Steps to RMA the RCM Based AIO Server in CNDP Deployment

Contents

Introduction

This document describes the detailed procedure for Return Material Authorization (RMA) for the Redundancy Configuration Manager (RCM) based All-in-One (AIO) server in Cloud Native Deployment Platform (CNDP) deployment for any hardware issues or Maintenance related activities.

Prerequisites

Requirements

Cisco recommends that you have knowledge of these topics:

- RCM
- Kubernetes

Components Used

The information in this document is based on the RCM version - rcm.2021.02.1.i18

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, ensure that you understand the potential impact of any command.

Know the RCM IP Schema

This document explains the RCM design that consists of two AIO nodes with two RCM Opscenters and one RCM CEE one each for the AIO node.

The target RCM AIO node for the RMA in this article is AIO-1 (AI0301) which contains both the RCM opscenters in the PRIMARY state.

POD_NAME	NODE_NAME	IP_ADDRESS	DEVICE_TYPE	OS_TYPE
UP0300	RCE301	10.1.2.9	RCM_CEE_AIO_1	opscenter
UP0300	RCE302	10.1.2.10	RCM_CEE_AIO_2	opscenter
UP0300	AI0301	10.1.2.7	RCM_K8_AIO_1	linux
UP0300	AI0302	10.1.2.8	RCM_K8_AIO_2	linux
UP0300	RM0301	10.1.2.3	RCM1_ACTIVE	opscenter
UP0300	RM0302	10.1.2.4	RCM1_STANDBY	opscenter
UP0300	RM0303	10.1.2.5	RCM2_ACTIVE	opscenter
UP0300	RM0304	10.1.2.6	RCM2_STANDBY	opscenter

Backup Procedure

Backup the Configuration

To begin with, collect the config backup of the running-config from RCM opscenters which runs on the target AIO node.

```
# show running-config | nomore
```

Collect the running-config from RCM CEE opscenters which runs on the target AIO node.

```
# show running-config | nomore
```

Precheck Procedure

Prechecks on AIO

Collect the command output from both AIO nodes and verify all the pods are in the Running state.

```
# kubectl get ns
# kubectl get pods -A -o wide
```

Sample Prechecks Output

Note the two RCM opscenters and one RCM CEE opscenter runs on the AIO-1 node

```
cloud-user@up0300-aio-1-master-1:~$ kubectl get ns
NAME              STATUS   AGE
cee-rce301       Active   110d <--
default           Active   110d
istio-system      Active   110d
kube-node-lease   Active   110d
kube-public       Active   110d
kube-system       Active   110d
nginx-ingress     Active   110d
rcm-rm0301       Active   110d <--
rcm-rm0303       Active   110d <--
registry          Active   110d
smi-certs         Active   110d
smi-node-label    Active   110d
smi-vips          Active   110d
cloud-user@up0300-aio-1-master-1:~$
```

Login to both the RCM opscenter of AIO-1 and verify the status.

```
[up0300-aio-1/rm0301] rcm# rcm show-status
message :
{"status":[" Fri Oct 29 07:21:11 UTC 2021 : State is MASTER"]}
[up0300-aio-1/rm0301] rcm#

[up0300-aio-1/rm0303] rcm# rcm show-status
message :
{"status":[" Fri Oct 29 07:22:18 UTC 2021 : State is MASTER"]}
[up0300-aio-1/rm0303] rcm#
```

Repeat the same steps on the AIO-2 node where the other two RCM opscenters corresponds to the AIO-1 node are present.

```
cloud-user@up0300-aio-2-master-1:~$ kubectl get ns
```

```
NAME                STATUS   AGE
cee-rce302          Active   105d  <--
default             Active   105d
istio-system        Active   105d
kube-node-lease     Active   105d
kube-public         Active   105d
kube-system         Active   105d
nginx-ingress       Active   105d
rcm-rm0302          Active   105d  <--
rcm-rm0304          Active   105d  <--
registry            Active   105d
smi-certs           Active   105d
smi-node-label      Active   105d
smi-vips            Active   105d
cloud-user@up0300-aio-2-master-1:~$
```

