

다양한 Ultra-M 구성 요소의 백업 및 복원 절차

목차

[소개](#)

[배경 정보](#)

[약어](#)

[백업 절차](#)

[OSPD 백업](#)

[백업 자동 구축](#)

[AutoIT-VNF 백업](#)

[AutoVNF 백업](#)

[ESC 백업](#)

[EM 백업](#)

[StarOS 백업](#)

[복원 절차](#)

[OSPD 복구](#)

[복구 자동 구축](#)

[AutoIT-VNF 복구](#)

[AutoVNF 복구](#)

[ESC 복구](#)

[EM 복구](#)

[StarOS 복구](#)

소개

이 문서에서는 StarOS VNF(Virtual Network Functions)를 호스팅하는 Ultra-M 설정에서 VM(Virtual Machine)을 백업 및 복원하는 데 필요한 단계에 대해 설명합니다.

배경 정보

Ultra-M은 VNF 구축을 간소화하기 위해 사전 패키지 및 검증된 가상화 모바일 패킷 코어 솔루션입니다. Ultra-M 솔루션은 다음과 같은 VM(가상 머신) 유형으로 구성됩니다.

- 자동 IT
- 자동 구축
- UAS(Ultra Automation Services)
- 요소 관리자(EM)
- Elastic Services Controller(ESC)
- 제어 기능(CF)
- 세션 함수(SF)

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



이 문서는 Cisco Ultra-M 플랫폼에 익숙한 Cisco 직원을 대상으로 합니다.

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	보편적으로 고유한 Identifier

백업 절차

OSPD 백업

1. OpenStack 스택 상태 및 노드 목록을 확인합니다.

```
[stack@director ~]$ source stackrc
[stack@director ~]$ openstack stack list --nested
[stack@director ~]$ ironic node-list
[stack@director ~]$ nova list
```

2. 모든 언더클라우드 서비스가 OSP-D 노드에서 로드, 활성 및 실행 상태인지 확인합니다.

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

UNIT	LOAD	ACTIVE	SUB	DESCRIPTION
neutron-dhcp-agent.service	loaded	active	running	OpenStack Neutron DHCP Agent
neutron-openvswitch-agent.service	loaded	active	running	OpenStack Neutron Open vSwitch Agent
neutron-ovs-cleanup.service	loaded	active	exited	OpenStack Neutron Open vSwitch Cleanup Utility
neutron-server.service	loaded	active	running	OpenStack Neutron Server
openstack-aodh-evaluator.service	loaded	active	running	OpenStack Alarm evaluator service
openstack-aodh-listener.service	loaded	active	running	OpenStack Alarm listener service
openstack-aodh-notifier.service	loaded	active	running	OpenStack Alarm notifier service
openstack-ceilometer-central.service	loaded	active	running	OpenStack ceilometer central agent
openstack-ceilometer-collector.service	loaded	active	running	OpenStack ceilometer collection service
openstack-ceilometer-notification.service	loaded	active	running	OpenStack ceilometer notification agent
openstack-glance-api.service	loaded	active	running	OpenStack Image Service (code-named Glance) API server
openstack-glance-registry.service	loaded	active	running	OpenStack Image Service (code-named Glance) Registry server
openstack-heat-api-cfn.service	loaded	active	running	Openstack Heat CFN-compatible API Service
openstack-heat-api.service	loaded	active	running	OpenStack Heat API Service
openstack-heat-engine.service	loaded	active	running	Openstack Heat Engine Service
openstack-ironic-api.service	loaded	active	running	OpenStack Ironic API service
openstack-ironic-conductor.service	loaded	active	running	OpenStack Ironic Conductor service
openstack-ironic-inspector-dnsmasq.service	loaded	active	running	PXE boot dnsmasq service for Ironic Inspector
openstack-ironic-inspector.service	loaded	active	running	Hardware introspection service for OpenStack Ironic
openstack-mistral-api.service	loaded	active	running	Mistral API Server
openstack-mistral-engine.service	loaded	active	running	Mistral Engine Server
openstack-mistral-executor.service	loaded	active	running	Mistral Executor Server
openstack-nova-api.service	loaded	active	running	OpenStack Nova API Server
openstack-nova-cert.service	loaded	active	running	OpenStack Nova Cert Server
openstack-nova-compute.service	loaded	active	running	OpenStack Nova Compute Server
openstack-nova-conductor.service	loaded	active	running	OpenStack Nova Conductor Server
openstack-nova-scheduler.service	loaded	active	running	OpenStack Nova Scheduler Server
openstack-swift-account-reaper.service	loaded	active	running	OpenStack Object Storage (swift) - Account Reaper
openstack-swift-account.service	loaded	active	running	OpenStack Object Storage (swift) - Account Server
openstack-swift-container-updater.service	loaded	active	running	OpenStack Object Storage (swift) - Container Updater
openstack-swift-container.service	loaded	active	running	OpenStack Object Storage

```

(swift) - Container Server
openstack-swift-object-updater.service    loaded active running OpenStack Object Storage
(swift) - Object Updater
openstack-swift-object.service           loaded active running OpenStack Object Storage
(swift) - Object Server
openstack-swift-proxy.service            loaded active running OpenStack Object Storage
(swift) - Proxy Server
openstack-zaqar.service                   loaded active running OpenStack Message Queuing
Service (code-named Zaqar) Server
openstack-zaqar@1.service                 loaded active running OpenStack Message Queuing
Service (code-named Zaqar) Server Instance 1
openvswitch.service                       loaded active exited  Open vSwitch

```

```

LOAD    = Reflects whether the unit definition was properly loaded.
ACTIVE  = The high-level unit activation state, i.e. generalization of SUB.
SUB     = The low-level unit activation state, values depend on unit type.

```

```

37 loaded units listed. Pass --all to see loaded but inactive units, too.
To show all installed unit files use 'systemctl list-unit-files'.

```

3. 백업 프로세스를 수행하기 전에 사용 가능한 디스크 공간이 충분한지 확인합니다. 이 대상 용량은 3.5GB 이상이어야 합니다.

```
[stack@director ~]$df -h
```

4. Undercloud 노드의 데이터를 `under-cloud-backup-[timestamp].tar.gz`라는 파일로 백업하고 백업 서버로 전송하기 위해 이러한 명령을 루트 사용자로 실행합니다.

```

[root@director ~]# mysqldump --opt --all-databases > /root/undercloud-all-databases.sql
[root@director ~]# tar --xattrs -czf undercloud-backup-`date +%F`.tar.gz /root/undercloud-all-
databases.sql
/etc/my.cnf.d/server.cnf /var/lib/glance/images /srv/node /home/stack
tar: Removing leading `/' from member names

```

백업 자동 구축

1. 자동 배포를 사용하려면 다음 데이터를 백업해야 합니다.

- CDB(구성 데이터베이스 자동 구축)
- 구성 자동 배포
- 로그 자동 구축
- Syslog 컨피그레이션

2. 활성화/비활성화 후 AutoDeploy Confd CDB 데이터의 백업 및 실행 구성이 필요하며, 데이터가 백업 서버로 전송되도록 합니다.

3. 자동 배포가 독립형 모드로 실행되며 이 데이터가 손실되면 배포를 정상적으로 비활성화할 수 없습니다. 따라서 컨피그레이션 및 CDB 데이터의 백업을 수행해야 합니다.

```

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:/home/ubuntu# service autodeploy status
autodeploy start/running, process 1313

```

```

root@auto-deploy:/home/ubuntu# service autodeploy stop
autodeploy 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
```

4. **autodeploy_cdb_backup.tar**를 백업 서버에 복사합니다.

5. 자동 구축에서 실행 중인 컨피그레이션의 백업을 수행한 다음 백업 서버로 전송합니다.

```
root@auto-deploy:/home/ubuntu# confd_cli -u admin -C
```

```
Welcome to the ConfD CLI
```

```
admin connected from 127.0.0.1 using console on auto-deploy
```

```
auto-deploy#show running-config | save backup-config-$(date +%Y%m%d).cfg
Replace the $(date +%Y%m%d) to appropriate date and POD reference
auto-deploy#
```

6. **AutoDeploy Config** 서비스를 시작합니다.

```
root@auto-deploy-iso-2007-uas-0:~# service uas-confd start
uas-confd start/running, process 13852
```

```
root@auto-deploy:/home/ubuntu# service autodeploy start
autodeploy start/running, process 8835
```

7. 스크립트 디렉토리로 이동하여 **AutoDeploy VM**에서 로그를 수집합니다.

