

OSDのPCRFによるUCS 240M4コンピューティング

内容

[概要](#)

[背景説明](#)

[ヘルスチェック](#)

[バックアップ](#)

[OSDコンピューティングノードでホストされるVMの特定](#)

[グレースフルパワーオフ](#)

[ESCをスタンバイモードに移行](#)

[Osd-Computeノードの削除](#)

[オーバークラウドから削除](#)

[サービスリストからのOsd-Computeノードの削除](#)

[Neutronエージェントの削除](#)

[NovaおよびIronicデータベースから削除](#)

[新しいコンピューティングノードのインストール](#)

[新しいOSD-ComputeノードをOvercloudに追加します](#)

[VMのリストア](#)

[Nova集約リストへの追加](#)

[ESC VMのリカバリ](#)

概要

このドキュメントでは、Cisco Policy Suite(CPS)Virtual Network Functions(VNF)をホストするUltra-Mセットアップで問題のあるosd-computeサーバを交換するために必要な手順について説明します。

背景説明

このドキュメントは、Cisco Ultra-Mプラットフォームに精通したシスコ担当者を対象としており、OSD-Compute Serverの交換時にOpenStackおよびCPS VNFレベルで実行する必要がある手順の詳細を説明しています。

注：このドキュメントの手順を定義するために、Ultra M 5.1.xリリースが検討されています。

ヘルスチェック

Osd-Computeノードを交換する前に、Red Hat OpenStack Platform環境の現在の状態を確認することが重要です。コンピューティングの交換プロセスがオンの場合に複雑にならないように、現在の状態を確認することをお勧めします。

OSPDから

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

ステップ1:15分ごとに生成されるultam-healthレポートからシステムの健全性を確認します。

```
[stack@director ~]# cd /var/log/cisco/ultram-health
ファイルultam_health_os.reportをチェックします。
```

サービスはneutron-sriov-nic-agent.service**だけ**で、XXXのステータスとして表示されます。

ステップ2：すべてのコントローラでrabbitmqが実行され、OSPDから実行されているかどうかを確認します。

```
[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
```

ステップ3:stonithが有効になっていることを確認します。

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

すべてのコントローラでPCSのステータスを確認

- すべてのコントローラノードがhaproxy-clone
- すべてのコントローラノードがMasterの下にあります
- すべてのコントローラノードがRabbitmqの下で起動されます
- 1台のコントローラノードがマスターで、2台のスレーブがredisの下にあります

OSPDから

```
[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
```

ステップ4：すべてのopenstackサービスがアクティブであることを確認します。OSPDから次のコマンドを実行します。

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

ステップ5：コントローラのCEPHステータスがHEALTH_OKであることを確認します。

```
[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
```

ステップ6: OpenStackコンポーネントのログを確認します。エラーを探します。

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
```

ステップ7:OSPDから、APIに対してこれらの検証を実行します。

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

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

```
[stack@director ~]$ glance image-list
```

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

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

ステップ8: サービスの健全性を確認します。

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
```

バックアップ

リカバリの場合は、次の手順を使用してOSPDデータベースのバックアップを取ることを推奨します。

ステップ1: Mysqlダンプを取得します。

```
[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
```

このプロセスにより、インスタンスの可用性に影響を与えることなく、ノードを確実に交換できます。

ステップ2: Cluster Manager VMからCPS VMをバックアップするには、次の手順を実行します。

```
[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
```

OSDコンピューティングノードでホストされるVMの特定

コンピューティングサーバでホストされているVMを特定します。

ステップ1: コンピューティングサーバにElastic Services Controller(ESC)が含まれています。

```
[stack@director ~]$ nova list --field name,host,networks | grep osd-compute-1  
| 50fd1094-9c0a-4269-b27b-cab74708e40c | esc | pod1-osd-compute-0.localdomain  
| tbl1-orch=172.16.180.6; tbl1-mgmt=172.16.181.3
```

注: ここに示す出力では、最初のカラムはUniversal Unique Identifier(UUID)に対応し、2番目のカラムはVM名を表し、3番目のカラムはVMが存在するホスト名を表しています。この出力のパラメータは、以降のセクションで使用します。

注: 置き換えるOSDコンピュートノードが完全にダウンしていてアクセスできない場合は、「NovaアグリゲーションリストからOsd-コンピュートノードを削除する」セクションに進みます。それ以外の場合は、次のセクションから進みます。

ステップ2: CEPHに、単一のOSDサーバを削除できる容量があることを確認します。

```
[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

ステップ3: ceph osdツリーのステータスがosd-computeサーバでupであることを確認します。

