

# さまざまなUltra-Mコンポーネントのバックアップおよびリストア手順 – CPS

## 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](#)

[ESCバックアップ](#)

[CPSバックアップ](#)

[復元手順](#)

[OSPDの回復](#)

[ESCリカバリ](#)

[CPSリカバリ](#)

## 概要

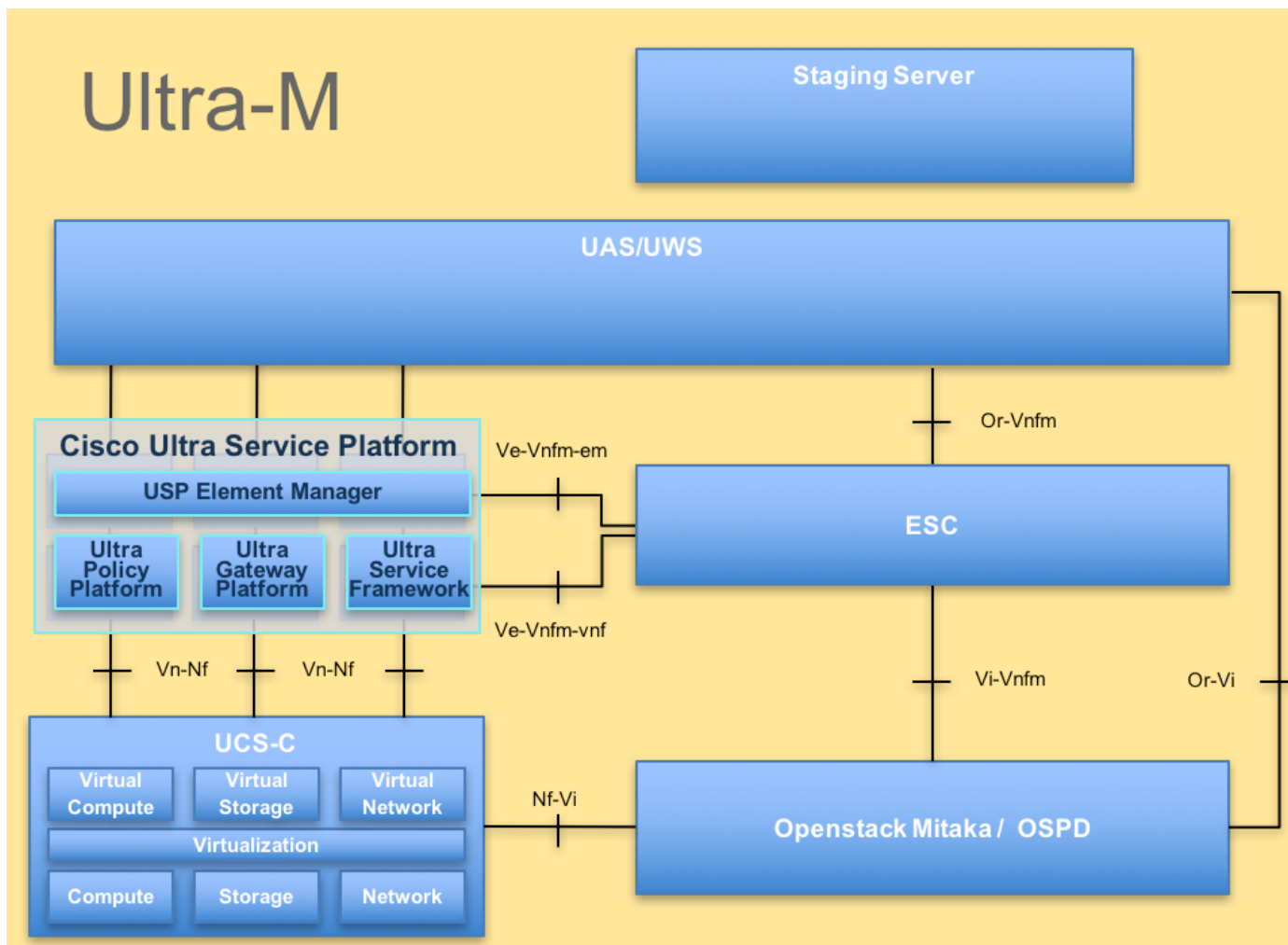
このドキュメントでは、CPS仮想ネットワーク機能(VNF)をホストするUltra-Mセットアップで仮想マシン(VM)をバックアップおよび復元するために必要な手順について説明します。

## 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。

- Elastic Services Controller ( ESC )
- Cisco Policy Suite(CPS)

