PCRF更换OSD-Compute UCS 240M4

目录

简介

本文档介绍在托管思科策略套件(CPS)虚拟网络功能(VNF)的Ultra-M设置中更换故障osd-compute服务器所需的步骤。

背景信息

本文档面向熟悉Cisco Ultra-M平台的思科人员，详细介绍在OSD-Compute服务器更换时在OpenStack和CPS VNF级别执行所需的步骤。

> **注意**：为了定义本文档中的步骤，我们考虑了Ultra M 5.1.x版本。

运行状况检查

在更换Osd-Compute节点之前，必须检查Red Hat OpenStack平台环境的当前状态。建议您检查当前状态，以避免在计算更换流程开启时出现问题。

来自OSPD

```
[root@director ~]$ su - stack
[stack@director ~]$ cd ansible
[stack@director ansible]$ ansible-playbook -i inventory-new openstack_verify.yml  -e
```

```
platform=pcrf
```

步骤1.从每15分钟生成的ultram-health报告检验系统的运行状况。

```
[stack@director ~]# cd /var/log/cisco/ultram-health
```

检查文件ultram_health_os.report。

唯一的服务应显示为XXX状态为neutron-sriov-nic-agent.service。

步骤2.检查是否为所有控制器运行rabbitmq，这些控制器从OSPD运行。

```
[stack@director ~]# for i in $(nova list| grep controller | awk '{print $12}'| sed
's/ctlplane=//g') ; do (ssh -o StrictHostKeyChecking=no heat-admin@$i "hostname;sudo rabbitmqctl
eval 'rabbit_diagnostics:maybe_stuck().'" ) & done
```

步骤3.检验Stonith是否已启用。

```
[stack@director ~]# sudo pcs property show stonith-enabled
```

对于所有控制器，验证PCS状态

- 所有控制器节点在haproxy-clone下启动
- 所有控制器节点都是Master在Galera下
- 在Rabbitmq下启动所有控制器节点
- 1个控制器节点为主节点，2个从节点

来自OSPD

```
[stack@director ~]$ for i in $(nova list| grep controller | awk '{print $12}'| sed
's/ctlplane=//g') ; do (ssh -o StrictHostKeyChecking=no heat-admin@$i "hostname;sudo pcs status"
) ;done
```

步骤4.从OSPD运行以下命令，验证所有openstack服务是否都处于活动状态：

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

步骤5.验证控制器的CEPH状态为HEALTH_OK。

```
[stack@director ~]# for i in $(nova list| grep controller | awk '{print $12}'| sed
's/ctlplane=//g') ; do (ssh -o StrictHostKeyChecking=no heat-admin@$i "hostname;sudo ceph -s" )
;done
```

步骤6.检验OpenStack组件日志。查找任何错误：

```
Neutron:
[stack@director ~]# sudo tail -n 20 /var/log/neutron/{dhcp-agent,l3-agent,metadata-
agent,openvswitch-agent,server}.log

Cinder:
[stack@director ~]# sudo tail -n 20 /var/log/cinder/{api,scheduler,volume}.log

Glance:
[stack@director ~]# sudo tail -n 20 /var/log/glance/{api,registry}.log
```

步骤7.从OSPD对API执行这些验证。

```
[stack@director ~]$ source
```

```
[stack@director ~]$ nova list

[stack@director ~]$ glance image-list

[stack@director ~]$ cinder list

[stack@director ~]$ neutron net-list
```

步骤8.检验服务的运行状况。

```
Every service status should be “up”:
[stack@director ~]$ nova service-list

Every service status should be “ :-)”:
[stack@director ~]$ neutron agent-list

Every service status should be “up”:
[stack@director ~]$ cinder service-list
```

备份

在恢复时，思科建议使用以下步骤备份OSPD数据库。

步骤1.执行Mysql转储。

```
[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
```

此过程可确保在不影响任何实例可用性的情况下更换节点。

步骤2.从Cluster Manager VM备份CPS VM:

```
[root@CM ~]# config_br.py -a export --all /mnt/backup/CPS_backup_$(date +\%Y-\%m-\%d).tar.gz

or

[root@CM ~]# config_br.py -a export --mongo-all --svn --etc --grafanadb --auth-htpasswd --
haproxy /mnt/backup/$(hostname)_backup_all_$(date +\%Y-\%m-\%d).tar.gz
```

识别OSD计算节点中托管的虚拟机

确定托管在计算服务器上的虚拟机：

步骤1.计算服务器包含弹性服务控制器(ESC)。

```
[stack@director ~]$ nova list --field name,host,networks | grep osd-compute-1
| 50fd1094-9c0a-4269-b27b-cab74708e40c | esc | pod1-osd-compute-0.localdomain
| tb1-orch=172.16.180.6; tb1-mgmt=172.16.181.3
```

注意：在此处显示的输出中，第一列对应于通用唯一标识符(UUID)，第二列是VM名称，第三列是VM所在的主机名。此输出的参数将用于后续部分。

注意：如果要更换的OSD计算节点完全关闭且无法访问，请继续参阅标题为"从新聚合列表中删除OSD计算节点"的部分。否则，请从下一节继续。

步骤2.检验CEPH是否具有可用容量，以便删除单个OSD服务器。

```
[root@pod1-osd-compute-0 ~]# sudo ceph df
GLOBAL:
   SIZE       AVAIL      RAW USED     %RAW USED
    13393G     11804G        1589G         11.87
POOLS:
   NAME        ID    USED      %USED     MAX AVAIL     OBJECTS
   rbd         0          0         0         3876G           0
   metrics     1     4157M      0.10         3876G      215385
   images      2     6731M      0.17         3876G         897
   backups     3         0         0         3876G           0
   volumes     4      399G      9.34         3876G      102373
   vms         5      122G      3.06         3876G       31863
```

