PCRF VM Recovery Procedure - OpenStack

Contents

[Introduction](#)
[Prerequisites](#)
[Requirements](#)
[Components Used](#)
[Backup Procedure](#)
[Step 1. Elastic Services Controller (ESC)](#)
[Step 2. Cisco Policy Suite Backup](#)
[Troubleshoot](#)

Introduction

The documents describes the procedure to recover the Virtual Cisco Policy and Charging Rules Function (vPCRF) instances deployed on an Ultra-M/OpenStack environment.

Prerequisites

Requirements

Cisco recommends that you have knowledge of these topics:

- OpenStack
- Cisco Policy Suite (CPS)
- Compute on which affected instances were deployed is now available
- Compute resources are available in the same availability zone as the affected instance

Components Used

This document is not restricted to specific software and hardware versions.

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, ensure that you understand the potential impact of any command.

Backup Procedure

Step 1. Elastic Services Controller (ESC)

Configurations in ESC-HA must be backed up monthly, before/after any scale-up or scale-down operation with the VNF and before/after configuration changes at ESC. It must be backed up in order to do a disaster recovery of ESC effectively.

ESC opdata as an XML

Follow this in order to export the ESC **opdata** as an XML:

1. Login to ESC with the use of admin credentials.

2. Export **opdata** to XML:

```
/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
```

3. Download this file to your local computer of **ftp/sftp** to a server outside the cloud.

4. All scripts and user-data files referenced in Deployment XMLs. Find all user-data files referenced in deployment XMLs of all VNFs from the opdata exported in the previous step.

```
grep "file://" /home/admin/ESC_config.xml | sort | uniq
```

Sample output:

```
<file>file://opt/cisco/esc/cisco-cps/config/gr/cfg/std/pcrf-cm_cloud.cfg</file>

<file>file://opt/cisco/esc/cisco-cps/config/gr/cfg/std/pcrf-oam_cloud.cfg</file>

<file>file://opt/cisco/esc/cisco-cps/config/gr/cfg/std/pcrf-pd_cloud.cfg</file>

<file>file://opt/cisco/esc/cisco-cps/config/gr/cfg/std/pcrf-qns_cloud.cfg</file>

<file>file://opt/cisco/esc/cisco-cps/config/gr/cfg/std/pcrf-sm_cloud.cfg</file>
```

5. Find all post-deployment script used to send CPS orchestration API.

6. Sample snippets of **post_deploy** script in **ESC opdata**

Sample output:

```
<policies>
      <policy>
         <name>PCRF_POST_DEPLOYMENT</name>
       <conditions>
          <condition>
            <name>LCS::POST_DEPLOY_ALIVE</name>
           </condition>
         </conditions>
            <actions>
               <action>
                 <name>FINISH_PCRF_INSTALLATION</name>
                 <type>SCRIPT</type>
                <properties>
 ----------
<property>
```

```
                  <name>script_filename</name>
                        <value>/opt/cisco/esc/cisco-cps/config/gr/tmo/cfg/../cps_init.py</value>
                        </property>
                        <property>
                          <name>script_timeout</name>
                          <value>3600</value>
                        </property>
                      </properties>
                    </action>
                  </actions>
                </policy>
              </policies>
```

Sample 2:

```
<policy>
  <name>PCRF_POST_DEPLOYMENT</name>
  <conditions>
    <condition>
      <name>LCS::POST_DEPLOY_ALIVE</name>
    </condition>
  </conditions>
  <actions>
     <action>
       <name>FINISH_PCRF_INSTALLATION</name>
       <type>SCRIPT</type>
       <properties>
         <property>
            <name>CLUMAN_MGMT_ADDRESS</name>
            <value>10.174.132.46</value>
          </property>
          <property>
            <name>CLUMAN_YAML_FILE</name>
            <value>/opt/cisco/esc/cisco-cps/config/vpcrf01/ cluman_orch_config.yaml</value>
          </property>
          <property>
            <name>script_filename</name>
            <value>/opt/cisco/esc/cisco-
cps/config/vpcrf01/vpcrf_cluman_post_deployment.py</value>
          </property>
          <property>
            <name>wait_max_timeout</name>
            <value>3600</value>
          </property>
        </properties>
      </action>
    </actions>
</policy>
```

If the deployment **ESC opdata** (extracted in the previous step) contains any of the highlighted files, back them up with the help of this command.

```
tar –zcf esc_files_backup.tgz /opt/cisco/esc/cisco-cps/config/
```

Download this file to your local computer of **ftp/sftp** to a server outside the cloud.

