

# Ultra-M UCS 240M4 서버에서 마더보드 교체 - vEPC

## 목차

[소개](#)

[배경 정보](#)

[약어](#)

[MoP 워크플로](#)

[Ultra-M 설정에서 마더보드 교체](#)

[컴퓨팅 노드의 마더보드 교체](#)

[컴퓨팅 노드에서 호스팅되는 VM 식별](#)

[정상 전원 끄기](#)

[사례 1. 컴퓨팅 노드가 SF VM만 호스팅함](#)

[사례 2. 컴퓨팅 노드가 CF/ESC/EM/UAS를 호스팅함](#)

[마더보드 교체](#)

[VM 복원](#)

[사례 1. 컴퓨팅 노드가 SF VM만 호스팅함](#)

[사례 2. 컴퓨팅 노드 호스트 UAS, ESC, EM 및 CF](#)

[OSD 컴퓨팅 노드의 마더보드 교체](#)

[유지 관리 모드로 전환](#)

[OSD-컴퓨팅 노드에서 호스팅되는 VM 식별](#)

[정상 전원 끄기](#)

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

[케이스 2. OSD-Compute Node Hosts Auto-Deploy/Auto-It/EM/UAS](#)

[자동 구축의 CDB 백업](#)

[Auto-IT에서 System.cfg 백업](#)

[마더보드 교체](#)

[유지 관리 모드에서 CEPH 이동](#)

[VM 복원](#)

[케이스 1. OSD-Compute Node 호스트 CF, ESC, EM 및 UAS](#)

[케이스 2. OSD-Compute Node가 자동 구축, 자동 구축, EM 및 UAS 호스팅](#)

[자동 IT VM 복구](#)

[컨트롤러 노드의 마더보드 교체](#)

[컨트롤러 상태 확인 및 유지 관리 모드로 클러스터 설정](#)

[마더보드 교체](#)

[클러스터 상태 복원](#)

## 소개

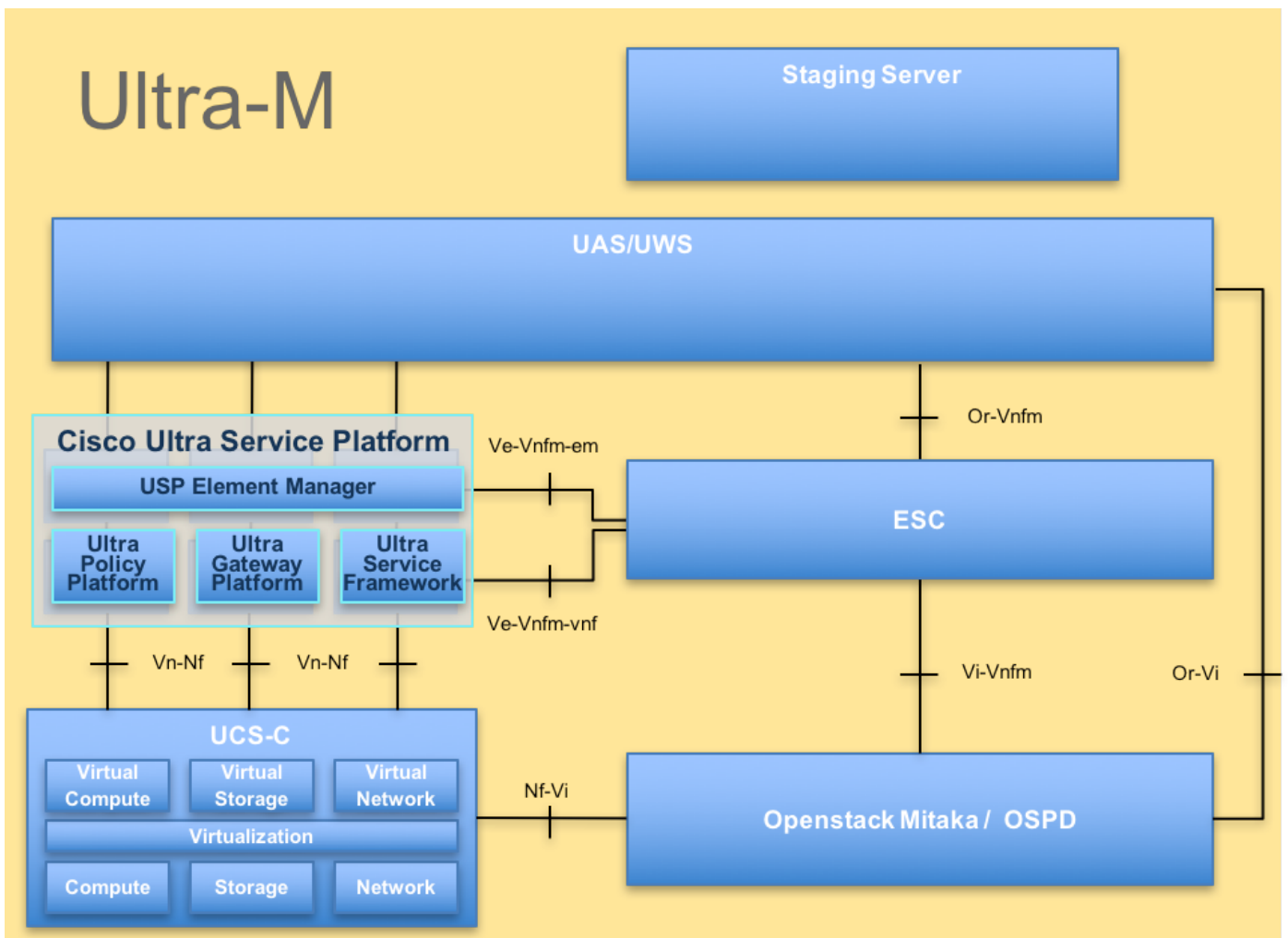
이 문서에서는 StarOS VNF(Virtual Network Functions)를 호스팅하는 Ultra-M 설정에서 서버의 결함이 있는 마더보드를 교체하는 데 필요한 단계를 설명합니다.

