PCRF更换控制器服务器UCS C240 M4

目录

简介

本文档介绍在托管CPS虚拟网络功能(VNF)的Ultra-M设置中更换故障控制器服务器所需的步骤。

先决条件

备份

在恢复时，思科建议使用以下步骤备份OSPD数据库(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
```

初步状态检查

在您继续执行更换过程之前，必须检查OpenStack环境和服务的当前状态并确保其正常运行。它有助于避免控制器更换过程中出现问题。

步骤1.检查OpenStack的状态和节点列表：

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

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

步骤2.检查控制器上的Pace maker状态。

登录其中一个活动控制器并检查起搏器状态。所有服务应在可用控制器上运行，并在故障控制器上停止。

```
[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
```

在本例中，控制器2处于脱机状态。因此，它将被更换。控制器0和控制器1运行正常，并运行集群服务。

步骤3.检查活动控制器中的MariaDB状态。

```
[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
```

验证每个活动控制器都存在以下线路：

wsrep_local_state_comment:已同步

wsrep_cluster_size:2

步骤4.检查活动控制器中的Rabbitmq状态。故障控制器不应出现在运行的节点列表中。

```
[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',[]}]}]
```

步骤5.检查OSP-D节点中是否所有下云服务都处于已加载、活动和运行状态。

