Backup and Restore Procedures for Various Ultra-M Components - CPS

Contents

Introduction

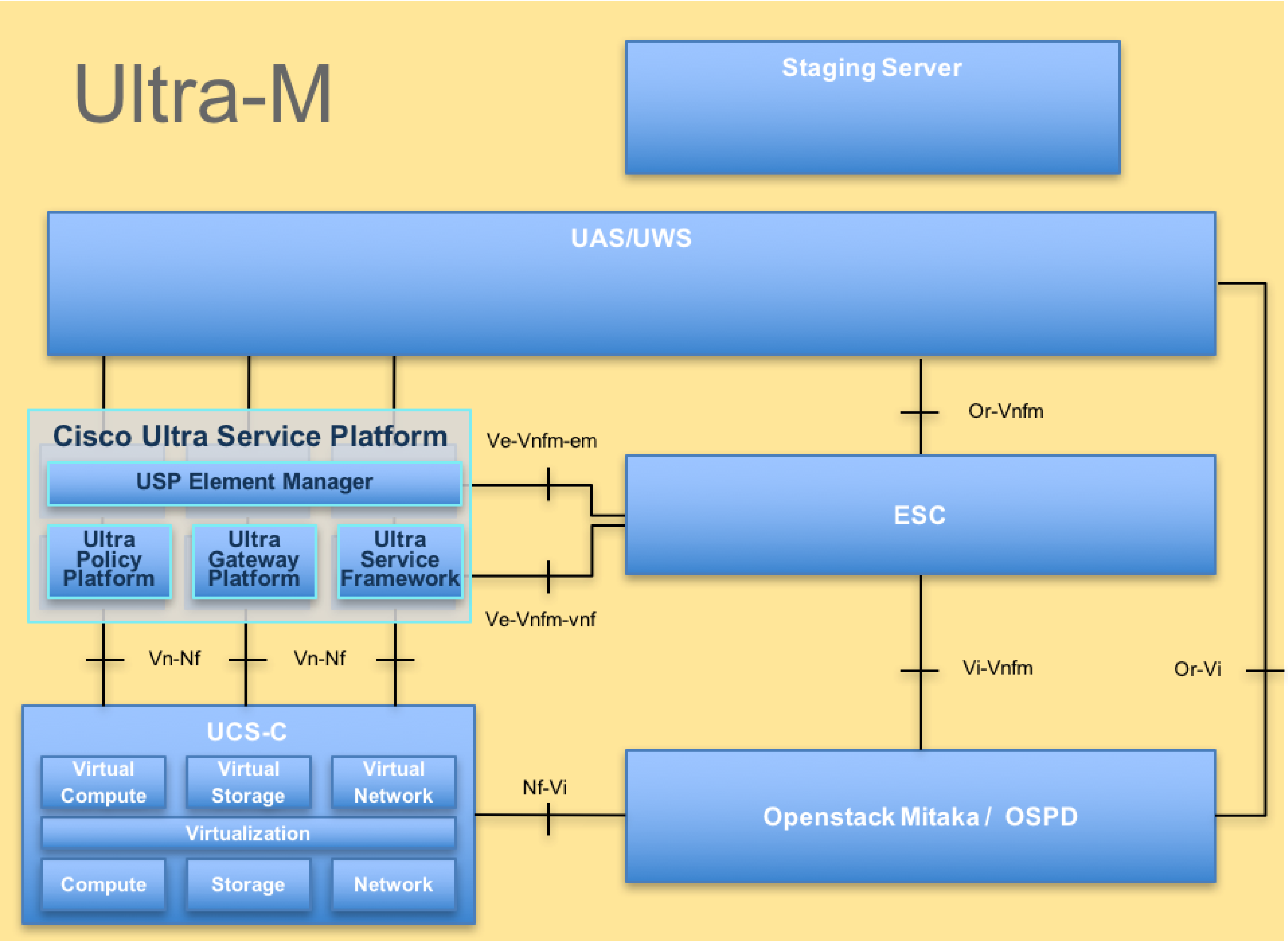
This document describes the steps required to Backup and Restore a Virtual Machine in an Ultra-M setup that hosts Calls CPS Virtual Network Functions.