# 배경 정보

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

- 컴퓨팅
- 개체 스토리지 디스크 - 컴퓨팅(OSD - 컴퓨팅)
- 컨트롤러
- OpenStack Platform - 디렉터(OSPD)

Ultra-M 및 관련 구성 요소의 고급 아키텍처는 다음 이미지에 설명되어 있습니다.



UltraM 아키텍처

이 문서는 Cisco Ultra-M 플랫폼에 익숙한 Cisco 담당자를 대상으로 하며, 서버에서 마더보드 교체 시 OpenStack 및 StarOS VNF 레벨에서 수행해야 하는 단계를 자세히 설명합니다.

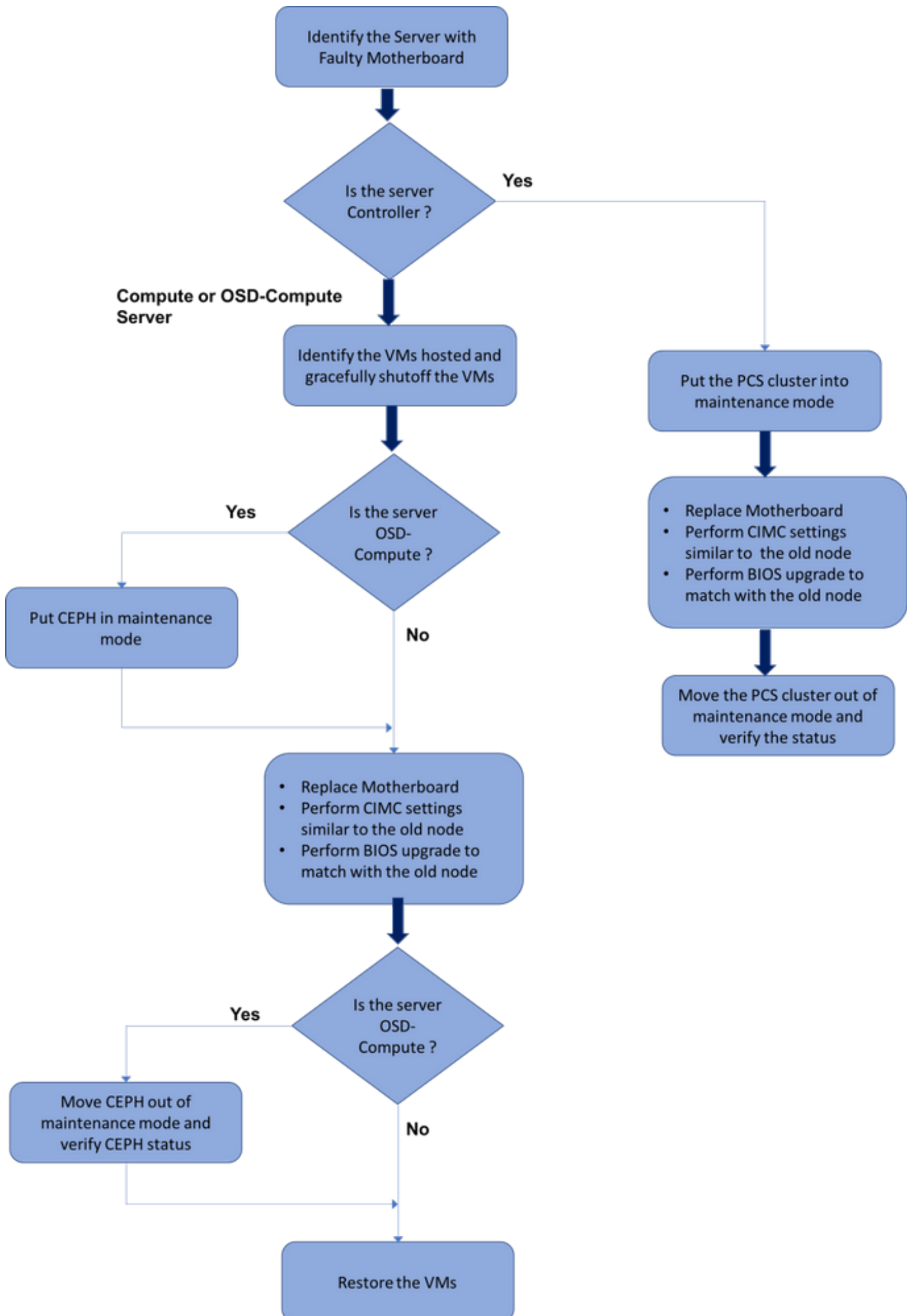
참고:Ultra M 5.1.x 릴리스는 이 문서의 절차를 정의하기 위해 고려됩니다.