この図に示すように、Ultra-Mの高レベルのアーキテクチャと関連するコンポーネントを示します。



注：Ultra M 5.1.xリリースは、このドキュメントの手順を定義するために検討されています。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の担当者を対象としています。

## 省略形

VNF	仮想ネットワーク機能
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
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、すべてのアンダークラウド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.これらのコマンドをrootユーザーとして実行し、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
```

## ESCバックアップ

1. ESCは、次に、VIMと対話することで仮想ネットワーク機能(VNF)を起動します。

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、Ultra-Mで1つの障害をサポートします。つまり、システムに1つの障害がある場合はシステムを回復します。

**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。

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、特定のVNF/POD/Siteでの次のアクティビティの後に実行することをお勧めします

4. health.shスクリプトを使用して、ESCの健全性が良好であることを確認します。

```
[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ESCデータベース

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

```
[admin@esc ~]# sudo bash
[root@esc ~]# cp /opt/cisco/esc/esc-scripts/esc_dbtool.py /opt/cisco/esc/esc-
scripts/esc_dbtool.py.bkup
[root@esc esc-scripts]# 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@esc esc-scripts]# escadm op_mode set --mode=maintenance
```

**2. 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@esc esc-scripts]# escadm op_mode show
```

3. ESCで使用できるデータベースバックアップ復元ツールを使用してデータベースをバックアップします。

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

4. ESCを[Operation Mode]に戻し、モードを確認します。

```
[root@esc scripts]# escadm op_mode set --mode=operation
```

```
[root@esc scripts]# escadm op_mode show
```

5. scriptsディレクトリに移動し、ログを収集します。

```
[root@esc scripts]# /opt/cisco/esc/esc-scripts
```

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

6. ESCのスナップショットを作成するには、まずESCをシャットダウンします。

```
shutdown -r now
```

7. OSPDからイメージスナップショットを作成します

```
nova image-create --poll escl esc_snapshot_27aug2018
```

8. スナップショットが作成されていることを確認します

```
openstack image list | grep esc_snapshot_27aug2018
```

9. OSPDからESCを起動します

```
nova start escl
```

10. 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

11. 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する

```
[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
```

## CPSバックアップ

## 1. CPS Cluster-Managerのバックアップの作成

次のコマンドを使用して、novaインスタンスを表示し、クラスタマネージャVMインスタンスの名前をメモします。

```
nova list
```

Escキーを押して列を停止します

```
/opt/cisco/esc/esc-confd/esc-cli/esc_nc_cli vm-action STOP <vm-name>
```

ステップ2：クラスタマネージャがシャットオフ状態であることを確認します。

```
admin@escl ~]$ /opt/cisco/esc/confd/bin/confd_cli
```

```
admin@escl> show esc_datamodel opdata tenants tenant Core deployments * state_machine
```

ステップ3：次のコマンドに示すように、novaスナップショットイメージを作成します。

```
nova image-create --poll
```

注：スナップショットに十分なディスク領域があることを確認します。

重要：スナップショットの作成後にVMが到達不能になった場合は、nova listコマンドを使用してVMのステータスを確認します。「シャットオフ」状態の場合は、VMを手動で開始する必要があります。

ステップ4：次のコマンドを使用してイメージリストを表示します。nova image-list図1:出力例

ID	Name	Status	Server
146719e8-d8a0-4d5a-9b15-2a669cfab81f	CPS_10.9.9_20160803_100301_112.iso	ACTIVE	
1955d56e-4ecf-4269-b53d-b30e73ad57f0	base_vm	ACTIVE	
2bbfb51c-cd05-4b7c-ad77-8362d76578db	cluman_snapshot	ACTIVE	4842ae5a-83a3-48fd-915b-6ca6361adb2c

ステップ5：スナップショットが作成されると、スナップショットイメージがOpenStack Glanceに保存されます。スナップショットをリモートデータストアに保存するには、スナップショットをダウンロードし、OSPDのファイルに転送します(/home/stack/CPS\_BACKUP)

イメージをダウンロードするには、OpenStackで次のコマンドを使用します。

```
glance image-download --file For example: glance image-download --file snapshot.raw 2bbfb51c-cd05-4b7c-ad77-8362d76578db
```

ステップ6：次のコマンドに示すように、ダウンロードしたイメージをリストします。

