

Ultra-M AutoVNF 클러스터 장애에 대한 복구 절차 - vEPC

목차

[소개](#)

[배경 정보](#)

[약어](#)

[MoP의 워크플로](#)

[사례 1. UAS 클러스터의 단일 장애 복구](#)

[상태 확인](#)

[UAS에 연결하려고 할 때 구성된 서버에 연결하지 못했습니다.](#)

[오류 상태에서 UAS 복구](#)

[사례 2. 세 가지 UAS\(AutoVNF\) 모두 오류 상태입니다.](#)

[uas-check.py 스크립트로 UAS 상태 확인](#)

[OpenStack 레벨에서 VM의 상태 확인](#)

[Zookeeper 보기 확인](#)

[AutoVNF 문제 해결 - 프로세스 및 작업](#)

[오류 상태의 여러 UAS 수정](#)

소개

이 문서에서는 StarOS VNF(Virtual Network Functions)를 호스팅하는 Ultra-M 설정에서 UAS(Ultra Automation Services) 또는 AutoVNF 클러스터 장애를 복구하는 데 필요한 단계를 설명합니다.

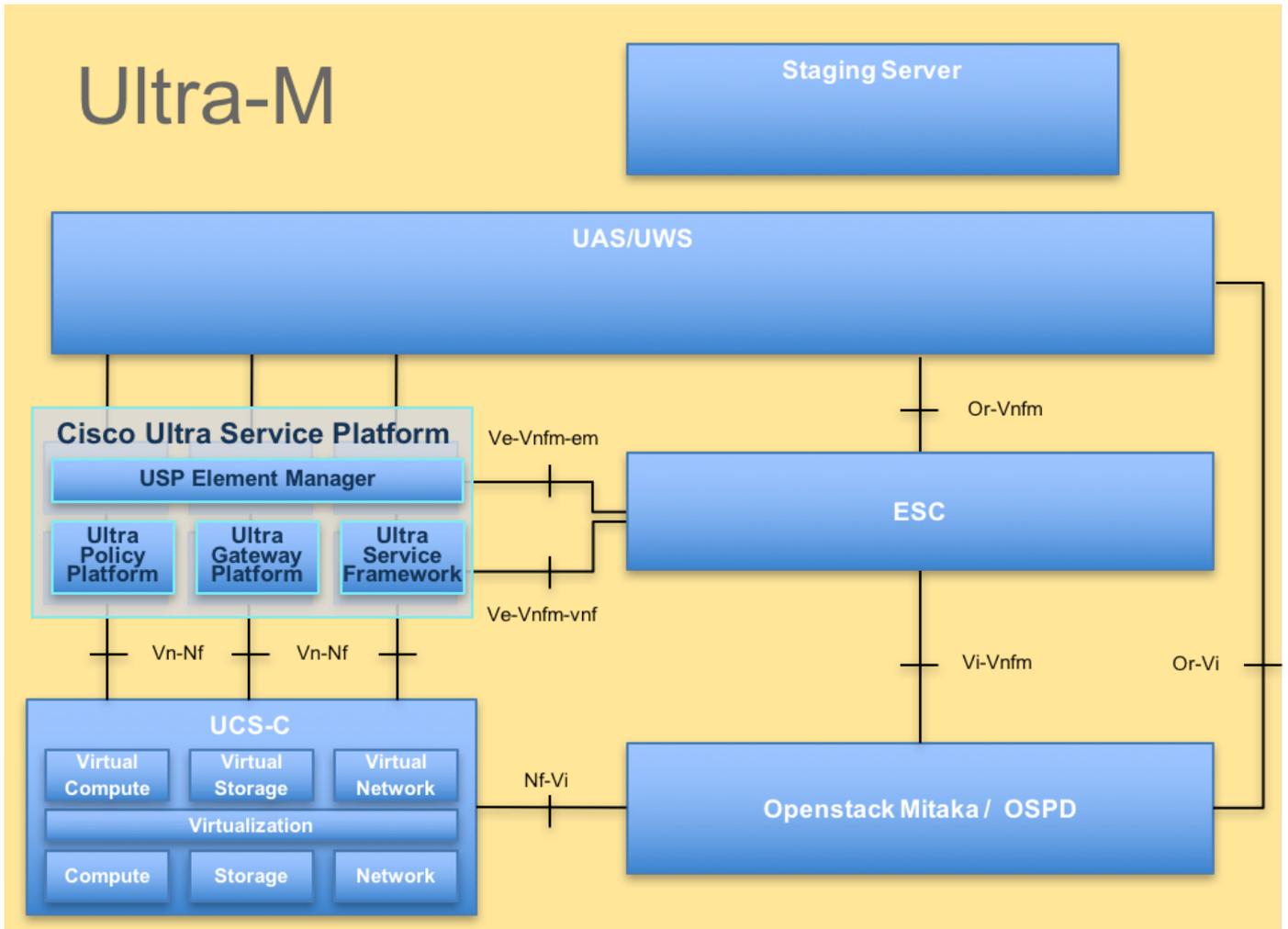
배경 정보

Ultra-M은 VNF의 구축을 간소화하기 위해 설계된, 사전 패키징되고 검증된 가상화된 모바일 패킷 코어 솔루션입니다.

Ultra-M 솔루션은 다음과 같은 VM(가상 머신) 유형으로 구성됩니다.

- 자동 IT
- 자동 구축
- UAS 또는 AutoVNF
- 요소 관리자(EM)
- Elastic Services Controller(ESC)
- 제어 기능(CF)
- 세션 기능(SF)

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



UltraM 아키텍처

이 문서는 Cisco Ultra-M 플랫폼에 대해 잘 알고 있는 Cisco 직원을 대상으로 합니다.

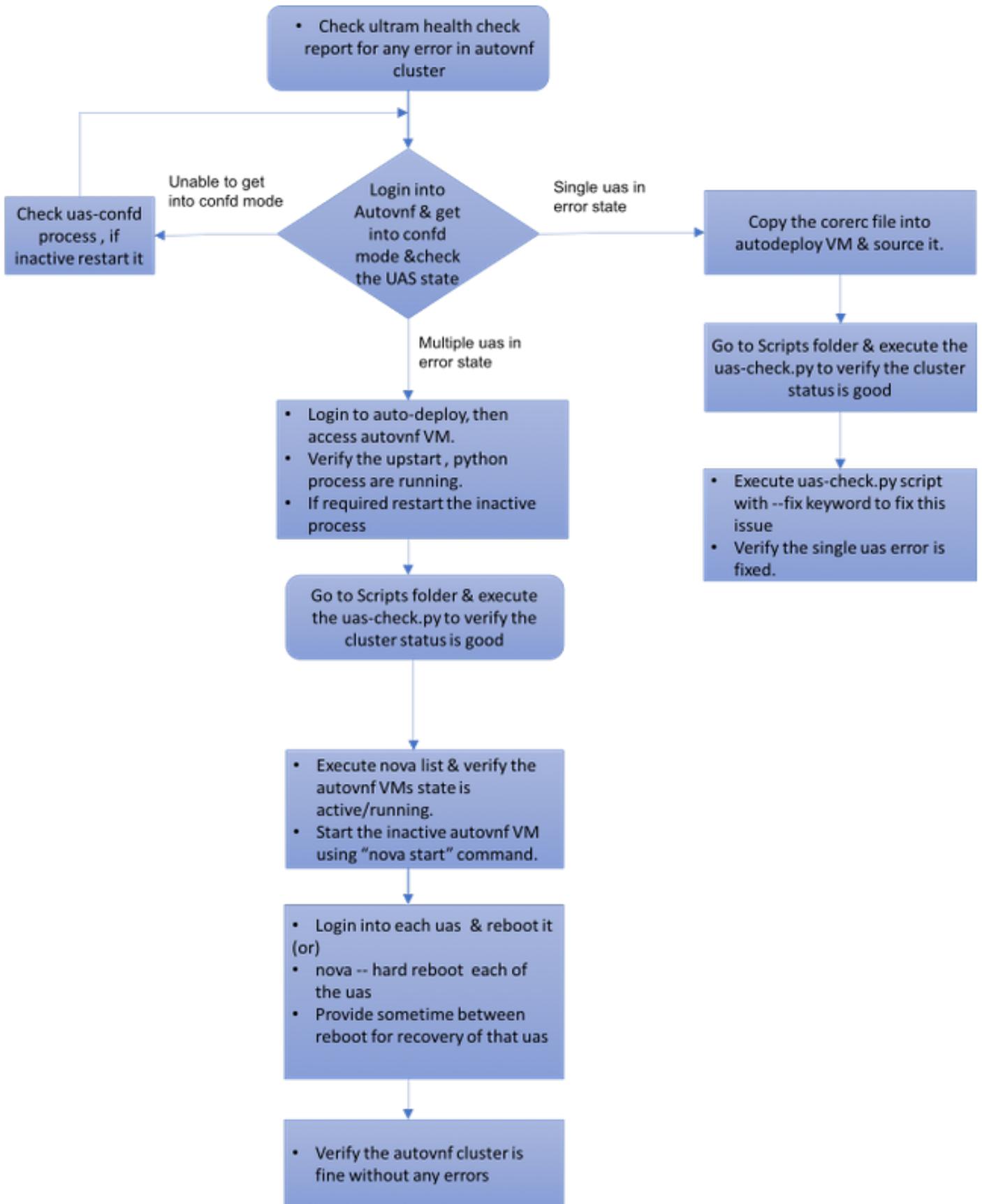
 참고: 이 문서의 절차를 정의하기 위해 Ultra M 5.1.x 릴리스가 고려됩니다.

