

CNDP導入でのRCMベースのAIOサーバのRMA手順

内容

[はじめに](#)

[前提条件](#)

[要件](#)

[使用するコンポーネント](#)

[RCM IPスキーマについて](#)

[バックアップ手順](#)

[設定のバックアップ](#)

[事前チェック手順](#)

[AIOの事前確認](#)

[サンプルPrechecks出力](#)

[実行手順](#)

[AIOノードをシャットダウンする前にRCMで実行する手順](#)

[AIOノードをシャットダウンする前にKubernetesノードで実行する手順](#)

[サーバメンテナンス手順](#)

[Kubernetesの復元手順](#)

[Kubernetesノードで実行する手順AIOノードの電源投入後](#)

[RCMの復元手順](#)

[アプリケーションを復元するためにCEEおよびRCMオペレーションセンターで実行する手順](#)

[確認手順](#)

はじめに

このドキュメントでは、ハードウェアの問題やメンテナンス関連のアクティビティに対するCloud Native Deployment Platform(CNDP)展開でのRedundancy Configuration Manager(RCM)ベースのオールインワン(AIO)サーバの返品許可(RMA)の詳細な手順について説明します。

前提条件

要件

次の項目に関する知識があることが推奨されます。

- RCMの場合
- クベルネテス

使用するコンポーネント

このドキュメントの情報は、RCMのバージョン(rcm.2021.02.1.i18)に基づくものです。

このドキュメントの情報は、特定のラボ環境にあるデバイスに基づいて作成されました。このドキュメントで使用するすべてのデバイスは、クリアな(デフォルト)設定で作業を開始しています。本稼働中のネットワークでは、各コマンドによって起こる可能性がある影響を十分確認してください。

RCM IPスキーマについて

このドキュメントでは、2つのRCM Opscenter(AIO)を持つ2つのAIOノードと、AIOノードごとに1つずつRCM CEEを持つ1つのAIOノードで構成されるRCM設計について説明します。

この記事のRMAの対象となるRCM AIOノードはAIO-1(AI0301)で、これにはPRIMARY状態の両方のRCMオスペンタが含まれています。

| 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 |

バックアップ手順

設定のバックアップ

まず、ターゲットのAIOノードで実行されているRCM opscentersからrunning-configのconfigバックアップを収集します。

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

ターゲットのAIOノードで実行されているRCM CEEオペレーションシステムからrunning-configを収集します。

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

事前チェック手順

AIOの事前確認

両方のAIOノードからコマンド出力を収集し、すべてのポッドがRunning状態であることを確認します。

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

サンプルPrechecks出力

AIO-1ノードで2つのRCMオペレーションセンターと1つのRCM CEEオペレーションセンターが稼働していることに注意してください

```
cld-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-ai0-1-master-1:~$
```

AIO-1のRCMオペレーションセンターの両方にログインし、ステータスを確認します。

```
[up0300-ai0-1/rm0301] rcm# rcm show-status
message :
{"status":[" Fri Oct 29 07:21:11 UTC 2021 : State is MASTER"]}
[up0300-ai0-1/rm0301] rcm#
```

```
[up0300-ai0-1/rm0303] rcm# rcm show-status
message :
{"status":[" Fri Oct 29 07:22:18 UTC 2021 : State is MASTER"]}
[up0300-ai0-1/rm0303] rcm#
```

AIO-1ノードに対応する他の2つのRCMのパターンが存在するAIO-2ノードで同じ手順を繰り返します。

```
cloud-user@up0300-ai0-2-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-ai0-2-master-1:~$
```

AIO-2のRCMオペレーションセンターの両方にログインし、ステータスを確認します。

```
[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#
```

実行手順

AIOノードをシャットダウンする前にRCMで実行する手順

1. AIO-1の両方のRCMがMASTERであるため、これらをBACKUPに移行できます。

a. そのためには、AIO-1サーバをシャットダウンする前に、アクティブRCMでrcm migrate primaryコマンドを実行する必要があります。

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

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

b. AIO-1のステータスがBACKUPになっていることを確認します。

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

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

c. AIO-2のステータスがMASTERになったことを確認し、MASTERになっていることを確認します。

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

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

d. rm0301とrm0303の両方でRCMシャットダウンを実行します。

```
[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. AIO-1で実行するCEE操作もシャットダウンする必要があります。使用するコマンドです。

```
[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
```

数分待つてから、システムが0.0 %と表示されていることを確認します。

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

3. documentation、smart-agent、ops-center-rcm、およびops-center-cee podsを除き、RCMおよびCEE名前空間にポッドがないことを確認します

```
# 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ノードをシャットダウンする前にKubernetesノードで実行する手順

Kubernetesノードをドレインして、関連するポッドとサービスが正常に終了するようにします。スケジューラはこのKubernetesノードを選択しなくなり、そのノードからポッドを削除します。一度に1つのノードをドレインしてください。

