

# ESC REST API

## ESC REST Northbound API Documentation

### Table of Contents

- [Resources managed by ESC REST](#)
- [Behavior of ESC REST operations](#)
- [ESC REST API Headers/Path/Body Parameters](#)
  - [Callbacks](#)
  - [Media Type](#)
  - [Internal IDs](#)
  - [Body](#)
  - [String Restrictions in Body/Internal IDs](#)
- [REST API Security](#)
  - [REST Authentication](#)
  - [REST Https Support](#)
- [REST API List](#)

### Resources managed by ESC REST

- Tenants
- Networks
- Subnets
- Flavors
- Images
- Volumes
- Routers
- Deployments

### Behavior of ESC REST operations

GET operations are synchronous the request/response parameters are described with an example of a request/response

POST, DELETE, and PUT operations are asynchronous. The asynchronous design use the “web hooks” approach using two independent "one-way" invocations - one to start a long-running operation (Client to ESC) and the other one to notify a requester that it is completed (ESC to client)

# ESC REST API Headers/Path/Body Parameters

## Callbacks

One of the header parameters of the operation request will contain a callback field, whose value is a URI where the client expects a callback. If this field is not present, no callback will be performed.

When the operation has completed (either successfully or unsuccessfully), ESC will perform a POST request to the callback URI, with an entity body containing the resource for this operation. The status will be returned in the HTTP header called `esc-status-code` and the status message will be returned in the HTTP header called `esc-status-message`.

ESC Clients can match a completion report back to the original request by comparing the value of the HTTP header called `esc-transaction-id` to the one returned in the initial status response.

ESC Clients are expected to provide a REST implementation for all the callbacks.

## Media Type

All POST, DELETE, and PUT operations that require request and/or response body must provide a header parameter for media type. Note, that ESC is currently tested to accept and return XML.

## Internal IDs

In some of the ESC REST API, there is a path parameter called `"*_internal_id"` where the asterisk is replaced by the resource type. ESC Clients must provide an internal id for its resource it wants to create. The internal ids are enforced so the ESC Clients are able to refer to them in future operations such as GET, PUT, or DELETE requests.

It is recommended that the ESC Client generated internal ids within the scope of the ESC.

NOTE: Reusing (recycling) of internal ids is not recommended. This practice can lead to confusion in trouble shooting.

## Body

In some of the POST, PUT, and DELETE requests you are required to provide a JSON/XML body. Depending on the resource, you will need to provide certain fields in the body to successfully execute a request. Check the body section for each API (bottom of page) to see what fields are supported.

## String Restrictions in Body/Internal IDs

Body  
Some string parameters/fields, such as `id`, `name`, are restricted by the underlying cloud provider, such as Openstack. The allowable range for these parameters is included in the Create section for each resource, in this notation: String minimum-length . . maximum length, for example, for Tenant name: String 1 . . 64

Generally, names are either 1..64 (for Tenant/User) or 1..255 (for Network/Subnet/Flavor/Image/VM).

Internal ID

Ids are restricted to 72 characters such as `internal_id 1..72` and `external_id 1..72`.

Internal id naming must conform the standards stated in RFC 3986 Uniform Resource Identifier (URI): Generic Syntax. See sections related to URI encoding, sections 2.2 Reserved Characters and 2.3 Unreserved Characters. ESC will accept an internal id specified in the request URL form the unreserved character set: unreserved = ALPHA / DIGIT / "-" / "." / "\_"

NOTE: the tilde “~” is not supported in the internal id character set.

The internal id received in a request must be unique within the scope of all the domain resources (tenant, network, subnet, service catalog, deployment ) For example, you cannot create 2 networks with the same internal\_network\_id, even if the networks are created under different tenants.

## REST API Security

### REST Authentication

#### Overview

ESC REST API uses http basic access authentication where the ESC client will have to provide a username and password when making ESC REST requests. The user name and password will be encoded with Base64 in transit, but not encrypted or hashed. HTTPS will be used in conjunction with Basic Authentication to provide the encryption.

#### Setup

By default, the REST authentication is disabled. To enable it pass the argument `--enable-https-rest` and/or `--enable-http-rest` and `--user_rest_pass admin:<password>` to ESC bootvm.py, ESC installation script.

#### Username and Password

The REST interface has only one default username/password (admin credentials).

The REST password can be updated using escadm tool from the ESC VM CLI:

```
$ escadm rest set --username admin --password <password>
```

You can also update the password through the REST API:

[http://\[ESC\\_VM\\_IP\]:8080/ESCManager/v0/authentication/setpassword?  
userName=yourUsername&password=yourPassword](http://[ESC_VM_IP]:8080/ESCManager/v0/authentication/setpassword?userName=yourUsername&password=yourPassword)

#### Sending an Authorized Request

To send an authorized request an ESC client should send the request with the following header:

```
Authorization: Basic <hashed-pass>
```

where is the Base64 encoded string of the default username/password

Most libraries and web clients have an interface for simply providing the username/password and the app will encode the username/password and add the HTTP Basic Auth header.

Example using the default admin credentials:

```
http://[ESC_VM_IP]:8080/ESCManager/v0/tenants/
```

Headers:

```
Content-Type    application/xml
Authorization   Basic <hashed-pass>
```

## REST Https Support

### Overview

ESC supports https communication over port 8443. ESC will generate a self-signed certificate that the client will need to trust to get the https communication going.

### Setup

By default, the REST https is disabled and restricted to localhost. To enable it pass the argument `--enable-https-rest` to ESC `bootvm.py`, ESC installation script.

### HA Considerations

In an HA setup every ESC HA VM will have its own self-signed generated certificate. Upon an HA switchover, ESC north bound client will be required to re-accept the new certificate it will be served.

## REST API List

### 01-Tenants : Manage ESC tenants

DELETE	/v0/tenants/{internal_tenant_id}	Delete a tenant from ESC		
<b>Implementation Notes</b> Delete a tenant from ESC				
<b>Parameters</b>				
Parameter	Value	Description	Parameter Type	Data Type
<b>callback</b>	<input type="text" value="(required)"/>	<b>Callback URL for ESC notifications</b>	header	string
<b>internal_tenant_id</b>	<input type="text" value="(required)"/>	<b>Internal tenant ID (max length 72 characters)</b>	path	string
<b>Response Messages</b>				
HTTP Status Code	Reason	Response Model	Headers	
200	Tenant deletion request accepted			

HTTP Status Code	Reason	Response Model	Headers
400	Invalid Request		
404	Invalid Tenant Id		

Try it out!

**GET** /v0/tenants/{internal\_tenant\_id} Get a tenant by its name

### Implementation Notes

Only tenants created using ESC will be checked

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
internal_tenant_id	<input type="text" value="(required)"/>	Internal tenant ID (max length 72 characters)	path	string

### Response Messages

HTTP Status Code	Reason	Response Model	Headers
200	Get request accepted		
404	Invalid Request - Invalid Tenant ID		

Try it out!

**POST** /v0/tenants/{internal\_tenant\_id} Create a tenant in ESC

### Implementation Notes

Creates a tenant which will be managed by ESC and can be used for creating other resources under it. Example request:

```
POST /v0/tenants/tentest HTTP/1.1
Content-Type: application/xml
Accept: application/xml
Callback: http://127.0.0.1:9010/
Body:
<tenant xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <name>example-tenant</name>
  <managed_resource>true</managed_resource>
</tenant>
```

After the tenant creation operation is done, ECS will send a callback to the client with the result. Example callback:

```
HTTP/1.1 201 OK
Content-Type: application/xml; charset=UTF-8
ESC_TRANSACTION_ID: d53f0047-2315-466c-b7e3-aa78e0a567e8
ESC-Status-Code: 200
ESC-Status-Message: Tenant successfully created
<tenant xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <enabled>true</enabled>
  <event_type>CREATE_TENANT</event_type>
  <external_tenant_id>484c620285ea4f588227ff3396215980</external_tenant_id>
  <internal_tenant_id>tentest</internal_tenant_id>
  <name>example-tenant</name>
  <managed_resource>true</managed_resource>
</tenant>
```

## Parameters

Parameter	Value	Description	Parameter Type	Data Type
-----------	-------	-------------	----------------	-----------

Parameter	Value	Description	Parameter Type	Data Type
<b>body</b>	<div style="border: 1px solid #ccc; padding: 5px; min-height: 80px;">(required)</div> <p>Parameter content type:  <input type="text" value="application/xml"/></p>	<b>Tenant configuration payload</b>	body	<div style="display: flex; border-bottom: 1px solid #ccc;"> <span style="border-right: 1px solid #ccc; padding: 2px 5px;">Model</span> <span style="padding: 2px 5px;">Model Schema</span> </div> <pre style="background-color: #fff9c4; padding: 10px;"> {   "internal_tenant_id": "string",   "external_tenant_id": "string",   "name": "string",   "enabled": true,   "vim_mapping": true,   "extensions": {     "extension": [       {         "name": "string",         "properties": {           "property": [             {               "name": "string",               "value": "string"             }           ]         }       }     ],     "containers": {       "container": [         {           "name": "string",           "properties": {             "property": [               {                 "name": "string",                 "value": "string"               }             ]           }         }       ]     }   } }, "event_type": "CREATE_TENANT", "managed_resource": true } </pre> <p style="font-size: small; text-align: right;">Click to set as parameter value</p>
<b>Callback</b>	<input type="text" value="(required)"/>	<b>Callback URL for ESC notifications</b>	header	string
<b>internal_tenant_id</b>	<input type="text" value="(required)"/>	<b>Internal tenant ID (max length 72 characters)</b>	path	string

## Response Messages

HTTP Status Code	Reason	Response Model	Headers
200	Tenant creation request accepted		
400	Invalid Request		
409	Conflicting Request		

Try it out!

**PUT** /v0/tenants/{internal\_tenant\_id} Update a tenant from ESC

### Implementation Notes

Update a tenant from ESC

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
-----------	-------	-------------	----------------	-----------



Parameter	Value	Description	Parameter Type	Data Type
<b>body</b>	<div style="border: 1px solid #ccc; padding: 5px; min-height: 80px;">(required)</div> <p>Parameter content type:  <input type="text" value="application/xml"/></p>	<b>Tenant payload</b>	body	<div style="display: flex; border-bottom: 1px solid #ccc;"> <span style="border-right: 1px solid #ccc; padding-right: 5px;">Model</span> <span>Model Schema</span> </div> <pre style="background-color: #fff9c4; padding: 10px;"> {   "internal_tenant_id": "string",   "external_tenant_id": "string",   "name": "string",   "enabled": true,   "vim_mapping": true,   "extensions": {     "extension": [       {         "name": "string",         "properties": {           "property": [             {               "name": "string",               "value": "string"             }           ]         }       }     ],     "containers": {       "container": [         {           "name": "string",           "properties": {             "property": [               {                 "name": "string",                 "value": "string"               }             ]           }         }       ]     }   ] }, "event_type": "CREATE_TENANT", "managed_resource": true } </pre> <p style="font-size: small; text-align: right;">Click to set as parameter value</p>
<b>Callback</b>	<div style="border: 1px solid #ccc; padding: 5px;">(required)</div>	<b>Callback URL for ESC notifications</b>	header	string

Parameter	Value	Description	Parameter Type	Data Type
<b>internal_tenant_id</b>	<input type="text" value="(required)"/>	<b>Internal tenant ID (max length 72 characters)</b>	path	string

### Response Messages

HTTP Status Code	Reason	Response Model	Headers
200	Tenant update request accepted		
400	Invalid Request		
404	Invalid Tenant Id		
500	Internal Server Error		

Try it out!

GET /v0/tenants

Get all tenants

### Implementation Notes

All tenants in ESC will be returned as a List

### Response Class (Status 200)

Model | Model Schema

```
[
  {
    "internal_tenant_id": "string",
    "external_tenant_id": "string",
    "name": "string",
    "enabled": true,
    "vim_mapping": true,
    "extensions": {
      "extension": [
        {
          "name": "string",
          "properties": {
            "property": [
              {
                "name": "string",
                "value": "string"
              }
            ]
          }
        }
      ]
    }
  }
],
```

```

"containers": {
  "container": [
    {
      "name": "string",
      "properties": {
        "property": [
          {
            "name": "string",
            "value": "string"
          }
        ]
      }
    }
  ]
}
},
"event_type": "CREATE_TENANT",
"managed_resource": true
}
]

```

Response Content Type  ▼

Try it out!