```
[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

ステップ4:CEPHプロセスがosd-computeサーバでアクティブになります。

```
[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
```

ステップ5：各cephインスタンスを無効にして停止し、各インスタンスをosdから削除して、ディレクトリをアンマウントします。cephインスタンスごとに繰り返します。

```
[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
```

(または)

ステップ6:Clean.shスクリプトを使用して、上記のタスクを一度に実行できます。

```
[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
```

すべてのOSDプロセスが移行/削除されると、ノードをオーバークラウドから削除できます。

注：CEPHが削除されると、VNF HD RAIDはDegraded状態になりますが、hdディスクにアクセスできる必要があります。

グレースフルパワーオフ

ESCをスタンバイモードに移行

ステップ1: コンピューティングノードでホストされているESCにログインし、マスター状態であるかどうかを確認します。存在する場合は、ESCをスタンバイモードに切り替えます。

```
[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!
```

ステップ2: Nova集約リストからOsd-Computeノードを削除します。

- novaアグリゲートをリストし、ホストされているVNFに基づいてコンピューティングサーバに対応するアグリゲートを特定します。通常、形式は<VNFNAME>-EM-MGMT<X>および<VNFNAME>-CF-MGMT<X>です

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

この例では、osd-computeサーバはesc1に属しています。したがって、対応する集約はesc1になります

ステップ3: 識別された集約からosd-computeノードを削除します。

```
nova aggregate-remove-host
```

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

ステップ4: osd-computeノードが集約から削除されているかどうかを確認します。ここで、集約の下にホストがリストされていないことを確認します。

```
nova aggregate-show
```



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

Osd-Computeノードの削除

このセクションで説明する手順は、コンピューティングノードでホストされるVMに関係なく共通です。

オーバークラウドから削除

ステップ1：次に示す内容のdelete_node.shという名前のスクリプトファイルを作成します。記載されているテンプレートが、スタック配置に使用されるdeploy.shスクリプトで使用されるテンプレートと同じであることを確認します。

```
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
```

ステップ2:OpenStackスタックの動作がCOMPLETE状態になるまで待ちます。

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

```

サービスリストからのOsd-Computeノードの削除

サービスリストからコンピューティングサービスを削除します。

```

[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
```

Neutronエージェントの削除

古い関連付けられたNeutronエージェントを削除し、コンピューティングサーバのvswitchエージェントを開きます。

```

[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
```

NovaおよびIronicデータベースから削除

Novaリストからノードを皮肉なデータベースとともに削除し、確認します。

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

