

OSD-Compute UCS 240M4 - vEPC 교체

목차

[소개](#)

[배경 정보](#)

[MoP의 워크플로](#)

[약어](#)

[사전 요구 사항](#)

[백업 OSPD](#)

[OSD-Compute 노드에서 호스팅되는 VM 식별](#)

[정상 전원 끄기](#)

[케이스 1. OSD-Compute Node Hosts CF/ESC/EM/UAS](#)

[CF 카드를 대기 상태로 마이그레이션](#)

[ESC에서 CF 및 EM VM 종료](#)

[ESC를 대기 모드로 마이그레이션](#)

-

[Nova Aggregate List\(Nova 집계 목록\)에서 OSD-Compute Node\(OSD-컴퓨팅 노드\) 제거](#)

[사례 2. OSD-Compute Node가 Auto-Deploy/Auto-IT/EM/UAS를 호스팅함](#)

[자동 구축의 CDB 백업](#)

[자동 IT에서 system.cfg 백업](#)

[OSD-Compute Node 삭제](#)

[서비스 목록에서 OSD-Compute 노드 삭제](#)

[중성자 에이전트 삭제](#)

[Nova 및 Ironic 데이터베이스에서 삭제](#)

[오버클라우드에서 삭제](#)

[새 컴퓨팅 노드 설치](#)

[오버클라우드에 새 OSD-Compute 노드 추가](#)

[사후 서버 교체 설정](#)

[VM 복원](#)

[케이스 1. CF, ESC, EM 및 UAS를 호스팅하는 OSD-컴퓨팅 노드](#)

[Nova Aggregate List에 추가](#)

[UAS VM 복구](#)

[ESC VM 복구](#)

[ESC에서 CF 및 EM VM 복구](#)

[사례 2. 자동 IT, 자동 배포, EM 및 UAS를 호스팅하는 OSD-Compute Node](#)

[자동 구축 VM 복구](#)

[Auto-IT VM 복구](#)

[ESC 복구 실패 처리](#)

[자동 구축 컨피그레이션 업데이트](#)

[Syslog 활성화](#)

소개

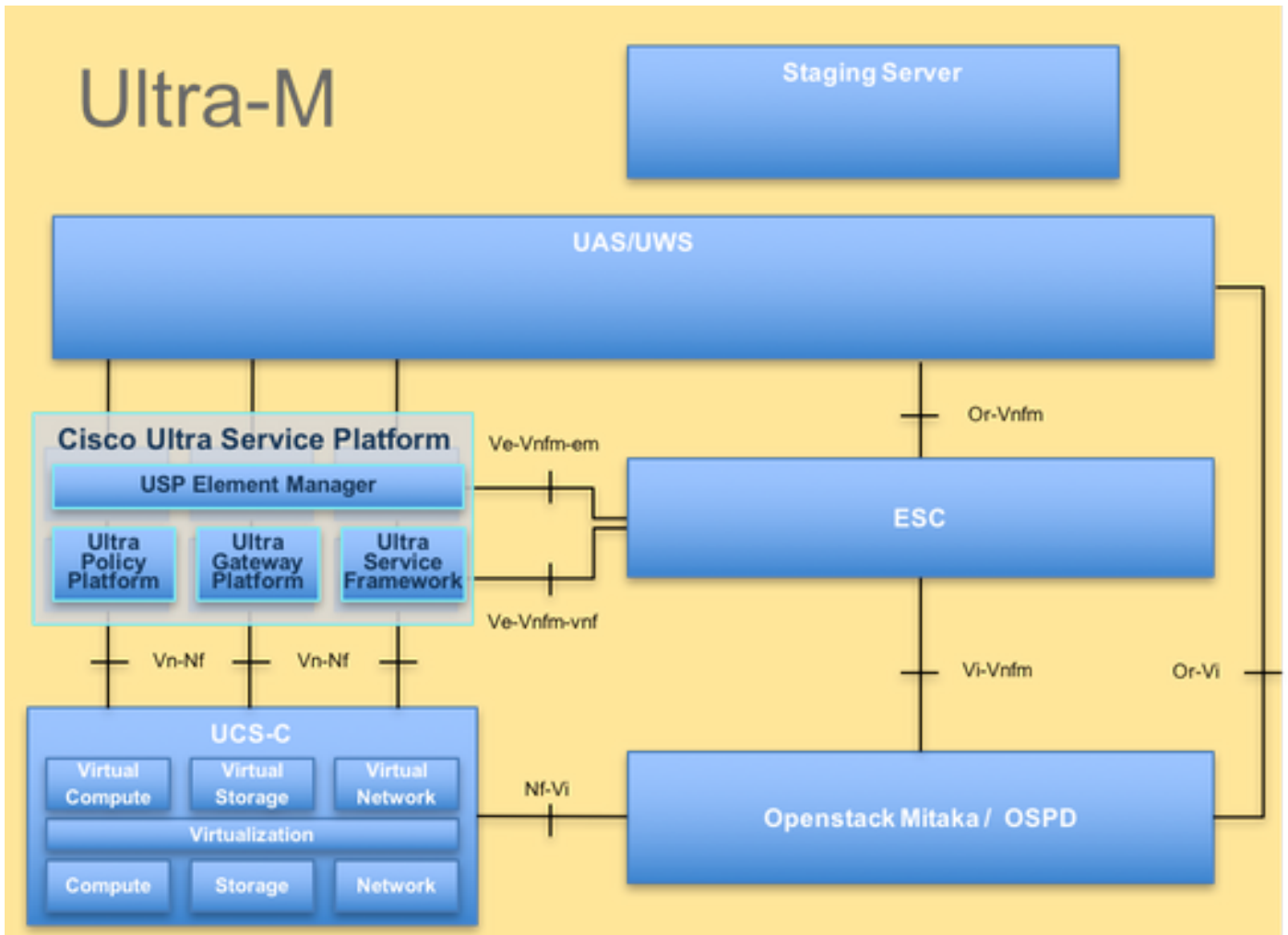
이 문서에서는 StarOS VNF(Virtual Network Functions)를 호스팅하는 Ultra-M 설정에서 결합이 있는 OSD(Object Storage Disk)-Compute 서버를 교체하는 데 필요한 단계에 대해 설명합니다.

배경 정보

Ultra-M은 VNF의 구축을 단순화하기 위해 설계된 사전 패키지 및 검증된 가상 모바일 패킷 코어 솔루션입니다. OpenStack은 Ultra-M용 VIM(Virtualized Infrastructure Manager)이며 다음 노드 유형으로 구성됩니다.

- 컴퓨팅
- OSD - 컴퓨팅
- 컨트롤러
- OpenStack 플랫폼 - 디렉터(OSPD)

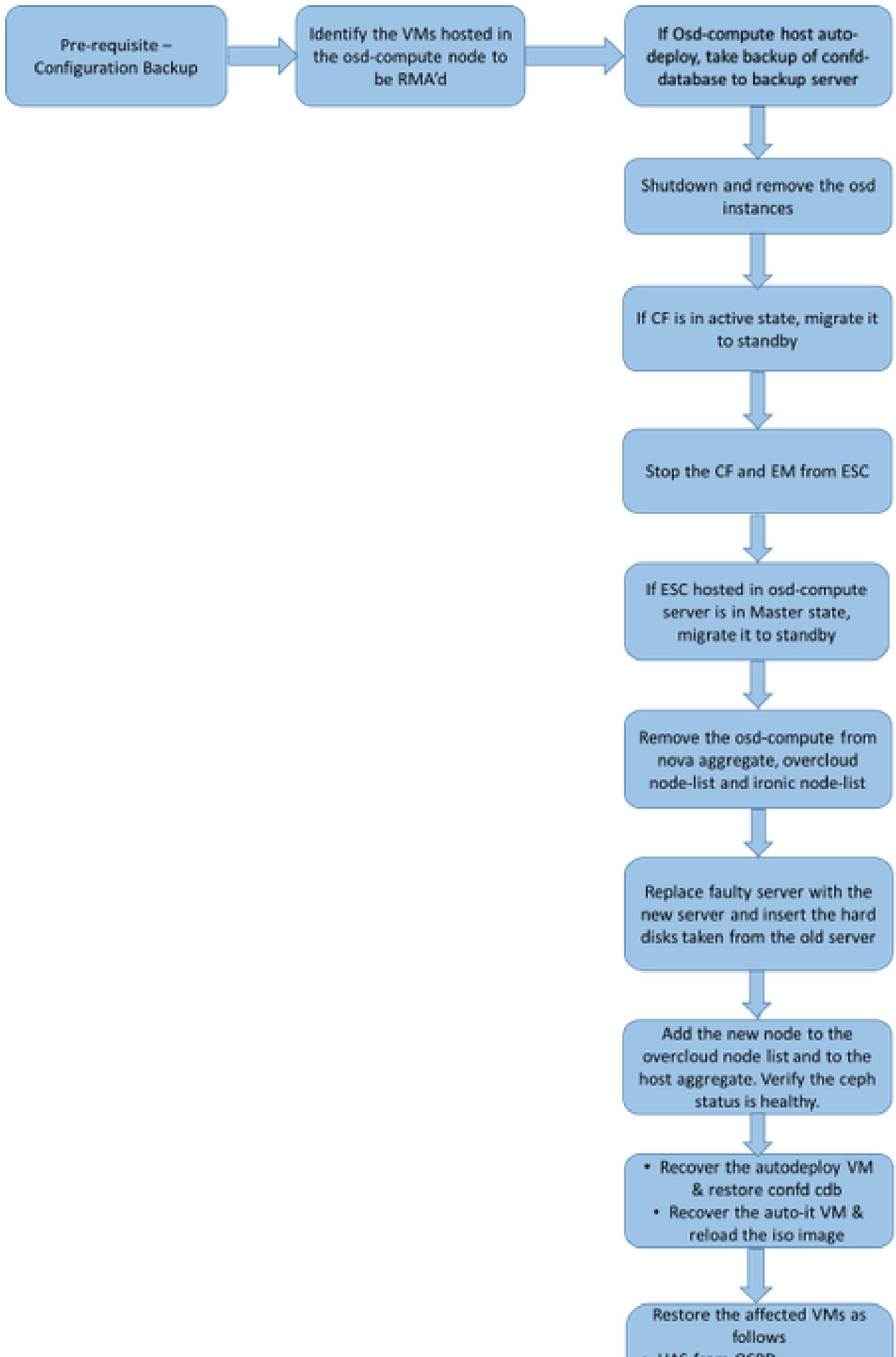
이 그림에는 Ultra-M의 고급 아키텍처와 관련 구성 요소가 나와 있습니다.