步骤3.验证osd-compute服务器上的osd树状态是否为up状态。

```
[heat-admin@pod1-osd-compute-0 ~]$ sudo ceph osd tree
ID WEIGHT   TYPE NAME                          UP/DOWN REWEIGHT PRIMARY-AFFINITY
-1 13.07996 root default
-2  4.35999     host pod1-osd-compute-0
 0  1.09000         osd.0                           up  1.00000          1.00000
 3  1.09000         osd.3                           up 1.00000          1.00000
 6  1.09000         osd.6                           up  1.00000          1.00000
 9  1.09000         osd.9                           up  1.00000          1.00000
-3  4.35999     host pod1-osd-compute-2
```

```
 1  1.09000        osd.1                          up  1.00000          1.00000

 4  1.09000         osd.4                          up  1.00000          1.00000

 7  1.09000         osd.7                          up  1.00000          1.00000

10  1.09000         osd.10                         up  1.00000          1.00000

-4  4.35999     host pod1-osd-compute-1

 2  1.09000         osd.2                          up  1.00000          1.00000

 5  1.09000         osd.5                          up  1.00000          1.00000

 8  1.09000         osd.8                          up  1.00000          1.00000

11  1.09000         osd.11                         up  1.00000          1.00000
```

步骤4. CEPH进程在osd-compute服务器上处于活动状态。

```
[root@pod1-osd-compute-0 ~]# systemctl list-units *ceph*

UNIT                                LOAD   ACTIVE SUB     DESCRIPTION

var-lib-ceph-osd-ceph\x2d11.mount loaded active mounted /var/lib/ceph/osd/ceph-11

var-lib-ceph-osd-ceph\x2d2.mount  loaded active mounted /var/lib/ceph/osd/ceph-2

var-lib-ceph-osd-ceph\x2d5.mount  loaded active mounted /var/lib/ceph/osd/ceph-5

var-lib-ceph-osd-ceph\x2d8.mount  loaded active mounted /var/lib/ceph/osd/ceph-8

ceph-osd@11.service               loaded active running Ceph object storage daemon

ceph-osd@2.service                loaded active running Ceph object storage daemon

ceph-osd@5.service                loaded active running Ceph object storage daemon

ceph-osd@8.service                loaded active running Ceph object storage daemon

system-ceph\x2ddisk.slice         loaded active active  system-ceph\x2ddisk.slice

system-ceph\x2dosd.slice          loaded active active  system-ceph\x2dosd.slice

ceph-mon.target                   loaded active active  ceph target allowing to start/stop all
ceph-mon@.service instances at once

ceph-osd.target                   loaded active active  ceph target allowing to start/stop all
ceph-osd@.service instances at once

ceph-radosgw.target               loaded active active  ceph target allowing to start/stop all
ceph-radosgw@.service instances at once

ceph.target                       loaded active active  ceph target allowing to start/stop all
ceph*@.service instances at once
```

步骤5.禁用并停止每个ceph实例，并从osd中删除每个实例，然后卸载目录。对每个ceph实例重复此步骤。

```
[root@pod1-osd-compute-0 ~]# systemctl disable ceph-osd@11

[root@pod1-osd-compute-0 ~]# systemctl stop ceph-osd@11

[root@pod1-osd-compute-0 ~]# ceph osd out 11
marked out osd.11.

[root@pod1-osd-compute-0 ~]# ceph osd crush remove osd.11
removed item id 11 name 'osd.11' from crush map

[root@pod1-osd-compute-0 ~]# ceph auth del osd.11
updated

[root@pod1-osd-compute-0 ~]# ceph osd rm 11
removed osd.11

[root@pod1-osd-compute-0 ~]# umount /var/lib/ceph/osd/ceph-11

[root@pod1-osd-compute-0 ~]# rm -rf /var/lib/ceph/osd/ceph-11
```

（或）

步骤6. Clean.sh脚本可用于立即执行上述任务。

```
[heat-admin@pod1-osd-compute-0 ~]$ sudo ls /var/lib/ceph/osd
ceph-11 ceph-3 ceph-6 ceph-8
[heat-admin@pod1-osd-compute-0 ~]$ /bin/sh clean.sh

[heat-admin@pod1-osd-compute-0 ~]$ cat clean.sh
#!/bin/sh
set -x
CEPH=`sudo ls /var/lib/ceph/osd`
for c in $CEPH
do
  i=`echo $c |cut -d'-' -f2`
```

```
  sudo systemctl disable ceph-osd@$i || (echo "error rc:$?"; exit 1)
  sleep 2
  sudo systemctl stop ceph-osd@$i || (echo "error rc:$?"; exit 1)
  sleep 2
  sudo ceph osd out $i || (echo "error rc:$?"; exit 1)
  sleep 2
  sudo ceph osd crush remove osd.$i || (echo "error rc:$?"; exit 1)
  sleep 2
  sudo ceph auth del osd.$i || (echo "error rc:$?"; exit 1)
  sleep 2
  sudo ceph osd rm $i || (echo "error rc:$?"; exit 1)
  sleep 2
  sudo umount /var/lib/ceph/osd/$c || (echo "error rc:$?"; exit 1)
  sleep 2
  sudo rm -rf /var/lib/ceph/osd/$c || (echo "error rc:$?"; exit 1)
  sleep 2
done
sudo ceph osd tree
```

迁移/删除所有OSD进程后，可以从超云中删除节点。

注意：删除CEPH后，VNF HD RAID将进入“已降级”状态，但必须仍可访问硬盘。

平稳关闭电源

将ESC迁移到备用模式

步骤1.登录到计算节点中托管的ESC，并检查它是否处于主状态。如果是，请将ESC切换到备用模式。

```
[admin@esc esc-cli]$ escadm status
0 ESC status=0 ESC Master Healthy
```

```
[admin@esc ~]$ sudo service keepalived stop
Stopping keepalived:                                       [  OK  ]

[admin@esc ~]$ escadm status
1 ESC status=0 In SWITCHING_TO_STOP state. Please check status after a while.
```

```
[admin@esc ~]$ sudo reboot
Broadcast message from admin@vnf1-esc-esc-0.novalocal
      (/dev/pts/0) at 13:32 ...
The system is going down for reboot NOW!
```

步骤2.从Nova Aggregate List中删除OSD-Compute节点。

- 列出nova聚合并根据它托管的VNF确定与计算服务器对应的聚合。通常，格式为<VNFNAME>-EM-MGMT<X>和<VNFNAME>-CF-MGMT<X>

```
[stack@director ~]$ nova aggregate-list
+----+------+-------------------+
| Id | Name | Availability Zone |
+----+------+-------------------+
| 3 | esc1 | AZ-esc1 |
| 6 | esc2 | AZ-esc2 |
| 9 | aaa | AZ-aaa |
+----+------+-------------------+
```