## 02-Networks : Manage ESC networks

**DELETE**

/v0/{internalTenantId}/networks/{internalNetworkId}

Delete a network from an existing tenant in ESC

### Implementation Notes

Delete a network from ESC

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
<b>Callback</b>	<input type="text" value="(required)"/>	<b>Callback URL for ESC notifications</b>	header	string
<b>internalNetworkId</b>	<input type="text" value="(required)"/>	<b>Internal network ID (max length 72 characters)</b>	path	string

Parameter	Value	Description	Parameter Type	Data Type
<b>internalTenantId</b>	<input type="text" value="(required)"/>	<b>Internal tenant ID of the network (max length 72 characters)</b>	path	string

### Response Messages

HTTP Status Code	Reason	Response Model	Headers
200	Request accepted		
400	Invalid request		
500	Internal server error		

[Try it out!](#)

**GET** `/v0/{internalTenantId}/networks/{internalNetworkId}` [Get a network by its tenant and internal network ID](#)

### Implementation Notes

Only networks created using ESC will be checked

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
<b>internalTenantId</b>	<input type="text" value="(required)"/>	<b>Internal tenant ID of the network (max length 72 characters)</b>	path	string
<b>internalNetworkId</b>	<input type="text" value="(required)"/>	<b>Internal network ID (max length 72 characters)</b>	path	string

### Response Messages

HTTP Status Code	Reason	Response Model	Headers
200	Request accepted		
404	Network Id not found		

[Try it out!](#)

**POST** `/v0/{internalTenantId}/networks/{internalNetworkId}` [Create a network under an existing tenant in ESC](#)

### Implementation Notes

## Create a network in ESC

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
<b>Callback</b>	<input type="text" value="(required)"/>	<b>Callback URL for ESC notifications</b>	header	string
<b>internalTenantId</b>	<input type="text" value="(required)"/>	<b>Internal tenant ID of the network (max length 72 characters)</b>	path	string
<b>internalNetworkId</b>	<input type="text" value="(required)"/>	<b>Internal network ID (max length 72 characters)</b>	path	string

Parameter	Value	Description	Parameter Type	Data Type
<b>body</b>	<div style="border: 1px solid #ccc; padding: 5px; min-height: 80px;">(required)</div> <p>Parameter content type:  <input type="text" value="application/xml"/></p>	<b>Network configuration payload</b>	body	<div style="border: 1px solid #ccc; padding: 5px;"> <div style="display: flex; justify-content: space-between;"> <span>Model</span> <span>Model Schema</span> </div> <pre style="background-color: #fff9c4; padding: 10px;"> {   "locator": {     "vim_id": "string",     "vim_type": "OPENSTACK",     "vim_project": "string",     "vim_project_id": "string",     "vim_region": "string",     "vim_vdc": "string"   },   "vimNetName": "string",   "internal_tenant_id": "string",   "internal_network_id": "string",   "internal_deployment_id": "string",   "name": "string",   "shared": true,   "external_tenant_id": "string",   "tenant_name": "string",   "external_network_id": "string",   "subnet": [     "string"   ],   "admin_state_up": true,   "locators": {     "switch_name": "string",     "datacenter": "string"   },   "event_type": "CREATE_NETWORK",   "switch_name": "string",   "router_external": true,   "provider_physical_network": "string",   "provider_network_type": "string",   "vlan_id": 0,   "number_of_ports": 0,   "provider_segmentation_id": "string" } </pre> </div>

Click to set as parameter value

### Response Messages

HTTP Status Code	Reason	Response Model	Headers
200	Request accepted		
400	Invalid request		
404	Network Id not found		
409	Conflicting request		
503	Unable to query VIM		

Try it out!

GET /v0/{internalTenantId}/networks Get all networks under a tenant

### Implementation Notes

All networks under that tenant in ESC will be returned as a List

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
<b>internalTenantId</b>	<input type="text" value="(required)"/>	<b>Internal tenant ID of the network (max length 72 characters)</b>	path	string

### Response Messages

HTTP Status Code	Reason	Response Model	Headers
200	Request accepted		

Try it out!

## 03-Subnets : Manage ESC subnets

DELETE /v0/{internalTenantId}/subnets/{internalSubnetId} Delete a subnet from an existing tenant in ESC

### Implementation Notes

Delete a subnet from ESC

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
<b>callback</b>	<input type="text" value="(required)"/>	<b>Callback URL for ESC notifications</b>	header	string
<b>internalTenantId</b>	<input type="text" value="(required)"/>	<b>Internal tenant ID of the subnet (max length 72 characters)</b>	path	string
<b>internalSubnetId</b>	<input type="text" value="(required)"/>	<b>Internal subnet ID (max length 72 characters)</b>	path	string

### Response Messages

HTTP Status Code	Reason	Response Model	Headers
200	Request accepted		
400	Invalid Request		
404	Not found		
500	Internal server error		

Try it out!

**GET** /v0/{internalTenantId}/subnets/{internalSubnetId} [Get a subnet by its tenant and internal subnet ID](#)

### Implementation Notes

Only subnets created using ESC will be checked

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
internalTenantId	<input type="text" value="(required)"/>	Internal tenant ID of the subnet (max length 72 characters)	path	string
internalSubnetId	<input type="text" value="(required)"/>	Internal subnet ID (max length 72 characters)	path	string

### Response Messages

HTTP Status Code	Reason	Response Model	Headers
200	Request accepted		
404	Not found		

Try it out!

**POST** /v0/{internalTenantId}/subnets/{internalSubnetId} [Create a subnet under an existing tenant in ESC](#)

### Implementation Notes

Create a subnet in ESC

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
-----------	-------	-------------	----------------	-----------



Parameter	Value	Description	Parameter Type	Data Type
Callback	<input type="text" value="(required)"/>	Callback URL for ESC notifications	header	string
internalTenantId	<input type="text" value="(required)"/>	Internal tenant ID of the subnet (max length 72 characters)	path	string
internalSubnetId	<input type="text" value="(required)"/>	Internal subnet ID (max length 72 characters)	path	string

body

Parameter content type:

Subnet configuration payload

body

Model | Model Schema

```

{
  "networkFromCdg": {
    "locator": {
      "vim_id": "string",
      "vim_type": "OPENSTACK",
      "vim_project": "string",
      "vim_project_id": "string",
      "vim_region": "string",
      "vim_vdc": "string"
    },
    "vimNetName": "string",
    "internal_tenant_id": "string",
    "internal_network_id": "string",
    "internal_deployment_id": "string",
    "name": "string",
    "shared": true,
    "external_tenant_id": "string",
    "tenant_name": "string",
    "external_network_id": "string",
    "subnet": [
      "string"
    ],
    "admin_state_up": true,
    "locators": {
      "switch_name": "string",
      "datacenter": "string"
    },
    "event_type": "CREATE_NETWORK",
    "switch_name": "string",
    "router_external": true,
    "provider_physical_network": "string",
    "provider_network_type": "string",
  }
}

```

Parameter	Value	Description	Parameter Type	Data Type
				<pre> "vlan_id": 0, "number_of_ports": 0, "provider_segmentation_id": "string" }, "internal_tenant_id": "string", "external_tenant_id": "string", "internal_network_id": "string", "external_network_id": "string", "internal_subnet_id": "string", "external_subnet_id": "string", "name": "string", "allocation_pools": [ { "start": "string", "end": "string" } ], "gateway_ip": "string", "ip_version": 0, "cidr": "string", "enable_dhcp": true, "event_type": "CREATE_SUBNET" } </pre>

Click to set as parameter value

## Response Messages

HTTP Status Code	Reason	Response Model	Headers
200	Request accepted		
400	Invalid Request		
404	Not found		
409	Conflicting Request		
500	Internal server error		

Try it out!

GET
/v0/{internalTenantId}/subnets
Get all subnets under a network

### Implementation Notes

All subnet under that network in ESC will be returned as a List

### Parameters

Parameter	Value	Description	Parameter Type	Data Type

Parameter	Value	Description	Parameter Type	Data Type
internalNetworkId	<input type="text"/>	Internal network ID of the subnet (max length 72 characters)	header	string
internalTenantId	<input type="text" value="(required)"/>	<b>Internal tenant ID of the subnet (max length 72 characters)</b>	path	string

**Response Messages**

HTTP Status Code	Reason	Response Model	Headers
200	Request accepted		
404	Not found		

[Try it out!](#)

## 04-Images : Manage ESC images

DELETE	/v0/images/{internalImageId}	Delete an image from ESC		
<b>Implementation Notes</b>				
Delete an image from ESC				
<b>Parameters</b>				
Parameter	Value	Description	Parameter Type	Data Type
callback	<input type="text" value="(required)"/>	<b>Callback URL for ESC notifications</b>	header	string
internalImageId	<input type="text" value="(required)"/>	<b>Internal image ID (max length 72 characters)</b>	path	string
<b>Response Messages</b>				
HTTP Status Code	Reason	Response Model	Headers	
200	Request accepted			
400	Invalid Request			
500	Internal server error			
<a href="#">Try it out!</a>				

### Implementation Notes

Images not created by ESC will not be checked

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
internalImageId	<input type="text" value="(required)"/>	Internal image ID (max length 72 characters)	path	string

### Response Messages

HTTP Status Code	Reason	Response Model	Headers
200	Request accepted		
400	Invalid Request		
404	Image Id not found		

[Try it out!](#)

### Implementation Notes

Creates an Image(Template) which will be managed by ESC and can be used for creating instance VMs. An optional properties list can be included in the image creation payload. This is only supported for creation of images on openstack. Image properties will be stored in openstack as image metadata. Property name and values are not validated by ESC.

Example 1 create image request (without properties):

```
POST /v0/images/my_image_id HTTP/1.1
Content-Type: application/xml
Accept: application/xml
Callback: http://127.0.0.1:9010/
Body:

<?xml version="1.0"?>
<image xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <name>uLinuxImg</name>
  <src>http://VAR_FILE_SERVER_IP/share/images/myULinux.qcow2</src>
  <disk_bus>virtio</disk_bus>
</image>
```

## Example 2 create image request with properties:

```
POST /v0/images/my_image_id HTTP/1.1
Content-Type: application/xml
Accept: application/xml
Callback: http://127.0.0.1:9010/
Body:

<?xml version="1.0"?>
<image xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <name>uLinuxImg</name>
  <src>http://VAR_FILE_SERVER_IP/share/images/myULinux.qcow2</src>
  <disk_bus>virtio</disk_bus>
  <properties>
    <property>
      <name>os_secure_boot</name>
      <value>enabled</value>
    </property>
    <property>
      <name>hw_rng_model</name>
      <value>virtio</value>
    </property>
    <property>
      <name>hw_vif_multiqueue_enabled</name>
      <value>>true</value>
    </property>
  </properties>
</image>
```

Example 3 create image with properties Use of image properties allow flexible assignment of image metadata, such as hw\_disk\_bus and hw\_vif\_model. This extends current limitation of ESC schema with disk\_bus := ide | scsi | virtio and e1000\_net := true | false and virtio\_net := true | false.

```
POST /v0/images/my_image_id HTTP/1.1
Content-Type: application/xml
Accept: application/xml
Callback: http://127.0.0.1:9010/
Body:

<?xml version="1.0"?>
<image xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <name>uLinuxImg</name>
  <src>http://VAR_FILE_SERVER_IP/share/images/myULinux.qcow2</src>
  <disk_bus>virtio</disk_bus>
  <properties>
    <property>
      <name>hw_disk_bus</name>
      <value>pcnet</value>
    </property>
```

```

    <property>
      <name>hw_vif_model</name>
      <value>pcnet</value>
    </property>
  </properties>
</image>

```

After the Image creation operation is done, ESC will send a callback to the client with the result.

Example callback:

```


HTTP/1.1 201 OK
Content-Type: application/xml; charset=UTF-8
ESC_TRANSACTION_ID: d53f0047-2315-466c-b7e3-aa78e0a567e8
ESC-Status-Code: 200
ESC-Status-Message: Image successfully created

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<image xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <disk_bus>virtio</disk_bus>
  <e1000_net>false</e1000_net>
  <event_type>CREATE_IMAGE</event_type>
  <external_image_id>5021843b-fcc5-2c1b-8fc4-935147f95872</external_image_id>
  <external_tenant_id>SystemAdminTenantId</external_tenant_id>
  <imageisenabled>>true</imageisenabled>
  <name>uLinuxImg</name>
  <internal_image_id>my_image_id</internal_image_id>
  <internal_tenant_id>SystemAdminTenantId</internal_tenant_id>
  <visibility>public</visibility>
  <src>http://VAR_FILE_SERVER_IP/share/images/myULinux.qcow2</src>
</image>

```

## Parameters

Parameter	Value	Description	Parameter Type	Data Type
<b>callback</b>	<input type="text" value="(required)"/>	<b>Callback URL for ESC notifications</b>	header	string
<b>internalImageId</b>	<input type="text" value="(required)"/>	<b>Internal image ID (max length 72 characters)</b>	path	string

Parameter	Value	Description	Parameter Type	Data Type
<b>body</b>	(required)  Parameter content type: application/xml ▼	<b>ESC datamodel with the image to create</b>	body	string

### Response Messages

HTTP Status Code	Reason	Response Model	Headers
200	Request accepted		
400	Invalid Request		
409	Conflicting Request		
500	Internal server error		

[Try it out!](#)

GET	/v0/images	Get all images	
<b>Implementation Notes</b>			
All images created in ESC will be returned as a List			
<b>Response Messages</b>			
HTTP Status Code	Reason	Response Model	Headers
200	Request accepted		

[Try it out!](#)

## 05-Flavors : Manage ESC flavors

DELETE	/v0/flavors/{internalFlavorId}	Delete a flavor from ESC		
<b>Implementation Notes</b>				
Delete a flavor from ESC				
<b>Parameters</b>				
Parameter	Value	Description	Parameter Type	Data Type

Parameter	Value	Description	Parameter Type	Data Type
<b>Callback</b>	<input type="text" value="(required)"/>	<b>Callback URL for ESC notifications</b>	header	string
<b>internalFlavorId</b>	<input type="text" value="(required)"/>	<b>Internal flavor ID (max length 72 characters)</b>	path	string

**Response Messages**

HTTP Status Code	Reason	Response Model	Headers
200	Request accepted		
404	Flavor Id not found		
500	Internal server error		

**GET** /v0/flavors/{internalFlavorId} Get the flavor by the internal flavor ID

**Implementation Notes**  
Flavors not created by ESC will not be checked

**Parameters**

Parameter	Value	Description	Parameter Type	Data Type
<b>internalFlavorId</b>	<input type="text" value="(required)"/>	<b>Internal flavor ID (max length 72 characters)</b>	path	string

**Response Messages**

HTTP Status Code	Reason	Response Model	Headers
200	Request accepted		
404	Flavor Id not found		
406	Operation not supported on VIM/Flavor part of service registration		

**POST** /v0/flavors/{internalFlavorId} Create a flavor in ESC



## Implementation Notes

Creates a Flavor which could be used for creating instance VMs.