```
ls -ltr *snapshot*
```

```
Example output: -rw-r--r--. 1 root root 10429595648 Aug 16 02:39 snapshot.raw
```

ステップ7：将来リストアするCluster Manager VMのスナップショットを保存します。

## 2.設定とデータベースをバックアップします。

1. config\_br.py -a export --all /var/tmp/backup/ATP1\_backup\_all\_\$(date +%Y-%m-%d).tar.gz OR
2. config\_br.py -a export --mongo-all /var/tmp/backup/ATP1\_backup\_mongoall\$(date +%Y-%m-%d).tar.gz
3. config\_br.py -a export --svn --etc --grafanadb --auth-htpasswd --haproxy /var/tmp/backup/ATP1\_backup\_svn\_etc\_grafanadb\_haproxy\_\$(date +%Y-%m-%d).tar.gz
4. mongodump - /var/qps/bin/support/env/env\_export.sh --mongo /var/tmp/env\_export\_\$(date +%Y-%m-%d).tgz
5. patches - cat /etc/broadhop/repositories, check which patches are installed and copy those patches to the backup directory /home/stack/CPS\_BACKUP on OSPD
6. backup the cronjobs by taking backup of the cron directory: /var/spool/cron/ from the Pcrfclient01/Cluman. Then move the file to CPS\_BACKUP on the OSPD.

他のバックアップが必要な場合は、 crontab -lから確認します

すべてのバックアップをOSPD /home/stack/CPS\_BACKUPに転送します

## 3. ESCマスターからのyamlファイルのバックアップ

```
/opt/cisco/esc/confd/bin/netconf-console --host 127.0.0.1 --port 830 -u <admin-user> -p <admin-password> --get-config > /home/admin/ESC_config.xml
```

OSPD /home/stack/CPS\_BACKUPでファイルを転送します

## 4. crontab -lエントリのバックアップ

crontab -lを使用してtxtファイルを作成し、リモートの場所 ( OSPD /home/stack/CPS\_BACKUP内 ) にftpします

## 5. LBおよびPCRFクライアントからルートファイルのバックアップを取る

```
Collect and scp the below configurations from both LBs and Pcrfclients  
route -n /etc/sysconfig/network-script/route-*
```

# 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

1. OSPDバックアップは古いOSPDサーバから利用できます。
2. 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。

## 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ハードリブートして影響を受けるVMを起動するとリカバリできます。ESCを回復するには、次の手順を実行します。

2. ESC VMのハードリブートが確認されたら、エラーまたはシャットダウン状態のVMを特定します。この例では、auto-test-vnfm1-ESC-0をリブートしています。

```
[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。次の手順に従います

```
[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. 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。

5. 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する必要があります。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
```

6. 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の両方ですべてが停止していることを確認します。

```
# escadm status
```

7. スクリプトを実行してデータベースを復元します。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>
```

8. 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。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
```

9. 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
```

10. ESCをOSPDスナップショットから再構築する必要がある場合は、バックアップ中に取得したスナップショットを使用してこのコマンドを使用します。

```
nova rebuild --poll --name esc_snapshot_27aug2018 esc1
```

11.再構築が完了したら、ESCのステータスを確認します

```
nova list --fileds name,host,status,networks | grep esc
```

12.次のコマンドでESCの状態を確認します

```
health.sh
```

```
Copy Datamodel to a backup file
```

```
/opt/cisco/esc/esc-confd/esc-cli/esc_nc_cli get esc_datamodel/opdata > /tmp/esc_opdata_`date +%Y%m%d%H%M%S`.txt
```

## 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、起動していることを確認します。ESCがマスターになると、ESCはVMの状態を修正し、VMを起動できます。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。
- VMが正常に復元されて稼働したら、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

認めます

```
root@abautotestvnm1em-0:/etc/rsyslog.d# pwd
/etc/rsyslog.d
```

```
root@abautotestvnm1em-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@abautotestvnm1em-0:/etc/rsyslog.d# ls /etc/rsyslog.conf
rsyslog.conf
```

## CPSリカバリ

### OpenStackのCluster Manager VMの復元

ステップ1: 次のコマンドに示すように、クラスタマネージャVMスナップショットをコントローラブレードにコピーします。

```
ls -ltr *snapshot*
```

Example output: -rw-r--r--. 1 root root 10429595648 Aug 16 02:39 snapshot.raw

ステップ2 DatastoreからOpenStackにスナップショットイメージをアップロードします。

```
glance image-create --name --file --disk-format qcow2 --container-format bare
```

ステップ3次の例に示すように、スナップショットがNovaコマンドでアップロードされているかどうかを確認します。