在本例中，osd-compute服务器属于esc1。因此，对应的聚合是**esc1**

步骤3.从识别的聚合中删除osd-compute节点。

nova aggregate-remove-host

```
[stack@director ~]$ nova aggregate-remove-host esc1 pod1-osd-compute-0.localdomain
```

步骤4.检验osd-compute节点是否已从聚合中删除。现在，确保主机未列在聚合下。

nova aggregate-show

```
[stack@director ~]$ nova aggregate-show esc1
[stack@director ~]$
```

OSD计算节点删除

本节中提到的步骤是通用的，与计算节点中托管的虚拟机无关。

从Overcloud中删除

步骤1.创建名为delete_node.sh的脚本文件，其内容如图所示。请确保所提及的模板与用于堆栈部署的**deploy.sh脚**本中使用的模板相同。

delete_node.sh

```
openstack overcloud node delete --templates -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 /home/stack/custom-templates/network.yaml -e
/home/stack/custom-templates/ceph.yaml -e /home/stack/custom-templates/compute.yaml -e
/home/stack/custom-templates/layout.yaml -e /home/stack/custom-templates/layout.yaml --stack
```

```
[stack@director ~]$ source stackrc
[stack@director ~]$ /bin/sh delete_node.sh
+ openstack overcloud node delete --templates -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 /home/stack/custom-templates/network.yaml -e
/home/stack/custom-templates/ceph.yaml -e /home/stack/custom-templates/compute.yaml -e
/home/stack/custom-templates/layout.yaml -e /home/stack/custom-templates/layout.yaml --stack
pod1 49ac5f22-469e-4b84-badc-031083db0533
Deleting the following nodes from stack pod1:
- 49ac5f22-469e-4b84-badc-031083db0533
Started Mistral Workflow. Execution ID: 4ab4508a-c1d5-4e48-9b95-ad9a5baa20ae

real   0m52.078s
user   0m0.383s
sys    0m0.086s
```

步骤2.等待OpenStack堆栈操作移至“完成”状态。

```
[stack@director ~]$  openstack stack list
+--------------------------------------+------------+-----------------+----------------------+--
--------------------+
| ID                                   | Stack Name | Stack Status    | Creation Time        |
Updated Time         |
+--------------------------------------+------------+-----------------+----------------------+--
--------------------+
| 5df68458-095d-43bd-a8c4-033e68ba79a0 | pod1 | UPDATE_COMPLETE | 2018-05-08T21:30:06Z | 2018-
05-08T20:42:48Z |
+--------------------------------------+------------+-----------------+----------------------+--
--------------------
```

从服务列表中删除OSD计算节点

从服务列表中删除计算服务。

```
[stack@director ~]$ source corerc
[stack@director ~]$ openstack compute service list | grep osd-compute-0
| 404 | nova-compute     | pod1-osd-compute-0.localdomain     | nova     | enabled | up    |
2018-05-08T18:40:56.000000 |

openstack compute service delete
```

```
[stack@director ~]$ openstack compute service delete 404
```

删除中子代理

删除旧的关联中子代理并打开计算服务器的vswitch代理。

```
[stack@director ~]$ openstack network agent list | grep osd-compute-0
| c3ee92ba-aa23-480c-ac81-d3d8d01dcc03 | Open vSwitch agent | pod1-osd-compute-0.localdomain
| None              | False  | UP    | neutron-openvswitch-agent |
| ec19cb01-abbb-4773-8397-8739d9b0a349 | NIC Switch agent   | pod1-osd-compute-0.localdomain
| None              | False  | UP    | neutron-sriov-nic-agent   |

openstack network agent delete
```

```
[stack@director ~]$ openstack network agent delete c3ee92ba-aa23-480c-ac81-d3d8d01dcc03
[stack@director ~]$ openstack network agent delete ec19cb01-abbb-4773-8397-8739d9b0a349
```

从Nova和Ironic数据库中删除

从nova列表中删除节点以及讽刺数据库，然后进行验证。

```
[stack@director ~]$ source stackrc

[stack@al01-pod1-ospd ~]$ nova list | grep osd-compute-0
| c2cfa4d6-9c88-4ba0-9970-857d1a18d02c | pod1-osd-compute-0 | ACTIVE | -          | Running
| ctlplane=192.200.0.114 |

[stack@al01-pod1-ospd ~]$ nova delete c2cfa4d6-9c88-4ba0-9970-857d1a18d02c

nova show
```

```
[stack@director ~]$ nova show pod1-osd-compute-0 | grep hypervisor
| OS-EXT-SRV-ATTR:hypervisor_hostname  | 4ab21917-32fa-43a6-9260-02538b5c7a5a

ironic node-delete
```

```
[stack@director ~]$ ironic node-delete 4ab21917-32fa-43a6-9260-02538b5c7a5a
[stack@director ~]$ ironic node-list (node delete must not be listed now)
```