Background Information

Ultra-M is a pre-packaged and validated, virtualized mobile packet core solution designed to simplify the deployment of Virtual Network Functions (VNFs). Ultra-M solution consists of these Virtual Machine (VM) types:

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

The high-level architecture of Ultra-M and the components involved are as shown in this image.


Ultra-M
Staging Server
UAS/UWS
Cisco Ultra Service Platform
USP Element Manager
Ultra Policy Platform
Ultra Gateway Platform
Ultra Service Framework
Ve-Vnfm-em
Ve-Vnfm-vnf
Or-Vnfm
ESC
Vn-Nf
Vn-Nf
Vi-Vnfm
Or-Vi
UCS-C
Virtual Compute
Virtual Storage
Virtual Network
Virtualization
Compute
Storage
Network
Nf-Vi
Openstack Mitaka / OSPD

Note: Ultra M 5.1.x release is considered in order to define the procedures in this document. This document is intended for the Cisco personnel who are familiar with Cisco Ultra-M platform.

Abbreviations

VNF	Virtual Network Function
ESC	Elastic Service Controller
MOP	Method of Procedure
OSD	Object Storage Disks
HDD	Hard Disk Drive
SSD	Solid State Drive
VIM	Virtual Infrastructure Manager
VM	Virtual Machine
UUID	Universally Unique IDentifier

Backup Procedure

OSPD Backup

1. Check the status of OpenStack stack and the node list.

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

2. Check if all the undercloud services are in loaded, active, and running status from the OSP-D node.

```
[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
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 servic
openstack-ceilometer-notification.service  loaded active running OpenStack ceilometer notification agen
openstack-glance-api.service               loaded active running OpenStack Image Service (code-named Gl
openstack-glance-registry.service          loaded active running OpenStack Image Service (code-named Gl
openstack-heat-api-cfn.service             loaded active running Openstack Heat CFN-compatible API Serv
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 In
openstack-ironic-inspector.service         loaded active running Hardware introspection service for Ope
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) - Acc
openstack-swift-account.service            loaded active running OpenStack Object Storage (swift) - Acc
openstack-swift-container-updater.service  loaded active running OpenStack Object Storage (swift) - Con
openstack-swift-container.service          loaded active running OpenStack Object Storage (swift) - Con
openstack-swift-object-updater.service     loaded active running OpenStack Object Storage (swift) - Obj
openstack-swift-object.service             loaded active running OpenStack Object Storage (swift) - Obj
openstack-swift-proxy.service              loaded active running OpenStack Object Storage (swift) - Pro
openstack-zaqar.service                    loaded active running OpenStack Message Queuing Service (cod
openstack-zaqar@1.service                  loaded active running OpenStack Message Queuing Service (cod
openvswitch.service                        loaded active exited  Open vSwitch

LOAD   = Reflects whether the unit definition was properly loaded.
ACTIVE = The high-level unit activation state, for example, 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. Confirm that you have sufficient disk space available before you perform the backup process. This tarball is expected to be at least 3.5 GB.

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

4. Execute these commands as the root user to backup the data from the undercloud node to a file named undercloud-backup-[timestamp].tar.gz and transfer it to the backup server.

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

ESC Backup

1. ESC, in turn, brings up Virtual Network Function (VNF) by interacting to VIM.

2. ESC has 1:1 redundancy in Ultra-M Solution. There are 2 ESC VMs deployed and support single failure in Ultra-M. for example, recover the system if there is a single failure in the system.

Note: If there is more than a single failure, it is not supported and can require redeployment of the system.

ESC backup details:

- Running configuration
- ConfD CDB DB
- ESC Logs
- Syslog configuration

3. The frequency of ESC DB backup is tricky and needs to be handled carefully as ESC monitors and maintains the various state machines for various VNF VMs deployed. It is advised that these backups are performed after these activities in given VNF/POD/Site.

4. Verify the health of ESC is good using health.sh script.

