

Modular Components RMA-PCRF

Contents

[Introduction](#)

[Background Information](#)

[Abbreviations](#)

[Troubleshoot Component RMA - Compute/OSD-Compute Node](#)

[Step 1. Graceful Shutdown](#)

[Identify the VMs Hosted in the Compute/OSD-Compute Node](#)

[For the Cluster Manager VM Graceful Shutdown](#)

[For the Active PD/loadbalancer VM Graceful Shutdown](#)

[For the Standby PD/loadbalancer VM Graceful Shutdown](#)

[For the PS/QNS VM Graceful Shutdown](#)

[For the OAM/pcrfclient VM Graceful Shutdown](#)

[For the Arbiter VM](#)

[Step 2. ESC Database Backup.](#)

[Step 3. Migrate ESC to Standby Mode.](#)

[Step 4. Replace the Faulty Component from the Compute/OSD-Compute Node.](#)

[Step 5. Restore the VMs.](#)

[VM Recovery from ESC](#)

[Recovery of ESC VM](#)

[Handle ESC Recovery Failure](#)

[Troubleshoot Component RMA - Controller Node](#)

[Step 1. Controller - Prechecks](#)

[Step 2. Move Controller Cluster to Maintenance Mode.](#)

[Step 3. Replace the Faulty Component from the Controller Node.](#)

[Step 4. Power on the Server.](#)

Introduction

This document describes the steps required to replace faulty components mentioned here in a Cisco Unified Computing System (UCS) server in an Ultra-M setup that hosts Cisco Policy Suite (CPS) Virtual Network Functions (VNFs).

- Dual In-line Memory Module (DIMM) Replacement MOP
- FlexFlash Controller Failure
- Solid State Drive (SSD) Failure
- Trusted Platform Module (TPM) Failure
- Raid Cache Failure
- Raid Controller/ Hot Bus Adapter (HBA) Failure
- PCI Riser Failure
- PCIe adapter Intel X520 10G Failure
- Modular LAN-on Motherboard (MLOM) Failure

- Fan tray RMA
- CPU Failure

Contributed by Nitesh Bansal, Cisco Advance Services.

Background Information

Ultra-M is a pre-packaged and validated virtualized solution designed to simplify the deployment of VNFs. OpenStack is the Virtualized Infrastructure Manager (VIM) for Ultra-M and consists of these node types:

- Compute
- Object Storage Disk - Compute (OSD - Compute)
- Controller
- OpenStack Platform - Director (OSPD)
- Ultra M 5.1.x release is considered for defining the procedures in this document.
- This document is intended for the Cisco personnel familiar with Cisco Ultra-M platform and it details the steps required to be carried out at OpenStack and CPS VNF level at the time of the Component Replacement in the server.

Before you replace a faulty component, it is important to check the current status of Red Hat Open Stack platform environment. It is recommended that you check the current state in order to avoid complications when the replacement process is on.

In case of recovery, Cisco recommends to take the back-up of the OSPD database with the help of these steps:

```
[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
```

This process ensures that a node can be replaced without affecting the availability of the instances.

Note: If a server is controller node please proceed to the section , otherwise please continue with the next section.

Abbreviations

VNF	Virtual Network Function
PD	Policy Director (Load Balancer)
PS	Policy Server (pcrfclient)
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
SM	Session Manager
QNS	Quantum Name Server
UUID	Universally Unique IDentifier

Troubleshoot Component RMA - Compute/OSD-Compute Node

Step 1. Graceful Shutdown

Identify the VMs Hosted in the Compute/OSD-Compute Node

The Compute/OSD-Compute may host multiple types of VM. Identify all and proceed with the individual steps along with the particular baremetal node and for the particular VM names hosted on this compute:

```
[stack@director ~]$ nova list --field name,host | grep compute-10
| 49ac5f22-469e-4b84-badc-031083db0533 | SVS1-tmo_cm_0_e3ac7841-7f21-45c8-9f86-3524541d6634
|
pod1-compute-10.localdomain |
| 49ac5f22-469e-4b84-badc-031083db0533 | SVS1-tmo_sm-s3_0_05966301-bd95-4071-817a-
0af43757fc88 |
pod1-compute-10.localdomain |
```

For the Cluster Manager VM Graceful Shutdown

Step 1. Create a Snapshot and FTP the file to other location outside the server or if possible outside the rack itself.

```
openstack image create --poll <cluman_instance_name> <cluman_snapshot>
```

Step 2. Stop the VM from ESC.