Example request:

```
POST /v0/flavors/my_flavor_id HTTP/1.1
Content-Type: application/xml
Accept: application/xml
Callback: http://127.0.0.1:9010/
Body:
<?xml version="1.0"?>
<flavor xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <name>lightVMFlavor</name>
  <vcpus>2</vcpus>
  <memory_mb>2048</memory_mb>
</flavor>
```

After the Flavor creation operation is done, ESC will send a callback to the client with the result.

Example callback:

```
HTTP/1.1 201 OK
Content-Type: application/xml; charset=UTF-8
ESC_TRANSACTION_ID: d53f0047-2315-466c-b7e3-aa78e0a567e8
ESC-Status-Code: 200
ESC-Status-Message: Flavor creation completed successfully created
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<flavor xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <ephemeral_disk_mb>0</ephemeral_disk_mb>
  <event_type>CREATE_FLAVOR</event_type>
  <external_flavor_id>fb63c114-de1a-40d8-abae-fa6011abd150</external_flavor_id>
  <name>lightVMFlavor</name>
  <internal_flavor_id>my_flavor_id</internal_flavor_id>
  <memory_mb>2048</memory_mb>
  <vcpus>2</vcpus>
</flavor>
```

## Parameters

Parameter	Value	Description	Parameter Type	Data Type
<b>Callback</b>	<input type="text" value="(required)"/>	<b>Callback URL for ESC notifications</b>	header	string
<b>internalFlavorId</b>	<input type="text" value="(required)"/>	<b>Internal flavor ID (max length 72 characters)</b>	path	string

Parameter	Value	Description	Parameter Type	Data Type
<b>body</b>	<div style="border: 1px solid black; padding: 5px; min-height: 80px;">(required)</div> <p>Parameter content type:  <input type="text" value="application/xml"/></p>	<b>ESC datamodel with the flavor to create</b>	body	string

### Response Messages

HTTP Status Code	Reason	Response Model	Headers
200	Request accepted		
400	Invalid Request		
404	Flavor Id not found		
406	Operation not supported on VIM/Flavor part of service registration		
409	Conflicting Request		
500	Internal server error		

[Try it out!](#)

GET	/v0/flavors	Get all flavors	
<b>Implementation Notes</b>			
All flavors created in ESC will be returned as a List			
<b>Response Messages</b>			
HTTP Status Code	Reason	Response Model	Headers
200	Request accepted		

[Try it out!](#)

## 06-Deployments : Manage ESC deployments

GET	/v0/deployments/config/{internal_deployment_id}	Find deployment configuration data by its internal ID
<b>Implementation Notes</b>		

Deployment configuration data will be returned if found in CDB

## Parameters

Parameter	Value	Description	Parameter Type	Data Type
<b>internal_deployment_id</b>	<input type="text" value="(required)"/>	<b>Internal ID of the deployment (max length 1024 characters)</b>	path	string

## Response Messages

HTTP Status Code	Reason	Response Model	Headers
200	Request accepted		
404	Deployment Id not found		

Try it out!

**DELETE** /v0/deployments/{internal\_deployment\_id}

Delete a deployment from ESC

## Implementation Notes

Delete a deployment from ESC

## Parameters

Parameter	Value	Description	Parameter Type	Data Type
<b>Callback</b>	<input type="text" value="(required)"/>	<b>Callback URL for ESC notifications</b>	header	string
<b>internal_deployment_id</b>	<input type="text" value="(required)"/>	<b>Internal ID of the deployment (max length 72 characters)</b>	path	string

## Response Messages

HTTP Status Code	Reason	Response Model	Headers
200	Request accepted		
400	Invalid Request		
404	Deployment Id not found		

Try it out!

### Implementation Notes

A deployment will be returned if found in ESC

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
Accept	<input type="text"/>		header	string
internal_deployment_id	<input type="text" value="(required)"/>	Internal ID of the deployment (max length 1024 characters)	path	string

### Response Messages

HTTP Status Code	Reason	Response Model	Headers
200	Request accepted		
404	Deployment Id not found		

Try it out!

### Implementation Notes

Creates a Simple deployment of Instance VMs.

**NOTE:** As deployment requirements vary a lot, please see ESC User guide for datamodel excerpts of specific functionality you may desire that's not displayed in the simple example below.

Example request:

```
POST /v0/deployments/my_dep_id HTTP/1.1
Content-Type: application/xml
Accept: application/xml
Callback: http://127.0.0.1:9010/
Callback-ESC-Events: http://127.0.0.1:9010/

Body:

<esc_datamodel xmlns="http://www.cisco.com/esc/esc" xmlns:ns2="urn:ietf:params:xml:ns:netconf:notification:1"
  <tenants>
    <tenant>
      <name>admin</name>
      <deployments>
```

```
<deployment>
  <name>MyDemoDepName</name>
  <vm_group>
    <name>ASA-1</name>
    <bootup_time>300</bootup_time>
    <reboot_time>100</reboot_time>
    <recovery_wait_time>10</recovery_wait_time>
    <image>ASA</image>
    <interfaces>
      <interface>
        <nicid>1</nicid>
        <network>MgtNetwork</network>
        <ip_address>18.0.0.16</ip_address>
      </interface>
    </interfaces>
    <scaling>
      <min_active>1</min_active>
      <max_active>1</max_active>
      <elastic>true</elastic>
      <static_ip_address_pool>
        <network>MgtNetwork</network>
        <ip_address>18.0.0.16</ip_address>
      </static_ip_address_pool>
    </scaling>
    <kpi_data>
      <kpi>
        <event_name>VM_ALIVE</event_name>
        <metric_value>1</metric_value>
        <metric_cond>GT</metric_cond>
        <metric_type>UINT32</metric_type>
        <metric_collector>
          <type>ICMPPing</type>
          <nicid>1</nicid>
          <poll_frequency>3</poll_frequency>
          <polling_unit>seconds</polling_unit>
          <continuous_alarm>>false</continuous_alarm>
        </metric_collector></kpi>
    </kpi_data>
    <rules>
      <admin_rules>
        <rule>
          <event_name>VM_ALIVE</event_name>
          <action>"ALWAYS log"</action>
          <action>"TRUE esc_vm_alive_notification"</action>
        </rule>
      </admin_rules>
    </rules>
    <config_data>
      <configuration>
        <dst>day0-config</dst>
        <file>http://MY_FILE_SERVER_IP:/share/images/day0-configASAv</file>
      </configuration>
    </config_data>
  </vm_group>
</deployment>
```

```

    </configuration>
    <configuration>
      <dst>idtoken</dst>
      <file>http://MY_FILE_SERVER_IP:/share/images/asa-smart.config</file>
    </configuration>
  </config_data>
  <placement>
    <type>zone_host</type>
    <zone>MY_CLUSTER1</zone>
  </placement>
  <volumes>
    <volume>
      <name>MY_DATASTORE1</name>
      <valid>1</valid>
    </volume>
  </volumes>
</vm_group>
</deployment>
</deployments>
</tenant>
</tenants>
</esc_datamodel>

```

While the Deployment operation is being processed successfully, ESC will send a VM\_DEPLOYED callback for each VM created on the VIM as part of this deployment

```

HTTP/1.1 201 OK
Content-Type: application/xml; charset=UTF-8
ESC_TRANSACTION_ID: d53f0047-2315-466c-b7e3-aa78e0a567e8
ESC-Status-Code: 200
ESC-Status-Message: VM Deployed

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<deployment xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <datacenter>
    <default>>false</default>
  </datacenter>
  <deployment_details>
    <host_uuid>host-14882</host_uuid>
    <host_name>10.85.103.13</host_name>
    <vm_uuid>50217d2d-8afb-41c2-cffd-d76b1239f51d</vm_uuid>
    <interfaces>
      <interface>
        <network_uuid>MgtNetwork</network_uuid>
        <ip_address>18.0.0.16</ip_address>
        <mac_address>00:50:56:a1:6a:12</mac_address>
        <nic_id>1</nic_id>
        <port_forwarding></port_forwarding>
        <port_uuid></port_uuid>
        <security_groups></security_groups>
      </interface>
    </interfaces>
  </deployment_details>
</deployment>

```

```

    <subnet_uuid></subnet_uuid>
    <type>virtual</type>
</interface>
</interfaces>
<vm_group_name>ASA-1</vm_group_name>
<vm_name>Sanity-vmware-de_ASA-1_0_b323effd-7f70-4055-b11e-0ef66f0a758d</vm_name>
<vm_state_machine_state>VM_INERT_STATE</vm_state_machine_state>
</deployment_details>
<deployment_name>MyDemoDepName</deployment_name>
<deployment_stage>VM_DEPLOYED</deployment_stage>
<external_deployment_id>8d084aab-6a66-4a70-8f47-42b91f7477ac</external_deployment_id>
<external_tenant_id>SystemAdminTenantId</external_tenant_id>
<internal_deployment_id>my_dep_id</internal_deployment_id>
<internal_tenant_id>SystemAdminTenantId</internal_tenant_id>
<service_state_machine_state>SERVICE_DEPLOYING_STATE</service_state_machine_state>
</deployment>

```

While the Deployment operation is being processed successfully, ESC will send a VM\_ALIVE callback for each monitorable/pingable VM created on the VIM as part of this deployment

```

HTTP/1.1 201 OK
Content-Type: application/xml; charset=UTF-8
ESC_TRANSACTION_ID: d53f0047-2315-466c-b7e3-aa78e0a567e8
ESC-Status-Code: 200
ESC-Status-Message: VM_Alive event received, VM ID: [Sanity-vmware-de_ASA-1_0_b323effd-7f70-4055-b11e-0ef66f0a758d]

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<deployment xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <datacenter>
    <default>false</default>
  </datacenter>
  <deployment_details>
    <host_uuid>host-14882</host_uuid>
    <host_name>10.85.103.13</host_name>
    <vm_uuid>50217d2d-8afb-41c2-cffd-d76b1239f51d</vm_uuid>
    <interfaces>
      <interface>
        <network_uuid>MgtNetwork</network_uuid>
        <ip_address>18.0.0.16</ip_address>
        <mac_address>00:50:56:a1:6a:12</mac_address>
        <nic_id>1</nic_id>
        <port_forwarding></port_forwarding>
        <port_uuid></port_uuid>
        <security_groups></security_groups>
        <subnet_uuid></subnet_uuid>
        <type>virtual</type>
      </interface>
    </interfaces>
    <vm_group_name>ASA-1</vm_group_name>
    <vm_name>Sanity-vmware-de_ASA-1_0_b323effd-7f70-4055-b11e-0ef66f0a758d</vm_name>

```

```

    <vm_state_machine_state>VM_ALIVE_STATE</vm_state_machine_state>
  </deployment_details>
  <deployment_name>MyDemoDepName</deployment_name>
  <deployment_stage>VM_ALIVE</deployment_stage>
  <external_deployment_id>8d084aab-6a66-4a70-8f47-42b91f7477ac</external_deployment_id>
  <external_tenant_id>SystemAdminTenantId</external_tenant_id>
  <internal_deployment_id>my_dep_id</internal_deployment_id>
  <internal_tenant_id>SystemAdminTenantId</internal_tenant_id>
  <service_state_machine_state>SERVICE_INERT_STATE</service_state_machine_state>
</deployment>

```

After the Deployment operation is done, ESC will send a SERVICE\_ALIVE callback to the client with the result.

Example callback:

```

HTTP/1.1 201 OK
Content-Type: application/xml; charset=UTF-8
ESC_TRANSACTION_ID: d53f0047-2315-466c-b7e3-aa78e0a567e8
ESC-Status-Code: 200
ESC-Status-Message: Service group deployment completed successfully!

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<deployment xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <datacenter>
    <default>false</default>
  </datacenter>
  <deployment_name>MyDemoDepName</deployment_name>
  <deployment_stage>SERVICE_ALIVE</deployment_stage>
  <external_deployment_id>8d084aab-6a66-4a70-8f47-42b91f7477ac</external_deployment_id>
  <external_tenant_id>SystemAdminTenantId</external_tenant_id>
  <internal_deployment_id>my_dep_id</internal_deployment_id>
  <internal_tenant_id>SystemAdminTenantId</internal_tenant_id>
  <service_state_machine_state>SERVICE_ACTIVE_STATE</service_state_machine_state>
</deployment>

```

## Parameters

Parameter	Value	Description	Parameter Type	Data Type
<b>Callback</b>	<input type="text" value="(required)"/>	<b>Callback URL for ESC notifications</b>	header	string
<b>Callback-ESC-Events</b>	<input type="text" value="(required)"/>	<b>Callback URL for ESC event notifications</b>	header	string



Parameter	Value	Description	Parameter Type	Data Type
<b>internal_deployment_id</b>	<input type="text" value="(required)"/>	<b>Internal ID of the deployment (max length 72 characters)</b>	path	string
<b>body</b>	<input type="text" value="(required)"/> <div style="border: 1px solid #ccc; width: 100px; height: 80px; margin: 5px 0;"></div> Parameter content type: <input type="text" value="application/xml"/>	<b>ESC datamodel to deploy</b>	body	string

### Response Messages

HTTP Status Code	Reason	Response Model	Headers
200	Request accepted		
400	Invalid Request		
404	An associated resource not found		
406	Not Acceptable		
409	Conflicting Request		
500	Internal server error		
503	Failed to query VIM		

[Try it out!](#)

**PUT** `/v0/deployments/{internal_deployment_id}` Update an existing deployment in ESC

### Implementation Notes

Updates an existing deployment. Updates could be addition or removal of VM groups, interfaces, networks, KPIs and Rules, Day 0 config.

After the service update operation is done, ESC will send a callback to the client with the result.

