

# Troubleshoot RCMChassisReload Due to RCM Initiated Reload

## Contents

[Introduction](#)

[Background Information](#)

[RCMChassisReload Trap](#)

[Data Collection and Analysis](#)

[Related Information](#)

## Introduction

This document describes how to investigate the `RCMChassisReload` trap due to `RCM Initiated Reload` in StarOS.

## Background Information

The Redundancy Configuration Manager (RCM) is a Cisco proprietary node/network function (NF) that provides redundancy of StarOS-based User Plane (UP) / User Plane Function (UPF). The RCM provides N:M redundancy of UP/UPFs wherein "N" is the number of Active UP/UPFs and is less than 10, and "M" is the number of Standby UP/UPF in the redundancy group. The RCM monitors its UP/UPFs for failures and initiates a switchover to a standby UP/UPFs.

Unplanned switchovers occur due to issues in the UP and it gets rebooted without manual intervention. When an unplanned switchover happens, the Bidirectional Forwarding Detection (BFD) monitor Pod detects that the UP has gone down and triggers the RCM Controller to start the switchover mechanism. The RCM controller chooses a Standby UP and the Redundancy Manager Pods to push configuration and checkpoints to the new UP.

The manual switchover can be initiated from the RCM with `rcm switchover source <source_ip_address> destination <dest_ip_address>` command.

## RCMChassisReload Trap

The `RCMChassisReload` trap due to `RCM Initiated Reload` is reported in StarOS.

Similar logs are reported when the reload occurs due to BFD failure and manual command execution.

Logs due to BFD failure:

```
2022-Nov-03+12:35:28.682 [snmp 22002 info] [1/0/6083 <vpnmgr:5> trap_api.c:11832] [software internal system syslog] Internal trap notification 1427 (RCMChassisReload) RCM Chassis Reload Reason: (2) RCM Initiated Reload
2022-Nov-03+12:35:28.682 [srp 84201 info] [1/0/6083 <vpnmgr:5> vpmngr_rcm.c:1492] [context: RCM,
```

```
contextID: 5] [software internal system syslog] RCM: Attempting to reload UPF.
2022-Nov-03+12:35:28.735 [snmp 22002 info] [1/0/5271 <cspectrl:0> trap_api.c:17907] [software
internal system syslog] Internal trap notification 1521 (CseShutDownNotify) Shutdown reason
"Reload chassis requested by CLI command"
```

Logs due to manual switchover command:

```
2022-Nov-03+12:41:04.984 [snmp 22002 info] [1/0/6073 <vpnmgr:5> trap_api.c:11832] [software
internal system syslog] Internal trap notification 1427 (RCMChassisReload) RCM Chassis Reload
Reason: (2) RCM Initiated Reload
2022-Nov-03+12:41:04.984 [srp 84201 info] [1/0/6073 <vpnmgr:5> vpnmgr_rcm.c:1492] [context: RCM,
contextID: 5] [software internal system syslog] RCM: Attempting to reload UPF.
2022-Nov-03+12:41:05.014 [snmp 22002 info] [1/0/5265 <cspectrl:0> trap_api.c:17907] [software
internal system syslog] Internal trap notification 1521 (CseShutDownNotify) Shutdown reason
"Reload chassis requested by CLI command"
```

## Data Collection and Analysis

The RCM switchovers are shown in the `rcm show-statistics switchover` command output.

In the example, the latest switchover at 15:28:14 on Nov 3 was due to BGP failover on the UP/UPF, while prior switchover was at 15:14:23 on Nov 3 due to manual command switchover from RCM.

```
[unknown] rcm# rcm show-statistics switchover
Thu Nov 3 15:39:10.486 UTC+00:00
message :
{
  "stats_history": [
  {
    "status": "Success",
    "started": "Nov 3 15:28:12.315",
    "ended": "Nov 3 15:28:14.107",
    "switchoverreason": "BGP Failure",
    "source_endpoint": "192.168.60.3",
    "destination_endpoint": "192.168.60.4"
  },
  {
    "status": "Success",
    "started": "Nov 3 15:13:48.808",
    "ended": "Nov 3 15:14:23.670",
    "switchoverreason": "Planned Switchover",
    "source_endpoint": "192.168.60.4",
    "destination_endpoint": "192.168.60.3"
  },
  {

```

In case the reason for the `RCMChassisReload` is not identified from the log then collect the data:

- Collect `show support details` from active and standby UP/UPFs.
- Collect `syslog` information from active and standby UP/UPFs.
- Collect `rcm support-summary` and `syslog` information from the RCM.
- Check RCM Pod events: Verify Kubernetes pods

```
ubuntu@CUPS-RCM-01:~$ kubectl get pods -n rcm
NAME READY STATUS RESTARTS AGE
documentation-65d698cfbb-1941g 1/1 Running 0 161d
etcd-rcm-etcd-cluster-0 2/2 Running 0 161d
grafana-dashboard-etcd-rcm-65bd789-t57pq 1/1 Running 0 161d
ops-center-rcm-ops-center-6f946946c7-wlpnq 5/5 Running 0 161d
```

```
prometheus-rules-etc-5c5cff47c6-vl7 1/1 Running 0 161d
rcm-bfdmgr-7fd47466c4-xm99h 1/1 Running 0 161d
rcm-checkpointmgr-0 1/1 Running 0 161d
rcm-checkpointmgr-1 1/1 Running 0 161d
rcm-checkpointmgr-2 1/1 Running 0 161d
rcm-checkpointmgr-3 1/1 Running 0 161d
rcm-configmgr-569f6d89c5-g7ztg 1/1 Running 0 161d
rcm-controller-775c4cc7bb-q96m6 1/1 Running 0 161d
smart-agent-rcm-ops-center-5c475b6bd-2plc6 1/1 Running 1 161d
```

## Collect the describe command of the checkpoint manager

```
ubuntu@CUPS-RCM-01:~$ kubectl describe pod rcm-checkpointmgr-0 -n rcm
```

```
Name: rcm-checkpointmgr-0
Namespace: rcm
Priority: 0
Node: rcm/10.10.1.1
Start Time: Wed, 01 Jun 2022 23:38:40 +0000
Labels: component=rcm-checkpointmgr
controller-revision-hash=rcm-checkpointmgr-566cdd886f
release=rcm-rcm-chkptmgr
statefulset.kubernetes.io/pod-name=rcm-checkpointmgr-0
Annotations: cnf.projectcalico.org/containerID:
0dea15df9e41a9195d9827cdb257430bab3257bad3417281fb6c8f3d3ed146cc
cnf.projectcalico.org/podIP: 10.42.0.72/32
cnf.projectcalico.org/podIPs: 10.42.0.72/32
prometheus.io/port: 8081
prometheus.io/scrape: true
sidecar.istio.io/inject: false
Status: Running
IP: 10.10.0.72
IPs:
IP: 10.10.0.72
Controlled By: StatefulSet/rcm-checkpointmgr
Containers:
rcm-checkpointmgr:
Container ID: docker://b86826c43e191f0266a1489ef6f0398b21c1801d6a79e40093aed6e3c023ba4d
Image: dockerhub.cisco.com/smi-fuse-docker-internal/mobile-cnat-rcm/rcm-
chkptmgr/v21.27.x/rcm_chkptmgr:0.0.5-38a8de3
Image ID: docker://sha256:adc4013783f60f6413fa81eb2bf16a652fddcd8d164e469368c2587560e42bc8
Ports: 9900/TCP, 9300/TCP, 8080/TCP, 8081/TCP
Host Ports: 0/TCP, 0/TCP, 0/TCP, 0/TCP
Command:
/usr/local/bin/run-app
State: Running
Started: Wed, 01 Jun 2022 23:38:44 +0000
Ready: True
Restart Count: 0
Environment:
K8S_NAMESPACE: rcm
GODEBUG: madvdontneed=1
GOGC: 25
GOTRACEBACK: crash
SERVICE_NAME: rcm-checkpointmgr
DATACENTER_NAME: DC
CLUSTER_NAME: Local
APPLICATION_NAME: RCM
RCM_CHKPT_POD_IP: (v1:status.podIP)
RCM_CHKPT_POD_NAME: rcm-checkpointmgr-0 (v1:metadata.name)
INFRA_PROMETHEUS_PORT: 8081
Mounts:
/config/rcm-logging from rcm-logging-volume (ro)
/var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-6828r (ro)
Conditions:
Type Status
```

```
Initialized True
Ready True
ContainersReady True
PodScheduled True
Volumes:
rcm-logging-volume:
Type: ConfigMap (a volume populated by a ConfigMap)
Name: infra-logging-conf
Optional: false
kube-api-access-6828r:
Type: Projected (a volume that contains injected data from multiple sources)
TokenExpirationSeconds: 3607
ConfigMapName: kube-root-ca.crt
ConfigMapOptional: <nil>
DownwardAPI: true
QoS Class: BestEffort
Node-Selectors: nodeType=RCM
Tolerations: node.kubernetes.io/not-ready:NoExecute op=Exists for 300s
node.kubernetes.io/unreachable:NoExecute op=Exists for 300s
Events: <none>
```

### Check the current logs of the Pod

```
ubuntu@CUPS-RCM-01:~$ kubectl logs rcm-checkpointmgr-0 -n rcm | more
2022/11/09 20:19:01.554 rcm-checkpointmgr [DEBUG] [TopologyData.go:295]
[infra.topology.core] Setting state of the application as APP_STARTED
2022/11/09 20:19:01.558 rcm-checkpointmgr [DEBUG] [TopologyData.go:295]
[infra.topology.core] Setting state of the application as APP_STARTED
```

If a Pod has crashed, the logs from the previous Pod can be collected with `-p` option

```
ubuntu@CUPS-RCM-01:~$ kubectl logs <pod name> -n rcm -p
```

## Related Information

[RCM Configuration and Administration Guide](#)