控制器伺服器UCS C240 M4的PCRF更換

目錄

簡介

本文檔介紹在託管CPS虛擬網路功能(VNF)的Ultra-M設定中更換故障控制器伺服器所需的步驟。

必要條件

備份

在進行恢復時，思科建議使用以下步驟備份OSPD資料庫(DB):

```
[root@director ~]# mysqldump --opt --all-databases > /root/undercloud-all-databases.sql
[root@director ~]# tar --xattrs -czf undercloud-backup-`date +%F`.tar.gz /root/undercloud-all-
databases.sql
/etc/my.cnf.d/server.cnf /var/lib/glance/images /srv/node /home/stack
tar: Removing leading `/' from member names
```

初步狀態檢查

請務必檢查OpenStack環境和服務的當前狀態，並確保其處於正常狀態，然後再繼續更換過程。它有助於避免控制器更換過程中的複雜性。

步驟1.檢查OpenStack的狀態和節點清單：

```
[stack@director ~]$ source stackrc
[stack@director ~]$ openstack stack list --nested
```

```
[stack@director ~]$ ironic node-list
[stack@director ~]$ nova list
```

步驟2.檢查控制器上的心臟起搏器狀態。

登入其中一個作用中控制器並檢查心臟起搏器狀態。所有服務應在可用控制器上運行並在出現故障的控制器上停止。

```
[stack@pod1-controller-0 ~]# pcs status

<snip>
Online: [ pod1-controller-0 pod1-controller-1 ]
OFFLINE: [ pod1-controller-2 ]
Full list of resources:
ip-11.120.0.109 (ocf::heartbeat:IPaddr2): Started pod1-controller-0
ip-172.25.22.109 (ocf::heartbeat:IPaddr2): Started pod1-controller-1
ip-192.200.0.107 (ocf::heartbeat:IPaddr2): Started pod1-controller-0

Clone Set: haproxy-clone [haproxy]
Started: [ pod1-controller-0 pod1-controller-1 ]
Stopped: [ pod1-controller-2 ]

Master/Slave Set: galera-master [galera]
Masters: [ pod1-controller-0 pod1-controller-1 ]
Stopped: [ pod1-controller-2 ]
ip-11.120.0.110 (ocf::heartbeat:IPaddr2): Started pod1-controller-0
ip-11.119.0.110 (ocf::heartbeat:IPaddr2): Started pod1-controller-1

Clone Set: rabbitmq-clone [rabbitmq]
Started: [ pod1-controller-0 pod1-controller-1 ]
Stopped: [ pod1-controller-2 ]

Master/Slave Set: redis-master [redis]
Masters: [ pod1-controller-0 ]
Slaves: [ pod1-controller-1 ]
Stopped: [ pod1-controller-2 ]

ip-11.118.0.104 (ocf::heartbeat:IPaddr2): Started pod1-controller-1
openstack-cinder-volume (systemd:openstack-cinder-volume): Started pod1-controller-0

my-ipmilan-for-controller-6 (stonith:fence_ipmilan): Started pod1-controller-1
my-ipmilan-for-controller-4 (stonith:fence_ipmilan): Started pod1-controller-0
my-ipmilan-for-controller-7 (stonith:fence_ipmilan): Started pod1-controller-0

Failed Actions:
Daemon Status:

corosync: active/enabled
pacemaker: active/enabled
pcsd: active/enabled
```

在此範例中，Controller-2處於離線狀態。因此，它將被取代。Controller-0和Controller-1運行正常且運行群集服務。

步驟3.檢查作用中控制器的MariaDB狀態。

```
[stack@director] nova list | grep control
| 4361358a-922f-49b5-89d4-247a50722f6d | pod1-controller-0 | ACTIVE | - | Running |
ctlplane=192.200.0.102 |
| d0f57f27-93a8-414f-b4d8-957de0d785fc | pod1-controller-1 | ACTIVE | - | Running |
```

```
ctlplane=192.200.0.110 |

[stack@director ~]$ for i in 192.200.0.102 192.200.0.110 ; do echo "*** $i ***" ; ssh heat-
admin@$i "sudo mysql --exec=\"SHOW STATUS LIKE 'wsrep_local_state_comment'\" ; sudo mysql --
exec=\"SHOW STATUS LIKE 'wsrep_cluster_size'\""; done
*** 192.200.0.152 ***
Variable_name     Value
wsrep_local_state_comment  Synced
Variable_name     Value
wsrep_cluster_size         2
*** 192.200.0.154 ***
Variable_name     Value
wsrep_local_state_comment  Synced
Variable_name     Value
wsrep_cluster_size         2
```

驗證每個作用中控制器是否存在以下線路：

wsrep_local_state_comment:已同步

wsrep_cluster_size:2

步驟4.檢查作用中控制器的Rabbitmq狀態。發生故障的控制器不應出現在運行的節點清單中。

```
[heat-admin@pod1-controller-0 ~] sudo rabbitmqctl cluster_status
Cluster status of node 'rabbit@pod1-controller-0' ...
[{nodes,[{disc,['rabbit@pod1-controller-0','rabbit@pod1-controller-1',
               'rabbit@pod1-controller-2']}]},
{running_nodes,['rabbit@pod1-controller-1',
                'rabbit@pod1-controller-0']},
 {cluster_name,<<"rabbit@pod1-controller-2.localdomain">>},
 {partitions,[]},
 {alarms,[{'rabbit@pod1-controller-1',[]},
         {'rabbit@pod1-controller-0',[]}]}]


[heat-admin@pod1-controller-1 ~] sudo rabbitmqctl cluster_status
Cluster status of node 'rabbit@pod1-controller-1' ...
[{nodes,[{disc,['rabbit@pod1-controller-0','rabbit@pod1-controller-1',
               'rabbit@pod1-controller-2']}]},
 {running_nodes,['rabbit@pod1-controller-0',
                 'rabbit@pod1-controller-1']},
 {cluster_name,<<"rabbit@pod1-controller-2.localdomain">>},
 {partitions,[]},
 {alarms,[{'rabbit@pod1-controller-0',[]},
         {'rabbit@pod1-controller-1',[]}]}]
```

步驟5.檢查是否所有底層雲服務都處於OSP-D節點的已載入、活動和運行狀態。

```
[stack@director ~]$ systemctl list-units "openstack*" "neutron*" "openvswitch*"

UNIT                                    LOAD   ACTIVE SUB     DESCRIPTION

neutron-dhcp-agent.service              loaded active running OpenStack Neutron DHCP Agent
neutron-openvswitch-agent.service       loaded active running OpenStack Neutron Open vSwitch
Agent
neutron-ovs-cleanup.service             loaded active exited  OpenStack Neutron Open vSwitch
Cleanup Utility
neutron-server.service                  loaded active running OpenStack Neutron Server
openstack-aodh-evaluator.service        loaded active running OpenStack Alarm evaluator
service
```

```
openstack-aodh-listener.service           loaded active running OpenStack Alarm listener
service
openstack-aodh-notifier.service           loaded active running OpenStack Alarm notifier
service
openstack-ceilometer-central.service      loaded active running OpenStack ceilometer central
agent
openstack-ceilometer-collector.service    loaded active running OpenStack ceilometer collection
service
openstack-ceilometer-notification.service  loaded active running OpenStack ceilometer
notification agent
openstack-glance-api.service              loaded active running OpenStack Image Service (code-
named Glance) API server
openstack-glance-registry.service         loaded active running OpenStack Image Service (code-
named Glance) Registry server
openstack-heat-api-cfn.service            loaded active running Openstack Heat CFN-compatible
API Service
openstack-heat-api.service                loaded active running OpenStack Heat API Service
openstack-heat-engine.service             loaded active running Openstack Heat Engine Service
openstack-ironic-api.service              loaded active running OpenStack Ironic API service
openstack-ironic-conductor.service        loaded active running OpenStack Ironic Conductor
service
openstack-ironic-inspector-dnsmasq.service loaded active running PXE boot dnsmasq service for
Ironic Inspector
openstack-ironic-inspector.service        loaded active running Hardware introspection service
for OpenStack Ironic
openstack-mistral-api.service             loaded active running Mistral API Server
openstack-mistral-engine.service          loaded active running Mistral Engine Server
openstack-mistral-executor.service        loaded active running Mistral Executor Server
openstack-nova-api.service                loaded active running OpenStack Nova API Server
openstack-nova-cert.service               loaded active running OpenStack Nova Cert Server
openstack-nova-compute.service            loaded active running OpenStack Nova Compute Server
openstack-nova-conductor.service          loaded active running OpenStack Nova Conductor Server
openstack-nova-scheduler.service          loaded active running OpenStack Nova Scheduler Server
openstack-swift-account-reaper.service    loaded active running OpenStack Object Storage
(swift) - Account Reaper
openstack-swift-account.service           loaded active running OpenStack Object Storage
(swift) - Account Server
openstack-swift-container-updater.service  loaded active running OpenStack Object Storage
(swift) - Container Updater
openstack-swift-container.service         loaded active running OpenStack Object Storage
(swift) - Container Server
openstack-swift-object-updater.service    loaded active running OpenStack Object Storage
(swift) - Object Updater
openstack-swift-object.service            loaded active running OpenStack Object Storage
(swift) - Object Server
openstack-swift-proxy.service             loaded active running OpenStack Object Storage
(swift) - Proxy Server
openstack-zaqar.service                   loaded active running OpenStack Message Queuing
Service (code-named Zaqar) Server
openstack-zaqar@1.service                 loaded active running OpenStack Message Queuing
Service (code-named Zaqar) Server Instance 1
openvswitch.service                       loaded active exited  Open vSwitch

LOAD   = Reflects whether the unit definition was properly loaded.
ACTIVE = The high-level unit activation state, i.e. generalization of SUB.
SUB    = The low-level unit activation state, values depend on unit type.

37 loaded units listed. Pass --all to see loaded but inactive units, too.
To show all installed unit files use 'systemctl list-unit-files'.
```