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



사례 1. UAS 클러스터의 단일 장애 복구

상태 확인

1. Ultra-M Manager는 Ultra-M 노드의 상태를 확인을 수행합니다. 보고서/var/log/cisco/ultram-health/directory 및 grep로 이동하여 UAS 보고서를 확인합니다.

```
<#root>
```

```
[stack@pod1-ospd ultram-health]$ more ultram_health_uas.report
```

```
-----  
VNF ID          | UAS Node | Status  | Error Info, if any  
-----
```

```
172.21.201.122  | autovnf | XXX    |
```

```
AutoVNF Cluster FAILED
```

```
: Node: 172.16.180.12, Status:
```

```
error
```

```
, Role: NA
```

```
172.21.201.122  | vnf-em  | :-)
```

```
172.21.201.122  | esc     | :-)
```

2. UAS 클러스터의 예상 상태는 세 UAS가 모두 활성화된 상태로 표시됩니다.

```
[stack@pod1-ospd ~]# ssh ubuntu@10.1.1.1  
password:
```

```
ubuntu@autovnf1-uas:~$ ncs_cli -u admin -C
```

```
autovnf1-uas-0#show uas
```

```
uas version 1.0.1-1
```

```
uas state ha-active
```

```
uas ha-vip 172.16.181.101
```

```
INSTANCE IP STATE ROLE
```

```
-----  
172.16.180.3 alive CONFD-MASTER
```

```
172.16.180.7 alive CONFD-SLAVE
```

```
172.16.180.12 alive NA
```

UAS에 연결하려고 할 때 구성된 서버에 연결하지 못했습니다.

1. 경우에 따라 구성된 서버에 연결할 수 없습니다.

```
ubuntu@autovnf1-uas-0:/opt/cisco/usp/uas/manager$ confd_cli -u admin -C  
Failed to connect to server
```

2. uas-confd 프로세스의 상태를 확인합니다.

<#root>

```
ubuntu@autovnf1-uas-0:/opt/cisco/usp/uas/manager$ sudo initctl status uas-confd
uas-confd
```

```
stop
```

```
/waiting
```

3. confd 서버가 실행되지 않으면 서비스를 다시 시작합니다.

<#root>

```
ubuntu@autovnf1-uas-0:/opt/cisco/usp/uas/manager$ sudo initctl
```

```
start
```

```
uas-confd
uas-confd start/running, process 7970
ubuntu@autovnf1-uas-0:/opt/cisco/usp/uas/manager$ confd_cli -u admin -C
```

```
Welcome to the ConfD CLI
```

```
admin connected from 172.16.180.9 using ssh on autovnf1-uas-0
```

오류 상태에서 UAS 복구

1. 클러스터 중 하나의 AutoVNF에 장애가 발생하는 경우 UAS 클러스터는 UAS 중 하나를 오류 상태로 표시합니다.

```
[stack@pod1-ospd ~]# ssh ubuntu@10.1.1.1
password:
```

```
ubuntu@autovnf1-uas:~$ ncs_cli -u admin -C
```

```
autovnf1-uas-0#show uas
uas version 1.0.1-1
uas state ha-active
uas ha-vip 172.16.181.101
INSTANCE IP STATE ROLE
-----
172.16.180.3 alive CONFD-MASTER
172.16.180.7 alive CONFD-SLAVE
172.16.180.12 alive error
```

2. OSPD 서버의 /home/stack에서 core 파일(VNF의 rc 파일)을 AutoDeploy에 복사하여 소싱합니다

3. uas-check.py 스크립트를 사용하여 UAS/AutoVNF의 상태를 확인합니다. autovnf1은 AutoVNF

이름입니다.

