

# Replace Motherboard in Ultra-M UCS 240M4 Server - vEPC

## Contents

[Introduction](#)

[Background Information](#)

[Abbreviations](#)

[Workflow of the MoP](#)

[Motherboard Replacement in Ultra-M Setup](#)

[Motherboard Replacement in Compute Node](#)

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

[Graceful Power Off](#)

[Case 1. Compute Node Hosts only SF VM](#)

[Case 2. Compute Node Hosts CF/ESC/EM/UAS](#)

[Replace Motherboard](#)

[Restore the VMs](#)

[Case 1. Compute Node Hosts only SF VM](#)

[Case 2. Compute Node Hosts UAS, ESC, EM and CF](#)

[Motherboard Replacement in OSD Compute Node](#)

[Put Ceph in Maintenance Mode](#)

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

[Graceful Power Off](#)

[Case 1. OSD-Compute Node Hosts CF/ESC/EM/UAS](#)

[Case 2. OSD-Compute Node Hosts Auto-Deploy/Auto-It/EM/UAS](#)

[Backup the CDB of Auto-Deploy](#)

[Backup System.cfg from Auto-IT](#)

[Replace Motherboard](#)

[Move Ceph Out of the Maintenance Mode](#)

[Restore the VMs](#)

[Case 1. OSD-Compute Node Hosts CF, ESC, EM and UAS](#)

[Case 2. OSD-Compute Node Hosts Auto-It, Auto-Deploy, EM and UAS](#)

[Recover Auto-IT VM](#)

[Replace Motherboard in Controller Node](#)

[Verify Controller Status and Put Cluster in Maintenance Mode](#)

[Replace Motherboard](#)

[Restore Cluster Status](#)

## Introduction

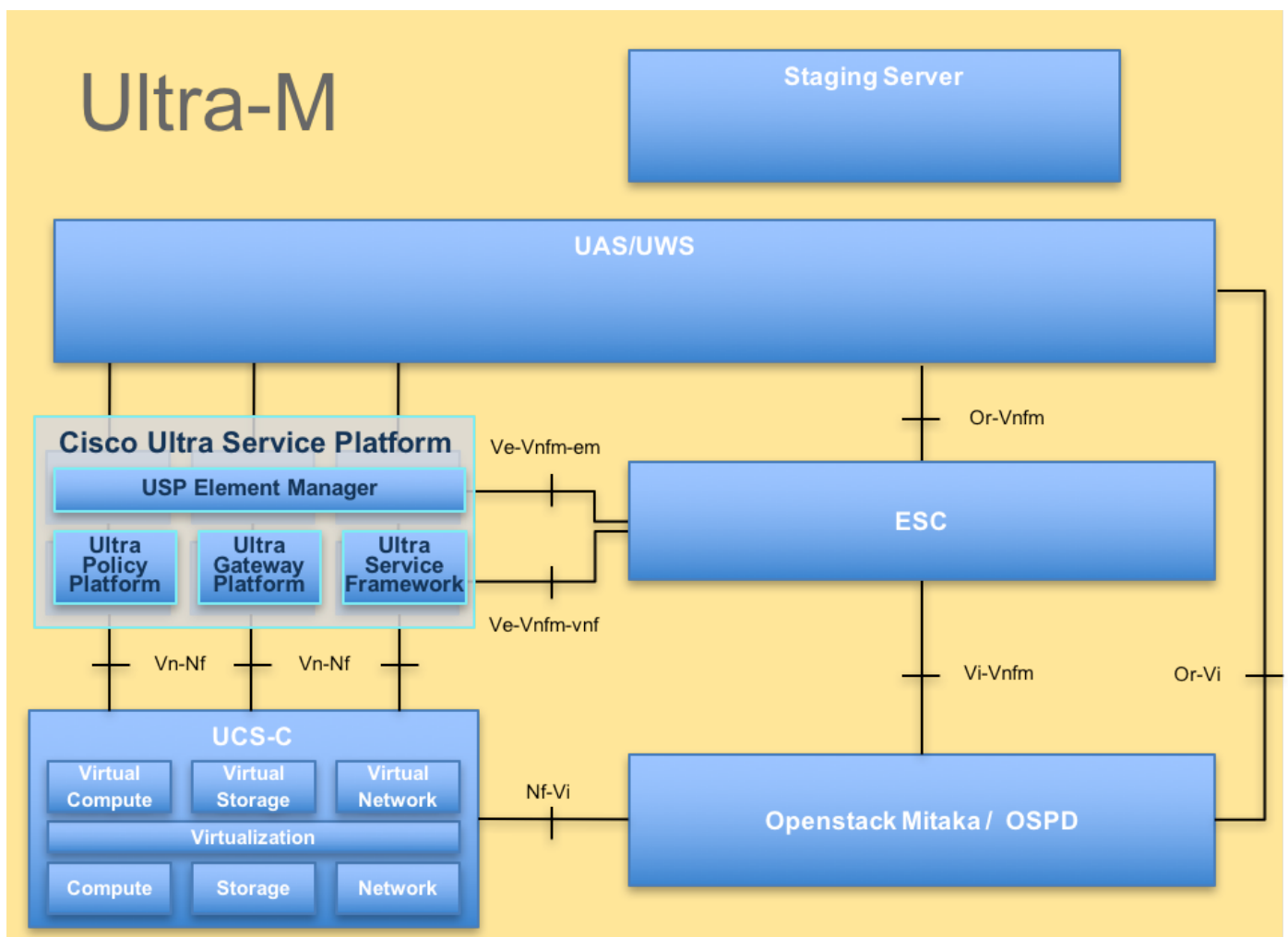
This document describes the steps required in order to replace the faulty Motherboard of a server in an Ultra-M setup that hosts StarOS Virtual Network Functions (VNFs).

# Background Information

Ultra-M is a pre-packaged and validated virtualized mobile packet core solution that is designed in order 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)

The high-level architecture of Ultra-M and the components involved are depicted in this image:



UltraM Architecture

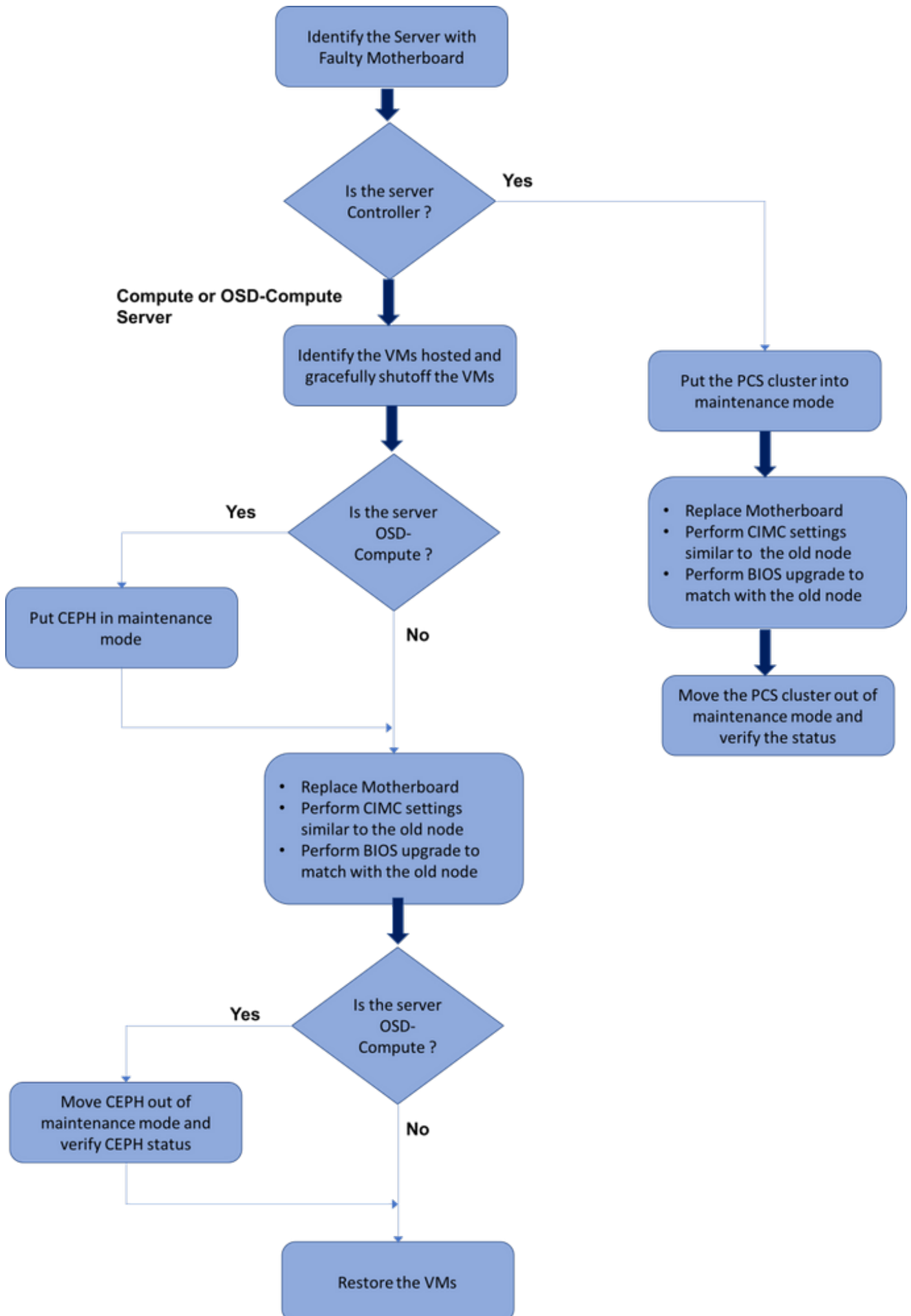
This document is intended for Cisco personnel who are familiar with Cisco Ultra-M platform and it details the steps that are required to be carried out at OpenStack and StarOS VNF level at the time of the Motherboard Replacement in a server.

**Note:** Ultra M 5.1.x release is considered in order to define the procedures in this document.

## Abbreviations

VNF	Virtual Network Function
CF	Control Function
SF	Service 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
EM	Element Manager
UAS	Ultra Automation Services
UUID	Universally Unique Identifier

## **Workflow of the MoP**



## Motherboard Replacement in Ultra-M Setup

In an Ultra-M setup, there can be scenarios where a motherboard replacement is required in these server types: Compute, OSD-Compute and Controller.

**Note:** The boot disks with the OpenStack installation are replaced after the replacement of the motherboard takes place. Hence, there is no requirement to add the node back to overcloud. Once the server is powered On after the replacement activity, it would enrol itself back to the overcloud stack.

## Motherboard Replacement in Compute Node

Before the activity, the VMs hosted in the Compute node are gracefully shutdown. Once the Motherboard has been replaced, the VMs are restored back.

### Identify the VMs Hosted in the Compute Node

Identify the VMs that are hosted on the Compute Server. There can be two possibilities:

The Compute Server contains only SF VM:

```
[stack@director ~]$ nova list --field name,host | grep compute-10
| 49ac5f22-469e-4b84-badc-031083db0533 | VNF2-DEPLOYM_s9_0_8bc6cc60-15d6-4ead-8b6a-
10e75d0e134d |
pod1-compute-10.localdomain |
```

The Compute Server contains CF/ESC/EM/UAS combination of VMs:

```
[stack@director ~]$ nova list --field name,host | grep compute-8
| 507d67c2-1d00-4321-b9d1-da879af524f8 | VNF2-DEPLOYM_XXXX_0_c8d98f0f-d874-45d0-af75-
88a2d6fa82ea | pod1-compute-8.localdomain |
| f9c0763a-4a4f-4bbd-af51-bc7545774be2 | VNF2-DEPLOYM_c1_0_df4be88d-b4bf-4456-945a-
3812653ee229 | pod1-compute-8.localdomain |
| 75528898-ef4b-4d68-b05d-882014708694 | VNF2-ESC-ESC-
0 | pod1-compute-8.localdomain |
| f5bd7b9c-476a-4679-83e5-303f0aae9309 | VNF2-UAS-uas-
0 | pod1-compute-8.localdomain |
```

**Note:** In the output shown here, the first column corresponds to the UUID, the second column is the VM name and the third column is the hostname where the VM is present. The parameters from this output will be used in the subsequent sections.

## Graceful Power Off

### Case 1. Compute Node Hosts only SF VM

Log in to the StarOS VNF and identify the card that corresponds to the SF VM. Use the UUID of the SF VM identified from the section **Identify the VMs hosted in the Compute Node**, and identify the card that corresponds to the UUID:

```
[local]VNF2# show card hardware
Tuesday night 08 16:49:42 UTC 2018
<snip>
Card 8:
  Card Type           : 4-Port Service Function Virtual Card
  CPU Packages        : 26 [#0, #1, #2, #3, #4, #5, #6, #7, #8, #9, #10, #11, #12, #13, #14,
#15, #16, #17, #18, #19, #20, #21, #22, #23, #24, #25]
  CPU Nodes           : 2
  CPU Cores/Threads   : 26
  Memory              : 98304M (qvpc-di-large)
  UUID/Serial Number  : 49AC5F22-469E-4B84-BADC-031083DB0533
<snip>
```

Check the status of the card:

```
[local]VNF2# show card table
Tuesday night 08 16:52:53 UTC 2018
```