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

8. 로그를 수집하려면 **collect-uas-logs.sh** 스크립트를 실행합니다.

```
sudo ./collect-uas-logs.sh
```

9. 자동 구축에서 ISO 이미지 백업을 가져와 백업 서버로 전송합니다.

```
root@POD1-5-1-7-2034-auto-deploy-uas-0:/home/ubuntu# /home/ubuntu/isos
root@POD1-5-1-7-2034-auto-deploy-uas-0:/home/ubuntu/isos# ll
```

```
total 4430888
drwxr-xr-x 2 root root 4096 Dec 20 01:17 ./
drwxr-xr-x 5 ubuntu ubuntu 4096 Dec 20 02:31 ../
-rwxr-xr-x 1 ubuntu ubuntu 4537214976 Oct 12 03:34 usp-5_1_7-2034.iso*
```

10. **syslog** 컨피그레이션을 수집하고 백업 서버에 저장합니다.

```
ubuntu@auto-deploy-vnf-iso-5-1-5-1196-uas-0:~$sudo su
root@auto-deploy-vnf-iso-5-1-5-1196-uas-0:/home/ubuntu#ls /etc/rsyslog.d/00-autodeploy.conf
00-autodeploy.conf
root@auto-deploy-vnf-iso-5-1-5-1196-uas-0:/home/ubuntu#ls /etc/rsyslog.conf
rsyslog.conf
```

AutoIT-VNF 백업

AutoIT-VNF는 스테이트리스 VM이므로 백업할 데이터베이스(DB)가 없습니다. AutoIT-VNF는 Ultra-M용 컨피그레이션 관리 리포지토리와 함께 패키지 관리를 담당하므로 이러한 백업을 반드시 수행해야 합니다.

1. day-0 StarOS 컨피그레이션을 백업하고 백업 서버로 전송합니다.

```
root@auto-it-vnf-iso-5-8-uas-0:/home/ubuntu# cd /opt/cisco/usp/uploads/
root@auto-it-vnf-iso-5-8-uas-0:/opt/cisco/usp/uploads# ll
total 12
drwxrwxr-x 2 uspadmin usp-data 4096 Nov  8 23:28 ./
drwxr-xr-x 15 root      root    4096 Nov  8 23:53 ../
-rw-rw-r-- 1 ubuntu    ubuntu   985 Nov  8 23:28 system.cfg
```

2. 스크립트 디렉토리로 이동하여 AutoIT VM에서 로그를 수집합니다.

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

3. 로그를 수집하기 위해 **collect-uas-logs.sh** 스크립트를 실행합니다.

```
sudo ./collect-uas-logs.sh
```

4. syslog 구성 백업을 수집하고 백업 서버에 저장합니다.

```
ubuntu@auto-it-vnf-iso-5-1-5-1196-uas-0:~$sudo su
```

```
root@auto-it-vnf-iso-5-1-5-1196-uas-0:/home/ubuntu#ls /etc/rsyslog.d/00-autoit-vnf.conf
```

```
00-autoit-vnf.conf
```

```
root@auto-it-vnf-iso-5-1-5-1196-uas-0:ls /etc/rsyslog.conf
```

```
rsyslog.conf
```

AutoVNF 백업

AutoVNF는 개별 VNF 및 VNF를 가동할 책임이 있습니다. AutoDeploy는 VNF 및 VNF를 모두 인스턴스화하는 데 필요한 컨피그레이션을 AutoVNF로 전송하고 AutoVNF는 이 작업을 수행합니다. VNF를 활성화하기 위해 AutoVNF는 VIM/OpenStack과 직접 통신하며 VNF가 작동하면 AutoVNF는 VNF를 사용하여 VNF를 표시합니다.

AutoVNF에는 1:N개의 리던던시가 있으며 Ultra-M 설정에는 3개의 AutoVNF VM이 실행되고 있습니다. 단일 AutoVNF 장애가 Ultra-M에서 지원되므로 복구가 가능합니다.

참고: 하나 이상의 장애가 있는 경우 지원되지 않으며 시스템 재구축이 필요할 수 있습니다.

AutoVNF 백업 세부 정보:

- 실행 중인 컨피그레이션
- ConfD CDB DB

- AutoVNF 로그(AutoVNF의 각 인스턴스에서)
- Syslog 컨피그레이션

지정된 사이트에서 활성화/비활성화하고 백업 서버에 업로드하기 전에 백업을 수행하는 것이 좋습니다.

1. 마스터 AutoVNF에 로그인하여 마스터 마스터인지 확인합니다.

```
root@auto-testautovnf1-uas-1:/home/ubuntu# confd_cli -u admin -C

Welcome to the ConfD CLI
admin connected from 127.0.0.1 using console on auto-testautovnf1-uas-1
auto-testautovnf1-uas-1#show uas
uas version 1.0.1-1
uas state ha-active
uas ha-vip 172.57.11.101

INSTANCE IP      STATE  ROLE
-----
172.57.12.6     alive  CONFID-SLAVE
172.57.12.7     alive  CONFID-MASTER
172.57.12.13    alive  NA

auto-testautovnf1-uas-1#exit

root@auto-testautovnf1-uas-1:/home/ubuntu# ip a

1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever

2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether fa:16:3e:c7:dc:89 brd ff:ff:ff:ff:ff:ff
    inet 172.57.12.7/24 brd 172.57.12.255 scope global eth0
        valid_lft forever preferred_lft forever
    inet6 fe80::f816:3eff:fec7:dc89/64 scope link
        valid_lft forever preferred_lft forever

3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether fa:16:3e:10:29:1b brd ff:ff:ff:ff:ff:ff
    inet 172.57.11.101/24 brd 172.57.11.255 scope global eth1
        valid_lft forever preferred_lft forever
    inet6 fe80::f816:3eff:fe10:291b/64 scope link
        valid_lft forever preferred_lft forever
```

2. 실행 중인 컨피그레이션의 백업을 수행하고 파일을 백업 서버로 전송합니다.

```
root@auto-testautovnf1-uas-1:/home/ubuntu# confd_cli -u admin -C

Welcome to the ConfD CLI
admin connected from 127.0.0.1 using console on auto-testautovnf1-uas-1
auto-testautovnf1-uas-1#show running-config | save running-autovnf-12202017.cfg
auto-testautovnf1-uas-1#exit

root@auto-testautovnf1-uas-1:/home/ubuntu# ll running-autovnf-12202017.cfg
-rw-r--r-- 1 root  root  18181 Dec 20 19:03 running-autovnf-12202017.cfg
```

3. CDB를 백업하고 파일을 백업 서버로 전송합니다.

```
root@auto-testautovnf1-uas-1:/opt/cisco/usp/uas/confd-6.3.1/var/confd# tar cvf
autovnf_cdb_backup.tar cdb/
cdb/
cdb/O.cdb
cdb/C.cdb
cdb/aaa_init.xml
cdb/vpc.xml
cdb/A.cdb
cdb/gilan.xml
```

```
root@auto-testautovnf1-uas-1:/opt/cisco/usp/uas/confd-6.3.1/var/confd#
root@auto-testautovnf1-uas-1:/opt/cisco/usp/uas/confd-6.3.1/var/confd# ll autovnf_cdb_backup.tar
-rw-r--r-- 1 root root 1198080 Dec 20 19:08 autovnf_cdb_backup.tar
```

4. 스크립트 디렉토리로 이동하여 로그를 수집하고 백업 서버로 전송합니다.

