

さまざまなUltra-Mコンポーネントのバックアップ/リストア手順

内容

[概要](#)

[背景説明](#)

[省略形](#)

[バックアップ手順](#)

[OSPDバックアップ](#)

[AutoDeployバックアップ](#)

[AutoIT-VNFバックアップ](#)

[AutoVNFバックアップ](#)

[ESCバックアップ](#)

[EMバックアップ](#)

[StarOSバックアップ](#)

[復元手順](#)

[OSPDの回復](#)

[AutoDeployのリカバリ](#)

[AutoIT-VNFリカバリ](#)

[AutoVNFリカバリ](#)

[ESCリカバリ](#)

[EMリカバリ](#)

[StarOSリカバリ](#)

概要

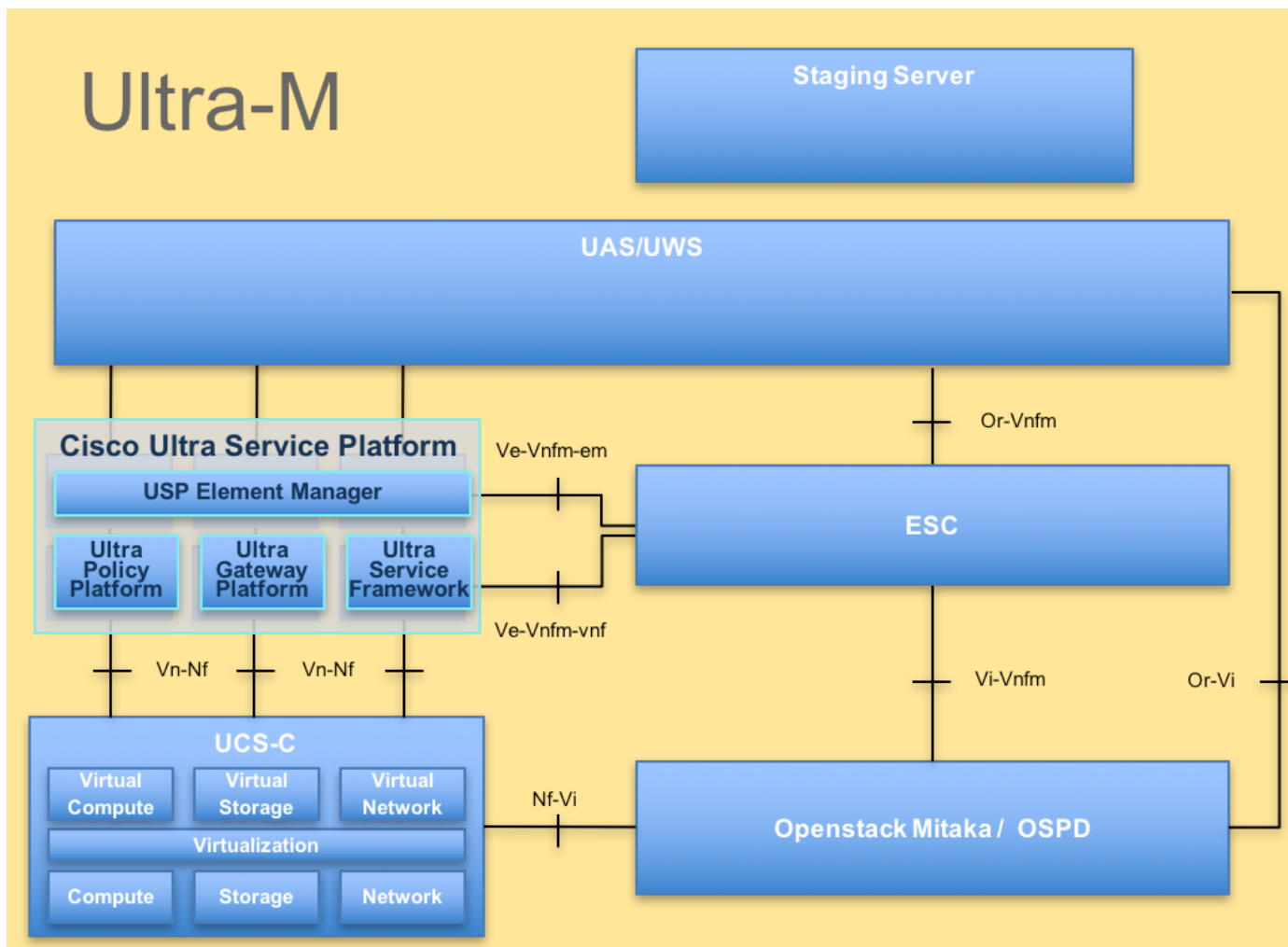
このドキュメントでは、StarOS Virtual Network Functions(VNF)をホストするUltra-Mセットアップで仮想マシン(VM)をバックアップおよび復元するために必要な手順について説明します。