```
nova image-list
```

図 2 : 出力例

ID	Name	Status	Server
146719e8-d8a0-4d5a-9b15-2a669cfab81f	CPS_10.9.9_20160803_100301_112.iso	ACTIVE	
1955d56e-4ecf-4269-b53d-b30e73ad57f0	base_vm	ACTIVE	
2bbfb51c-cd05-4b7c-ad77-8362d76578db	cluman_snapshot	ACTIVE	4842ae5a-83a3-48fd-915b-6ca6361adb2c
5eebff44-658a-49a5-a170-1978f6276d18	imported_image	ACTIVE	

ステップ4クラスタマネージャVMが存在するかどうかによって、次の手順でカラムを作成するか、カラムを再構築するかを選択できます。

- Cluster Manager VMインスタンスが存在しない場合は、次の例に示すように、HeatまたはNovaコマンドを使用してCluman VMを作成します。

ESCを使用して列VMを作成する

```
/opt/cisco/esc/esc-confd/esc-cli/esc_nc_cli edit-config /opt/cisco/esc/cisco-cps/config/gr/tmo/gen/<original_xml_filename>
```

PCRFクラスタは、上記のコマンドを使用して起動し、次にconfig\_br.py restoreで取得したバックアップからクラスタマネージャの設定を復元し、バックアップで取得したダンプからmongorestoreします

```
delete - nova boot --config-drive true --image "" --flavor "" --nic net-id="v4-fixed-ip=" --nic net-id="network_id,v4-fixed-ip=ip_address" --block-device-mapping "/dev/vdb=2edbac5e-55de-4d4c-a427-ab24ebe66181:::0" --availability-zone "az-2:megh-os2-compute2.cisco.com" --security-groups cps_secgrp "cluman"
```

・ Cluster Manager VMインスタンスが存在する場合は、nova rebuildコマンドを使用して、アップロードされたスナップショットを使用してCluman VMインスタンスを再構築します。

```
nova rebuild <instance_name> <snapshot_image_name>
```

以下に、いくつかの例を示します。

```
nova rebuild cps-cluman-5f3tujqvbi67 cluman_snapshot
```

ステップ5次に示すように、すべてのインスタンスをリストし、新しいCluster Managerインスタンスが作成されて実行されていることを確認します。

```
nova list
```

図3.出力例

ID	Name	Status	Task State	Power State	Networks
ac3d2dbc-7b0e-4df4-a690-7f84ca3032bd	cluman	ACTIVE	-	Running	management=172.20.67.34; internal=172.20.70.34

## システム上の最新のパッチを復元する

- Copy the patch files to cluster manager which were backed up in OSPD  
/home/stack/CPS\_BACKUP
- Login to the Cluster Manager as a root user.
- Untar the patch by executing the following command: tar -xvzf [patch name].tar.gz
- Edit /etc/broadhop/repositories and add the following entry: file:/// \$path\_to\_the plugin/[component name]
- Run build\_all.sh script to create updated QPS packages:  
/var/qps/install/current/scripts/build\_all.sh
- Shutdown all software components on the target VMs: runonall.sh sudo monit stop all
- Make sure all software components are shutdown on target VMs: statusall.sh

注：ソフトウェアコンポーネントはすべて、現在のステータスとして[Not Monitored]と表示されている必要があります。

- Update the qns VMs with the new software using reinit.sh script:  
/var/qps/install/current/scripts/upgrade/reinit.sh
- Restart all software components on the target VMs: runonall.sh sudo monit start all
- Verify that the component is updated, run: about.sh

## クロジョブの復元

1.バックアップしたファイルをOSPDからCluman/Pcrfclient01に移動します。

2.コマンドを実行して、バックアップからcronjobをアクティブにします。

```
#crontab Cron-backup
```

3.以下のコマンドでcronjobsがアクティブになっているかどうかを確認します。

```
#crontab -l
```

## クラスタ内の個々のVMのリストア

pcrfclient01 VMを再配置するには、次の手順を実行します。

ステップ1 Cluster Manager VMにルートユーザとしてログインします。

手順2次のコマンドを使用して、SVNリポジトリのUUIDをメモします。

```
svn info http://pcrfclient02/repos | grep UUID
```

このコマンドは、リポジトリのUUIDを出力します。

以下に、いくつかの例を示します。リポジトリUUID:ea50bbd2-5726-46b8-b807-10f4a7424f0e

ステップ3次の例に示すように、Cluster Managerでバックアップポリシービルダーの設定データをインポートします。