Slot	Card Type	Oper State	SPOF	Attach
1: CFC	Control Function Virtual Card	Active	No	
2: CFC	Control Function Virtual Card	Standby	-	
3: FC	4-Port Service Function Virtual Card	Active	No	
4: FC	4-Port Service Function Virtual Card	Active	No	
5: FC	4-Port Service Function Virtual Card	Active	No	
6: FC	4-Port Service Function Virtual Card	Active	No	
7: FC	4-Port Service Function Virtual Card	Active	No	
<b>8: FC</b>	<b>4-Port Service Function Virtual Card</b>	<b>Active</b>	<b>No</b>	
9: FC	4-Port Service Function Virtual Card	Active	No	
10: FC	4-Port Service Function Virtual Card	Standby	-	

If the card is in the active state, move the card to standby state:

```
[local]VNF2# card migrate from 8 to 10
```

Log in to the ESC node that corresponds to the VNF and check the status of the SF VM:

```
[admin@VNF2-esc-esc-0 ~]$ cd /opt/cisco/esc/esc-confd/esc-cli
[admin@VNF2-esc-esc-0 esc-cli]$ ./esc_nc_cli get esc_datamodel | egrep --color
"<state>|<vm_name>|<vm_id>|<deployment_name>"
<snip>
<state>SERVICE_ACTIVE_STATE</state>
  <vm_name>VNF2-DEPLOYM_c1_0_df4be88d-b4bf-4456-945a-3812653ee229</vm_name>
  <state>VM_ALIVE_STATE</state>
  <vm_name> VNF2-DEPLOYM_s9_0_8bc6cc60-15d6-4ead-8b6a-10e75d0e134d</vm_name>
  <state>VM_ALIVE_STATE</state>
<snip>
```

Stop the SF VM with the use of its VM Name. (VM Name noted from section **Identify the VMs hosted in the Compute Node**)

```
[admin@VNF2-esc-esc-0 esc-cli]$ ./esc_nc_cli vm-action STOP VNF2-DEPLOYM_s9_0_8bc6cc60-15d6-4ead-8b6a-10e75d0e134d
```

Once it is stopped, the VM must enter the **SHUTOFF** state:

```
[admin@VNF2-esc-esc-0 ~]$ cd /opt/cisco/esc/esc-confd/esc-cli
[admin@VNF2-esc-esc-0 esc-cli]$ ./esc_nc_cli get esc_datamodel | egrep --color
"<state>|<vm_name>|<vm_id>|<deployment_name>"
<snip>
<state>SERVICE_ACTIVE_STATE</state>
    <vm_name>VNF2-DEPLOYM_c1_0_df4be88d-b4bf-4456-945a-3812653ee229</vm_name>
    <state>VM_ALIVE_STATE</state>
    <vm_name>VNF2-DEPLOYM_c3_0_3e0db133-c13b-4e3d-ac14-
    <state>VM_ALIVE_STATE</state>
    <vm_name>VNF2-DEPLOYM_s9_0_8bc6cc60-15d6-4ead-8b6a-10e75d0e134d</vm_name>
    <state>VM_SHUTOFF_STATE</state>
<snip>
```

## Case 2. Compute Node Hosts CF/ESC/EM/UAS

Log in to the StarOS VNF and identify the card that corresponds to the CF VM. Use the UUID of the CF VM identified from the section **Identify the VMs hosted in the Compute Node**, and find the card that corresponds to the UUID:

```
[local]VNF2# show card hardware
Tuesday night 08 16:49:42 UTC 2018
<snip>
Card 2:
  Card Type           : Control Function Virtual Card
  CPU Packages        : 8 [#0, #1, #2, #3, #4, #5, #6, #7]
  CPU Nodes           : 1
  CPU Cores/Threads   : 8
  Memory              : 16384M (qvpc-di-large)
  UUID/Serial Number  : F9C0763A-4A4F-4BBD-AF51-BC7545774BE2
<snip>
```

Check the status of the card:

```
[local]VNF2# show card table
Tuesday night 08 16:52:53 UTC 2018
```

Slot	Card Type	Oper State	SPOF	Attach
1: CFC	Control Function Virtual Card	Standby	-	
2: CFC	Control Function Virtual Card	Active	No	
3: FC	4-Port Service Function Virtual Card	Active	No	
4: FC	4-Port Service Function Virtual Card	Active	No	
5: FC	4-Port Service Function Virtual Card	Active	No	
6: FC	4-Port Service Function Virtual Card	Active	No	
7: FC	4-Port Service Function Virtual Card	Active	No	
8: FC	4-Port Service Function Virtual Card	Active	No	
9: FC	4-Port Service Function Virtual Card	Active	No	
10: FC	4-Port Service Function Virtual Card	Standby	-	

If the card is in the active state, move the card to the standby state:

```
[local]VNF2# card migrate from 2 to 1
```

Log in to the ESC node that corresponds to the VNF and check the status of the VMs:

```
[admin@VNF2-esc-esc-0 ~]$ cd /opt/cisco/esc/esc-confd/esc-cli
[admin@VNF2-esc-esc-0 esc-cli]$ ./esc_nc_cli get esc_datamodel | egrep --color
"<state>|<vm_name>|<vm_id>|<deployment_name>"
```

```

<snip>
<state>SERVICE_ACTIVE_STATE</state>
    <vm_name>VNF2-DEPLOYM_c1_0_df4be88d-b4bf-4456-945a-3812653ee229</vm_name>
    <state>VM_ALIVE_STATE</state>
    <vm_name>VNF2-DEPLOYM_c3_0_3e0db133-c13b-4e3d-ac14-
    <state>VM_ALIVE_STATE</state>
<deployment_name>VNF2-DEPLOYMENT-em</deployment_name>
    <vm_id>507d67c2-1d00-4321-b9d1-da879af524f8</vm_id>
    <vm_id>dc168a6a-4aeb-4e81-abd9-91d7568b5f7c</vm_id>
    <vm_id>9ffec58b-4b9d-4072-b944-5413bf7fcf07</vm_id>
    <state>SERVICE_ACTIVE_STATE</state>
    <vm_name>VNF2-DEPLOYM_XXXX_0_c8d98f0f-d874-45d0-af75-88a2d6fa82ea</vm_name>
    <state>VM_ALIVE_STATE</state>
<snip>

```

Stop the CF and EM VM one-by-one with the use of its VM Name. (VM Name noted from section **Identify the VMs hosted in the Compute Node**)

```

[admin@VNF2-esc-esc-0 esc-cli]$ ./esc_nc_cli vm-action STOP VNF2-DEPLOYM_c1_0_df4be88d-b4bf-4456-945a-3812653ee229

```

```

[admin@VNF2-esc-esc-0 esc-cli]$ ./esc_nc_cli vm-action STOP VNF2-DEPLOYM_XXXX_0_c8d98f0f-d874-45d0-af75-88a2d6fa82ea