```
[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'.
```

在控制器集群中禁用隔离

```
[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
```

安装新控制器节点

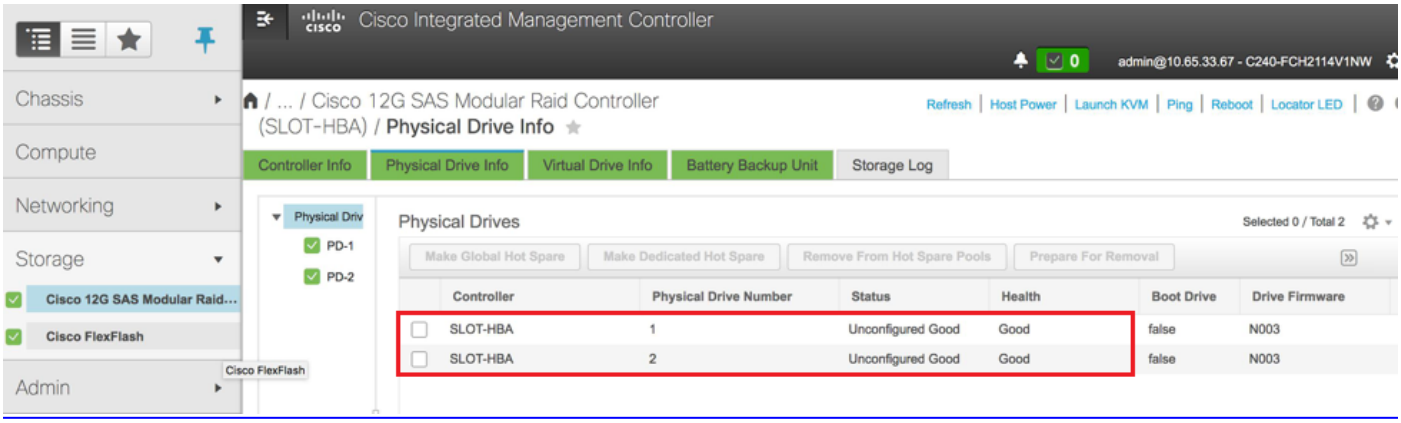
步骤1.安装新UCS C240 M4服务器的步骤和初始设置步骤可参阅《Cisco UCS C240 M4服务器安装和服务指南》

步骤2.使用CIMC IP登录到服务器。

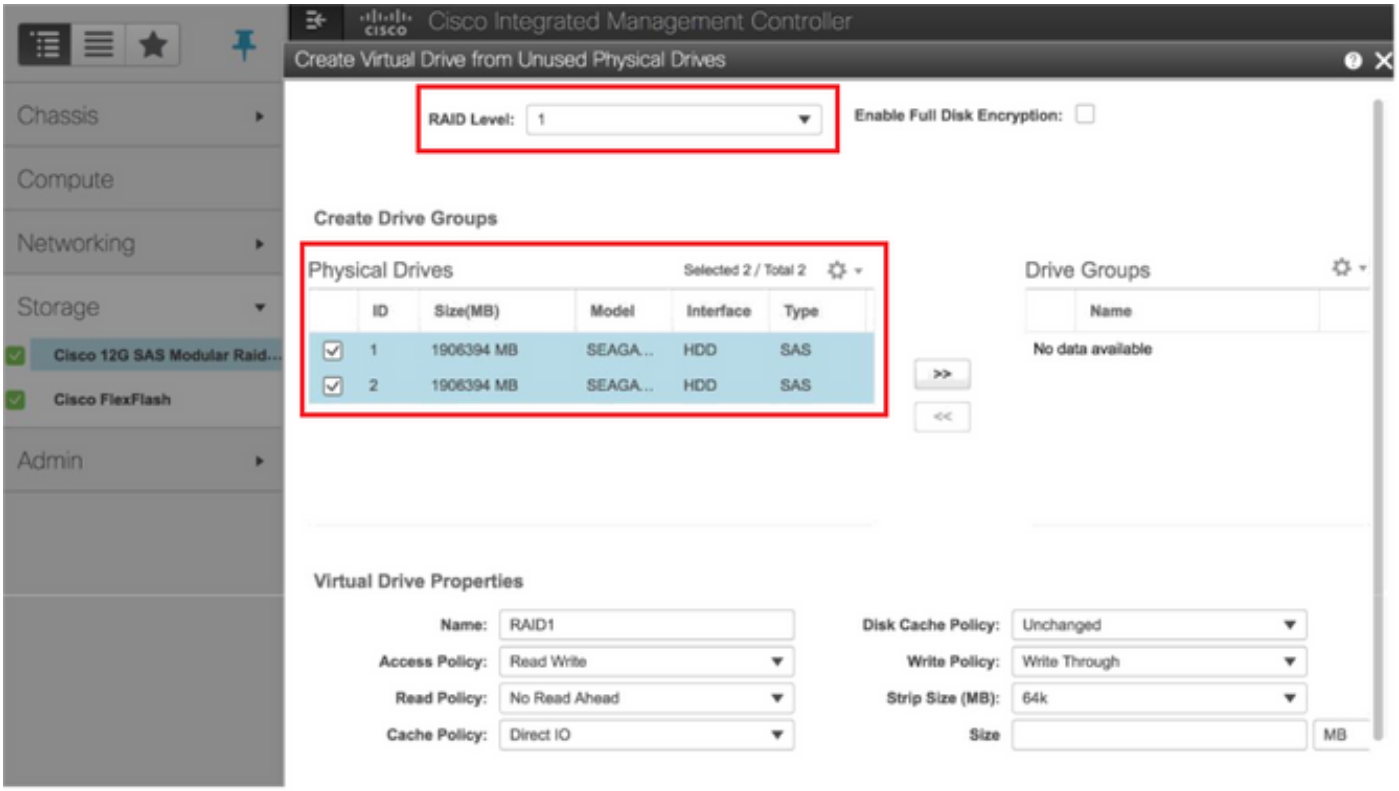
步骤3.如果固件与之前使用的推荐版本不同，则执行BIOS升级。BIOS升级步骤如下：

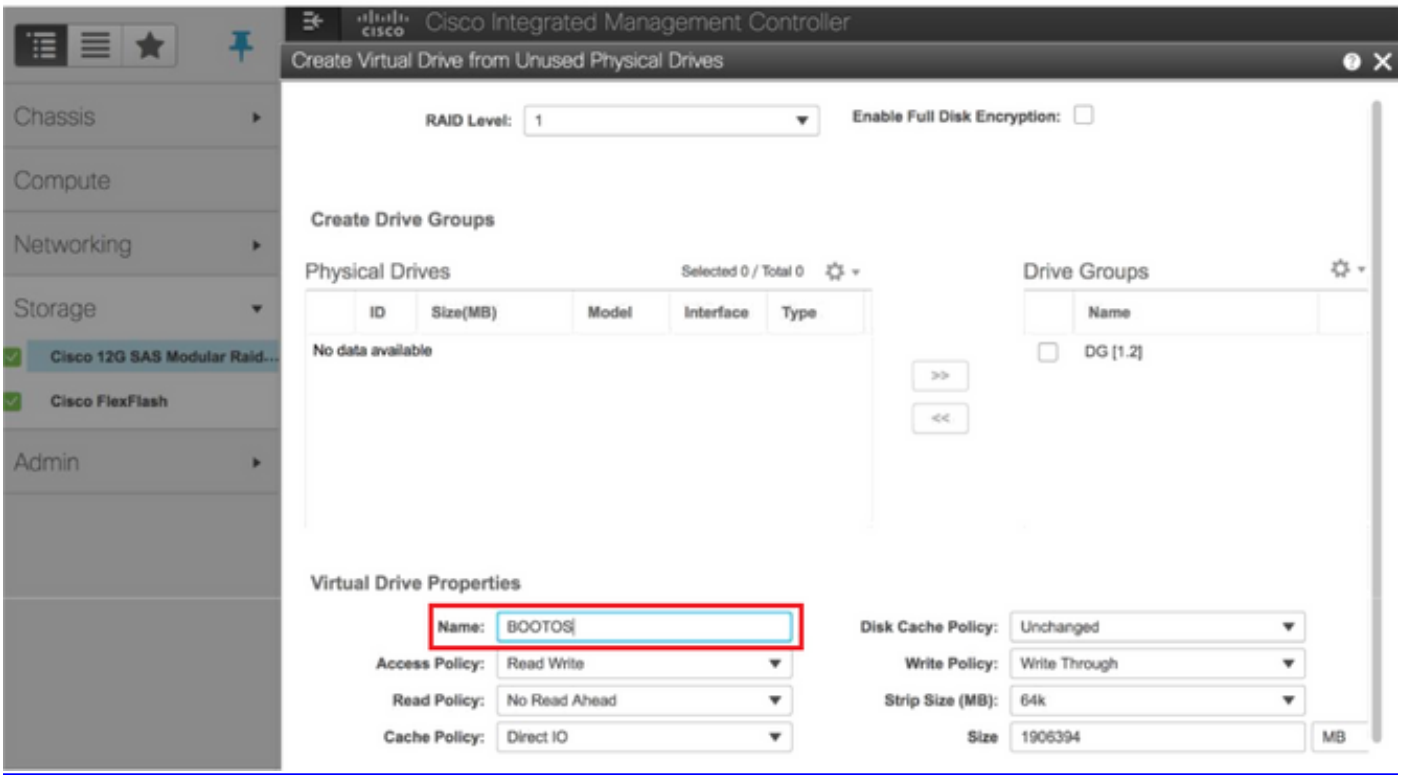
Cisco UCS C系列机架式服务器BIOS升级指南