在控制器群集中禁用隔離功能

```
[root@pod1-controller-0 ~]# sudo pcs property set stonith-enabled=false
[root@pod1-controller-0 ~]# pcs property show

Cluster Properties:
 cluster-infrastructure: corosync
 cluster-name: tripleo_cluster
 dc-version: 1.1.15-11.el7_3.4-e174ec8
 have-watchdog: false
 last-lrm-refresh: 1510809585
 maintenance-mode: false
 redis_REPL_INFO: pod1-controller-0
stonith-enabled: false

Node Attributes:
 pod1-controller-0: rmq-node-attr-last-known-rabbitmq=rabbit@pod1-controller-0
 pod1-controller-1: rmq-node-attr-last-known-rabbitmq=rabbit@pod1-controller-1
 pod1-controller-2: rmq-node-attr-last-known-rabbitmq=rabbit@pod1-controller-2
```

安裝新控制器節點

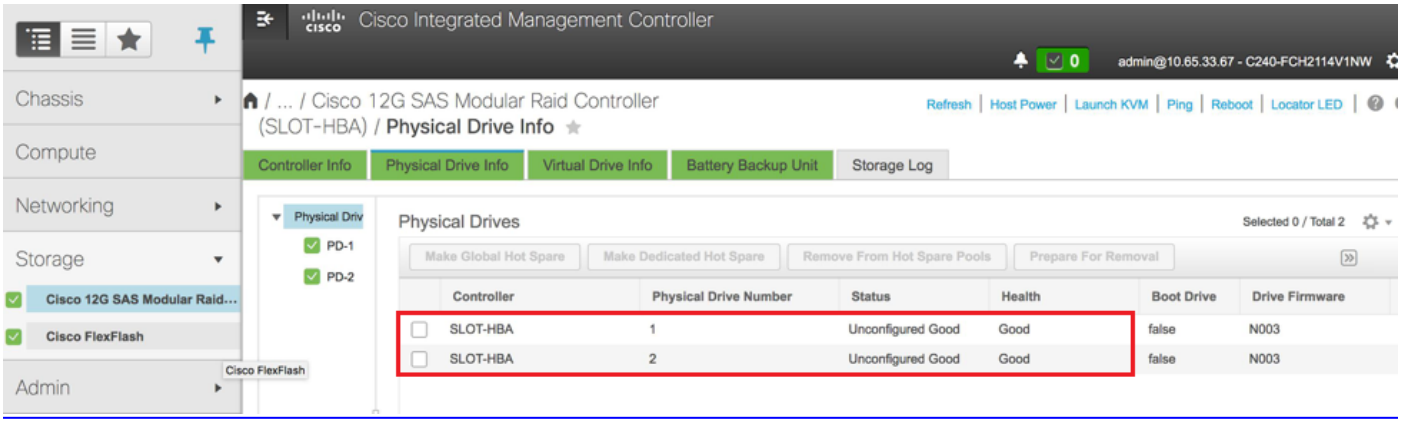
步驟1.安裝新UCS C240 M4伺服器的步驟和初始設定步驟可從Cisco UCS C240 M4伺服器安裝和服務指南中參考

步驟2.使用CIMC IP登入伺服器。

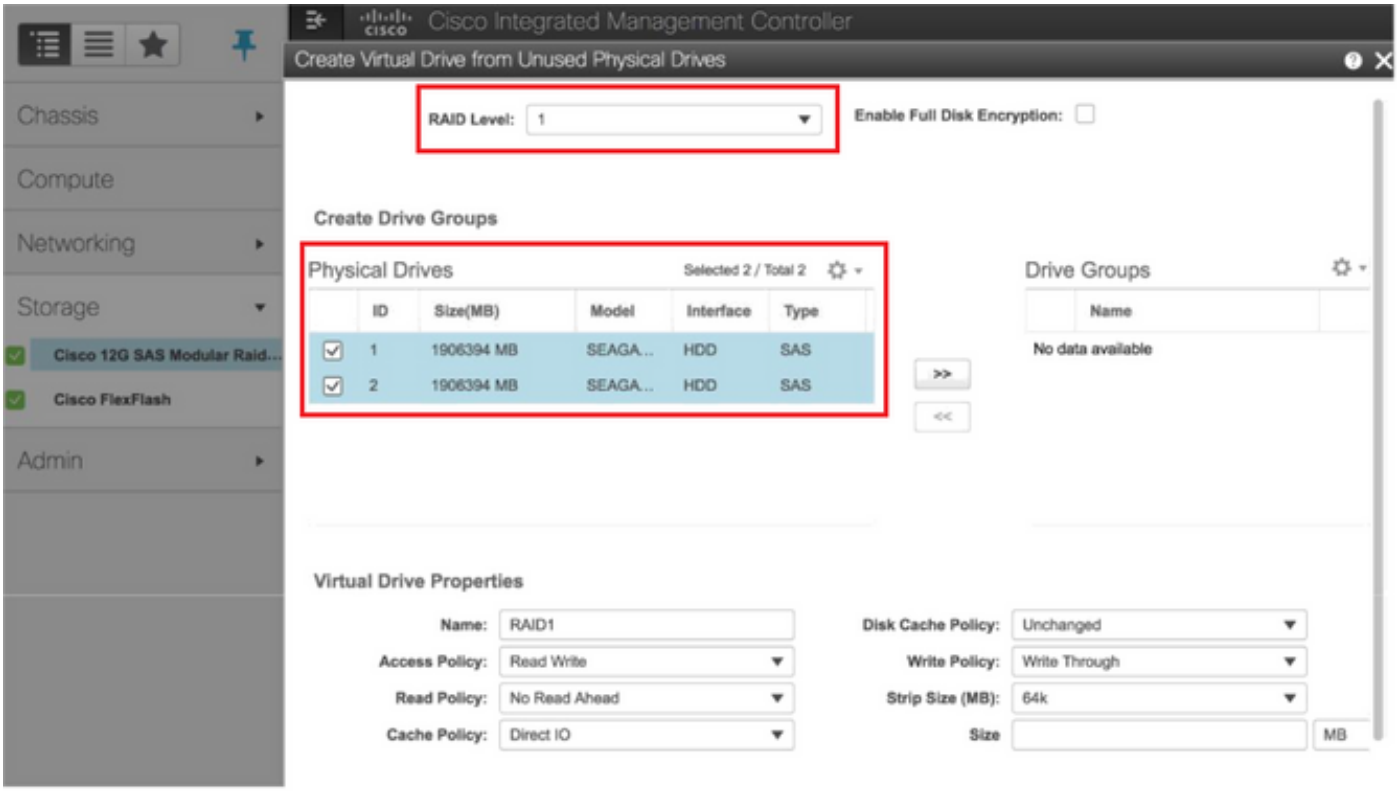
步驟3.如果韌體與以前使用的推薦版本不一致，請執行BIOS升級。BIOS升級步驟如下：