> **Note**: Although **opdata** is synced between ESC primary and standby, directories that contain user-data, XML and post-deploy scripts are not synced across both instances. It is suggested that customers push the contents of the directory that contain these files with the use of SCP or SFTP, these files should be constant across ESC-Primary and ESC-Standby

in order to recover a deployment when ESC VM which was primary at the time of deployment is not available.

Suggested Backups Schedule in ESC

These are suggested crontab entries for the root user to be added in ESC Primary and ESC Standby. You can, however, modify the hours/day/month as per their requirements and frequency of changes in the network.

```
30 01 * * * tar –zcf /home/admin/esc_files_backup_$(date +"\%Y-\%m-\%d").tgz
/opt/cisco/esc/cisco-cps/config/
00 02 * * * /opt/cisco/esc/confd/bin/netconf-console --host 127.0.0.1 --port 830 -u <admin-user>
-p <admin-password> --get-config > /home/admin/ESC_config_$(date +"\%Y-\%m-\%d").xml
```

Step 2. Cisco Policy Suite Backup

Cluster Manager acts as a puppet primary for one CPS cluster. Thus, it becomes necessary to take a snapshot of this instance. Also, Cisco provided backup and restore utility can be used to gather backups of mongoDB, policy configuration, grafana DB, users, networks and other pcrf configuration files. These files should be frequently backed up with the use of CPS backup utility and stored in a location outside Ultra-M cloud.

Snapshot of Cluster Manager VM

Cluster Manager Instance Snapshot must be backed up monthly, also before and after any configuration changes, patch updates and upgrades. Old snapshots can be deleted after successful activities to save disk space. This procedure describes the steps to backup the cluster manager instance as a snapshot:

1. This command to view the nova instances and note the name of the cluster manager VM instance:

```
nova list
```

2. Create a nova snapshot image as shown here:

```
nova image-create --poll <cluman_instance_name> <cluman_snapshot_name>
```

Sample output:

```
Server snapshotting... 100% complete

Finished
```

> **Note**: Ensure that you have enough disk space for the snapshot. Cluster Manager sometimes becomes unreachable at the time of snapshot creation and resumes itself after the snapshot has been created. If the instance remains unreachable even after snapshot process has finished, check the status of VM with the use of the **nova list** command. If it is in **SHUTOFF** state, you need to start the VM manually, with the use of **nova start** command.

3. Ensure the snapshot image is created with this command.

```
glance image-list
```

Sample output:

```
+--------------------------------------+-----------------------------------+
| ID                                   | Name                              |
+--------------------------------------+-----------------------------------+
| 1683d05f-2a9f-46d8-877d-10982ee819e1 | cluman_backup_image               |
| 30f2ece1-6438-4ef7-b4cf-44a0e7de183e | CPS_13.1.1.release.iso            |
| d38321a1-27c1-4c47-bc0f-24aedab5867a | CPS_13.1.1_Base                   |
+--------------------------------------+-----------------------------------+
```

4. When you perform any platform changes where Ceph might get impacted, it is always suggested to convert the snapshot of Cluster Manager to a QCOW file and save it to some remote location.

```
glance image-download --file /var/Pcrf/cluman_snapshot.raw <image-id of the snapshot>
```

5. Download this file to your local computer of **ftp/sftp** to a server outside the cloud.

Backup of CPS Configurations and Database

1. For backing up CPS configurations and database contents, **config_br.py** utility is inbuilt in the CPS platform. Details about the use of the **config_br.py** utility are present in CPS Backup and Restore Guide. This is a sample crontab in cluster manager to backup all configuration and databases at 0100 hrs every day.

```
00 01 * * * /var/platform/modules/config_br.py -a export -–all /mnt/backup/backup_$(date +\%Y-
\%m-\%d).tar
```

2. MongoDB can alternatively be backed up with the use of **mongodump**.

```
30 01 * * * mongodump --host sessionmgr01 –port 27721 --out /mnt/backup/mongo_admin_27721_$(date
+\%Y-\%m-\%d)/

30 01 * * * mongodump --host sessionmgr01 –port 27720 --out /mnt/backup/mongo_spr_27720_$(date
+\%Y-\%m-\%d)/

30 01 * * * mongodump --host sessionmgr01 –port 27718 --out /mnt/backup/mongo_bal_27718_$(date
+\%Y-\%m-\%d)/

30 01 * * * mongodump --host sessionmgr01 –port 27719 --out
/mnt/backup/mongo_report_27721_$(date +\%Y-\%m-\%d)/
```

3. Backup Orchestration YAML.

```
curl -i -X GET http://<Cluster Manager IP>:8458/api/system/config -H "Content-Type:
application/yaml" > /mnt/backup/CPS_orc_$(date +\%Y-\%m-\%d).yaml
```

If CPS orchestration API is used to configure the system, it is suggested to back up this configuration as well.