步骤4.检验物理驱动器的状态。它必须是未配**置的正**常。导航至**存储> Cisco 12G SAS模块化RAID控制器(SLOT-HBA)>物理驱动器信息。**

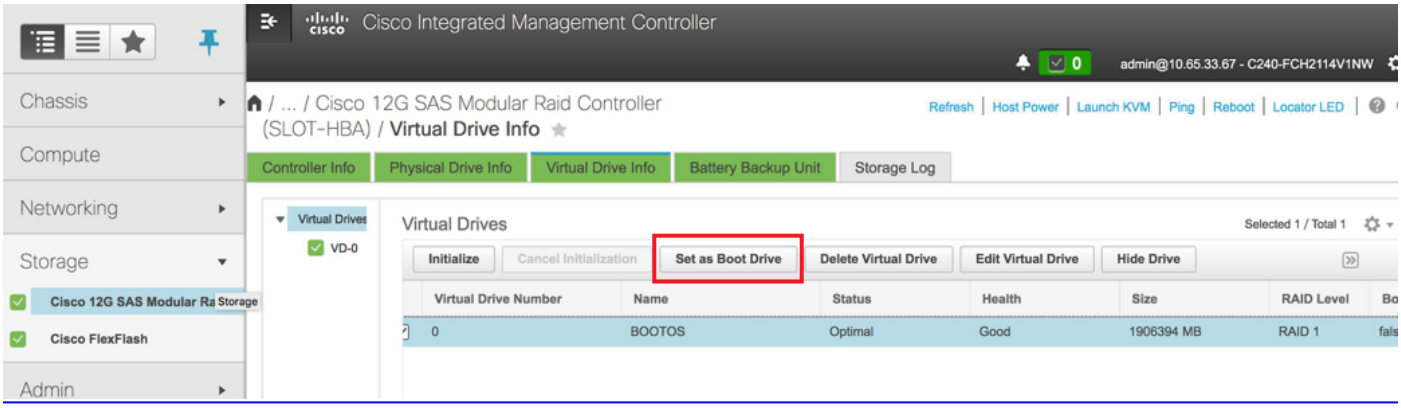


步骤5.要从RAID级别为1的物理驱动器创建虚拟驱动器：导航至**存储> Cisco 12G SAS模块化RAID控制器(SLOT-HBA)>控制器信息>从未使用的物理驱动器创建虚拟驱动器**，如图所示。

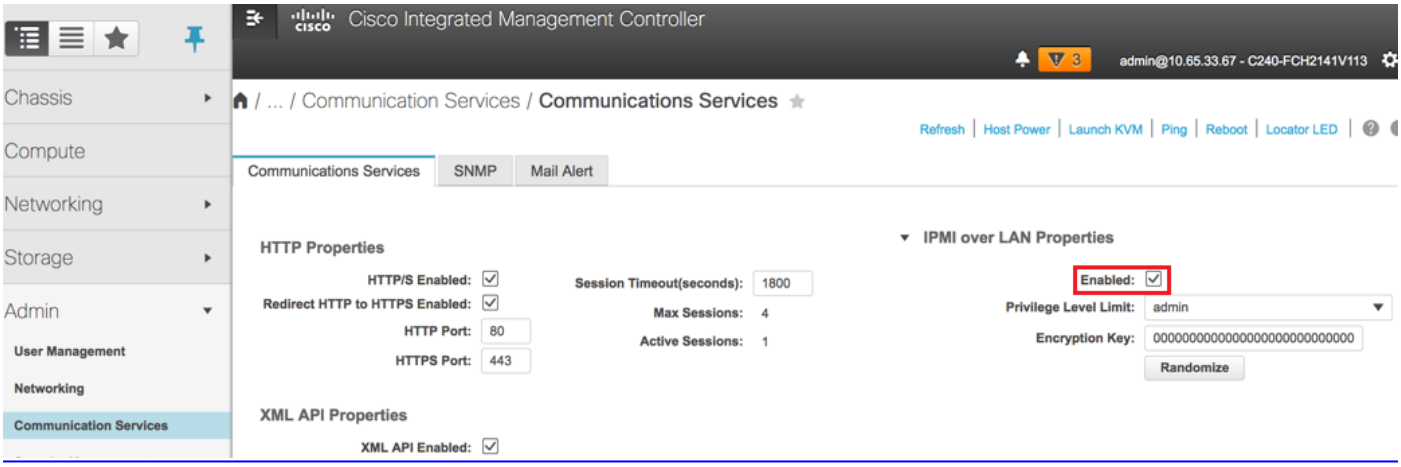




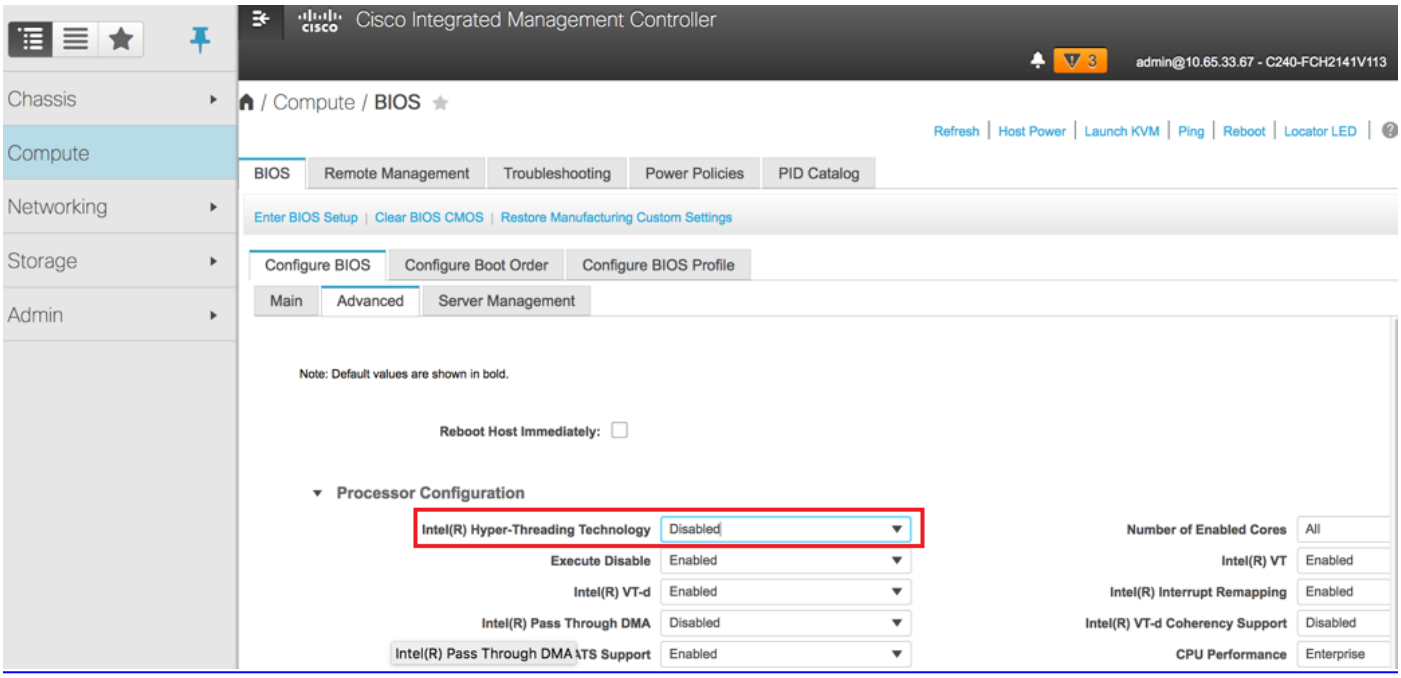
- 选择VD并配置“Set as Boot Drive(设**置为引导驱动器)**”：



步骤6.要启用LAN上的IPMI，请导航到**Admin > Communication Services > Communication Services**。



步骤7.要禁用超线程，请导航至“计算”>“BIOS”>“配置BIOS”>“高级”>“处理器配置”，如图所示。



注意：此处显示了该映像，本节中提及的配置步骤参考固件版本3.0(3e)，如果您使用其他版本，可能会略有变化。

在Overcloud中更换控制器节点

本节介绍将故障控制器替换为超云中的新控制器所需的步骤。为此，将重新使用用于启动堆栈的deploy.sh脚本。在部署时，在ControllerNodesPostDeployment阶段，更新将因Puppet模块中的某些限制而失败。重新启动部署脚本之前需要手动干预。

准备删除故障控制器节点

步骤1.确定故障控制器的索引。索引是OpenStack服务器列表输出中控制器名称上的数字后缀。在本例中，索引为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 |
```

