

Étapes de RMA du serveur AIO basé sur le RCM dans le déploiement CNDP

Table des matières

[Introduction](#)

[Conditions préalables](#)

[Exigences](#)

[Composants utilisés](#)

[Connaître le schéma IP du RCM](#)

[Procédure de sauvegarde](#)

[Sauvegarder la configuration](#)

[Procédure De Contrôle Préalable](#)

[Prévérifications sur l'AIO](#)

[Exemple de résultats de précontrôles](#)

[Procédure D'Exécution](#)

[Étapes d'exécution sur le RCM avant l'arrêt du noeud AIO](#)

[Étapes à exécuter sur le noeud Kubernetes avant d'arrêter le noeud AIO](#)

[Procédure de maintenance du serveur](#)

[Procédure de restauration Kubernetes](#)

[Étapes à exécuter sur le noeud Kubernetes Après l'alimentation sur le noeud AIO](#)

[Procédure de restauration du RCM](#)

[Étapes à exécuter sur les centres d'opérations CEE et RCM pour restaurer l'application](#)

[Procédure de vérification](#)

Introduction

Ce document décrit la procédure détaillée pour l'autorisation de retour de matériel (RMA) pour le déploiement du serveur tout-en-un basé sur le gestionnaire de configuration de redondance (RCM) dans la plate-forme de déploiement natif cloud (CNDP) pour tout problème matériel ou toute activité liée à la maintenance.

Conditions préalables

Exigences

Cisco vous recommande de prendre connaissance des rubriques suivantes :

- MCR
- Kubernetes

Composants utilisés

Les informations contenues dans ce document sont basées sur la version du RCM - 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. Si votre réseau est en ligne, assurez-vous de bien comprendre l'incidence possible des commandes.

Connaître le schéma IP du RCM

Ce document explique la conception du RCM, qui se compose de deux noeuds AIO avec deux Opscenters RCM et un CEE RCM, un pour chaque noeud AIO.

Dans cet article, le noeud AIO cible du RCM pour la RMA est AIO-1 (AI0301), qui contient les deux opérateurs du RCM à l'état PRIMARY.

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

Procédure de sauvegarde

Sauvegarder la configuration

Pour commencer, collectez la sauvegarde de la configuration en cours à partir des opscenters du RCM qui s'exécutent sur le noeud AIO cible.

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

Collectez la configuration en cours à partir des opscenters CEE du RCM qui s'exécute sur le noeud AIO cible.

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

Procédure De Contrôle Préalable

Prévérifications sur l'AIO

Collectez le résultat de la commande à partir des deux noeuds AIO et vérifiez que tous les pods sont à l'état En cours d'exécution.

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

Exemple de résultats de précontrôles

Notez que les deux opscenteurs du RCM et un opscenteur CEE du RCM fonctionnent sur le noeud AIO-1

```
ccloud-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
ccloud-user@up0300-aio-1-master-1:~$
```

Connectez-vous à l'opscenter RCM de l'AIO-1 et vérifiez l'état.

```
[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#
```

Répétez les mêmes étapes sur le noeud AIO-2 où les deux autres opscenters RCM correspondant au noeud AIO-1 sont présents.

```
ccloud-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
ccloud-user@up0300-aio-2-master-1:~$
```

Connectez-vous à l'opscenter RCM de l'AIO-2 et vérifiez l'état.

```
[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#
```

Procédure D'Exécution

Étapes d'exécution sur le RCM avant l'arrêt du noeud AIO

1. Comme les deux RCM sur AIO-1 sont MASTER, vous pouvez les faire migrer vers BACKUP.

a. Pour ce faire, vous devez exécuter la commande `rcm migrate primary` sur les RCM actifs avant d'arrêter le serveur AIO-1.

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

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

- b. Vérifiez que l'état est maintenant BACKUP sur AIO-1.

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

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

- c. Vérifiez que l'état est maintenant MASTER sur AIO-2 et assurez-vous qu'ils sont MASTER.

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

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

- d. Arrêtez le RCM sur les routeurs rm0301 et 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. Nous devons également arrêter les opérations CEE qui s'exécutent sur l'AIO-1, commandes utilisées.

```
[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
```

Patientez quelques minutes et vérifiez que le système affiche 0,0 %.

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

3. Vérifiez qu'il n'existe pas de pods pour les espaces de noms RCM et CEE, à l'exception des pods documentation, smart-agent, ops-center-rcm et ops-center-cee

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

Étapes à exécuter sur le noeud Kubernetes avant d'arrêter le noeud AIO

Drainez le noeud Kubernetes de sorte que les pods et les services associés soient correctement terminés. Le planificateur ne sélectionnerait plus ce noeud Kubernetes et n'expulserait plus les pods de ce noeud. Déchargez un seul noeud à la fois.

Connectez-vous à SMI Cluster Manager.

```
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   -

```

Drainez le noeud maître :

```

[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

```

Marquez le noeud maître-1 en mode maintenance :

```

[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

```

Exécutez la synchronisation du cluster et surveillez les journaux pour l'action de synchronisation :

```

[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

```

Exemple de résultat pour les journaux de synchronisation de cluster :

```
[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
```

Procédure de maintenance du serveur

Mettez le serveur hors tension à partir de CIMC avec précaution. Procédez à l'activité de maintenance liée au matériel telle que définie dans la musique d'attente matérielle et assurez-vous que toutes les vérifications de l'état sont effectuées après la mise sous tension du serveur.

Remarque : cet article ne couvre pas la musique d'attente d'activité de maintenance ou de matériel pour le serveur, car elle diffère de l'énoncé du problème

Procédure de restauration Kubernetes

Étapes à exécuter sur le noeud Kubernetes Après l'alimentation sur le noeud AIO

Connectez-vous à SMI Cluster Manager :

```
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,20
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
                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  -
```

Désactivez l'indicateur de maintenance pour que le maître-1 soit à nouveau ajouté au 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
```

Restaurez les pods et les services du noeud maître avec l'action de synchronisation du cluster.

```
[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
```

Surveillez les journaux pour l'action de synchronisation.

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

Vérifiez l'état du cluster du maître AIO-1.

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

Exemple de sortie :

```
[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
```

Procédure de restauration du RCM

Étapes à exécuter sur les centres d'opérations CEE et RCM pour restaurer l'application

Mettez à jour les composants CEE opscenter et RCM opscenter en mode d'exécution.

Configurez le mode d'exécution pour 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
```

Attendez quelques minutes et vérifiez que le système est à 100,0 %.

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

Configurez le mode running pour 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>
```

Patientez quelques minutes et vérifiez que le système est à 100,0 %.

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

Configurez le mode running pour 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>
```

Attendez quelques minutes et vérifiez que le système est à 100,0 %.

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

Procédure de vérification

Vérifiez que les pods sont tous à l'état UP et Running sur les deux noeuds AIO avec ces commandes.

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

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

À propos de cette traduction

Cisco a traduit ce document en traduction automatisée vérifiée par une personne dans le cadre d'un service mondial permettant à nos utilisateurs d'obtenir le contenu d'assistance dans leur propre langue.

Il convient cependant de noter que même la meilleure traduction automatisée ne sera pas aussi précise que celle fournie par un traducteur professionnel.