서버 UCS C240 M4 - vEPC의 결함 있는 구성 요 소 교체

목차
<u>소개</u>
<u>배경 정보</u>
<u>약어</u>
<u>MoP의 워크플로</u>
사전 요구 사항
<u>백업</u>
<u>구성 요소 RMA - 컴퓨팅/OSD-컴퓨팅 노드</u>
<u>Compute/OSD-Compute 노드에서 호스팅되는 VM 식별</u>
정상 전원 끄기
<u>사례 1. 컴퓨팅 노드가 SF VM만 호스팅함</u>
케이스 2. 컴퓨팅/OSD-컴퓨팅 노드가 호스트 CF/ESC/EM/UAS
<u>Compute/OSD-Compute 도드에서 결함이 있는 구성 요소 교제</u>
<u>VM 목원</u>
<u>사례 1. 컴퓨팅 노드가 SF VM만 호스팅함</u> 케이스 2. 컴퓨팅/OSD-컴퓨팅 노도가 호스트 CE ESC EM 및 UAS
<u> </u>
<u>자동 구축 컨피그레이션 업데이트</u>
<u>구성 요소 RMA - 컨트롤러 노드</u>
<u>사전 점검</u>
<u>컨트롤러 클러스터를 유지 관리 모드로 이동</u>
<u>컨트롤러 노드에서 결함이 있는 구성 요소 교체</u>
<u>서버 전원 켜기</u>

소개

이 문서에서는 StarOS VNF(가상 네트워크 기능)를 호스팅하는 Ultra-M 설정에서 UCS(Unified Computing System) 서버에 언급된 결함 있는 구성 요소를 교체하는 데 필요한 단계에 대해 설명합 니다.

- 듀얼 인라인 메모리 모듈(DIMM) 교체 MOP
- FlexFlash 컨트롤러 오류
- SSD(Solid State Drive) 장애
- TPM(Trusted Platform Module) 실패
- Raid 캐시 실패
- Raid 컨트롤러/HBA(Hot Bus Adapter) 장애
- PCI 라이저 장애
- PCIe 어댑터 Intel X520 10G 오류

- 모듈형 MLOM(LAN-on Motherboard) 오류
- 팬 트레이 RMA
- CPU 오류

배경 정보

Ultra-M은 VNF의 구축을 간소화하도록 설계된 사전 패키지 및 검증된 가상 모바일 패킷 코어 솔루 션입니다. OpenStack은 Ultra-M용 VIM(Virtualized Infrastructure Manager)이며 다음 노드 유형으 로 구성됩니다.

- 컴퓨팅
- 개체 스토리지 디스크 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, 서버의 구성 요소 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.

약어

VNF	가상 네트워크 기능
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
엠	요소 관리자
UAS	Ultra Automation 서비스
UUID	보편적으로 고유한 식별자

MoP의 워크플로



사전 요구 사항

백업

결함이 있는 구성 요소를 교체하기 전에 Red Hat OpenStack Platform 환경의 현재 상태를 확인하는 것이 중요합니다. 교체 프로세스가 켜져 있을 때 합병증을 피하기 위해 현재 상태를 확인하는 것이 좋습니다. 이러한 교체 흐름으로 달성할 수 있습니다.

복구의 경우 다음 단계를 사용하여 OSPD 데이터베이스를 백업하는 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, 특히 교체할 컴퓨팅/OSD-컴퓨팅 노드가 CF(Control Function) VM(Virtual Machine)을 호스트하는 경우 StarOS 구성을 백업하는 것이 좋습니다.

✤ 참고: 서버가 컨트롤러 노드인 경우 "" 섹션으로 진행합니다. 그렇지 않으면 다음 섹션을 계속 합니다.

구성 요소 RMA - 컴퓨팅/OSD-컴퓨팅 노드

Compute/OSD-Compute 노드에서 호스팅되는 VM 식별

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

• 서버에는 SF(서비스 기능) VM만 포함되어 있습니다.