Login to both the RCM opscenter of AIO-2 and verify the status.

```
[up0300-aio-2/rm0302] rcm# rcm show-status
message :
{"status":[" Fri Oct 29 09:32:54 UTC 2021 : State is BACKUP"]}
[up0300-aio-2/rm0302] rcm#

[up0300-aio-2/rm0304] rcm# rcm show-status
message :
{"status":[" Fri Oct 29 09:33:51 UTC 2021 : State is BACKUP"]}
[up0300-aio-2/rm0304] rcm#
```

Execution Procedure

Steps to Execute on RCM Before Shut Down AIO Node

1. As both the RCMs on AIO-1 are MASTER, you can migrate them to BACKUP.

a. To do that, you have to execute the **rcm migrate primary** command on the Active RCMs before you shut off the AIO-1 server.

```
[up0300-aio-1/rm0301] rcm# rcm migrate primary

[up0300-aio-1/rm0303] rcm# rcm migrate primary
```

b. Verify the status is now BACKUP on AIO-1.

```
[up0300-aio-1/rm0301] rcm# rcm show-status

[up0300-aio-1/rm0303] rcm# rcm show-status
```

c. Verify the status is now MASTER on AIO-2 and ensure they are MASTER.

```
[up0300-aio-1/rm0302] rcm# rcm show-status

[up0300-aio-1/rm0304] rcm# rcm show-status
```

d. Perform RCM shutdown on both rm0301 and rm0303.

```
[up0300-aio-2/rm0301] rcm# config
Entering configuration mode terminal
[up0300-aio-2/rm0301] rcm(config)# system mode shutdown
```

```
[up0300-aio-1/rce301] rcm(config)# commit comment <CRNUMBER>

[up0300-aio-2/rm0303] rcm# config
Entering configuration mode terminal
[up0300-aio-2/rm0303] rcm(config)# system mode shutdown
[up0300-aio-1/rce303] rcm(config)# commit comment <CRNUMBER>
```

2. We also have to shut down the CEE ops that run on the AIO-1, commands used.

```
[up0300-aio-1/rce301] cee# config
Entering configuration mode terminal
[up0300-aio-1/rce301] cee(config)# system mode shutdown
[up0300-aio-1/rce301] cee(config)# commit comment <CRNUMBER>
[up0300-aio-1/rce301] cee(config)# exit
```

Wait a couple of minutes and check the system to show 0.0%.

```
[up0300-aio-1/rce301] cee# show system
```

3. Verify there are no pods for RCM and CEE namespaces except for documentation, smart-agent, ops-center-rcm and ops-center-cee pods

```
# kubectl get pods -n rcm-rm0301 -o wide
# kubectl get pods -n rcm-rm0303 -o wide
# kubectl get pods -n cee-rce302 -o wide
```

Steps to Execute on Kubernetes Node Before Shut Down AIO Node

Drain the Kubernetes node so the pods and services associated are gracefully terminated. The scheduler would no longer select this Kubernetes node and evict pods from that node. Please drain a single node at a time.