安装新计算节点

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》

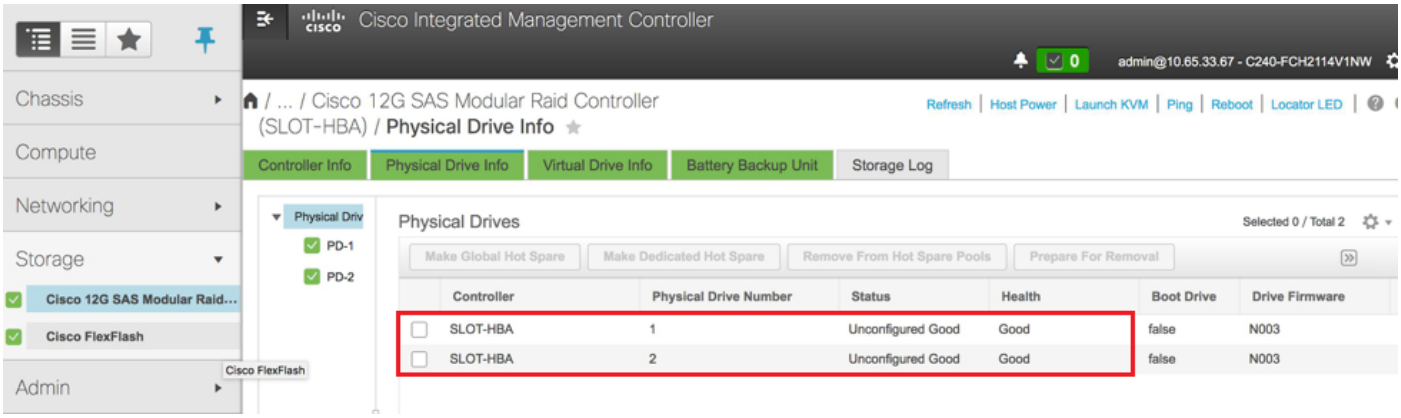
步骤1.安装服务器后，将硬盘作为旧服务器插入各插槽中。

步骤2.使用CIMC IP登录服务器。

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

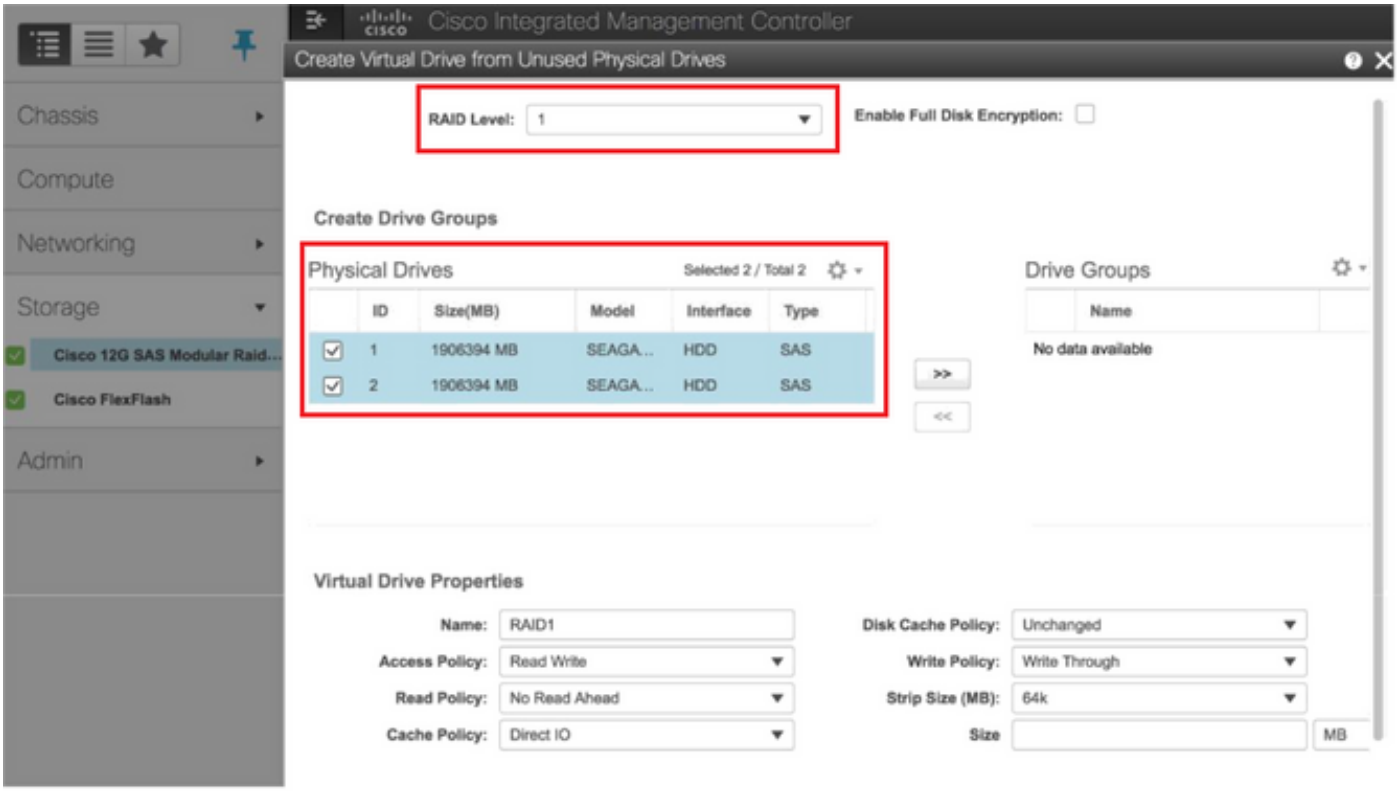
步骤4.检验物理驱动器的状态。它必须是未映**像良好**。