```

After it stops, the VMs must enter the **SHUTOFF** state:

```

[admin@VNF2-esc-esc-0 ~]$ cd /opt/cisco/esc/esc-confd/esc-cli
[admin@VNF2-esc-esc-0 esc-cli]$ ./esc_nc_cli get esc_datamodel | egrep --color
"<state>|<vm_name>|<vm_id>|<deployment_name>"
<snip>
<state>SERVICE_ACTIVE_STATE</state>
    <vm_name>VNF2-DEPLOYM_c1_0_df4be88d-b4bf-4456-945a-3812653ee229</vm_name>
    <state>VM_SHUTOFF_STATE</state>
    <vm_name>VNF2-DEPLOYM_c3_0_3e0db133-c13b-4e3d-ac14-
    <state>VM_ALIVE_STATE</state>
<deployment_name>VNF2-DEPLOYMENT-em</deployment_name>
    <vm_id>507d67c2-1d00-4321-b9d1-da879af524f8</vm_id>
    <vm_id>dc168a6a-4aeb-4e81-abd9-91d7568b5f7c</vm_id>
    <vm_id>9ffec58b-4b9d-4072-b944-5413bf7fcf07</vm_id>
    <state>SERVICE_ACTIVE_STATE</state>
    <vm_name>VNF2-DEPLOYM_XXXX_0_c8d98f0f-d874-45d0-af75-88a2d6fa82ea</vm_name>
    <state>VM_SHUTOFF_STATE</state>
<snip>

```

Log in to the ESC hosted in the Compute 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!
```

## Replace Motherboard

The steps in order to replace the motherboard in a UCS C240 M4 server can be referred from: [Cisco UCS C240 M4 Server Installation and Service Guide](#)

Log in to the server with the use of the CIMC IP.

Perform BIOS upgrade if the firmware is not as per the recommended version used previously. Steps for BIOS upgrade are given here: [Cisco UCS C-Series Rack-Mount Server BIOS Upgrade Guide](#)

## Restore the VMs

### Case 1. Compute Node Hosts only SF VM

The SF 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 |
```

Recover the SF VM 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.ZpRCGieiuW  
<?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>
```

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].
```

Ensure that the SF card comes up as standby SF in the VNF.

### Case 2. Compute Node Hosts UAS, ESC, EM and CF

#### Recovery of UAS VM

Check the status of the UAS VM in the nova list and delete it:

```
[stack@director ~]$ nova list | grep VNF2-UAS-uas-0
| 307a704c-a17c-4cdc-8e7a-3d6e7e4332fa | VNF2-UAS-uas-0
| ACTIVE | - | Running | VNF2-UAS-uas-
orchestration=172.168.11.10; VNF2-UAS-uas-management=172.168.10.3
[stack@tb5-ospd ~]$ nova delete VNF2-UAS-uas-0
Request to delete server VNF2-UAS-uas-0 has been accepted.
```

In order to recover the AutoVNF-UAS VM, run the UAS-check script to check state. It must report an error. Then run again with **--fix** option in order to recreate the missing UAS VM:

```
[stack@director ~]$ cd /opt/cisco/usp/uas-installer/scripts/
[stack@director scripts]$ ./uas-check.py auto-vnf VNF2-UAS
2017-12-08 12:38:05,446 - INFO: Check of AutoVNF cluster started
2017-12-08 12:38:07,925 - INFO: Instance 'vnf1-UAS-uas-0' status is 'ERROR'
2017-12-08 12:38:07,925 - INFO: Check completed, AutoVNF cluster has recoverable errors

[stack@director scripts]$ ./uas-check.py auto-vnf VNF2-UAS --fix
2017-11-22 14:01:07,215 - INFO: Check of AutoVNF cluster started
2017-11-22 14:01:09,575 - INFO: Instance VNF2-UAS-uas-0' status is 'ERROR'
2017-11-22 14:01:09,575 - INFO: Check completed, AutoVNF cluster has recoverable errors
2017-11-22 14:01:09,778 - INFO: Removing instance VNF2-UAS-uas-0'
2017-11-22 14:01:13,568 - INFO: Removed instance VNF2-UAS-uas-0'
2017-11-22 14:01:13,568 - INFO: Creating instance VNF2-UAS-uas-0' and attaching volume 'VNF2-
UAS-uas-vol-0'
2017-11-22 14:01:49,525 - INFO: Created instance 'VNF2-UAS-uas-0'
```

Log in to **AutoVNF-UAS**. Wait for a few minutes and then you will see that the UAS returns to the good state:

```
VNF2-autovnf-uas-0#show uas
uas version 1.0.1-1
uas state ha-active
uas ha-vip 172.17.181.101
INSTANCE IP STATE ROLE
-----
172.17.180.6 alive CONFID-SLAVE
172.17.180.7 alive CONFID-MASTER
172.17.180.9 alive NA
```

## Recovery of ESC VM

Check the status of the ESC VM from the nova list and delete it:

```
stack@director scripts]$ nova list |grep ESC-1
| c566efbf-1274-4588-a2d8-0682e17b0d41 | VNF2-ESC-ESC-
1 | ACTIVE | - | Running | VNF2-
UAS-uas-orchestration=172.168.11.14; VNF2-UAS-uas-
management=172.168.10.4
|
[stack@director scripts]$ nova delete VNF2-ESC-ESC-1
Request to delete server VNF2-ESC-ESC-1 has been accepted.
```

From AutoVNF-UAS, find the ESC deployment transaction and in the log for the transaction, find