<#root>

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

<#root>

```
[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-38	12653ee229	
pod1-compute-8.localdomain 75528898-ef4b-4d68-b05d-882014708694		
VNF2-ESC-0		
f5bd7b9c-476a-4679-83e5-303f0aae9309	pod1-compute-8.localdomain	I
VNF2-UAS-uas-0		
	pod1-compute-8.localdomain	I

✤ 참고: 여기에 표시된 출력에서 첫 번째 열은 UUID(Universally Unique IDentifier)에 해당하고, 두 번째 열은 VM 이름이며, 세 번째 열은 VM이 있는 호스트 이름입니다. 이 출력의 매개변수 는 후속 섹션에서 사용됩니다.

정상 전원 끄기

사례 1. 컴퓨팅 노드가 SF VM만 호스팅함

SF 카드를 대기 상태로 마이그레이션

• StarOS VNF에 로그인하고 SF VM에 해당하는 카드를 식별합니다. "Identify the VMs hosted in the Compute/OSD-Compute Node(컴퓨팅/OSD-컴퓨팅 노드에서 호스팅되는 VM 식별)" 섹 션에서 식별된 SF VM의 UUID를 사용하고 UUID에 해당하는 카드를 식별합니다.

<#root>

[local]VNF2# show card hardware Tuesday might 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, #

      CPU Nodes
      : 2

      CPU Cores/Threads
      : 26

      Memory
      : 98304M (qvpc-di-large)

      UUID/Serial Number
      :
```

49AC5F22-469E-4B84-BADC-031083DB0533

• 카드 상태를 확인합니다.

<#root>

[local]VNF2# show card table Tuesday might 08 16:52:53 UTC 2018

rucsuuj mi				
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 전환합니다.

[local]VNF2# card migrate from 8 to 10

ESC에서 SF VM 종료

• VNF에 해당하는 ESC 노드에 로그인하고 SF 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>
VNF2-DEPLOYM_c1_0_df4be88d-b4bf-4456-945a-3812653ee229
VM_ALIVE_STATE
VNF2-DEPLOYM_s9_0_8bc6cc60-15d6-4ead-8b6a-10e75d0e134d
```

VM_ALIVE_STATE

</state> <snip>

> 해당 VM 이름을 사용하여 SF VM을 중지합니다. (VM 이름은 "Identify the VMs hosted in the Compute/OSD-Compute Node(컴퓨팅/OSD-컴퓨팅 노드에서 호스팅되는 VM 식별)" 섹션에 서 명시됩니다.)

[admin@VNF2-esc-esc-0 esc-cli]\$./esc_nc_cli vm-action STOP VNF2-DEPLOYM_s9_0_8bc6cc60-15d6-4ead-8b6a-1

• 일단 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>
VNF2-DEPLOYM_c1_0_df4be88d-b4bf-4456-945a-3812653ee229
VM_ALIVE_STATE
VNF2-DEPLOYM_c3_0_3e0db133-c13b-4e3d-ac14-
VM_ALIVE_STATE
VNF2-DEPLOYM_s9_0_8bc6cc60-15d6-4ead-8b6a-10e75d0e134d
```

VM_SHUTOFF_STATE

</state>

케이스 2. 컴퓨팅/OSD-컴퓨팅 노드가 호스트 CF/ESC/EM/UAS

CF 카드를 대기 상태로 마이그레이션

• StarOS VNF에 로그인하고 CF VM에 해당하는 카드를 식별합니다. "노드에서 호스팅되는 VM 식별" 섹션에서 식별된 CF VM의 UUID를 사용하고 UUID에 해당하는 카드를 찾습니다.

[local]VNF2# show card har	rdv	vare
Tuesday might 08 16:49:42	U٦	FC 2018
<snip></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></snip>		

• 카드 상태를 확인합니다.

Slot	Card Type	Oper State	SPOF	Attach
Tuesday migh [.]	t 08 16:52:53 UTC 2018			
[local]VNF2#	show card table			

1.	CEC	Control	Functio	n Virtual	Card		Standhy	_
т.	CIC	Concror	Tuncero	in vircua	Caru		Scandby	
2:	CFC	Control	Functio	n Virtual	Card		Active	No
3:	FC	4-Port S	Service	Function	Virtual	Card	Active	No
4:	FC	4-Port S	ervice	Function	Virtual	Card	Active	No
5:	FC	4-Port S	ervice	Function	Virtual	Card	Active	No
6:	FC	4-Port S	ervice	Function	Virtual	Card	Active	No
7:	FC	4-Port S	Service	Function	Virtual	Card	Active	No
8:	FC	4-Port S	Service	Function	Virtual	Card	Active	No
9:	FC	4-Port S	Service	Function	Virtual	Card	Active	No
10:	FC	4-Port S	ervice	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>
```

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

VM_ALIVE_STATE

</state>

VNF2-DEPLOYM_c3_0_3e0db133-c13b-4e3d-ac14-VM_ALIVE_STATE <deployment_name>VNF2-DEPLOYMENT-em</deployment_name> 507d67c2-1d00-4321-b9d1-da879af524f8 dc168a6a-4aeb-4e81-abd9-91d7568b5f7c 9ffec58b-4b9d-4072-b944-5413bf7fcf07 SERVICE_ACTIVE_STATE VNF2-DEPLOYM_XXXX_0_c8d98f0f-d874-45d0-af75-88a2d6fa82ea

VM_ALIVE_STATE

</state> <snip>

• VM 이름을 사용하여 CF 및 EM VM을 하나씩 중지합니다. (VM 이름은 "Identify the VMs hosted in the Compute/OSD-Compute Node(컴퓨팅/OSD-컴퓨팅 노드에서 호스팅되는 VM