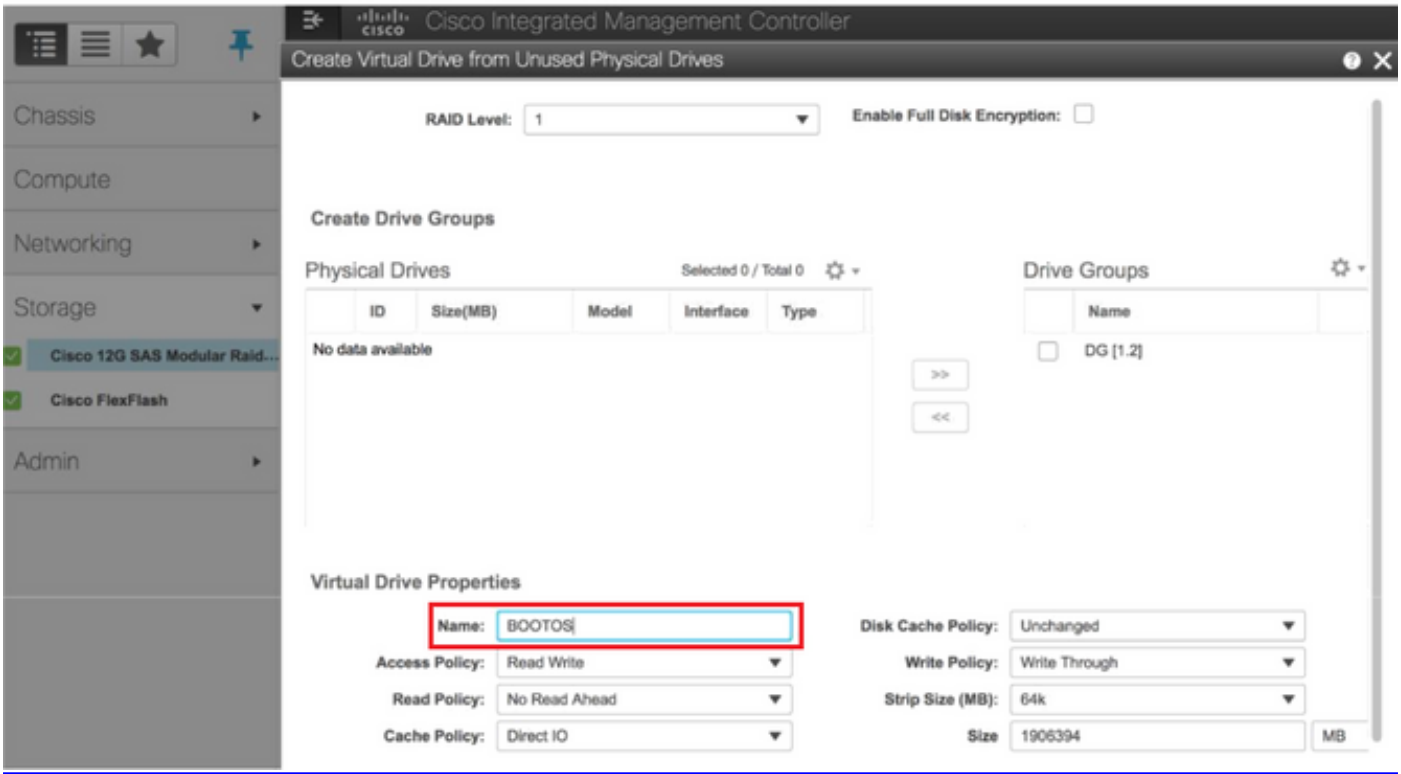
Cisco UCS C系列機架式伺服器BIOS升級指南

步驟4.檢驗物理驅動器的狀態。必須是**Unconfigured Good**。導航到**儲存> Cisco 12G SAS模組化Raid控制器(SLOT-HBA)>物理驅動器資訊。**

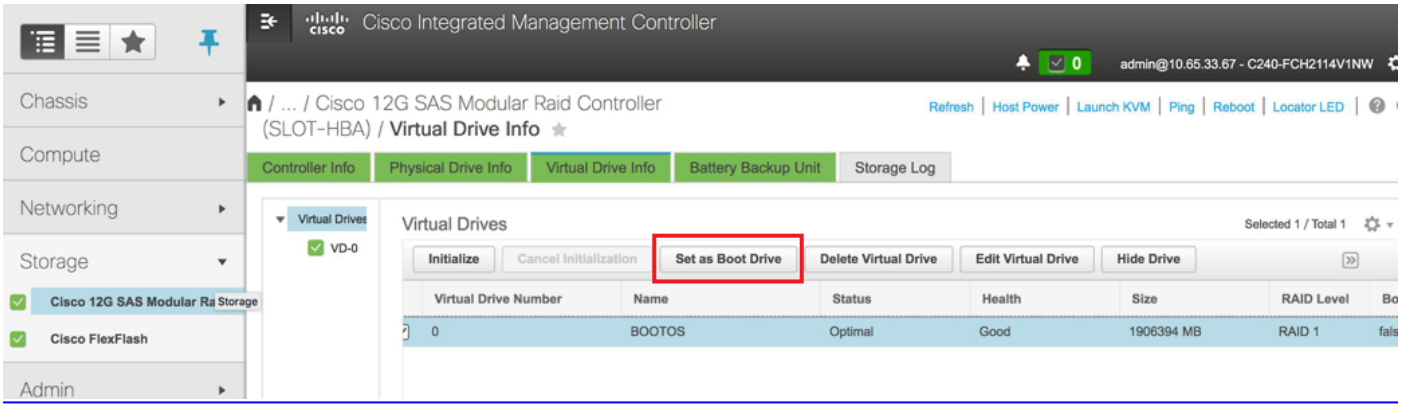


步驟5.要從具有RAID級別1的物理驅動器建立虛擬驅動器：導覽至**Storage > Cisco 12G SAS Modular Raid Controller(SLOT-HBA)> Controller Info > Create Virtual Drive from Unused Physical Drives**，如下圖所示。