## 약어

VNF            가상 네트워크 기능

CF	제어 기능
SF	서비스 기능
ESC	Elastic Service Controller
MOP	절차 방법
OSD	개체 스토리지 디스크
HDD	하드 디스크 드라이브
SSD	솔리드 스테이트 드라이브
VIM	가상 인프라 관리자
VM	가상 머신
EM	요소 관리자
UAS	Ultra Automation 서비스
UUID	범용 고유 식별자

## MoP 워크플로



## Ultra-M 설정에서 마더보드 교체

Ultra-M 설정에서는 다음과 같은 서버 유형에 마더보드 교체가 필요한 경우가 있을 수 있습니다. 컴퓨팅, OSD-컴퓨팅 및 컨트롤러

**참고:** 마더보드를 교체한 후 OpenStack이 설치된 부팅 디스크가 교체됩니다. 따라서 노드를 다시 오버클라우드에 추가할 필요가 없습니다. 교체 작업 후 서버의 전원이 켜지면 오버클라우드 스택에 다시 등록됩니다.

## 컴퓨팅 노드의 마더보드 교체

활동 전에 컴퓨팅 노드에 호스팅된 VM이 정상적으로 차단됩니다. 마더보드를 교체하면 VM이 다시 복원됩니다.

### 컴퓨팅 노드에서 호스팅되는 VM 식별

컴퓨팅 서버에서 호스팅되는 VM을 식별합니다. 두 가지 가능성이 있습니다.

컴퓨팅 서버에는 SF VM만 포함됩니다.

```
[stack@director ~]$ nova list --field name,host | grep compute-10
| 49ac5f22-469e-4b84-badc-031083db0533 | VNF2-DEPLOYM_s9_0_8bc6cc60-15d6-4ead-8b6a-10e75d0e134d
| pod1-compute-10.localdomain |
```

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

```
[stack@director ~]$ nova list --field name,host | grep compute-8
| 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 이름이고, 세 번째 열은 VM이 있는 호스트 이름입니다. 이 출력의 매개변수는 후속 섹션에서 사용됩니다.

## 정상 전원 끄기

### 사례 1. 컴퓨팅 노드가 SF VM만 호스팅함

StarOS VNF에 로그인하고 SF VM에 해당하는 카드를 식별합니다. Identify the VMs hosted in the

**Compute Node(컴퓨팅 노드에서 호스팅되는 VM 식별) 섹션에서 식별된 SF VM의 UUID를 사용하고 UUID에 해당하는 카드를 식별합니다.**

```
[local]VNF2# show card hardware
Tuesday night 08 16:49:42 UTC 2018
<snip>
Card 8:
  Card Type           : 4-Port Service Function Virtual Card
  CPU Packages        : 26 [#0, #1, #2, #3, #4, #5, #6, #7, #8, #9, #10, #11, #12, #13, #14, #15, #16, #17, #18, #19, #20, #21, #22, #23, #24, #25]
  CPU Nodes           : 2
  CPU Cores/Threads   : 26
  Memory              : 98304M (qvpc-di-large)
  UUID/Serial Number  : 49AC5F22-469E-4B84-BADC-031083DB0533
<snip>
```

카드의 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             Active       No
2: CFC        Control Function Virtual Card             Standby     -
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 이동합니다.

```
[local]VNF2# card migrate from 8 to 10
VNF에 해당하는 ESC 노드에 로그인하고 SF VM의 상태를 확인합니다.
```

```
[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_id>|<deployment_name>"
<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_s9_0_8bc6cc60-15d6-4ead-8b6a-10e75d0e134d</vm_name>
      <state>VM_ALIVE_STATE</state>
<snip>
```

VM 이름을 사용하여 SF VM을 중지합니다.(섹션에서 설명한 VM 이름은 컴퓨팅 노드에서 호스팅되는 VM을 식별합니다)◆◆

```
[admin@VNF2-esc-esc-0 esc-cli]$ ./esc_nc_cli vm-action STOP VNF2-DEPLOYM_s9_0_8bc6cc60-15d6-4ead-8b6a-10e75d0e134d
```

일단 중지되면 VM은 SHUTOFF 상태로 들어가야 합니다.

```
[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_id>|<deployment_name>"
<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>
    <vm_name>VNF2-DEPLOYM_s9_0_8bc6cc60-15d6-4ead-8b6a-10e75d0e134d</vm_name>
    <state>VM_SHUTOFF_STATE</state>
</snip>
```

## 사례 2. 컴퓨팅 노드가 CF/ESC/EM/UAS를 호스팅함

StarOS VNF에 로그인하고 CF VM에 해당하는 카드를 식별합니다. Identify the VMs hosted in the **Compute Node**(컴퓨팅 노드에서 호스팅되는 VM 식별) 섹션에서 식별된 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 (qvpc-di-large)
  UUID/Serial Number  : F9C0763A-4A4F-4BBD-AF51-BC7545774BE2
<snip>
```

카드의 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 이동합니다.

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

VNF에 해당하는 ESC 노드에 로그인하고 VM의 상태를 확인합니다.

```
[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_id>|<deployment_name>"
```

```

<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.(섹션에서 설명한 VM 이름은 컴퓨팅 노드에서 호스팅되는 VM을 식별합니다)◆◆

```

[admin@VNF2-esc-esc-0 esc-cli]$ ./esc_nc_cli vm-action STOP VNF2-DEPLOYM_c1_0_df4be88d-b4bf-4456-945a-3812653ee229

```

```

[admin@VNF2-esc-esc-0 esc-cli]$ ./esc_nc_cli vm-action STOP VNF2-DEPLOYM_XXXX_0_c8d98f0f-d874-45d0-af75-88a2d6fa82ea

```

VM이 중지되면 SHUTOFF 상태를 입력해야 합니다.