SMI Cluster Managerにログインします。

```
cloud-user@bot-deployer-cm-primary:~$ kubectl get svc -n smi-cm
```

| NAME | TYPE | CLUSTER-IP | EXTERNAL-IP | PORT(S) |
|---|-----------|----------------|----------------|--------------|
| cluster-files-offline-smi-cluster-deployer | ClusterIP | 10.102.108.177 | <none> | 8080/TCP |
| iso-host-cluster-files-smi-cluster-deployer | ClusterIP | 10.102.255.174 | 192.168.0.102 | 80/TCP |
| iso-host-ops-center-smi-cluster-deployer | ClusterIP | 10.102.58.99 | 192.168.0.100 | 3001/TCP |
| netconf-ops-center-smi-cluster-deployer | ClusterIP | 10.102.108.194 | 10.244.110.193 | 3022/TCP, 22 |
| ops-center-smi-cluster-deployer | ClusterIP | 10.102.156.123 | <none> | 8008/TCP, 20 |

```
squid-proxy-node-port          NodePort    10.102.73.130    <none>          3128:31677/  
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  -
```

マスターノードをドレインします。

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

マスター1ノードをメンテナンスモードにマークします。

```
[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
```

クラスタの同期を実行し、同期アクションのログを監視します。

```
[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
```

クラスタ同期ログの出力例：

```

[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
cmts-worker2           : ok=1    changed=0    unreachable=0    failed=0    skipped=2    rescued=0
cmts-worker3           : ok=1    changed=0    unreachable=0    failed=0    skipped=2    rescued=0
master1                : ok=1    changed=0    unreachable=0    failed=0    skipped=2    rescued=0
master2                : ok=1    changed=0    unreachable=0    failed=0    skipped=2    rescued=0
master3                : ok=1    changed=0    unreachable=0    failed=0    skipped=2    rescued=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

```

サーバメンテナンス手順

CIMCからサーバの電源を正常にオフにします。ハードウェアMoPで定義されているハードウェア関連のメンテナンスアクティビティを続行し、サーバの電源がオンになった後にすべてのヘルスチェックに合格することを確認します。

注：この記事では、サーバのハードウェアまたはメンテナンス作業のMoPは問題の説明とは異なる

るため、扱いません

Kubernetesの復元手順

Kubernetesノードで実行する手順AIOノードの電源投入後

SMI Cluster Managerにログインします。

```
cloud-user@bot-deployer-cm-primary:~$ kubectl get svc -n smi-cm
NAME                                TYPE             CLUSTER-IP      EXTERNAL-IP      PORT(S)
cluster-files-offline-smi-cluster-deployer ClusterIP        10.102.108.177  <none>           8080/TCP
iso-host-cluster-files-smi-cluster-deployer ClusterIP        10.102.255.174  192.168.0.102   80/TCP
iso-host-ops-center-smi-cluster-deployer ClusterIP        10.102.58.99    192.168.0.100   3001/TCP
netconf-ops-center-smi-cluster-deployer ClusterIP        10.102.108.194  10.244.110.193  3022/TCP,22
ops-center-smi-cluster-deployer      ClusterIP        10.102.156.123  <none>           8008/TCP,20
squid-proxy-node-port                NodePort        10.102.73.130   <none>           3128:31677/

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

NAME                                LOCK TO          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                      -
```

クラスタに再び追加するマスター1のメンテナンスフラグをオフにします。

```
[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
```

クラスタ同期アクションを使用して、マスターノードポッドとサービスを復元します。

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

同期操作のログを監視します。

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

AIO-1マスターのクラスタステータスを確認します。

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

出力例：

```
[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の復元手順

アプリケーションを復元するためにCEEおよびRCMオペレーションセンターで実行する手順

CEE opscenterとRCM opscenterを実行モードに更新します。

rce301の実行モードを設定します。

```
[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
```

数分待ち、システムが100.0%であることを確認します。

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

rm0301の実行モードを設定します。

```
[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>
```

数分待ち、システムが100.0%であることを確認します。

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

rm0303の実行モードを設定します。

```
[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>
```

数分待ち、システムが100.0%であることを確認します。

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

確認手順

次のコマンドを使用して、両方のAIOノードでポッドがすべてアップ状態で稼働状態であることを確認します。

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

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

翻訳について

シスコは世界中のユーザにそれぞれの言語でサポート コンテンツを提供するために、機械と人による翻訳を組み合わせて、本ドキュメントを翻訳しています。ただし、最高度の機械翻訳であっても、専門家による翻訳のような正確性は確保されません。シスコは、これら翻訳の正確性について法的責任を負いません。原典である英語版（リンクからアクセス可能）もあわせて参照することを推奨します。