the **boot\_vm.py** command line in order to create the ESC instance:

```
ubuntu@VNF2-uas-uas-0:~$ sudo -i
root@VNF2-uas-uas-0:~# confd_cli -u admin -C
Welcome to the ConfD CLI
admin connected from 127.0.0.1 using console on VNF2-uas-uas-0
VNF2-uas-uas-0#show transaction
TX ID                                TX TYPE                                DEPLOYMENT ID
TIMESTAMP                            STATUS
-----
35eefc4a-d4a9-11e7-bb72-fa163ef8df2b  vnf-deployment                        VNF2-DEPLOYMENT  2017-11-
29T02:01:27.750692-00:00  deployment-success
73d9c540-d4a8-11e7-bb72-fa163ef8df2b  vnf-deployment                        VNF2-ESC        2017-11-
29T01:56:02.133663-00:00  deployment-success
```

```
VNF2-uas-uas-0#show logs 73d9c540-d4a8-11e7-bb72-fa163ef8df2b | display xml
<config xmlns="http://tail-f.com/ns/config/1.0">
  <logs xmlns="http://www.cisco.com/usp/nfv/usp-autovnf-oper">
    <tx-id>73d9c540-d4a8-11e7-bb72-fa163ef8df2b</tx-id>
    <log>2017-11-29 01:56:02,142 - VNF Deployment RPC triggered for deployment: VNF2-ESC,
deactivate: 0
2017-11-29 01:56:02,179 - Notify deployment
..
2017-11-29 01:57:30,385 - Creating VNF 'VNF2-ESC-ESC-1' with [python //opt/cisco/vnf-
staging/bootvm.py VNF2-ESC-ESC-1 --flavor VNF2-ESC-ESC-flavor --image 3fe6b197-961b-4651-af22-
dfd910436689 --net VNF2-UAS-uas-management --gateway_ip 172.168.10.1 --net VNF2-UAS-uas-
orchestration --os_auth_url http://10.1.2.5:5000/v2.0 --os_tenant_name core --os_username *****
--os_password ***** --bs_os_auth_url http://10.1.2.5:5000/v2.0 --bs_os_tenant_name core --
bs_os_username ***** --bs_os_password ***** --esc_ui_startup false --esc_params_file
/tmp/esc_params.cfg --encrypt_key ***** --user_pass ***** --user_confid_pass ***** --kad_vif
eth0 --kad_vip 172.168.10.7 --ipaddr 172.168.10.6 dhcp --ha_node_list 172.168.10.3 172.168.10.6
--file root:0755:/opt/cisco/esc/esc-
scripts/esc_volume_em_staging.sh:/opt/cisco/usp/uas/autovnf/vnfms/esc-
scripts/esc_volume_em_staging.sh --file root:0755:/opt/cisco/esc/esc-
scripts/esc_vpc_chassis_id.py:/opt/cisco/usp/uas/autovnf/vnfms/esc-scripts/esc_vpc_chassis_id.py
--file root:0755:/opt/cisco/esc/esc-scripts/esc-vpc-di-internal-
keys.sh:/opt/cisco/usp/uas/autovnf/vnfms/esc-scripts/esc-vpc-di-internal-keys.sh
Save the boot_vm.py line to a shell script file (esc.sh) and update all the username ***** and
password ***** lines with the correct information (typically core/<PASSWORD>). You need to
remove the --encrypt_key option as well. For user_pass and user_confid_pass, you need to use
the format – username: password (example - admin:<PASSWORD>).
```

Find the URL to **bootvm.py** from running-config and get the **bootvm.py** file to the **AutoVNF UAS VM**. In this case, 10.1.2.3 is the AutoIT VM's IP:

```
root@VNF2-uas-uas-0:~# confd_cli -u admin -C
Welcome to the ConfD CLI
admin connected from 127.0.0.1 using console on VNF2-uas-uas-0
VNF2-uas-uas-0#show running-config autovnf-vnfm:vnfm
...
configs bootvm
  value http:// 10.1.2.3:80/bundles/5.1.7-2007/vnfm-bundle/bootvm-2_3_2_155.py
!

root@VNF2-uas-uas-0:~# wget http://10.1.2.3:80/bundles/5.1.7-2007/vnfm-bundle/bootvm-
2_3_2_155.py
--2017-12-01 20:25:52-- http://10.1.2.3 /bundles/5.1.7-2007/vnfm-bundle/bootvm-2_3_2_155.py
```

```

Connecting to 10.1.2.3:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 127771 (125K) [text/x-python]
Saving to: 'bootvm-2_3_2_155.py'
100%[=====]
127,771 --.-K/s in 0.001s
2017-12-01 20:25:52 (173 MB/s) - 'bootvm-2_3_2_155.py' saved [127771/127771]

```

### Create a `/tmp/esc_params.cfg` file:

```
root@VNF2-uas-uas-0:~# echo "openstack.endpoint=publicURL" > /tmp/esc_params.cfg
```

### Execute shell script in order to deploy **ESC** from the UAS node:

```

root@VNF2-uas-uas-0:~# /bin/sh esc.sh
+ python ./bootvm.py VNF2-ESC-ESC-1 --flavor VNF2-ESC-ESC-flavor --image 3fe6b197-961b-4651-af22-dfd910436689
--net VNF2-UAS-uas-management --gateway_ip 172.168.10.1 --net VNF2-UAS-uas-orchestration --os_auth_url
http://10.1.2.5:5000/v2.0 --os_tenant_name core --os_username core --os_password <PASSWORD> --bs_os_auth_url
http://10.1.2.5:5000/v2.0 --bs_os_tenant_name core --bs_os_username core --bs_os_password <PASSWORD>
--esc_ui_startup false --esc_params_file /tmp/esc_params.cfg --user_pass admin:<PASSWORD> --user_confid_pass
admin:<PASSWORD> --kad_vif eth0 --kad_vip 172.168.10.7 --ipaddr 172.168.10.6 dhcp --ha_node_list 172.168.10.3
172.168.10.6 --file root:0755:/opt/cisco/esc/esc-scripts/esc_volume_em_staging.sh:/opt/cisco/usp/uas/autovnf/vnfms/esc-scripts/esc_volume_em_staging.sh
--file root:0755:/opt/cisco/esc/esc-scripts/esc_vpc_chassis_id.py:/opt/cisco/usp/uas/autovnf/vnfms/esc-scripts/esc_vpc_chassis_id.py
--file root:0755:/opt/cisco/esc/esc-scripts/esc-vpc-di-internal-keys.sh:/opt/cisco/usp/uas/autovnf/vnfms/esc-scripts/esc-vpc-di-internal-keys.sh

```

### Log in to new **ESC** and verify the Backup state:

```

ubuntu@VNF2-uas-uas-0:~$ ssh admin@172.168.11.14
...
#####
#   ESC on VNF2-esc-esc-1.novalocal is in BACKUP state.
#####

[admin@VNF2-esc-esc-1 ~]$ escadm status
0 ESC status=0 ESC Backup Healthy

[admin@VNF2-esc-esc-1 ~]$ health.sh
===== ESC HA (BACKUP) =====
ESC HEALTH PASSED

```

### Recover CF and EM VMs from ESC

### Check the status of the CF and EM VMs from the nova list. They must be in the **ERROR** state:

```

[stack@director ~]$ source corerc
[stack@director ~]$ nova list --field name,host,status |grep -i err
| 507d67c2-1d00-4321-b9d1-da879af524f8 | VNF2-DEPLOYM_XXXX_0_c8d98f0f-d874-45d0-af75-88a2d6fa82ea | None | ERROR|

```

```
| f9c0763a-4a4f-4bbd-af51-bc7545774be2 | VNF2-DEPLOYM_c1_0_df4be88d-b4bf-4456-945a-3812653ee229 |None | ERROR
```