```

[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_id>|<deployment_name>"
<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>

```

<snip>  
 컴퓨팅 노드에서 호스팅되는 ESC에 로그인하여 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!
```

## 마더보드 교체

UCS C240 M4 서버의 마더보드를 교체하기 위한 단계는 [Cisco UCS C240 M4 Server 설치 및 서비스 가이드](#)에서 참조할 수 있습니다.

CIMC IP를 사용하여 서버에 로그인합니다.

펌웨어가 이전에 사용한 권장 버전에 따라 다르면 BIOS 업그레이드를 수행합니다. BIOS 업그레이드 단계는 다음과 같습니다. [Cisco UCS C-Series Rack-Mount Server BIOS 업그레이드 가이드](#)

## VM 복원

### 사례 1. 컴퓨팅 노드가 SF VM만 호스팅함

SF VM은 nova 목록에서 오류 상태가 됩니다.

```
[stack@director ~]$ nova list |grep VNF2-DEPLOYM_s9_0_8bc6cc60-15d6-4ead-8b6a-10e75d0e134d
| 49ac5f22-469e-4b84-badc-031083db0533 | VNF2-DEPLOYM_s9_0_8bc6cc60-15d6-4ead-8b6a-10e75d0e134d
| ERROR | - | NOSTATE |
```

ESC에서 SF VM을 복구합니다.

```
[admin@VNF2-esc-esc-0 ~]$ sudo /opt/cisco/esc/esc-confd/esc-cli/esc_nc_cli recovery-vm-action DO
VNF2-DEPLOYM_s9_0_8bc6cc60-15d6-4ead-8b6a-10e75d0e134d
[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/.ssh/confd_id_dsa --privKeyType=dsa --rpc=/tmp/esc_nc_cli.ZpRCGiieuW
<?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>
```

양세초로그를 모니터링합니다.

```
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-
DEPLOYM_s9_0_8bc6cc60-15d6-4ead-8b6a-10e75d0e134d].
```

SF 카드가 VNF에서 대기 SF로 나타나는지 확인합니다.

### 사례 2. 컴퓨팅 노드 호스트 UAS, ESC, EM 및 CF

## 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
| ACTIVE | - | Running | VNF2-UAS-uas-
orchestration=172.168.11.10; VNF2-UAS-uas-management=172.168.10.3
[stack@tb5-ospd ~]$ nova delete VNF2-UAS-uas-0
Request to delete server VNF2-UAS-uas-0 has been accepted.
```

AutoVNF-UAS VM을 복구하려면 UAS 확인 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-vol-0'
2017-11-22 14:01:49,525 - INFO: Created instance 'VNF2-UAS-uas-0'
```

AutoVNF-UAS에 로그인합니다. 몇 분 정도 기다리면 UAS가 정상 상태로 돌아갑니다.

```
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
```

## 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 | ACTIVE | - | Running | VNF2-
UAS-uas-orchestration=172.168.11.14; VNF2-UAS-uas-
management=172.168.10.4
|
[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 트랜잭션의 로그에서 **boot\_vm.py** 명령줄을 찾아 ESC 인스턴스를 생성합니다.

```
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                            STATUS
-----
35eefc4a-d4a9-11e7-bb72-fal63ef8df2b  vnf-deployment                        VNF2-DEPLOYMENT  2017-11-
29T02:01:27.750692-00:00  deployment-success
73d9c540-d4a8-11e7-bb72-fal63ef8df2b  vnf-deployment                        VNF2-ESC        2017-11-
29T01:56:02.133663-00:00  deployment-success
```

```
VNF2-uas-uas-0#show logs 73d9c540-d4a8-11e7-bb72-fal63ef8df2b | 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-fal63ef8df2b</tx-id>
    <log>2017-11-29 01:56:02,142 - VNF Deployment RPC triggered for deployment: VNF2-ESC,
deactivate: 0
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 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 *****
--os_password ***** --bs_os_auth_url http://10.1.2.5:5000/v2.0 --bs_os_tenant_name core --
bs_os_username ***** --bs_os_password ***** --esc_ui_startup false --esc_params_file
/tmp/esc_params.cfg --encrypt_key ***** --user_pass ***** --user_confid_pass ***** --kad_vif
eth0 --kad_vip 172.168.10.7 --ipaddr 172.168.10.6 dhcp --ha_node_list 172.168.10.3 172.168.10.6
--file root:0755:/opt/cisco/esc/esc-
scripts/esc_volume_em_staging.sh:/opt/cisco/usp/uas/autovnf/vnfms/esc-
scripts/esc_volume_em_staging.sh --file root:0755:/opt/cisco/esc/esc-
scripts/esc_vpc_chassis_id.py:/opt/cisco/usp/uas/autovnf/vnfms/esc-scripts/esc_vpc_chassis_id.py
--file root:0755:/opt/cisco/esc/esc-scripts/esc-vpc-di-internal-
keys.sh:/opt/cisco/usp/uas/autovnf/vnfms/esc-scripts/esc-vpc-di-internal-keys.sh
boot_vm.py 줄을 셸 스크립트 파일(esc.sh)에 저장하고 모든 사용자 이름 ***** 및 비밀번호 ***** 줄
을 올바른 정보(일반적으로 코어/<PASSWORD>)로 업데이트합니다. —encrypt_key 옵션도 제거해
야 합니다.user_pass 및 user_confid_pass의 경우 username 형식을 사용해야 합니다.password(예:
admin:<PASSWORD>).
```

running-config에서 bootvm.py에 대한 URL을 찾아 bootvm.py 파일을 AutoVNF UAS VM으로 가져옵니다. 이 경우 10.1.2.3은 AutoIT 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 --.-K/s in 0.001s
2017-12-01 20:25:52 (173 MB/s) - 'bootvm-2_3_2_155.py' saved [127771/127771]
```

### /tmp/esc\_params.cfg 파일 만들기:

```
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-dfd910436689
--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_url
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_confid_pass
admin:<PASSWORD> --kad_vif eth0 --kad_vip 172.168.10.7 --ipaddr 172.168.10.6 dhcp --ha_node_list 172.168.10.3
172.168.10.6 --file root:0755:/opt/cisco/esc/esc-scripts/esc_volume_em_staging.sh:/opt/cisco/usp/uas/autovnf/vnfms/esc-scripts/esc_volume_em_staging.sh
--file root:0755:/opt/cisco/esc/esc-scripts/esc_vpc_chassis_id.py:/opt/cisco/usp/uas/autovnf/vnfms/esc-scripts/esc_vpc_chassis_id.py
--file root:0755:/opt/cisco/esc/esc-scripts/esc-vpc-di-internal-keys.sh:/opt/cisco/usp/uas/autovnf/vnfms/esc-scripts/esc-vpc-di-internal-keys.sh
```

### 새 ESC에 로그인하여 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 상태에 있어야 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 | None | ERROR |
| f9c0763a-4a4f-4bbd-af51-bc7545774be2 | VNF2-DEPLOYM_c1_0_df4be88d-b4bf-4456-945a-
3812653ee229 |None | ERROR
```

ESC 마스터에 로그인하고 영향받는 각 EM 및 CF VM에 대해 **recovery-vm-action**을 실행합니다.인내심을 가지세요.ESC는 복구 작업을 예약하며 몇 분 동안 이 작업이 수행되지 않을 수 있습니다.양세초로그를 모니터링합니다.

```
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-DEPLOYMENT-_VNF2-D_0_a6843886-77b4-4f38-b941-74eb527113a8
[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/.ssh/confd_id_dsa --privKeyType=dsa --rpc=/tmp/esc_nc_cli.ZpRCGieiuW
<?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_0_a6843886-77b4-4f38-b941-74eb527113a8]
```

새 EM에 로그인하여 EM 상태가 작동 중인지 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 로그인하고 CF 카드가 대기 상태인지 확인합니다.

### ESC 복구 실패 처리

예기치 않은 상태로 인해 ESC가 VM을 시작하지 못하는 경우 마스터 ESC를 재부팅하여 ESC 전환을 수행하는 것이 좋습니다.ESC 전환은 약 1분 정도 걸립니다.새 마스터 ESC에서 health.sh 스크립트를 실행하여 상태가 작동 중인지 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.log을 모니터링할 수 있습니다.5~7분 후에

VM이 복구되지 않는 경우, 사용자는 이동하여 영향을 받는 VM을 수동으로 복구해야 합니다.

## OSD 컴퓨팅 노드의 마더보드 교체

활동 전에 컴퓨팅 노드에 호스팅된 VM이 정상적으로 종료되고 Ceph가 유지 관리 모드로 전환됩니다. 마더보드를 교체하면 VM이 다시 복원되고 Ceph가 유지 보수 모드에서 벗어납니다.

### 유지 관리 모드로 전환

서버에서 CEPH osd 트리 상태가 작동 중인지 확인합니다.

```
[heat-admin@pod1-osd-compute-1 ~]$ 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

OSD Compute(OSD 컴퓨팅) 노드에 로그인하고 Ceph를 유지 관리 모드로 전환합니다.

```
[root@pod1-osd-compute-1 ~]# sudo ceph osd set norebalance
```

```
[root@pod1-osd-compute-1 ~]# sudo ceph osd set noout
```

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

```
cluster eb2bb192-b1c9-11e6-9205-525400330666
health HEALTH_WARN
noout,norebalance,sortbitwise,require_jewel_osds flag(s) set
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 e194: 12 osds: 12 up, 12 in
flags noout,norebalance,sortbitwise,require_jewel_osds
pgmap v584865: 704 pgs, 6 pools, 531 GB data, 344 kobjects
1585 GB used, 11808 GB / 13393 GB avail
704 active+clean
client io 463 kB/s rd, 14903 kB/s wr, 263 op/s rd, 542 op/s wr
```

**참고:**Ceph를 제거하면 VNF HD RAID가 저하됨 상태로 전환되지만 HDD에 계속 액세스할 수 있어야 합니다.

## OSD-컴퓨팅 노드에서 호스팅되는 VM 식별

OSD 컴퓨팅 서버에서 호스팅되는 VM을 식별합니다. 두 가지 가능성이 있습니다.

osd-compute 서버에는 VM의 EM(Element Manager)/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 | pod1-osd-compute-0.localdomain
|
| 2d051522-bce2-4809-8d63-0c0e17f251dc | AUTO-IT-VNF2-uas-0 | pod1-osd-compute-0.localdomain |
| 507d67c2-1d00-4321-b9d1-da879af524f8 | VNF2-DEPLOYM_XXXX_0_c8d98f0f-d874-45d0-af75-
88a2d6fa82ea | pod1-osd-compute-0.localdomain |
| f5bd7b9c-476a-4679-83e5-303f0aae9309 | VNF2-UAS-uas-0 | pod1-osd-compute-0.localdomain |
```