步骤2.创建Yaml文件~templates/remove-controller.yaml，以定义要删除的节点。使用上一步中找到的索引查找资源列表中的条目。

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

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

parameter_defaults:
 CorosyncSettleTries: 5
```

步骤3.制作部署脚本的副本，以安装过云并插入一行，以包括以前创建的remove-controller.yaml文件。

```
[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
```

步骤4.使用此处提到的命令确定要更换的控制器的ID，并将其移至维护模式。

```
[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        |
```

步骤5.为了确保数据库在更换过程时运行，请从起搏器控制中删除Galera并在其中一个主用控制器上运行此命令。

```
[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>
```

准备添加新控制器节点

步骤1.仅使用**新控制器详**细信息创建controllerRMA.json文件。确保新控制器上的索引号以前未使用过。通常，递增到下一个最高控制器编号。

示例：最早的是控制器2，因此创建控制器3。

注意：注意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>"
       }
   ]
}
```

步骤2.使用上一步骤中创建的json文件导入新节点。

```
[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
```

步骤3.将节点设置为管理状态。

```
[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 |
```

步骤4.运行内省。

```
[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       |
```

步骤5.使用新控制器属性标记可用节点。确保使用为新控制器指定的控制器ID，如controllerRMA.json文件中所用。

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

步骤6.在部署脚本中，有一个名为layout.yaml的自定义模板，该模板除其他外，还指定为各种接口的控制器分配了哪些IP地址。在新堆栈中，为控制器0、控制器1和控制器2定义了3个地址。添加新控制器时，请确保按顺序为每个子网添加下一个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
```

步骤7.现在运行以前创建的deploy-removecontroller.sh，以删除旧节点并添加新节点。

注意：此步骤在ControllerNodesDeployment_Step1中预期会失败。此时需要手动干预。