背景説明

Ultra-Mは、VNFの導入を簡素化するように設計された、パッケージ化および検証済みの仮想化モジュールコアソリューションです。Ultra-Mソリューションは、次の仮想マシン(VM)タイプで構成されます。

- 自動IT
- 自動導入
- Ultra Automation Services (UAS)
- 要素マネージャ(EM)
- Elastic Services Controller (ESC)
- 制御機能(CF)
- セッション機能(SF)

Ultra-Mのアーキテクチャと関連するコンポーネントを次の図に示します。



このドキュメントは、Cisco Ultra-Mプラットフォームに精通しているシスコ担当者を対象としています。

注：このドキュメントの手順を定義するために、Ultra M 5.1.xリリースが検討されています。

省略形

VNF	仮想ネットワーク機能
CF	制御機能
SF	サービス機能
ESC	Elastic Service Controller
MOP	手続きの方法
OSD	オブジェクトストレージディスク
HDD	ハードディスクドライブ
SSD	ソリッドステートドライブ
VIM	仮想インフラストラクチャマネージャ
VM	仮想マシン
EM	エレメント マネージャ
UAS	Ultra Automation Services
UUID	ユニバーサル一意IDentifier

バックアップ手順

OSPDバックアップ

1. OpenStackスタックの状態とノードリストを確認します。

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

2. OSP-Dノードから、すべてのUndercloudサービスがロード済み、アクティブ、および実行中の状態であるかどうかを確認します。

```
[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.バックアッププロセスを実行する前に、十分なディスク領域があることを確認します。このtarballは少なくとも3.5 GBである必要があります。

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

4. Undercloudノードからundercloud-backup-[timestamp].tar.gzという名前のファイルにデータをバックアップし、バックアップサーバに転送するには、ルートユーザとしてこれらのコマンドを実行します。

```

[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

```

AutoDeployバックアップ

1. AutoDeployでは、次のデータをバックアップする必要があります。

- AutoDeploy Confd Configuration Database(CDB)
- AutoDeployの設定
- AutoDeployログ
- Syslogの設定

2. AutoDeploy Confd CDBデータのバックアップと実行コンフィギュレーションは、アクティブ化/非アクティブ化のたびに必要であり、データがバックアップサーバに転送されることを確認します。

3. AutoDeployはスタンドアロンモードで実行され、このデータが失われた場合、展開を正常に非アクティブ化することはできません。したがって、設定とCDBデータのバックアップを取る必要があります。

```
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**をバックアップ・サーバにコピーします。

5. AutoDeployの実行コンフィギュレーションをバックアップし、バックアップサーバに転送します。

```
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
auto-deploy#
```

6. AutoDeploy Confdサービスを開始します。

```
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. **scripts**ディレクトリに移動し、AutoDeploy VMからログを収集します。

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

8. ログを収集するために、**collect-uas-logs.sh**スクリプトを起動します。

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

9. AutoDeployからISOイメージのバックアップを取り、バックアップサーバに転送します。