컴퓨팅 서버에는 VM의 CF(Control Function)/ESC(Elastic Services Controller)/EM(Element Manager)/(UAS) 조합이 포함되어 있습니다.

```
[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 이름이고, 세 번째 열은 VM이 있는 호스트 이름입니다. 이 출력의 매개변수는 후속 섹션에서 사용됩니다.

## 정상 전원 끄기

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

CF/ESC/EM/UAS VM의 전원을 정상적으로 끄는 절차는 VM이 컴퓨팅 또는 OSD-컴퓨팅 노드에서 호스팅되는지 여부와 상관없이 동일합니다. 컴퓨팅 노드의 마더보드 교체에서 VM의 전원을 정상적으로 끄려면 다음 단계를 수행합니다.

### 케이스 2. OSD-Compute Node Hosts Auto-Deploy/Auto-It/EM/UAS

#### 자동 구축의 CDB 백업

자동 구축 confd cdb 데이터를 정기적으로 또는 활성화/비활성화 후 백업하고 백업 서버에 저장합니다. 자동 배포가 중복되지 않으며 이 데이터가 손실되면 배포를 정상적으로 비활성화할 수 없습니다.

AutoDeploy VM에 로그인하고 백업을 구성하는 cdb 디렉토리에 로그인합니다.

```
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/O.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:** 서버를 백업하려면 autodeploy\_cdb\_backup.tar를 복사합니다.

## Auto-IT에서 System.cfg 백업

system.cfg 파일의 백업을 사용하여 서버를 백업합니다.

```
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
```

```
100% 565 0.6KB/s 00:00
```

```
ubuntu@auto-it-vnf-uas-0:~$
```

**참고:**EM/UAS VM의 전원을 정상적으로 켜는 절차는 VM이 컴퓨팅 또는 OSD-컴퓨팅 노드에



서 호스팅되는지 여부에 관계없이 동일합니다.

이러한 VM의 전원을 정상적으로 끄려면 컴퓨팅 노드에서 마더보드 교체 단계를 수행하십시오.

## 마더보드 교체

UCS C240 M4 서버의 마더보드를 교체하기 위한 단계는 [Cisco UCS C240 M4 Server 설치 및 서비스 가이드](#)에서 참조할 수 있습니다.

CIMC IP를 사용하여 서버에 로그인합니다.

펌웨어가 이전에 사용한 권장 버전에 따라 다르면 BIOS 업그레이드를 수행합니다. BIOS 업그레이드 단계는 다음과 같습니다. [Cisco UCS C-Series Rack-Mount Server BIOS 업그레이드 가이드](#)

## 유지 관리 모드에서 CEPH 이동

OSD Compute(OSD 컴퓨팅) 노드에 로그인하고 Ceph를 유지 보수 모드에서 꺼냅니다.

```
[root@pod1-osd-compute-1 ~]# sudo ceph osd unset norebalance
[root@pod1-osd-compute-1 ~]# sudo ceph osd unset noout

[root@pod1-osd-compute-1 ~]# sudo ceph status

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 e196: 12 osds: 12 up, 12 in
flags sortbitwise,require_jewel_osds
pgmap v584954: 704 pgs, 6 pools, 531 GB data, 344 kobjects
1585 GB used, 11808 GB / 13393 GB avail
704 active+clean
client io 12888 kB/s wr, 0 op/s rd, 81 op/s wr
```

## VM 복원

### 케이스 1. OSD-Compute Node 호스트 CF, ESC, EM 및 UAS

CF/ESC/EM/UAS VM의 복원 절차는 VM이 컴퓨팅 또는 OSD-컴퓨팅 노드에서 호스팅되는지 여부에 관계없이 동일합니다. Case 2. Compute Node Hosts CF/ESC/EM/UAS의 단계를 따라 VM을 복원합니다.

### 케이스 2. OSD-Compute Node가 자동 구축, 자동 구축, EM 및 UAS 호스팅

#### VM 자동 구축 복구

OSPD에서 자동 배포 VM이 영향을 받았지만 여전히 ACTIVE/Running이 표시되면 먼저 삭제해야

합니다. 자동 구축이 영향을 받지 않은 경우 자동 VM 복구로 건너뛴니다.

```
[stack@director ~]$ nova list |grep auto-deploy
| 9b55270a-2dcd-4ac1-aba3-bf041733a0c9 | auto-deploy-ISO-2007-uas-
0 | ACTIVE | - | Running | mgmt=172.16.181.12,
10.1.2.7 [stack@director ~]$ cd /opt/cisco/usp/uas-installer/scripts
```

```
[stack@director ~]$ ./auto-deploy-booting.sh --floating-ip 10.1.2.7 --delete
자동 배포가 삭제되면 동일한 floatingip 주소로 다시 생성합니다.
```

```
[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://10.84.123.4:5000/v2.0' tenant 'core' user 'core', ISO 'default'
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 '/opt/cisco/usp/uas-installer/images/usp-uas-1.0.1-1504.qcow2'
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
```

Autodeploy.cfg 파일, ISO 및 confd\_backup tar 파일을 백업 서버에서 복사하여 VM을 자동 구축하  
고 백업 tar 파일에서 confd cdb 파일을 복원합니다.

```
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/O.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 컨피그레이션이 제대로 로드되었는지 확인합니다. autodeploy.cfg를 새  
osd-compute 이름으로 업데이트합니다. 섹션- 최종 단계: 자동 배포 구성을 업데이트합니다.