Log in to ESC Master, run **recovery-vm-action** for each impacted EM and CF VM. Be patient. ESC schedules the recovery-action and it might not happen for a few minutes. Monitor the **yangesc.log**:

```
sudo /opt/cisco/esc/esc-confd/esc-cli/esc_nc_cli recovery-vm-action DO <VM-Name>
```

```
[admin@VNF2-esc-esc-0 ~]$ sudo /opt/cisco/esc/esc-confd/esc-cli/esc_nc_cli recovery-vm-action DO VNF2-DEPLOYMENT-_VNF2-D_0_a6843886-77b4-4f38-b941-74eb527113a8 [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.ZpRCG1ieuW <?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>
```

```
[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-DEPLOYMENT-_VNF2-D_0_a6843886-77b4-4f38-b941-74eb527113a8]
```

Log in to new EM and verify that the EM state is up:

```
ubuntu@VNF2vnfddeploymentem-1:~$ /opt/cisco/ncs/current/bin/ncs_cli -u admin -C admin connected from 172.17.180.6 using ssh on VNF2vnfddeploymentem-1 admin@scm# show ems EM VNFM ID SLA SCM PROXY ----- 2 up up up 3 up up up
```

Log in to the StarOS VNF and verify that the CF card is in the standby state.

## Handle ESC Recovery Failure

In cases where ESC fails to start the VM due to an unexpected state, it is recommended to perform an ESC switchover by rebooting the Master ESC. The ESC switchover takes about a minute. Run the script **health.sh** on the new Master ESC in order to check if the status is up. Master ESC to start the VM and fix the VM state. This recovery task takes 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 needs to go and do the manual recovery of the impacted VM(s).

# Motherboard Replacement in OSD Compute Node

Before the activity, the VMs hosted in the Compute node are gracefully shutdown and the Ceph is put into the maintenance mode. Once the motherboard has been replaced, the VMs are restored back and Ceph is moved out of the maintenance mode.

## Put Ceph in Maintenance Mode

Verify that the **ceph osd tree** status are up in the server.

```
[heat-admin@pod1-osd-compute-1 ~]$ sudo ceph osd tree
```

ID	WEIGHT	TYPE	NAME	UP/DOWN	REWEIGHT	PRIMARY-AFFINITY
-1	13.07996	root	default			
-2	4.35999	host	pod1-osd-compute-0			
0	1.09000		osd.0	up	1.00000	1.00000
3	1.09000		osd.3	up	1.00000	1.00000
6	1.09000		osd.6	up	1.00000	1.00000
9	1.09000		osd.9	up	1.00000	1.00000
-3	4.35999	host	pod1-osd-compute-2			
1	1.09000		osd.1	up	1.00000	1.00000
4	1.09000		osd.4	up	1.00000	1.00000
7	1.09000		osd.7	up	1.00000	1.00000
10	1.09000		osd.10	up	1.00000	1.00000
-4	4.35999	host	pod1-osd-compute-1			
2	1.09000		osd.2	up	1.00000	1.00000
5	1.09000		osd.5	up	1.00000	1.00000
8	1.09000		osd.8	up	1.00000	1.00000
11	1.09000		osd.11	up	1.00000	1.00000

Log into the OSD Compute node and put Ceph in the maintenance mode.

```
[root@pod1-osd-compute-1 ~]# sudo ceph osd set norebalance
```

```
[root@pod1-osd-compute-1 ~]# sudo ceph osd set noout
```

```
[root@pod1-osd-compute-1 ~]# 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 {pod1-controller-0=11.118.0.40:6789/0,pod1-controller-1=11.118.0.41:6789/0,pod1-controller-2=11.118.0.42:6789/0}
election epoch 58, quorum 0,1,2 pod1-controller-0,pod1-controller-1,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
```

**Note:** When Ceph is removed, VNF HD RAID goes into the Degraded state but HDD must still be accessible.

## Identify the VMs Hosted in the Osd-Compute Node

Identify the VMs that are hosted on the OSD Compute server. There can be two possibilities:

The osd-compute server contains Element Manager (EM)/UAS/Auto-Deploy/Auto-IT combination of VMs:

```
[stack@director ~]$ nova list --field name,host | grep osd-compute-0
| c6144778-9afd-4946-8453-78c817368f18 | AUTO-DEPLOY-VNF2-uas-0 | pod1-osd-compute-0.localdomain
|
| 2d051522-bce2-4809-8d63-0c0e17f251dc | AUTO-IT-VNF2-uas-0 | pod1-osd-compute-0.localdomain |
| 507d67c2-1d00-4321-b9d1-da879af524f8 | VNF2-DEPLOYM_XXXX_0_c8d98f0f-d874-45d0-af75-
88a2d6fa82ea | pod1-osd-compute-0.localdomain |
| f5bd7b9c-476a-4679-83e5-303f0aae9309 | VNF2-UAS-uas-0 | pod1-osd-compute-0.localdomain |
```

The Compute Server contains Control Function (CF)/Elastic Services Controller (ESC)/Element Manager (EM)/ (UAS) combination of VMs:

```
[stack@director ~]$ nova list --field name,host | grep osd-compute-1
| 507d67c2-1d00-4321-b9d1-da879af524f8 | VNF2-DEPLOYM_XXXX_0_c8d98f0f-d874-45d0-af75-
88a2d6fa82ea | pod1-compute-8.localdomain |
| f9c0763a-4a4f-4bbd-af51-bc7545774be2 | VNF2-DEPLOYM_c1_0_df4be88d-b4bf-4456-945a-
3812653ee229 | pod1-compute-8.localdomain |
| 75528898-ef4b-4d68-b05d-882014708694 | VNF2-ESC-ESC-
0 | pod1-compute-8.localdomain |
| f5bd7b9c-476a-4679-83e5-303f0aae9309 | VNF2-UAS-uas-
0 | pod1-compute-8.localdomain |
```

**Note:** In the output shown here, the first column corresponds to the UUID, the second column is the VM name and the third column is the hostname where the VM is present. The parameters from this output will be used in subsequent sections.

## Graceful Power Off

### Case 1. OSD-Compute Node Hosts CF/ESC/EM/UAS

Procedure to gracefully power of CF/ESC/EM/UAS VMs is same irrespective of whether the VMs are hosted in Compute or OSD-Compute node. Follow steps from **Motherboard Replacement in Compute Node** to gracefully power off the VMs.

### Case 2. OSD-Compute Node Hosts Auto-Deploy/Auto-It/EM/UAS

#### Backup the CDB of Auto-Deploy

Backup the autodeploy confd cdb data periodically or after every activation/deactivation and save the file to a backup server. Autodeploy is not redundant and if this data is lost, you will not be able to gracefully deactivate the deployment.

