

# CNDP 구축에서 RCM 기반 AIO 서버를 RMA하는 단계

## 목차

[소개](#)

[사전 요구 사항](#)

[요구 사항](#)

[사용되는 구성 요소](#)

[RCM IP 스키마 이해](#)

[백업 절차](#)

[구성 백업](#)

[사전 검사 절차](#)

[AIO에 대한 사전 검사](#)

[샘플 미리 검사 출력](#)

[실행 절차](#)

[AIO 노드를 종료하기 전에 RCM에서 실행하는 단계](#)

[AIO 노드를 종료하기 전에 Kubernetes 노드에서 실행하는 단계](#)

[서버 유지 관리 절차](#)

[Kubernetes 복원 절차](#)

[AIO 노드의 Kubernetes 노드 Post Power에서 실행하는 단계](#)

[RCM 복원 절차](#)

[CEE 및 RCM Ops-Centers에서 애플리케이션을 복원하는 단계](#)

[확인 절차](#)

## 소개

이 문서에서는 하드웨어 문제 또는 유지 관리 관련 활동에 대한 CNDP(Cloud Native Deployment Platform) 구축의 RCM(Redundancy Configuration Manager) 기반 AIO(All-in-One) 서버에 대한 RMA(Return Material Authorization)의 자세한 절차에 대해 설명합니다.

## 사전 요구 사항

### 요구 사항

다음 주제에 대한 지식을 보유하고 있으면 유용합니다.

- RCM
- 쿠버네티스

### 사용되는 구성 요소

이 문서의 정보는 RCM 버전 - rcm.2021.02.1.i18을 기반으로 합니다.

이 문서의 정보는 특정 랩 환경의 디바이스를 토대로 작성되었습니다. 이 문서에 사용된 모든 디바이스는 초기화된(기본) 컨피그레이션으로 시작되었습니다. 현재 네트워크가 작동 중인 경우 모든 명령의 잠재적인 영향을 미리 숙지하시기 바랍니다.

## RCM IP 스키마 이해

이 문서에서는 AIO 노드에 대해 RCM Opscenters 2개와 RCM CEE 1개가 있는 AIO 노드 2개로 구성된 RCM 설계에 대해 설명합니다.

이 문서에서 RMA의 대상 RCM AIO 노드는 PRIMARY 상태의 RCM opscnters를 모두 포함하는 AIO-1(AI0301)입니다.

POD_NAME	NODE_NAME	IP_ADDRESS	DEVICE_TYPE	OS_TYPE
UP0300	RCE301	10.1.2.9	RCM_CEE_AIO_1	opscnter
UP0300	RCE302	10.1.2.10	RCM_CEE_AIO_2	opscnter
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	opscnter
UP0300	RM0302	10.1.2.4	RCM1_STANDBY	opscnter
UP0300	RM0303	10.1.2.5	RCM2_ACTIVE	opscnter
UP0300	RM0304	10.1.2.6	RCM2_STANDBY	opscnter

## 백업 절차

### 구성 백업

먼저 대상 AIO 노드에서 실행되는 RCM opscenters에서 running-config의 컨피그레이션 백업을 수집합니다.

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

대상 AIO 노드에서 실행되는 RCM CEE opscenters에서 running-config를 수집합니다.

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

## 사전 검사 절차

### AIO에 대한 사전 검사

두 AIO 노드에서 명령 출력을 수집하고 모든 Pod가 Running(실행 중) 상태인지 확인합니다.

```
# kubectl get ns
```

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

### 샘플 미리 검사 출력

AIO-1 노드에서 RCM opscenters 2개와 RCM CEE opscenter 1개가 실행됩니다

```
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:~$
```

AIO-1의 RCM opscenter에 모두 로그인하고 상태를 확인합니다.

```
[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#
```

다른 두 RCM opscenters 가 AIO-1 노드에 해당하는 AIO-2 노드에서 동일한 단계를 반복합니다.

```
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:~$

```

AIO-2의 RCM opscenter에 모두 로그인하고 상태를 확인합니다.

```

[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#

```

## 실행 절차

### AIO 노드를 종료하기 전에 RCM에서 실행하는 단계

1. AIO-1의 RCM은 모두 MASTER이므로 BACKUP으로 마이그레이션할 수 있습니다.

a. 이를 위해서는 AIO-1 서버를 종료하기 전에 Active RCM에서 `rcm migrate primary` 명령을 실행해야 합니다.