```
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
```

SERVICE

SITE

```

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

```

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

## 자동 IT VM 복구

OSPD에서 자동-it VM이 영향을 받았지만 여전히 활성/실행으로 표시되면 삭제해야 합니다. 자동-적용이 영향을 받지 않은 경우 다음으로 건너뛸니다.

```

[stack@director ~]$ nova list |grep auto-it
| 580faf80-1d8c-463b-9354-781ea0c0b352 | auto-it-vnf-ISO-2007-uas-
0 | ACTIVE | - | Running | mgmt=172.16.181.3,
10.1.2.8 [stack@director ~]$ 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 실행하여 자동 IT 재생성:

```
[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://10.84.123.4:5000/v2.0' tenant 'core' user 'core', ISO 'default'
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 '/opt/cisco/usp/uas-installer/images/usp-uas-1.0.1-1504.qcow2'
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 자동 IT IP 주소는 10.1.2.8입니다. 로드하는 데 몇 분 정도 걸립니다.

```
[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"
    }
  ]
}

```

}  
VNF system.cfg 파일을 OSPD Auto-Deploy 디렉터리에서 Auto-IT VM으로 복사합니다.

```
[stack@director autodeploy]$ scp system-vnf* ubuntu@10.1.2.8:.  
  
ubuntu@10.1.2.8's password:  
system-  
vnf1.cfg  
    100% 1197    1.2KB/s   00:00  
system-vnf2.cfg  
                                100% 1197    1.2KB/s   00:00  
  
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 VM이 컴퓨팅 또는 OSD-Compute에서 호스팅되는지 여부에 관계없이 동일합니다. 컴퓨팅 노드에서 마더보드 교체의 단계를 수행하여 이러한 VM의 전원을 정상적으로 끕니다.

## 컨트롤러 노드의 마더보드 교체

### 컨트롤러 상태 확인 및 유지 관리 모드로 클러스터 설정

OSPD에서 컨트롤러에 로그인하고 PC가 정상 상태인지 확인합니다. Online(온라인) 및 Galera(갤러리) 3개 컨트롤러 모두 3개의 컨트롤러를 모두 Master(마스터)로 표시합니다.

```
[heat-admin@pod1-controller-0 ~]$ sudo pcs status  
Cluster name: tripleo_cluster  
Stack: corosync  
Current DC: pod1-controller-2 (version 1.1.15-11.e17_3.4-e174ec8) - partition with quorum  
Last updated: Mon Dec 4 00:46:10 2017 Last change: Wed Nov 29 01:20:52 2017 by hacluster via  
crmd on pod1-controller-0  
  
3 nodes and 22 resources configured  
  
Online: [ pod1-controller-0 pod1-controller-1 pod1-controller-2 ]  
  
Full list of resources:  
  
ip-11.118.0.42 (ocf::heartbeat:IPaddr2): Started pod1-controller-1  
ip-11.119.0.47 (ocf::heartbeat:IPaddr2): Started pod1-controller-2  
ip-11.120.0.49 (ocf::heartbeat:IPaddr2): Started pod1-controller-1  
ip-192.200.0.102 (ocf::heartbeat:IPaddr2): Started pod1-controller-2  
Clone Set: haproxy-clone [haproxy]  
Started: [ pod1-controller-0 pod1-controller-1 pod1-controller-2 ]  
Master/Slave Set: galera-master [galera]  
Masters: [ pod1-controller-0 pod1-controller-1 pod1-controller-2 ]  
ip-11.120.0.47 (ocf::heartbeat:IPaddr2): Started pod1-controller-2  
Clone Set: rabbitmq-clone [rabbitmq]  
Started: [ pod1-controller-0 pod1-controller-1 pod1-controller-2 ]  
Master/Slave Set: redis-master [redis]  
Masters: [ pod1-controller-2 ]  
Slaves: [ pod1-controller-0 pod1-controller-1 ]  
ip-10.84.123.35 (ocf::heartbeat:IPaddr2): Started pod1-controller-1
```

```
openstack-cinder-volume (systemd:openstack-cinder-volume): Started pod1-controller-2
my-ipmilan-for-controller-0 (stonith:fence_ipmilan): Started pod1-controller-0
my-ipmilan-for-controller-1 (stonith:fence_ipmilan): Started pod1-controller-0
my-ipmilan-for-controller-2 (stonith:fence_ipmilan): Started pod1-controller-0
```

Daemon Status:

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

클러스터를 유지 관리 모드로 설정합니다.

```
[heat-admin@pod1-controller-0 ~]$ sudo pcs cluster standby
```

```
[heat-admin@pod1-controller-0 ~]$ sudo pcs status
```

```
Cluster name: tripleo_cluster
Stack: corosync
Current DC: pod1-controller-2 (version 1.1.15-11.e17_3.4-e174ec8) - partition with quorum
Last updated: Mon Dec 4 00:48:24 2017 Last change: Mon Dec 4 00:48:18 2017 by root via
crm_attribute on pod1-controller-0
```