```
config_br.py -a import --etc-oam --svn --stats --grafanadb --auth-htpasswd --users  
/mnt/backup/oam_backup_27102016.tar.gz
```

**注：**多くの導入設定データを定期的にバックアップするcronジョブを実行します。詳細はSubversion Repository Backupを参照してください。

ステップ4 Cluster Managerで最新の設定を使用してVMアーカイブファイルを生成するには、次のコマンドを実行します。

```
/var/qps/install/current/scripts/build/build_svn.sh
```

ステップ5 pcrfclient01 VMを導入するには、次のいずれかを実行します。

OpenStackで、HEATテンプレートまたはNovaコマンドを使用してVMを再作成します。詳細については、『CPS Installation Guide for OpenStack』を参照してください。

ステップ6 pcrfclient01とpcrfclient02の間のSVNマスター/スレーブ同期を、次の一連のコマンドを

実行してマスターとして再確立します。

SVNがすでに同期されている場合は、これらのコマンドを発行しないでください。

SVNが同期しているかどうかを確認するには、pcrfclient02から次のコマンドを実行します。

値が返された場合、SVNはすでに同期されています。

```
/usr/bin/svn propget svn:sync-from-url --revprop -r0 http://pcrfclient01/repos  
pcrfclient01から次のコマンドを実行します。
```

```
/bin/rm -fr /var/www/svn/repos
```

```
/usr/bin/svnadmin create /var/www/svn/repos
```

```
/usr/bin/svn propset --revprop -r0 svn:sync-last-merged-rev 0 http://pcrfclient02/repos-proxy-  
sync
```

```
/usr/bin/svnadmin setuuid /var/www/svn/repos/ "Enter the UUID captured in step 2"
```

```
/etc/init.d/vm-init-client /
```

```
var/qps/bin/support/recover_svn_sync.sh
```

ステップ7 pcrfclient01がアービターVMでもある場合は、次の手順を実行します。

a)システム設定に基づいてmongodbの開始/停止スクリプトを作成します。すべての導入で、これらのデータベースがすべて設定されているわけではありません。

注：どのデータベースをセットアップする必要があるかを確認するには、  
/etc/broadhop/mongoConfig.cfgを参照してください。

```
cd /var/qps/bin/support/mongo
```

```
build_set.sh --session --create-scripts
```

```
build_set.sh --admin --create-scripts
```

```
build_set.sh --spr --create-scripts
```

```
build_set.sh --balance --create-scripts
```

```
build_set.sh --audit --create-scripts
```

```
build_set.sh --report --create-scripts
```

b)mongoプロセスを開始します。

```
/usr/bin/systemctl start sessionmgr-XXXXX
```

c)アービタが起動するまで待ち、diagnostics.sh —get\_replica\_statusを実行して、レプリカセットの状態を確認します。

pcrfclient02 VMを再配置するには、次の手順を実行します。

ステップ1 Cluster Manager VMにルートユーザとしてログインします

ステップ2 Cluster Managerで最新の設定を使用してVMアーカイブファイルを生成するには、次のコマンドを実行します。

```
/var/qps/install/current/scripts/build/build_svn.sh
```

ステップ3 pcrfclient02 VMを導入するには、次のいずれかを実行します。  
OpenStackで、HEATテンプレートまたはNovaコマンドを使用してVMを再作成します。詳細については、『CPS Installation Guide for OpenStack』を参照してください。

ステップ4 pcrfclient01へのシエルを保護します。

```
ssh pcrfclient01
```

ステップ5 pcrfclient01からSVNリポジトリを回復するには、次のスクリプトを実行します。

```
/var/qps/bin/support/recover_svn_sync.sh
```

**sessionmgr VMを再配置するには、次の手順を実行します。**

ステップ1 Cluster Manager VMにルートユーザとしてログインします

ステップ2 sessionmgr VMを展開し、失敗または破損したVMを置き換えるには、次のいずれかを実行します。

OpenStackで、HEATテンプレートまたはNovaコマンドを使用してVMを再作成します。詳細については、『CPS Installation Guide for OpenStack』を参照してください

ステップ3 : システム設定に基づいて、mongodbの開始/停止スクリプトを作成します。

すべての導入で、これらのデータベースがすべて設定されているわけではありません。どのデータベースをセットアップする必要があるかを確認するには、/etc/broadhop/mongoConfig.cfgを参照してください