```
[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
```

可以使用以下命令监控部署的进度/状态：

```
[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                                    |

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

手动干预

步骤1.在OSP-D服务器上，运行OpenStack server list命令以列出可用的控制器。新添加的控制器应显示在列表中。

```
[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 |
```

步骤2.连接到其中一个活动控制器（而不是新添加的控制器）并查看文件/etc/corosync/corosycn.conf。查找向每个控制器分配节点的节点列表。查找失败节点的条目并记录其节点ID:

```
[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
   }
}
```

步骤3.登录到每个活动控制器。删除故障节点并重新启动服务。在这种情况下，请删除Pod1-controller-2。请勿在新添加的控制器上执行此操作。

```
[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
```

步骤4.从其中一个活动控制器运行此命令以从群集中删除故障节点。

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

步骤5.从其中一个活动控制器运行此命令，以便从rabbitmq群集中删除故障节点。

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

步骤6.从MongoDB中删除失败节点。为此，您需要找到活动的Mongo节点。使用netstat查找主机的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
```

步骤7.登录到节点并检查是否是使用上一命令中的IP地址和端口号的主节点。

```
[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
}
```

如果节点不是主节点，请登录到其他活动控制器并执行相同步骤。

步骤8.从主设备中，使用rs.status()命令列**出可用节**点。查找旧节点/无响应节点并标识mongo节点名称。

```
[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
}
```

步骤9.从主设备中，使用rs.remove命令删**除故障节**点。运行此命令时会看到一些错误，但再次检查状态，以查找节点已被删除：

```
[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
```

步骤10.运行此命令以更新活动控制器节点的列表。在此列表中包含新控制器节点。

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

步骤11.将这些文件从已存在的控制器复制到新控制器：

/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
```

步骤12.从已存在的一个控制器中运行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
```

步骤13.登录到每个控制器并查看文件/etc/corosync/corosync.conf。确保列出新控制器，并且分配给该控制器的节点是序列中下一个先前未使用的编号。确保在所有3个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，修改后/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
}
```

步骤14.在活动控制器上重新启动corosync。请勿在新控制器上启动corosync。

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

步骤15.从一个作用控制器启动新控制器节点。

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

步骤16.从一个控制器重新启动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
```

步骤17.群集处于维护模式。禁用维护模式以启动服务。

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

步骤18.检查PC的Galera状态，直到3个控制器全部列为Galera的主控制器。

注意：对于大型设置，同步数据库可能需要一些时间。

```
[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 ]
```

步骤19.将集群切换到维护模式。

```
[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
```

步骤20.重新运行之前运行的部署脚本。这次应该会成功。

```
[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
```

验证控制器中的超云服务

确保所有托管服务在控制器节点上正常运行。

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

最终确定L3代理路由器

检查路由器以确保L3代理正确托管。执行此检查时，请确保源到overcloudrc文件。

步骤1.查找路由器名称。

```
[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 |
```

在本例中，路由器的名称是main。

步骤2.列出所有L3代理，以查找故障节点和新节点的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          |
```

步骤3.在本例中，应从路由器中删除与pod1-controller-2.localdomain对应的L3代理，并将与pod1-controller-3.localdomain对应的L3代理添加到路由器。

```
[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
```

步骤4.检查更新的L3代理列表。

```
[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   |
+--------------------------------------+-----------------------------------+----------------+---
----+----------+
```

步骤5.列出从已删除的控制器节点运行的所有服务并删除它们。

```
[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 ~]$
```

最终确定计算服务

步骤1.检查从**删除的节**点中保留的nova service-list项并将其删除。

```
[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
```

步骤2.确保控制台**身份**验证进程在所有控制器上运行，或使用以下命令重新启动该进程：**pcs resource restart 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 | -
```

在控制器节点上重新启动隔离

步骤1.检查所有控制器以查找通向下云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
```

步骤2.检查当前的Stonith配置。删除对旧控制器节点的任何引用。

```
[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
```

步骤3.为新控制器添加Stonith配置。

```
[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
```

步骤4.重新从任何控制器启动隔离并验证状态。

```
[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
```