```
[root@auto-test-vnfm1-esc-0 admin]# escadm status
0 ESC status=0 ESC Primary 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 (Primary) with DRBD =================

DRBD_ROLE_CHECK=0
MNT_ESC_DATABSE_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. Take the backup of the Running configuration and transfer the file to the backup server.

```
[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
```

Backup ESC Database

1. Log into ESC VM and execute this command before you take the backup.

```
[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

#Set ESC to mainenance mode
[root@esc esc-scripts]# escadm op_mode set --mode=maintenance
```

2. Check ESC mode and ensure it is in maintenance mode.

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

3. Backup database using database backup restore tool available in ESC.

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

4. Set ESC Back to Operation Mode and confirm the mode.

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

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

5. Navigate to the scripts directory and collect the logs.

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

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

6. To create a snapshot of the ESC, first shutdown the ESC.

```
shutdown -r now
```

7. From OSPD, create an image snapshot.

```
nova image-create --poll esc1 esc_snapshot_27aug2018
```

8. Verify that the snapshot is created.

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

9. Start the ESC from OSPD.

```
nova start esc1
```

10. Repeat the same procedure on standby ESC VM and transfer the logs to backup server.

11. Collect syslog configuration backup on both the ESC VMS and transfer them to backup server.

```
[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 Backup

Step 1. Create a Backup of CPS Cluster-Manager.

Use this command in order to view the nova instances and note the name of the cluster manager VM instance:

```
nova list
```

Stop the Cluman from ESC.

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

Step 2. Verify Cluster Manager is in SHUTOFF state.

```
admin@esc1 ~]$ /opt/cisco/esc/confd/bin/confd_cli

admin@esc1> show esc_datamodel opdata tenants tenant Core deployments * state_machine
```

Step 3. Create a nova snapshot image as shown in the this command:

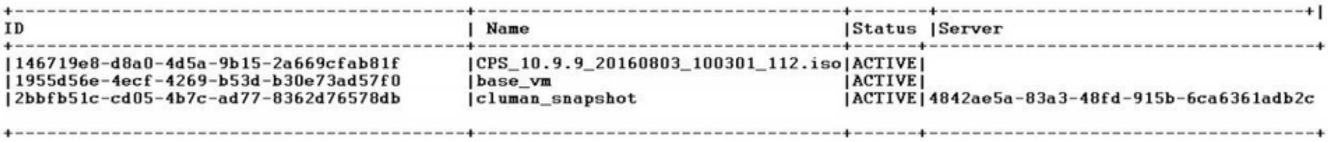
```
nova image-create --poll <cluman-vm-name> <snapshot-name>
```

Note: Ensure that you have enough disk space for the snapshot.

.Important - In case VM becomes unreachable after snapshot creation, check status of VM using nova list command. If it is in SHUTOFF state, you need to start the VM manually.

Step 4. View the image list with this command: nova image-list

Image 1: Example Output



```
+---------------------------------------+-----------------------------------+------+----------------------------------+|
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

+---------------------------------------+-----------------------------------+------+----------------------------------+
```

Step 5. When a snapshot is created, the snapshot image is stored in OpenStack Glance. To store the snapshot in a remote data store, download the snapshot and transfer the file in OSPD
to (/home/stack/CPS_BACKUP).

To download the image, use this command in OpenStack:

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

Step 6. List the downloaded images as shown in this command:

```
ls –ltr *snapshot*
```

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

Step 7. Store the snapshot of the Cluster Manager VM to restore in the future.

2. Backup the configuration and database.

```
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
4. mongodump - /var/qps/bin/support/env/env_export.sh --mongo /var/tmp/env_export_$date.tgz
5. patches - cat /etc/broadhop/repositories, check which patches are installed and copy those patches t
6. backup the cronjobs by taking backup of the cron directory: /var/spool/cron/ from the Pcrfclient01/C
```

Verify from the crontab-l if any other backup is needed.

Transfer all the backups to the OSPD /home/stack/CPS_BACKUP.

3. Backup yaml file from ESC Primary.

```
/opt/cisco/esc/confd/bin/netconf-console --host 127.0.0.1 --port 830 -u <admin-user> -p <admin-password
```

Transfer the file in OSPD /home/stack/CPS_BACKUP.

4. Back up crontab -l entries.

Create a txt file with crontab -l and ftp it to remote location (in OSPD /home/stack/CPS_BACKUP).

5. Take a backup of the route files from LB and PCRF client.

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

Restore Procedure

OSPD Recovery

OSPD recovery procedure is performed based on these assumptions.

1. OSPD backup is available from old OSPD server.

2. OSPD Recovery can be done on the new server which is the replacement of the old OSPD server in the system. .

ESC Recovery

1. ESC VM is recoverable if the VM is in error or shutdown state do hard reboot to bring up of the impacted VM. Execute these steps to recover ESC.

2. Identify the VM which is in ERROR or Shutdown state, once identified hard-reboot the ESC VM. In this example, you are rebooting auto-test-vnfm1-ESC-0.