```
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設定を収集し、バックアップサーバに保存します。

```
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バックアップ

AutoIT-VNFはステートレスVMであるため、バックアップが必要なデータベース(DB)はありません。AutoIT-VNFは、Ultra-Mの構成管理リポジトリとともにパッケージ管理を行います。したがって、これらのバックアップを実行することが不可欠です。

1. ゼロデイStarOS設定のバックアップをとり、バックアップサーバに転送します。

```
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. scriptsディレクトリに移動し、AutoIT VMからログを収集します。

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

3. ログを収集するために、collect-uas-logs.shスクリプトを起動します。

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

4. syslog設定のバックアップを収集し、バックアップサーバに保存します。

```
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バックアップ

AutoVNFは、個々のVNFMとVNFを起動する責任があります。AutoDeployはVNFMとVNFの両方をインスタンス化するために必要な設定をAutoVNFに送信し、AutoVNFはこの操作を実行します。VNFMを起動するために、AutoVNFはVIM/OpenStackと直接通信し、VNFMが起動した後、AutoVNFはVNFMを使用してVNFを起動します。

AutoVNFには1:Nの冗長性があり、Ultra-Mセットアップでは、3つのAutoVNF VMが実行されてい

ます。Ultra-Mでは単一のAutoVNF障害がサポートされ、リカバリが可能です。

注：複数の障害が発生した場合は、サポートされず、システムの再展開が必要になる可能性があります。

AutoVNFバックアップの詳細：

- Running Configuration
- ConfD CDB DB
- AutoVNFログ (AutoVNFの各インスタンスから)
- Syslogの設定

特定のサイトでアクティブ化/非アクティブ化を行い、バックアップサーバにアップロードする前に、バックアップを取ることをお勧めします。

1. マスターAutoVNFにログインし、confd-masterであることを確認します。

```
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  CONFD-SLAVE
172.57.12.7   alive  CONFD-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.実行コンフィギュレーションのバックアップを取り、ファイルをバックアップサーバに転送します。

```
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. CDBのバックアップを取り、ファイルをバックアップサーバに転送します。

```
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. scriptsディレクトリに移動し、ログを収集してバックアップサーバに転送します。

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

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

5. AutoVNFのスタンバイインスタンスにログインし、次の手順を実行して、ログを収集し、バックアップサーバに転送します。

6. マスターおよびスタンバイAutoVNF VMでsyslog設定をバックアップし、バックアップサーバに転送します。

```
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バックアップ

1. AutoVNFは、VIMと直接対話することによってUltra-MソリューションでESCを起動する責任があります。AutoVNF/EMはVNF固有の設定をESCおよびESCに渡し、VNFはVIMと対話することで起動します。

2. ESCには、Ultra-Mソリューションの1:1の冗長性があります。2つのESC VMが導入されており

、Ultra-Mで単一障害をサポートします。つまり、システムに単一障害がある場合は、システムを回復できます。

注：複数の障害が発生した場合は、サポートされず、システムの再展開が必要になる可能性があります。

ESCバックアップの詳細：

- Running Configuration
- ConfD CDB DB
- ESCログ
- Syslog設定

3. ESC DBバックアップの頻度は複雑で、ESCが導入された各種VNF VMのさまざまな状態マシンを監視および維持するため、慎重に処理する必要があります。これらのバックアップは、指定されたVNF/POD/Siteのアクティビティに従った後に実行することをお勧めします。

4. ESCの状態がhealth.shスクリプトを使用して良好であることを確認してください。

```
[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. 実行コンフィギュレーションのバックアップを取り、ファイルをバックアップサーバに転送します。

```
[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
```

バックアップデータベース

1. ESCをメンテナンスモードに設定します。

2. ESC VMにログインし、バックアップを取る前にこのコマンドを実行します。

```
[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モードをチェックし、メンテナンスモードであることを確認します。

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

4. ESCで利用可能なDBバックアップ復元ツールを使用したバックアップDB。

