在CNDP部署中对基于RCM的AIO服务器进行RMA的步骤

目录

简介

本文档介绍在云本地部署平台(CNDP)中为任何硬件问题或维护相关活动部署基于冗余配置管理器(RCM)的一体化(AIO)服务器提供退货授权(RMA)的详细过程。

先决条件

要求

Cisco 建议您了解以下主题：

- RCM
- 库贝内特斯

使用的组件

本文档中的信息基于RCM版本 — rcm.2021.02.1.i18

本文档中的信息都是基于特定实验室环境中的设备编写的。本文档中使用的所有设备最初均采用原

始（默认）配置。如果您的网络处于活动状态，请确保您了解所有命令的潜在影响。

了解RCM IP方案

本文档说明了RCM设计，该设计包括两个AIO节点，两个RCM Opscenter，一个RCM CEE，每个AIO节点。

本文中RMA的目标RCM AIO节点是AIO-1(AI0301)，它包含处于PRIMARY状态的RCM操作器。

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

备份过程

备份配置

首先，从目标AIO节点上运行的RCM运算机收集运行配置的配置备份。

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

从在目标AIO节点上运行的RCM CEE计算机收集运行配置。

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

预检程序

预检AIO

从两个AIO节点收集命令输出，并验证所有Pod是否都处于运行状态。

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

预检输出示例

请注意，AIO-1节点上运行了两台RCM机和一个RCM CEE机箱

```
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:~$
```

登录AIO-1的RCM操作中心并验证状态。

```
[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节点上重复相同的步骤，其他两个RCM运算器对应于AIO-1节点。

```
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:~$
```

登录AIO-2的RCM操作中心并验证状态。

```
[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#
```

执行过程

关闭AIO节点之前在RCM上执行的步骤

1. 由于AIO-1上的RCM都是MASTER，因此您可以将它们迁移到BACKUP。

a.为此，在关闭AIO-1服务器之前，**必须在**Active RCM上执行rcm migrate primary命令。

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

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

b.验证AIO-1上的状态现在是BACKUP。

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

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

c.验证AIO-2上的状态现在为MASTER，并确保它们为MASTER。

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

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

d.在rm0301和rm0303上执行RCM关闭。

```
[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.我们还必须关闭在AIO-1上运行的CEE操作，即使用的命令。

```
[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
```

等待几分钟，然后检查系统显示0.0%。

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

3.验证RCM和CEE命名空间是否没有可配置的命名空间，但文档、smart-agent、ops-center-rcm和ops-center-cee pod除外

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

关闭AIO节点之前在Kubernetes节点上执行的步骤

耗尽Kubernetes节点，使关联的Pod和服务正常终止。调度程序将不再选择此Kubernetes节点并从该节点中逐出pod。请一次耗尽一个节点。

登录SMI集群管理器。

```
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   -
```

耗尽主节点：

```
[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
```

将主节点1标记为维护模式：

```
[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
```

运行群集同步并监控同步操作的日志：

```
[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
```

群集同步日志的输出示例：

```
[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
```

服务器维护过程

从CIMC正常关闭服务器电源。继续执行硬件MoP中定义的与硬件相关的维护活动，并确保在服务器通电后通过所有运行状况检查。

注意：本文未介绍服务器的硬件或维护活动MoP，因为它们与问题陈述不同

Kubernetes恢复程序

在AIO节点上Kubernetes节点通电时执行的步骤

登录SMI集群管理器：

```
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   -
```

关闭要重新添加到集群的主1的维护标志。

```
[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复主节点Pod和服务。

```
[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
```

监控同步操作的日志。

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

检查AIO-1主机的集群状态。

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

示例输出：

```
[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恢复过程

在CEE和RCM Ops-Center上执行步骤以恢复应用程序

将CEE opscenter和RCM opscenter更新为运行模式。

配置rce301的运行模式。

```
[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
```

等待几分钟，检查系统为100.0%。

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

配置rm**0301**的运行模式。

```
[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>
```

等待几分钟，验证系统是否为100.0%。

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

配置rm**0303**的运行模式。

```
[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>
```

等待几分钟，检查系统为100.0%。

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

验证过程

使用这些命令验证两个AIO节点上的Pod都处于**UP**和**Running**状态。

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

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