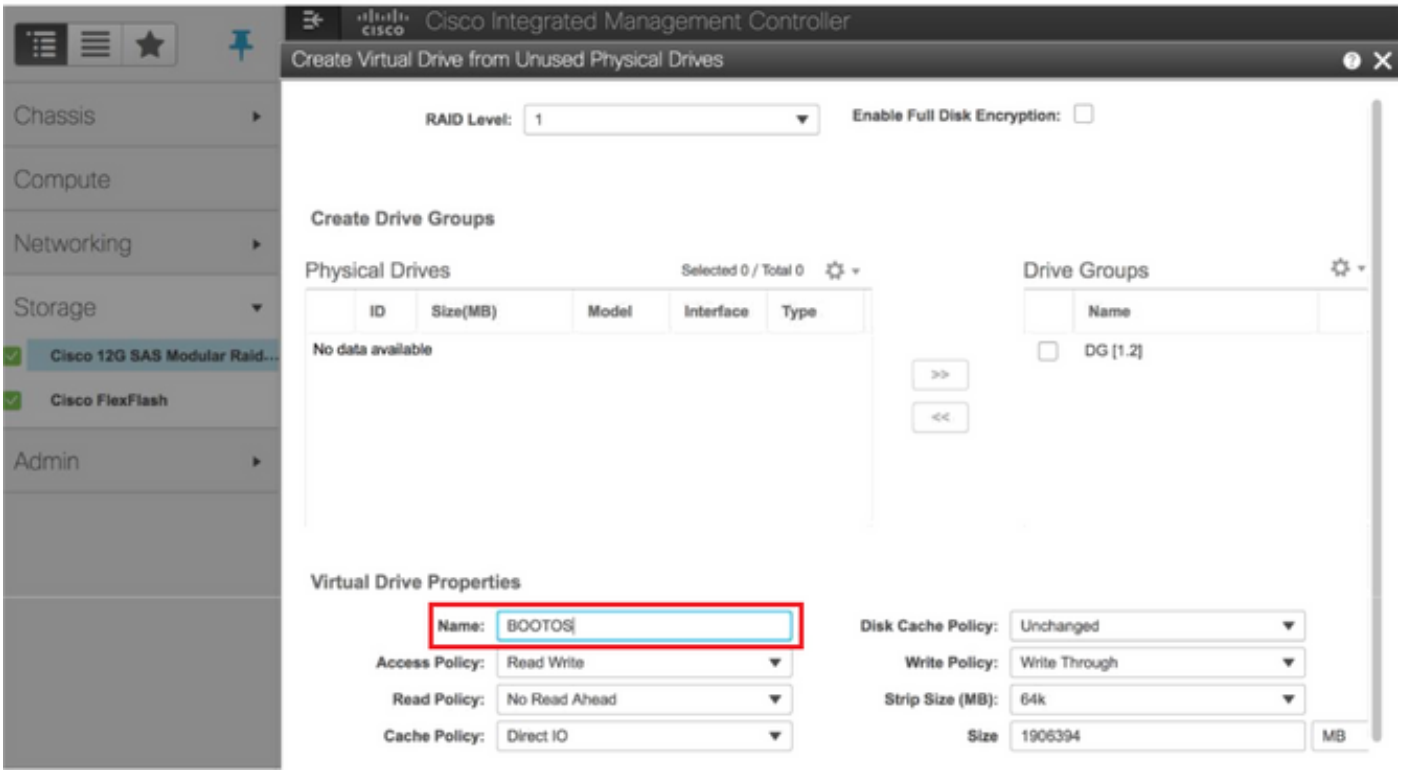
步骤5.从RAID级别为1的物理驱动器创建虚拟驱动器。



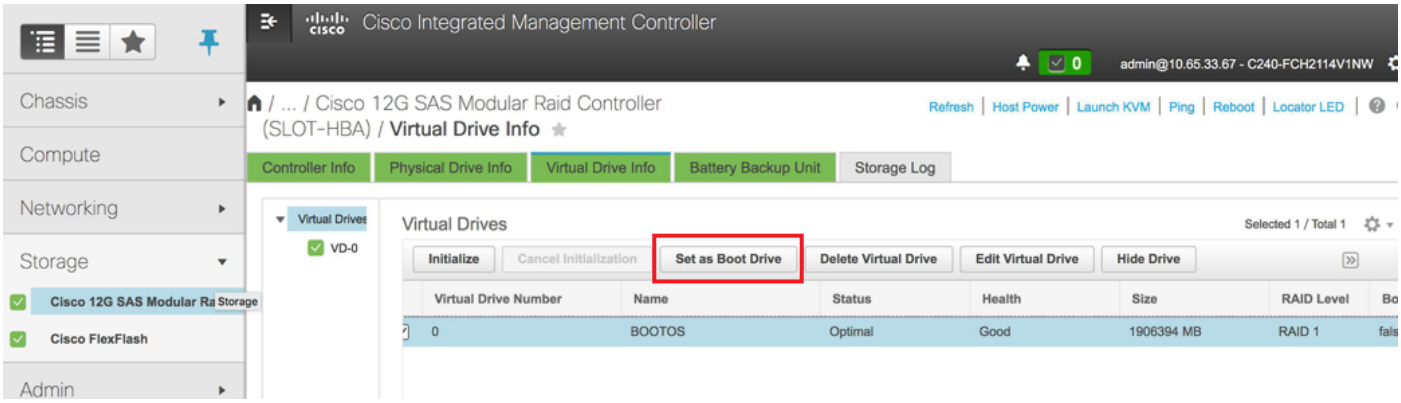
步骤6.导航至存储部分，选择Cisco 12G Sas模块化RAID控制器，并验证RAID控制器的状态和运行状况，如图所示。

> **注意**：以上图像仅用于说明，在实际OSD-Compute CIMC中，您会看到插槽[1,2,3,7,8,9,10]中的七个物理驱动器处于未映像良好状态，因为没有从它们创建虚拟驱动器。