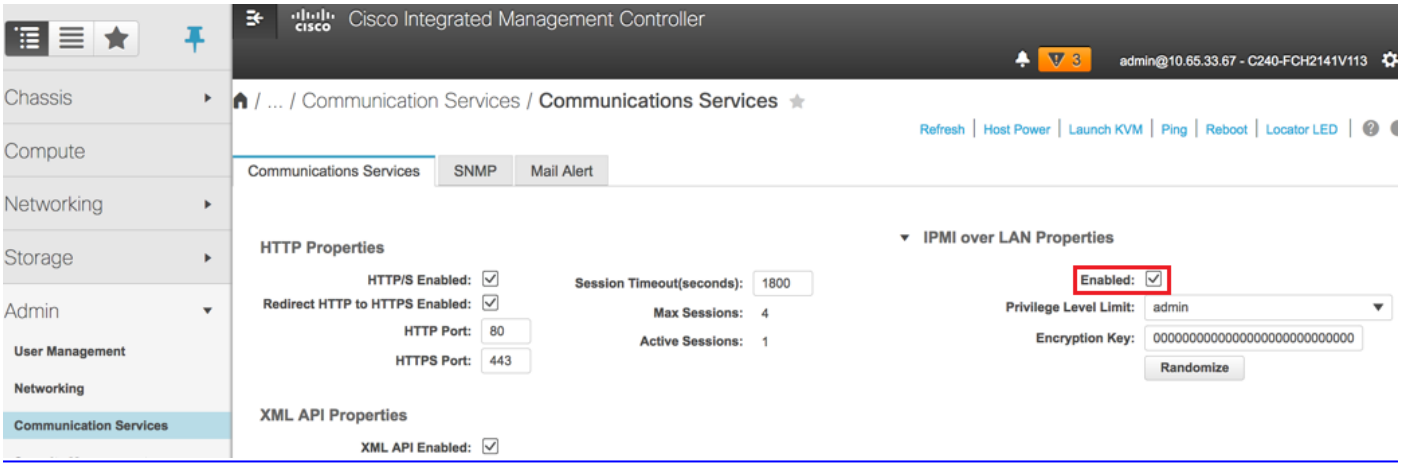




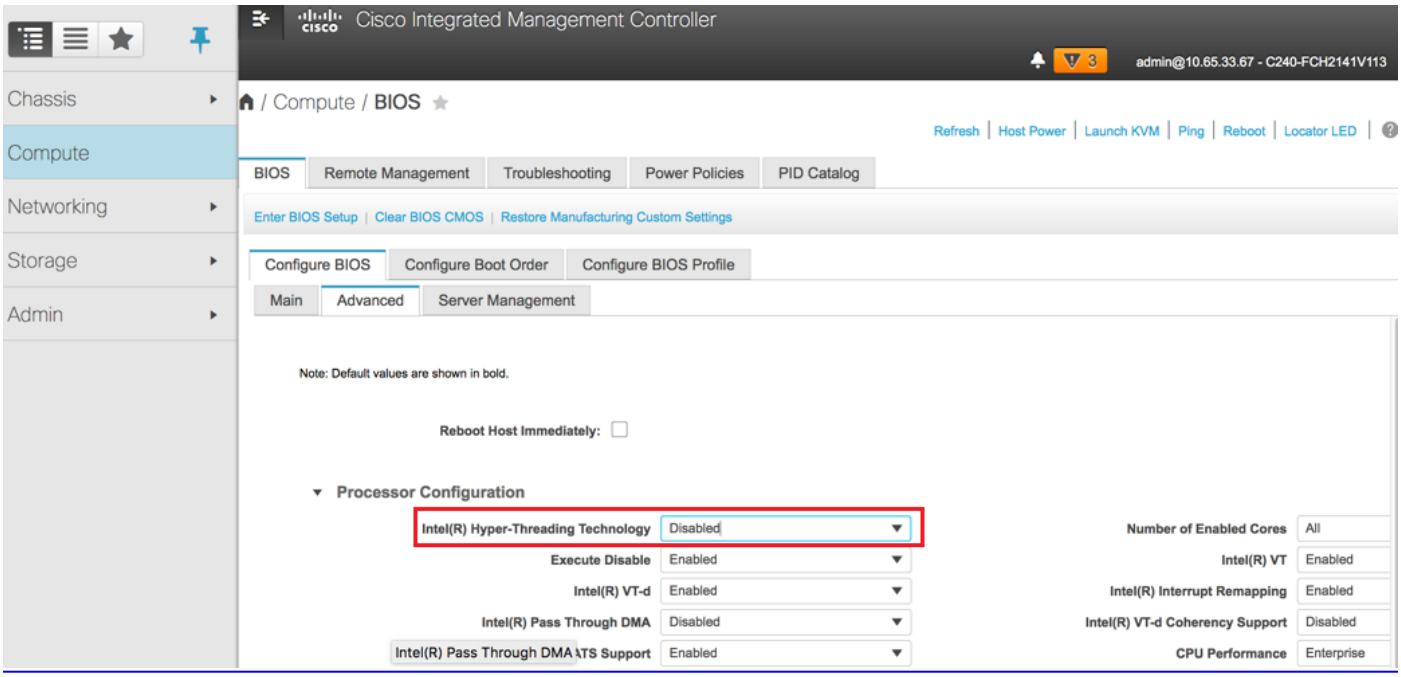
- 選擇VD並配置**Set as Boot Drive:**



步驟6.要啟用IPMI over LAN，請導航到**Admin > Communication Services > Communication Services**。



步驟7。若要停用超執行緒，請導覽至**Compute > BIOS > Configure BIOS > Advanced > Processor Configuration**，如下圖所示。



附註：此處顯示影象，本節中提到的配置步驟是參照韌體版本3.0(3e)進行的，如果您使用其他版本，可能會有細微的差異。

Overcloud中的控制器節點更換

本節說明使用重疊雲中的新控制器替換有故障的控制器所需的步驟。為此，將重新使用用於啟動堆疊的**deploy.sh**指令碼。在部署時，在ControllerNodesPostDeployment階段，由於Puppet模組中的某些限制，更新將失敗。重新啟動部署指令碼之前需要手動干預。

準備刪除失敗的控制器節點

步驟1.識別發生故障的控制器的索引。索引是OpenStack伺服器清單輸出中控制器名稱的數字字尾。在本例中，索引為2:

```
[stack@director ~]$ nova list | grep controller
| 5813a47e-af27-4fb9-8560-75decd3347b4 | pod1-controller-0  | ACTIVE | -          | Running
| ctlplane=192.200.0.152 |
| 457f023f-d077-45c9-bbea-dd32017d9708 | pod1-controller-1  | ACTIVE | -          | Running
| ctlplane=192.200.0.154 |
| d13bb207-473a-4e42-a1e7-05316935ed65 | pod1-controller-2  | ACTIVE | -          | Running
| ctlplane=192.200.0.151 |
```

步驟2.建立定義要**刪除的**節點的Yaml檔案~templates/remove-controller.yaml。將上一步中找到的索引用於資源清單中的條目。

```
[stack@director ~]$ cat templates/remove-controller.yaml

parameters:
 ControllerRemovalPolicies:
   [{'resource_list': ['2']}]

parameter_defaults:
 CorosyncSettleTries: 5
```

步驟3.製作用於安裝重疊雲的部署指令碼的副本，並插入一行，以便包含之前建立的**remove-controller.yaml**檔案。

```
[stack@director ~]$ cp deploy.sh deploy-removeController.sh
[stack@director ~]$ cat deploy-removeController.sh
time openstack overcloud deploy --templates \
-r ~/custom-templates/custom-roles.yaml \
-e /home/stack/templates/remove-controller.yaml \
-e /usr/share/openstack-tripleo-heat-templates/environments/puppet-pacemaker.yaml \
-e /usr/share/openstack-tripleo-heat-templates/environments/network-isolation.yaml \
-e /usr/share/openstack-tripleo-heat-templates/environments/storage-environment.yaml \
-e /usr/share/openstack-tripleo-heat-templates/environments/neutron-sriov.yaml \
-e ~/custom-templates/network.yaml \
-e ~/custom-templates/ceph.yaml \
-e ~/custom-templates/compute.yaml \
-e ~/custom-templates/layout-removeController.yaml \
-e ~/custom-templates/rabbitmq.yaml \
--stack pod1 \
--debug \
--log-file overcloudDeploy_$(date +%m_%d_%y__%H_%M_%S).log \
--neutron-flat-networks phys_pcie1_0,phys_pcie1_1,phys_pcie4_0,phys_pcie4_1 \
--neutron-network-vlan-ranges datacentre:101:200 \
--neutron-disable-tunneling \
--verbose --timeout 180
```

步驟4.使用此處提及的命令，確定要替換的控制器的ID，並將其移至維護模式。

```
[stack@director ~]$ nova list | grep controller

| 5813a47e-af27-4fb9-8560-75decd3347b4 | pod1-controller-0  | ACTIVE | -          | Running
| ctlplane=192.200.0.152 |

| 457f023f-d077-45c9-bbea-dd32017d9708 | pod1-controller-1  | ACTIVE | -          | Running
| ctlplane=192.200.0.154 |

| d13bb207-473a-4e42-a1e7-05316935ed65 | pod1-controller-2  | ACTIVE | -          | Running
| ctlplane=192.200.0.151 |


[stack@director ~]$ openstack baremetal node list | grep d13bb207-473a-4e42-a1e7-05316935ed65

| e7c32170-c7d1-4023-b356-e98564a9b85b | None | d13bb207-473a-4e42-a1e7-05316935ed65 | power
off   | active             | False       |


[stack@b10-ospd ~]$ openstack baremetal node maintenance set e7c32170-c7d1-4023-b356-
e98564a9b85b


[stack@director~]$ openstack baremetal node list | grep True

| e7c32170-c7d1-4023-b356-e98564a9b85b | None | d13bb207-473a-4e42-a1e7-05316935ed65 | power
off   | active             | True        |
```

步驟5. 為了確保資料庫在替換過程時運行，請從起搏器控制元件中刪除Galera，並在其中一個活動控制器上運行此命令。

```
[root@pod1-controller-0 ~]# sudo pcs resource unmanage galera
[root@pod1-controller-0 ~]# sudo pcs status

Cluster name: tripleo_cluster
Stack: corosync
Current DC: pod1-controller-0 (version 1.1.15-11.el7_3.4-e174ec8) - partition with quorum
Last updated: Thu Nov 16 16:51:18 2017                  Last change: Thu Nov 16 16:51:12 2017
by root via crm_resource on pod1-controller-0
3 nodes and 22 resources configured
Online: [ pod1-controller-0 pod1-controller-1 ]
OFFLINE: [ pod1-controller-2 ]

Full list of resources:

 ip-11.120.0.109          (ocf::heartbeat:IPaddr2):        Started pod1-controller-0
 ip-172.25.22.109        (ocf::heartbeat:IPaddr2):        Started pod1-controller-1
 ip-192.200.0.107        (ocf::heartbeat:IPaddr2):        Started pod1-controller-0

 Clone Set: haproxy-clone [haproxy]
    Started: [ pod1-controller-0 pod1-controller-1 ]
    Stopped: [ pod1-controller-2 ]

Master/Slave Set: galera-master [galera] (unmanaged)
    galera         (ocf::heartbeat:galera):          Master pod1-controller-0 (unmanaged)
    galera         (ocf::heartbeat:galera):          Master pod1-controller-1 (unmanaged)

    Stopped: [ pod1-controller-2 ]
 ip-11.120.0.110          (ocf::heartbeat:IPaddr2):        Started pod1-controller-0
 ip-11.119.0.110          (ocf::heartbeat:IPaddr2):        Started pod1-controller-1

<snip>
```