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

```
sudo ./collect-uas-logs.sh
```

5. AutoVNF의 대기 인스턴스에 로그인하고 로그를 수집하고 백업 서버로 전송하기 위해 다음 단계를 수행합니다.

6. 마스터 및 대기 AutoVNF VM에서 syslog 컨피그레이션을 백업하고 백업 서버로 전송합니다.

```
ubuntu@auto-testautovnf1-uas-1:~$sudo su
root@auto-testautovnf1-uas-1:/home/ubuntu#ls /etc/rsyslog.d/00-autovnf.conf
00-autovnf.conf
```

```
root@auto-testautovnf1-uas-1:/home/ubuntu#ls /etc/rsyslog.conf
rsyslog.conf
```

ESC 백업

1. AutoVNF는 VIM과 직접 상호 작용하여 Ultra-M 솔루션에서 ESC를 가동하기 위한 책임이 있습니다. AutoVNF/EM은 VNF 특정 컨피그레이션을 ESC로 전달하고 ESC는 VIM에 상호 작용하여 VNF를 표시합니다.

2. ESC는 Ultra-M 솔루션에서 1:1 이중화를 가집니다. 두 개의 ESC VM이 구축되어 Ultra-M에서 단일 장애를 지원합니다. 즉, 시스템에 단일 오류가 있을 경우 시스템을 복구할 수 있습니다.

참고: 하나 이상의 장애가 있는 경우 지원되지 않으며 시스템 재구축이 필요할 수 있습니다.

ESC 백업 세부 정보:

- 실행 중인 컨피그레이션
- ConfD CDB DB
- ESC 로그
- Syslog 컨피그레이션

3. ESC DB 백업 빈도는 까다롭기 때문에 ESC가 구축된 다양한 VNF VM에 대해 다양한 상태 시스

템을 모니터링하고 유지 관리할 때 주의해서 처리해야 합니다. 이러한 백업은 지정된 VNF/POD/사이트의 작업을 수행한 후에 수행하는 것이 좋습니다.

4. ESC의 상태가 health.sh 스크립트를 사용하는 데 적합한지 확인합니다.

```
[root@auto-test-vnfm1-esc-0 admin]# escadm status
0 ESC status=0 ESC Master Healthy

[root@auto-test-vnfm1-esc-0 admin]# health.sh
esc ui is disabled -- skipping status check
esc_monitor start/running, process 836
esc_mona is up and running ...
vimmanager start/running, process 2741
vimmanager start/running, process 2741
esc_confd is started
tomcat6 (pid 2907) is running... [ OK ]
postgresql-9.4 (pid 2660) is running...
ESC service is running...
Active VIM = OPENSTACK
ESC Operation Mode=OPERATION

/opt/cisco/esc/esc_database is a mountpoint

===== ESC HA (MASTER) with DRBD =====

DRBD_ROLE_CHECK=0
MNT_ESC_DATABASE_CHECK=0
VIMMANAGER_RET=0
ESC_CHECK=0
STORAGE_CHECK=0
ESC_SERVICE_RET=0
MONA_RET=0
ESC_MONITOR_RET=0

=====

ESC HEALTH PASSED
```

5. 실행 중인 컨피그레이션의 백업을 수행하고 파일을 백업 서버로 전송합니다.

```
[root@auto-test-vnfm1-esc-0 admin]# /opt/cisco/esc/confd/bin/confd_cli -u admin -C
admin connected from 127.0.0.1 using console on auto-test-vnfm1-esc-0.novalocal
auto-test-vnfm1-esc-0# show running-config | save /tmp/running-esc-12202017.cfg
auto-test-vnfm1-esc-0#exit

[root@auto-test-vnfm1-esc-0 admin]# ll /tmp/running-esc-12202017.cfg
-rw-----. 1 tomcat tomcat 25569 Dec 20 21:37 /tmp/running-esc-12202017.cfg
```

데이터베이스 백업

1. ESC를 유지 관리 모드로 설정합니다.

2. 백업을 수행하기 전에 ESC VM에 로그인하고 이 명령을 실행합니다.

```
[admin@auto-test-vnfm1-esc-0 admin]# sudo bash
[root@auto-test-vnfm1-esc-0 admin]# cp /opt/cisco/esc/esc-scripts/esc_dbtool.py
/opt/cisco/esc/esc-scripts/esc_dbtool.py.bkup
[root@auto-test-vnfm1-esc-0 admin]# sudo sed -i "s,'pg_dump','/usr/pgsql-9.4/bin/pg_dump,"
```

```
/opt/cisco/esc/esc-scripts/esc_dbtool.py
```

```
#Set ESC to mainenance mode
```

```
[root@auto-test-vnfm1-esc-0 admin]# escadm op_mode set --mode=maintenance
```

3. ESC 모드를 선택하고 유지 관리 모드에 있는지 확인합니다.

```
[root@auto-test-vnfm1-esc-0 admin]# escadm op_mode show
```

4. ESC에서 DB 백업 복원 도구를 사용하여 DB를 백업합니다.

```
[root@auto-test-vnfm1-esc-0 admin]# sudo /opt/cisco/esc/esc-scripts/esc_dbtool.py backup --file  
scp://<username>:<password>@<backup_vm_ip>:<filename>
```

5. ESC를 다시 작동 모드로 설정하고 모드를 확인합니다.

```
[root@auto-test-vnfm1-esc-0 admin]# escadm op_mode set --mode=operation
```

```
[root@auto-test-vnfm1-esc-0 admin]# escadm op_mode show
```

6. 스크립트 디렉토리로 이동하여 로그를 수집합니다.

```
[root@auto-test-vnfm1-esc-0 admin]# /opt/cisco/esc/esc-scripts
```

```
sudo ./collect_esc_log.sh
```

7. 대기 ESC VM에서 동일한 절차를 반복하고 로그를 백업 서버로 전송합니다.

8. ESC VMS 양쪽에서 syslog 구성 백업을 수집하고 이를 백업 서버로 전송합니다.

```
[admin@auto-test-vnfm2-esc-1 ~]$ cd /etc/rsyslog.d
```

```
[admin@auto-test-vnfm2-esc-1 rsyslog.d]$ls /etc/rsyslog.d/00-escmanager.conf  
00-escmanager.conf
```

```
[admin@auto-test-vnfm2-esc-1 rsyslog.d]$ls /etc/rsyslog.d/01-messages.conf  
01-messages.conf
```

```
[admin@auto-test-vnfm2-esc-1 rsyslog.d]$ls /etc/rsyslog.d/02-mona.conf  
02-mona.conf
```

```
[admin@auto-test-vnfm2-esc-1 rsyslog.d]$ls /etc/rsyslog.conf  
rsyslog.conf
```

EM 백업

1. VNF/ESC가 작동되면 AutoVNF는 ESC를 사용하여 EM 클러스터를 가져옵니다. EM 클러스터가 작동하면 EM은 ESC와 상호 작용하여 VNF(VPC/StarOS)를 표시합니다.

2. EM은 Ultra-M 솔루션에서 1:N의 이중화를 지원합니다. 3개의 EM VM으로 구성된 클러스터와 Ultra-M은 단일 VM 장애 복구를 지원합니다.

참고: 하나 이상의 장애가 있는 경우 지원되지 않으며 시스템 재구축이 필요할 수 있습니다.

EM 백업 세부 정보:

- 실행 중인 컨피그레이션
- NCS DB
- EM 로그(NCS 액티브/스탠바이 VM 모두에서)
- Syslog 컨피그레이션

3. EM DB 백업 빈도는 까다롭기 때문에 ESC가 구축된 다양한 VNF VM에 대해 다양한 상태 시스템을 모니터링하고 유지 관리할 때 신중하게 처리해야 합니다. 지정된 VNF/POD/사이트의 작업을 수행한 후에 이러한 백업을 수행하는 것이 좋습니다.