```
[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. ESCを[Operation Mode]に戻し、モードを確認します。

```
[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. scriptsディレクトリに移動し、ログを収集します。

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

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

7. スタンバイESC VMで同じ手順を繰り返し、ログをバックアップサーバに転送します。

8. ESC VMSの両方でsyslog設定バックアップを収集し、バックアップサーバに転送します。

```
[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
```

EMバックアップ

1. VNF/ESCが起動した後、AutoVNFはESCを使用してEMクラスタを起動します。EMクラスタが起動すると、EMはESCと対話してVNF(VPC/StarOS)を起動します。

2. EMは、Ultra-Mソリューションで1:Nの冗長性を備えています。3つのEM VMのクラスタがあり、Ultra-Mは単一のVM障害のリカバリをサポートします。

注：複数の障害が発生した場合は、サポートされず、システムの再展開が必要になる可能性があります。

EMバックアップの詳細：

- Running Configuration
- NCS DB
- EMログ (両方のNCSアクティブ/スタンバイVMから)
- Syslog設定

3. EM DBバックアップの頻度は複雑で、ESCが導入された各種VNF VMのさまざまな状態マシンを監視および維持するため、慎重に処理する必要があります。これらのバックアップは、特定のVNF/POD/Siteのアクティビティに従った後に実行することをお勧めします。

4. EM実行コンフィギュレーションをバックアップし、ファイルをバックアップサーバに転送します。

```
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バックアップを取り、ファイルをバックアップサーバに転送します。

```
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. scriptsディレクトリに移動し、ログを収集してバックアップサーバに転送します。

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

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

- 同じ手順を繰り返して、NCSスタンバイVMからログを収集します
- すべてのEM VMでsyslog設定をバックアップし、バックアップサーバに転送します

```
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-vnmf-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バックアップ

StarOSの場合、この情報をバックアップする必要があります。

- Charging Data Records(CDR)は、定期的によりリモートサーバでバックアップする必要があります
- システム導入時にDay-0設定を適用
- Day-N設定
- SSD
- Syslog設定

復元手順

OSPDの回復

OSPD回復手順は、次の前提条件に基づいて実行されます

- OSPDバックアップは古いOSPDサーバから利用できます
- OSPDの回復は、システム内の古いOSPDサーバを置き換える新しいサーバで行われます。

AutoDeployのリカバリ

1. AutoDeploy VMは、VMがエラーまたはシャットダウン状態のときに回復可能です。ハードリブートを実行して、影響を受けるVMを起動します。AutoDeployの回復に役立つかどうかを確認するには、次のチェックを実行します。

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. AutoDeployが回復不能な場合は、次の手順に従って以前の状態に復元します。以前に取得したバックアップを使用します。

```
[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. AutoDeployを削除した後、同じfloatingipアドレスを使用してAutoDeployを再び作成します。

```
[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ファイル、ISO、およびconfd_backup tarファイルをバックアップサーバからAutoDeploy VMにコピーします。

5. バックアップtarファイルからconfd cdbファイルを復元します。

```
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. VMが正常に復元されて実行されている場合。以前の正常な既知のバックアップからすべてのsyslog固有の設定が復元されていることを確認します。

```
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リカバリ

1. AutoIT-VNF VMは回復可能です。VMがエラーまたはシャットダウン状態の場合は、ハードリブートを実行して、影響を受けるVMを起動します。AutoIT-VNFを回復するには、次の手順を実行します。

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-VNFが回復不能な場合は、次の手順に従って、以前の状態に復元します。バックアップファイルを使用します。

```
[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ステージングスクリプトを実行してAuto-ITを再作成し、以前に使用したのと同じ浮動IPを使用してください。

```
[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. PODで使用されるISOイメージは、AutoIT-VNFでリロードする必要があります。

```
[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. VNFsystem.cfgファイルをリモートサーバからAutoIT-VNF VMにコピーします。この例では、AutoDeployからAutoIT-VNF VMにコピーされます。