Log in to AutoDeploy VM and backup confd cdb directory.

```
ubuntu@auto-deploy-iso-2007-uas-0:~ $sudo -i
```

```
root@auto-deploy-iso-2007-uas-0:~#service uas-confd stop
uas-confd stop/waiting
```

```
root@auto-deploy-iso-2007-uas-0:~# cd /opt/cisco/usp/uas/confd-6.3.1/var/confd
root@auto-deploy-iso-2007-uas-0:/opt/cisco/usp/uas/confd-6.3.1/var/confd#tar cvf
autodeploy_cdb_backup.tar cdb/
cdb/
cdb/O.cdb
cdb/C.cdb
cdb/aaa_init.xml
cdb/A.cdb
```

```
root@auto-deploy-iso-2007-uas-0:~# service uas-confd start
uas-confd start/running, process 13852
```

**Note:** opy autodeploy\_cdb\_backup.tar in order to back up the server.

## Backup System.cfg from Auto-IT

Take a backup of **system.cfg** file to backup the server:

```
Auto-it = 10.1.1.2
Backup server = 10.2.2.2
```

```
[stack@director ~]$ ssh ubuntu@10.1.1.2
ubuntu@10.1.1.2's password:
Welcome to Ubuntu 14.04.3 LTS (GNU/Linux 3.13.0-76-generic x86_64)
```

```
* Documentation: https://help.ubuntu.com/
```

```
System information as of Wed Jun 13 16:21:34 UTC 2018
```

```
System load:  0.02          Processes:      87
Usage of /:   15.1% of 78.71GB  Users logged in:  0
Memory usage: 13%          IP address for eth0: 172.16.182.4
Swap usage:   0%
```

```
Graph this data and manage this system at:
https://landscape.canonical.com/
```

```
Get cloud support with Ubuntu Advantage Cloud Guest:
http://www.ubuntu.com/business/services/cloud
```

```
Cisco Ultra Services Platform (USP)
Build Date: Wed Feb 14 12:58:22 EST 2018
Description: UAS build assemble-uas#1891
shal: bf02ced
```

```
ubuntu@auto-it-vnf-uas-0:~$ scp -r /opt/cisco/usp/uploads/system.cfg root@10.2.2.2:/home/stack
root@10.2.2.2's password:
system.cfg
```

```
100% 565 0.6KB/s 00:00
```

```
ubuntu@auto-it-vnf-uas-0:~$
```

**Note:** Procedure to gracefully power of EM/UAS VMs is same irrespective of whether the VMs are hosted in Compute or OSD-Compute node.



Follow steps from **Motherboard Replacement in Compute Node** in order to gracefully power off these VMs.

## Replace Motherboard

The steps in order to replace the motherboard in a UCS C240 M4 server can be referred from: [Cisco UCS C240 M4 Server Installation and Service Guide](#)

Log in to the server with the use of the CIMC IP.

Perform BIOS upgrade if the firmware is not as per the recommended version used previously. Steps for BIOS upgrade are given here: [Cisco UCS C-Series Rack-Mount Server BIOS Upgrade Guide](#)

## Move Ceph Out of the Maintenance Mode

Log into the OSD Compute node and move Ceph out of the maintenance mode.

```
[root@pod1-osd-compute-1 ~]# sudo ceph osd unset norebalance
[root@pod1-osd-compute-1 ~]# sudo ceph osd unset noout

[root@pod1-osd-compute-1 ~]# sudo ceph status

cluster eb2bb192-b1c9-11e6-9205-525400330666
health HEALTH_OK
monmap e1: 3 mons at {pod1-controller-0=11.118.0.40:6789/0,pod1-controller-1=11.118.0.41:6789/0,pod1-controller-2=11.118.0.42:6789/0}
election epoch 58, quorum 0,1,2 pod1-controller-0,pod1-controller-1,pod1-controller-2
osdmap e196: 12 osds: 12 up, 12 in
flags sortbitwise,require_jewel_osds
pgmap v584954: 704 pgs, 6 pools, 531 GB data, 344 kobjects
1585 GB used, 11808 GB / 13393 GB avail
704 active+clean
client io 12888 kB/s wr, 0 op/s rd, 81 op/s wr
```

## Restore the VMs

### Case 1. OSD-Compute Node Hosts CF, ESC, EM and UAS

Procedure to restore of CF/ESC/EM/UAS VMs is same irrespective of whether the VMs are hosted in Compute or OSD-Compute node. Follow steps from **Case 2. Compute Node Hosts CF/ESC/EM/UAS** to restore the VMs.

### Case 2. OSD-Compute Node Hosts Auto-It, Auto-Deploy, EM and UAS

#### Recover Auto-Deploy VM

From OSPD, if auto-deploy VM was impacted but still shows ACTIVE/Running, you will need to delete it first. If auto-deploy was not impacted, skip to Recovery of Auto-it VM:

```
[stack@director ~]$ nova list |grep auto-deploy
| 9b55270a-2dcd-4ac1-aba3-bf041733a0c9 | auto-deploy-ISO-2007-uas-
0 | ACTIVE | - | Running | mgmt=172.16.181.12,
10.1.2.7 [stack@director ~]$ cd /opt/cisco/usp/uas-installer/scripts
```

```
[stack@director ~]$ ./auto-deploy-booting.sh --floating-ip 10.1.2.7 --delete
```

Once auto-deploy is deleted, create it again with same **floatingip** address:

```
[stack@director ~]$ cd /opt/cisco/usp/uas-installer/scripts
```

```
[stack@director scripts]$ ./auto-deploy-booting.sh --floating-ip 10.1.2.7
```

```
2017-11-17 07:05:03,038 - INFO: Creating AutoDeploy deployment (1 instance(s)) on
'http://10.84.123.4:5000/v2.0' tenant 'core' user 'core', ISO 'default'
2017-11-17 07:05:03,039 - INFO: Loading image 'auto-deploy-ISO-5-1-7-2007-usp-uas-1.0.1-
1504.qcow2' from '/opt/cisco/usp/uas-installer/images/usp-uas-1.0.1-1504.qcow2'
2017-11-17 07:05:14,603 - INFO: Loaded image 'auto-deploy-ISO-5-1-7-2007-usp-uas-1.0.1-
1504.qcow2'
2017-11-17 07:05:15,787 - INFO: Assigned floating IP '10.1.2.7' to IP '172.16.181.7'
2017-11-17 07:05:15,788 - INFO: Creating instance 'auto-deploy-ISO-5-1-7-2007-uas-0'
2017-11-17 07:05:42,759 - INFO: Created instance 'auto-deploy-ISO-5-1-7-2007-uas-0'
2017-11-17 07:05:42,759 - INFO: Request completed, floating IP: 10.1.2.7
```