4. EM에서 실행 중인 구성을 백업하고 파일을 백업 서버로 전송합니다.

```
ubuntu@vnfd1deploymentem-0:~$ sudo -i
root@vnfd1deploymentem-0:~# ncs_cli -u admin -C

admin connected from 127.0.0.1 using console on vnfd1deploymentem-0
admin@scm# show running-config | save em-running-12202017.cfg

root@vnfd1deploymentem-0:~# ll em-running-12202017.cfg
-rw-r--r-- 1 root root 19957 Dec 20 23:01 em-running-12202017.cfg
```

5. EM NCS DB 백업을 수행하고 파일을 백업 서버로 전송합니다.

```
ubuntu@vnfd1deploymentem-0:~$ sudo -i
root@vnfd1deploymentem-0:~# cd /opt/cisco/em/git/em-scm/ncs-cdb
root@vnfd1deploymentem-0:/opt/cisco/em/git/em-scm/ncs-cdb# ll

total 472716
drwxrwxr-x 2 root root      4096 Dec 20 02:53 ./
drwxr-xr-x 9 root root      4096 Dec 20 19:22 ../
-rw-r--r-- 1 root root        770 Dec 20 02:48 aaa_users.xml
-rw-r--r-- 1 root root     70447 Dec 20 02:53 A.cdb
-rw-r--r-- 1 root root 483927031 Dec 20 02:48 C.cdb
-rw-rw-r-- 1 root root         47 Jul 27 05:53 .gitignore
-rw-rw-r-- 1 root root        332 Jul 27 05:53 global-settings.xml
-rw-rw-r-- 1 root root        621 Jul 27 05:53 jvm-defaults.xml
-rw-rw-r-- 1 root root       3392 Jul 27 05:53 nacm.xml
-rw-r--r-- 1 root root       6156 Dec 20 02:53 O.cdb
-rw-r--r-- 1 root root     13041 Dec 20 02:48 startup-vnfd.xml

root@vnfd1deploymentem-0:/opt/cisco/em/git/em-scm/ncs-cdb#
root@vnfd1deploymentem-0:/opt/cisco/em/git/em-scm# tar cvf em_cdb_backup.tar ncs-cdb

ncs-cdb/
ncs-cdb/O.cdb
ncs-cdb/C.cdb
ncs-cdb/nacm.xml
ncs-cdb/jvm-defaults.xml
ncs-cdb/A.cdb
ncs-cdb/aaa_users.xml
ncs-cdb/global-settings.xml
ncs-cdb/.gitignore
ncs-cdb/startup-vnfd.xml
```

```
root@vnfd1deploymentem-0:/opt/cisco/em/git/em-scm# ll em_cdb_backup.tar
-rw-r--r-- 1 root root 484034560 Dec 20 23:06 em_cdb_backup.tar
```

6. 스크립트 디렉토리로 이동하여 로그를 수집하고 백업 서버로 전송합니다.

```
/opt/cisco/em-scripts
```

```
sudo ./collect-em-logs.sh
```

- NCS 대기 VM에서 로그를 수집하려면 동일한 절차를 반복합니다.
- 모든 EM VM에서 syslog 컨피그레이션을 백업하고 백업 서버로 전송합니다.

```
root@vnfd1deploymentem-0:/etc/rsyslog.d# pwd  
/etc/rsyslog.d
```

```
root@vnfd1deploymentem-0:/etc/rsyslog.d# ll  
total 28
```

```
drwxr-xr-x  2 root root 4096 Jun  7 18:38 ./  
drwxr-xr-x 86 root root 4096 Jun  6 20:33 ../  
-rw-r--r--  1 root root  319 Jun  7 18:36 00-vmmf-proxy.conf  
-rw-r--r--  1 root root  317 Jun  7 18:38 01-ncs-java.conf  
-rw-r--r--  1 root root  311 Mar 17  2012 20-ufw.conf  
-rw-r--r--  1 root root  252 Nov 23  2015 21-cloudinit.conf  
-rw-r--r--  1 root root 1655 Apr 18  2013 50-default.conf
```

```
root@vnfd1deploymentem-0:/etc/rsyslog.d# ls /etc/rsyslog.conf  
rsyslog.conf
```

StarOS 백업

StarOS의 경우 이 정보를 백업해야 합니다.

- CDR(Charging Data Records)은 정기적으로 원격 서버에서 백업해야 합니다.
- 시스템 구축 시 Day-0 컨피그레이션 적용
- Day-N 컨피그레이션
- SSD
- Syslog 컨피그레이션

복원 절차

OSPD 복구

OSPD 복구 절차는 이러한 가정을 기반으로 수행됩니다.

- 이전 OSPD 서버에서 OSPD 백업을 사용할 수 있습니다.
- OSPD 복구는 시스템에서 이전 OSPD 서버를 대체하는 새 서버에서 수행됩니다.

복구 자동 구축

1. VM이 오류 또는 종료 상태일 때 VM 자동 배포를 복구할 수 있습니다. 영향을 받는 VM을 표시하려면 하드 재부팅을 수행하십시오. 이러한 검사를 실행하여 AutoDeploy를 복구하는 데 도움이 되는지 확인합니다.

Checking AutoDeploy Processes

Verify that key processes are running on the AutoDeploy VM:

```
root@auto-deploy-iso-2007-uas-0:~# initctl status autodeploy
autodeploy start/running, process 1771
```

```
root@auto-deploy-iso-2007-uas-0:~# ps -ef | grep java
root      1788  1771  0 May24 ?          00:00:41 /usr/bin/java -jar
/opt/cisco/usp/apps/autodeploy/autodeploy-1.0.jar com.cisco.usp.autodeploy.Application --
autodeploy.transaction-log-store=/var/log/cisco-uas/autodeploy/transactions
Stopping/Restarting AutoDeploy Processes
```

#To start the AutoDeploy process:

```
root@auto-deploy-iso-2007-uas-0:~# initctl start autodeploy
autodeploy start/running, process 11094
```

#To stop the AutoDeploy process:

```
root@auto-deploy-iso-2007-uas-0:~# initctl stop autodeploy
autodeploy stop/waiting
```

#To restart the AutoDeploy process:

```
root@auto-deploy-iso-2007-uas-0:~# initctl restart autodeploy
autodeploy start/running, process 11049
```

#If the VM is in ERROR or shutdown state, hard-reboot the AutoDeploy VM

```
[stack@pod1-ospd ~]$ nova list |grep auto-deploy
| 9b55270a-2dcd-4ac1-aba3-bf041733a0c9 | auto-deploy-ISO-2007-uas-
0 | ACTIVE | - | running | mgmt=172.16.181.12,
10.84.123.39
```

```
[stack@pod1-ospd ~]$ nova reboot -hard 9b55270a-2dcd-4ac1-aba3-bf041733a0c9
```

2. 자동 배포를 복구할 수 없는 경우 다음 절차를 수행하여 이전 상태로 복원합니다. 이전에 수행한 백업을 사용합니다.

```
[stack@pod1-ospd ~]$ nova list |grep auto-deploy
| 9b55270a-2dcd-4ac1-aba3-bf041733a0c9 | auto-deploy-ISO-2007-uas-
0 | ACTIVE | - | running | mgmt=172.16.181.12,
10.84.123.39 [stack@pod1-ospd ~]$ cd /opt/cisco/usp/uas-installer/scripts
```

```
[stack@pod1-ospd ~]$ ./auto-deploy-booting.sh --floating-ip 10.1.1.2 --delete
```

3. 자동 배포가 삭제된 후 동일한 폴더 주소로 다시 만듭니다.

```
[stack@pod1-ospd ~]$ cd /opt/cisco/usp/uas-installer/scripts
```

```
[stack@pod1-ospd scripts]$ ./auto-deploy-booting.sh --floating-ip 10.1.1.2
```

```
2017-11-17 07:05:03,038 - INFO: Creating AutoDeploy deployment (1 instance(s)) on
'http://10.1.1.2:5000/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 '10.1.1.2' 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: 10.1.1.2]
```

4. Autodeploy.cfg 파일, ISO 및 confd_backup tar 파일을 백업 서버에서 AutoDeploy VM으로 복사합니다.

5. 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
```

#Restart AutoDeploy process

```
root@auto-deploy-iso-2007-uas-0~# service autodeploy restart
autodeploy start/running, process 2144
```

#Check that confd was loaded properly by checking earlier transactions.

```
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
```

6. VM이 복구 · 실행되는 경우 이전에 성공한 알려진 백업에서 모든 syslog 관련 컨피그레이션이 복원되었는지 확인합니다.