```
[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の設定で参照されるように、AutoIT-VNF上の適切な場所にファイルをコピーします。こちらをご覧ください。

```
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. VMが正常に復元されて実行されている場合は、以前の正常な既知のバックアップからすべてのsyslog固有の設定が復元されていることを確認します。

```
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リカバリ

1. VMがエラーまたはシャットダウン状態の場合、AutoVNF VMは回復可能です。ハードリブートを実行して、影響を受けるVMを起動します。AutoVNFを回復するには、次の手順を実行します。

2.エラーまたはシャットダウン状態のVMを特定します。AutoVNF VMをハードリブートします。

この例では、**auto-testautovnf1-uas-2**をリブートします。

```
[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. VMが起動したら、VMがクラスタに戻ることを確認します。

```
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 VMを上記の手順で回復できない場合は、次の手順に従って回復する必要があります

o

```
[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 VMを回復するには、uas-checkスクリプトを実行して状態を確認してください。エラーを報告する必要があります。次に、—fixオプションを使用して再度実行し、欠落しているUAS VMを再作成します。

```
[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. マスターAutoVNF VMにログインします。リカバリの数分以内に、新しく作成されたインスタンスがクラスタに参加し、アライブ状態になる必要があります。

```
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. VMが正常に復元されて実行されている場合は、以前の正常な既知のバックアップからすべてのsyslog固有の設定が復元されていることを確認します。すべてのAutoVNF VMで復元されていることを確認します。

```
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リカバリ

1. ESC VMは、VMがエラーまたはシャットダウン状態の場合は回復可能です。影響を受けるVMを起動するには、ハードリブートを実行してください。ESCを回復するには、次の手順を実行します。

2. エラーまたはシャットダウン状態のVMを特定し、ESC VMをハードリブートします。この例では、auto-test-vnfm1-ESC-0がリブートされます。

```
[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 VMが削除され、再度起動する必要がある場合は、次の手順に従います。

```
[stack@pod1-ospd scripts]$ nova list |grep ESC-1
| c566efbf-1274-4588-a2d8-0682e17b0d41 | vnf1-ESC-ESC-
1 | ACTIVE | - | running | vnf1-
UAS-uas-orchestration=172.168.11.14; vnf1-UAS-uas-
management=172.168.10.4
|
```

```
[stack@pod1-ospd scripts]$ nova delete vnf1-ESC-ESC-1
```

Request to delete server vnf1-ESC-ESC-1 has been accepted.

4. AutoVNF-UASからESC展開トランザクションを検索し、トランザクションのログで boot_vm.pyコマンド行を検索して、ESCインスタンスを作成します。

```
ubuntu@vnf1-uas-uas-0:~$ sudo -i
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 transaction
```

TX ID TIMESTAMP	TX TYPE STATUS	DEPLOYMENT ID
35eefc4a-d4a9-11e7-bb72-fa163ef8df2b 2017-11-29T02:01:27.750692-00:00	vnf-deployment deployment-success	vnf1-DEPLOYMENT
73d9c540-d4a8-11e7-bb72-fa163ef8df2b 2017-11-29T01:56:02.133663-00:00	vnfm-deployment deployment-success	vnf1-ESC

```
vnf1-uas-uas-0#show logs 73d9c540-d4a8-11e7-bb72-fa163ef8df2b | display xml
```

```
<config xmlns="http://tail-f.com/ns/config/1.0">
  <logs xmlns="http://www.cisco.com/usp/nfv/usp-autovnf-oper">
    <tx-id>73d9c540-d4a8-11e7-bb72-fa163ef8df2b</tx-id>
    <log>2017-11-29 01:56:02,142 - VNF Deployment RPC triggered for deployment: vnf1-ESC,
deactivate: 0
```