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

Step 3. Verify if the VM is stopped.

```
[admin@esc ~]$ cd /opt/cisco/esc/esc-confd/esc-cli
[admin@esc ~]$ ./esc_nc_cli get esc_datamodel | egrep --color
"<state>|<vm_name>|<vm_id>|<deployment_name>"
<snip>
<state>SERVICE_ACTIVE_STATE</state>
SVS1-tmo_cm_0_e3ac7841-7f21-45c8-9f86-3524541d6634
VM_SHUTOFF_STATE</state>
```

For the Active PD/loadbalancer VM Graceful Shutdown

Step 1. Login to Active lb & stop the services as below

- Switch the lb from active to standby

```
service corosync restart
```

- stop services on standby lb

```
service monit stop
service qns stop
```

Step 2. From the ESC Master.

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

Step 3. Verify if the VM is stopped.

```
admin@esc ~]$ cd /opt/cisco/esc/esc-confd/esc-cli
[admin@esc ~]$ ./esc_nc_cli get esc_datamodel | egrep --color
"<state>|<vm_name>|<vm_id>|<deployment_name>"
<snip>
<state>SERVICE_ACTIVE_STATE</state>
          SVS1-tmo_cm_0_e3ac7841-7f21-45c8-9f86-3524541d6634
          VM_SHUTOFF_STATE</state>
```

For the Standby PD/loadbalancer VM Graceful Shutdown

Step 1. Login to standby lb & stop the services.

```
service monit stop
service qns stop
```

Step 2. From the ESC Master.

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

Step 3. Verify if the VM is stopped.

```
[admin@esc ~]$ cd /opt/cisco/esc/esc-confd/esc-cli
[admin@esc ~]$ ./esc_nc_cli get esc_datamodel | egrep --color
"<state>|<vm_name>|<vm_id>|<deployment_name>"
<snip>
<state>SERVICE_ACTIVE_STATE</state>
          SVS1-tmo_cm_0_e3ac7841-7f21-45c8-9f86-3524541d6634
          VM_SHUTOFF_STATE</state>
```

For the PS/QNS VM Graceful Shutdown

Step 1. Stop the service:

```
service monit stop
service qns stop
```

Step 2. From the ESC Master.

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

Step 3. Verify if the VM is stopped.

```
[dmin@esc ~]$ cd /opt/cisco/esc/esc-confd/esc-cli
[dmin@esc ~]$ ./esc_nc_cli get esc_datamodel | egrep --color
"<state>|<vm_name>|<vm_id>|<deployment_name>"
<snip>
<state>SERVICE_ACTIVE_STATE</state>
          SVS1-tmo_cm_0_e3ac7841-7f21-45c8-9f86-3524541d6634
          VM_SHUTDOWN_STATE</state>
```

For the SM VM Graceful Shutdown

Step 1. Stop the all the mongo services present in the sessionmgr.

```
[root@sessionmg01 ~]# cd /etc/init.d
[root@sessionmg01 init.d]# ls -l sessionmgr*

[root@sessionmg01 ~]# /etc/init.d/sessionmgr-27717 stop Stopping mongod: [ OK ]
[root@ sessionmg01 ~]# /etc/init.d/sessionmgr-27718 stop Stopping mongod: [ OK ]
[root@ sessionmg01 ~]# /etc/init.d/sessionmgr-27719 stop Stopping mongod: [ OK ]
```

Step 2. From the ESC Master.