```
ubuntu@auto-deploy-vnf-iso-5-1-5-1196-uas-0:~$sudo su
root@auto-deploy-vnf-iso-5-1-5-1196-uas-0:/home/ubuntu#ls /etc/rsyslog.d/00-autodeploy.conf
00-autodeploy.conf
```

```
root@auto-deploy-vnf-iso-5-1-5-1196-uas-0:/home/ubuntu#ls /etc/rsyslog.conf
rsyslog.conf
```

AutoIT-VNF 복구

1. AutoIT-VNF VM을 복구할 수 있습니다. VM이 오류 또는 종료 상태인 경우 영향을 받는 VM을 표시하려면 하드 재부팅을 수행합니다. AutoIT-VNF를 복구하려면 다음 단계를 수행하십시오.

Checking AutoIT-VNF Processes

Verify that key processes are running on the AutoIT-VNF VM:

```
root@auto-it-vnf-iso-5-1-5-1196-uas-0:~# service autoit status
AutoIT-VNF is running.
```

#Stopping/Restarting AutoIT-VNF Processes

```
root@auto-it-vnf-iso-5-1-5-1196-uas-0:~# service autoit stop
AutoIT-VNF API server stopped.
```

#To restart the AutoIT-VNF processes:

```
root@auto-it-vnf-iso-5-1-5-1196-uas-0:~# service autoit restart
AutoIT-VNF API server stopped.
Starting AutoIT-VNF
```

```
/opt/cisco/usp/apps/auto-it/vnf
AutoIT API server started.
```

#If the VM is in ERROR or shutdown state, hard-reboot the AutoDeploy VM

```
[stack@pod1-ospd ~]$ nova list |grep auto-it
| 1c45270a-2dcd-4ac1-aba3-bf041733d1a1 | auto-it-vnf-ISO-2007-uas-
0 | ACTIVE | - | running | mgmt=172.16.181.13,
10.84.123.40
```

```
[stack@pod1-ospd ~]$ nova reboot -hard 1c45270a-2dcd-4ac1-aba3-bf041733d1a1
```

2. AutoIT-VNF를 복구할 수 없는 경우 다음 절차를 수행하여 이전 상태로 복원합니다. 백업 파일을 사용합니다.

```
[stack@pod1-ospd ~]$ nova list |grep auto-it
| 580faf80-1d8c-463b-9354-781ea0c0b352 | auto-it-vnf-ISO-2007-uas-
0 | ACTIVE | - | running | mgmt=172.16.181.3,
10.84.123.42 [stack@pod1-ospd ~]$ cd /opt/cisco/usp/uas-installer/scripts
```

```
[stack@pod1-ospd ~]$ ./ auto-it-vnf-staging.sh --floating-ip 10.1.1.3 --delete
```

3. auto-it-vnf 스테이징 스크립트를 실행하여 Auto-IT를 다시 생성하고 이전에 사용한 것과 동일한 부동 IP를 사용해야 합니다.

```
[stack@pod1-ospd ~]$ cd /opt/cisco/usp/uas-installer/scripts
```

```
[stack@pod1-ospd scripts]$ ./auto-it-vnf-staging.sh --floating-ip 10.1.1.3
```

```
2017-11-16 12:54:31,381 - INFO: Creating StagingServer deployment (1 instance(s)) on
'http://10.1.1.3:5000/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 '10.1.1.3' 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: 10.1.1.3
```

4. POD에 사용된 ISO 이미지는 AutoIT-VNF에서 다시 로드해야 합니다.

```
[stack@pod1-ospd ~]$ cd images/5_1_7-2007/isos
```

```
[stack@pod1-ospd isos]$ curl -F file=@usp-5_1_7-2007.iso http://10.1.1.3:5001/isos
```

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

Note: 10.1.1.3 is AutoIT-VNF IP in the above command.

#Validate that ISO is correctly loaded.

```
[stack@pod1-ospd isos]$ curl http://10.1.1.3:5001/isos
```

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

5. VNF **system.cfg** 파일을 원격 서버에서 AutoIT-VNF VM으로 복사합니다. 이 예에서는 AutoDeploy에서 AutoIT-VNF VM으로 복사됩니다.

```
[stack@pod1-ospd autodeploy]$ scp system-vnf* ubuntu@10.1.1.3:.  
ubuntu@10.1.1.3'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
```

6. AutoDeploy 컨피그레이션에서 참조되는 대로 AutoIT-VNF의 적절한 위치에 파일을 복사합니다. 여기를 참조하십시오.

```
ubuntu@auto-it-vnf-iso-2007-uas-0:~$ sudo -i  
root@auto-it-vnf-iso-2007-uas-0:~$ cp -rp system-vnf1.cfg system-vnf2.cfg  
/opt/cisco/usp/uploads/  
root@auto-it-vnf-iso-2007-uas-0:~$ ls /opt/cisco/usp/uploads/  
system-vnf1.cfg system-vnf2.cfg
```

7. VM이 성공적으로 복원되고 실행 중인 경우, 이전에 성공한 알려진 백업에서 모든 syslog 관련 컨피그레이션이 복원되었는지 확인합니다.

```
root@auto-deploy-vnf-iso-5-1-5-1196-uas-0:/home/ubuntu#ls /etc/rsyslog.d/00-autoit-vnf.conf  
00-autoit-vnf.conf
```

```
root@auto-deploy-vnf-iso-5-1-5-1196-uas-0:ls /etc/rsyslog.conf  
rsyslog.conf
```

AutoVNF 복구

1. VM이 오류 또는 종료 상태인 경우 AutoVNF VM을 복구할 수 있습니다. 영향을 받는 VM을 표시하려면 하드 리부팅을 수행합니다. AutoVNF를 복구하려면 다음 단계를 실행합니다.

2. 오류 또는 종료 상태의 VM을 식별합니다. AutoVNF VM을 하드 재부팅합니다.

이 예에서는 auto-testautovnf1-uas-2를 재부팅합니다.

```
[root@tb1-baremetal scripts]# nova list | grep "auto-testautovnf1-uas-[0-2]"

| 3834a3e4-96c5-49de-a067-68b3846fba6b | auto-testautovnf1-uas-0 | ACTIVE | - | running | auto-testautovnf1-uas-orchestration=172.57.12.6; auto-testautovnf1-uas-management=172.57.11.8
|
| 0fbfec0c-f4b0-4551-807b-50c5fe9d3ea7 | auto-testautovnf1-uas-1 | ACTIVE | - | running | auto-testautovnf1-uas-orchestration=172.57.12.7; auto-testautovnf1-uas-management=172.57.11.12
|
| 432e1a57-00e9-4e58-8bef-2a20652df5bf | auto-testautovnf1-uas-2 | ACTIVE | - | running | auto-testautovnf1-uas-orchestration=172.57.12.13; auto-testautovnf1-uas-management=172.57.11.4
|
```

```
[root@tb1-baremetal scripts]# nova reboot --hard 432e1a57-00e9-4e58-8bef-2a20652df5bf
Request to reboot server <Server: auto-testautovnf1-uas-2> has been accepted.
```

```
[root@tb1-baremetal scripts]#
```

3. VM이 나타나면 클러스터에 다시 가입하는지 확인합니다.

```
root@auto-testautovnf1-uas-1:/opt/cisco/usp/uas/scripts# confd_cli -u admin -C
Welcome to the ConfD CLI
admin connected from 127.0.0.1 using console on auto-testautovnf1-uas-1
```

```
auto-testautovnf1-uas-1#show uas
```

```
uas version 1.0.1-1
uas state ha-active
uas ha-vip 172.57.11.101
```

```
INSTANCE IP STATE ROLE
-----
172.57.12.6 alive CONFD-SLAVE
172.57.12.7 alive CONFD-MASTER
172.57.12.13 alive NA
```

4. AutoVNF VM을 앞서 설명한 절차에 따라 복구할 수 없는 경우 이러한 단계의 도움을 받아 복구해야 합니다.