```
2017-11-29 01:56:02,179 - Notify deployment
```

```
..
2017-11-29 01:57:30,385 - Creating VNF '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_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]...
```

5. boot_vm.py行をシェルスクリプトファイル(esc.sh)に保存し、すべてのユーザ名*****とパスワード*****行を正しい情報(通常はcore/Cisco@123)で更新します。-encrypt_keyオプションも削除する必要があります。user_passおよびuser_confid_passの場合は、-user_passwd username:password(例: admin:Cisco@123)の形式を使用する必要があります。

次に、running-configからbootvm.pyへのURLを探し、bootvm.pyファイルをautovnf-uas VMに取得します。この場合、10.1.1.3はAuto-ITです。

```
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スクリプトをオプションで実行するシェルスクリプトを実行します。

```
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	02T13:28:32.000000	2017-12-
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	02T13:27:49Z	2017-12-
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
e859f6642ab6 | 2601b8ec-8ff8-4285-810a-

| 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
1 | vnf1-esc-esc-

| os-extended-volumes:volumes_attached | []

| progress
0

| security_groups
"default"] | [{"name": "default"}, {"name":

| status
ACTIVE

| tenant_id
fd4b15df46c6469cbacf5b80dcc98a5c

| updated
02T13:28:32Z | 2017-12-

| user_id
d3b51d6f705f4826b22817f27505c6cd

```

7. OSPDから、新しいESC VMがアクティブ/実行中であることを確認します。

```
[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 VMが回復不能で、データベースの復元が必要な場合は、以前に取得したバックアップからデータベースを復元してください。

9. ESCデータベースを復元するには、データベースを復元する前にESCサービスが停止していることを確認します。ESC HAの場合は、最初にセカンダリVMで、次にプライマリVMで実行します。

```
# service keepalived stop
```

10. ESCサービスのステータスを確認し、HAのプライマリVMとセカンダリVMの両方ですべての機能が停止していることを確認します。

```
# escadm status
```

11. データベースを復元するためにスクリプトを実行します。新しく作成されたESCインスタンスへのDBの復元の一環として、ツールはインスタンスの1つをプライマリESCに昇格し、そのDBフォルダをDRBDデバイスにマウントして、PostgreSQLデータベースを起動します。

```
# /opt/cisco/esc/esc-scripts/esc_dbtool.py restore --file
scp://<username>:<password>@<backup_vm_ip>:<filename>
```

12. ESCサービスを再起動して、データベースの復元を完了します。

13.両方のVMでHAを実行するには、キープアライブサービスを再起動します。

```
# service keepalived start
```

14. VMが正常に復元されて実行されると、以前の正常な既知のバックアップからすべてのsyslog固有の設定が復元されていることを確認します。すべてのESC VMで復元されていることを確認します。

```
[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リカバリ

1. EM VMが、1つまたはその他の条件によりNone/Error状態になっている場合、ユーザは該当するEM VMを回復するために所定の順序に従うことができます。

2. ESC/VNFMは、EM VMを監視するコンポーネントです。そのため、EMがエラー状態の場合、ESCはEM VMの自動回復を試みます。何らかの理由で、ESCがリカバ리를正常に完了できない場合、ESCはそのVMをエラー状態としてマークします。

3.このようなシナリオでは、基盤となるインフラストラクチャの問題が修正されると、ユーザはEM VMを手動でリカバリできます。根本的な問題が修正された後に限り、この手動回復を実行することが重要です。

4.エラー状態のVMを特定します。

```
[stack@pod1-ospd ~]$ source corerc  
[stack@pod1-ospd ~]$ nova list --field name,host,status |grep -i err  
| c794207b-a51e-455e-9a53-3b8ff3520bb9 | vnfm1-DEPLOYMENT-_vnfm1-D_0_a6843886-77b4-4f38-b941-  
74eb527113a8 | None | ERROR |
```

5. ESC Masterにログインし、影響を受けるEMおよびCF VMごとにrecovery-vm-actionを実行します。我慢しなさい。ESCはリカバリアクションをスケジュールし、数分間実行されない可能性があります。

```
ubuntu@vnfm1-uas-uas-1:~$ ssh admin@172.168.10.3
```

