

# Troubleshoot PCRf VM Recovery-Arbiter-Openstack

## Contents

[Introduction](#)

[Prerequisites](#)

[Requirements](#)

[Components Used](#)

[Troubleshoot](#)

[Power on Arbiter from SHUTOFF State](#)

[Recover any Instance from ERROR State](#)

[Recover Arbiter/arbiterVIP](#)

[Verify](#)

## Introduction

This document describes how to recover Cisco Virtual Policy and Charging Rules Function (vPCRf) instances deployed on Ultra-M/Openstack deployment.

Contributed by Nitesh Bansal, Cisco Advance Services.

## Prerequisites

### Requirements

Cisco recommends that you have knowledge on these topics:

- Openstack
- 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

The information in this document is based on CPS and applicable to all 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.

## Troubleshoot

### Power on Arbiter from SHUTOFF State

If any instance is in SHUTOFF state due to a planned shutdown or some other reason, please use the following procedure to start the instance and enable its monitoring in Elastic Service Controller (ESC).

Step 1. Check the state of instance via OpenStack.

```
source /home/stack/destackovsrc-Pcrf
nova list --fields name,host,status | grep arbiter
| c5e4ebd4-803d-45c1-bd96-fd6e459b7ed6 | r5-arbiter_arb_0_2eb86cbf-07e5-4e14-9002-8990588b8957 |
destackovs-compute-2 | SHUTOFF|
```

Step 2. Check if the compute is available and ensure state is up.

```
source /home/stack/destackovsrc
nova hypervisor-show destackovs-compute-2 | egrep 'status|state'
| state | up |
| status | enabled |
```

Step 3. Login to ESC Master as admin user and Check the state of instance in opdata.

```
/opt/cisco/esc/esc-confd/esc-cli/esc_nc_cli get esc_datamodel/opdata | grep arbiter
r5-arbiter_arb_0_2eb86cbf-07e5-4e14-9002-8990588b8957 VM_ERROR_STATE
```

Step 4. Power on the instance from openstack.

```
source /home/stack/destackovsrc-Pcrf
nova start r5-arbiter_arb_0_2eb86cbf-07e5-4e14-9002-8990588b8957
```

Step 5. Wait five minutes for the instance to boot up and come to active state.

```
source /home/stack/destackovsrc-Pcrf
nova list --fields name,status | grep cm
| c5e4ebd4-803d-45c1-bd96-fd6e459b7ed6 | r5-arbiter_arb_0_2eb86cbf-07e5-4e14-9002-8990588b8957 |
ACTIVE |
```

Step 6. Enable VM Monitor in ESC after instance is in active state.

```
/opt/cisco/esc/esc-confd/esc-cli/esc_nc_cli vm-action ENABLE_MONITOR r5-arbiter_arb_0_2eb86cbf-
07e5-4e14-9002-8990588b8957
```

For Further recovery of instance configurations, refer instance type specific procedures provided below.

## Recover any Instance from ERROR State

If state of CPS instance in openstack is in ERROR state, please use the following procedure to start the instance:

Step 1. Reset the state of 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 r5-arbiter_arb_0_2eb86cbf-07e5-4e14-9002-8990588b8957
nova reboot --hard r5-arbiter_arb_0_2eb86cbf-07e5-4e14-9002-8990588b8957
```

Step 2. Login to ESC Master as admin user and Check the state of instance in opdata.

```
/opt/cisco/esc/esc-confd/esc-cli/esc_nc_cli get esc_datamodel/opdata | grep arbiter
r5-arbiter_arb_0_2eb86cbf-07e5-4e14-9002-8990588b8957 VM_ERROR_STATE
```

Step 3. Check if the compute is available and runs fine.

```
source /home/stack/destackovsrc
nova hypervisor-show destackovs-compute-2 | egrep 'status|state'
| state                | up                |
| status               | enabled           |
```

Step 4. Check the state of instance in OpenStack.

```
source /home/stack/destackovsrc-Pcrf
nova list --fields name,host,status | grep arbiter
| c5e4ebd4-803d-45c1-bd96-fd6e459b7ed6 | r5-arbiter_arb_0_2eb86cbf-07e5-4e14-9002-8990588b8957 |
destackovs-compute-2 | ERROR|
```

Step 5. Wait five minutes for the instance to boot up and come to active state.

```
source /home/stack/destackovsrc-Pcrf
nova list --fields name,status | grep arbiter
| c5e4ebd4-803d-45c1-bd96-fd6e459b7ed6 | r5-arbiter_arb_0_2eb86cbf-07e5-4e14-9002-8990588b8957 |
ACTIVE |
```

Step 6. If Cluster Manager changes state to ACTIVE after reboot, enable VM Monitor in ESC after Cluster Manager instance is in active state.

```
/opt/cisco/esc/esc-confd/esc-cli/esc_nc_cli vm-action ENABLE_MONITOR r5-arbiter_arb_0_2eb86cbf-
07e5-4e14-9002-8990588b8957
```

Step 7. Post recovery to running/active state, refer instance type specific procedure to recover config/data from backup.

## Recover Arbiter/arbitervip

If an arbiter instance/pcrfclient is recently recovered and the arbiter is not in diagnostics.sh get\_replica\_status output then follow this procedure.

If deployment has dedicated arbiter VM use steps 1 to 3 , for arbitervip additionally run step 4, then run these steps:

1. On the cluster manager, run this command to create the mongodb start/stop scripts based on the system configuration:

```
cd /var/qps/bin/support/mongo
build_set.sh --all --create-scripts
```

2. On PCRFCLIENTXX or (and) arbiter run this command to list all processes that you need to start.

```
cd etc/init.d/  
ll | grep sessionmgr
```

3. On PCRFCLIENTXX or (and) arbiter for each file listed in the last output, run this command, replace xxxxx with port numbers, example for 27717 here:

```
/etc/init.d/sessionmgr-xxxxx start  
Example:  
/etc/init.d/sessionmgr-27717 start
```

4. If arbiter vip is used, check whether any of the pcs resources on pcrfclient01 require cleanup with the help of this commands:

```
pcs resource show | grep -v started
```

If any output is returned by the command in step 4 cleanup the pcs resource using following command, for multiple pcs resources that are not started repeat the command for each resource:

```
pcs resource cleanup <resource-name>
```

## Verify

Verify the health of replica status :

```
Run diagnostics.sh on pcrfclient01
```

If the arbiter runs as an arbiter and not an arbiter/pcrfclient then to verify the VM whether it is fully recovered or not, you can perform these steps:

1. On the primary arbiter, all mongo processes should run and it can be verified with this command on arbiter:

```
ps -aef | grep mongo
```

2. Verify all the processes under monit monitoring are in a good(Running/Monitored) state for arbiter.

```
monit summary
```