Login to the SMI Cluster Manager.

```
cloud-user@bot-deployer-cm-primary:~$ kubectl get svc -n smi-cm
NAME                                          TYPE        CLUSTER-IP       EXTERNAL-IP
PORT(S)                                                  AGE
cluster-files-offline-smi-cluster-deployer    ClusterIP   10.102.108.177   <none>
8080/TCP                                                 78d
iso-host-cluster-files-smi-cluster-deployer   ClusterIP   10.102.255.174   192.168.0.102
80/TCP                                                   78d
iso-host-ops-center-smi-cluster-deployer      ClusterIP   10.102.58.99     192.168.0.100
3001/TCP                                                 78d
netconf-ops-center-smi-cluster-deployer       ClusterIP   10.102.108.194   10.244.110.193
3022/TCP,22/TCP                                          78d
ops-center-smi-cluster-deployer               ClusterIP   10.102.156.123   <none>
8008/TCP,2024/TCP,2022/TCP,7681/TCP,3000/TCP,3001/TCP   78d
squid-proxy-node-port                         NodePort    10.102.73.130    <none>
3128:31677/TCP                                           78d
cloud-user@bot-deployer-cm-primary:~$ ssh -p 2024 admin@<Cluster IP of ops-center-smi-cluster-
deployer>

     Welcome to the Cisco SMI Cluster Deployer on bot-deployer-cm-primary
     Copyright © 2016-2020, Cisco Systems, Inc.
     All rights reserved.
admin connected from 192.168.0.100 using ssh on ops-center-smi-cluster-deployer-686b66d9cd-nfzx8
[bot-deployer-cm-primary] SMI Cluster Deployer#
[bot-deployer-cm-primary] SMI Cluster Deployer# show clusters
```

```
                  LOCK TO
NAME               VERSION
----------------------------
cp0100-smf-data  -
cp0100-smf-ims   -
cp0200-smf-data  -
cp0200-smf-ims   -
up0300-aio-1     -     <--
up0300-aio-2     -
up0300-upf-data  -
up0300-upf-ims   -
```

Drain the master node:

```
[bot-deployer-cm-primary] SMI Cluster Deployer# clusters up0300-aio-1 nodes master-1 actions
sync drain remove-node true
This would run drain on the node, disrupting pods running on the node.  Are you sure? [no,yes]
yes
message accepted
```

Mark the master-1 node into maintenance mode:

```
[bot-deployer-cm-primary] SMI Cluster Deployer# config
Entering configuration mode terminal
[bot-deployer-cm-primary] SMI Cluster Deployer(config)# clusters up0300-aio-1
[bot-deployer-cm-primary] SMI Cluster Deployer(config-clusters-up0300-aio-1)# nodes master-1
[bot-deployer-cm-primary] SMI Cluster Deployer(config-nodes-master1)# maintenance true
[bot-deployer-cm-primary] SMI Cluster Deployer(config-nodes-master1)# commit
Commit complete.
[bot-deployer-cm-primary] SMI Cluster Deployer(config-nodes-master1)# end
```

Run Cluster sync and monitor the logs for the sync action:

```
[bot-deployer-cm-primary] SMI Cluster Deployer# clusters up0300-aio-1 nodes master-1 actions
sync
This would run sync.  Are you sure? [no,yes] yes
message accepted
[bot-deployer-cm-primary] SMI Cluster Deployer# clusters up0300-aio-1 nodes master-1 actions
sync logs
```