이 문서는 Cisco Ultra-M 플랫폼에 익숙한 Cisco 직원을 대상으로 하며, 컴퓨팅 서버 교체 시 OpenStack 및 StarOS VNF 레벨에서 수행해야 하는 단계에 대해 자세히 설명합니다.

 참고: 이 문서의 절차를 정의하기 위해 Ultra M 5.1.x 릴리스가 고려됩니다.

MoP의 워크플로



| | |
|-------|----------------------------|
| CF | 제어 기능 |
| SF | 서비스 기능 |
| Esc 키 | Elastic Service Controller |
| 자루걸레 | 절차 방법 |
| OSD | 개체 스토리지 디스크 |
| HDD | 하드 디스크 드라이브 |
| SSD | SSD(Solid State Drive) |
| 빔 | 가상 인프라 관리자 |
| VM | 가상 머신 |
| 엠펜 | 요소 관리자 |
| UAS | Ultra Automation 서비스 |
| UUID | 보편적으로 고유한 식별자 |

사전 요구 사항

백업 OSPD

OSD-Compute 노드를 교체하기 전에 Red Hat OpenStack Platform 환경의 현재 상태를 확인하는 것이 중요합니다. Compute replacement process(컴퓨팅 교체 프로세스)가 켜져 있을 때 복잡성을 피하기 위해 현재 상태를 확인하는 것이 좋습니다. 이러한 교체 흐름으로 달성할 수 있습니다.

복구의 경우 다음 단계를 사용하여 OSPD 데이터베이스(DB)의 백업을 수행하는 것이 좋습니다.

```
[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
/etc/my.cnf.d/server.cnf /var/lib/glance/images /srv/node /home/stack
tar: Removing leading `/' from member names
```

이 프로세스는 인스턴스의 가용성에 영향을 주지 않고 노드를 교체할 수 있도록 합니다. 또한 교체할 계산 노드가 CF VM을 호스팅하는 경우 특히 StarOS 구성을 백업하는 것이 좋습니다.

OSD-Compute 노드에서 호스팅되는 VM 식별

컴퓨팅 서버에서 호스팅되는 VM을 식별합니다. 다음과 같은 두 가지 가능성이 있습니다.

OSD-Compute 서버에는 다음과 같은 VM의 EM/UAS/Auto-Deploy/Auto-IT 조합이 포함되어 있습니다.

```
[stack@director ~]$ nova list --field name,host | grep osd-compute-0
```

```
| c6144778-9afd-4946-8453-78c817368f18 | AUTO-DEPLOY-VNF2-uas-0 | pod
| 2d051522-bce2-4809-8d63-0c0e17f251dc | AUTO-IT-VNF2-uas-0 | pod
| 507d67c2-1d00-4321-b9d1-da879af524f8 | VNF2-DEPLOYM_XXXX_0_c8d98f0f-d874-45d0-af75-88a2d6fa82ea | pod
| f5bd7b9c-476a-4679-83e5-303f0aae9309 | VNF2-UAS-uas-0 | pod
```

컴퓨팅 서버에는 VM의 CF/ESC/EM/UAS 조합이 포함되어 있습니다.

<#root>

```
[stack@director ~]$ nova list --field name,host | grep osd-compute-1
| 507d67c2-1d00-4321-b9d1-da879af524f8 |
```

VNF2-DEPLOYM_XXXX_0_c8d98f0f-d874-45d0-af75-88a2d6fa82ea

```
| pod1-compute-8.localdomain |
| f9c0763a-4a4f-4bbd-af51-bc7545774be2 |
```

VNF2-DEPLOYM_c1_0_df4be88d-b4bf-4456-945a-3812653ee229


```
| pod1-compute-8.localdomain |
| 75528898-ef4b-4d68-b05d-882014708694 |
```

VNF2-ESC-ESC-0

```
| pod1-compute-8.localdomain |
| f5bd7b9c-476a-4679-83e5-303f0aae9309 |
```

VNF2-UAS-uas-0

```
| pod1-compute-8.localdomain |
```

 참고: 여기에 표시된 출력에서 첫 번째 열은 UUID에 해당하고, 두 번째 열은 VM 이름이며, 세 번째 열은 VM이 있는 호스트 이름입니다. 이 출력의 매개변수는 후속 섹션에서 사용됩니다.

단일 OSD 서버를 제거하기 위해 Ceph에 가용 용량이 있는지 확인합니다.

<#root>

```
[root@pod1-osd-compute-1 ~]#
```

```
sudo ceph df
```

GLOBAL:

| SIZE | AVAIL | RAW USED | %RAW USED |
|--------|--------|----------|-----------|
| 13393G | 11804G | 1589G | 11.87 |

POOLS:


```
11 1.09000          osd.11                up 1.00000          1.00000
```

Ceph 프로세스는 OSD-Compute 서버에서 활성 상태입니다.

```
[root@pod1-osd-compute-1 ~]# 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 |
| ceph-osd.target | loaded | active | active | ceph target allowing to start/stop all ceph-osd |
| ceph-radosgw.target | loaded | active | active | ceph target allowing to start/stop all ceph-rad |
| ceph.target | loaded | active | active | ceph target allowing to start/stop all ceph*@.s |

각 Ceph 인스턴스를 비활성화 및 중지하고 OSD에서 각 인스턴스를 제거한 다음 디렉토리를 마운트 해제합니다. 각 Ceph 인스턴스에 대해 반복합니다.

```
[root@pod1-osd-compute-1 ~]# systemctl disable ceph-osd@11
```

```
[root@pod1-osd-compute-1 ~]# systemctl stop ceph-osd@11
```

```
[root@pod1-osd-compute-1 ~]# ceph osd out 11
```


marked out osd.11.

```
[root@pod1-osd-compute-1 ~]# ceph osd crush remove osd.11
```

```
removed item id 11 name 'osd.11' from crush map
```

```
[root@pod1-osd-compute-1 ~]# ceph auth del osd.11
```

```
updated
```

```
[root@pod1-osd-compute-1 ~]# ceph osd rm 11
```

```
removed osd.11
```

```
[root@pod1-osd-compute-1 ~]# umount /var/lib/ceph/osd/ceph-11
```

```
[root@pod1-osd-compute-1 ~]# rm -rf /var/lib/ceph/osd/ceph-11
```

또는

Clean.sh 스크립트를 사용하여 다음 작업을 수행할 수 있습니다.

```
[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
```

모든 OSD 프로세스가 마이그레이션/삭제된 후에는 노드를 오버클라우드에서 제거할 수 있습니다.

 참고: Ceph를 제거하면 VNF HD RAID가 Degraded(저하됨) 상태로 전환되지만 HD 디스크에 계속 액세스할 수 있어야 합니다.

정상 전원 끄기

케이스 1. OSD-Compute Node Hosts CF/ESC/EM/UAS

CF 카드를 대기 상태로 마이그레이션

StarOS VNF에 로그인하고 CF VM에 해당하는 카드를 식별합니다. OSD-Compute Node(OSD 컴퓨팅 노드)에서 호스팅되는 VM 식별 섹션에서 식별한 CF VM의 UUID를 사용하고 UUID에 해당하는 카드를 찾습니다.

```
[local]VNF2# show card hardware
Tuesday night 08 16:49:42 UTC 2018
<snip>
Card 2:
```

```

Card Type           : Control Function Virtual Card
CPU Packages        : 8 [#0, #1, #2, #3, #4, #5, #6, #7]
CPU Nodes           : 1
CPU Cores/Threads   : 8
Memory              : 16384M (qvmc-di-large)
UUID/Serial Number  : F9C0763A-4A4F-4BBD-AF51-BC7545774BE2
<snip>

```

카드 상태를 확인합니다.

```

[local]VNF2# show card table
Tuesday night 08 16:52:53 UTC 2018
Slot      Card Type                               Oper State   SPOF  Attach
-----
1: CFC    Control Function Virtual Card             Standby      -
2: CFC    Control Function Virtual Card             Active       No
3: FC     4-Port Service Function Virtual Card     Active       No
4: FC     4-Port Service Function Virtual Card     Active       No
5: FC     4-Port Service Function Virtual Card     Active       No
6: FC     4-Port Service Function Virtual Card     Active       No
7: FC     4-Port Service Function Virtual Card     Active       No
8: FC     4-Port Service Function Virtual Card     Active       No
9: FC     4-Port Service Function Virtual Card     Active       No
10: FC    4-Port Service Function Virtual Card     Standby      -

```

카드가 활성 상태인 경우 카드를 대기 상태로 전환합니다.

```
[local]VNF2# card migrate from 2 to 1
```

ESC에서 CF 및 EM VM 종료

VNF에 해당하는 ESC 노드에 로그인하고 VM의 상태를 확인합니다.