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

Step 3. Verify if the VM is stopped.

```
[admin@esc ~]$ cd /opt/cisco/esc/esc-confd/esc-cli
[admin@esc ~]$ ./esc_nc_cli get esc_datamodel | egrep --color
"<state>|<vm_name>|<vm_id>|<deployment_name>"
<snip>
<state>SERVICE_ACTIVE_STATE</state>
          SVS1-tmo_cm_0_e3ac7841-7f21-45c8-9f86-3524541d6634
          VM_SHUTDOWN_STATE</state>
```

For the OAM/pcrfclient VM Graceful Shutdown

Step 1. Check if the policy SVN is in sync through these commands, If a value is returned, then SVN is already in sync and you don't need to sync it from PCRFCLIENT02. You should skip Recovery from the last backup can still be used if required.

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

Step 2. Re-establish SVN master/slave synchronization between the pcrfclient01 and pcrfclient02 with pcrfclient01 as the master by executing the series of commands on PCRFCLIENT01.

```
/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 3. Take a backup of the SVN in cluster manager.

```
config_br.py -a export --svn /mnt/backup/svn_backup_pcrfclient.tgz
```

Step 4. Shutdown the services in pcrfclient.

```
service monit stop
service qns stop
```

Step 5. From the ESC Master:

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

Step 6. Verify if the VM is stopped.

```
[admin@esc ~]$ cd /opt/cisco/esc/esc-confd/esc-cli
[admin@esc ~]$ ./esc_nc_cli get esc_datamodel | egrep --color
"<state>|<vm_name>|<vm_id>|<deployment_name>"
<snip>
<state>SERVICE_ACTIVE_STATE</state>
          SVS1-tmo_cm_0_e3ac7841-7f21-45c8-9f86-3524541d6634
          VM_SHUTOFF_STATE</state>
```

For the Arbiter VM

Step 1. Login to arbiter and shutdown the services.

```
[root@SVS10AM02 init.d]# ls -lrt sessionmgr*
-rwxr-xr-x 1 root root 4382 Jun 21 07:34 sessionmgr-27721
-rwxr-xr-x 1 root root 4406 Jun 21 07:34 sessionmgr-27718
-rwxr-xr-x 1 root root 4407 Jun 21 07:34 sessionmgr-27719
-rwxr-xr-x 1 root root 4429 Jun 21 07:34 sessionmgr-27717
-rwxr-xr-x 1 root root 4248 Jun 21 07:34 sessionmgr-27720
```

```
service monit stop
service qns stop
/etc/init.d/sessionmgr-[portno.] stop , where port no is the db port in the arbiter.
```

Step 2. From the ESC Master.

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

Step 3. Verify if the VM is stopped.

```
[admin@esc ~]$ cd /opt/cisco/esc/esc-confd/esc-cli
```

```
[admin@esc ~]$ ./esc_nc_cli get esc_datamodel | egrep --color
"<state>|<vm_name>|<vm_id>|<deployment_name>"
<snip>
<state>SERVICE_ACTIVE_STATE</state>
        SVS1-tmo_cm_0_e3ac7841-7f21-45c8-9f86-3524541d6634
        VM_SHUTDOWN_STATE</state>
```

For the Elastic Services Controller (ESC)

Step 1. 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. This must be backed up in order to do a disaster recovery of ESC effectively

1. Login to ESC using admin Credentials and 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
```

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

Step 2. Backup the PCRF cloud configuration All scripts and user-data files referenced in Deployment XMLs.

1. Find all user-data files referenced in deployment XMLs of all VNFs from the opdata exported in previous step. 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>
```

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

3. Sample snippets of post deploy script in esc opdata.

Sample 1:

```
<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>
          -----
        </properties>
      </action>
    </actions>
  </policy>
  <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, take the backup.

Sample Backup 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 cloud.