```
[stack@pod1-ospd ~]$ nova list | grep vnf1-UAS-uas-0
| 307a704c-a17c-4cdc-8e7a-3d6e7e4332fa | vnf1-UAS-uas-0 | ACTIVE | - | running | vnf1-UAS-uas-orchestration=172.168.11.10; vnf1-UAS-uas-management=172.168.10.3
```

```
[stack@pod1-ospd ~]$ nova delete vnf1-UAS-uas-0
Request to delete server vnf1-UAS-uas-0 has been accepted.
```

5. autovnf-uas VM을 복구하려면 uas-check 스크립트를 실행하여 상태를 확인합니다. 오류를 보고

해야 합니다.그런 다음 다음으로 다시 실행 —누락된 UAS VM을 다시 생성하려면 수정 옵션을 선택 합니다.

```
[stack@pod1-ospd ~]$ cd /opt/cisco/usp/uas-installer/scripts/  
[stack@pod1-ospd scripts]$ ./uas-check.py auto-vnf vnf1-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@tb3-ospd scripts]$ ./uas-check.py auto-vnf vnf1-UAS --fix
```

```
2017-11-22 14:01:07,215 - INFO: Check of AutoVNF cluster started  
2017-11-22 14:01:09,575 - INFO: Instance vnf1-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 vnf1-UAS-uas-0'  
2017-11-22 14:01:13,568 - INFO: Removed instance vnf1-UAS-uas-0'  
2017-11-22 14:01:13,568 - INFO: Creating instance vnf1-UAS-uas-0' and attaching volume 'vnf1-UAS-uas-vol-0'  
2017-11-22 14:01:49,525 - INFO: Created instance 'vnf1-UAS-uas-0'
```

```
[stack@tb3-ospd scripts]$ ./uas-check.py auto-vnf vnf1-UAS
```

```
2017-11-16 13:11:07,472 - INFO: Check of AutoVNF cluster started  
2017-11-16 13:11:09,510 - INFO: Found 3 ACTIVE AutoVNF instances  
2017-11-16 13:11:09,511 - INFO: Check completed, AutoVNF cluster is fine
```

6. 마스터 AutoVNF VM에 로그인합니다.복구 후 몇 분 이내에 새로 생성된 인스턴스는 클러스터에 가입하고 활성 상태여야 합니다.

```
tb3-bxb-vnf1-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
```

#if uas-check.py --fix fails, you may need to copy this file and execute again.

```
[stack@tb3-ospd]$ mkdir -p /opt/cisco/usp/apps/auto-it/common/uas-deploy/  
[stack@tb3-ospd]$ cp /opt/cisco/usp/uas-installer/common/uas-deploy/userdata-uas.txt  
/opt/cisco/usp/apps/auto-it/common/uas-deploy/
```

7. VM이 성공적으로 복원되고 실행 중인 경우, 이전에 성공한 알려진 백업에서 모든 syslog 관련 컨 피그레이션이 복원되었는지 확인합니다.모든 AutoVNF VM에서 복원되었는지 확인합니다.

```
ubuntu@auto-testautovnf1-uas-1:~$sudo su  
root@auto-testautovnf1-uas-1:/home/ubuntu#ls /etc/rsyslog.d/00-autovnf.conf  
00-autovnf.conf
```

```
root@auto-testautovnf1-uas-1:/home/ubuntu#ls /etc/rsyslog.conf  
rsyslog.conf
```

ESC 복구

1. VM이 오류 또는 종료 상태인 경우 ESC VM을 복구할 수 있습니다. 영향을 받는 VM을 표시하려면 하드 재부팅을 수행하십시오.ESC를 복구하려면 다음 단계를 실행합니다.

2. ESC VM을 하드 재부팅한 후 오류 또는 종료 상태의 VM을 식별합니다.이 예에서는 auto-test-vnfm1-ESC-0이 재부팅됩니다.

```
[root@tb1-baremetal scripts]# nova list | grep auto-test-vnfm1-ESC-
| f03e3cac-a78a-439f-952b-045aea5b0d2c | auto-test-vnfm1-ESC-
0 | ACTIVE | - | running | auto-testautovnf1-
uas-orchestration=172.57.12.11; auto-testautovnf1-uas-
management=172.57.11.3
|
| 79498e0d-0569-4854-a902-012276740bce | auto-test-vnfm1-ESC-
1 | ACTIVE | - | running | auto-testautovnf1-
uas-orchestration=172.57.12.15; auto-testautovnf1-uas-
management=172.57.11.5
|
```

```
[root@tb1-baremetal scripts]# [root@tb1-baremetal scripts]# nova reboot --hard f03e3cac-a78a-
439f-952b-045aea5b0d2c\
Request to reboot server <Server: auto-test-vnfm1-ESC-0> has been accepted.
```

```
[root@tb1-baremetal scripts]#
```

3. ESC VM이 삭제되어 다시 시작해야 하는 경우 다음 단계를 수행하십시오.

```
[stack@pod1-ospd scripts]$ nova list | grep ESC-1
| c566efbf-1274-4588-a2d8-0682e17b0d41 | vnfm1-ESC-ESC-
1 | ACTIVE | - | running | vnfm1-
UAS-uas-orchestration=172.168.11.14; vnfm1-UAS-uas-
management=172.168.10.4
|
```

```
[stack@pod1-ospd scripts]$ nova delete vnfm1-ESC-ESC-1
Request to delete server vnfm1-ESC-ESC-1 has been
accepted.
```

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

```
ubuntu@vnfm1-uas-uas-0:~$ sudo -i
root@vnfm1-uas-uas-0:~# confd_cli -u admin -C
Welcome to the ConfD CLI
```

```
admin connected from 127.0.0.1 using console on vnfm1-uas-uas-0
vnfm1-uas-uas-0#show transaction
```

TX ID	TX TYPE	DEPLOYMENT ID	TIMESTAMP	STATUS
35eefc4a-d4a9-11e7-bb72-fa163ef8df2b	vnfm-deployment	vnfm1-DEPLOYMENT	2017-11-29T02:01:27.750692-00:00	deployment-success
73d9c540-d4a8-11e7-bb72-fa163ef8df2b	vnfm-deployment	vnfm1-ESC	2017-11-29T01:56:02.133663-00:00	deployment-success

```
vnfm1-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 - VNFM Deployment RPC triggered for deployment: vnf1-ESC,
deactivate: 0

2017-11-29 01:56:02,179 - Notify deployment
..
2017-11-29 01:57:30,385 - Creating VNFM 'vnf1-ESC-ESC-1' with [python //opt/cisco/vnf-
staging/bootvm.py vnf1-ESC-ESC-1 --flavor vnf1-ESC-ESC-flavor --image 3fe6b197-961b-4651-af22-
dfd910436689 --net vnf1-UAS-uas-management --gateway_ip 172.168.10.1 --net vnf1-UAS-uas-
orchestration --os_auth_url http://10.1.1.5:5000/v2.0 --os_tenant_name core --os_username *****
--os_password ***** --bs_os_auth_url http://10.1.1.5:5000/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_confd_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]...
```

5. boot_vm.py 줄을 셸 스크립트 파일(esc.sh)에 저장하고 모든 사용자 이름 ***** 및 비밀번호 ***** 행을 올바른 정보로 업데이트합니다(일반적으로 core/Cisco@123). -encrypt_key 옵션도 제거해야 합니다.user_pass 및 user_config_pass의 경우 -user_passwd username:password 형식을 사용해야 합니다(예: admin:Cisco@123).

이제 running-config에서 bootvm.py에 대한 URL을 찾아 bootvm.py 파일을 autovnf-uas VM에 가져옵니다.10.1.1.3은 이 경우 Auto-IT입니다.