準備新增新控制器節點

步驟1。建立一個controllerRMA.json檔案，並僅包含新的控制器詳細資訊。請確保以前未使用過新控制器上的索引號。通常，遞增到下一個最高控制器編號。

範例：之前的版本最高的是Controller-2，因此請建立Controller-3。

附註：請記住json格式。

```
[stack@director ~]$ cat controllerRMA.json
{
   "nodes": [
       {
           "mac": [
               <MAC_ADDRESS>
           ],
           "capabilities": "node:controller-3,boot_option:local",
           "cpu": "24",
           "memory": "256000",
           "disk": "3000",
           "arch": "x86_64",
           "pm_type": "pxe_ipmitool",
           "pm_user": "admin",
           "pm_password": "<PASSWORD>",
           "pm_addr": "<CIMC_IP>"
       }
   ]
}
```

步驟2.使用在上一步中建立的json檔案匯入新節點。

```
[stack@director ~]$ openstack baremetal import --json controllerRMA.json

Started Mistral Workflow. Execution ID: 67989c8b-1225-48fe-ba52-3a45f366e7a0

Successfully registered node UUID 048ccb59-89df-4f40-82f5-3d90d37ac7dd

Started Mistral Workflow. Execution ID: c6711b5f-fa97-4c86-8de5-b6bc7013b398

Successfully set all nodes to available.

[stack@director ~]$ openstack baremetal node list | grep available

| 048ccb59-89df-4f40-82f5-3d90d37ac7dd | None | None                                    | power
off   | available          | False
```

步驟3.將節點設定為管理狀態。

```
[stack@director ~]$ openstack baremetal node manage 048ccb59-89df-4f40-82f5-3d90d37ac7dd
[stack@director ~]$ openstack baremetal node list | grep off
| 048ccb59-89df-4f40-82f5-3d90d37ac7dd | None | None | power off | manageable | False |
```

步驟4.運行內檢。

```
[stack@director ~]$ openstack overcloud node introspect 048ccb59-89df-4f40-82f5-3d90d37ac7dd --
provide
Started Mistral Workflow. Execution ID: f73fb275-c90e-45cc-952b-bfc25b9b5727
```

```
Waiting for introspection to finish...
Successfully introspected all nodes.
Introspection completed.
Started Mistral Workflow. Execution ID: a892b456-eb15-4c06-b37e-5bc3f6c37c65
Successfully set all nodes to available

[stack@director ~]$ openstack baremetal node list | grep available
| 048ccb59-89df-4f40-82f5-3d90d37ac7dd | None | None                                  | power
off   | available          | False       |
```

步驟5.使用新的控制器屬性標籤可用節點。確保使用為新控制器指定的控制器ID，如controllerRMA.json檔案中使用。

```
[stack@director ~]$ openstack baremetal node set --property capabilities='node:controller-
3,profile:control,boot_option:local' 048ccb59-89df-4f40-82f5-3d90d37ac7dd
```

步驟6.在部署指令碼中，有一個名為layout.yaml的自定義模板，該模板除了其他功能之外，還指定將哪些IP地址分配給各個介面的控制器。在新堆疊上，為Controller-0、Controller-1和Controller-2定義了3個地址。新增新控制器時，請確保按順序為每個子網新增下一個IP地址。

```
ControllerIPs:
internal_api:
- 11.120.0.10
- 11.120.0.11
- 11.120.0.12
 - 11.120.0.13
tenant:
- 11.117.0.10
- 11.117.0.11
- 11.117.0.12
 - 11.117.0.13
storage:
- 11.118.0.10
- 11.118.0.11
- 11.118.0.12
 - 11.118.0.13
storage_mgmt:
- 11.119.0.10
- 11.119.0.11
- 11.119.0.12
 - 11.119.0.13
```

步驟7.現在運行之前建立的deploy-removecontroller.sh，以刪除舊節點並新增新節點。

附註：在ControllerNodesDeployment_Step1中，此步驟預期失敗。此時，需要手動干預。

```
[stack@b10-ospd ~]$ ./deploy-addController.sh
START with options: [u'overcloud', u'deploy', u'--templates', u'-r', u'/home/stack/custom-
templates/custom-roles.yaml', u'-e', u'/usr/share/openstack-tripleo-heat-
templates/environments/puppet-pacemaker.yaml', u'-e', u'/usr/share/openstack-tripleo-heat-
templates/environments/network-isolation.yaml', u'-e', u'/usr/share/openstack-tripleo-heat-
templates/environments/storage-environment.yaml', u'-e', u'/usr/share/openstack-tripleo-heat-
templates/environments/neutron-sriov.yaml', u'-e', u'/home/stack/custom-templates/network.yaml',
u'-e', u'/home/stack/custom-templates/ceph.yaml', u'-e', u'/home/stack/custom-
templates/compute.yaml', u'-e', u'/home/stack/custom-templates/layout-removeController.yaml',
u'-e', u'/home/stack/custom-templates/rabbitmq.yaml', u'--stack', u'newtonoc', u'--debug', u'--
log-file', u'overcloudDeploy_11_15_17__07_46_35.log', u'--neutron-flat-networks',
u'phys_pcie1_0,phys_pcie1_1,phys_pcie4_0,phys_pcie4_1', u'--neutron-network-vlan-ranges',
u'datacentre:101:200', u'--neutron-disable-tunneling', u'--verbose', u'--timeout', u'180']
 :
```

```
DeploymentError: Heat Stack update failed
END return value: 1

real     42m1.525s
user     0m3.043s
sys      0m0.614s
```

可以使用以下命令監控部署的進度/狀態：

```
[stack@director~]$ openstack stack list --nested | grep -iv complete

+--------------------------------------+------------------------------------------------------------
-------------------------------------------------------------------------------------------------
-----------------+-----------------+----------------------+----------------------+--------------
------------------------+

| ID                                   | Stack
Name
                                                                  | Stack Status    | Creation
Time        | Updated Time         | Parent                                 |

+--------------------------------------+------------------------------------------------------------
-------------------------------------------------------------------------------------------------
-----------------+-----------------+----------------------+----------------------+--------------
------------------------+

| c1e338f2-877e-4817-93b4-9a3f0c0b3d37 | pod1-AllNodesDeploySteps-5psegydpwxij-
ComputeDeployment_Step1-swnuzjixac43
                                                                      | UPDATE_FAILED   |
2017-10-08T14:06:07Z | 2017-11-16T18:09:43Z | e90f00ef-2499-4ec3-90b4-d7def6e97c47 |

| 1db4fef4-45d3-4125-bd96-2cc3297a69ff | pod1-AllNodesDeploySteps-5psegydpwxij-
ControllerDeployment_Step1-
hmn3hpruubcn
   | UPDATE_FAILED   | 2017-10-08T14:03:05Z | 2017-11-16T18:12:12Z | e90f00ef-2499-4ec3-90b4-
d7def6e97c47 |

| e90f00ef-2499-4ec3-90b4-d7def6e97c47 | pod1-AllNodesDeploySteps-
5psegydpwxij
                                          | UPDATE_FAILED   | 2017-10-08T13:59:25Z | 2017-11-
16T18:09:25Z | 6c4b604a-55a4-4a19-9141-28c844816c0d |

| 6c4b604a-55a4-4a19-9141-28c844816c0d |
pod1
                                                                  | UPDATE_FAILED   | 2017-10-
08T12:37:11Z | 2017-11-16T17:35:35Z | None                                   |

+--------------------------------------+------------------------------------------------------------
-------------------------------------------------------------------------------------------------
-----------------+-----------------+----------------------+----------------------+--------------
------------------------+
```

手動干預

步驟1.在OSP-D伺服器上，執行OpenStack server list指令以列出可用控制器。新增的控制器應該會出現在清單中。

```
[stack@director ~]$ openstack server list | grep controller
| 3e6c3db8-ba24-48d9-b0e8-1e8a2eb8b5ff | pod1-controller-3  | ACTIVE | ctlplane=192.200.0.103 |
overcloud-full |
| 457f023f-d077-45c9-bbea-dd32017d9708 | pod1-controller-1  | ACTIVE | ctlplane=192.200.0.154 |
```

```
overcloud-full |
| 5813a47e-af27-4fb9-8560-75decd3347b4 | pod1-controller-0  | ACTIVE | ctlplane=192.200.0.152 |
overcloud-full |
```

步驟2.連線至其中一個作用中控制器（不是新增的控制器），並檢視檔案/etc/corosync/corosycn.conf。找到nodelist，該清單為每個控制器分配nodeid。找到失敗節點的條目，並記下其nodeid:

```
[root@pod1-controller-0 ~]# cat /etc/corosync/corosync.conf
totem {
   version: 2
   secauth: off
   cluster_name: tripleo_cluster
   transport: udpu
   token: 10000
}

nodelist {
   node {
       ring0_addr: pod1-controller-0
       nodeid: 5
   }
   node {
       ring0_addr: pod1-controller-1
       nodeid: 7
   }
   node {
       ring0_addr: pod1-controller-2
       nodeid: 8
   }
}
```