Sample output for cluster sync logs:

```
[installer-master] SMI Cluster Deployer#  clusters kali-stacked nodes cmts-worker1-1 actions
sync logs
Example Cluster Name: kali-stacked
Example WorkerNode: cmts-worker1
logs 2020-10-06 20:01:48.023 DEBUG cluster_sync.kali-stacked.cmts-worker1: Cluster name: kali-
stacked
2020-10-06 20:01:48.024 DEBUG cluster_sync.kali-stacked.cmts-worker1: Node name: cmts-worker1
2020-10-06 20:01:48.024 DEBUG cluster_sync.kali-stacked.cmts-worker1: debug: false
2020-10-06 20:01:48.024 DEBUG cluster_sync.kali-stacked.cmts-worker1: remove_node: true
PLAY [Check required variables] ************************************************
TASK [Gathering Facts] *********************************************************
Tuesday 06 October 2020  20:01:48 +0000 (0:00:00.017)       0:00:00.017 *******
ok: [master3]
ok: [master1]
ok: [cmts-worker1]
ok: [cmts-worker3]
ok: [cmts-worker2]
ok: [master2]
TASK [Check node_name] *********************************************************
```

```
Tuesday 06 October 2020  20:01:50 +0000 (0:00:02.432)       0:00:02.450 *******
skipping: [master1]
skipping: [master2]
skipping: [master3]
skipping: [cmts-worker1]
skipping: [cmts-worker2]
skipping: [cmts-worker3]
PLAY [Wait for ready and ensure uncordoned] ************************************
TASK [Cordon and drain node] ***************************************************
Tuesday 06 October 2020  20:01:51 +0000 (0:00:00.144)       0:00:02.594 *******
skipping: [master1]
skipping: [master2]
skipping: [master3]
skipping: [cmts-worker2]
skipping: [cmts-worker3]
TASK [upgrade/cordon : Cordon/Drain/Delete node] *******************************
Tuesday 06 October 2020  20:01:51 +0000 (0:00:00.205)       0:00:02.800 *******
changed: [cmts-worker1 -> 172.22.18.107]
PLAY RECAP *********************************************************************
cmts-worker1               : ok=2    changed=1    unreachable=0    failed=0    skipped=1
rescued=0    ignored=0
cmts-worker2               : ok=1    changed=0    unreachable=0    failed=0    skipped=2
rescued=0    ignored=0
cmts-worker3               : ok=1    changed=0    unreachable=0    failed=0    skipped=2
rescued=0    ignored=0
master1                    : ok=1    changed=0    unreachable=0    failed=0    skipped=2
rescued=0    ignored=0
master2                    : ok=1    changed=0    unreachable=0    failed=0    skipped=2
rescued=0    ignored=0
master3                    : ok=1    changed=0    unreachable=0    failed=0    skipped=2
rescued=0    ignored=0
Tuesday 06 October 2020  20:02:29 +0000 (0:00:38.679)       0:00:41.479 *******
===============================================================================
2020-10-06 20:02:30.057 DEBUG cluster_sync.kali-stacked.cmts-worker1: Cluster sync successful
2020-10-06 20:02:30.058 DEBUG cluster_sync.kali-stacked.cmts-worker1: Ansible sync done
2020-10-06     0:02:30.058 INFO cluster_sync.kali-stacked.cmts-worker1: _sync finished.  Opening
lock
```

Server Maintenance Procedure

Power Off the server from CIMC gracefully. Proceed with the hardware-related maintenance activity as defined in the Hardware MoP and ensure all the health checks are passed after the server is powered ON.

Note: This article does not cover the hardware or maintenance activity MoP for the server as they differ from the problem statement

Kubernetes Restore Procedure

Steps to Execute on Kubernetes Node Post Power on AIO Node

Login to the SMI Cluster Manager:

```
cloud-user@bot-deployer-cm-primary:~$ kubectl get svc -n smi-cm
NAME                                         TYPE        CLUSTER-IP       EXTERNAL-IP
PORT(S)                                                AGE
cluster-files-offline-smi-cluster-deployer    ClusterIP   10.102.108.177   <none>
8080/TCP                                               78d
```

```
iso-host-cluster-files-smi-cluster-deployer   ClusterIP   10.102.255.174   192.168.0.102
80/TCP                                                  78d
iso-host-ops-center-smi-cluster-deployer      ClusterIP   10.102.58.99     192.168.0.100
3001/TCP                                                78d
netconf-ops-center-smi-cluster-deployer       ClusterIP   10.102.108.194   10.244.110.193
3022/TCP,22/TCP                                         78d
ops-center-smi-cluster-deployer               ClusterIP   10.102.156.123   <none>
8008/TCP,2024/TCP,2022/TCP,7681/TCP,3000/TCP,3001/TCP   78d
squid-proxy-node-port                         NodePort    10.102.73.130    <none>
3128:31677/TCP                                          78d
cloud-user@bot-deployer-cm-primary:~$ ssh -p 2024 admin@<ClusterIP of ops-center-smi-cluster-
deployer>
     Welcome to the Cisco SMI Cluster Deployer on bot-deployer-cm-primary
     Copyright © 2016-2020, Cisco Systems, Inc.
     All rights reserved.
admin connected from 192.168.0.100 using ssh on ops-center-smi-cluster-deployer-686b66d9cd-nfzx8
[bot-deployer-cm-primary] SMI Cluster Deployer#
[bot-deployer-cm-primary] SMI Cluster Deployer# show clusters
                  LOCK TO
NAME               VERSION
----------------------------
cp0100-smf-data  -
cp0100-smf-ims   -
cp0200-smf-data  -
cp0200-smf-ims   -
up0300-aio-1     -     <--
up0300-aio-2     -
up0300-upf-data  -
up0300-upf-ims   -
```