Example request of adding VM group:

```

PUT /v0/deployments/my_dep_id HTTP/1.1
Content-Type: application/xml
Accept: application/xml
Callback: http://127.0.0.1:9010/
Callback-ESC-Events: http://127.0.0.1:9010/

```

Body:

```
<esc_datamodel xmlns:ns2="urn:ietf:params:xml:ns:netconf:notification:1.0" xmlns="http://www.cisco.com/e
<tenants>
<tenant>
<name>admin</name>
<deployments>
<deployment>
<name>MyDemoDepName</name>
<vm_group>
<name>ASA-1</name>
<bootup_time>300</bootup_time>
<recovery_wait_time>0</recovery_wait_time>
<image>ASA</image>
<interfaces>
<interface>
<nicid>1</nicid>
<network>MgtNetwork</network>
<ip_address>18.0.0.16</ip_address>
</interface>
</interfaces>
<scaling>
<min_active>1</min_active>
<max_active>1</max_active>
<elastic>true</elastic>
<static_ip_address_pool>
<network>MgtNetwork</network>
<ip_address>18.0.0.16</ip_address>
</static_ip_address_pool>
</scaling>
<kpi_data>
<kpi>
<event_name>VM_ALIVE</event_name>
<metric_value>1</metric_value>
<metric_cond>GT</metric_cond>
<metric_type>UIN32</metric_type>
<metric_collector>
<type>ICMPPing</type>
<nicid>1</nicid>
<poll_frequency>3</poll_frequency>
<polling_unit>seconds</polling_unit>
<continuous_alarm>false</continuous_alarm>
</metric_collector></kpi>
</kpi_data>
<rules>
<admin_rules>
<rule>
<event_name>VM_ALIVE</event_name>
<action>"ALWAYS log"</action>
<action>"TRUE esc_vm_alive_notification"</action>
```

```
</rule>
</admin_rules>
</rules>
<config_data>
  <configuration>
    <dst>day0-config</dst>
    <file>http://MY_FILE_SERVER_IP:/share/images/day0-configASAv</file>
  </configuration>
  <configuration>
    <dst>idtoken</dst>
    <file>http://MY_FILE_SERVER_IP:/share/images/asa-smart.config</file>
  </configuration>
</config_data>
<placement>
  <type>zone_host</type>
  <zone>MY_CLUSTER1</zone>
</placement>
<volumes>
  <volume>
    <name>MY_DATASTORE1</name>
    <valid>1</valid>
  </volume>
</volumes>
</vm_group>
<name>CSR-1</name>
<bootup_time>300</bootup_time>
<recovery_wait_time>0</recovery_wait_time>
<image>ASA</image>
<interfaces>
  <interface>
    <nicid>1</nicid>
    <network>MgtNetwork</network>
    <ip_address>18.0.0.16</ip_address>
  </interface>
</interfaces>
<scaling>
  <min_active>1</min_active>
  <max_active>1</max_active>
  <elastic>true</elastic>
  <static_ip_address_pool>
    <network>MgtNetwork</network>
    <ip_address>18.0.0.16</ip_address>
  </static_ip_address_pool>
</scaling>
<kpi_data>
  <kpi>
    <event_name>VM_ALIVE</event_name>
    <metric_value>1</metric_value>
    <metric_cond>GT</metric_cond>
    <metric_type>UINT32</metric_type>
    <metric_collector>
```

```

<type>ICMPPing</type>
<nicid>1</nicid>
<poll_frequency>3</poll_frequency>
<polling_unit>seconds</polling_unit>
<continuous_alarm>>false</continuous_alarm>
</metric_collector></kpi>
</kpi_data>
<rules>
  <admin_rules>
    <rule>
      <event_name>VM_ALIVE</event_name>
      <action>"ALWAYS log"</action>
      <action>"TRUE esc_vm_alive_notification"</action>
    </rule>
  </admin_rules>
</rules>
<config_data>
  <configuration>
    <dst>day0-config</dst>
    <file>http://MY_FILE_SERVER_IP:/share/images/day0-configASAv</file>
  </configuration>
  <configuration>
    <dst>idtoken</dst>
    <file>http://MY_FILE_SERVER_IP:/share/images/asa-smart.config</file>
  </configuration>
</config_data>
<placement>
  <type>zone_host</type>
  <zone>MY_CLUSTER1</zone>
</placement>
<volumes>
  <volume>
    <name>MY_DATASTORE1</name>
    <valid>1</valid>
  </volume>
</volumes>

</deployment>
</deployments>
</tenant>
</tenants>
</esc_datamodel>

```

While the Deployment Update operation is being processed successfully, ESC will send a VM\_DEPLOYED callback for each VM created on the VIM as part of this update request.

```

HTTP/1.1 201 OK
Content-Type: application/xml; charset=UTF-8
ESC_TRANSACTION_ID: d53f0047-2315-466c-b7e3-aa78e0a567e8
ESC-Status-Code: 200

```

ESC-Status-Message: VM Deployed

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<deployment xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
<datacenter><default>false</default></datacenter>
<deployment_details>
<host_uuid>host-14882</host_uuid>
<host_name>10.85.103.13</host_name>
<vm_uuid>50217d2d-8afb-41c2-cffd-d76b1239f51e</vm_uuid>
<interfaces>
<interface>
<network_uuid>MgtNetwork</network_uuid>
<ip_address>18.0.0.17</ip_address>
<mac_address>00:50:56:a1:6a:13</mac_address>
<nic_id>1</nic_id>
<port_forwarding></port_forwarding>
<port_uuid></port_uuid>
<security_groups></security_groups>
<subnet_uuid></subnet_uuid>
<type>virtual</type>
</interface>
</interfaces>
<vm_group_name>CSR-1</vm_group_name>
<vm_name>Sanity-vmware-de_CSR-1_0_b323effd-7f70-4055-b11e-0ef66f0a758d</vm_name>
<vm_state_machine_state>VM_INERT_STATE</vm_state_machine_state>
</deployment_details>
<deployment_name>MyDemoDepName</deployment_name>
<deployment_stage>VM_DEPLOYED</deployment_stage>
<external_deployment_id>8d084aab-6a66-4a70-8f47-42b91f7477ad</external_deployment_id>
<external_tenant_id>SystemAdminTenantId</external_tenant_id>
<internal_deployment_id>my_dep_id</internal_deployment_id>
<internal_tenant_id>SystemAdminTenantId</internal_tenant_id>
<service_state_machine_state>SERVICE_DEPLOYING_STATE</service_state_machine_state>
</deployment>
```

While the Deployment Update operation is being processed successfully, ESC will send a VM\_ALIVE callback for each monitorable/pingable VM on the VIM created as part of this update request.

HTTP/1.1 201 OK

Content-Type: application/xml; charset=UTF-8

ESC\_TRANSACTION\_ID: d53f0047-2315-466c-b7e3-aa78e0a567e8

ESC-Status-Code: 200

ESC-Status-Message: VM\_Alive event received, VM ID: [Sanity-vmware-de\_CSR-1\_0\_b323effd-7f70-4055-b11e-0ef66f

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<deployment xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
<datacenter><default>false</default></datacenter>
<deployment_details>
<host_uuid>host-14882</host_uuid>
<host_name>10.85.103.13</host_name>
```

```

<vm_uuid>50217d2d-8afb-41c2-cffd-d76b1239f51e</vm_uuid>
<interfaces>
<interface>
<network_uuid>MgtNetwork</network_uuid>
<ip_address>18.0.0.17</ip_address>
<mac_address>00:50:56:a1:6a:13</mac_address>
<nic_id>1</nic_id>
<port_forwarding></port_forwarding>
<port_uuid></port_uuid>
<security_groups></security_groups>
<subnet_uuid></subnet_uuid>
<type>virtual</type>
</interface>
</interfaces>
<vm_group_name>CSR-1</vm_group_name>
<vm_name>Sanity-vmware-de_CSR-1_0_b323effd-7f70-4055-b11e-0ef66f0a758d</vm_name>
<vm_state_machine_state>VM_ALIVE_STATE</vm_state_machine_state>
</deployment_details>
<deployment_name>MyDemoDepName</deployment_name>
<deployment_stage>VM_ALIVE</deployment_stage>
<external_deployment_id>8d084aab-6a66-4a70-8f47-42b91f7477ac</external_deployment_id>
<external_tenant_id>SystemAdminTenantId</external_tenant_id>
<internal_deployment_id>my_dep_id</internal_deployment_id>
<internal_tenant_id>SystemAdminTenantId</internal_tenant_id>
<service_state_machine_state>SERVICE_INERT_STATE</service_state_machine_state>
</deployment>

```

After the Deployment Update operation is done, ESC will send a SERVICE\_UPDATED callback to the client with the result.

Example callback:

```

HTTP/1.1 201 OK
Content-Type: application/xml; charset=UTF-8
ESC_TRANSACTION_ID: d53f0047-2315-466c-b7e3-aa78e0a567e8
ESC-Status-Code: 200
ESC-Status-Message: Service group update completed successfully

```

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<deployment xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
<datacenter><default>>false</default></datacenter>
<deployment_name>MyDemoDepName</deployment_name>
<deployment_stage>SERVICE_UPDATED</deployment_stage>
<external_deployment_id>8d084aab-6a66-4a70-8f47-42b91f7477ac</external_deployment_id>
<external_tenant_id>SystemAdminTenantId</external_tenant_id>
<internal_deployment_id>my_dep_id</internal_deployment_id>
<internal_tenant_id>SystemAdminTenantId</internal_tenant_id>
<service_state_machine_state>SERVICE_ACTIVE_STATE</service_state_machine_state>
</deployment>

```

If a VM specific resource like interface, is updated, then a VM\_UPDATED followed by a SERVICE\_UPDATED notification is received.

Example callback:

```
HTTP/1.1 201 OK
Content-Type: application/xml; charset=UTF-8
ESC_TRANSACTION_ID: d53f0047-2315-466c-b7e3-aa78e0a567e8
ESC-Status-Code: 200
ESC-Status-Message: VM has been updated successfully. vm: VM ID: [Sanity-vmware-de_CSR-1_0_b323effd-7f70-405
```

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<deployment xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
<datacenter><default>>false</default></datacenter>
<deployment_details>
<host_uuid>host-14882</host_uuid>
<host_name>10.85.103.13</host_name>
<vm_uuid>50217d2d-8afb-41c2-cffd-d76b1239f51e</vm_uuid>
<interfaces>
<interface>
<network_uuid>MgtNetwork</network_uuid>
<ip_address>18.0.0.17</ip_address>
<mac_address>00:50:56:a1:6a:13</mac_address>
<nic_id>1</nic_id>
<port_forwarding></port_forwarding>
<port_uuid></port_uuid>
<security_groups></security_groups>
<subnet_uuid></subnet_uuid>
<type>virtual</type>
</interface>
<interface>
<network_uuid>MgtNetwork2</network_uuid>
<ip_address>18.0.1.18</ip_address>
<mac_address>00:50:56:a1:6b:14</mac_address>
<nic_id>2</nic_id>
<port_forwarding></port_forwarding>
<port_uuid></port_uuid>
<security_groups></security_groups>
<subnet_uuid></subnet_uuid>
<type>virtual</type>
</interface>
</interfaces>
<vm_group_name>CSR-1</vm_group_name>
<vm_name>Sanity-vmware-de_CSR-1_0_b323effd-7f70-4055-b11e-0ef66f0a758d</vm_name>
<vm_state_machine_state>VM_ALIVE_STATE</vm_state_machine_state>
</deployment_details>
<deployment_name>MyDemoDepName</deployment_name>
<deployment_stage>VM_ALIVE</deployment_stage>
<external_deployment_id>8d084aab-6a66-4a70-8f47-42b91f7477ac</external_deployment_id>
<external_tenant_id>SystemAdminTenantId</external_tenant_id>
```

```

<internal_deployment_id>my_dep_id</internal_deployment_id>
<internal_tenant_id>SystemAdminTenantId</internal_tenant_id>
<service_state_machine_state>SERVICE_INERT_STATE</service_state_machine_state>
</deployment>

```

Followed by:

```

HTTP/1.1 201 OK
Content-Type: application/xml; charset=UTF-8
ESC_TRANSACTION_ID: d53f0047-2315-466c-b7e3-aa78e0a567e8
ESC-Status-Code: 200
ESC-Status-Message: Service group update completed successfully

```

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<deployment xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
<datacenter><default>false</default></datacenter>
<deployment_name>MyDemoDepName</deployment_name>
<deployment_stage>SERVICE_UPDATED</deployment_stage>
<external_deployment_id>8d084aab-6a66-4a70-8f47-42b91f7477ac</external_deployment_id>
<external_tenant_id>SystemAdminTenantId</external_tenant_id>
<internal_deployment_id>my_dep_id</internal_deployment_id>
<internal_tenant_id>SystemAdminTenantId</internal_tenant_id>
<service_state_machine_state>SERVICE_ACTIVE_STATE</service_state_machine_state>
</deployment>

```

## Parameters

Parameter	Value	Description	Parameter Type	Data Type
Callback	<input type="text" value="(required)"/>	Callback URL for ESC notifications	header	string
Callback-ESC-Events	<input type="text" value="(required)"/>	Callback URL for ESC event notifications	header	string
internal_deployment_id	<input type="text" value="(required)"/>	Internal ID of the deployment (max length 72 characters)	path	string
body	<input type="text" value="(required)"/> <div style="border: 1px solid #ccc; height: 100px; width: 100%;"></div>	ESC datamodel to update	body	string

Parameter content type:  
 ▼



## Response Messages

HTTP Status Code	Reason	Response Model	Headers
200	Request accepted		
400	Invalid Request		
404	An associated resource not found		
406	Not Acceptable		
409	Conflicting Request		
500	Internal server error		

Try it out!

GET /v0/deployments

Find all deployments

### Implementation Notes

All deployments created in ESC will be returned as a List

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
Accept	<input type="text"/>		header	string

### Response Messages

HTTP Status Code	Reason	Response Model	Headers
200	Request accepted		

Try it out!

## 07-Deployment Operations : Operations to a deployment

POST /v0/{internal\_tenant\_id}/deployments/service/{internal\_deployment\_id}

Handle deployment operation request with the operation type as payload

### Implementation Notes

Performs actions like stop, start, reboot, disable\_monitoring or enable\_monitoring on a deployment. Example request:

```
POST /v0/tenants/test-tenant1/deployments/vm/test-dep1-cirros_overid_0_cfeefcee-a1fe-4e94-9ffc-81792b265788
Content-Type: application/xml
```

Accept: application/json

Callback: http://127.0.0.1:9010/

Body:

```
<service_operation xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <operation>stop</operation>
</service_operation>
```

While the stop deployment operation is being performed, ESC will send a VM\_STOPPED callback to the client for each VM stopped as part of this request. Example callback:

HTTP/1.1 200 OK

Content-Type: application/xml; charset=UTF-8

ESC\_TRANSACTION\_ID: 5e47b5ff-bc59-4ece-b15b-8941910def74

ESC-Status-Code: 200

ESC-Status-Message: VM successfully stopped

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<deployment xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <datacenter>
    <default>>false</default>
  </datacenter>
  <deployment_details>
    <host_uuid>17be6eae40795ec068ce77a05fcb1c47f7ac42dfba160261719d4087</host_uuid>
    <host_name>my-ucs-3</host_name>
    <vm_uuid>cb86a952-c030-46f6-96f6-e81069bcea3a</vm_uuid>
    <interfaces>
      <interface>
        <network_uuid>943fda9e-79f8-400c-b442-3506f102721a</network_uuid>
        <gateway>192.168.0.1</gateway>
        <ip_address>192.168.0.135</ip_address>
        <mac_address>fa:16:3e:3c:17:b0</mac_address>
        <netmask>255.255.255.0</netmask>
        <nic_id>0</nic_id>
        <port_forwarding></port_forwarding>
        <port_uuid>263eaf20-2662-4a6b-a1f1-ed76001a1b5c</port_uuid>
        <security_groups></security_groups>
        <subnet_uuid>e313b95c-ca1f-4c81-8d60-c9e721a85d0b</subnet_uuid>
        <type>virtual</type>
      </interface>
    </interfaces>
    <vm_group_name>overridegrp2</vm_group_name>
    <vm_name>test-dep1-cirros_overid_0_cfeefcee-a1fe-4e94-9ffc-81792b265788</vm_name>
    <vm_state_machine_state>VM_SHUTDOWN_STATE</vm_state_machine_state>
  </deployment_details>
  <deployment_name>test-dep1-cirros</deployment_name>
  <deployment_stage>VM_STOPPED</deployment_stage>
  <external_deployment_id>5c278010-ad0f-4e29-bdce-f4b6c6946c38</external_deployment_id>
  <external_tenant_id>29f2cd7934dc4311ad3a34d72d1ab7b8</external_tenant_id>
  <internal_deployment_id>test-tenant1test-dep1-cirros</internal_deployment_id>
```

```
<internal_tenant_id>test-tenant1</internal_tenant_id>
<service_state_machine_state>SERVICE_STOPPED_STATE</service_state_machine_state>
</deployment>
```

After the stop deployment operation is done successfully, ESC will send a SERVICE\_STOPPED callback to the client with the result. Example callback:

```
HTTP/1.1 200 OK
Content-Type: application/xml; charset=UTF-8
ESC_TRANSACTION_ID: 5e47b5ff-bc59-4ece-b15b-8941910def74
ESC-Status-Code: 200
ESC-Status-Message: Service group suspension completed successfully

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<deployment xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <datacenter>
    <default>false</default>
  </datacenter>
  <deployment_details>
    <host_uuid>17be6eae40795ec068ce77a05fcb1c47f7ac42dfba160261719d4087</host_uuid>
    <host_name>my-ucs-3</host_name>
    <vm_uuid>cb86a952-c030-46f6-96f6-e81069bcea3a</vm_uuid>
    <interfaces>
      <interface>
        <network_uuid>943fda9e-79f8-400c-b442-3506f102721a</network_uuid>
        <gateway>192.168.0.1</gateway>
        <ip_address>192.168.0.135</ip_address>
        <mac_address>fa:16:3e:3c:17:b0</mac_address>
        <netmask>255.255.255.0</netmask>
        <nic_id>0</nic_id>
        <port_forwarding></port_forwarding>
        <port_uuid>263eaf20-2662-4a6b-a1f1-ed76001a1b5c</port_uuid>
        <security_groups></security_groups>
        <subnet_uuid>e313b95c-ca1f-4c81-8d60-c9e721a85d0b</subnet_uuid>
        <type>virtual</type>
      </interface>
    </interfaces>
    <vm_group_name>overridegrp2</vm_group_name>
    <vm_name>test-dep1-cirros_overid_0_cfeefcee-a1fe-4e94-9ffc-81792b265788</vm_name>
    <vm_state_machine_state>VM_SHUTDOWN_STATE</vm_state_machine_state>
  </deployment_details>
  <deployment_name>test-dep1-cirros</deployment_name>
  <deployment_stage>SERVICE_STOPPED</deployment_stage>
  <external_deployment_id>5c278010-ad0f-4e29-bdce-f4b6c6946c38</external_deployment_id>
  <external_tenant_id>29f2cd7934dc4311ad3a34d72d1ab7b8</external_tenant_id>
  <internal_deployment_id>test-tenant1test-dep1-cirros</internal_deployment_id>
  <internal_tenant_id>test-tenant1</internal_tenant_id>
  <service_state_machine_state>SERVICE_STOPPED_STATE</service_state_machine_state>
</deployment>
```

## Parameters

Parameter	Value	Description	Parameter Type	Data Type
Callback	<input type="text" value="(required)"/>	Callback URL for ESC notifications	header	string
Callback-ESC-Events	<input type="text" value="(required)"/>	Callback URL for ESC event notifications	header	string
internal_tenant_id	<input type="text" value="(required)"/>	Internal tenant ID (max length 72 characters)	path	string
internal_deployment_id	<input type="text" value="(required)"/>	Internal deployment ID (max length 72 characters)	path	string
body	<div style="border: 1px solid #ccc; padding: 5px; min-height: 100px;"><input type="text" value="(required)"/></div> <p>Parameter content type: <input type="text" value="application/xml"/></p>	Deployment operation payload specifying the type of operation	body	<div style="display: flex; justify-content: space-between;"><span>Model</span><span>Model Schema</span></div> <div style="border: 1px solid #ccc; padding: 5px; background-color: #fff9c4;"><pre>{   "operation": "stop",   "properties": {} }</pre></div> <p>Click to set as parameter value</p>

## Response Messages

HTTP Status Code	Reason	Response Model	Headers
200	Deployment operation request accepted		
400	Missing callback or, Deployment operation is not applicable		
404	Deployment or tenant not found		

Try it out!

## 08-VM Operations : Operations to a VM

**POST** /v0/{internal\_tenant\_id}/deployments/migrate-vm/{vm\_name}

Handle VM migrate operation request with the operation type as payload

## Parameters

Parameter	Value	Description	Parameter Type	Data Type
Callback	<input type="text" value="(required)"/>	Callback URL for notifications	header	string
Callback-ESC-Events	<input type="text" value="(required)"/>	Callback URL for ESC event notifications	header	string
internal_tenant_id	<input type="text" value="(required)"/>	Internal tenant ID (max length 72 characters)	path	string
vm_name	<input type="text" value="(required)"/>	VM name	path	string
body	<div><input type="text" value="(required)"/> Parameter content type: <input type="text" value="application/xml"/></div>	Migrate VM operation payload specifying the type of operation	body	<div>Model   Model Schema</div> <pre>{   "migrationType": "live",   "host": "string",   "blockMigration": "auto",   "operation": "migrate" }</pre> <p>Click to set as parameter value</p>

## Response Messages

HTTP Status Code	Reason	Response Model	Headers
200	VM operation request accepted		
400	Missing callback or, VM operation is not applicable		
404	VM or tenant not found		

Try it out!

**POST** /v0/{internal\_tenant\_id}/deployments/recovery-vm/{vm\_name}

Handle recovery VM operation request with the operation type as payload

## Parameters

Parameter	Value	Description	Parameter Type	Data Type
Callback	<input type="text"/>	Callback URL for notifications	header	string

Parameter	Value	Description	Parameter Type	Data Type
Callback-ESC-Events	<input type="text"/>	Callback URL for ESC event notifications	header	string
internal_tenant_id	<input type="text" value="(required)"/>	Internal tenant ID (max length 72 characters)	path	string
vm_name	<input type="text" value="(required)"/>	VM name	path	string
body	<input type="text" value="(required)"/> Parameter content type: <input type="text" value="application/xml"/>	Recovery VM operation payload specifying the type of operation: recovery_do	body	Model   Model Schema <pre>{   "operation": "string",   "checkVmBeforeRecovery": true,   "properties": {} }</pre> Click to set as parameter value

## Response Messages

HTTP Status Code	Reason	Response Model	Headers
200	Recovery VM operation request accepted		
400	Recovery VM operation is not applicable		
404	VM or tenant not found		

[Try it out!](#)

**POST** /v0/{internal\_tenant\_id}/deployments/snapshot-vm/{vm\_name}

Handle VM snapshot operation request with the operation type as payload

## Implementation Notes

Create a snapshot of an existing VM managed by ESC.

## Parameters

Parameter	Value	Description	Parameter Type	Data Type
Callback	<input type="text" value="(required)"/>	Callback URL for notifications	header	string

Parameter	Value	Description	Parameter Type	Data Type
Callback-ESC-Events	<input type="text" value="(required)"/>	Callback URL for ESC event notifications	header	string
internal_tenant_id	<input type="text" value="(required)"/>	Internal tenant ID (max length 72 characters)	path	string
vm_name	<input type="text" value="(required)"/>	VM name	path	string
body	<input type="text" value="(required)"/> Parameter content type: <input type="text" value="application/xml"/>	VM snapshot operation payload specifying the type of operation and the name of snapshot	body	Model   Model Schema <pre>{   "operation": "snapshot",   "name": "string",   "id": "string",   "metadata": {} }</pre> Click to set as parameter value

### Response Messages

HTTP Status Code	Reason	Response Model	Headers
201	VM snapshot operation request accepted		
400	Missing callback or, VM operation is not applicable		
404	VM or tenant not found		

[Try it out!](#)

**POST** /v0/{internal\_tenant\_id}/deployments/vm/{vm\_name}

Handle VM operation request with the operation type as payload

### Implementation Notes

Performs actions like stop, start, reboot, confirm\_resize, revert\_resize, disable\_monitoring or enable\_monitoring on an existing VM managed by ESC. Example request:

```
POST /v0/test-tenant1/deployments/vm/test-dep1-cirros_overid_0_cfefcfce-a1fe-4e94-9ffc-81792b265788 HTTP/1.1
Content-Type: application/xml
Accept: application/json
Callback: http://127.0.0.1:9010/
Callback-ESC-Events: http://127.0.0.1:9010/
```

Body:

```
<vm_operation xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <operation>stop</operation>
</vm_operation>
```

After the VM stop operation is done, ESC will send a VM\_STOP\_COMPLETE callback to the client with the result. Example callback:

```
HTTP/1.1 200 OK
Content-Type: application/xml; charset=UTF-8
ESC_TRANSACTION_ID: 4546c40a-d6e3-4abe-a7e7-51afd4fc5362
ESC-Status-Code: 200
ESC-Status-Message: Successfully stopped VM [test-dep1-cirros_overid_0_cfeefcee-a1fe-4e94-9ffc-81792b265788]

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<esc_event xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <deployment_name>test-dep1-cirros</deployment_name>
  <event_details></event_details>
  <event_type>VM_STOP_COMPLETE</event_type>
  <external_deployment_id>5c278010-ad0f-4e29-bdce-f4b6c6946c38</external_deployment_id>
  <external_tenant_id>29f2cd7934dc4311ad3a34d72d1ab7b8</external_tenant_id>
  <internal_deployment_id>test-tenant1test-dep1-cirros</internal_deployment_id>
  <internal_tenant_id>test-tenant1</internal_tenant_id>
  <vm_source>
    <interfaces>
      <network_uuid>943fda9e-79f8-400c-b442-3506f102721a</network_uuid>
      <gateway>192.168.0.1</gateway>
      <ip_address>192.168.0.135</ip_address>
      <mac_address>fa:16:3e:3c:17:b0</mac_address>
      <netmask>255.255.255.0</netmask>
      <nic_id>0</nic_id>
      <port_forwarding></port_forwarding>
      <port_uuid>263eaf20-2662-4a6b-a1f1-ed76001a1b5c</port_uuid>
      <security_groups></security_groups>
      <subnet_uuid>e313b95c-ca1f-4c81-8d60-c9e721a85d0b</subnet_uuid>
      <type>virtual</type>
    </interfaces>
    <host_uuid>17be6eae40795ec068ce77a05fcb1c47f7ac42dfba160261719d4087</host_uuid>
    <host_name>my-ucs-3</host_name>
    <vm_uuid>cb86a952-c030-46f6-96f6-e81069bcea3a</vm_uuid>
    <vm_group_name>overridegrp2</vm_group_name>
    <vm_name>test-dep1-cirros_overid_0_cfeefcee-a1fe-4e94-9ffc-81792b265788</vm_name>
  </vm_source>
</esc_event>
```

## Parameters

Parameter	Value	Description	Parameter Type	Data Type
-----------	-------	-------------	----------------	-----------