步驟3.登入每個作用中控制器。刪除出現故障的節點並重新啟動服務。在這種情況下，請移除pod1-controller-2。不要在新新增的控制器上執行此操作。

```
[root@pod1-controller-0 ~]# sudo pcs cluster localnode remove pod1-controller-2
pod1-controller-2: successfully removed!
[root@pod1-controller-0 ~]# sudo pcs cluster reload corosync
Corosync reloaded

[root@pod1-controller-1 ~]# sudo pcs cluster localnode remove pod1-controller-2
pod1-controller-2: successfully removed!
[root@pod1-controller-1 ~]# sudo pcs cluster reload corosync
Corosync reloaded
```

步驟4. 從其中一個活動控制器運行此命令，以便從群集中刪除故障節點。

```
[root@pod1-controller-0 ~]# sudo crm_node -R pod1-controller-2 --force
```

步驟5.從其中一個活動控制器運行此命令，以便從rabbitmq群集中刪除故障節點。

```
[root@pod1-controller-0 ~]# sudo rabbitmqctl forget_cluster_node rabbit@pod1-controller-2
Removing node 'rabbit@newtonoc-controller-2' from cluster ...
```

步驟6.從MongoDB中刪除故障節點。為此，您需要找到活動的Mongo節點。使用netstat查詢主機的IP地址。

```
[root@pod1-controller-0 ~]# sudo netstat -tulnp | grep 27017
tcp        0      0 11.120.0.10:27017       0.0.0.0:*               LISTEN
219577/mongod
```

步驟7.使用先前命令的IP位址和連線埠號碼登入節點並檢查其是否為主機。

```
[heat-admin@pod1-controller-0 ~]$ echo "db.isMaster()" | mongo --host 11.120.0.10:27017
MongoDB shell version: 2.6.11
connecting to: 11.120.0.10:27017/test
{
        "setName" : "tripleo",
        "setVersion" : 9,
        "ismaster" : true,
        "secondary" : false,
        "hosts" : [
                "11.120.0.10:27017",
                "11.120.0.12:27017",
                "11.120.0.11:27017"
        ],
        "primary" : "11.120.0.10:27017",
        "me" : "11.120.0.10:27017",
        "electionId" : ObjectId("5a0d2661218cb0238b582fb1"),
        "maxBsonObjectSize" : 16777216,
        "maxMessageSizeBytes" : 48000000,
        "maxWriteBatchSize" : 1000,
        "localTime" : ISODate("2017-11-16T18:36:34.473Z"),
        "maxWireVersion" : 2,
        "minWireVersion" : 0,
        "ok" : 1
}
```

如果節點不是主節點，請登入到另一個活動控制器並執行相同步驟。

步驟8.在主機上使用**rs.status()**命令列**出可用節**點。查詢舊/無響應節點並標識mongo節點名稱。

```
[root@pod1-controller-0 ~]# mongo --host 11.120.0.10
MongoDB shell version: 2.6.11
connecting to: 11.120.0.10:27017/test
<snip>
tripleo:PRIMARY> rs.status()
{
        "set" : "tripleo",
        "date" : ISODate("2017-11-14T13:27:14Z"),
        "myState" : 1,
        "members" : [
                {
                        "_id" : 0,
                        "name" : "11.120.0.10:27017",
                        "health" : 1,
                        "state" : 1,
                        "stateStr" : "PRIMARY",
                        "uptime" : 418347,
                        "optime" : Timestamp(1510666033, 1),
                        "optimeDate" : ISODate("2017-11-14T13:27:13Z"),
                        "electionTime" : Timestamp(1510247693, 1),
                        "electionDate" : ISODate("2017-11-09T17:14:53Z"),
                        "self" : true
                },
                {
                        "_id" : 2,
                        "name" : "11.120.0.12:27017",
```

```
                        "health" : 1,
                        "state" : 2,
                        "stateStr" : "SECONDARY",
                        "uptime" : 418347,
                        "optime" : Timestamp(1510666033, 1),
                        "optimeDate" : ISODate("2017-11-14T13:27:13Z"),
                        "lastHeartbeat" : ISODate("2017-11-14T13:27:13Z"),
                        "lastHeartbeatRecv" : ISODate("2017-11-14T13:27:13Z"),
                        "pingMs" : 0,
                        "syncingTo" : "11.120.0.10:27017"
                },
                {
                        "_id" : 3,
                        "name" : "11.120.0.11:27017
                        "health" : 0,
                        "state" : 8,
                        "stateStr" : "(not reachable/healthy)",
                        "uptime" : 0,
                        "optime" : Timestamp(1510610580, 1),
                        "optimeDate" : ISODate("2017-11-13T22:03:00Z"),
                        "lastHeartbeat" : ISODate("2017-11-14T13:27:10Z"),
                        "lastHeartbeatRecv" : ISODate("2017-11-13T22:03:01Z"),
                        "pingMs" : 0,
                        "syncingTo" : "11.120.0.10:27017"
                }
        ],
        "ok" : 1
}
```

步驟9.使用rs.remove指令，從主機刪除失敗的節點。運行此命令時出現一些錯誤，但再次檢查狀態以發現節點已刪除：

```
[root@pod1-controller-0 ~]$ mongo --host 11.120.0.10
<snip>
tripleo:PRIMARY> rs.remove('11.120.0.12:27017')
2017-11-16T18:41:04.999+0000 DBClientCursor::init call() failed
2017-11-16T18:41:05.000+0000 Error: error doing query: failed at src/mongo/shell/query.js:81
2017-11-16T18:41:05.001+0000 trying reconnect to 11.120.0.10:27017 (11.120.0.10) failed
2017-11-16T18:41:05.003+0000 reconnect 11.120.0.10:27017 (11.120.0.10) ok

tripleo:PRIMARY> rs.status()
{
        "set" : "tripleo",
        "date" : ISODate("2017-11-16T18:44:11Z"),
        "myState" : 1,
        "members" : [
                {
                        "_id" : 3,
                        "name" : "11.120.0.11:27017",
                        "health" : 1,
                        "state" : 2,
                        "stateStr" : "SECONDARY",
                        "uptime" : 187,
                        "optime" : Timestamp(1510857848, 3),
                        "optimeDate" : ISODate("2017-11-16T18:44:08Z"),
                        "lastHeartbeat" : ISODate("2017-11-16T18:44:11Z"),
                        "lastHeartbeatRecv" : ISODate("2017-11-16T18:44:09Z"),
                        "pingMs" : 0,
                        "syncingTo" : "11.120.0.10:27017"
                },
                {
                        "_id" : 4,
                        "name" : "11.120.0.10:27017",
```

```
                        "health" : 1,
                        "state" : 1,
                        "stateStr" : "PRIMARY",
                        "uptime" : 89820,
                        "optime" : Timestamp(1510857848, 3),
                        "optimeDate" : ISODate("2017-11-16T18:44:08Z"),
                        "electionTime" : Timestamp(1510811232, 1),
                        "electionDate" : ISODate("2017-11-16T05:47:12Z"),
                        "self" : true
                }
        ],
        "ok" : 1
}
tripleo:PRIMARY> exit
bye
```

步驟10.運行此命令可更新活動控制器節點清單。在此清單中包括新控制器節點。

```
[root@pod1-controller-0 ~]# sudo pcs resource update galera wsrep_cluster_address=gcomm://pod1-controller-0,pod1-controller-1,pod1-controller-2
```

步驟11.將這些檔案從已存在之控制器複製到新控制器：

/etc/sysconfig/clustercheck

/root/.my.cnf

```
On existing controller:

[root@pod1-controller-0 ~]# scp /etc/sysconfig/clustercheck stack@192.200.0.1:/tmp/.
[root@pod1-controller-0 ~]# scp /root/.my.cnf stack@192.200.0.1:/tmp/my.cnf

On new controller:

[root@pod1-controller-3 ~]# cd /etc/sysconfig

[root@pod1-controller-3 sysconfig]# scp stack@192.200.0.1:/tmp/clustercheck .

[root@pod1-controller-3 sysconfig]# cd /root

[root@pod1-controller-3 ~]# scp stack@192.200.0.1:/tmp/my.cnf .my.cnf
```

步驟12.從已存在的控制器之一運行cluster node add命令。

```
[root@pod1-controller-1 ~]# sudo pcs cluster node add pod1-controller-3

Disabling SBD service...
pod1-controller-3: sbd disabled
pod1-controller-0: Corosync updated
pod1-controller-1: Corosync updated

Setting up corosync...
pod1-controller-3: Succeeded
Synchronizing pcsd certificates on nodes pod1-controller-3...
pod1-controller-3: Success

Restarting pcsd on the nodes in order to reload the certificates...
pod1-controller-3: Success
```