```

[stack@a101-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@a101-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)
```

新しいコンピューティングノードのインストール

新しいUCS C240 M4サーバのインストール手順と初期セットアップ手順については、『[Cisco UCS C240 M4サーバインストールおよびサービスガイド](#)』を参照してください。

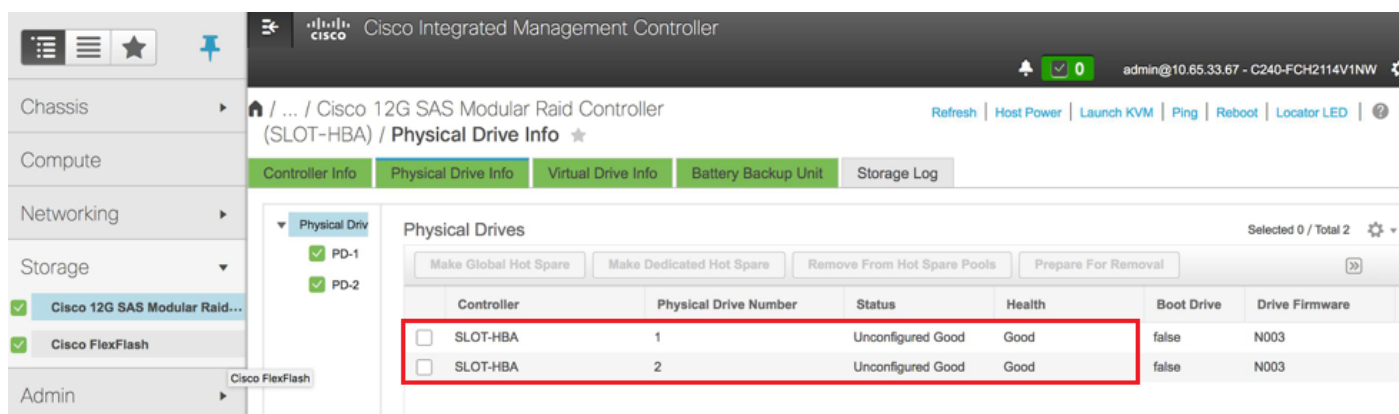
ステップ1：サーバのインストール後、ハードディスクを古いサーバとしてそれぞれのスロットに挿入します。

ステップ2:CIMC IPを使用してサーバにログインします。

ステップ3：ファームウェアが以前に使用した推奨バージョンと異なる場合は、BIOSアップグレードを実行します。BIOSアップグレードの手順は次のとおりです。[Cisco UCS CシリーズラックマウントサーバBIOSアップグレードガイド](#)

ステップ4：物理ドライブのステータスを確認します。それはUnconimaged Goodである必要があります。

ステップ5:RAIDレベル1の物理ドライブから仮想ドライブを作成します。



Controller	Physical Drive Number	Status	Health	Boot Drive	Drive Firmware
<input type="checkbox"/> SLOT-HBA	1	Unconfigured Good	Good	false	N003
<input type="checkbox"/> SLOT-HBA	2	Unconfigured Good	Good	false	N003

ステップ6：ストレージセクションに移動し、Cisco 12G Sas Modular Raid Controllerを選択し、図に示すようにRAIDコントローラのステータスと健全性を確認します。

注：上の図は例示の目的に限られます。実際のOSD-Compute CIMCでは、スロット [1,2,3,7,8,9,10]の7つの物理ドライブがunconimaged状態になっています。仮想ドライブは作成されません。

The screenshot shows the 'Create Virtual Drive from Unused Physical Drives' interface. The RAID Level is set to 1. Two physical drives are selected in the 'Physical Drives' table. The 'Virtual Drive Properties' section shows the name 'RAID1' and various policies.

ID	Size(MB)	Model	Interface	Type
1	1906394 MB	SEAGA...	HDD	SAS
2	1906394 MB	SEAGA...	HDD	SAS

Virtual Drive Properties:

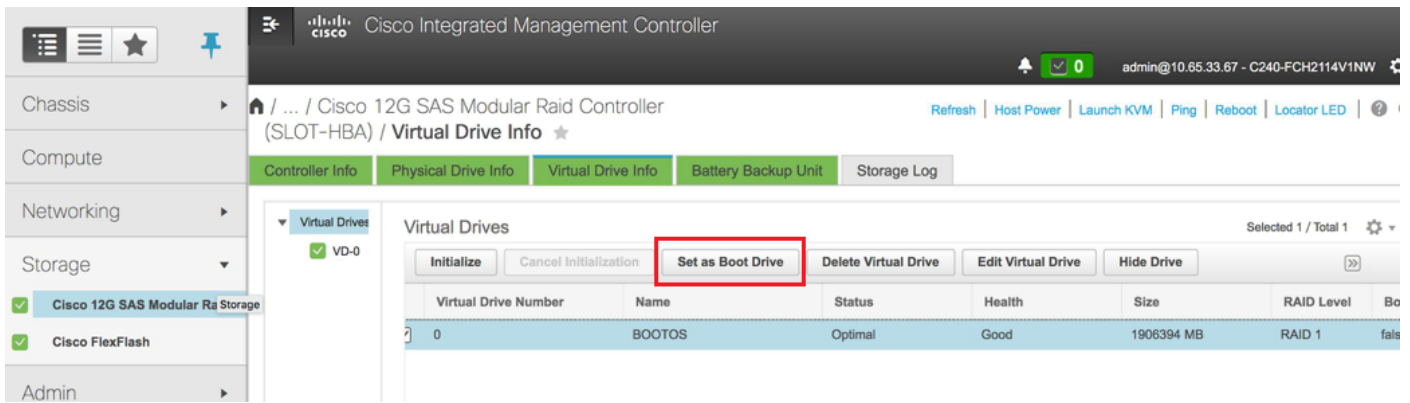
- Name: RAID1
- Access Policy: Read Write
- Read Policy: No Read Ahead