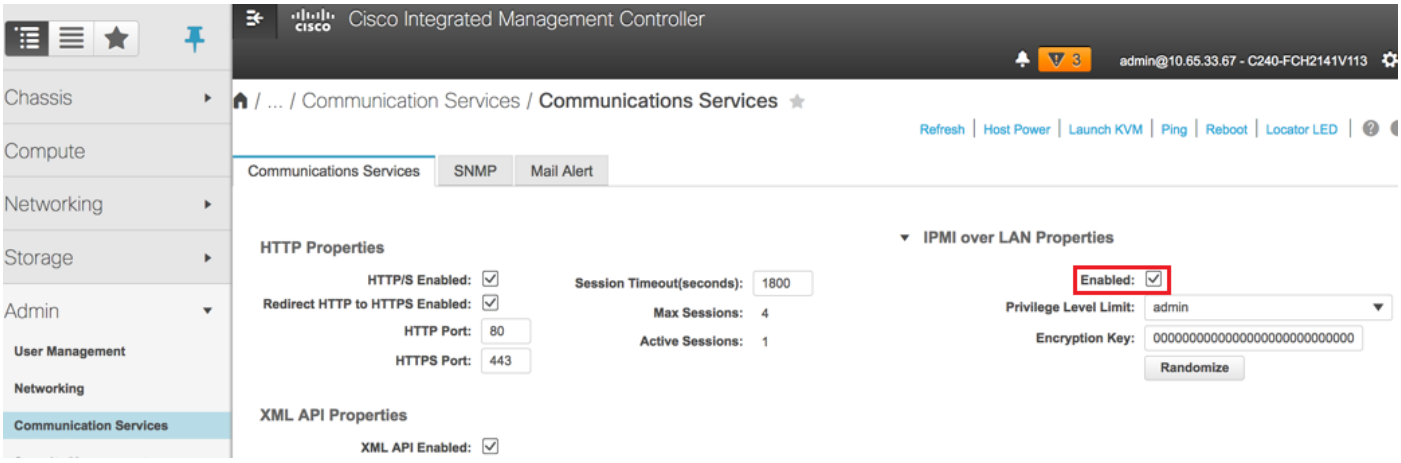




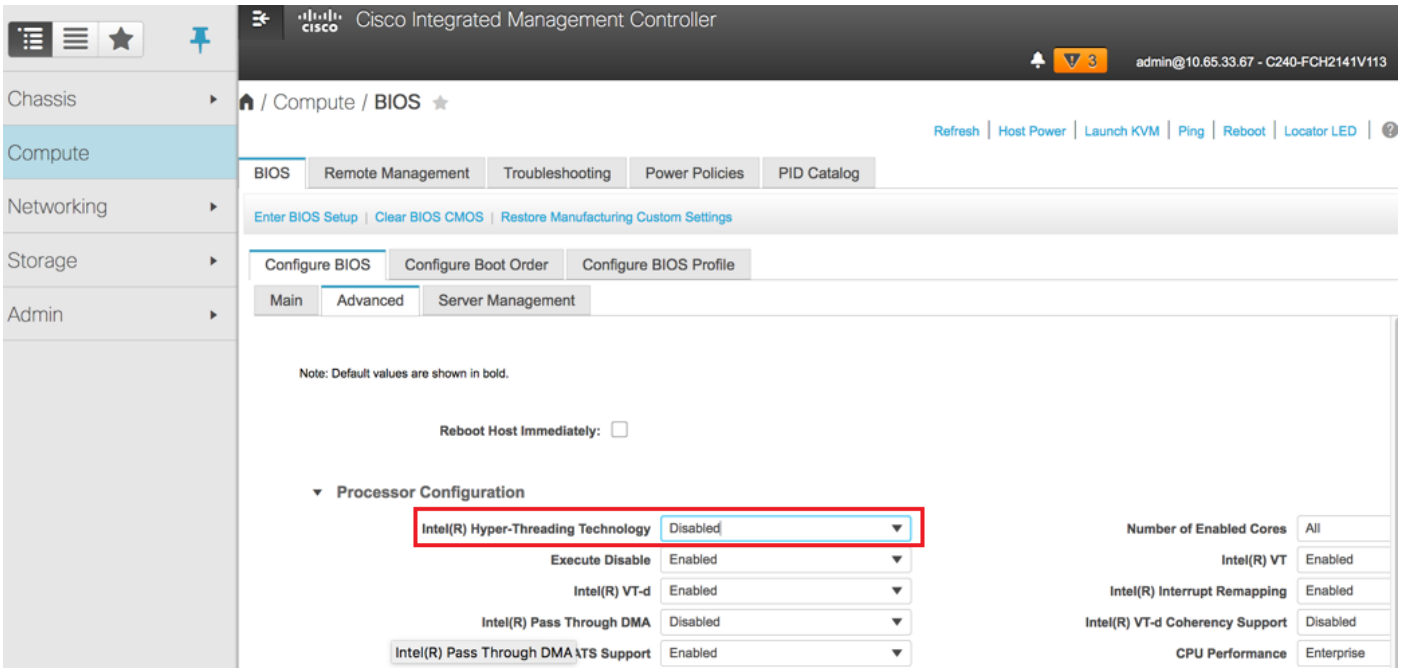
步骤7.现在，从控制器信息下的未使用物理驱动器创建虚拟驱动器，位于**Cisco 12G SAS模块化Raid控制器**下。



步骤8.选择VD并将设置**为引导驱动器。**



步骤9.从“管理”选项卡下的“通信服务”中启用IPMI over LAN。



步骤10.从“计算”节点下的高级BIOS配置中禁用超线程，如图所示。

步骤11.与使用物理驱动器1和2创建的BOOTOS VD类似，创建四个虚拟驱动器作为

日志 — 从物理驱动器号3

OSD1 — 从物理驱动器7

OSD2 — 从物理驱动器8

OSD3 — 从物理驱动器9

OSD4 — 从物理驱动器号10

步骤7.最后，物理驱动器和虚拟驱动器必须相似。

> **注意：**此处显示的图像和本节中提及的配置步骤均参考固件版本3.0(3e)，如果您使用其他版本，可能会略有变化。

将新的OSD-Compute节点添加到Overcloud

本节中提到的步骤是通用的，与计算节点托管的VM无关。

步骤1.添加具有不同索引的计算服务器。

创建仅**包含要添**加的新计算服务器详细信息的add_node.json文件。 确保以前未使用过新osd-compute服务器的索引号。通常，增加下一个最高的计算值。

示例：在2-vnf系统中，最先的是osd-compute-0，因此创建了osd-compute-3。

> **注意：**注意json格式。

```
[stack@director ~]$ cat add_node.json
{
   "nodes":[
       {
           "mac":[
               "<MAC_ADDRESS>"
           ],
           "capabilities": "node:osd-compute-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":"192.100.0.5"
       }
   ]
}
```

步骤2.导入json文件。

```
[stack@director ~]$ openstack baremetal import --json add_node.json
Started Mistral Workflow. Execution ID: 78f3b22c-5c11-4d08-a00f-8553b09f497d
Successfully registered node UUID 7eddfa87-6ae6-4308-b1d2-78c98689a56e
Started Mistral Workflow. Execution ID: 33a68c16-c6fd-4f2a-9df9-926545f2127e
```

```
Successfully set all nodes to available.
```

步骤3.使用上一步中记录的UUID运行节点内省。

```
[stack@director ~]$ openstack baremetal node manage 7eddfa87-6ae6-4308-b1d2-78c98689a56e
[stack@director ~]$ ironic node-list |grep 7eddfa87
| 7eddfa87-6ae6-4308-b1d2-78c98689a56e | None | None                                    | power off
  | manageable         | False       |

[stack@director ~]$ openstack overcloud node introspect 7eddfa87-6ae6-4308-b1d2-78c98689a56e --
provide
Started Mistral Workflow. Execution ID: e320298a-6562-42e3-8ba6-5ce6d8524e5c
Waiting for introspection to finish...
Successfully introspected all nodes.
Introspection completed.
Started Mistral Workflow. Execution ID: c4a90d7b-ebf2-4fcb-96bf-e3168aa69dc9
Successfully set all nodes to available.

[stack@director ~]$ ironic node-list |grep available
| 7eddfa87-6ae6-4308-b1d2-78c98689a56e | None | None                                    | power off
  | available          | False       |
```

步骤4.在OsdComputeIPs下将IP地址添加到custom-templates/layout.yml。在本例中，在替换osd-compute-0时，您会将该地址添加到每种类型的列表末尾。

```
OsdComputeIPs:

   internal_api:

   - 11.120.0.43

   - 11.120.0.44

   - 11.120.0.45

   - 11.120.0.43   <<< take osd-compute-0 .43 and add here

   tenant:

   - 11.117.0.43

   - 11.117.0.44

   - 11.117.0.45

   - 11.117.0.43   << and here

   storage:

   - 11.118.0.43

   - 11.118.0.44

   - 11.118.0.45

   - 11.118.0.43   << and here

   storage_mgmt:
```

```
   - 11.119.0.43

   - 11.119.0.44

   - 11.119.0.45

   - 11.119.0.43    << and here
```