步驟13.登入每個控制器並檢視/etc/corosync/corosync.conf檔案。確保列出新控制器，且未指派給該控制器的節點是序列中先前未使用的下一個編號。確保在所有3個控制器上完成此更改。

```
[root@pod1-controller-1 ~]# cat /etc/corosync/corosync.conf
totem {
   version: 2
   secauth: off
   cluster_name: tripleo_cluster
   transport: udpu
   token: 10000
}
nodelist {
   node {
       ring0_addr: pod1-controller-0
       nodeid: 5
   }
   node {
       ring0_addr: pod1-controller-1
       nodeid: 7
   }
   node {
       ring0_addr: pod1-controller-3
        nodeid: 6
   }
}
quorum {
   provider: corosync_votequorum
}
logging {
   to_logfile: yes
   logfile: /var/log/cluster/corosync.log
   to_syslog: yes
}
```

例如，修改後/etc/corosync/corosync.conf:

```
totem {
version: 2
secauth: off
cluster_name: tripleo_cluster
transport: udpu
token: 10000
}
nodelist {
   node {
       ring0_addr: pod1-controller-0
       nodeid: 5
   }
   node {
       ring0_addr: pod1-controller-1
       nodeid: 7
   }
   node {
       ring0_addr: pod1-controller-3
       nodeid: 9
   }
}
quorum {
   provider: corosync_votequorum
}
logging {
   to_logfile: yes
   logfile: /var/log/cluster/corosync.log
   to_syslog: yes
```

```
}
```

步驟14.在作用中控制器上重新啟動corosync。不要在新控制器上啟動corosync。

```
[root@pod1-controller-0 ~]# sudo pcs cluster reload corosync
[root@pod1-controller-1 ~]# sudo pcs cluster reload corosync
```

步驟15.從其中一個作用控制器啟動新控制器節點。

```
[root@pod1-controller-1 ~]# sudo pcs cluster start pod1-controller-3
```

步驟16.從其中一個作用控制器重新啟動Galera。

```
[root@pod1-controller-1 ~]#  sudo pcs cluster start pod1-controller-3

pod1-controller-0: Starting Cluster...

[root@pod1-controller-1 ~]# sudo pcs resource cleanup galera
Cleaning up galera:0 on pod1-controller-0, removing fail-count-galera
Cleaning up galera:0 on pod1-controller-1, removing fail-count-galera
Cleaning up galera:0 on pod1-controller-3, removing fail-count-galera
 * The configuration prevents the cluster from stopping or starting 'galera-master' (unmanaged)

Waiting for 3 replies from the CRMd... OK

[root@pod1-controller-1 ~]#
[root@pod1-controller-1 ~]# sudo pcs resource manage galera
```

步驟17.群集處於維護模式。禁用維護模式以使服務啟動。

```
[root@pod1-controller-2 ~]# sudo pcs property set maintenance-mode=false --wait
```

步驟18.檢查Galera的PC狀態，直到所有3個控制器都列為Galera的主控制器。

附註：對於大型設定，同步資料庫可能需要一些時間。

```
[root@pod1-controller-1 ~]# sudo pcs status | grep galera -A1

Master/Slave Set: galera-master [galera]
     Masters: [ pod1-controller-0 pod1-controller-1 pod1-controller-3 ]
```

步驟19.將群集切換到維護模式。

```
[root@pod1-controller-1~]# sudo pcs property set maintenance-mode=true --wait

[root@pod1-controller-1 ~]# pcs cluster status
Cluster Status:
 Stack: corosync
 Current DC: pod1-controller-0 (version 1.1.15-11.el7_3.4-e174ec8) - partition with quorum
 Last updated: Thu Nov 16 19:17:01 2017                  Last change: Thu Nov 16 19:16:48 2017
by root via cibadmin on pod1-controller-1
              *** Resource management is DISABLED ***
  The cluster will not attempt to start, stop or recover services

PCSD Status:
 pod1-controller-3: Online
 pod1-controller-0: Online
 pod1-controller-1: Online
```

步驟20.重新運行之前運行的部署指令碼。這一次應該會成功。

```
[stack@director ~]$ ./deploy-addController.sh
START with options: [u'overcloud', u'deploy', u'--templates', u'-r', u'/home/stack/custom-
templates/custom-roles.yaml', u'-e', u'/usr/share/openstack-tripleo-heat-
templates/environments/puppet-pacemaker.yaml', u'-e', u'/usr/share/openstack-tripleo-heat-
templates/environments/network-isolation.yaml', u'-e', u'/usr/share/openstack-tripleo-heat-
templates/environments/storage-environment.yaml', u'-e', u'/usr/share/openstack-tripleo-heat-
templates/environments/neutron-sriov.yaml', u'-e', u'/home/stack/custom-templates/network.yaml',
u'-e', u'/home/stack/custom-templates/ceph.yaml', u'-e', u'/home/stack/custom-
templates/compute.yaml', u'-e', u'/home/stack/custom-templates/layout-removeController.yaml',
u'--stack', u'newtonoc', u'--debug', u'--log-file', u'overcloudDeploy_11_14_17__13_53_12.log',
u'--neutron-flat-networks', u'phys_pcie1_0,phys_pcie1_1,phys_pcie4_0,phys_pcie4_1', u'--neutron-
network-vlan-ranges', u'datacentre:101:200', u'--neutron-disable-tunneling', u'--verbose', u'--
timeout', u'180']
options: Namespace(access_key='', access_secret='***', access_token='***',
access_token_endpoint='', access_token_type='', aodh_endpoint='', auth_type='',
auth_url='https://192.200.0.2:13000/v2.0', authorization_code='', cacert=None, cert='',
client_id='', client_secret='***', cloud='', consumer_key='', consumer_secret='***', debug=True,
default_domain='default', default_domain_id='', default_domain_name='', deferred_help=False,
discovery_endpoint='', domain_id='', domain_name='', endpoint='', identity_provider='',
identity_provider_url='', insecure=None, inspector_api_version='1', inspector_url=None,
interface='', key='', log_file=u'overcloudDeploy_11_14_17__13_53_12.log', murano_url='',
old_profile=None, openid_scope='', os_alarming_api_version='2',
os_application_catalog_api_version='1', os_baremetal_api_version='1.15', os_beta_command=False,
os_compute_api_version='', os_container_infra_api_version='1',
os_data_processing_api_version='1.1', os_data_processing_url='', os_dns_api_version='2',
os_identity_api_version='', os_image_api_version='1', os_key_manager_api_version='1',
os_metrics_api_version='1', os_network_api_version='', os_object_api_version='',
os_orchestration_api_version='1', os_project_id=None, os_project_name=None,
os_queues_api_version='2', os_tripleoclient_api_version='1', os_volume_api_version='',
os_workflow_api_version='2', passcode='', password='***', profile=None, project_domain_id='',
project_domain_name='', project_id='', project_name='admin', protocol='', redirect_uri='',
region_name='', roles='', timing=False, token='***', trust_id='', url='', user='',
user_domain_id='', user_domain_name='', user_id='', username='admin', verbose_level=3,
verify=None)
Auth plugin password selected

Starting new HTTPS connection (1): 192.200.0.2
"POST /v2/action_executions HTTP/1.1" 201 1696
HTTP POST https://192.200.0.2:13989/v2/action_executions 201
Overcloud Endpoint: http://172.25.22.109:5000/v2.0
Overcloud Deployed
clean_up DeployOvercloud:
END return value: 0

real     54m17.197s
user     0m3.421s
sys       0m0.670s
```

驗證控制器中的超雲服務

確保所有託管服務在控制器節點上正常運行。

```
[heat-admin@pod1-controller-2 ~]$ sudo pcs status
```

完成L3代理路由器

檢查路由器以確保第3層代理正確託管。執行此檢查時，請確保源位置為overcloudc檔案。

步驟1.查詢路由器名稱。

```
[stack@director~]$ source corerc
[stack@director ~]$ neutron router-list

+--------------------------------------+------+-----------------------------------------------
-----------------+-------------+------+

| id                                   | name |
external_gateway_info                                          | distributed | ha   |

+--------------------------------------+------+-----------------------------------------------
-----------------+-------------+------+

| d814dc9d-2b2f-496f-8c25-24911e464d02 | main | {"network_id": "18c4250c-e402-428c-87d6-
a955157d50b5",            | False       | True |
```