```
root@vnf1-uas-uas-0:~# confd_cli -u admin -C
Welcome to the ConfD CLI
admin connected from 127.0.0.1 using console on vnf1-uas-uas-0

vnf1-uas-uas-0#show running-config autovnf-vnfm:vnfm
...
configs bootvm
  value http://10.1.1.3:80/bundles/5.1.7-2007/vnfm-bundle/bootvm-2_3_2_155.py
!

root@vnf1-uas-uas-0:~# wget http://10.1.1.3:80/bundles/5.1.7-2007/vnfm-bundle/bootvm-
2_3_2_155.py

--2017-12-01 20:25:52-- http://10.1.1.3/bundles/5.1.7-2007/vnfm-bundle/bootvm-2_3_2_155.py
Connecting to 10.1.1.3:80... 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]
Create a /tmp/esc_params.cfg file.

root@vnf1-uas-uas-0:~# echo "openstack.endpoint=publicURL" > /tmp/esc_params.cfg
```

6. bootvm.py python 스크립트를 실행하는 셸 스크립트를 옵션과 함께 실행합니다.

```
root@vnf1-uas-uas-0:~# /bin/sh esc.sh
```

```
+ python ./bootvm.py vnf1-ESC-ESC-1 --flavor vnf1-ESC-ESC-flavor --image 3fe6b197-961b-4651-af22-dfd910436689 --net vnf1-UAS-uas-management --gateway_ip 172.168.10.1 --net vnf1-UAS-uas-orchestration --os_auth_url http://10.1.1.5:5000/v2.0 --os_tenant_name core --os_username core --os_password Cisco@123 --bs_os_auth_url http://10.1.1.5:5000/v2.0 --bs_os_tenant_name core --bs_os_username core --bs_os_password Cisco@123 --esc_ui_startup false --esc_params_file /tmp/esc_params.cfg --user_pass admin:Cisco@123 --user_confd_pass admin:Cisco@123 --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
```

```

+-----+-----+
| Property | |
Value |
|
+-----+-----+
| OS-DCF:diskConfig |
MANUAL |
|
| OS-EXT-AZ:availability_zone |
mgmt |
|
| OS-EXT-SRV-ATTR:host |
1.localdomain | tb5-ultram-osd-compute-
|
| OS-EXT-SRV-ATTR:hypervisor_hostname |
1.localdomain | tb5-ultram-osd-compute-
|
| OS-EXT-SRV-ATTR:instance_name |
000001eb | instance-
|
| OS-EXT-STS:power_state |
1 |
|
| OS-EXT-STS:task_state |
| -
|
| OS-EXT-STS:vm_state |
active |

```

```

| OS-SRV-USG:launched_at          | 2017-12-
02T13:28:32.000000

| OS-SRV-USG:terminated_at        | -

| accessIPv4                       |

| accessIPv6                       |

| addresses                        | {"vnf1-UAS-uas-orchestration": [{"OS-EXT-IPS-
MAC:mac_addr": "fa:16:3e:d7:c6:19", "version": 4, "addr": "172.168.11.14", "OS-EXT-IPS:type":
"fixed"}], "vnf1-UAS-uas-management": [{"OS-EXT-IPS-MAC:mac_addr": "fa:16:3e:31:ee:cd",
"version": 4, "addr": "172.168.10.6", "OS-EXT-IPS:type": "fixed"}]}
| config_drive                     | True

| created                          | 2017-12-
02T13:27:49Z

| flavor                            | {"id": "457623b6-05d5-403c-b2e4-aa3b6a0c9d32", "links":
[{"href": "http://10.1.1.5:8774/flavors/457623b6-05d5-403c-b2e4-aa3b6a0c9d32", "rel":
"bookmark"}]}

| hostId                           |
f5d2bbf0c5a7df34cf2e6f62ae0702ef120ff82f81c3f7664ffb35e9

| id                                | 2601b8ec-8ff8-4285-810a-
e859f6642ab6

| image                            | {"id": "3fe6b197-961b-4651-af22-dfd910436689", "links":
[{"href": "http://10.1.1.5:8774/images/3fe6b197-961b-4651-af22-dfd910436689", "rel":
"bookmark"}]}

| key_name                          | -

| metadata                         | {}

| name                              | vnf1-esc-esc-
1

```

```

| os-extended-volumes:volumes_attached | []
| progress |
0
| security_groups | [{"name": "default"}, {"name":
"default"}]
| status |
ACTIVE
| tenant_id |
fd4b15df46c6469cbacf5b80dcc98a5c
| updated | 2017-12-
02T13:28:32Z
| user_id |
d3b51d6f705f4826b22817f27505c6cd

```

7. OSPD에서 새 ESC VM이 활성/실행 중인지 확인합니다.

```

[stack@pod1-ospd ~]$ nova list|grep -i esc
| 934519a4-d634-40c0-a51e-fc8d55ec7144 | vnf1-ESC-ESC-
0 | ACTIVE | - | running | vnf1-
UAS-uas-orchestration=172.168.11.13; vnf1-UAS-uas-
management=172.168.10.3
| 2601b8ec-8ff8-4285-810a-e859f6642ab6 | vnf1-ESC-ESC-
1 | ACTIVE | - | running | vnf1-
UAS-uas-orchestration=172.168.11.14; vnf1-UAS-uas-
management=172.168.10.6

```

#Log in to new ESC and verify Backup state. You may execute health.sh on ESC Master too.

```
ubuntu@vnf1-uas-uas-0:~$ ssh admin@172.168.11.14
```

```

...
#####
# ESC on vnf1-esc-esc-1.novalocal is in BACKUP state.
#####

```

```
[admin@vnf1-esc-esc-1 ~]$ escadm status
0 ESC status=0 ESC Backup Healthy
```

```
[admin@vnf1-esc-esc-1 ~]$ health.sh
===== ESC HA (BACKUP) =====
```

=====

ESC HEALTH PASSED

[admin@vnf1-esc-esc-1 ~]\$ cat /proc/drbd

version: 8.4.7-1 (api:1/proto:86-101)

GIT-hash: 3a6a769340ef93b1ba2792c6461250790795db49 build by mockbuild@Build64R6, 2016-01-12
13:27:11

1: cs:Connected ro:Secondary/Primary ds:UpToDate/UpToDate C r-----

ns:0 nr:504720 dw:3650316 dr:0 al:8 bm:0 lo:0 pe:0 ua:0 ap:0 ep:1 wo:f oos:0

8. ESC VM을 복구할 수 없고 데이터베이스를 복원해야 하는 경우 이전에 수행한 백업에서 데이터베이스를 복원하십시오.

9. ESC 데이터베이스 복원의 경우 데이터베이스를 복원하기 전에 ESC 서비스가 중지되었는지 확인하십시오. ESC HA의 경우 먼저 보조 VM에서 실행한 다음 기본 VM에서 실행합니다.

service keepalived stop

10. ESC 서비스 상태를 확인하고 HA용 기본 및 보조 VM 모두에서 모든 것이 중지되었는지 확인합니다.

escadm status

11. 데이터베이스를 복원하려면 스크립트를 실행합니다. DB를 새로 생성된 ESC 인스턴스로 복원하는 과정에서 이 틀은 인스턴스 중 하나를 기본 ESC로 승격하고 DB 폴더를 DRBD 디바이스에 마운트하고 PostgreSQL 데이터베이스를 시작합니다.

/opt/cisco/esc/esc-scripts/esc_dbtool.py restore --file

scp://<username>:<password>@<backup_vm_ip>:<filename>

12. 데이터베이스 복원을 완료하려면 ESC 서비스를 다시 시작하십시오.

13. 두 VM에서 HA를 실행하려면 keepalive 서비스를 다시 시작합니다.

service keepalived start

14. VM이 성공적으로 복원되고 실행되면 이전에 성공한 알려진 백업에서 모든 syslog 관련 컨피그레이션이 복원되었는지 확인합니다. 모든 ESC VM에서 복원되었는지 확인합니다.

[admin@auto-test-vnfm2-esc-1 ~]\$

[admin@auto-test-vnfm2-esc-1 ~]\$ cd /etc/rsyslog.d

[admin@auto-test-vnfm2-esc-1 rsyslog.d]\$ ls /etc/rsyslog.d/00-escmanager.conf
00-escmanager.conf

[admin@auto-test-vnfm2-esc-1 rsyslog.d]\$ ls /etc/rsyslog.d/01-messages.conf
01-messages.conf

[admin@auto-test-vnfm2-esc-1 rsyslog.d]\$ ls /etc/rsyslog.d/02-mona.conf
02-mona.conf

[admin@auto-test-vnfm2-esc-1 rsyslog.d]\$ ls /etc/rsyslog.conf
rsyslog.conf

EM 복구

1. EM VM이 하나 또는 다른 조건 때문에 None/Error(없음/오류) 상태인 경우 사용자는 지정된 시퀀스를 따라 영향을 받는 EM VM을 복구할 수 있습니다.

2. ESC/VNFM은 EM VM을 모니터링하는 구성 요소이므로 EM이 오류 상태인 경우 ESC는 EM VM을 자동 복구하려고 시도합니다.어떤 이유로든 ESC가 복구를 성공적으로 완료할 수 없는 경우 ESC는 해당 VM을 오류 상태로 표시합니다.

3. 이 경우 기본 인프라 문제가 해결되면 사용자는 EM VM을 수동으로 복구할 수 있습니다.기본 문제를 해결한 후에만 이 수동 복구를 실행하는 것이 중요합니다.

4. 오류 상태의 VM을 식별합니다.