```
식별)" 섹션에서 명시됩니다.)
```

[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>
```

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

</vm_name>

VM_SHUTOFF_STATE

</state>

VNF2-DEPLOYM_c3_0_3eOdb133-c13b-4e3d-ac14-VM_ALIVE_STATE <deployment_name>VNF2-DEPLOYMENT-em</deployment_name> 507d67c2-1d00-4321-b9d1-da879af524f8 dc168a6a-4aeb-4e81-abd9-91d7568b5f7c 9ffec58b-4b9d-4072-b944-5413bf7fcf07 SERVICE_ACTIVE_STATE

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

</vm_name>

VM_SHUTOFF_STATE

```
<snip>
```

ESC를 대기 모드로 마이그레이션

• 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.

✤ 참고: OSD-Compute 노드에서 결함이 있는 구성 요소를 교체하려면 구성 요소 교체를 진행하 기 전에 Ceph를 서버의 Maintenance(유지 관리)에 넣으십시오.

<#root>

```
[admin@osd-compute-0 ~]$ sudo ceph osd set norebalance set norebalance
```

```
[admin@osd-compute-0 ~]\$ sudo ceph osd set noout set noout
```

[admin@osd-compute-0 ~]\$ 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 {tb3-ultram-pod1-controller-0=11.118.0.40:6789/0,tb3-ultram-pod1-controller-1
election epoch 58, quorum 0,1,2 tb3-ultram-pod1-controller-0,tb3-ultram-pod1-controller-1,t
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

Compute/OSD-Compute 노드에서 결함이 있는 구성 요소 교체

지정된 서버의 전원을 끕니다. UCS C240 M4 서버에서 결함 있는 구성 요소를 교체하는 절차는 다음 링크에서 참조할 수 있습니다.

<u>서버 구성 요소 교체</u>

VM 복원

사례 1. 컴퓨팅 노드가 SF VM만 호스팅함

ESC에서 SF VM 복구

• 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 | E

• ESC에서 SF VM을 복구합니다.

[admin@VNF2-esc-esc-0 ~]\$ sudo /opt/cisco/esc/esc-confd/esc-cli/esc_nc_cli recovery-vm-action D0 VNF2-D
[sudo] password for admin:

• yangesc.log를 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

• VNF에서 SF 카드가 대기 SF로 나타나는지 확인합니다.

케이스 2. 컴퓨팅/OSD-컴퓨팅 노드가 호스트 CF, ESC, EM 및 UAS

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@tb5-ospd ~]\$ nova delete VNF2-UAS-uas-0 Request to delete server VNF2-UAS-uas-0 has been accepted.

 autovnf-uas VM을 복구하려면 상태를 확인하기 위해 uas-check 스크립트를 실행합니다. 오류 를 보고해야 합니다. 그런 다음 누락된 UAS VM을 다시 생성하려면 다음 옵션을 수정하여 다 시 실행합니다.

```
[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 - INF0: Check of AutoVNF cluster started
2017-12-08 12:38:07,925 - INF0: Instance 'vnf1-UAS-uas-0' status is 'ERROR'
2017-12-08 12:38:07,925 - INF0: 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 - INF0: Check of AutoVNF cluster started
2017-11-22 14:01:09,575 - INF0: Instance VNF2-UAS-uas-0' status is 'ERROR'
2017-11-22 14:01:09,575 - INF0: Check completed, AutoVNF cluster has recoverable errors
2017-11-22 14:01:09,575 - INF0: Check completed, AutoVNF cluster has recoverable errors
2017-11-22 14:01:09,778 - INF0: Removing instance VNF2-UAS-uas-0'
2017-11-22 14:01:13,568 - INF0: Removed instance VNF2-UAS-uas-0'
2017-11-22 14:01:13,568 - INF0: Creating instance VNF2-UAS-uas-0' and attaching volume 'VNF2-UAS-uas-vo
2017-11-22 14:01:49,525 - INF0: Created instance 'VNF2-UAS-uas-0'
```