Parameter	Value	Description	Parameter Type	Data Type
<b>Callback</b>	<input type="text" value="(required)"/>	<b>Callback URL for notifications</b>	header	string
<b>Callback-ESC-Events</b>	<input type="text" value="(required)"/>	<b>Callback URL for ESC event notifications</b>	header	string
<b>internal_tenant_id</b>	<input type="text" value="(required)"/>	<b>Internal tenant ID (max length 72 characters)</b>	path	string
<b>vm_name</b>	<input type="text" value="(required)"/>	<b>VM name</b>	path	string
<b>body</b>	<input type="text" value="(required)"/> <div style="border: 1px solid #ccc; height: 100px; width: 100%;"></div> Parameter content type: <input type="text" value="application/xml"/>	<b>VM operation payload specifying the type of operation</b>	body	Model   Model Schema <pre style="background-color: #fff9c4; padding: 5px;">{   "operation": "stop" }</pre> Click to set as parameter value

## Response Messages

HTTP Status Code	Reason	Response Model	Headers
200	VM operation request accepted		
400	Missing callback or, VM operation is not applicable		
404	VM or tenant not found		

[Try it out!](#)

**POST** `/internal/{internal_tenant_id}/deployments/snapshot-vm/{vm_name}`  
 Handle VM snapshot operation request with the operation type as payload

## Implementation Notes

vm\_operation\_notes

## Parameters

Parameter	Value	Description	Parameter Type	Data Type
<b>Callback</b>	<input type="text" value="(required)"/>	<b>Callback URL for notifications</b>	header	string

Parameter	Value	Description	Parameter Type	Data Type
<b>Callback-ESC-Events</b>	<input type="text" value="(required)"/>	<b>Callback URL for ESC event notifications</b>	header	string
<b>internal_tenant_id</b>	<input type="text" value="(required)"/>	<b>Internal tenant ID (max length 72 characters)</b>	path	string
<b>vm_name</b>	<input type="text" value="(required)"/>	<b>VM name</b>	path	string
<b>body</b>	<input type="text" value="(required)"/> <div style="border: 1px solid #ccc; width: 100px; height: 80px; margin-top: 5px;"></div> Parameter content type: <input type="text" value="application/xml"/>	<b>VM operation payload specifying the type of operation</b>	body	Model   Model Schema <pre style="background-color: #fff9c4; padding: 10px; border: 1px solid #ccc;">{   "operation": "snapshot",   "name": "string",   "id": "string",   "metadata": {} }</pre> <p style="font-size: small; margin-top: 5px;">Click to set as parameter value</p>

### Response Messages

HTTP Status Code	Reason	Response Model	Headers
201	VM snapshot operation request accepted		
400	Missing callback or, VM operation is not applicable		
404	VM or tenant not found		

Try it out!

## 09-Configuration Parameters

GET
/v0/config
Retrieve all ESC config parameters

### Implementation Notes

Retrieve all ESC config parameters

### Response Messages

HTTP Status Code	Reason	Response Model	Headers
200	Get all parameters request accepted		

Try it out!

GET /v0/config/{category}

Retrieve all ESC config parameters under a category

### Implementation Notes

Retrieve all ESC config parameters under a category

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
<b>category</b>	<input type="text" value="(required)"/>	<b>ESC config parameter category</b>	path	string

### Response Messages

HTTP Status Code	Reason	Response Model	Headers
200	Get category request accepted.		
400	Invalid category name provided.		
404	Category not found.		

Try it out!

GET /v0/config/{category}/{key}

Retrieve a ESC config parameters under a category, with a given key

### Implementation Notes

Retrieve a ESC config parameters under a category, with a given key

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
<b>category</b>	<input type="text" value="(required)"/>	<b>ESC config parameter category</b>	path	string
<b>key</b>	<input type="text" value="(required)"/>	<b>ESC config parameter key</b>	path	string

### Response Messages

HTTP Status Code	Reason	Response Model	Headers
------------------	--------	----------------	---------

HTTP Status Code	Reason	Response Model	Headers
200	Get key under category request accepted.		
400	Invalid key or category name provided.		
404	Key or category not found.		

Try it out!

**POST** /v0/config/{category}/{key} Create a new key under a category.

**Implementation Notes**

Create a new key under a category.

**Response Class (Status 200)**

Model | Model Schema

```

{
  "category": "DEFAULT",
  "key": "MAX_LOGS",
  "value": {},
  "type": "STRING"
}

```

Response Content Type  ▼

**Parameters**

Parameter	Value	Description	Parameter Type	Data Type
category	<input type="text" value="(required)"/>	ESC config parameter category	path	string
key	<input type="text" value="(required)"/>	ESC config parameter key	path	string
body	<input type="text" value="(required)"/>	ESC config parameter value	body	string

Parameter content type:

▼

Try it out!

### Implementation Notes

Change an ESC config parameter under a category, with a given key.

Format: PUT /v0/config/{category}/{key}/{value}

Example request:

curl -X PUT -H "accept: Application/xml" <http://127.0.0.1:8080/ESCManager/v0/config/log/level/debug>

Example response:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<escconfigparameter>
  <category>LOG</category>
  <key>LEVEL</key>
  <type>STRING</type>
  <value xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xs="http://www.w3.org/2001/XMLSchema" />
</escconfigparameter>
```

Details of Config Parameters: Please refer to installation and user guide for more details.

Category: DEFAULT Key: VM\_RECOVERY\_RETRIES\_MAX Default Value: 3 Type: Int Description: Number of maximum recovery attempts allowed per VM.

Category: OPENSTACK Key: ENDPOINT Default Value: adminURL Type: String Description: The parameter to set up the keystone endpoint value of ESC. Options : adminURL, publicURL

Category: LOG Key: LEVEL Default Value: INFO Type: String Description: Level of logging of ESCManager. Options: INFO, TRACE, DEBUG.

Category: AFFINITY Key: FILTER Default Value: SameHostFilter Type: String Description: A constant string used to build PolicyEngine and initializing VM policy table. Options: SameHostFilter, ServerGroupAffinity

Category: ANTI-AFFINITY Key: FILTER Default Value: DifferentHostFilter Type: String Description: A constant string used to build PolicyEngine and initializing VM policy table. Options: DifferentHostFilter, ServerGroupAffinity

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
category	<input type="text" value="(required)"/>	ESC config parameter category	path	string
key	<input type="text" value="(required)"/>	ESC config parameter key	path	string

Parameter	Value	Description	Parameter Type	Data Type
value	<input type="text" value="(required)"/>	ESC config parameter value	path	string

### Response Messages

HTTP Status Code	Reason	Response Model	Headers
200	Update value request accepted.		
400	Invalid request.		
404	Associated resource not found.		
406	Not acceptable.		
409	Conflicting resource.		
500	Internal server error.		

[Try it out!](#)

## 10-Volumes : Manage ESC volumes

**DELETE** /v0/{internalTenantId}/volumes/{internalVolumeId} Delete a volume from an existing tenant in ESC

### Implementation Notes

Delete a volume from ESC

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
callback	<input type="text" value="(required)"/>	Callback URL for ESC notifications	header	string
internalVolumeId	<input type="text" value="(required)"/>	Internal volume ID (max length 72 characters)	path	string
internalTenantId	<input type="text" value="(required)"/>	Internal tenant ID of the volume (max length 72 characters)	path	string

### Response Messages

HTTP Status Code	Reason	Response Model	Headers
200	Request accepted		

HTTP Status Code	Reason	Response Model	Headers
400	Invalid request		
500	Internal server error		

Try it out!

**GET** /v0/{internalTenantId}/volumes/{internalVolumeId} Get a volume by its tenant and internal volume ID

### Implementation Notes

Only volumes created using ESC will be checked

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
internalTenantId	<input type="text" value="(required)"/>	Internal tenant ID of the volume (max length 72 characters)	path	string
internalVolumeId	<input type="text" value="(required)"/>	Internal volume ID (max length 72 characters)	path	string

### Response Messages

HTTP Status Code	Reason	Response Model	Headers
200	Request accepted		
404	Volume Id not found		

Try it out!

**POST** /v0/{internalTenantId}/volumes/{internalVolumeId} Create a volume under an existing tenant in ESC

### Implementation Notes

Create a volume in ESC

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
callback	<input type="text" value="(required)"/>	Callback URL for ESC notifications	header	string

Parameter	Value	Description	Parameter Type	Data Type
<b>internalTenantId</b>	<input type="text" value="(required)"/>	<b>Internal tenant ID of the volume (max length 72 characters)</b>	path	string
<b>internalVolumeId</b>	<input type="text" value="(required)"/>	<b>Internal volume ID (max length 72 characters)</b>	path	string
<b>body</b>	<input type="text" value="(required)"/> Parameter content type: <input type="text" value="application/xml"/>	<b>Volume configuration payload</b>	body	Model   Model Schema <pre>{   "volume_id": "string",   "volume_snapshot_id": "string" }</pre> Click to set as parameter value

### Response Messages

HTTP Status Code	Reason	Response Model	Headers
200	Request accepted		
400	Invalid request		
404	Volume Id not found		
409	Conflicting request		
503	Unable to query VIM		

[Try it out!](#)

**GET** `/v0/{internalTenantId}/volumes` Get all volumes under a tenant

### Implementation Notes

All volumes under that tenant in ESC will be returned as a List

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
<b>internalTenantId</b>	<input type="text" value="(required)"/>	<b>Internal tenant ID of the volume (max length 72 characters)</b>	path	string

### Response Messages



HTTP Status Code	Reason	Response Model	Headers
200	Request accepted		

Try it out!

## 11-ESC Operation Mode

**POST** /v0/operationmode/{escOpMode} Set operation mode in ESC

**Implementation Notes**  
Set operation mode in ESC

**Response Class (Status 200)**

Model | Model Schema

```

{
  "message": "string",
  "mode": "string"
}

```

Response Content Type

**Parameters**

Parameter	Value	Description	Parameter Type	Data Type
escOpMode	<input type="text" value="MAINTENANCE"/>	ESC maintenance mode.	path	string

Try it out!

**GET** /v0/operationmode Get operation mode in ESC

**Implementation Notes**  
Set operation mode in ESC

**Response Class (Status 200)**

Model | Model Schema

```

{
  "message": "string",
  "mode": "string"
}

```

Response Content Type  ▼

Try it out!

## 12-Host Actions

**GET** /v0/hosts/{hostName}/status Get the current status of the host

### Implementation Notes

This is to see if the host can be schedule to deploy instances

### Response Class (Status 200)

Response Content Type  ▼

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
Client-Transaction-Id	<input type="text"/>		header	string
hostName	<input type="text" value="(required)"/>		path	string

Try it out!

**POST** /v0/hosts/{hostName}/disable Disable a host

### Implementation Notes

To disable a host

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
Callback-ESC-Events	<input type="text" value="(required)"/>	Callback URL for host action notifications	header	string
hostName	<input type="text" value="(required)"/>	Target host name	path	string

### Response Messages

HTTP Status Code	Reason	Response Model	Headers
default	successful operation		

Try it out!

POST /v0/hosts/{hostName}/enable

Enable a host

### Implementation Notes

To enable a host

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
Callback-ESC-Events	<input type="text" value="(required)"/>	Callback URL for host action notifications	header	string
hostName	<input type="text" value="(required)"/>	Target host name	path	string

### Response Messages

HTTP Status Code	Reason	Response Model	Headers
default	successful operation		

Try it out!

## 13-ESC VIM Connectors : Manage ESC VIM Connectors

PUT /v0/vims/{vim\_id}/check\_status

Trigger VIM connector status checking

### Implementation Notes

Trigger VIM connector status checking

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
vim_id	<input type="text" value="(required)"/>	VIM ID	path	string

### Response Messages

HTTP Status Code	Reason	Response Model	Headers
200	Request accepted		
400	Invalid Request		
404	Not Found		

HTTP Status Code	Reason	Response Model	Headers
500	Internal Server Error		

Try it out!

**GET** /v0/vims Get all VIM Connectors

**Implementation Notes**  
All VIM Connectors in ESC will be returned as a List

**Response Class (Status 200)**

Model | Model Schema

```
[
  {
    "id": "string",
    "type": "OPENSTACK",
    "properties": {
      "property": [
        {
          "name": "string",
          "value": "string",
          "encrypted_value": "string"
        }
      ]
    },
    "event_type": "string"
  }
]
```

Response Content Type  ▼

Try it out!

**POST** /v0/vims Create a VIM connector in ESC

**Implementation Notes**  
Create a VIM connector in ESC

**Parameters**

Parameter	Value	Description	Parameter Type	Data Type
-----------	-------	-------------	----------------	-----------

Parameter	Value	Description	Parameter Type	Data Type
<b>Callback</b>	<input type="text" value="(required)"/>	<b>Callback URL for ESC notifications</b>	header	string
<b>body</b>	<input type="text" value="(required)"/> <div style="border: 1px solid #ccc; width: 100px; height: 100px; margin: 5px;"></div> Parameter content type: <input type="text" value="application/xml"/>	<b>VIM Connector configuration payload</b>	body	Model   Model Schema <pre> {   "id": "string",   "type": "OPENSTACK",   "properties": {     "property": [       {         "name": "string",         "value": "string",         "encrypted_value": "string"       }     ]   },   "event_type": "string" }           </pre> <p>Click to set as parameter value</p>

### Response Messages

HTTP Status Code	Reason	Response Model	Headers
200	Request accepted		
400	Invalid Request		
500	Internal Server Error		

**GET** /v0/vims/default\_vim\_connector Get default VIM Connector name

### Response Class (Status 200)

Model | Model Schema