- Cache Policy: Direct IO
- Disk Cache Policy: Unchanged
- Write Policy: Write Through
- Strip Size (MB): 64k
- Size: [] MB

The screenshot shows the 'Create Virtual Drive from Unused Physical Drives' interface. The RAID Level is set to 1. No physical drives are selected. The 'Virtual Drive Properties' section shows the name 'BOOTOS' and various policies.

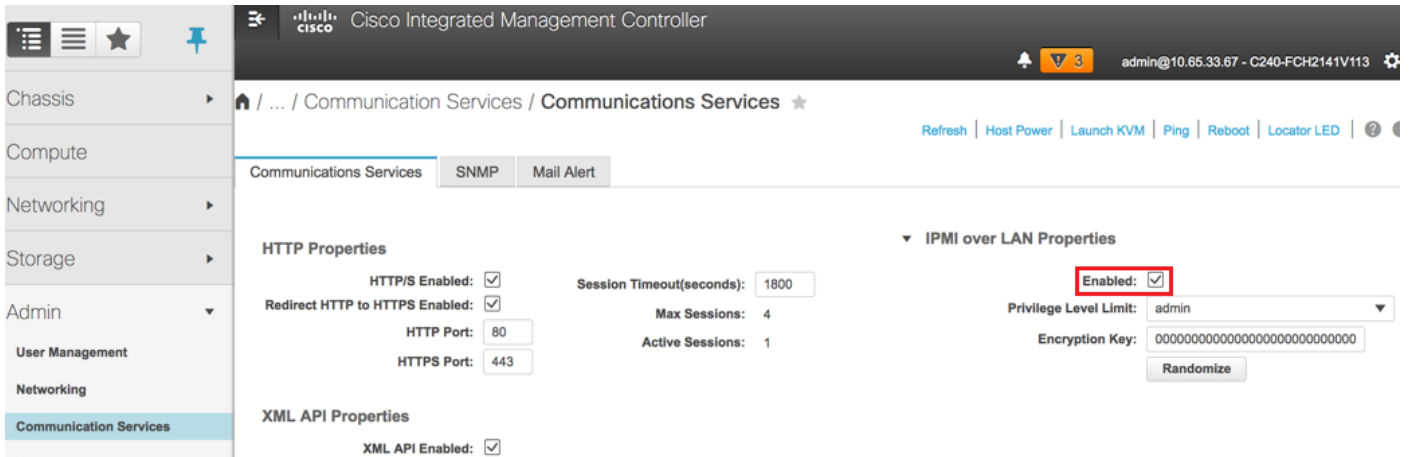
Virtual Drive Properties:

- Name: BOOTOS
- Access Policy: Read Write
- Read Policy: No Read Ahead
- Cache Policy: Direct IO
- Disk Cache Policy: Unchanged
- Write Policy: Write Through
- Strip Size (MB): 64k
- Size: 1906394 MB

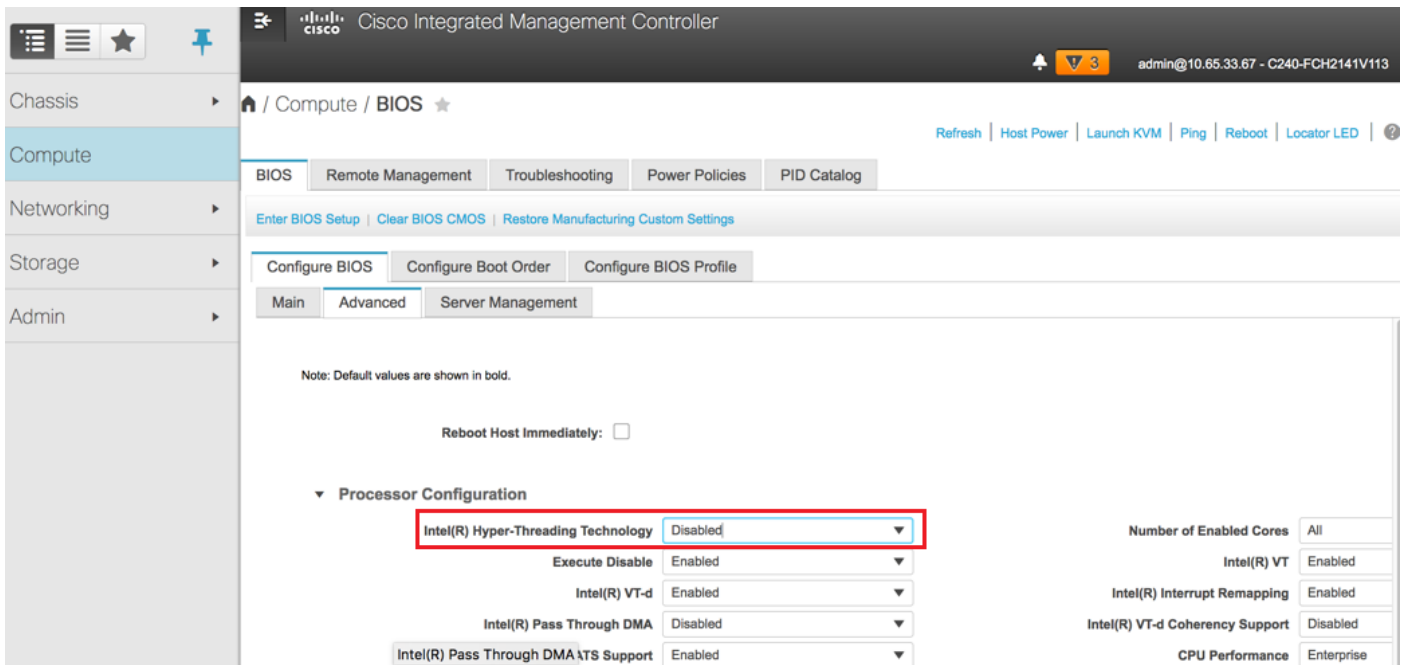
ステップ7：コントローラ情報から未使用の物理ドライブから仮想ドライブを作成します。Cisco 12G SASモジュラRaidコントローラの下に。



ステップ8:VDを選択し、ブートドライブとして設定します。



ステップ9:[Admin]タブの[Communication services]からIPMI over LANを有効にします。



ステップ10 : 図に示すように、ComputeノードのAdvance BIOS設定からハイパースレッディングをディセーブルにします。

ステップ11 : 物理ドライブ1および2で作成されたBOOTOS VDと同様に、次の4つの仮想ドライブを作成します。

ジャーナル – 物理ドライブ番号3から

OSD1 – 物理ドライブ番号7から

OSD2 – 物理ドライブ番号8から

OSD3 – 物理ドライブ番号9から

OSD4 – 物理ドライブ番号10から

ステップ7：最後に、物理ドライブと仮想ドライブは類似している必要があります。

注：このセクションで説明するイメージと設定手順は、ファームウェアバージョン3.0(3e)を参照するもので、他のバージョンで作業する場合は、若干の違いがあります。

新しいOSD-ComputeノードをOvercloudに追加します

このセクションで説明する手順は、コンピューティングノードによってホストされるVMに関係なく共通です。

ステップ1：異なるインデックスを持つコンピューティングサーバを追加します。

追加する新しいコンピュートサーバの詳細のみを含むadd_node.jsonファイルを作成します。新しいosd-computeサーバのインデックス番号が以前に使用されていないことを確認します。通常、次に高い計算値を増分します。

例：最も前のバージョンはosd-compute-0で、2-vnfシステムの場合はosd-compute-3が作成されました。

注：json形式に注意してください。