• autovnf-uas에 로그인합니다. 몇 분 정도 기다리면 UAS가 정상 상태로 돌아와야 합니다.



```
[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 찾 아 ESC 인스턴스를 생성합니다.

<#root>

ubuntu@VNF2-uas-uas-0:~\$ sudo -i root@VNF2-uas-uas-0:~# confd_cli -u a Welcome to the ConfD CLI admin connected from 127.0.0.1 using VNF2-uas-uas-0#show transaction	dmin -C console on VNF2-u	as-uas-0	
TX ID	ΤΧ ΤΥΡΕ	DEPLOYMENT ID	TIMESTAMP
35eefc4a-d4a9-11e7-bb72-fa163ef8df2b 73d9c540-d4a8-11e7-bb72-fa163ef8df2b	vnf-deployment vnfm-deployment	VNF2-DEPLOYMENT VNF2-ESC	2017-11-29T02:01:27.750692-00:0 2017-11-29T01:56:02.133663-00:0
<pre>VNF2-uas-uas-0#show logs 73d9c540-d4a <config xmlns="http://tail-f.com/ns/c</td><td>8-11e7-bb72-fa163
onfig/1.0"> sp/nfv/usp-autovn 163ef8df2bFM Deployment RPC oyment</config></pre>	ef8df2b display <u>f-oper</u> "> > triggered for de	<pre>xml ployment: VNF2-ESC, deactivate: </pre>	
bootym by WNE2-ESC-ESC-1flavor WNE	PM VNFZ-ESC-ESC-	I with [python /	/opt/clsco/viil-staging/ 961b-4651-af22-dfd910436689pei
Decourter and a second of the	2-BOC-BOC-LIAVOI	image Steubis/-	2010-1031-0122-010310430003He

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

• running-config에서 bootvm.py를 실행하여 autovnf-uas VM에 bootvm.py 파일을 가져오기 위 한 URL을 찾습니다. 이 경우 10.1.2.3이 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 0K
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 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-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_aut 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
```

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

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 | Non
```

• ESC 마스터에 로그인하여 영향을 받는 각 EM 및 CF VM에 대해 recovery-vm-action을 실행 합니다. 인내심을 가져라. ESC는 복구 작업을 예약하며 몇 분 동안 발생하지 않을 수 있습니 다. 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 D0 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"?>

• 새 EM에 로그인하고 EM 상태가 정상인지 확인합니다.

• StarOS VNF에 로그인하여 CF 카드가 대기 상태인지 확인합니다

ESC 복구 실패 처리

예기치 않은 상태로 인해 ESC가 VM을 시작하지 못하는 경우 Cisco에서는 마스터 ESC를 재부팅하 여 ESC 전환을 수행하는 방법을 권장합니다. ESC 전환에는 약 1분이 소요됩니다. 새 마스터 ESC에서 "health.sh" 스크립트를 실행하여 상태가 up 상태인지 확인합니다. VM을 시작하고 VM 상 태를 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 영향받는 VM을 수동으로 복구해야 합니다.

자동 구축 컨피그레이션 업데이트

• AutoDeploy VM에서 autodeploy.cfg를 편집하고 기존 컴퓨팅 서버를 새 컴퓨팅 서버로 교체합 니다. 그런 다음 conpd_cli에서 replace를 로드합니다. 이 단계는 나중에 성공적인 구축을 비활 성화하는 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

• 컨피그레이션이 변경된 후 uas-confd 및 자동 배포 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

구성 요소 RMA - 컨트롤러 노드

사전 점검

• OSPD에서 컨트롤러에 로그인하고 pc가 양호한 상태인지 확인합니다. 세 컨트롤러 모두 Online 및 Galera에서 세 컨트롤러를 모두 Master로 표시합니다.

✤ 참고: 정상 클러스터에는 2개의 활성 컨트롤러가 필요하므로 남아 있는 2개의 컨트롤러가 온 라인 및 활성 상태인지 확인합니다.

<#root>

[heat-admin@pod1-controller-0 ~]\$ sudo pcs status Cluster name: tripleo_cluster Stack: corosync Current DC: pod1-controller-2 (version 1.1.15-11.el7_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 3 nodes and 22 resources configured

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