```

{
  "defaultVimConnectorId": "string",
  "event_type": "string"
}
          
```

Response Content Type

POST

/v0/vims/default\_vim\_connector

Create a Default VIM connector in ESC

## Implementation Notes

Create a Defaukt VIM connector in ESC

## Parameters

Parameter	Value	Description	Parameter Type	Data Type
<b>callback</b>	<input type="text" value="(required)"/>	<b>Callback URL for ESC notifications</b>	header	string
<b>body</b>	<input type="text" value="(required)"/> Parameter content type: <input type="text" value="application/xml"/>	<b>Default VIM Connector configuration payload</b>	body	Model   Model Schema <pre>{   "defaultVimConnectorId": "string",   "event_type": "string" }</pre> Click to set as parameter value

## Response Messages

HTTP Status Code	Reason	Response Model	Headers
200	Request accepted		
400	Invalid Request		
500	Internal Server Error		

Try it out!

DELETE

/v0/vims/{vim\_id}

Delete a VIM Connector from ESC

## Implementation Notes

Delete a VIM Connector from ESC

## Parameters

Parameter	Value	Description	Parameter Type	Data Type
<b>callback</b>	<input type="text" value="(required)"/>	<b>Callback URL for ESC notifications</b>	header	string
<b>vim_id</b>	<input type="text" value="(required)"/>	<b>VIM ID</b>	path	string

## Response Messages

HTTP Status Code	Reason	Response Model	Headers
200	VIM Connector deletion request accepted		
400	Invalid Request		
404	VIM Id not found		
500	Internal Server Error		

Try it out!

GET	/v0/vims/{vim_id}	Get a VIM Connector by its id		
<b>Parameters</b>				
Parameter	Value	Description	Parameter Type	Data Type
<b>vim_id</b>	<input type="text" value="(required)"/>	<b>VIM ID</b>	path	string
<b>Response Messages</b>				
HTTP Status Code	Reason	Response Model	Headers	
200	Get request accepted			
404	VIM Id not found			

Try it out!

PUT	/v0/vims/{vim_id}	Update an existing VIM connector in ESC		
<b>Implementation Notes</b>				
Update an existing VIM connector in ESC				
<b>Parameters</b>				
Parameter	Value	Description	Parameter Type	Data Type
<b>callback</b>	<input type="text" value="(required)"/>	<b>Callback URL for ESC notifications</b>	header	string
<b>vim_id</b>	<input type="text" value="(required)"/>	<b>VIM ID</b>	path	string

Parameter	Value	Description	Parameter Type	Data Type
<b>body</b>	(required)          Parameter content type: application/xml ▾	<b>Update VIM Connector configuration payload</b>	body	Model   Model Schema <pre>{   "id": "string",   "type": "OPENSTACK",   "properties": {     "property": [       {         "name": "string",         "value": "string",         "encrypted_value": "string"       }     ]   },   "event_type": "string" }</pre> Click to set as parameter value

### Response Messages

HTTP Status Code	Reason	Response Model	Headers
200	Request accepted		
400	Invalid Request		
500	Internal Server Error		

[Try it out!](#)

## 14-ESC VIM Users : Manage ESC VIM Users

**GET** /v0/vims/{vim\_id}/vim\_users Get all VIM Users

### Implementation Notes

All VIM Users in ESC will be returned as a List

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
<b>vim_id</b>	(required)		path	string

### Response Messages

HTTP Status Code	Reason	Response Model	Headers
200	Get request accepted		



HTTP Status Code	Reason	Response Model	Headers
404	Unknown VIM Id provided		

Try it out!

**POST** /v0/vims/{vim\_id}/vim\_users Create a VIM user under an existing VIM in ESC

### Implementation Notes

Create a VIM user in ESC

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
Callback	<input type="text" value="(required)"/>	Callback URL for ESC notifications	header	string
vim_id	<input type="text" value="(required)"/>	Vim id	path	string

<b>body</b>	<input type="text" value="(required)"/>	<b>Vim user configuration payload</b>	body	Model	Model Schema
				<pre> {   "credentials": {     "properties": {       "property": [         {           "name": "string",           "value": "string",           "encrypted_value": "string"         }       ]     }   },   "id": "string",   "vim_id": "string",   "event_type": "string" } </pre>	

Click to set as parameter value

### Response Messages

HTTP Status Code	Reason	Response Model	Headers
200	Request accepted		
400	Invalid Request		
500	Internal Server Error		

Try it out!

**DELETE** /v0/vims/{vim\_id}/vim\_users/{vim\_user\_id}

Delete a VIM User from ESC

### Implementation Notes

Delete a VIM User from ESC

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
<b>callback</b>	<input type="text" value="(required)"/>	<b>Callback URL for ESC notifications</b>	header	string
<b>vim_id</b>	<input type="text" value="(required)"/>	<b>Vim id</b>	path	string
<b>vim_user_id</b>	<input type="text" value="(required)"/>	<b>Vim user id</b>	path	string

### Response Messages

HTTP Status Code	Reason	Response Model	Headers
200	Request accepted		
400	Invalid Request		
404	VIM User Id not found		
500	Internal Server Error		

Try it out!

**GET** /v0/vims/{vim\_id}/vim\_users/{vim\_user\_id}

Get a VIM User by its id

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
<b>vim_id</b>	<input type="text" value="(required)"/>	<b>Vim id</b>	path	string
<b>vim_user_id</b>	<input type="text" value="(required)"/>	<b>Vim user id</b>	path	string

### Response Messages

HTTP Status Code	Reason	Response Model	Headers
200	Get request accepted		

HTTP Status Code	Reason	Response Model	Headers
404	VIM User Id not found		

Try it out!

**PUT** /v0/vims/{vim\_id}/vim\_users/{vim\_user\_id} Update a VIM user under an existing VIM in ESC

### Implementation Notes

Update a VIM user in ESC

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
Callback	<input type="text" value="(required)"/>	Callback URL for ESC notifications	header	string
vim_id	<input type="text" value="(required)"/>	Vim id	path	string
vim_user_id	<input type="text" value="(required)"/>	Vim user id	path	string

body

Vim user configuration payload

body

Model | Model Schema