```
[stack@pod1-ospd ~]$ source corerc
[stack@pod1-ospd ~]$ nova list --field name,host,status |grep -i err
| c794207b-a51e-455e-9a53-3b8ff3520bb9 | vnf1-DEPLOYMENT-_vnf1-D_0_a6843886-77b4-4f38-b941-74eb527113a8 | None | ERROR |
```

5. 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.

```
ubuntu@vnf1-uas-uas-1:~$ ssh admin@172.168.10.3
```

...

```
[admin@vnf1-esc-esc-0 ~]$ sudo /opt/cisco/esc/esc-confd/esc-cli/esc_nc_cli recovery-vm-action DO vnf1-DEPLOYMENT-_vnf1-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.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>
```

6. 명령이 완료될 때까지 **/var/log/esc/yangesc.log**를 모니터링합니다.

```
[admin@vnf1-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 [vnf1-DEPLOYMENT-_vnf1-D_0_a6843886-77b4-4f38-b941-74eb527113a8]
```

```
#Log in to new EM and verify EM state is up.
```

```
ubuntu@vnf1vnfddeploymentem-1:~$ /opt/cisco/ncs/current/bin/ncs_cli -u admin -C
```

```
admin connected from 172.17.180.6 using ssh on vnf1vnfddeploymentem-1
```

```
admin@scm# show ems
```

```
EM VNFM
```

```
ID SLA SCM PROXY
```

```
-----
2 up up up
```

ESC가 VM을 시작하지 못하는 경우

1. 예기치 않은 상태로 인해 ESC가 VM을 시작하지 못하는 경우도 있습니다. 해결 방법은 마스터 ESC를 재부팅하여 ESC 전환을 수행하는 것입니다. ESC 전환은 약 1분 정도 걸립니다. 새 마스터 ESC에서 health.sh를 실행하여 작동 여부를 확인합니다. ESC가 마스터가 되면 ESC는 VM 상태를 수정하고 VM을 시작할 수 있습니다. 이 작업이 예약되었으므로 작업이 완료될 때까지 5-7분 기다려야 합니다.
2. /var/log/esc/yangesc.log 및 /var/log/esc/escmanager.log를 모니터링할 수 있습니다. 5~7분 후에도 VM이 복구되지 않는 경우, 사용자는 이동 후 영향을 받는 VM을 수동으로 복구해야 합니다.
3. VM이 복구 · 실행되면 모든 syslog 관련 컨피그레이션이 이전에 성공한 알려진 백업에서 복원되었는지 확인합니다. 모든 ESC VM에서 복원되었는지 확인합니다.

```
root@abautotestvnmlem-0:/etc/rsyslog.d# pwd
/etc/rsyslog.d
```

```
root@abautotestvnmlem-0:/etc/rsyslog.d# ll
```

```
total 28
drwxr-xr-x  2 root root 4096 Jun  7 18:38 ./
drwxr-xr-x 86 root root 4096 Jun  6 20:33 ../]
-rw-r--r--  1 root root  319 Jun  7 18:36 00-vmmf-proxy.conf
-rw-r--r--  1 root root  317 Jun  7 18:38 01-ncs-java.conf
-rw-r--r--  1 root root  311 Mar 17 2012 20-ufw.conf
-rw-r--r--  1 root root  252 Nov 23 2015 21-cloudinit.conf
-rw-r--r--  1 root root 1655 Apr 18 2013 50-default.conf
```

```
root@abautotestvnmlem-0:/etc/rsyslog.d# ls /etc/rsyslog.conf
rsyslog.conf
```

StarOS 복구

1. 하나 또는 다른 조건 때문에 StarOS VM 중 하나가 None/Error(없음/오류) 상태로 나타날 경우 사용자는 이 시퀀스를 따라 영향을 받는 StarOS VM을 복구할 수 있습니다.
2. ESC/VNFM은 StarOS VM을 모니터링하는 구성 요소이므로 CF/SF VM이 오류 상태인 경우 ESC는 CF/SF VM을 자동 복구하려고 시도합니다. 어떤 이유로든 ESC가 복구를 성공적으로 완료할 수 없는 경우 ESC는 해당 VM을 오류 상태로 표시합니다.
3. 이 경우 기본 인프라 문제가 해결되면 사용자는 CF/SF VM을 수동으로 복구할 수 있습니다. 기본 문제를 해결한 후에만 이 수동 복구를 실행하는 것이 중요합니다.
4. 오류 상태에서 VM을 식별합니다.

```
[stack@pod1-ospd ~]$ source corerc
```

```
[stack@pod1-ospd ~]$ nova list --field name,host,status |grep -i err
| c794207b-a51e-455e-9a53-3b8ff3520bb9 | vnf1-DEPLOYMENT-s4_0_c2b19084-26b3-4c9c-8639-62428a4cb3a3 | None | ERROR |
```

5. ESC 마스터에 로그인하고 영향을 받는 각 EM 및 CF VM.Be에 대해 recovery-vm-action을 실행합니다. ESC는 복구 작업을 예약하며 몇 분 동안 이 작업이 수행되지 않을 수 있습니다.

```
ubuntu@vnf1-uas-uas-1:~$ ssh admin@172.168.10.3
```

```
...
```

```
[admin@vnf1-esc-esc-0 ~]$ sudo /opt/cisco/esc/esc-confd/esc-cli/esc_nc_cli recovery-vm-action DO vnf1-DEPLOYMENT-_s4_0_c2b19084-26b3-4c9c-8639-62428a4cb3a3
```

```
[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>
```

```
##Monitor the /var/log/esc/yangesc.log until command completes.
```

```
[admin@vnf1-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 [vnf1-DEPLOYMENT-_s4_0_c2b19084-26b3-4c9c-8639-62428a4cb3a3]
```

6. 또한 StarOS에서 **show card** 탭을 실행하여 동일한 항목을 검증합니다. 복구된 VM이 SF인 경우 원하는 경우 사용자가 활성화해야 할 수 있습니다. 필요한 StarOS 컨피그레이션을 변경합니다.

```
[local]VNF1# show card tab
```

```
Saturday December 02 14:40:20 UTC 2017
```

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	Standby	-	
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	Active	No	

ESC가 VM을 시작하지 못하는 경우

예기치 않은 상태로 인해 ESC가 VM을 시작하지 못하는 경우도 있습니다. 해결 방법은 마스터 ESC를 재부팅하여 ESC 전환을 수행하는 것입니다. ESC 전환은 약 1분 정도 걸립니다. 새 마스터 ESC에서 health.sh를 실행하여 작동 중인지 확인합니다. ESC가 마스터가 되면 ESC는 VM 상태를 수정하고 VM을 시작할 수 있습니다. 이 작업이 예약되었으므로 작업이 완료될 때까지 5-7분 기다려야 합니다. /var/log/esc/yangesc.log 및 /var/log/esc/escmanager.log을 모니터링할 수 있습니다. .5~7분 후에 VM이 복구되는 것을 볼 수 없는 경우, 이동하여 영향을 받는 VM을 수동으로 복구해야 합니다.