...

```
[admin@vnfm1-esc-esc-0 ~]$ sudo /opt/cisco/esc/esc-confd/esc-cli/esc_nc_cli recovery-vm-action DO  
vnfm1-DEPLOYMENT-_vnfm1-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.コマンドが完了するまで/var/log/esc/yangesc.logを監視します。

```
[admin@vnmf1-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 [vnmf1-DEPLOYMENT-
_vnmf1-D_0_a6843886-77b4-4f38-b941-74eb527113a8]
```

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

```
ubuntu@vnmf1vnmfddeploymentem-1:~$ /opt/cisco/ncs/current/bin/ncs_cli -u admin -C
```

```
admin connected from 172.17.180.6 using ssh on vnmf1vnmfddeploymentem-1
```

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

```
EM          VNFM

ID  SLA  SCM  PROXY
-----
2   up   up   up
3   up   up   up
```

ESCがVMの起動に失敗した場合

1. 予期しない状態が原因で、ESCがVMの起動に失敗する場合があります。回避策は、マスターESCをリブートしてESCスイッチオーバーを実行することです。ESCスイッチオーバーには約1分かかります。新しいマスターESCでhealth.shを実行し、起動しているかどうかを確認します。ESCがマスターになると、ESCがVMの状態を修正してVMを起動する場合があります。この操作はスケジュールされているため、完了するまで5～7分待つ必要があります。

2. /var/log/esc/yangesc.logと/var/log/esc/escmanager.logを監視できます。5～7分後にVMがリカバリされない場合は、影響を受けるVMを手動でリカバリする必要があります。

3. VMが正常に復元されて実行されると、以前の正常な既知のバックアップから、すべてのsyslog固有の設定が復元されていることを確認します。すべてのESC VMで復元されていることを確認します。

```
root@abautotestvnmf1em-0:/etc/rsyslog.d# pwd
/etc/rsyslog.d
```

```
root@abautotestvnmf1em-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-vnmf-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@abautotestvnmflem-0:/etc/rsyslog.d# ls /etc/rsyslog.conf
rsyslog.conf
```

StarOSリカバリ

1. StarOS VMの1つがNone/Error状態になった場合、ユーザは影響を受けるStarOS VMを回復するために、この順序に従うことができます。

2. ESC/VNFMはStarOS VMを監視するコンポーネントであるため、CF/SF VMがエラー状態の場合、ESCはCF/SF VMの自動回復を試みます。何らかの理由で、ESCがリカバ리를正常に完了できない場合、ESCはそのVMをエラー状態としてマークします。

3.このようなシナリオでは、基盤となるインフラストラクチャの問題が修正されると、ユーザはCF/SF VMを手動でリカバリできます。この手動回復は、根本的な問題を修正した後でのみ実行することが重要です。

4.エラー状態のVMを特定します。

```
[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 Masterにログインし、影響を受けるEMおよびCF VMごとにrecovery-vm-actionを実行します。ESCはリカバリアクションをスケジュールし、数分間実行されない可能性があります。

```
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.また、StarOSでshow cardタブを実行して、同じことを確認します。リカバリされたVMがSFの場合、必要に応じてアクティブにする必要があります。必要なStarOS設定を変更します。

```
[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
```

ESCがVMの起動に失敗した場合

場合によっては、予期しない状態が原因で、ESCがVMの起動に失敗することがあります。回避策は、マスターESCをリポートしてESCスイッチオーバーを実行することです。ESCスイッチオーバーには約1分かかります。新しいマスタESCでhealth.shを実行し、起動していることを確認します。ESCがマスターになると、ESCがVMの状態を修正してVMを起動する場合があります。この操作はスケジュールされているため、完了するまで5～7分待つ必要があります。

/var/log/esc/yangesc.logと/var/log/esc/escmanager.logを監視できます。5～7分後にVMが回復しない場合は、影響を受けるVMを手動でリカバリする必要があります。