

在CNDP部署中對基於RCM的AIO伺服器進行RMA的步驟

目錄

[簡介](#)

[必要條件](#)

[需求](#)

[採用元件](#)

[瞭解RCM IP方案](#)

[備份過程](#)

[備份配置](#)

[預檢查過程](#)

[對AIO進行預檢查](#)

[預檢查輸出示例](#)

[執行過程](#)

[關閉AIO節點之前在RCM上執行的步驟](#)

[關閉AIO節點之前在Kubernetes節點上執行的步驟](#)

[伺服器維護程式](#)

[Kubernetes還原過程](#)

[在AIO節點上加電後Kubernetes節點上執行的步驟](#)

[RCM還原過程](#)

[在CEE和RCM運營中心執行恢復應用程式的步驟](#)

[驗證程式](#)

簡介

本檔案介紹在雲原生部署平台(CNDP)部署中針對基於冗餘配置管理器(RCM)的多功能一體(AIO)伺服器的返回材料授權(RMA)的詳細程式，以解決任何硬體問題或維護相關活動。

必要條件

需求

思科建議您瞭解以下主題：

- RCM
- 庫伯內特斯

採用元件

本檔案中的資訊是根據RCM版本 — rcm.2021.02.1.i18

本文中的資訊是根據特定實驗室環境內的裝置所建立。文中使用到的所有裝置皆從已清除（預設

) 的組態來啟動。如果您的網路運作中，請確保您瞭解任何指令可能造成的影響。

瞭解RCM IP方案

本檔案將說明RCM設計，包括兩個AIO節點，其中兩個RCM Openters和一個RCM CEE，每個AIO節點一個。

本文中RMA的目標RCM AIO節點是AIO-1(AI0301)，它包含處於主狀態的RCM運算子。

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

備份過程

備份配置

首先，從目標AIO節點上運行的RCM操作程式收集運行配置的配置備份。

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

從目標AIO節點上運行的RCM CEE操作程式收集運行配置。

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

預檢查過程

對AIO進行預檢查

收集兩個AIO節點的命令輸出並驗證所有Pod是否均處於Running狀態。

```
# kubectl get ns
```

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

預檢查輸出示例

請注意，兩個RCM操作員和一個在AIO-1節點上運行的RCM CEE操作中心

```
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:~$
```

登入到AIO-1的兩個RCM操作中心並驗證狀態。

```
[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#
```

在AIO-2節點上重複相同步驟，其中其他兩個RCM運算子對應於AIO-1節點。

```
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:~$
```

登入到AIO-2的兩個RCM操作中心並驗證狀態。

```
[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#
```

執行過程

關閉AIO節點之前在RCM上執行的步驟

1. 由於AIO-1上的兩個RCM都是主控制器，因此您可以將其遷移到BACKUP。

a. 為此，您必須先在活動RCM上執行`rcm migrate primary`命令，然後再關閉AIO-1伺服器。

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

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

b. 驗證AIO-1上的狀態現在是否為BACKUP。

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

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

c. 驗證AIO-2上的狀態現在是否為MASTER，並確保其為MASTER。

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

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

d. 對rm0301和rm0303執行RCM關閉。

```
[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. 我們還必須關閉AIO-1上運行的CEE操作（使用的命令）。

```
[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
```

等待幾分鐘，然後檢查系統以顯示0.0%。

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

3. 驗證RCM和CEE名稱空間沒有Pod，除了文檔、智慧代理、ops-center-rcm和ops-center-cee Pod

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

關閉AIO節點之前在Kubernetes節點上執行的步驟

排出Kubernetes節點，以便正常終止關聯的池和服務。排程式不再選擇此Kubernetes節點並從該節點中移除Pod。請一次排出單個節點。

登入到SMI群集管理器。

```
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 -

排出主節點：

```
[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
```

將主機1節點標籤為維護模式：

```
[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
```

運行群集同步並監視同步操作的日誌：

```
[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
```

群集同步日誌的輸出示例：

```
[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

```

伺服器維護程式

從CIMC正常關閉伺服器。按照硬體MoP中的定義繼續硬體相關維護活動，並確保在伺服器加電後通過所有運行狀況檢查。

附註：本文不包括伺服器的硬體或維護活動MoP，因為它們與問題陳述不同

Kubernetes還原過程

在AIO節點上加電後Kubernetes節點上執行的步驟

登入到SMI群集管理器：

```

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  -
```

關閉要重新新增到集群中的master-1的維護標誌。

[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

使用群集同步操作恢復主節點Pod和服務。

[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

監視同步操作的日誌。

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

sync logs

檢查AIO-1主機的群集狀態。

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

輸出示例：

[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還原過程

在CEE和RCM運營中心執行恢復應用程式的步驟

將CEE opscenter和RCM opscenter更新為運行模式。

配置rce301的運行模式。

```
[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
```

等待幾分鐘，檢查系統是否為100.0%。

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

配置rm0301的運行模式。

```
[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>
```

等待幾分鐘，驗證系統是否為100.0%。

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

配置rm0303的運行模式。

```
[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>
```

等待幾分鐘，檢查系統是否為100.0%。

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

驗證程式

使用以下命令驗證兩個AIO節點上的Pod是否均處於UP和Running狀態。

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

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