```
[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"
    }
  ]
}
```

ステップ2: jsonファイルをインポートします。

```
[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.
```

ステップ3：前のステップでメモしたUUIDを使用して、ノードのイントロスペクションを実行します。

```
[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 |
```

ステップ4: OsdComputeIPsの下のcustom-templates/layout.ymlにIPアドレスを追加します。この場合、osd-compute-0を置き換えると、そのアドレスが各タイプのリストの最後に追加されます。

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

```

ステップ5: 新しいコンピューターノードをオーバークラウドスタックに追加するため、**deploy.sh**スクリプトを実行します。このスクリプトは、以前にスタックの展開に使用していました。

```

[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

```

ステップ6: openstackスタックのステータスがCOMPLETEになるまで待ちます。

```

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

```

ステップ7: 新しいosd-computeノードがアクティブ状態であることを確認します。

```

[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 |
```

ステップ8 : 新しいosd-computeサーバにログインし、cephプロセスを確認します。最初は、cephが回復したため、ステータスはHEALTH_WARNです。

```
[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
```

ステップ9 : ただし、短時間 (20分) が経過すると、CEPHはHEALTH_OK状態に戻ります。

```
[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

VMのリストア

Nova集約リストへの追加

osd-computeノードを集約ホストに追加し、ホストが追加されているかどうかを確認します。

```
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のリカバリ

ステップ1: ノバリストからESC VMのステータスを確認し、削除します。

```
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

ステップ2: OSPDでECS-Imageディレクトリに移動し、ESCリリースのbootvm.pyとqcow2が存在することを確認します (ディレクトリに移動しない場合)。

```
[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
```

ステップ3 : イメージを作成します。

```
[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
```

ステップ4:ESCイメージが存在することを確認します。

```
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

ステップ5 : イメージディレクトリにこのファイルを作成し、ESCインスタンスを起動します。

```
[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_confid_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
```

ステップ6：新しいESCにログインし、バックアップの状態を確認します。

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

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