```
cd /var/qps/bin/support/mongo
```

```
build_set.sh --session --create-scripts  
build_set.sh --admin --create-scripts  
build_set.sh --spr --create-scripts  
build_set.sh --balance --create-scripts  
build_set.sh --audit --create-scripts  
build_set.sh --report --create-scripts
```

ステップ4 sessionmgr VMへのシエルを保護し、mongoプロセスを開始します。

```
ssh sessionmgrXX
```

```
/usr/bin/systemctl start sessionmgr-XXXXX
```

ステップ5メンバーが開始し、セカンダリメンバーが同期するまで待つから、diagnostics.sh  
—get\_replica\_statusを実行して、データベースの状態を確認します。

ステップ6 Session Managerデータベースをリストアするには、—mongo-allまたは—mongoオプションを使用してバックアップが実行されたかどうかに応じて、次のいずれかのコマンド例を使

用します。

```
• config_br.py -a import --mongo-all --users /mnt/backup/Name of backup
```

or

```
• config_br.py -a import --mongo --users /mnt/backup/Name of backup
```

**Policy Director (ロードバランサ) VMを再配置するには、次の手順に従います。**

ステップ1 Cluster Manager VMにルートユーザとしてログインします。

ステップ2 Cluster Managerでバックアップポリシービルダーの設定データをインポートするには、次のコマンドを実行します。

```
config_br.py -a import --network --haproxy --users /mnt/backup/lb_backup_27102016.tar.gz
```

ステップ3 Cluster Managerで最新の設定を使用してVMアーカイブファイルを生成するには、次のコマンドを実行します。

```
/var/qps/install/current/scripts/build/build_svn.sh
```

ステップ4 lb01 VMを導入するには、次のいずれかを実行します。

OpenStackで、HEATテンプレートまたはNovaコマンドを使用してVMを再作成します。詳細については、『CPS Installation Guide for OpenStack』を参照してください。

**ポリシーサーバ(QNS)VMを再導入するには、次の手順を実行します。**

ステップ1 Cluster Manager VMにルートユーザとしてログインします。

ステップ2次の例に示すように、Cluster Managerでバックアップポリシービルダーの設定データをインポートします。

```
config_br.py -a import --users /mnt/backup/qns_backup_27102016.tar.gz
```

ステップ3 Cluster Managerで最新の設定を使用してVMアーカイブファイルを生成するには、次のコマンドを実行します。

```
/var/qps/install/current/scripts/build/build_svn.sh
```

ステップ4 qns VMを導入するには、次のいずれかを実行します。

OpenStackで、HEATテンプレートまたはNovaコマンドを使用してVMを再作成します。詳細については、『CPS Installation Guide for OpenStack』を参照してください

## データベース復元の一般的な手順

ステップ1次のコマンドを実行して、データベースを復元します。



`config_br.py -a import --mongo-all /mnt/backup/backup_$(date +%Y%m%d).tar.gz` where `$(date +%Y%m%d)` is the timestamp when the export was made.

たとえば、

```
config_br.py -a import --mongo-all /mnt/backup/backup_27092016.tgz
```

ステップ2データベースにログインし、実行中でアクセス可能かどうかを確認します。

1.セッションマネージャにログインします。

```
mongo --host sessionmgr01 --port $port
```

ここで、`$port`はチェックするデータベースのポート番号です。たとえば、27718がデフォルトのバランスポートです。

2.次のコマンドを実行して、データベースを表示します。

```
show dbs
```

3.次のコマンドを実行して、mongoシェルをデータベースに切り替えます。

```
use $db
```

`$db`は、前のコマンドで表示されたデータベース名です。

「use」コマンドは、mongoシェルをそのデータベースに切り替えます。

たとえば、

```
use balance_mgmt
```

4.コレクションを表示するには、次のコマンドを実行します。

```
show collections
```

5.コレクション内のレコード数を表示するには、次のコマンドを実行します。

```
db.$collection.count()
```

```
For example, db.account.count()
```

上の例では、Balanceデータベース(`balance_mgmt`)のコレクション「`account`」内のレコード数を示します。

## Subversionリポジトリの復元

バックアップからPolicy Builder設定データを復元するには、次のコマンドを実行します。

```
config_br.py -a import --svn /mnt/backup/backup_$(date +%Y%m%d).tgz
```

 where, `$(date +%Y%m%d)` is the date when the cron created the backup file.

## グラフダッシュボードの復元

次のコマンドを使用して、Grafanaダッシュボードを復元できます。

```
config_br.py -a import --grafanadb /mnt/backup/
```

## 復元の検証

データを復元した後、次のコマンドを実行して作業システムを確認します。

```
/var/qps/bin/diag/diagnostics.sh
```