步骤5.运行deploy.sh脚本，此脚本以前用于部署堆栈，以便将新的计算节点添加到超云堆栈。

```
[stack@director ~]$ ./deploy.sh
++ openstack overcloud deploy --templates -r /home/stack/custom-templates/custom-roles.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
/home/stack/custom-templates/network.yaml -e /home/stack/custom-templates/ceph.yaml -e
/home/stack/custom-templates/compute.yaml -e /home/stack/custom-templates/layout.yaml --stack
ADN-ultram --debug --log-file overcloudDeploy_11_06_17__16_39_26.log --ntp-server 172.24.167.109
--neutron-flat-networks phys_pcie1_0,phys_pcie1_1,phys_pcie4_0,phys_pcie4_1 --neutron-network-
vlan-ranges datacentre:1001:1050 --neutron-disable-tunneling --verbose --timeout 180
…
Starting new HTTP connection (1): 192.200.0.1
"POST /v2/action_executions HTTP/1.1" 201 1695
HTTP POST http://192.200.0.1:8989/v2/action_executions 201
Overcloud Endpoint: http://10.1.2.5:5000/v2.0
Overcloud Deployed
clean_up DeployOvercloud:
END return value: 0

real    38m38.971s
user    0m3.605s
sys     0m0.466s
```

步骤6.等待openstack堆栈状态为COMPLETE。

```
[stack@director ~]$  openstack stack list
+--------------------------------------+------------+----------------+----------------------+--
--------------------+
| ID                                   | Stack Name | Stack Status    | Creation Time        |
Updated Time         |
+--------------------------------------+------------+----------------+----------------------+--
--------------------+
| 5df68458-095d-43bd-a8c4-033e68ba79a0 | pod1 | UPDATE_COMPLETE | 2017-11-02T21:30:06Z | 2017-
11-06T21:40:58Z |
+--------------------------------------+------------+----------------+----------------------+--
--------------------+
```

步骤7.检查新osd-compute节点是否处于活动状态。

```
[stack@director ~]$ source stackrc
[stack@director ~]$ nova list |grep osd-compute-3
| 0f2d88cd-d2b9-4f28-b2ca-13e305ad49ea | pod1-osd-compute-3    | ACTIVE | -          | Running
  | ctlplane=192.200.0.117 |

[stack@director ~]$ source corerc
[stack@director ~]$ openstack hypervisor list |grep osd-compute-3
| 63 | pod1-osd-compute-3.localdomain    |
```

步骤8.登录新的osd-compute服务器并检查接口进程。初始时，状态在HEALTH_WARN中，当开始恢复时。

```
[heat-admin@pod1-osd-compute-3 ~]$ sudo ceph -s

   cluster eb2bb192-b1c9-11e6-9205-525400330666

    health HEALTH_WARN

           223 pgs backfill_wait

           4 pgs backfilling

           41 pgs degraded

           227 pgs stuck unclean

           41 pgs undersized

           recovery 45229/1300136 objects degraded (3.479%)

           recovery 525016/1300136 objects misplaced (40.382%)

    monmap e1: 3 mons at {Pod1-controller-0=11.118.0.40:6789/0,Pod1-controller-
1=11.118.0.41:6789/0,Pod1-controller-2=11.118.0.42:6789/0}

           election epoch 58, quorum 0,1,2 Pod1-controller-0,Pod1-controller-1,Pod1-controller-2

    osdmap e986: 12 osds: 12 up, 12 in; 225 remapped pgs

           flags sortbitwise,require_jewel_osds

     pgmap v781746: 704 pgs, 6 pools, 533 GB data, 344 kobjects

           1553 GB used, 11840 GB / 13393 GB avail

           45229/1300136 objects degraded (3.479%)

           525016/1300136 objects misplaced (40.382%)

                477 active+clean

                186 active+remapped+wait_backfill

                 37 active+undersized+degraded+remapped+wait_backfill

                  4 active+undersized+degraded+remapped+backfilling
```

步骤9.但是，在短时间（20分钟）后，CEPH将返回HEALTH_OK状态。

```
[heat-admin@pod1-osd-compute-3 ~]$ sudo ceph -s

   cluster eb2bb192-b1c9-11e6-9205-525400330666

    health HEALTH_OK

    monmap e1: 3 mons at {Pod1-controller-0=11.118.0.40:6789/0,Pod1-controller-
1=11.118.0.41:6789/0,Pod1-controller-2=11.118.0.42:6789/0}

           election epoch 58, quorum 0,1,2 Pod1-controller-0,Pod1-controller-1,Pod1-controller-2
```

```
    osdmap e1398: 12 osds: 12 up, 12 in
           flags sortbitwise,require_jewel_osds
     pgmap v784311: 704 pgs, 6 pools, 533 GB data, 344 kobjects
           1599 GB used, 11793 GB / 13393 GB avail
                704 active+clean
 client io 8168 kB/s wr, 0 op/s rd, 32 op/s wr
```

```
[heat-admin@pod1-osd-compute-3 ~]$ sudo ceph osd tree
ID WEIGHT   TYPE NAME                          UP/DOWN REWEIGHT PRIMARY-AFFINITY
-1 13.07996 root default
-2        0     host pod1-osd-compute-0
-3  4.35999     host pod1-osd-compute-2
 1  1.09000         osd.1                           up  1.00000          1.00000
 4  1.09000         osd.4                           up  1.00000          1.00000
 7  1.09000         osd.7                           up  1.00000          1.00000
10  1.09000         osd.10                          up  1.00000          1.00000
-4  4.35999     host pod1-osd-compute-1
 2  1.09000         osd.2                           up  1.00000          1.00000
 5  1.09000         osd.5                           up  1.00000          1.00000
 8  1.09000         osd.8                           up  1.00000          1.00000
11  1.09000         osd.11                          up  1.00000          1.00000
-5  4.35999     host pod1-osd-compute-3
 0  1.09000         osd.0                           up  1.00000          1.00000
 3  1.09000         osd.3                           up  1.00000          1.00000
 6  1.09000         osd.6                           up  1.00000          1.00000
 9  1.09000         osd.9                           up  1.00000          1.00000
```