Note:- Although opdata is synced between ESC master and slave, directories containing user-data, xml and post deploy scripts are not synced across both instances. It is suggested that customers can push the contents of directory containing these files using scp or sftp, these files should be constant across ESC-Master and ESC-Standby in order to recover a deployment when ESC VM which was master during deployment is not available do to any unforeseen circumstances.

Step 2. ESC Database Backup.

Step 1. Collect the logs from the both ESC VMs and back it up.

```
$ collect_esc_log.sh
$ scp /tmp/<log_package_file> <username>@<backup_vm_ip>:<filepath>
```

Step 2. Backup the database from Master ECS node.

Step 3. Switch to the root user and check the status of the primary ESC and validate the output value is **Master**.

```
$ sudo bash
$ escadm status
```

Set ESC to maintenance mode & verify

```
$ sudo escadm op_mode set --mode=maintenance
$ escadm op_mode show
```

Step 4. Use a variable to set the file name and include date info and Call the backup tool and provide the filename variable from the previous step.

```
fname=esc_db_backup_$(date -u +"%y-%m-%d-%H-%M-%S")
$ sudo /opt/cisco/esc/esc-scripts/esc_dbtool.py backup -- file /tmp/atlpod-esc-master-$(fname).tar
```

Step 5. Check the backup file in your backup storage and ensure the file is there.

Step 6. Put Master ESC back into normal operation mode.

```
$ sudo escadm op_mode set --mode=operation
```

If dbtool backup utility fails, apply the following workaround once in ESC node. Then repeat step 6.

```
$ sudo sed -i "s,'pg_dump','usr/pgsql-9.4/bin/pg_dump,'"
/opt/cisco/esc/esc-scripts/esc_dbtool.py
```

Step 3. Migrate ESC to Standby Mode.

Step 1. Log in to the ESC hosted in the node and check if it is in the master state. If yes, switch the ESC to standby mode.

```
[admin@VNF2-esc-esc-0 esc-cli]$ escadm status
0 ESC status=0 ESC Master Healthy

[admin@VNF2-esc-esc-0 ~]$ sudo service keepalived stop Stopping
keepalived:
[ OK ]
[admin@VNF2-esc-esc-0 ~]$ escadm status
```

1 ESC status=0 In SWITCHING_TO_STOP state. Please check status after a while.

```
[admin@VNF2-esc-esc-0 ~]$ sudo reboot
Broadcast message from admin@vnf1-esc-esc-0.novalocal
(/dev/pts/0) at 13:32 ...
The system is going down for reboot NOW!
```

Step 2. Once VM is ESC Standby, shutdown the VM by the command: **shutdown -r now**

Note: If the faulty component is to be replaced on OSD-Compute node, put the CEPH into Maintenance on the server before proceeding with the component replacement.

```
[admin@osd-compute-0 ~]$ sudo ceph osd set norebalance
set norebalance
[admin@osd-compute-0 ~]$ sudo ceph osd set noout
set noout
[admin@osd-compute-0 ~]$ sudo ceph status
  cluster eb2bb192-b1c9-11e6-9205-525400330666
    health HEALTH_WARN
      noout,norebalance,sortbitwise,require_jewel_osds flag(s) set
    monmap e1: 3 mons at {tb3-ultram-pod1-controller-0=11.118.0.40:6789/0,tb3-ultram-pod1-
controller-1=11.118.0.41:6789/0,tb3-ultram-pod1-controller-2=11.118.0.42:6789/0}
      election epoch 58, quorum 0,1,2 tb3-ultram-pod1-controller-0,tb3-ultram-pod1-
controller-1,tb3-ultram-pod1-controller-2
    osdmap e194: 12 osds: 12 up, 12 in
      flags noout,norebalance,sortbitwise,require_jewel_osds
    pgmap v584865: 704 pgs, 6 pools, 531 GB data, 344 kobjects
      1585 GB used, 11808 GB / 13393 GB avail
        704 active+clean
    client io 463 kB/s rd, 14903 kB/s wr, 263 op/s rd, 542 op/s wr
```

Step 4. Replace the Faulty Component from the Compute/OSD-Compute Node.