```
<#root>
```

```

[admin@VNF2-esc-esc-0 ~]$ cd /opt/cisco/esc/esc-confd/esc-cli
[admin@VNF2-esc-esc-0 esc-cli]$ ./esc_nc_cli get esc_datamodel | egrep --color "<state>|<vm_name>|<vm_i
<snip>
<state>SERVICE_ACTIVE_STATE</state>
      <vm_name>VNF2-DEPLOYM_c1_0_df4be88d-b4bf-4456-945a-3812653ee229</vm_name>
<state>
VM_ALIVE_STATE
</state>
      <vm_name>VNF2-DEPLOYM_c3_0_3e0db133-c13b-4e3d-ac14-
<state>VM_ALIVE_STATE</state>
<deployment_name>VNF2-DEPLOYMENT-em</deployment_name>
      <vm_id>507d67c2-1d00-4321-b9d1-da879af524f8</vm_id>

```

```
<vm_id>dc168a6a-4aeb-4e81-abd9-91d7568b5f7c</vm_id>
<vm_id>9ffec58b-4b9d-4072-b944-5413bf7fcf07</vm_id>
<state>SERVICE_ACTIVE_STATE</state>
  <vm_name>VNF2-DEPLOYM_XXXX_0_c8d98f0f-d874-45d0-af75-88a2d6fa82ea</vm_name>
</state>
```

VM_ALIVE_STATE

```
</state>
<snip>
```

VM 이름을 사용하여 CF 및 EM VM을 하나씩 중지합니다. 섹션에 명시된 VM 이름 OSD-Compute Node에서 호스팅되는 VM을 식별합니다.

```
[admin@VNF2-esc-esc-0 esc-cli]$ ./esc_nc_cli vm-action STOP VNF2-DEPLOYM_c1_0_df4be88d-b4bf-4456-945a-3
[admin@VNF2-esc-esc-0 esc-cli]$ ./esc_nc_cli vm-action STOP VNF2-DEPLOYM_XXXX_0_c8d98f0f-d874-45d0-af75
```

VM이 중지되면 VM은 SHUTOFF 상태로 전환해야 합니다.

```
<#root>
```

```
[admin@VNF2-esc-esc-0 ~]$ cd /opt/cisco/esc/esc-confd/esc-cli
[admin@VNF2-esc-esc-0 esc-cli]$ ./esc_nc_cli get esc_datamodel | egrep --color "<state>|<vm_name>|<vm_i
<snip>
<state>SERVICE_ACTIVE_STATE</state>
  <vm_name>
```

```
VNF2-DEPLOYM_c1_0_df4be88d-b4bf-4456-945a-3812653ee229
```

```
</vm_name>
```

```
<state>
```

VM_SHUTOFF_STATE

```
</state>
```

```
  <vm_name>VNF2-DEPLOYM_c3_0_3e0db133-c13b-4e3d-ac14-
  <state>VM_ALIVE_STATE</state>
<deployment_name>VNF2-DEPLOYMENT-em</deployment_name>
  <vm_id>507d67c2-1d00-4321-b9d1-da879af524f8</vm_id>
  <vm_id>dc168a6a-4aeb-4e81-abd9-91d7568b5f7c</vm_id>
  <vm_id>9ffec58b-4b9d-4072-b944-5413bf7fcf07</vm_id>
  <state>SERVICE_ACTIVE_STATE</state>
  <vm_name>
```

```
VNF2-DEPLOYM_XXXX_0_c8d98f0f-d874-45d0-af75-88a2d6fa82ea
```

```
</vm_name>
```

VM_SHUTOFF_STATE

<snip>

ESC를 대기 모드로 마이그레이션

컴퓨팅 노드에서 호스팅되는 ESC에 로그인하고 마스터 상태인지 확인합니다. 대답이 "예"인 경우 ESC를 대기 모드로 전환합니다.

```
[admin@VNF2-esc-esc-0 esc-cli]$ escadm status
0 ESC status=0 ESC Master Healthy
```

```
[admin@VNF2-esc-esc-0 ~]$ sudo service keepalived stop
Stopping keepalived: [ OK ]
```

```
[admin@VNF2-esc-esc-0 ~]$ escadm status
1 ESC status=0 In SWITCHING_TO_STOP state. Please check status after a while.
```

```
[admin@VNF2-esc-esc-0 ~]$ 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!
```

Nova Aggregate List(Nova 집계 목록)에서 OSD-Compute Node(OSD-컴퓨팅 노드) 제거

Nova 집계를 나열하고, Compute 서버가 호스팅하는 VNF를 기반으로 Compute 서버에 해당하는 집계를 식별합니다. 일반적으로 <VNFNAME>-EM-MGMT<X> 및 <VNFNAME>-CF-MGMT<X> 형식입니다.

<#root>

```
[stack@director ~]$ nova aggregate-list
+-----+-----+-----+
| Id | Name                | Availability Zone |
+-----+-----+-----+
| 29 |
```

POD1-AUTOIT

```
| mgmt |
| 57 | VNF1-SERVICE1 | - |
| 60 | VNF1-EM-MGMT1 | - |
| 63 | VNF1-CF-MGMT1 | - |
| 66 |
```

VNF2-CF-MGMT2

```
| - |
| 69 |
```

VNF2-EM-MGMT2

```
| - |
| 72 | VNF2-SERVICE2 | - |
| 75 | VNF3-CF-MGMT3 | - |
| 78 | VNF3-EM-MGMT3 | - |
| 81 | VNF3-SERVICE3 | - |
```

+-----+

이 경우 OSD-Compute 서버는 VNF2에 속합니다. 따라서 해당하는 집계는 VNF2-CF-MGMT2 및 VNF2-EM-MGMT2입니다.

식별된 집계에서 OSD-Compute 노드를 제거합니다.

<#root>

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

```
[stack@director ~]$ nova aggregate-remove-host VNF2-CF-MGMT2 pod1-osd-compute-0.localdomain
[stack@director ~]$ nova aggregate-remove-host VNF2-EM-MGMT2 pod1-osd-compute-0.localdomain
[stack@director ~]$ nova aggregate-remove-host POD1-AUTOIT pod1-osd-compute-0.localdomain
```

OSD-Compute 노드가 집계에서 제거되었는지 확인합니다. 이제 Host(호스트)가 aggregates(집계) 아래에 나열되지 않았는지 확인합니다.

<#root>

```
nova aggregate-show
```

```
[stack@director ~]$ nova aggregate-show VNF2-CF-MGMT2  
[stack@director ~]$ nova aggregate-show VNF2-EM-MGMT2  
[stack@director ~]$ nova aggregate-show POD1-AUTOIT
```

사례 2. OSD-Compute Node가 Auto-Deploy/Auto-IT/EM/UAS를 호스팅함

자동 구축의 CDB 백업

자동 배포 구성 cdb 데이터를 정기적으로 또는 활성화/비활성화할 때마다 백업하고 파일을 백업 서버에 저장합니다. 자동 배포는 중복되지 않으며 이 데이터가 손실될 경우 배포를 비활성화하기가 어렵습니다.

Auto-Deploy VM and backup confid cdb 디렉터리에 로그인합니다.

```
<#root>
```

```
ubuntu@auto-deploy-iso-2007-uas-0:~$
```

```
sudo -i
```

```
root@auto-deploy-iso-2007-uas-0:~#
```

```
service uas-confd stop
```

```
uas-confd stop/waiting
```

```
root@auto-deploy-iso-2007-uas-0:~#
```

```
cd /opt/cisco/usp/uas/confd-6.3.1/var/confd
```

```
root@auto-deploy-iso-2007-uas-0:/opt/cisco/usp/uas/confd-6.3.1/var/confd#
```

```
tar cvf autodeploy_cdb_backup.tar cdb/
```

```
cdb/
```

```
cdb/0.cdb
```

```
cdb/C.cdb
```

```
cdb/aaa_init.xml
```

cdb/A.cdb

```
root@auto-deploy-iso-2007-uas-0:~#
```

```
service uas-confd start
```

```
uas-confd start/running, process 13852
```

 참고: autodeploy_cdb_backup.tar을 백업 서버에 복사합니다.

자동 IT에서 system.cfg 백업

system.cfg 파일의 백업을 백업 서버로 가져옵니다.

```
Auto-it = 10.1.1.2  
Backup server = 10.2.2.2
```

```
[stack@director ~]$ ssh ubuntu@10.1.1.2  
ubuntu@10.1.1.2's password:  
Welcome to Ubuntu 14.04.3 LTS (GNU/Linux 3.13.0-76-generic x86_64)
```

```
* Documentation: https://help.ubuntu.com/
```

```
System information as of Wed Jun 13 16:21:34 UTC 2018
```


```
System load: 0.02          Processes:            87  
Usage of /: 15.1% of 78.71GB Users logged in:     0  
Memory usage: 13%         IP address for eth0: 172.16.182.4  
Swap usage: 0%
```

```
Graph this data and manage this system at:  
https://landscape.canonical.com/
```

```
Get cloud support with Ubuntu Advantage Cloud Guest:  
http://www.ubuntu.com/business/services/cloud
```

```
Cisco Ultra Services Platform (USP)  
Build Date: Wed Feb 14 12:58:22 EST 2018  
Description: UAS build assemble-uas#1891  
sha1: bf02ced
```

```
ubuntu@auto-it-vnf-uas-0:~$ scp -r /opt/cisco/usp/uploads/system.cfg root@10.2.2.2:/home/stack  
root@10.2.2.2's password:  
system.cfg  
ubuntu@auto-it-vnf-uas-0:~$
```


 참고: OSD-Compute-0에서 호스팅되는 EM/UAS의 정상 종료를 위해 수행할 절차는 두 경우 모두 동일합니다. 동일한 내용은 Case.1을 참조하십시오.

OSD-Compute Node 삭제

이 섹션에서 설명하는 단계는 컴퓨팅 노드에 호스팅된 VM에 관계없이 공통적으로 적용됩니다.

서비스 목록에서 OSD-Compute 노드 삭제

서비스 목록에서 컴퓨팅 서비스를 삭제합니다.

<#root>