Turn off the maintenance flag for the master-1 to be added back into cluster.

```
[bot-deployer-cm-primary] SMI Cluster Deployer# config
Entering configuration mode terminal
[bot-deployer-cm-primary] SMI Cluster Deployer(config)# clusters up0300-aio-1
[bot-deployer-cm-primary] SMI Cluster Deployer(config-clusters-up0300-aio-1)# nodes master-1
[bot-deployer-cm-primary] SMI Cluster Deployer(config-nodes-master-1)# maintenance false
[bot-deployer-cm-primary] SMI Cluster Deployer(config-nodes-master-1)# commit
Commit complete.
[bot-deployer-cm-primary] SMI Cluster Deployer(config-nodes-master-1)# end
```

Restore the master node pods and services with cluster sync action.

```
[bot-deployer-cm-primary] SMI Cluster Deployer# clusters up0100-aio-1 nodes master-1 actions
sync run debug true
This would run sync.  Are you sure? [no,yes] yes
message accepted
```

Monitor the logs for the sync action.

```
[bot-deployer-cm-primary] SMI Cluster Deployer# clusters up0100-aio-1 nodes master-1 actions
sync logs
```

Check the cluster status of the AIO-1 master.

```
[bot-deployer-cm-primary] SMI Cluster Deployer# clusters up0300-aio-1 actions k8s cluster-status
```

Sample output:

```
[installer-] SMI Cluster Deployer# clusters kali-stacked actions k8s cluster-status
```

```
pods-desired-count 67
pods-ready-count 67
pods-desired-are-ready true
etcd-healthy true
all-ok true
```

RCM Restore procedure

Steps to Execute on CEE and RCM Ops-Centers to Restore Application

Update CEE opscenter and RCM opscenter into running mode.

Configure the running mode for rce301.

```
[up0300-aio-1/rce301] cee# config
Entering configuration mode terminal
[up0300-aio-1/rce301] cee(config)# system mode running
[up0300-aio-1/rce301] cee(config)# commit comment <CRNUMBER>
[up0300-aio-1/rce301] cee(config)# exit
```

Wait for a couple of minutes and check the system is at 100.0%.

```
[up0300-aio-1/rce301] cee# show system
```

Configure the **running** mode for rm0301.

```
[up0300-aio-2/rm0301] rcm# config
Entering configuration mode terminal
[up0300-aio-2/rm0301] rcm(config)# system mode running
[up0300-aio-1/rce301] rcm(config)# commit comment <CRNUMBER>
```

Wait for a couple of minutes and verify the system is at 100.0%.

```
[up0300-aio-1/rm0301] cee# show system
```

Configure the **running** mode for rm0303.

```
[up0300-aio-2/rm0303] rcm# config
Entering configuration mode terminal
[up0300-aio-2/rm0303] rcm(config)# system mode running
[up0300-aio-1/rce303] rcm(config)# commit comment <CRNUMBER>
```

Wait for a couple of minutes and check the system is at 100.0%.

```
[up0300-aio-1/rm0303] cee# show system
```

Verification Procedure

Verify the pods are all UP and **Running** state on both the AIO nodes with these commands.

```
on AIO nodes:
kubectl get ns
kubectl get pods -A -o wide
```

on RCM ops-centers:

```
rcm show-status
```