```
ip-11.118.0.42 (ocf::heartbeat:IPaddr2):
                                                    Started pod1-controller-1
                                                    Started pod1-controller-2
 ip-11.119.0.47 (ocf::heartbeat:IPaddr2):
 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
my-ipmilan-for-pod1-controller-0
                                        (stonith:fence_ipmilan): Started pod1-controller-0
my-ipmilan-for-pod1-controller-1
                                        (stonith:fence_ipmilan): Started pod1-controller-0
                                       (stonith:fence_ipmilan): Started pod1-controller-0
my-ipmilan-for-pod1-controller-2
Daemon Status:
 corosync: active/enabled
 pacemaker: active/enabled
```

```
pcsd: active/enabled
```

컨트롤러 클러스터를 유지 관리 모드로 이동

• 스탠바이에서 업데이트된 컨트롤러의 pcs 클러스터 사용:

<#root>

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

sudo pcs cluster standby

• pcs 상태를 다시 확인하고 pcs 클러스터가 이 노드에서 중지되었는지 확인합니다.

<#root>

[heat-admin@pod1-controller-0 ~]\$

sudo pcs status

Cluster name: tripleo_cluster Stack: corosync Current DC: pod1-controller-2 (version 1.1.15-11.el7_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 3 nodes and 22 resources configured

Node pod1-controller-0: standby

```
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
                                   (systemd:openstack-cinder-volume):
                                                                                   Started pod1-controller
 openstack-cinder-volume
                                         (stonith:fence_ipmilan): Started pod1-controller-1
(stonith:fence_ipmilan): Started pod1-controller-1
my-ipmilan-for-pod1-controller-0
my-ipmilan-for-pod1-controller-1
my-ipmilan-for-pod1-controller-2
                                         (stonith:fence_ipmilan): Started pod1-controller-2
Daemon Status:
  corosync: active/enabled
  pacemaker: active/enabled
  pcsd: active/enabled
```

또한 다른 2개의 컨트롤러의 pcs 상태는 노드를 대기 상태로 표시해야 합니다.

컨트롤러 노드에서 결함이 있는 구성 요소 교체

지정된 서버의 전원을 끕니다. UCS C240 M4 서버에서 결함이 있는 구성 요소를 교체하는 단계는 다음 위치에서 참조할 수 있습니다.

<u>서버 구성 요소 교체</u>

서버 전원 켜기

• 서버 전원을 켜고 서버가 작동하는지 확인합니다.

<#root>

[stack@tb5-ospd ~]\$ source stackrc [stack@tb5-ospd ~]\$ nova list |grep pod1-controller-0 | 1ca946b8-52e5-4add-b94c-4d4b8a15a975 | pod1-controller-0 |

ACTIVE

| - |

Running

```
| ctlplane=192.200.0.112 |
```

 영향을 받는 컨트롤러에 로그인하고, unstandby를 사용하여 대기 모드를 제거합니다. 컨트롤 러가 클러스터와 함께 온라인 상태로 전환되고 Galera가 세 컨트롤러 모두를 Master로 표시 하는지 확인합니다. 이 작업에는 몇 분 정도 걸릴 수 있습니다.

<#root>

```
[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.el7_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
3 nodes and 22 resources configured
Online: [ podl-controller-0 podl-controller-1 podl-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
                           (ocf::heartbeat:IPaddr2):
                                                               Started pod1-controller-2
 ip-192.200.0.102
 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

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

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

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

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

• ceph와 같은 일부 모니터 서비스의 상태가 정상인지 확인할 수 있습니다.

<#root>

[heat-admin@pod1-controller-0 ~]\$

sudo ceph -s

```
cluster eb2bb192-b1c9-11e6-9205-525400330666
health
```

HEALTH_OK

```
monmap e1: 3 mons at {pod1-controller-0=11.118.0.10:6789/0,pod1-controller-1=11.118.0.11:6789/0,pod
election epoch 70, quorum 0,1,2 pod1-controller-0,pod1-controller-1,pod1-controller-2
osdmap e218: 12 osds: 12 up, 12 in
flags sortbitwise,require_jewel_osds
pgmap v2080888: 704 pgs, 6 pools, 714 GB data, 237 kobjects
2142 GB used, 11251 GB / 13393 GB avail
704 active+clean
client io 11797 kB/s wr, 0 op/s rd, 57 op/s wr
```

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 번 역 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.