恢复虚拟机

Nova聚合列表的附加项

将osd-compute节点添加到聚合主机并验证主机是否已添加。

nova aggregate-add-host

```
[stack@director ~]$ nova aggregate-add-host esc1 pod1-osd-compute-3.localdomain
```

nova aggregate-show

```
[stack@director ~]$ nova aggregate-show esc1
+----+------+-------------------+---------------------------------------+--------------------
--------------------+
| Id | Name | Availability Zone | Hosts | Metadata |
+----+------+-------------------+---------------------------------------+--------------------
--------------------+
| 3 | esc1 | AZ-esc1 | 'pod1-osd-compute-3.localdomain' | 'availability_zone=AZ-esc1',
'esc1=true' |
+----+------+-------------------+---------------------------------------+--------------------
--------------------+
```

ESC VM恢复

步骤1.从nova列表中检查ESC VM的状态并将其删除。

```
stack@director scripts]$ nova list |grep esc
| c566efbf-1274-4588-a2d8-0682e17b0d41 | esc                                                    |
ACTIVE | -          | Running     | VNF2-UAS-uas-orchestration=172.168.11.14; VNF2-UAS-uas-
management=172.168.10.4
                                |
[stack@director scripts]$ nova delete esc
Request to delete server esc has been accepted.

If can not delete esc then use command: nova force-delete esc
```

步骤2.在OSPD中，导航到ECS-Image目录，并确保ESC版本的bootvm.py和qcow2存在（如果不将其移动到目录）。

```
[stack@atospd ESC-Image-157]$ ll

total 30720136

-rw-r--r--. 1 root  root       127724 Jan 23 12:51 bootvm-2_3_2_157a.py

-rw-r--r--. 1 root  root           55 Jan 23 13:00 bootvm-2_3_2_157a.py.md5sum

-rw-rw-r--. 1 stack stack 31457280000 Jan 24 11:35 esc-2.3.2.157.qcow2
```

步骤3.创建映像。

```
[stack@director ESC-image-157]$  glance image-create --name ESC-2_3_2_157 --disk-format "qcow2"
--container "bare" --file /home/stack/ECS-Image-157/ESC-2_3_2_157.qcow2
```

步骤4.检验ESC映像是否存在。

```
stack@director ~]$ glance image-list
+--------------------------------------+--------------------------------------+
| ID                                   | Name                                 |
+--------------------------------------+--------------------------------------+
| 8f50acbe-b391-4433-aa21-98ac36011533 | ESC-2_3_2_157|
| 2f67f8e0-5473-467c-832b-e07760e8d1fa | tmobile-pcrf-13.1.1.iso              |
| c5485c30-45db-43df-831d-61046c5cfd01 | tmobile-pcrf-13.1.1.qcow2            |
| 2f84b9ec-61fa-46a3-a4e6-45f14c93d9a9 | tmobile-pcrf-13.1.1_cco_20170825.iso |
| 25113ecf-8e63-4b81-a73f-63606781ef94 | wscaaa01-sept072017                  |
| 595673e8-c99c-40c2-82b1-7338325024a9 | wscaaa02-sept072017                  |
| 8bce3a60-b3b0-4386-9e9d-d99590dc9033 | wscaaa03-sept072017                  |
| e5c835ad-654b-45b0-8d36-557e6c5fd6e9 | wscaaa04-sept072017                  |
| 879dfcde-d25c-4314-8da0-32e4e73ffc9f | WSP1_cluman_12_07_2017               |
| 7747dd59-c479-4c8a-9136-c90ec894569a | WSP2_cluman_12_07_2017               |
+--------------------------------------+--------------------------------------+

[stack@ ~]$ openstack flavor list
+--------------------------------------+------------+--------+------+-----------+-------+-------
----+
| ID                                   | Name       |    RAM | Disk | Ephemeral | VCPUs | Is
Public |
+--------------------------------------+------------+--------+------+-----------+-------+-------
----+
| 1e4596d5-46f0-46ba-9534-cfdea788f734 | pcrf-smb   | 100352 |  100 |         0 |     8 | True
|
| 251225f3-64c9-4b19-a2fc-032a72bfe969 | pcrf-oam   |  65536 |  100 |         0 |    10 | True
|
| 4215d4c3-5b2a-419e-b69e-7941e2abe3bc | pcrf-pd    |  16384 |  100 |         0 |    12 | True
|
| 4c64a80a-4d19-4d52-b818-e904a13156ca | pcrf-qns   |  14336 |  100 |         0 |    10 | True
|
| 8b4cbba7-40fd-49b9-ab21-93818c80a2e6 | esc-flavor |   4096 |    0 |         0 |     4 | True
|
| 9c290b80-f80a-4850-b72f-d2d70d3d38ea | pcrf-sm    | 100352 |  100 |         0 |    10 | True
|
| e993fc2c-f3b2-4f4f-9cd9-3afc058b7ed1 | pcrf-arb   |  16384 |  100 |         0 |     4 | True
|
| f2b3b925-1bf8-4022-9f17-433d6d2c47b5 | pcrf-cm    |  14336 |  100 |         0 |     6 | True
|
+--------------------------------------+------------+--------+------+-----------+-------+-------
----+
```

步骤5.在image目录下创建此文件并启动ESC实例。

```
[root@director ESC-IMAGE]# cat esc_params.conf
openstack.endpoint = publicURL

[root@director ESC-IMAGE]./bootvm-2_3_2_157a.py esc --flavor esc-flavor --image ESC-2_3_2_157 --
net tb1-mgmt --gateway_ip 172.16.181.1 --net tb1-orch --enable-http-rest --avail_zone AZ-esc1 --
user_pass "admin:Cisco123" --user_confd_pass "admin:Cisco123" --bs_os_auth_url
http://10.250.246.137:5000/v2.0 --kad_vif eth0 --kad_vip 172.16.181.5 --ipaddr 172.16.181.4 dhcp
--ha_node_list 172.16.181.3 172.16.181.4 --esc_params_file esc_params.conf
```

bootvm.pyESC VMESC HAESC

步骤6.登录新的ESC并验证备份状态。

```
[admin@esc ~]$ escadm status
0 ESC status=0 ESC Backup Healthy

[admin@VNF2-esc-esc-1 ~]$ health.sh
============== ESC HA (BACKUP) ================================================
ESC HEALTH PASSED
```