Power off the specified server. The steps in order to replace a faulty component on UCS C240 M4 server can be referred from:

[Replacing the Server Components](#)

Refer to the Persistent Logging in the below procedure and execute as needed

Step 5. Restore the VMs.

VM Recovery from ESC

1. The VM would be in error state in the nova list.

```
[stack@director ~]$ nova list |grep VNF2-DEPLOYM_s9_0_8bc6cc60-15d6-4ead-8b6a-10e75d0e134d
| 49ac5f22-469e-4b84-badc-031083db0533 | VNF2-DEPLOYM_s9_0_8bc6cc60-15d6-4ead-8b6a-
10e75d0e134d | ERROR | - | NOSTATE |
```

2. Recover the VMs from the ESC.

```
[admin@VNF2-esc-esc-0 ~]$ sudo /opt/cisco/esc/esc-confd/esc-cli/esc_nc_cli recovery-vm-
action DO VNF2-DEPLOYM_s9_0_8bc6cc60-15d6-4ead-8b6a-10e75d0e134d
[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>
```

3. Monitor the yangesc.log

```
admin@VNF2-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 [VNF2-
DEPLOYM_s9_0_8bc6cc60-15d6-4ead-8b6a-10e75d0e134d].
```

4. Verify the all the Services in the VMs being started.

Recovery of ESC VM

1. Login to ESC via console and verify the status.
2. Start the processes if not already started

```
[admin@esc ~]$ sudo service keepalived start
```

```
[admin@esc ~]$ escadm status 0 ESC status=0 ESC Slave Healthy
```

Handle ESC Recovery Failure

In cases where ESC fails to start the VM due to an unexpected state, Cisco recommends performing an ESC switchover by rebooting the Master ESC. The ESC switchover would take about a minute. Run the script "health.sh" on the new Master ESC to check if the status is up. Master ESC to start the VM and fix the VM state. This recovery task would take up to 5 minutes 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, the user would need to go and do the manual recovery of the impacted VM(s).

In case the ESC VM is not recovered, follow the procedure to deploy a new ESC VM. Contact Cisco Support for the procedure.

Troubleshoot Component RMA - Controller Node

Step 1. Controller - Prechecks

From OSPD, login to the controller and verify pcs is in a good state – all three controllers Online and galera showing all three controllers as master.

Note: A healthy cluster requires 2 active controllers so verify that the remaining two controllers are Online and active.

```
heat-admin@pod1-controller-0 ~]$ sudo pcs status
Cluster name: tripleo_cluster
Stack: corosync
Current DC: pod1-controller-2 (version 1.1.15-11.e17_3.4-e174ec8) - partition with quorum
Last updated: Mon Dec  4 00:46:10 2017                Last change: Wed Nov 29 01:20:52
2017 by hacluster via crmd on pod1-controller-0
3 nodes and 22 resources configured
Online: [ pod1-controller-0 pod1-controller-1 pod1-controller-2 ]
Full list of resources:
ip-11.118.0.42 (ocf::heartbeat:IPaddr2):           Started pod1-controller-1
ip-11.119.0.47 (ocf::heartbeat:IPaddr2):           Started pod1-controller-2
ip-11.120.0.49 (ocf::heartbeat:IPaddr2):           Started pod1-controller-1
ip-192.200.0.102 (ocf::heartbeat:IPaddr2):         Started pod1-controller-2
Clone Set: haproxy-clone [haproxy]
  Started: [ pod1-controller-0 pod1-controller-1 pod1-controller-2 ]
Master/Slave Set: galera-master [galera]
  Masters: [ pod1-controller-0 pod1-controller-1 pod1-controller-2 ]
ip-11.120.0.47 (ocf::heartbeat:IPaddr2):           Started pod1-controller-2
Clone Set: rabbitmq-clone [rabbitmq]
  Started: [ pod1-controller-0 pod1-controller-1 pod1-controller-2 ]
Master/Slave Set: redis-master [redis]
  Masters: [ pod1-controller-2 ]
  Slaves: [ pod1-controller-0 pod1-controller-1 ]
ip-10.84.123.35 (ocf::heartbeat:IPaddr2):           Started pod1-controller-1
openstack-cinder-volume (systemd:openstack-cinder-volume): Started pod1-
controller-2
my-ipmilan-for-pod1-controller-0 (stonith:fence_ipmilan): Started pod1-controller-0
my-ipmilan-for-pod1-controller-1 (stonith:fence_ipmilan): Started pod1-controller-0
my-ipmilan-for-pod1-controller-2 (stonith:fence_ipmilan): Started pod1-controller-0
Daemon Status:
  corosync: active/enabled
  pacemaker: active/enabled
  pcsd: active/enabled
```