```
[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      | 2018-05-08
```

```
openstack compute service delete
```

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

중성자 에이전트 삭제

이전 관련 중성자 에이전트를 삭제하고 컴퓨팅 서버에 대한 vswitch 에이전트를 엽니다.

<#root>

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

```
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 및 Ironic 데이터베이스에서 삭제

nova 목록 및 아이러니 데이터베이스에서 노드를 삭제하고 확인합니다.

```
<#root>
```

```
[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 | ct1p1
```

```
[stack@a101-pod1-ospd ~]$ nova delete c2cfa4d6-9c88-4ba0-9970-857d1a18d02c
```

```
nova show
```

```
| grep hypervisor
```

```
[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)
```

오버클라우드에서 삭제

표시된 내용과 함께 delete_node.sh라는 스크립트 파일을 생성합니다. 언급된 템플릿이 스택 배포에 사용되는 deploy.sh 스크립트에 사용된 템플릿과 동일한지 확인하십시오.

```
<#root>
```

```
delete_node.sh
```

```
openstack overcloud node delete
```

```
--templates -e /usr/share/openstack-tripleo-heat-templates/environments/puppet-pacemaker.yaml -e /usr/
```

```
[stack@director ~]$ source stackrc
[stack@director ~]$ /bin/sh delete_node.sh
+ openstack overcloud node delete --templates -e /usr/share/openstack-tripleo-heat-templates/environmen
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
```