<#root>

```
ubuntu@auto-deploy-iso-590-uas-0:~$
```

```
/opt/cisco/usp/apps/auto-it/scripts/uas-check.py auto-vnf autovnf1
```

```
2017-11-17 14:52:20,186 - INFO: Check of AutoVNF cluster started
2017-11-17 14:52:22,172 - INFO: Found 2 AutoVNF instance(s), 3 expected
2017-11-17 14:52:22,172 - INFO: Instance 'autovnf1-uas-2' is missing
2017-11-17 14:52:22,172 - INFO: Check completed, AutoVNF cluster has recoverable errors
```

4. uas-check.py 스크립트 및 add—fix 키워드를 사용하여 UAS를 복구합니다.

```
ubuntu@auto-deploy-iso-590-uas-0:~$ /opt/cisco/usp/apps/auto-it/scripts/uas-check.py auto-vnf autovnf1
2017-11-17 14:52:27,493 - INFO: Check of AutoVNF cluster started
2017-11-17 14:52:29,215 - INFO: Found 2 AutoVNF instance(s), 3 expected
2017-11-17 14:52:29,215 - INFO: Instance 'autovnf1-uas-2' is missing
2017-11-17 14:52:29,215 - INFO: Check completed, AutoVNF cluster has recoverable errors
2017-11-17 14:52:29,386 - INFO: Creating instance 'autovnf1-uas-2' and attaching volume 'autovnf1-uas-v
2017-11-17 14:52:47,600 - INFO: Created instance 'autovnf1-uas-2'
```

5. 새로 생성된 UAS가 활성 상태이며 클러스터의 일부입니다.

<#root>

```
autovnf1-uas-0#show uas
uas version 1.0.1-1
uas state ha-active
uas ha-vip 172.16.181.101
INSTANCE IP STATE ROLE
-----
172.16.180.3 alive CONFD-MASTER
172.16.180.7 alive CONFD-SLAVE
172.16.180.13 alive NA
```

사례 2. 세 가지 UAS(AutoVNF) 모두 오류 상태입니다.

1. Ultra-M Manager는 Ultra-M 노드의 상태 확인을 수행합니다.

<#root>

```
[stack@pod1-ospd ultram-health]$ more ultram_health_uas.report
```

```
-----  
VNF ID          | UAS Node | Status  | Error Info, if any  
-----
```

```
172.21.201.122 | autovnf | XXX     |
```

AutoVNF Cluster FAILED

: Node: 172.16.180.12, Status:

error

, Role: NA, Node: 172.16.180.9, Status:

error

, Role: NA, Node: 172.16.180.10, Status:

error

, Role: NA

```
172.21.201.122 | vnf-em  | :-)     |
```

```
172.21.201.122 | esc     | :-)     |  
-----
```

2. 출력에서 관찰한 것처럼 Ultra-M manager는 AutoVNF에 오류가 있음을 보고하고 클러스터의 3개 UAS가 모두 오류 상태를 표시합니다.

uas-check.py 스크립트로 UAS 상태 확인

1. Auto-Deploy에 로그인하여 AutoVNF UAS에 액세스하여 상태를 확인할 수 있는지 확인합니다.

```
ubuntu@auto-deploy-iso-590-uas-0:~$ /opt/cisco/usp/apps/auto-it/scripts$ ./uas-check.py auto-vnf autovnf
```

```
2017-12-05 11:41:09,834 - INFO: Check of AutoVNF cluster started
```

```
2017-12-05 11:41:11,342 - INFO: Found 3 ACTIVE AutoVNF instances
```

```
2017-12-05 11:41:11,343 - INFO: Check completed, AutoVNF cluster is fine
```

2. Auto-Deploy, Secure Shell(SSH)에서 AutoVNF 노드로 이동한 다음 confd 모드로 들어갑니다. show uas를 사용하여 상태를 확인합니다.

```
ubuntu@auto-deploy-iso-590-uas-0:~$ ssh ubuntu@172.16.180.9
```

```
password:
```

```
autovnf1-uas-1#show uas
```

```
uas version 1.0.1-1
```

```
uas state ha-active
```

```
uas ha-vip 172.16.181.101
```

```
INSTANCE IP    STATE  ROLE
```

```
-----  
172.16.180.9   error  NA  
172.16.180.10 error  NA  
172.16.180.12 error  NA
```

3. 세 UAS 노드 모두에서 상태를 확인하는 것이 좋습니다.