Step 2. Move Controller Cluster to Maintenance Mode.

1. Put the pcs cluster on the controller being updated in standby.

```
[heat-admin@pod1-controller-0 ~]$ sudo pcs cluster standby
```

2. Check pcs status again and make sure the pcs cluster stopped on this node.

```

[heat-admin@pod1-controller-0 ~]$ sudo pcs status
Cluster name: tripleo_cluster
Stack: corosync
Current DC: pod1-controller-2 (version 1.1.15-11.el7_3.4-e174ec8) - partition with quorum
Last updated: Mon Dec  4 00:48:24 2017                Last change: Mon Dec  4
00:48:18 2017 by root via crm_attribute on pod1-controller-0
3 nodes and 22 resources configured
Node pod1-controller-0: standby
Online: [ pod1-controller-1 pod1-controller-2 ]
Full list of resources:
 ip-11.118.0.42 (ocf::heartbeat:IPaddr2):           Started pod1-controller-1
 ip-11.119.0.47 (ocf::heartbeat:IPaddr2):           Started pod1-controller-2
 ip-11.120.0.49 (ocf::heartbeat:IPaddr2):           Started pod1-controller-1
 ip-192.200.0.102 (ocf::heartbeat:IPaddr2):         Started pod1-controller-2
Clone Set: haproxy-clone [haproxy]
  Started: [ pod1-controller-1 pod1-controller-2 ]
  Stopped: [ pod1-controller-0 ]
Master/Slave Set: galera-master [galera]
  Masters: [ pod1-controller-1 pod1-controller-2 ]
  Slaves: [ pod1-controller-0 ]
 ip-11.120.0.47 (ocf::heartbeat:IPaddr2):           Started pod1-controller-2
Clone Set: rabbitmq-clone [rabbitmq]
  Started: [ pod1-controller-0 pod1-controller-1 pod1-controller-2 ]
Master/Slave Set: redis-master [redis]
  Masters: [ pod1-controller-2 ]
  Slaves: [ pod1-controller-1 ]
  Stopped: [ pod1-controller-0 ]
 ip-10.84.123.35 (ocf::heartbeat:IPaddr2):           Started pod1-controller-1
openstack-cinder-volume (systemd:openstack-cinder-volume): Started
pod1-controller-2
 my-ipmilan-for-pod1-controller-0 (stonith:fence_ipmilan): Started pod1-controller-
1
 my-ipmilan-for-pod1-controller-1 (stonith:fence_ipmilan): Started pod1-controller-
1
 my-ipmilan-for-pod1-controller-2 (stonith:fence_ipmilan): Started pod1-controller-
2
Daemon Status:
 corosync: active/enabled
 pacemaker: active/enabled
 pcsd: active/enabled

```

3. Also pcs status on the other 2 controllers should show the node as standby.

Step 3. Replace the Faulty Component from the Controller Node.

Power off the specified server. The steps in order to replace a faulty component on UCS C240 M4 server can be referred from:

[Replacing the Server Components](#)

Step 4. Power on the Server.