COMPLETE 상태로 전환하려면 OpenStack 스택 작업을 기다립니다.

```
[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:... |

새 컴퓨팅 노드 설치

- 새 UCS C240 M4 서버를 설치하는 단계 및 초기 설정 단계는 다음에서 참조할 수 있습니다.

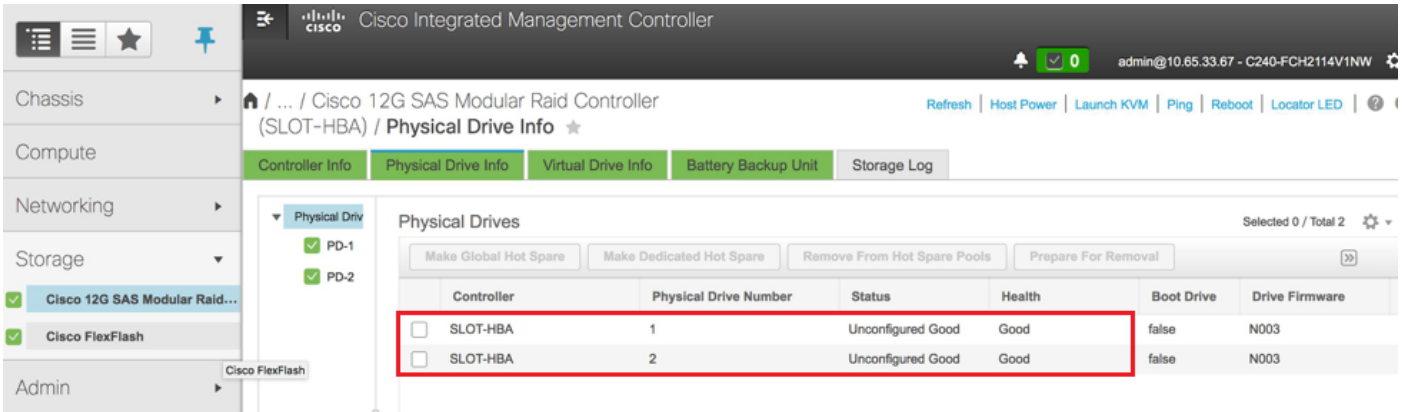
[Cisco UCS C240 M4 서버 설치 및 서비스 가이드](#)

- 서버 설치 후 하드 디스크를 이전 서버로 각 슬롯에 삽입합니다
- CIMC IP를 사용하여 서버에 로그인합니다


- 펌웨어가 이전에 사용한 권장 버전에 따라 다르면 BIOS 업그레이드를 수행합니다. BIOS 업그레이드 단계는 다음과 같습니다.

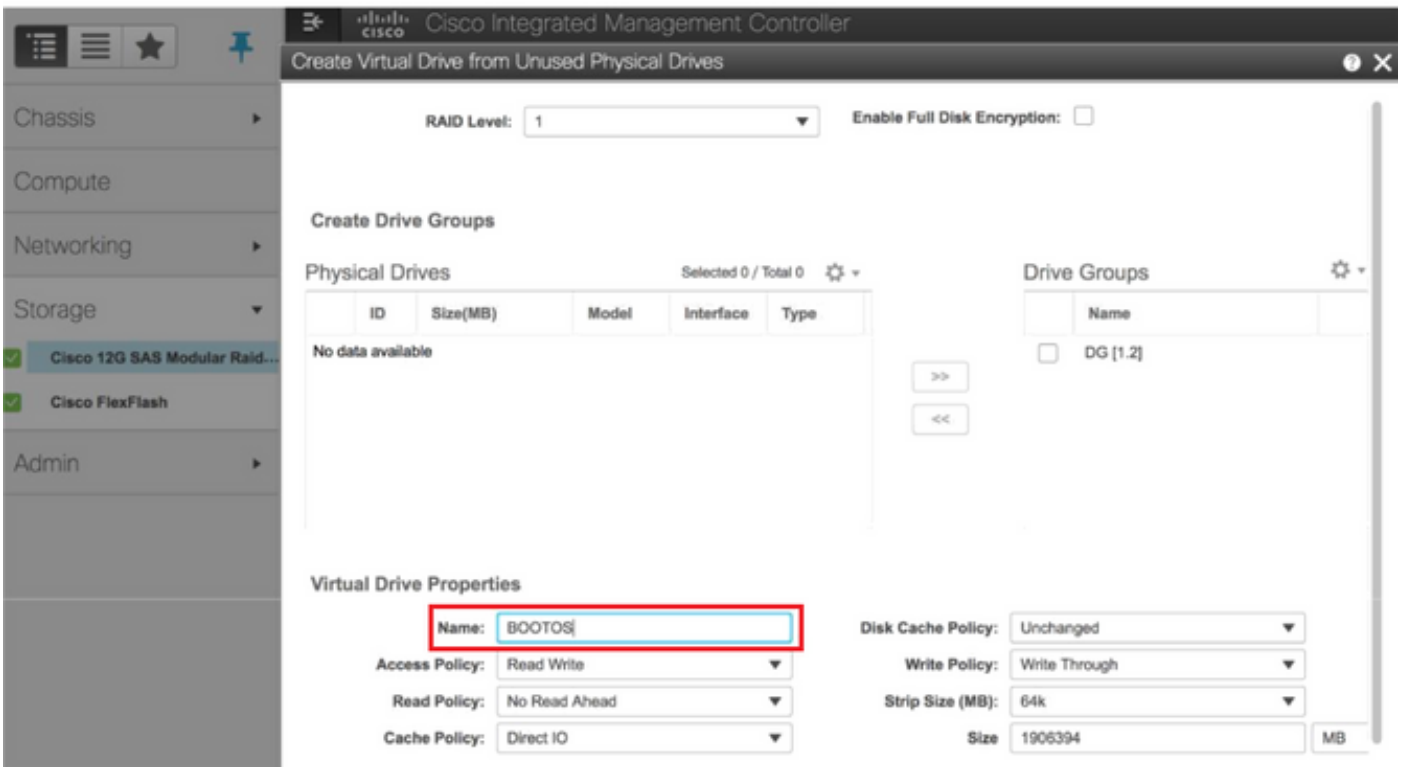
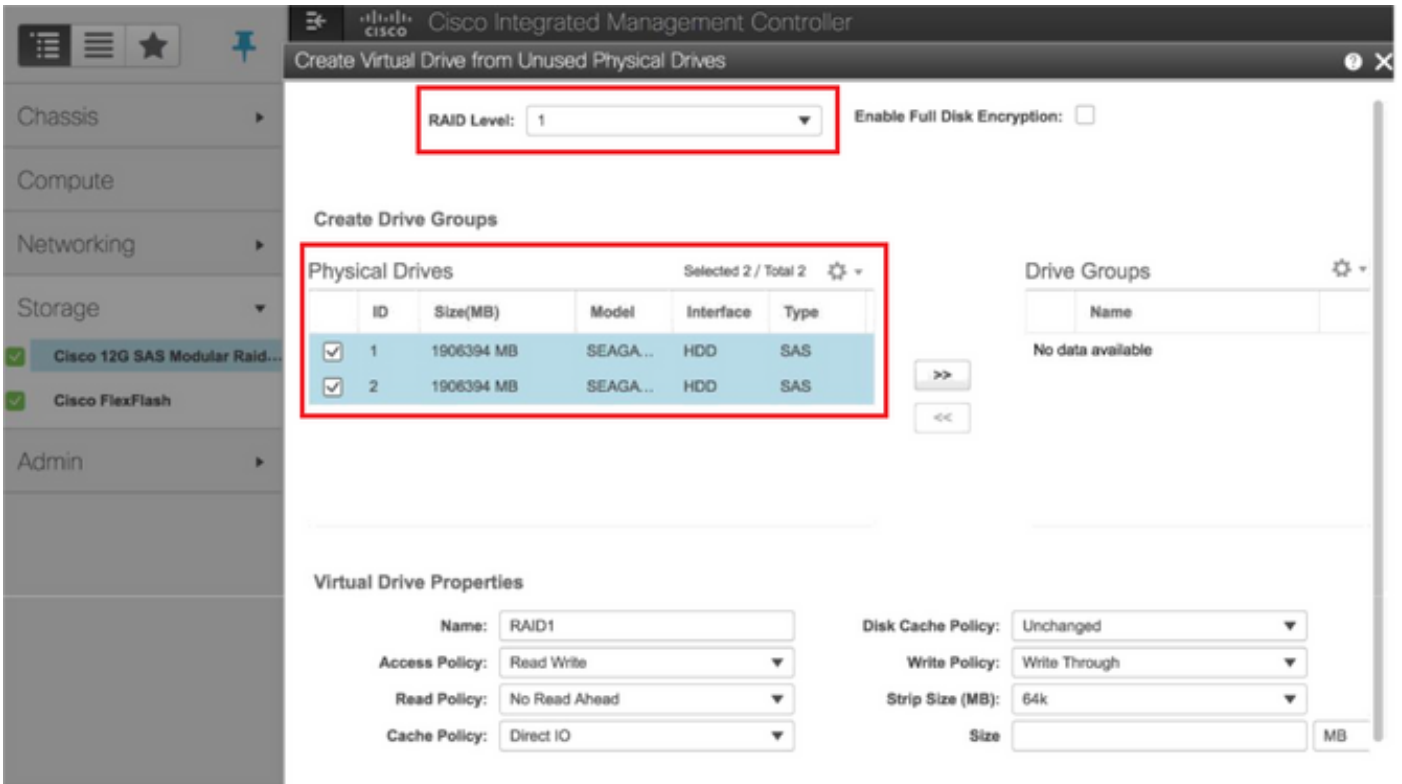
[Cisco UCS C-Series 랙 마운트 서버 BIOS 업그레이드 가이드](#)

- 물리적 드라이브의 상태를 확인합니다. 그것은 분명히 분명하다
- RAID 레벨 1의 물리적 드라이브에서 가상 드라이브 생성

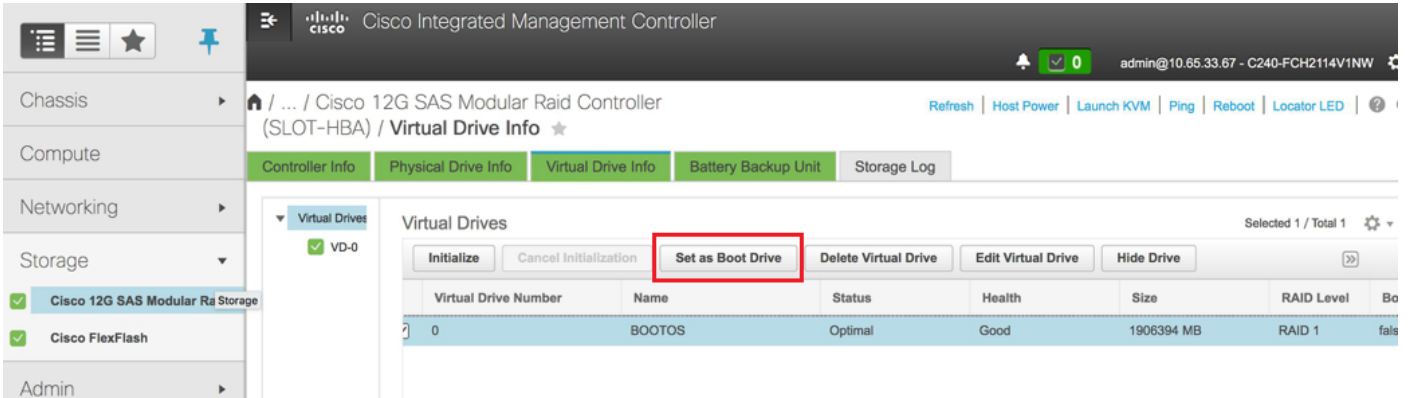


스토리지 > Cisco 12G SAS 모듈형 Raid 컨트롤러(SLOT-HBA) > 물리적 드라이브 정보

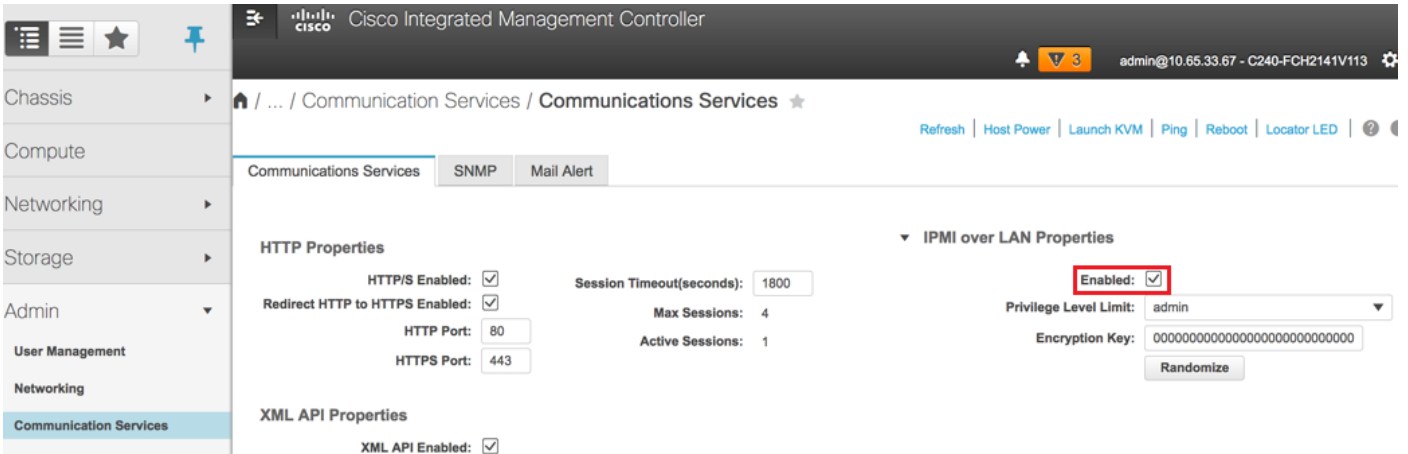
 참고: 이 이미지는 예시를 위한 것이며, 실제 OSD-Compute CIMC에서 가상 드라이브가 생성되지 않으므로 구성되지 않음(Unconfigured) 상태의 슬롯에 있는 7개의 물리적 드라이브 (1,2,3,7,8,9,10)를 볼 수 있습니다.



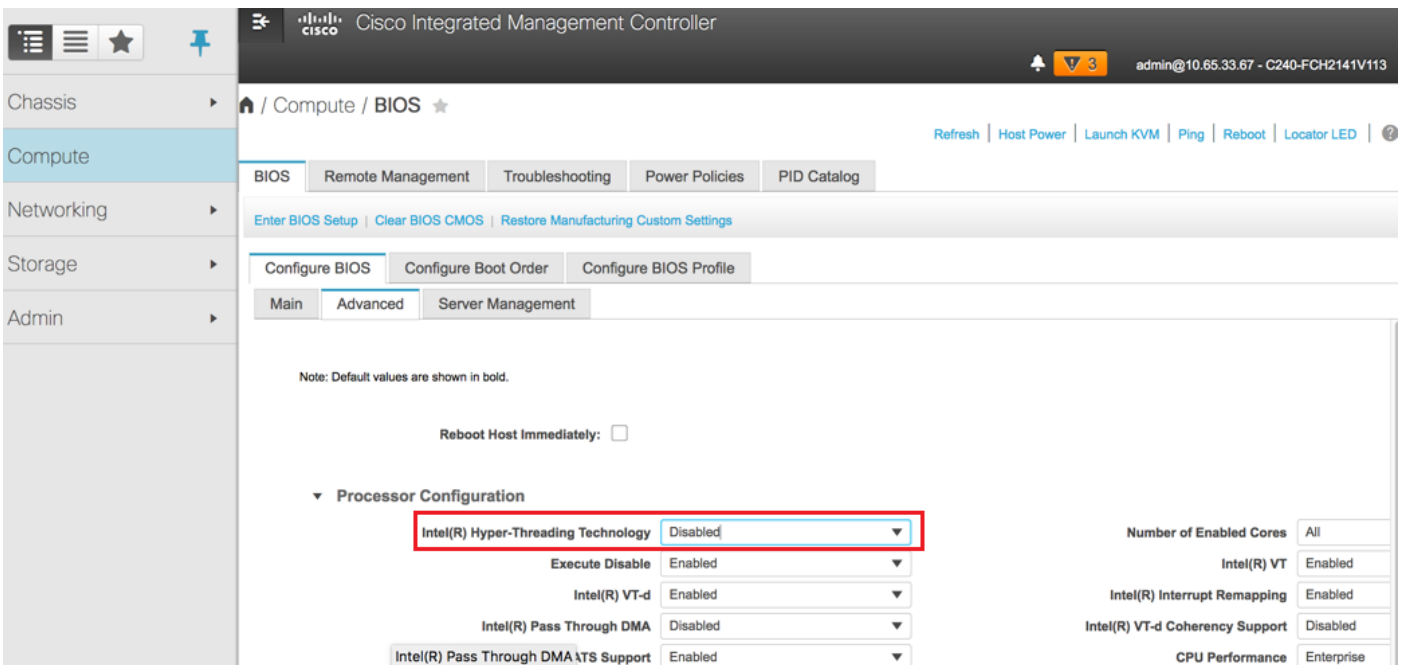
스토리지 > Cisco 12G SAS 모듈형 Raid 컨트롤러(SLOT-HBA) > 컨트롤러 정보 > 사용되지 않은 물리적 드라이브에서 가상 드라이브 생성



VD를 선택하고 "Set as Boot Drive(부팅 드라이브로 설정)"를 구성합니다.



IPMI over LAN 활성화: Admin(관리) > Communication Services(통신 서비스) > Communication Services(통신 서비스)



하이퍼스레딩을 비활성화합니다. Compute(컴퓨팅) > BIOS > Configure BIOS > Advanced(고급) > Processor Configuration(프로세서 컨피그레이션)

- 물리적 드라이브 1 및 2로 생성된 BOOTOS VD와 마찬가지로

JOURNAL(저널) > 물리적 드라이브 번호 3에서

OSD1 > 물리적 드라이브 번호 7에서

OSD2 > 8번 물리적 드라이브에서

OSD3 > 물리적 드라이브 번호 9에서

OSD4 > 물리적 드라이브 번호 10에서

- 마지막으로, 물리적 드라이브 및 가상 드라이브는 이미지에 표시된 것과 유사해야 합니다.

| Virtual Drive Number | Name | Status | Health | Size | RAID Level | Boot Drive |
|---------------------------------------|---------|---------|--------|------------|------------|------------|
| <input checked="" type="checkbox"/> 0 | BOOTOS | Optimal | Good | 285148 MB | RAID 1 | true |
| <input type="checkbox"/> 1 | JOURNAL | Optimal | Good | 456809 MB | RAID 0 | false |
| <input type="checkbox"/> 2 | OSD1 | Optimal | Good | 1143455 MB | RAID 0 | false |
| <input type="checkbox"/> 3 | OSD2 | Optimal | Good | 1143455 MB | RAID 0 | false |
| <input type="checkbox"/> 4 | OSD3 | Optimal | Good | 1143455 MB | RAID 0 | false |
| <input type="checkbox"/> 5 | OSD4 | Optimal | Good | 1143455 MB | RAID 0 | false |

| Controller | Physical Drive Number | Status | Health | Boot Drive | Drive Firmware | Coerced Size | Model | Type |
|-----------------------------------|-----------------------|--------|--------|------------|----------------|--------------|---------|------|
| <input type="checkbox"/> SLOT-HBA | 1 | Online | Good | false | 5704 | 1143455 MB | TOSHIBA | HDD |
| <input type="checkbox"/> SLOT-HBA | 2 | Online | Good | false | 5704 | 1143455 MB | TOSHIBA | HDD |
| <input type="checkbox"/> SLOT-HBA | 3 | Online | Good | false | CS01 | 456809 MB | ATA | SSD |
| <input type="checkbox"/> SLOT-HBA | 7 | Online | Good | false | N004 | 1143455 MB | SEAGATE | HDD |
| <input type="checkbox"/> SLOT-HBA | 8 | Online | Good | false | 5704 | 1143455 MB | TOSHIBA | HDD |
| <input type="checkbox"/> SLOT-HBA | 9 | Online | Good | false | N004 | 1143455 MB | SEAGATE | HDD |
| <input type="checkbox"/> SLOT-HBA | 10 | Online | Good | false | N004 | 1143455 MB | SEAGATE | HDD |

가상 드라이브
물리적 드라이브

참고: 여기에 표시된 이미지 및 이 섹션에서 설명한 컨피그레이션 단계는 펌웨어 버전 3.0(3e)을 참조하며, 다른 버전에서 작업하는 경우 약간의 차이가 있을 수 있습니다.


오버클라우드에 새 OSD-Compute 노드 추가

이 섹션에서 설명하는 단계는 컴퓨팅 노드에 의해 호스팅되는 VM에 관계없이 일반적인 단계입니다

다른 인덱스로 Compute 서버를 추가합니다.

추가할 새 컴퓨팅 서버의 세부 정보만 포함된 add_node.json 파일을 만듭니다. 새 OSD-Compute 서버의 인덱스 번호가 이전에 사용되지 않았는지 확인합니다. 일반적으로 다음으로 높은 컴퓨팅 값을 증가시킵니다.

예: 가장 높은 선행은 OSD-Compute-0으로 2-vnf 시스템의 경우 OSD-Compute-3을 생성했습니다.

 참고: json 형식에 유의하십시오.