Note: All backups must be stored/transferred outside CPS VNF and preferably outside the cloud on which CPS is deployed.

Troubleshoot

CPS VNF Instance Recovery Procedures

Power on any Instance from SHUTOFF State.

If any instance is in SHUTOFF state due to a planned shutdown or some other reason, use this procedure to start the instance and enable it's monitoring in ESC.

1. Check the state of an instance via OpenStack.

```
source /home/stack/destackovsrc-Pcrf
nova list --fields name,host,status | grep cm
| c5e4ebd4-803d-45c1-bd96-fd6e459b7ed6 | cm_0_170d9c14-0221-4609-87e3-d752e636f57f| destackovs-
compute-2  | SHUTOFF|
```

2. Check if the Compute is available and ensure that the state is up.

```
source /home/stack/destackovsrc-Pcrf
nova list --fields name,host,status | grep cm
| c5e4ebd4-803d-45c1-bd96-fd6e459b7ed6 | cm_0_170d9c14-0221-4609-87e3-d752e636f57f| destackovs-
compute-2  | SHUTOFF|
```

3. Login to ESC Primary as an admin user and check the state of instance in **opdata**.

```
echo "show esc_datamodel opdata tenants tenant Pcrf deployments * state_machine | tab" |
/opt/cisco/esc/confd/bin/confd_cli -u admin –C | grep cm
cm_0_170d9c14-0221-4609-87e3-d752e636f57f VM_ERROR_STATE
```

4. Power on the instance from OpenStack.

```
source /home/stack/destackovsrc-Pcrf

nova start cm_0_170d9c14-0221-4609-87e3-d752e636f57f
```

5. Wait for five minutes for the instance to boot up and come to the **ACTIVE** state.

```
source /home/stack/destackovsrc-Pcrf
nova list --fields name,status | grep cm
| c5e4ebd4-803d-45c1-bd96-fd6e459b7ed6 |cm_0_170d9c14-0221-4609-87e3-d752e636f57f| ACTIVE |
```

6. Enable VM Monitor in ESC after the instance is in an **ACTIVE** state.

```
/opt/cisco/esc/esc-confd/esc-cli/esc_nc_cli vm-action ENABLE_MONITOR cm_0_170d9c14-0221-4609-
87e3-d752e636f57f
```

For further recovery of instance configurations, refer to instance **type specific** procedures provided here.

Recover any Instance from ERROR State

The following procedure can be used if the state of CPS instance in OpenStack is ERROR:

1. Check the state of an instance in OpenStack.

```
source /home/stack/destackovsrc-Pcrf
nova list --fields name,host,status | grep cm
| c5e4ebd4-803d-45c1-bd96-fd6e459b7ed6 | cm_0_170d9c14-0221-4609-87e3-d752e636f57f| destackovs-compute-2Â  | ERROR|
```

2. Check if the Compute is available and running fine.

```
source /home/stack/destackovsrc-Pcrf
nova list --fields name,host,status | grep cm
| c5e4ebd4-803d-45c1-bd96-fd6e459b7ed6 | cm_0_170d9c14-0221-4609-87e3-d752e636f57f| destackovs-compute-2  | ERROR|
```

3. Login to ESC Primary as an admin user and check the state of an instance in **opdata**.

```
echo "show esc_datamodel opdata tenants tenant Pcrf deployments * state_machine | tab" | /opt/cisco/esc/confd/bin/confd_cli -u admin –C | grep cm

cm_0_170d9c14-0221-4609-87e3-d752e636f57f VM_ERROR_STATE
```

4. Reset the state of the instance to force the instance back to an **ACTIVE** state instead of an error state. Once done, reboot your instance.

```
source /home/stack/destackovsrc-Pcrf

nova reset-state –active cm_0_170d9c14-0221-4609-87e3-d752e636f57f
nova reboot –-hard  cm_0_170d9c14-0221-4609-87e3-d752e636f57f
```

5. Wait for five minutes for the instance to boot up and come to an **ACTIVE** state.

```
source /home/stack/destackovsrc-Pcrf
nova list â€“fields name,status | grep cm
| c5e4ebd4-803d-45c1-bd96-fd6e459b7ed6 |cm_0_170d9c14-0221-4609-87e3-d752e636f57f| ACTIVE |
```

6. If Cluster Manager changes state to **ACTIVE** after reboot, enable VM Monitor in ESC.

```
/opt/cisco/esc/esc-confd/esc-cli/esc_nc_cli vm-action ENABLE_MONITOR cm_0_170d9c14-0221-4609-87e3-d752e636f57f
```

7. Post recovery to RUNNING/ACTIVE state, refer instance type specific procedure in order to recover config/data from backup.