```
[root@tb1-baremetal scripts]# nova list | grep auto-test-vnfm1-ESC-

| f03e3cac-a78a-439f-952b-045aea5b0d2c | auto-test-vnfm1-ESC-0                                         |
| 79498e0d-0569-4854-a902-012276740bce | auto-test-vnfm1-ESC-1                                         |

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

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

3. If ESC VM is deleted and needs to be brought up again. Use these sequence of steps.

```
[stack@pod1-ospd scripts]$ nova list |grep ESC-1
| c566efbf-1274-4588-a2d8-0682e17b0d41 | vnf1-ESC-ESC-1

[stack@pod1-ospd scripts]$ nova delete vnf1-ESC-ESC-1
Request to delete server vnf1-ESC-ESC-1 has been accepted.
```

4. If ESC VM is unrecoverable and requires the restore of the database, restore the database from the previously taken backup.

5. For ESC database restore, you have to ensure the esc service is stopped before restoring the database; for ESC HA, execute in secondary VM first, and then the primary VM.

```
# service keepalived stop
```

6. Check ESC service status and ensure everything is stopped in both Primary and Secondary VMs for HA.

```
# escadm status
```

7. Execute the script to restore the database. As part of the restoration of the DB to the newly created ESC instance, the tool can also promote one of the instances to be a primary ESC, mount its DB folder to the drbd device and can start the PostgreSQL database.

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

8. Restart ESC service to complete the database restore. For HA execute in both VMs, restart the keepalived service.

```
# service keepalived start
```

9. Once the VM is successfully restored and running; ensure all the syslog specific configuration is restored from the previous successful known backup. ensure it is restored in all the ESC VMs.

```
[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. If the ESC needs to be rebuilt from OSPD snapshot, use this command with the use of snapshot taken during backup.

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

11. Check the status of the ESC after rebuild is complete.

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

12. Check ESC health with this command.

```
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%
```

When ESC Fails to Start VM

- In some cases, ESC can fail to start the VM due to an unexpected state. A workaround is to perform an ESC switchover by rebooting the Primary ESC. The ESC switchover can take about a minute. Execute health.sh on the new Primary ESC to verify it is up. When the ESC becomes Primary, ESC can fix the VM state and start the VM. Since this operation is scheduled, you must wait 5-7 minutes for it to complete.
- You can monitor /var/log/esc/yangesc.log and /var/log/esc/escmanager.log. If you do NOT see VM getting recovered after 5-7 minutes, user would need to go and do the manual recovery of the impacted VM(s).
- Once the VM is successfully restored and running; ensure all the syslog specific configuration is restored from the previous successful known backup. Ensure it is restored in all the ESC VMs

```
root@abautotestvnfm1em-0:/etc/rsyslog.d# pwd
/etc/rsyslog.d

root@abautotestvnfm1em-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@abautotestvnfm1em-0:/etc/rsyslog.d# ls /etc/rsyslog.conf
rsyslog.conf
```

CPS Recovery

Restore Cluster Manager VM in OpenStack.

Step 1. Copy the cluster manager VM snapshot to the controller blade as shown in this command:

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

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

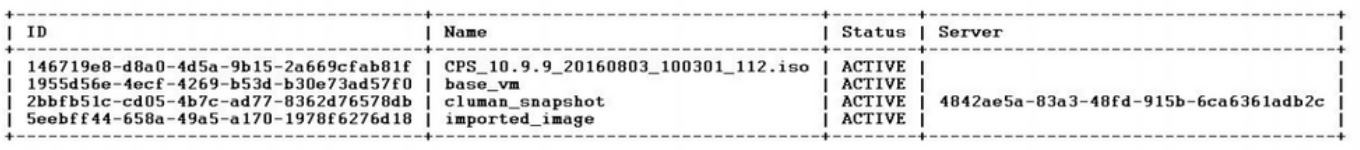
Step 2. Upload the snapshot image to OpenStack from Datastore:

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

Step 3. Verify whether the snapshot is uploaded with a Nova command as shown in this example:

```
nova image-list
```

Image 2: Example Output



```
+--------------------------------------+-----------------------------------+--------+--------------------------------------+
| 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 |                                      |
+--------------------------------------+-----------------------------------+--------+--------------------------------------+
```

Step 4. Depending on whether the cluster manager VM exists or not, you can choose to create the cluman or rebuild the cluman:

- If the Cluster Manager VM instance does not exist, create the Cluman VM with an Heat or Nova command as shown in this example:

Create the Cluman VM with ESC.

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

The PCRF cluster can spawn with the help of the previous command, and then restore the cluster manager configurations from the backups taken with config_br.py restore, mongorestore from dump taken in backup.

```
delete - nova boot --config-drive true --image "" --flavor "" --nic net-id=",v4-fixed-ip=" --nic net-id
```

• If the Cluster Manager VM instance exists, use a nova rebuild command to rebuild the Cluman VM instance with the uploaded snapshot as shown:

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

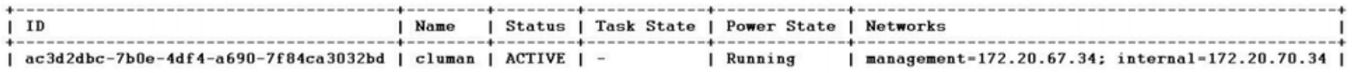
For example:

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

Step 5. List all the instances as shown, and verify that the new cluster manager instance is created and running:

```
nova list
```

Image 3. Example Output



```
+--------------------------------------+--------+--------+------------+-------------+------------------------------------------------+
| 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 |
```

Restore the latest patches on the system.

```
1.       Copy the patch files to cluster manager which were backed up in OSPD /home/stack/CPS_BACKUP
2.       Login to the Cluster Manager as a root user.
3.       Untar the patch by executing this command:  tar -xvzf [patch name].tar.gz
4.       Edit /etc/broadhop/repositories and add this entry:  file:///$path_to_the plugin/[component nam
5.       Run build_all.sh script to create updated QPS packages:  /var/qps/install/current/scripts/buil
6.       Shutdown all software components on the target VMs:  runonall.sh sudo monit stop all
7.       Make sure all software components are shutdown on target VMs:  statusall.sh
```

Note: The software components must all display Not Monitored as the current status.

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

Restore the Cronjobs.

1. Move the backed-up file from OSPD to the Cluman/Pcrfclient01.

2. Run the command to activate the cronjob from backup.

<#root>

```
#crontab Cron-backup
```

3. Check if the cronjobs have been activated by this command.

```
#crontab -l
```

Restore Individual VMs in the Cluster.

To redeploy the pcrfclient01 VM:

Step 1. Log in to the Cluster Manager VM as the root user.

Step 2. Remember the UUID of SVN repository using this command:

<#root>

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

The command can output the UUID of the repository.

For example: Repository UUID: ea50bbd2-5726-46b8-b807-10f4a7424f0e

Step 3. Import the backup Policy Builder configuration data on the Cluster Manager, as shown in this example:

<#root>

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

Note: Many deployments run a cron job that backs up configuration data regularly. See Subversion Repository Backup for more details.

Step 4. To generate the VM archive files on the Cluster Manager using the latest configurations, execute this command:

<#root>

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

Step 5. To deploy the pcrfclient01 VM, perform one of these:

In OpenStack, use the HEAT template or the Nova command to re-create the VM. For more information, see CPS Installation Guide for OpenStack.

Step 6. Re-establish SVN primary/secondary synchronization between the pcrfclient01 and pcrfclient02 with pcrfclient01 as the primary by executing these series of commands.

If SVN is already synchronized, do not issue these commands.

To check if SVN is in sync, run this command from pcrfclient02.

If a value is returned, then SVN is already in sync:

<#root>

```
/usr/bin/svn propget svn:sync-from-url --revprop -r0 http://pcrfclient01/repos
```

Execute this commands from pcrfclient01:

<#root>

```
/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
```

Step 7. If pcrfclient01 is also the arbiter VM, then execute these steps:

a) Create the mongodb start/stop scripts based on the system configuration. Not all deployments have all these databases configured.

Note: Refer to /etc/broadhop/mongoConfig.cfg to determine which databases need to be set up.

<#root>

```
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) Start the mongo process:

<#root>

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

c) Wait for the arbiter to start, then run diagnostics.sh --get_replica_status to check the health of the replica set.