3 nodes and 22 resources configured

**Node pod1-controller-0: standby**

```
Online: [ pod1-controller-1 pod1-controller-2 ]
```

Full list of resources:

```
ip-11.118.0.42 (ocf::heartbeat:IPaddr2): Started pod1-controller-1
ip-11.119.0.47 (ocf::heartbeat:IPaddr2): Started pod1-controller-2
ip-11.120.0.49 (ocf::heartbeat:IPaddr2): Started pod1-controller-1
ip-192.200.0.102 (ocf::heartbeat:IPaddr2): Started pod1-controller-2
Clone Set: haproxy-clone [haproxy]
Started: [ pod1-controller-1 pod1-controller-2 ]
Stopped: [ pod1-controller-0 ]
Master/Slave Set: galera-master [galera]
Masters: [ pod1-controller-1 pod1-controller-2 ]
Slaves: [ pod1-controller-0 ]
ip-11.120.0.47 (ocf::heartbeat:IPaddr2): Started pod1-controller-2
Clone Set: rabbitmq-clone [rabbitmq]
Started: [ pod1-controller-0 pod1-controller-1 pod1-controller-2 ]
Master/Slave Set: redis-master [redis]
Masters: [ pod1-controller-2 ]
Slaves: [ pod1-controller-1 ]
Stopped: [ pod1-controller-0 ]
ip-10.84.123.35 (ocf::heartbeat:IPaddr2): Started pod1-controller-1
```

```
openstack-cinder-volume (systemd:openstack-cinder-volume): Started pod1-controller-2
my-ipmilan-for-controller-0 (stonith:fence_ipmilan): Started pod1-controller-1
my-ipmilan-for-controller-1 (stonith:fence_ipmilan): Started pod1-controller-1
my-ipmilan-for-controller-2 (stonith:fence_ipmilan): Started pod1-controller-2
```

## 마더보드 교체

UCS C240 M4 서버의 마더보드를 교체하기 위한 단계는 [Cisco UCS C240 M4 Server 설치 및 서비스 가이드](#)에서 참조할 수 있습니다.

CIMC IP를 사용하여 서버에 로그인합니다.

펌웨어가 이전에 사용한 권장 버전에 따라 다르면 BIOS 업그레이드를 수행합니다. BIOS 업그레이드 단계는 다음과 같습니다. [Cisco UCS C-Series Rack-Mount Server BIOS 업그레이드 가이드](#)

## 클러스터 상태 복원

영향을 받은 컨트롤러에 로그인하고 unstandby를 설정하여 대기 모드를 제거합니다. 컨트롤러가 클러스터와 함께 온라인 상태인지, Galera에서 세 컨트롤러를 모두 Master로 표시하는지 확인합니다. 몇 분 정도 걸릴 수 있습니다.

```
[heat-admin@pod1-controller-0 ~]$ sudo pcs cluster unstandby
```

```
[heat-admin@pod1-controller-0 ~]$ sudo pcs status
```

```
Cluster name: tripleo_cluster
```

```
Stack: corosync
```

```
Current DC: pod1-controller-2 (version 1.1.15-11.e17_3.4-e174ec8) - partition with quorum
```

```
Last updated: Mon Dec 4 01:08:10 2017 Last change: Mon Dec 4 01:04:21 2017 by root via
```

```
crm_attribute on pod1-controller-0
```

```
3 nodes and 22 resources configured
```

```
Online: [ pod1-controller-0 pod1-controller-1 pod1-controller-2 ]
```

```
Full list of resources:
```

```
ip-11.118.0.42 (ocf::heartbeat:IPaddr2): Started pod1-controller-1
```

```
ip-11.119.0.47 (ocf::heartbeat:IPaddr2): Started pod1-controller-2
```

```
ip-11.120.0.49 (ocf::heartbeat:IPaddr2): Started pod1-controller-1
```

```
ip-192.200.0.102 (ocf::heartbeat:IPaddr2): Started pod1-controller-2
```

```
Clone Set: haproxy-clone [haproxy]
```

```
Started: [ pod1-controller-0 pod1-controller-1 pod1-controller-2 ]
```

```
Master/Slave Set: galera-master [galera]
```

```
Masters: [ pod1-controller-0 pod1-controller-1 pod1-controller-2 ]
```

```
ip-11.120.0.47 (ocf::heartbeat:IPaddr2): Started pod1-controller-2
```

```
Clone Set: rabbitmq-clone [rabbitmq]
```

```
Started: [ pod1-controller-0 pod1-controller-1 pod1-controller-2 ]
```

```
Master/Slave Set: redis-master [redis]
```

```
Masters: [ pod1-controller-2 ]
```

```
Slaves: [ pod1-controller-0 pod1-controller-1 ]
```

```
ip-10.84.123.35 (ocf::heartbeat:IPaddr2): Started pod1-controller-1
```

```
openstack-cinder-volume (systemd:openstack-cinder-volume): Started pod1-controller-2
```

```
my-ipmilan-for-controller-0 (stonith:fence_ipmilan): Started pod1-controller-1
```

```
my-ipmilan-for-controller-1 (stonith:fence_ipmilan): Started pod1-controller-1
```

```
my-ipmilan-for-controller-2 (stonith:fence_ipmilan): Started pod1-controller-2
```

```
Daemon Status:
```

```
corosync: active/enabled
```

```
pacemaker: active/enabled
```

```
pcsd: active/enabled
```