```
[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"
    }
  ]
}
```

json 파일 가져오기:

<#root>

```
[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.
```

이전 단계에서 기록한 UUID를 사용하여 노드 자체 검사를 실행합니다.

<#root>

```
[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 | ma

<#root>

```
[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 | av

OsdComputeIPs 아래의 custom-templates/layout.yml에 IP 주소를 추가합니다. 이 경우 OSD-Compute-0을 교체할 때 각 유형에 대한 목록의 끝에 해당 주소를 추가합니다.

<#root>

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

오버클라우드 스택에 새 컴퓨팅 노드를 추가하기 위해 이전에 스택 구축에 사용되었던 `deploy.sh` 스크립트를 실행합니다.

```
[stack@director ~]$ ./deploy.sh
++ openstack overcloud deploy --templates -r /home/stack/custom-templates/custom-roles.yaml -e /usr/sha...
...
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
```

OpenStack 스택 상태가 COMPLETE가 될 때까지 기다립니다.

```
[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 | 2017-11-02T21:30:06Z |

새 OSD-Compute 노드가 활성 상태인지 확인합니다.

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

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

새 OSD-Compute 서버에 로그인하고 Ceph 프로세스를 선택합니다. 초기에는 Ceph가 복구될 때 상태가 HEALTH_WARN이 됩니다.

<#root>

```
[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
```

그러나 짧은 기간(20분) 후 Ceph는 HEALTH_OK 상태로 돌아갑니다.

```
<#root>
```

```
[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 | osd.1 | up | 1.00000 | 1.00000 |
| 4 | 1.09000 | osd | osd.4 | up | 1.00000 | 1.00000 |
| 7 | 1.09000 | osd | osd.7 | up | 1.00000 | 1.00000 |
| 10 | 1.09000 | osd | 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 |

사후 서버 교체 설정

오버클라우드에 서버를 추가한 후 아래 링크를 참조하여 이전 서버에 있던 설정을 적용하십시오.

VM 복원

케이스 1. CF, ESC, EM 및 UAS를 호스팅하는 OSD-컴퓨팅 노드

Nova Aggregate List에 추가

OSD-Compute 노드를 aggregate-hosts에 추가하고 호스트가 추가되었는지 확인합니다. 이 경우 OSD-Compute 노드는 CF 및 EM 호스트 집계에 모두 추가되어야 합니다.

<#root>

```
nova aggregate-add-host
```

```
[stack@director ~]$ nova aggregate-add-host VNF2-CF-MGMT2 pod1-osd-compute-3.localdomain
[stack@director ~]$ nova aggregate-add-host VNF2-EM-MGMT2 pod1-osd-compute-3.localdomain
```

```
[stack@directr ~]$ nova aggregate-add-host POD1-AUTOIT pod1-osd-compute-3.localdomain
```

```
nova aggregate-show
```

```
[stack@director ~]$ nova aggregate-show VNF2-CF-MGMT2
```

```
[stack@director ~]$ nova aggregate-show VNF2-EM-MGMT2
```

```
[stack@director ~]$ nova aggregate-show POD1-AUTOITT
```

UAS VM 복구

nova 목록에서 UAS VM의 상태를 확인하고 삭제합니다.

```
[stack@director ~]$ nova list | grep VNF2-UAS-uas-0
```

```
| 307a704c-a17c-4cdc-8e7a-3d6e7e4332fa | VNF2-UAS-uas-0
```

```
[stack@director ~]$ nova delete VNF2-UAS-uas-0
```

```
Request to delete server VNF2-UAS-uas-0 has been accepted.
```

autovnf-uas VM을 복구하려면 uas-check 스크립트를 실행하여 상태를 확인합니다. 오류를 보고해야 합니다. 그런 다음 `--fix` 옵션을 사용하여 다시 실행하여 누락된 UAS VM을 다시 생성합니다.

```
[stack@director ~]$ cd /opt/cisco/usp/uas-installer/scripts/
```

```
[stack@director scripts]$ ./uas-check.py auto-vnf VNF2-UAS
```

```
2017-12-08 12:38:05,446 - INFO: Check of AutoVNF cluster started
```

```
2017-12-08 12:38:07,925 - INFO: Instance 'vnf1-UAS-uas-0' status is 'ERROR'
```

```
2017-12-08 12:38:07,925 - INFO: Check completed, AutoVNF cluster has recoverable errors
```

```
[stack@director scripts]$ ./uas-check.py auto-vnf VNF2-UAS --fix
```

```
2017-11-22 14:01:07,215 - INFO: Check of AutoVNF cluster started
```

```
2017-11-22 14:01:09,575 - INFO: Instance VNF2-UAS-uas-0' status is 'ERROR'
```

```
2017-11-22 14:01:09,575 - INFO: Check completed, AutoVNF cluster has recoverable errors
```

```
2017-11-22 14:01:09,778 - INFO: Removing instance VNF2-UAS-uas-0'
```

```
2017-11-22 14:01:13,568 - INFO: Removed instance VNF2-UAS-uas-0'
```

```
2017-11-22 14:01:13,568 - INFO: Creating instance VNF2-UAS-uas-0' and attaching volume 'VNF2-UAS-uas-vo
```

```
2017-11-22 14:01:49,525 - INFO: Created instance 'VNF2-UAS-uas-0'
```

autovnf-uas에 로그인합니다. 몇 분 정도 기다린 후 UAS가 정상 상태로 돌아와야 합니다.

```
VNF2-autovnf-uas-0#show uas
```

```
uas version 1.0.1-1
```


```
uas state ha-active
```

```
uas ha-vip 172.17.181.101
```

```

INSTANCE IP    STATE  ROLE
-----
172.17.180.6  alive  CONFD-SLAVE
172.17.180.7  alive  CONFD-MASTER
172.17.180.9  alive  NA

```

 참고: uas-check.py —수정이 실패할 경우 이 파일을 복사하고 다시 실행해야 할 수 있습니다.

```

[stack@director ~]$ mkdir -p /opt/cisco/usp/apps/auto-it/common/uas-deploy/
[stack@director ~]$ cp /opt/cisco/usp/uas-installer/common/uas-deploy/userdata-uas.txt /opt/cisco/usp/a

```

ESC VM 복구

nova 목록에서 ESC VM의 상태를 확인하고 삭제합니다.