```

{
  "credentials": {
    "properties": {
      "property": [
        {
          "name": "string",
          "value": "string",
          "encrypted_value": "string"
        }
      ]
    }
  },
  "id": "string",
  "vim_id": "string",
  "event_type": "string"
}

```

Click to set as parameter value

Parameter content type:

### Response Messages

HTTP Status Code	Reason	Response Model	Headers
200	Request accepted		
400	Invalid Request		

HTTP Status Code	Reason	Response Model	Headers
500	Internal Server Error		

Try it out!

## 15-System States

POST	/v0/systemstate/{name}/reset	Reset system state of a given name												
<h3>Implementation Notes</h3> <p>Reset system state of a given name</p>														
<h3>Parameters</h3> <table border="1"> <thead> <tr> <th>Parameter</th> <th>Value</th> <th>Description</th> <th>Parameter Type</th> <th>Data Type</th> </tr> </thead> <tbody> <tr> <td>name</td> <td><input type="text" value="(required)"/></td> <td>System state name</td> <td>path</td> <td>string</td> </tr> </tbody> </table>			Parameter	Value	Description	Parameter Type	Data Type	name	<input type="text" value="(required)"/>	System state name	path	string		
Parameter	Value	Description	Parameter Type	Data Type										
name	<input type="text" value="(required)"/>	System state name	path	string										
<h3>Response Messages</h3> <table border="1"> <thead> <tr> <th>HTTP Status Code</th> <th>Reason</th> <th>Response Model</th> <th>Headers</th> </tr> </thead> <tbody> <tr> <td>200</td> <td>Reset system state of specified name request accepted.</td> <td></td> <td></td> </tr> <tr> <td>404</td> <td>Name not found.</td> <td></td> <td></td> </tr> </tbody> </table> <p>Try it out!</p>			HTTP Status Code	Reason	Response Model	Headers	200	Reset system state of specified name request accepted.			404	Name not found.		
HTTP Status Code	Reason	Response Model	Headers											
200	Reset system state of specified name request accepted.													
404	Name not found.													

GET	/v0/systemstate/{name}	Retrieve system state of a given name										
<h3>Implementation Notes</h3> <p>Retrieve system state of a given name</p>												
<h3>Parameters</h3> <table border="1"> <thead> <tr> <th>Parameter</th> <th>Value</th> <th>Description</th> <th>Parameter Type</th> <th>Data Type</th> </tr> </thead> <tbody> <tr> <td>name</td> <td><input type="text" value="(required)"/></td> <td>System state name</td> <td>path</td> <td>string</td> </tr> </tbody> </table>			Parameter	Value	Description	Parameter Type	Data Type	name	<input type="text" value="(required)"/>	System state name	path	string
Parameter	Value	Description	Parameter Type	Data Type								
name	<input type="text" value="(required)"/>	System state name	path	string								
<h3>Response Messages</h3> <table border="1"> <thead> <tr> <th>HTTP Status Code</th> <th>Reason</th> <th>Response Model</th> <th>Headers</th> </tr> </thead> <tbody> <tr> <td>200</td> <td>Get system state of specified name request accepted.</td> <td></td> <td></td> </tr> </tbody> </table>			HTTP Status Code	Reason	Response Model	Headers	200	Get system state of specified name request accepted.				
HTTP Status Code	Reason	Response Model	Headers									
200	Get system state of specified name request accepted.											

HTTP Status Code	Reason	Response Model	Headers
404	Name not found.		

[Try it out!](#)

## 16-File Servers

GET	/v0/fileserver	Get all File Servers	
<b>Implementation Notes</b>			
All file servers created will be returned as a List			
<b>Response Messages</b>			
HTTP Status Code	Reason	Response Model	Headers
200	Request accepted		

[Try it out!](#)

DELETE	/v0/fileserver/{internalServerId}	Delete a File Server		
<b>Implementation Notes</b>				
Delete a File Server				
<b>Parameters</b>				
Parameter	Value	Description	Parameter Type	Data Type
<b>callback</b>	<input type="text" value="(required)"/>	<b>Callback URL for ESC notifications</b>	header	string
<b>internalServerId</b>	<input type="text" value="(required)"/>	<b>File Server ID</b>	path	string
<b>Response Messages</b>				
HTTP Status Code	Reason	Response Model	Headers	
200	Request accepted			
404	Not found			
500	Internal server error			

[Try it out!](#)

### Implementation Notes

Matching File server will be returned if exists

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
<b>internalServerId</b>	<input type="text" value="(required)"/>	<b>File Server Id</b>	path	string

### Response Messages

HTTP Status Code	Reason	Response Model	Headers
200	Request accepted		
404	Not found		

[Try it out!](#)

### Implementation Notes

Create a File Server

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
<b>callback</b>	<input type="text" value="(required)"/>	<b>Callback URL for ESC notifications</b>	header	string
<b>internalServerId</b>	<input type="text" value="(required)"/>	<b>File Server ID</b>	path	string
<b>body</b>	<div style="border: 1px solid #ccc; padding: 5px; min-height: 100px;"><p>(required)</p></div> <p>Parameter content type: <input type="text" value="application/xml"/></p>	<b>ESC datamodel with the file server payload</b>	body	string

### Response Messages

HTTP Status Code	Reason	Response Model	Headers
200	Request accepted		

HTTP Status Code	Reason	Response Model	Headers
204	No content		
400	Invalid Request		
409	Conflicting Request		

Try it out!

**PUT** /v0/fileserver/{internalServerId} Update a File Server

**Implementation Notes**

Update a File Server

**Parameters**

Parameter	Value	Description	Parameter Type	Data Type
<b>callback</b>	<input type="text" value="(required)"/>	<b>Callback URL for ESC notifications</b>	header	string
<b>internalServerId</b>	<input type="text" value="(required)"/>	<b>File Server ID</b>	path	string
<b>body</b>	<input type="text" value="(required)"/> <div style="border: 1px solid #ccc; height: 80px; width: 100%;"></div>	<b>ESC datamodel with the file server payload</b>	body	string

Parameter content type:

**Response Messages**

HTTP Status Code	Reason	Response Model	Headers
200	Request accepted		
204	No content		
400	Invalid Request		
404	Not Found		

Try it out!

**17-Dynamic Mappings**

**DELETE** /v0/dynamic\_mapping/actions Delete all non-CORE dynamic mapping actions from ESC

## Implementation Notes

Deletes all non-CORE dynamic mapping actions from ESC

## Response Messages

HTTP Status Code	Reason	Response Model	Headers
200	Delete all non-core actions request accepted		

Try it out!

**GET** /v0/dynamic\_mapping/actions Retrieve all the dynamic mapping actions defined in ESC

## Implementation Notes

Retrieves all the dynamic mapping actions defined in ESC

## Response Messages

HTTP Status Code	Reason	Response Model	Headers
200	Get all actions request accepted		

Try it out!

**POST** /v0/dynamic\_mapping/actions Create new dynamic mapping action(s)

## Implementation Notes

create\_dynamic\_mapping\_actions\_notes

## Parameters

Parameter	Value	Description	Parameter Type	Data Type
<b>body</b>	(required) <div style="border: 1px solid #ccc; height: 80px; width: 100%;"></div>	<b>Actions payload</b>	body	string

Parameter content type:

## Response Messages

HTTP Status Code	Reason	Response Model	Headers
------------------	--------	----------------	---------



HTTP Status Code	Reason	Response Model	Headers
200	Create new action request accepted		
400	'Exception Message' or 'No action definition found.'		
409	Action [actionName] already exists.		

Try it out!

DELETE	/v0/dynamic_mapping/actions/{action_name}	Delete a dynamic mapping action from ESC		
<b>Implementation Notes</b>				
Deletes a dynamic mapping action from ESC				
<b>Parameters</b>				
Parameter	Value	Description	Parameter Type	Data Type
<b>action_name</b>	<input type="text" value="(required)"/>	<b>Name of the action you want to delete</b>	path	string
<b>Response Messages</b>				
HTTP Status Code	Reason	Response Model	Headers	
200	Delete action request accepted			
400	'Invalid action name provided', 'Action [actionName] is a CORE action, cannot delete.', or 'Action [actionName] is currently referenced in a deployment, cannot delete.'			
404	Action [actionName] not found.			

Try it out!

GET	/v0/dynamic_mapping/actions/{action_name}	Retrieve a dynamic mapping action by name
<b>Implementation Notes</b>		
Retrieves a dynamic mapping action by name		
<b>Parameters</b>		

Parameter	Value	Description	Parameter Type	Data Type
action_name	<input type="text" value="(required)"/>	Name of the action you want to retrieve	path	string

**Response Messages**

HTTP Status Code	Reason	Response Model	Headers
200	Get action request accepted		
400	Invalid action name provided [actionName].		
404	Action [actionName] not found.		

[Try it out!](#)

DELETE	/v0/dynamic_mapping/metrics	Delete all non-CORE dynamic mapping metrics from ESC	
<b>Implementation Notes</b>			
Deletes all non-CORE dynamic mapping metrics from ESC			
<b>Response Messages</b>			
HTTP Status Code	Reason	Response Model	Headers
200	Delete all non-core metrics request accepted		

[Try it out!](#)

GET	/v0/dynamic_mapping/metrics	Retrieve all the dynamic mapping metrics defined in ESC	
<b>Implementation Notes</b>			
Retrieves all the dynamic mapping metrics defined in ESC			
<b>Response Messages</b>			
HTTP Status Code	Reason	Response Model	Headers
200	Get all metrics request accepted		

[Try it out!](#)

POST	/v0/dynamic_mapping/metrics	Create new dynamic mapping metric(s)
------	-----------------------------	--------------------------------------

## Implementation Notes

create\_dynamic\_mapping\_metrics\_notes

## Parameters

Parameter	Value	Description	Parameter Type	Data Type
<b>body</b>	<div style="border: 1px solid #ccc; padding: 5px; min-height: 100px;">(required)</div> <p>Parameter content type: application/xml ▼</p>	<b>Metrics payload</b>	body	string

## Response Messages

HTTP Status Code	Reason	Response Model	Headers
200	Create new metric request accepted		
400	'Exception Message' or 'No metric definition found.'		
409	Metric [metricName] already exists.		

Try it out!

**DELETE** /v0/dynamic\_mapping/metrics/{metric\_name}

Delete a dynamic mapping metric from ESC

## Implementation Notes

Deletes a dynamic mapping metric from ESC

## Parameters

Parameter	Value	Description	Parameter Type	Data Type
<b>metric_name</b>	<div style="border: 1px solid #ccc; padding: 2px;">(required)</div>	<b>Name of the metric you want to delete</b>	path	string

## Response Messages

HTTP Status Code	Reason	Response Model	Headers
200	Delete metric request accepted		

HTTP Status Code	Reason	Response Model	Headers
400	'Invalid metric name provided', 'Metric [metricName] is a CORE metric, cannot delete.', or 'Metric [metricName] is currently referenced in a deployment, cannot delete.'		
404	Metric [metricName] not found.		

[Try it out!](#)

**GET** `/v0/dynamic_mapping/metrics/{metric_name}` Retrieve a dynamic mapping metric by name

**Implementation Notes**  
Retrieves a dynamic mapping metric by name

**Parameters**

Parameter	Value	Description	Parameter Type	Data Type
<code>metric_name</code>	<input type="text" value="(required)"/>	<b>Name of the metric you want to retrieve</b>	path	string

**Response Messages**

HTTP Status Code	Reason	Response Model	Headers
200	Get metric request accepted		
400	Invalid metric name provided [metricName].		
404	Metric [metricName] not found.		

[Try it out!](#)

## 18-VM Snapshots

**GET** `/v0/snapshots` Get all VMSnapshots

**Implementation Notes**  
Return all VMSnapshots created as a List

**Parameters**

Parameter	Value	Description	Parameter Type	Data Type
-----------	-------	-------------	----------------	-----------

Parameter	Value	Description	Parameter Type	Data Type
internalTenantId	<input type="text"/>		query	string
generatedVMName	<input type="text"/>		query	string

### Response Messages

HTTP Status Code	Reason	Response Model	Headers
200	Request accepted		

[Try it out!](#)

**POST** `/v0/snapshots/{snapshotId}/volumes` Create a volume from a snapshot

### Implementation Notes

Creation of a bootable volume from an ESC created volume snapshot. This is an asynchronous operation

### Parameters

Parameter	Value	Description	Parameter Type	Data Type				
snapshotId	<input type="text" value="(required)"/>	Snapshot ID	path	string				
callback	<input type="text" value="(required)"/>	Callback URL for ESC notifications	header	string				
body	<input type="text" value="(required)"/>	Volume name	body	<table border="1"> <thead> <tr> <th>Model</th> <th>Model Schema</th> </tr> </thead> <tbody> <tr> <td></td> <td><pre>{}</pre></td> </tr> </tbody> </table> <p><a href="#">Click to set as parameter value</a></p>	Model	Model Schema		<pre>{}</pre>
Model	Model Schema							
	<pre>{}</pre>							

Parameter content type:

### Response Messages

HTTP Status Code	Reason	Response Model	Headers
202	Request Accepted		
404	Snapshot not found		

[Try it out!](#)

**DELETE** `/v0/snapshots/{snapshotIdOrName}` Handle VM snapshot delete request

## Implementation Notes

Delete VM snapshot

## Parameters

Parameter	Value	Description	Parameter Type	Data Type
<b>Callback</b>	<input type="text" value="(required)"/>	<b>Callback URL for notifications</b>	header	string
<b>Callback-ESC-Events</b>	<input type="text" value="(required)"/>	<b>Callback URL for ESC event notifications</b>	header	string
<b>snapshotIdOrName</b>	<input type="text" value="(required)"/>	<b>Snapshot Id/Name</b>	path	string

## Response Messages

HTTP Status Code	Reason	Response Model	Headers
200	VM snapshot delete operation request accepted		
404	Snapshot image not found		
409	The request could not be processed because snapshot name is not unique		

Try it out!

GET /internal/snapshots

Get all VMSnapshots

## Implementation Notes

Return all VMSnapshots created as a List

## Parameters

Parameter	Value	Description	Parameter Type	Data Type
<b>internalTenantId</b>	<input type="text"/>		query	string
<b>generatedVMName</b>	<input type="text"/>		query	string

## Response Messages

HTTP Status Code	Reason	Response Model	Headers
200	Request accepted		

Try it out!

POST

/internal/snapshots/{snapshotId}/volumes

Create a volume from a snapshot

### Implementation Notes

Creation of a bootable volume from an ESC created volume snapshot. This is an asynchronous operation

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
snapshotId	<input type="text" value="(required)"/>	Snapshot ID	path	string
callback	<input type="text" value="(required)"/>	Callback URL for ESC notifications	header	string
body	<input type="text" value="(required)"/> <div style="border: 1px solid #ccc; height: 80px; width: 100%;"></div> Parameter content type: <input type="text" value="application/json"/>	Volume name	body	Model   Model Schema <div style="border: 1px solid #ccc; background-color: #fff9c4; padding: 5px; display: inline-block;">{ }</div> Click to set as parameter value

### Response Messages

HTTP Status Code	Reason	Response Model	Headers
202	Request Accepted		
404	Snapshot not found		

[Try it out!](#)

DELETE

/internal/snapshots/{snapshotIdOrName}

Handle VM snapshot delete request

### Implementation Notes

Delete VM snapshot

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
callback	<input type="text" value="(required)"/>	Callback URL for notifications	header	string
callback-ESC-Events	<input type="text" value="(required)"/>	Callback URL for ESC event notifications	header	string

Parameter	Value	Description	Parameter Type	Data Type
<b>snapshotIdOrName</b>	<input type="text" value="(required)"/>	<b>Snapshot Id/Name</b>	path	string

### Response Messages

HTTP Status Code	Reason	Response Model	Headers
200	VM snapshot delete operation request accepted		
404	Snapshot image not found		
409	The request could not be processed because snapshot name is not unique		

[Try it out!](#)

## 19-Routers : Manage ESC Routers

**DELETE** /v0/{internalTenantId}/routers/{internalRouterId} Delete a router from an existing tenant in ESC

### Implementation Notes

Delete a router from ESC

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
<b>callback</b>	<input type="text" value="(required)"/>	<b>Callback URL for ESC notifications</b>	header	string
<b>internalRouterId</b>	<input type="text" value="(required)"/>	<b>Internal router ID (max length 72 characters)</b>	path	string
<b>internalTenantId</b>	<input type="text" value="(required)"/>	<b>Internal tenant ID of the router (max length 72 characters)</b>	path	string

### Response Messages

HTTP Status Code	Reason	Response Model	Headers
200	Request accepted		
400	Invalid request		
500	Internal server error		



Try it out!

POST /v0/{internalTenantId}/routers/{internalRouterId}

Create a router under an existing tenant in ESC

### Implementation Notes

Create a router in ESC

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
<b>Callback</b>	<input type="text" value="(required)"/>	<b>Callback URL for ESC notifications</b>	header	string
<b>internalTenantId</b>	<input type="text" value="(required)"/>	<b>Internal tenant ID of the network (max length 72 characters)</b>	path	string
<b>internalRouterId</b>	<input type="text" value="(required)"/>	<b>Internal router ID (max length 72 characters)</b>	path	string

Parameter	Value	Description	Parameter Type	Data Type
<b>body</b>	<div style="border: 1px solid #ccc; padding: 5px; min-height: 80px;">(required)</div> <p>Parameter content type:  <input type="text" value="application/xml"/></p>	<b>Router configuration payload</b>	body	<div style="display: flex; border-bottom: 1px solid #ccc;"> <div style="border-right: 1px solid #ccc; padding: 5px;">Model</div> <div style="padding: 5px;">Model Schema</div> </div> <pre style="background-color: #fff9c4; padding: 10px;"> {   "internal_tenant_id": "string",   "internal_router_id": "string",   "name": "string",   "external_tenant_id": "string",   "tenant_name": "string",   "external_router_id": "string",   "admin_state": true,   "external_network": "string",   "description": "string",   "external_network_id": "string",   "interface": [     {       "internalAttachInterfaceId": "string",       "routerName": "string",       "subnetId": "string",       "internal_router_id": "string",       "router_id": "string",       "subnet": "string",       "port_id": "string",       "event_type": "ATTACH_INTERFACE"     }   ],   "route": [     {       "routerId": "string",       "internalId": "string",       "internal_router_id": "string",       "destination": "string",       "next_hop": "string",       "route_name": "string",       "event_type": "ATTACH_INTERFACE"     }   ],   "event_type": "CREATE_ROUTER",   "snat_enable": true,   "isdistributed": true } </pre> <p style="text-align: right; font-size: small;">Click to set as parameter value</p>

## Response Messages

HTTP Status Code	Reason	Response Model	Headers
200	Request accepted		
400	Invalid request		
404	Router Id not found		

HTTP Status Code	Reason	Response Model	Headers
409	Conflicting request		
503	Unable to query VIM		

[Try it out!](#)

**PUT** `/v0/{internalTenantId}/routers/{internalRouterId}` [Update a Router](#)

### Implementation Notes

Update a Router

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
<b>Callback</b>	<input type="text" value="(required)"/>	<b>Callback URL for ESC notifications</b>	header	string
<b>internalTenantId</b>	<input type="text" value="(required)"/>	<b>Internal tenant ID of the network (max length 72 characters)</b>	path	string
<b>internalRouterId</b>	<input type="text" value="(required)"/>	<b>Internal router ID (max length 72 characters)</b>	path	string

Parameter	Value	Description	Parameter Type	Data Type
<b>body</b>	<div style="border: 1px solid #ccc; padding: 5px; min-height: 80px;">(required)</div> <p>Parameter content type:  <input type="text" value="application/xml"/></p>	<b>Router configuration payload</b>	body	Model   Model Schema <div style="border: 1px solid #ccc; padding: 10px; background-color: #fff9c4; margin-top: 5px;"> <pre> {   "internal_tenant_id": "string",   "internal_router_id": "string",   "name": "string",   "external_tenant_id": "string",   "tenant_name": "string",   "external_router_id": "string",   "admin_state": true,   "external_network": "string",   "description": "string",   "external_network_id": "string",   "interface": [     {       "internalAttachInterfaceId": "string",       "routerName": "string",       "subnetId": "string",       "internal_router_id": "string",       "router_id": "string",       "subnet": "string",       "port_id": "string",       "event_type": "ATTACH_INTERFACE"     }   ],   "route": [     {       "routerId": "string",       "internalId": "string",       "internal_router_id": "string",       "destination": "string",       "next_hop": "string",       "route_name": "string",       "event_type": "ATTACH_INTERFACE"     }   ],   "event_type": "CREATE_ROUTER",   "snat_enable": true,   "isdistributed": true } </pre> </div> <p style="text-align: right; margin-top: 5px;"><a href="#">Click to set as parameter value</a></p>

### Response Messages

HTTP Status Code	Reason	Response Model	Headers
200	Request accepted		
204	No content		
400	Invalid Request		

HTTP Status Code	Reason	Response Model	Headers
404	Not Found		

[Try it out!](#)

## SSL Management

PUT	/internal/ssl/reload	Reload ESC Truststore from file system	
<b>Implementation Notes</b> Reload ESC Truststore from file system			
<b>Response Messages</b>			
HTTP Status Code	Reason	Response Model	Headers
204	ESC Truststore reloaded successfully		
500	Failed to reload ESC truststore		

[Try it out!](#)

[ BASE URL: /ESCManager , API VERSION: 2.2.0 ]