1. Power on the server and verify server comes up.

```
[stack@tb5-ospd ~]$ source stackrc
[stack@tb5-ospd ~]$ nova list |grep pod1-controller-0
| 1ca946b8-52e5-4add-b94c-4d4b8a15a975 | pod1-controller-0 | ACTIVE | - |
Running | ctlplane=192.200.0.112 |
```

2. Login to the impacted controller, remove standby mode by setting **unstandby**. Verify controller comes Online with cluster and galera shows all three controllers as Master. This may take a few minutes.

```
[heat-admin@pod1-controller-0 ~]$ sudo pcs cluster unstandby
[heat-admin@pod1-controller-0 ~]$ sudo pcs status
Cluster name: tripleo_cluster
Stack: corosync
Current DC: pod1-controller-2 (version 1.1.15-11.e17_3.4-e174ec8) - partition with quorum
Last updated: Mon Dec 4 01:08:10 2017 Last change: Mon Dec 4
01:04:21 2017 by root via crm_attribute on pod1-controller-0
3 nodes and 22 resources configured
Online: [ pod1-controller-0 pod1-controller-1 pod1-controller-2 ]
Full list of resources:
 ip-11.118.0.42 (ocf::heartbeat:IPAddr2): Started pod1-controller-1
 ip-11.119.0.47 (ocf::heartbeat:IPAddr2): Started pod1-controller-2
 ip-11.120.0.49 (ocf::heartbeat:IPAddr2): Started pod1-controller-1
 ip-192.200.0.102 (ocf::heartbeat:IPAddr2): Started pod1-controller-2
Clone Set: haproxy-clone [haproxy]
 Started: [ pod1-controller-0 pod1-controller-1 pod1-controller-2 ]
Master/Slave Set: galera-master [galera]
 Masters: [ pod1-controller-0 pod1-controller-1 pod1-controller-2 ]
 ip-11.120.0.47 (ocf::heartbeat:IPAddr2): Started pod1-controller-2
Clone Set: rabbitmq-clone [rabbitmq]
 Started: [ pod1-controller-0 pod1-controller-1 pod1-controller-2 ]
Master/Slave Set: redis-master [redis]
 Masters: [ pod1-controller-2 ]
 Slaves: [ pod1-controller-0 pod1-controller-1 ]
 ip-10.84.123.35 (ocf::heartbeat:IPAddr2): Started pod1-controller-1
 openstack-cinder-volume (systemd:openstack-cinder-volume): Started
 pod1-controller-2
 my-ipmilan-for-pod1-controller-0 (stonith:fence_ipmilan): Started pod1-controller-1
 my-ipmilan-for-pod1-controller-1 (stonith:fence_ipmilan): Started pod1-controller-1
 my-ipmilan-for-pod1-controller-2 (stonith:fence_ipmilan): Started pod1-controller-2

Daemon Status:
 corosync: active/enabled
 pacemaker: active/enabled
 pcsd: active/enabled
```

3. You may check some of the monitor services such as ceph that they are in a healthy state.

```
[heat-admin@pod1-controller-0 ~]$ sudo ceph -s
cluster eb2bb192-b1c9-11e6-9205-525400330666
health HEALTH_OK
monmap e1: 3 mons at {pod1-controller-0=11.118.0.10:6789/0,pod1-controller-1=11.118.0.11:6789/0,pod1-controller-2=11.118.0.12:6789/0}
election epoch 70, quorum 0,1,2 pod1-controller-0,pod1-controller-1,pod1-controller-2
osdmap e218: 12 osds: 12 up, 12 in
flags sortbitwise,require_jewel_osds
pgmap v2080888: 704 pgs, 6 pools, 714 GB data, 237 kobjects
```

2142 GB used, 11251 GB / 13393 GB avail
704 active+clean
client io 11797 kB/s wr, 0 op/s rd, 57 op/s wr