在本範例中，路由器的名稱是main。

步驟2.列出所有L3代理，以便找到故障節點和新節點的UUID。

```
[stack@director ~]$ neutron agent-list | grep "neutron-l3-agent"

| 70242f5c-43ab-4355-abd6-9277f92e4ce6 | L3 agent           | pod1-controller-0.localdomain  |
nova              | :-)   | True           | neutron-l3-agent          |
| 8d2ffbcb-b6ff-42cd-b5b8-da31d8da8a40 | L3 agent           | pod1-controller-2.localdomain  |
nova              | xxx   | True           | neutron-l3-agent          |
| a410a491-e271-4938-8a43-458084ffe15d | L3 agent           | pod1-controller-3.localdomain  |
nova              | :-)   | True           | neutron-l3-agent          |
| cb4bc1ad-ac50-42e9-ae69-8a256d375136 | L3 agent           | pod1-controller-1.localdomain  |
nova              | :-)   | True           | neutron-l3-agent          |
```

步驟3.在本例中，應從路由器中移除對應於pod1-controller-2.localdomain的第3層代理，並將對應於pod1-controller-3.localdomain的L3代理新增到路由器中。

```
[stack@director ~]$ neutron l3-agent-router-remove 8d2ffbcb-b6ff-42cd-b5b8-da31d8da8a40  main

Removed router main from L3 agent

[stack@director ~]$ neutron l3-agent-router-add a410a491-e271-4938-8a43-458084ffe15d main

Added router main to L3 agent
```

步驟4.檢查L3-Agent的更新清單。

```
[stack@director ~]$ neutron l3-agent-list-hosting-router main

+--------------------------------------+-----------------------------------+----------------+---
----+----------+
| id                                   | host                              | admin_state_up |
alive | ha_state |
+--------------------------------------+-----------------------------------+----------------+---
----+----------+

| 70242f5c-43ab-4355-abd6-9277f92e4ce6 | pod1-controller-0.localdomain | True           | :-)
| standby  |
| a410a491-e271-4938-8a43-458084ffe15d | pod1-controller-3.localdomain | True           | :-)
| standby  |
| cb4bc1ad-ac50-42e9-ae69-8a256d375136 | pod1-controller-1.localdomain | True           | :-)
```

```
| active   |
+--------------------------------------+-----------------------------------+---------------+---
----+----------+
```

步驟5.列出從已移除的控制器節點運行的所有服務，並將其移除。

```
[stack@director ~]$ neutron  agent-list | grep controller-2

| 877314c2-3c8d-4666-a6ec-69513e83042d | Metadata agent     | pod1-controller-2.localdomain
|                   | xxx   | True           | neutron-metadata-agent    |
| 8d2ffbcb-b6ff-42cd-b5b8-da31d8da8a40 | L3 agent           | pod1-controller-2.localdomain  |
nova              | xxx   | True           | neutron-l3-agent          |
| 911c43a5-df3a-49ec-99ed-1d722821ec20 | DHCP agent         | pod1-controller-2.localdomain  |
nova              | xxx   | True           | neutron-dhcp-agent        |
| a58a3dd3-4cdc-48d4-ab34-612a6cd72768 | Open vSwitch agent | pod1-controller-2.localdomain
|                   | xxx   | True           | neutron-openvswitch-agent |

[stack@director ~]$ neutron agent-delete 877314c2-3c8d-4666-a6ec-69513e83042d
Deleted agent(s): 877314c2-3c8d-4666-a6ec-69513e83042d
[stack@director ~]$ neutron agent-delete 8d2ffbcb-b6ff-42cd-b5b8-da31d8da8a40
Deleted agent(s): 8d2ffbcb-b6ff-42cd-b5b8-da31d8da8a40
[stack@director ~]$ neutron agent-delete 911c43a5-df3a-49ec-99ed-1d722821ec20
Deleted agent(s): 911c43a5-df3a-49ec-99ed-1d722821ec20
[stack@director ~]$ neutron agent-delete a58a3dd3-4cdc-48d4-ab34-612a6cd72768
Deleted agent(s): a58a3dd3-4cdc-48d4-ab34-612a6cd72768

[stack@director ~]$ neutron agent-list | grep controller-2
[stack@director ~]$
```

最終確定計算服務

步驟1。檢查從移除的節點中留下的新星服務清單項並將其刪除。

```
[stack@director ~]$ nova service-list | grep controller-2

| 615 | nova-consoleauth | pod1-controller-2.localdomain  | internal          | enabled | down
| 2017-11-16T16:08:14.000000 | -               |
| 618 | nova-scheduler   | pod1-controller-2.localdomain  | internal          | enabled | down
| 2017-11-16T16:08:13.000000 | -               |
| 621 | nova-conductor   | pod1-controller-2.localdomain  | internal          | enabled | down
| 2017-11-16T16:08:14.000000 | -

[stack@director ~]$ nova service-delete 615
[stack@director ~]$ nova service-delete 618
[stack@director ~]$ nova service-delete 621

stack@director ~]$ nova service-list | grep controller-2
```

步驟2. 確保consoleauth程式在所有控制器上執行，或使用以下命令重新啟動該程式：pc資源重新啟動openstack-nova-consoleauth:

```
[stack@director ~]$ nova service-list | grep consoleauth

| 601 | nova-consoleauth | pod1-controller-0.localdomain  | internal          | enabled | up
| 2017-11-16T21:00:10.000000 | -               |
| 608 | nova-consoleauth | pod1-controller-1.localdomain  | internal          | enabled | up
| 2017-11-16T21:00:13.000000 | -               |
| 622 | nova-consoleauth | pod1-controller-3.localdomain  | internal          | enabled | up
| 2017-11-16T21:00:13.000000 | -
```

在控制器節點上重新啟動圍欄

步驟1. 檢查所有控制器中是否有通往底層雲192.0.0.0/8的IP路由

```
[root@pod1-controller-3 ~]# ip route
default via 172.25.22.1 dev vlan101
11.117.0.0/24 dev vlan17  proto kernel  scope link  src 11.117.0.12
11.118.0.0/24 dev vlan18  proto kernel  scope link  src 11.118.0.12
11.119.0.0/24 dev vlan19  proto kernel  scope link  src 11.119.0.12
11.120.0.0/24 dev vlan20  proto kernel  scope link  src 11.120.0.12
169.254.169.254 via 192.200.0.1 dev eno1
172.25.22.0/24 dev vlan101  proto kernel  scope link  src 172.25.22.102
192.0.0.0/8 dev eno1  proto kernel  scope link  src 192.200.0.103
```

步驟2.檢查當前的石塊配置。刪除對舊控制器節點的任何引用。

```
[root@pod1-controller-3 ~]# sudo pcs stonith show --full
 Resource: my-ipmilan-for-controller-6 (class=stonith type=fence_ipmilan)
 Attributes: pcmk_host_list=pod1-controller-1 ipaddr=192.100.0.1 login=admin
passwd=Csco@123Starent lanplus=1
 Operations: monitor interval=60s (my-ipmilan-for-controller-6-monitor-interval-60s)
 Resource: my-ipmilan-for-controller-4 (class=stonith type=fence_ipmilan)
 Attributes: pcmk_host_list=pod1-controller-0 ipaddr=192.100.0.14 login=admin
passwd=Csco@123Starent lanplus=1
 Operations: monitor interval=60s (my-ipmilan-for-controller-4-monitor-interval-60s)
Resource: my-ipmilan-for-controller-7 (class=stonith type=fence_ipmilan)
 Attributes: pcmk_host_list=pod1-controller-2 ipaddr=192.100.0.15 login=admin
passwd=Csco@123Starent lanplus=1
 Operations: monitor interval=60s (my-ipmilan-for-controller-7-monitor-interval-60s)

[root@pod1-controller-3 ~]# pcs stonith delete my-ipmilan-for-controller-7
Attempting to stop: my-ipmilan-for-controller-7...Stopped
```

步驟3. 新控制器新增斯通組態。

```
[root@pod1-controller-3 ~]sudo pcs stonith create my-ipmilan-for-controller-8 fence_ipmilan
pcmk_host_list=pod1-controller-3 ipaddr=<CIMC_IP> login=admin passwd=<PASSWORD> lanplus=1 op
monitor interval=60s
```

步驟4.從任何控制器重新啟動隔離並驗證狀態。

```
[root@pod1-controller-1 ~]# sudo pcs property set stonith-enabled=true
[root@pod1-controller-3 ~]# pcs status

<snip>
my-ipmilan-for-controller-1 (stonith:fence_ipmilan): Started pod1-controller-3
my-ipmilan-for-controller-0 (stonith:fence_ipmilan): Started pod1-controller-3
my-ipmilan-for-controller-3 (stonith:fence_ipmilan): Started pod1-controller-3
```