```

stack@director scripts]$ nova list |grep ESC-1
| c566efbf-1274-4588-a2d8-0682e17b0d41 | VNF2-ESC-ESC-1
[stack@director scripts]$ nova delete VNF2-ESC-ESC-1
Request to delete server VNF2-ESC-ESC-1 has been accepted.

```

AutoVNF-UAS에서 ESC 구축 트랜잭션을 찾고 트랜잭션 로그에서 boot_vm.py 명령줄을 찾아 ESC 인스턴스를 생성합니다.

<#root>

```

ubuntu@VNF2-uas-uas-0:~$ sudo -i
root@VNF2-uas-uas-0:~# confd_cli -u admin -C
Welcome to the ConfD CLI
admin connected from 127.0.0.1 using console on VNF2-uas-uas-0
VNF2-uas-uas-0#show transaction
TX ID                                TX TYPE          DEPLOYMENT ID    TIMESTAMP
-----
35eefc4a-d4a9-11e7-bb72-fa163ef8df2b  vnf-deployment  VNF2-DEPLOYMENT  2017-11-29T02:01:27.750692-00:00
73d9c540-d4a8-11e7-bb72-fa163ef8df2b  vnf-deployment  VNF2-ESC         2017-11-29T01:56:02.133663-00:00

VNF2-uas-uas-0#show logs 73d9c540-d4a8-11e7-bb72-fa163ef8df2b | display xml
<config xmlns="http://tail-f.com/ns/config/1.0">
  <logs xmlns="http://www.cisco.com/usp/nfv/usp-autovnf-oper">
    <tx-id>73d9c540-d4a8-11e7-bb72-fa163ef8df2b</tx-id>
    <log>2017-11-29 01:56:02,142 - VNF Deployment RPC triggered for deployment: VNF2-ESC, deactivate:
2017-11-29 01:56:02,179 - Notify deployment
    ..
2017-11-29 01:57:30,385 - Creating VNF 'VNF2-ESC-ESC-1' with [python //opt/cisco/vnf-staging/

```

```
bootvm.py VNF2-ESC-ESC-1 --flavor VNF2-ESC-ESC-flavor --image 3fe6b197-961b-4651-af22-dfd910436689 --net
```

boot_vm.py 행을 셸 스크립트 파일(esc.sh)에 저장하고 모든 사용자 이름 ***** 및 비밀번호 ***** 행을 올바른 정보(일반적으로 core/<PASSWORD>)로 업데이트합니다. -encrypt_key 옵션도 제거해야 합니다. user_pass 및 user_conpd_pass의 경우 username 형식을 사용해야 합니다. password(예: admin:<PASSWORD>).

bootvm.py가 running-config에서 유지되도록 하기 위한 URL을 찾고 bootvm.py 파일을 autovnf-uas VM으로 가져옵니다. 이 경우 10.1.2.3은 Auto-IT VM의 IP입니다.

```
root@VNF2-uas-uas-0:~# confd_cli -u admin -C
Welcome to the ConfD CLI
admin connected from 127.0.0.1 using console on VNF2-uas-uas-0
VNF2-uas-uas-0#show running-config autovnf-vnfm:vnfm
...
configs bootvm
  value http:// 10.1.2.3:80/bundles/5.1.7-2007/vnfm-bundle/bootvm-2_3_2_155.py
!
```

```
root@VNF2-uas-uas-0:~# wget http://10.1.2.3:80/bundles/5.1.7-2007/vnfm-bundle/bootvm-2_3_2_155.py
--2017-12-01 20:25:52-- http://10.1.2.3/bundles/5.1.7-2007/vnfm-bundle/bootvm-2_3_2_155.py
Connecting to 10.1.2.3:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 127771 (125K) [text/x-python]
Saving to: 'bootvm-2_3_2_155.py'
100%[=====] 127,771
2017-12-01 20:25:52 (173 MB/s) - 'bootvm-2_3_2_155.py' saved [127771/127771]
```

/tmp/esc_params.cfg 파일을 만듭니다.

```
root@VNF2-uas-uas-0:~# echo "openstack.endpoint=publicURL" > /tmp/esc_params.cfg
```

UAS 노드에서 ESC를 구축하려면 셸 스크립트를 실행합니다.

```
root@VNF2-uas-uas-0:~# /bin/sh esc.sh
+ python ./bootvm.py VNF2-ESC-ESC-1 --flavor VNF2-ESC-ESC-flavor --image 3fe6b197-961b-4651-af22-dfd910
--net VNF2-UAS-uas-management --gateway_ip 172.168.10.1 --net VNF2-UAS-uas-orchestration --os_auth_url
http://10.1.2.5:5000/v2.0 --os_tenant_name core --os_username core --os_password <PASSWORD> --bs_os_auth
http://10.1.2.5:5000/v2.0 --bs_os_tenant_name core --bs_os_username core --bs_os_password <PASSWORD>
--esc_ui_startup false --esc_params_file /tmp/esc_params.cfg --user_pass admin:<PASSWORD> --user_confd_
admin:<PASSWORD> --kad_vif eth0 --kad_vip 172.168.10.7 --ipaddr 172.168.10.6 dhcp --ha_node_list 172.16
172.168.10.6 --file root:0755:/opt/cisco/esc/esc-scripts/esc_volume_em_staging.sh:/opt/cisco/usp/uas/au
--file root:0755:/opt/cisco/esc/esc-scripts/esc_vpc_chassis_id.py:/opt/cisco/usp/uas/autovnf/vnfms/esc-
```



```
--file root:0755:/opt/cisco/esc/esc-scripts/esc-vpc-di-internal-keys.sh:/opt/cisco/usp/uas/autovnf/vnfm
```

새 ESC에 로그인하고 백업 상태를 확인합니다.

```
ubuntu@VNF2-uas-uas-0:~$ ssh admin@172.168.11.14
```

```
...  
#####  
#   ESC on VNF2-esc-esc-1.novalocal is in BACKUP state.  
#####
```

```
[admin@VNF2-esc-esc-1 ~]$ escadm status  
0 ESC status=0 ESC Backup Healthy
```

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

ESC에서 CF 및 EM VM 복구

nova 목록에서 CF 및 EM VM의 상태를 확인합니다. 오류 상태여야 합니다.

```
[stack@director ~]$ source corerc  
[stack@director ~]$ nova list --field name,host,status |grep -i err  
| 507d67c2-1d00-4321-b9d1-da879af524f8 | VNF2-DEPLOYM_XXXX_0_c8d98f0f-d874-45d0-af75-88a2d6fa82ea | Non  
| f9c0763a-4a4f-4bbd-af51-bc7545774be2 | VNF2-DEPLOYM_c1_0_df4be88d-b4bf-4456-945a-3812653ee229 | No
```

ESC 마스터에 로그인하여 영향을 받는 각 EM 및 CF VM에 대해 recovery-vm-action을 실행합니다. 인내심을 가져라. ESC는 복구 작업을 예약하며 몇 분 동안 발생하지 않을 수 있습니다. yangesc.log를 모니터링합니다.

<#root>

```
sudo /opt/cisco/esc/esc-confd/esc-cli/esc_nc_cli recovery-vm-action DO
```

```
[admin@VNF2-esc-esc-0 ~]$ sudo /opt/cisco/esc/esc-confd/esc-cli/esc_nc_cli recovery-vm-action DO VNF2-D  
[sudo] password for admin:
```

Recovery VM Action

```
/opt/cisco/esc/confd/bin/netconf-console --port=830 --host=127.0.0.1 --user=admin --privKeyFile=/root/.
<?xml version="1.0" encoding="UTF-8"?>
<rpc-reply xmlns="urn:iETF:params:xml:ns:netconf:base:1.0" message-id="1">
  <ok/>
</rpc-reply>
```

```
[admin@VNF2-esc-esc-0 ~]$ tail -f /var/log/esc/yangesc.log
...
14:59:50,112 07-Nov-2017 WARN Type: VM_RECOVERY_COMPLETE
14:59:50,112 07-Nov-2017 WARN Status: SUCCESS
14:59:50,112 07-Nov-2017 WARN Status Code: 200
14:59:50,112 07-Nov-2017 WARN Status Msg: Recovery: Successfully recovered VM [VNF2-DEPLOYMENT-_VNF2-D
```

새 EM에 로그인하고 EM 상태가 up인지 확인합니다.

```
ubuntu@VNF2vnfddeploymentem-1:~$ /opt/cisco/ncs/current/bin/ncs_cli -u admin -C
admin connected from 172.17.180.6 using ssh on VNF2vnfddeploymentem-1
admin@scm# show ems
EM          VNFM
ID  SLA  SCM  PROXY
-----
2   up   up   up
3   up   up   up
```

StarOS VNF에 로그인하여 CF 카드가 대기 상태인지 확인합니다.

사례 2. 자동 IT, 자동 배포, EM 및 UAS를 호스팅하는 OSD-Compute Node

자동 구축 VM 복구

OSPD에서 자동 구축 VM이 영향을 받았지만 여전히 ACTIVE/Running으로 표시되면 먼저 삭제해야 합니다. 자동 구축이 영향을 받지 않은 경우 Recovery of Auto-it VM(자동 VM 복구)으로 건너뛰니다.

```
<#root>
```

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

```
nova list |grep auto-deploy
```

```
| 9b55270a-2dcd-4ac1-aba3-bf041733a0c9 | auto-deploy-ISO-2007-uas-0
```

```
cd /opt/cisco/usp/uas-installer/scripts
```

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

```
./auto-deploy-booting.sh --floating-ip 10.1.2.7 --delete
```

자동 구축이 삭제되면 동일한 floatingip 주소로 다시 생성합니다.