To redeploy the pcrfclient02 VM:

Step 1. Log in to the Cluster Manager VM as the root user.

Step 2. To generate the VM archive files on the Cluster Manager using the latest configurations, execute this command:

<#root>

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

Step 3. To deploy the pcrfclient02 VM, perform one of these:
In OpenStack, use the HEAT template or the Nova command to re-create the VM. For more information, see CPS Installation Guide for OpenStack.

Step 4. Secure shell to the pcrfclient01:

<#root>

```
ssh pcrfclient01
```

Step 5. Run this script to recover the SVN repos from pcrfclient01:

<#root>

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

To redeploy a sessionmgr VM:

Step 1. Log in to the Cluster Manager VM as the root user.

Step 2. To deploy the sessionmgr VM and replace the failed or corrupt VM, perform one of these:

In OpenStack, use the HEAT template or the Nova command to re-create the VM. For more information, see CPS Installation Guide for OpenStack.

Step 3. Create the mongodb start/stop scripts based on the system configuration.

Not all deployments have all these databases configured. Refer to /etc/broadhop/mongoConfig.cfg to determine which databases need to be set up.

```
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
```

Step 4. Secure shell to the sessionmgr VM and start the mongo process:

<#root>

```
ssh sessionmgrXX

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

Step 5. Wait for the members to start and for the secondary members to synchronize, then run diagnostics.sh --get_replica_status to check the health of the database.

Step 6. To restore Session Manager database, use one of these example commands depending on whether the backup was performed with --mongo-all or --mongo option:

<#root>

-

```
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
```

To redeploy the Policy Director (Load Balancer) VM:

Step 1. Log in to the Cluster Manager VM as the root user.

Step 2. To import the backup Policy Builder configuration data on the Cluster Manager, execute this command:

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

Step 3. To generate the VM archive files on the Cluster Manager using the latest configurations, execute this command:

<#root>

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

Step 4. To deploy the lb01 VM, perform one of these:

In OpenStack, use the HEAT template or the Nova command to re-create the VM. For more information, see CPS Installation Guide for OpenStack.

To redeploy the Policy Server (QNS) VM:

Step 1. Log in to the Cluster Manager VM as the root user.

Step 2. Import the backup Policy Builder configuration data on the Cluster Manager, as shown in this example:

<#root>

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

Step 3. To generate the VM archive files on the Cluster Manager using the latest configurations, execute this command:

<#root>

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

Step 4. To deploy the qns VM, perform one of these:
In OpenStack, use the HEAT template or the Nova command to re-create the VM. For more information,

see CPS Installation Guide for OpenStack.

General Procedure for Database Restore.

Step 1. Execute this command to restore the database:

<#root>

```
config_br.py –a import --mongo-all /mnt/backup/backup_$date.tar.gz where $date is the timestamp when the
```

For example,

<#root>

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

Step 2. Log in to the database and verify whether it is running and is accessible:

1. Log into session manager:

<#root>

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

where $port is the port number of the database to check. For example, 27718 is the default Balance port.

2. Display the database by executing this command:

<#root>

```
show dbs
```

3. Switch the mongo shell to the database by executing this command:

<#root>

```
use $db
```

where $db is a database name displayed in the previous command.

The **use** command switches the mongo shell to that database.

For example,

<#root>

```
use balance_mgmt
```

4. To display the collections, execute this command:

<#root>

```
show collections
```

5. To display the number of records in the collection, execute this command:

<#root>

```
db.$collection.count()

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

The previous example can show the number of records in the collection account in the Balance database (balance_mgmt).

Subversion Repository Restore.

To restore the Policy Builder Configuration Data from a backup, execute this command:

<#root>

```
config_br.py –a import --svn /mnt/backup/backup_$date.tgz where, $date is the date when the cron created
```

Restore Grafana Dashboard.

You can restore Grafana dashboard using this command:

<#root>

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

Validating the Restore.

After restoring the data, verify the working system by executing this command:

<#root>

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