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

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

b. AIO-1에서 상태가 BACKUP인지 확인합니다.

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

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

c. 상태가 AIO-2에서 MASTER인지 확인하고 MASTER인지 확인합니다.

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

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

d. rm0301 및 rm0303에서 RCM 종료를 수행합니다.

```
[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. 사용된 AIO-1 명령에서 실행되는 CEE도 종료해야 합니다.

```
[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
```

몇 분 정도 기다렸다가 0.0%가 표시되도록 시스템을 확인합니다.

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

3. 문서, 스마트 에이전트, ops-center-rcm 및 ops-center-cee pod 외에는 RCM 및 CEE 네임스페이스용 포드가 없는지 확인합니다.

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

## AIO 노드를 종료하기 전에 Kubernetes 노드에서 실행하는 단계

Kubernetes 노드를 드레이하여 연결된 포드 및 서비스가 정상적으로 종료되도록 합니다. 스케줄러는 더 이상 이 Kubernetes 노드를 선택하고 해당 노드에서 포드를 제거하지 않습니다. 한 번에 하나의 노드를 드레이닝하십시오.

SMI Cluster Manager에 로그인합니다.

```
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     -
```

마스터 노드를 드레이닝합니다.

```
[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
```

마스터-1 노드를 유지 관리 모드로 표시합니다.

```
[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
```

클러스터 동기화를 실행하고 동기화 작업에 대한 로그를 모니터링합니다.

```
[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
```

클러스터 동기화 로그의 샘플 출력:

```
[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

```

## 서버 유지 관리 절차

CIMC에서 서버의 전원을 정상적으로 끕니다. 하드웨어 MoP에 정의된 대로 하드웨어 관련 유지 관리 작업을 진행하고 서버의 전원이 켜진 후 모든 상태 확인이 전달되었는지 확인합니다.

참고: 이 문서에서는 서버의 하드웨어 또는 유지 보수 활동 MoP가 문제 설명서와 다르므로 이에 대해서는 다루지 않습니다

## Kubernetes 복원 절차

### AIO 노드의 Kubernetes 노드 Post Power에서 실행하는 단계

SMI Cluster Manager에 로그인합니다.

```

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   -

```

클러스터에 다시 추가할 master-1에 대한 유지 관리 플래그를 해제합니다.

```

[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

```

클러스터 동기화 작업을 사용하여 마스터 노드 포드 및 서비스를 복원합니다.

```

[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

```

동기화 작업에 대한 로그를 모니터링합니다.

```

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

```

AIO-1 마스터의 클러스터 상태를 확인합니다.

```

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

```

```

[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 복원 절차

### CEE 및 RCM Ops-Centers에서 애플리케이션을 복원하는 단계

CEE opscenter 및 RCM opscenter를 실행 모드로 업데이트합니다.

rce301의 실행 모드를 구성합니다.

```
[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
```

몇 분 정도 기다렸다가 시스템이 100.0%인지 확인합니다.

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

rm0301의 실행 모드를 구성합니다.

```
[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>
```

몇 분 정도 기다렸다가 시스템이 100.0%인지 확인합니다.

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

rm0303의 실행 모드를 구성합니다.

```
[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>
```

몇 분 정도 기다렸다가 시스템이 100.0%인지 확인합니다.

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

## 확인 절차

이 명령을 사용하여 두 AIO 노드에서 Pod가 모두 UP 및 Running 상태인지 확인합니다.

```
on AIO nodes:
kubect1 get ns
kubect1 get pods -A -o wide
```

```
on RCM ops-centers:
rcm show-status
```