```
<#root>
```

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

```
cd /opt/cisco/usp/uas-installer/scripts
```

```
[stack@director scripts]$
```

```
./auto-deploy-booting.sh --floating-ip 10.1.2.7
```

```
2017-11-17 07:05:03,038 - INFO: Creating AutoDeploy deployment (1 instance(s)) on 'http://10.84.123.4:5
```

```
2017-11-17 07:05:03,039 - INFO: Loading image 'auto-deploy-ISO-5-1-7-2007-usp-uas-1.0.1-1504.qcow2' from
```

```
2017-11-17 07:05:14,603 - INFO: Loaded image 'auto-deploy-ISO-5-1-7-2007-usp-uas-1.0.1-1504.qcow2'
```

```
2017-11-17 07:05:15,787 - INFO: Assigned floating IP '10.1.2.7' to IP '172.16.181.7'
```

```
2017-11-17 07:05:15,788 - INFO: Creating instance 'auto-deploy-ISO-5-1-7-2007-uas-0'
```

```
2017-11-17 07:05:42,759 - INFO: Created instance 'auto-deploy-ISO-5-1-7-2007-uas-0'
```

```
2017-11-17 07:05:42,759 - INFO: Request completed, floating IP: 10.1.2.7
```

백업 서버에서 Autodeploy.cfg 파일, ISO 및 conpd_backup tar 파일을 복사하여 VM을 자동 배포하고 백업 tar 파일에서 conpd cdb 파일을 복원합니다.

```
<#root>
```

```
ubuntu@auto-deploy-iso-2007-uas-0:~#
```

```
sudo -i
```

```
ubuntu@auto-deploy-iso-2007-uas-0:#
```

```
service uas-confd stop
```

```
uas-confd stop/waiting
```

```
root@auto-deploy-iso-2007-uas-0:#
```

```
cd /opt/cisco/usp/uas/confd-6.3.1/var/confd
```

```
root@auto-deploy-iso-2007-uas-0:/opt/cisco/usp/uas/confd-6.3.1/var/confd#
```

```
tar xvf /home/ubuntu/ad_cdb_backup.tar
```

```
cdb/
```

```
cdb/0.cdb
```

```
cdb/C.cdb
```

```
cdb/aaa_init.xml
```

```
cdb/A.cdb
```

```
root@auto-deploy-iso-2007-uas-0~#
```

```
service uas-confd start
```

```
uas-confd start/running, process 2036
```

이전 트랜잭션을 확인하여 confd가 제대로 로드되었는지 확인합니다. autodeploy.cfg를 새 OSD-Compute 이름으로 업데이트합니다. 섹션 - 마지막 단계를 참조하십시오. 자동 배포 구성 업데이트:

```
<#root>
```

```
root@auto-deploy-iso-2007-uas-0:~#
```

```
confd_cli -u admin -C
```

```
Welcome to the ConfD CLI
```

```
admin connected from 127.0.0.1 using console on auto-deploy-iso-2007-uas-0
```

```
auto-deploy-iso-2007-uas-0#
```

```
show transaction
```

| TX ID | TX TYPE | SERVICE ID | DEPLOYMENT DATE AND TIME | STATUS | SITE ID |
|---------------|--------------------|---------------|-------------------------------|--------------------|------------|
| 1512571978613 | service-deployment | tb5bxb | 2017-12-06T14:52:59.412+00:00 | deployment-success | |

```
auto-deploy-iso-2007-uas-0# exit
```

Auto-IT VM 복구

OSPD에서 auto-it VM이 영향을 받았지만 여전히 ACTIVE/Running으로 표시되는 경우 삭제해야 합니다. auto-it가 영향을 받지 않은 경우 다음 VM으로 건너뛵니다.

```
<#root>
```

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

```
nova list |grep auto-it
```

```
| 580faf80-1d8c-463b-9354-781ea0c0b352 | auto-it-vnf-ISO-2007-uas-0
```

```
cd /opt/cisco/usp/uas-installer/scripts
```

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

```
./
```

```
auto-it-vnf-staging.sh --floating-ip 10.1.2.8 --delete
```

auto-it-vnf 스테이징 스크립트를 실행하고 auto-it를 다시 생성합니다.

```
<#root>
```

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

```
cd /opt/cisco/usp/uas-installer/scripts
```

```
[stack@director scripts]$
```

```
./auto-it-vnf-staging.sh --floating-ip 10.1.2.8
```

```
2017-11-16 12:54:31,381 - INFO: Creating StagingServer deployment (1 instance(s)) on 'http://10.84.123.1'
```

```
2017-11-16 12:54:31,382 - INFO: Loading image 'auto-it-vnf-ISO-5-1-7-2007-usp-uas-1.0.1-1504.qcow2' from 'http://10.84.123.1'
```

```
2017-11-16 12:54:51,961 - INFO: Loaded image 'auto-it-vnf-ISO-5-1-7-2007-usp-uas-1.0.1-1504.qcow2'
```

```
2017-11-16 12:54:53,217 - INFO: Assigned floating IP '10.1.2.8' to IP '172.16.181.9'
```

```
2017-11-16 12:54:53,217 - INFO: Creating instance 'auto-it-vnf-ISO-5-1-7-2007-uas-0'
```

2017-11-16 12:55:20,929 - INFO: Created instance 'auto-it-vnf-ISO-5-1-7-2007-uas-0'

2017-11-16 12:55:20,930 - INFO: Request completed, floating IP: 10.1.2.8

ISO 이미지를 다시 로드합니다. 이 경우 auto-it IP 주소는 10.1.2.8입니다. 이 작업은 로드하는 데 몇 분 정도 걸립니다.

<#root>

```
[stack@director ~]$ cd images/5_1_7-2007/isos
```

```
[stack@director isos]$
```

```
curl -F file=@usp-5_1_7-2007.iso http://10.1.2.8:5001/isos
```

```
{  
  "iso-id": "5.1.7-2007"  
}
```

to check the ISO image:

```
[stack@director isos]$ curl http://10.1.2.8:5001/isos
```

```
{  
  "isos": [  
    {  
      "iso-id": "5.1.7-2007"  
    }  
  ]  
}
```

OSPD Auto-Deploy 디렉토리에서 자동 VM으로 VNF system.cfg 파일을 복사합니다.


```
[stack@director autodeploy]$ scp system-vnf* ubuntu@10.1.2.8:.
```

ubuntu@10.1.2.8's password:

system-vnf1.cfg

system-vnf2.cfg

```
ubuntu@auto-it-vnf-iso-2007-uas-0:~$ pwd
/home/ubuntu
ubuntu@auto-it-vnf-iso-2007-uas-0:~$ ls
system-vnf1.cfg  system-vnf2.cfg
```

 참고: EM 및 UAS VM의 복구 절차는 두 경우 모두 동일합니다. 동일한 내용은 Case.1 섹션을 참조하십시오.

ESC 복구 실패 처리

예기치 않은 상태로 인해 ESC가 VM을 시작하지 못하는 경우 마스터 ESC를 재부팅하여 ESC 전환을 수행하는 것이 좋습니다. ESC 전환에는 약 1분이 소요됩니다. 새 마스터 ESC에서 스크립트 health.sh를 실행하여 상태가 up인지 확인합니다. VM을 시작하고 VM 상태를 수정하려면 마스터 ESC를 누릅니다. 이 복구 작업을 완료하는 데 최대 5분이 소요됩니다.

/var/log/esc/yangesc.log 및 /var/log/esc/escmanager.log를 모니터링할 수 있습니다. 5~7분 후에 VM이 복구되지 않으면 사용자는 직접 손상된 VM을 수동으로 복구해야 합니다.

자동 구축 컨피그레이션 업데이트

AutoDeploy VM에서 auto-deploy.cfg를 편집하고 기존 OSD-Compute 서버를 새 서버로 교체합니다. 그런 다음 conpd_cli에서 replace를 로드합니다. 이 단계는 나중에 성공적인 구축을 비활성화하는 데 필요합니다.

```
root@auto-deploy-iso-2007-uas-0:/home/ubuntu# confd_cli -u admin -C
Welcome to the Confd CLI
admin connected from 127.0.0.1 using console on auto-deploy-iso-2007-uas-0
auto-deploy-iso-2007-uas-0#config
Entering configuration mode terminal
auto-deploy-iso-2007-uas-0(config)#load replace autodeploy.cfg
Loading.      14.63 KiB parsed in 0.42 sec (34.16 KiB/sec)

auto-deploy-iso-2007-uas-0(config)#commit
Commit complete.
auto-deploy-iso-2007-uas-0(config)#end
```

컨피그레이션 변경 후 uas-confd 및 Auto-Deploy 서비스를 다시 시작합니다.

```
root@auto-deploy-iso-2007-uas-0:~# service uas-confd restart
uas-confd stop/waiting
uas-confd start/running, process 14078
```

```
root@auto-deploy-iso-2007-uas-0:~# service uas-confd status
uas-confd start/running, process 14078
```

```
root@auto-deploy-iso-2007-uas-0:~# service autodeploy restart
autodeploy stop/waiting
autodeploy start/running, process 14017
root@auto-deploy-iso-2007-uas-0:~# service autodeploy status
autodeploy start/running, process 14017
```

Syslog 활성화

UCS 서버, Openstack 구성 요소 및 복구된 VM에 대해 syslog를 활성화하려면 다음 섹션을 따르십시오

아래 링크의 "UCS 및 Openstack 구성 요소에 대해 syslog 다시 활성화" 및 "VNF에 대해 syslog 활성화"

이 번역에 관하여

Cisco는 전 세계 사용자에게 다양한 언어로 지원 콘텐츠를 제공하기 위해 기계 번역 기술과 수작업 번역을 병행하여 이 문서를 번역했습니다. 아무리 품질이 높은 기계 번역이라도 전문 번역가의 번역 결과물만큼 정확하지는 않습니다. Cisco Systems, Inc.는 이 같은 번역에 대해 어떠한 책임도 지지 않으며 항상 원본 영문 문서(링크 제공됨)를 참조할 것을 권장합니다.