OpenStack 레벨에서 VM의 상태 확인

nova 목록에서 AutoVNF VM의 상태를 확인합니다. 필요한 경우 nova start를 수행하여 VM 차단을 시작합니다.

```
[stack@pod1-ospd ultram-health]$ nova list | grep autovnf  
| 83870eed-b4e9-47b3-976d-cc3eddecf866 | autovnf1-uas-0  
| 201d9ce5-538c-42f7-a46c-fc8cdef1eabf | autovnf1-uas-1  
| 6c6d25cd-21b6-42b9-87ff-286220faa2ff | autovnf1-uas-2
```

Zookeeper 보기 확인

1. 리더로서의 모드를 확인하기 위해 사육사의 상태를 확인합니다.

```
ubuntu@autovnf1-uas-0:/var/log/upstart$ /opt/cisco/usp/packages/zookeeper/current/bin/zkServer.sh status  
ZooKeeper JMX enabled by default  
Using config: /opt/cisco/usp/packages/zookeeper/current/bin/../conf/zoo.cfg  
Mode: leader
```

2. 동물원 사육사는 정상적으로 일어나 있을 것.

AutoVNF 문제 해결 - 프로세스 및 작업

1. 노드의 오류 상태에 대한 이유를 식별합니다. AutoVNF를 실행하려면 다음과 같이 실행 및 실행해야 하는 프로세스 집합이 있습니다.

AutoVNF

uws-ae

uas-confd

cluster_manager

uas_manager

```
ubuntu@autovnf1-uas-0:~$ sudo initctl list | grep uas
```

uas-confd stop/waiting ===> this is not good, the uas-confd process is not running

uas_manager start/running, process 2143

<#root>

```
root@autovnf1-uas-1:/home/ubuntu# sudo initctl list
```

....

uas-confd start/running, process 1780

....

autovnf start/running, process 1908

....

....

uws-ae start/running, process 1909

....

....

cluster_manager start/running, process 1827

....

.....

uas_manager start/running, process 1697

.....

.....

2. 다음 python 프로세스가 실행 중인지 확인합니다.

uas_manager.py

cluster_manager.py

usp_autovnf.py

```
root@autovnf1-uas-1:/home/ubuntu# ps -aef | grep pyth
root      1819   1697   0 Jun13 ?        00:00:50 python /opt/cisco/usp/uas/manager/uas_manager.py
root      1858   1827   0 Jun13 ?        00:09:21 python /opt/cisco/usp/uas/manager/cluster_manager.py
root      1908     1   0 Jun13 ?        00:01:00 python /opt/cisco/usp/uas/autovnf/usp_autovnf.py
root      25662 24750   0 13:16 pts/7    00:00:00 grep --color=auto pyth
```

3. 필요한 프로세스 중 하나가 시작/실행 상태가 아닌 경우 프로세스를 다시 시작하고 상태를 확인합니다. 오류 상태로 계속 표시되는 경우 다음 섹션에서 설명한 절차를 수행하여 이 문제를 해결하십시오.

오류 상태의 여러 UAS 수정

1. nova —OSPD에서 <vm 이름>을(를) 하드 리부팅합니다. 다음 UAS로 진행하기 전에 이 VM을 복구할 수 있는 시간을 주십시오. 모든 UAS VM에서 이를 수행합니다.

또는

2. 각 UAS에 로그인하고 sudo reboot를 사용합니다. 복구를 기다린 다음 다른 UAS VM으로 진행합니다.

트랜잭션 로그의 경우 다음을 확인하십시오.

```
/var/log/upstart/autovnf.log
```

```
show logs xxx | display xml
```

이렇게 하면 문제가 해결되고 오류 상태에서 UAS가 복구됩니다.

1. ultram_health_check 보고서 사용과 동일하게 확인합니다.

```
[stack@pod1-ospd ultram-health]$ more ultram_health_uas.report
```

```
-----
```

VNF ID	UAS Node	Status	Error Info, if any
172.21.201.122	autovnf	:~)	
172.21.201.122	vnf-em	:~)	
172.21.201.122	esc	:~)	

```
-----
```

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.