Copy the **Autodeploy.cfg** file, ISO and the **confd\_backup** tar file from your backup server to autodeploy VM and restore **confd cdb** files from backup tar file:

```
ubuntu@auto-deploy-iso-2007-uas-0:~# sudo -i
```

```
ubuntu@auto-deploy-iso-2007-uas-0:~# service uas-confd stop
uas-confd stop/waiting
```

```
root@auto-deploy-iso-2007-uas-0:~# cd /opt/cisco/usp/uas/confd-6.3.1/var/confd
root@auto-deploy-iso-2007-uas-0:/opt/cisco/usp/uas/confd-6.3.1/var/confd# tar xvf
/home/ubuntu/ad_cdb_backup.tar
```

```
cdb/
cdb/O.cdb
cdb/C.cdb
cdb/aaa_init.xml
cdb/A.cdb
```

```
root@auto-deploy-iso-2007-uas-0:~# service uas-confd start
uas-confd start/running, process 2036
```

Verify that **confd** was loaded properly by checking earlier transactions. Update the **autodeploy.cfg** with new **osd-compute** name. See Section- **Final Step: Update AutoDeploy Configuration**.

```
root@auto-deploy-iso-2007-uas-0:~# confd_cli -u admin -C
```

```
Welcome to the ConfD CLI
admin connected from 127.0.0.1 using console on auto-deploy-iso-2007-uas-0
```

```
auto-deploy-iso-2007-uas-0#show transaction
```

```

SERVICE
SITE
DEPLOYMENT
SITE TX AUTOVNF VNF AUTOVNF
```

TX ID	TX TYPE	ID	DATE AND TIME
STATUS	ID	ID	TX ID
1512571978613	service-deployment	tb5bxb	2017-12-06T14:52:59.412+00:00 deployment-success

auto-deploy-iso-2007-uas-0# exit

## Recover Auto-IT VM

From OSPD, if auto-it VM was impacted but still shows as ACTIVE/Running, you need to delete it. If auto-it wasn't impacted, skip to the next:

```
[stack@director ~]$ nova list |grep auto-it
| 580faf80-1d8c-463b-9354-781ea0c0b352 | auto-it-vnf-ISO-2007-uas-
0 | ACTIVE | - | Running | mgmt=172.16.181.3,
10.1.2.8 [stack@director ~]$ cd /opt/cisco/usp/uas-installer/scripts
```

```
[stack@director ~]$ ./ auto-it-vnf-staging.sh --floating-ip 10.1.2.8 --delete
```

Recreate Auto-IT by running Auto-IT-VNF staging script:

```
[stack@director ~]$ cd /opt/cisco/usp/uas-installer/scripts
```

```
[stack@director scripts]$ ./auto-it-vnf-staging.sh --floating-ip 10.1.2.8
```

```
2017-11-16 12:54:31,381 - INFO: Creating StagingServer deployment (1 instance(s)) on
'http://10.84.123.4:5000/v2.0' tenant 'core' user 'core', ISO 'default'
2017-11-16 12:54:31,382 - INFO: Loading image 'auto-it-vnf-ISO-5-1-7-2007-usp-uas-1.0.1-
1504.qcow2' from '/opt/cisco/usp/uas-installer/images/usp-uas-1.0.1-1504.qcow2'
2017-11-16 12:54:51,961 - INFO: Loaded image 'auto-it-vnf-ISO-5-1-7-2007-usp-uas-1.0.1-
1504.qcow2'
2017-11-16 12:54:53,217 - INFO: Assigned floating IP '10.1.2.8' to IP '172.16.181.9'
2017-11-16 12:54:53,217 - INFO: Creating instance 'auto-it-vnf-ISO-5-1-7-2007-uas-0'
2017-11-16 12:55:20,929 - INFO: Created instance 'auto-it-vnf-ISO-5-1-7-2007-uas-0'
2017-11-16 12:55:20,930 - INFO: Request completed, floating IP: 10.1.2.8
```

Reload the ISO image. In this case, Auto-IT IP address is 10.1.2.8. This will take a few minutes to load:

```
[stack@director ~]$ cd images/5_1_7-2007/isos
```

```
[stack@director isos]$ curl -F file=@usp-5_1_7-2007.iso http://10.1.2.8:5001/isos
{
  "iso-id": "5.1.7-2007"
}
```

to check the ISO image:

```
[stack@director isos]$ curl http://10.1.2.8:5001/isos
```

```
{
  "isos": [
    {
      "iso-id": "5.1.7-2007"
    }
  ]
}
```

Copy the VNF **system.cfg** files from OSPD Auto-Deploy directory to Auto-IT VM:

```
[stack@director autodeploy]$ scp system-vnf* ubuntu@10.1.2.8:.
```

```
ubuntu@10.1.2.8's password:
```

```
system-
```

```
vnf1.cfg
```

```
100% 1197 1.2KB/s 00:00
```

```
system-vnf2.cfg
```

```
100% 1197 1.2KB/s 00:00
```

```
ubuntu@auto-it-vnf-iso-2007-uas-0:~$ pwd
```

```
/home/ubuntu
```

```
ubuntu@auto-it-vnf-iso-2007-uas-0:~$ ls
```

```
system-vnf1.cfg system-vnf2.cfg
```

**Note:** Recovery procedure of EM and UAS VM are same irrespective of whether the VM is hosted in Compute or OSD-Compute. Follow steps from **Replace Motherboard in Compute Node** in order to gracefully power off these VMs.

## Replace Motherboard in Controller Node

### Verify Controller Status and Put Cluster in Maintenance Mode

From OSPD, log in to the controller and verify that the PCs are in a good state – all three controllers Online and Galera shows all three controllers as Master.

```
[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-controller-0 (stonith:fence_ipmilan): Started pod1-controller-0
```

```
my-ipmilan-for-controller-1 (stonith:fence_ipmilan): Started pod1-controller-0
```

```
my-ipmilan-for-controller-2 (stonith:fence_ipmilan): Started pod1-controller-0
```

Daemon Status:

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

Put the cluster in maintenance mode:

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

```
[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 00:48:24 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-controller-0 (stonith:fence_ipmilan): Started pod1-controller-1  
my-ipmilan-for-controller-1 (stonith:fence_ipmilan): Started pod1-controller-1  
my-ipmilan-for-controller-2 (stonith:fence_ipmilan): Started pod1-controller-2
```

## Replace Motherboard

The steps in order to replace the motherboard in a UCS C240 M4 server can be referred from: [Cisco UCS C240 M4 Server Installation and Service Guide](#)

Log in to the server with the use of the CIMC IP.

Perform BIOS upgrade if the firmware is not as per the recommended version used previously. Steps for BIOS upgrade are given here: [Cisco UCS C-Series Rack-Mount Server BIOS Upgrade Guide](#)

## Restore Cluster Status

Log in to the controller that has been impacted, remove standby mode by setting **unstandby**. Verify whether the controller comes online with cluster and Galera shows all three controllers as Master. This might 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-controller-0 (stonith:fence_ipmilan): Started pod1-controller-1
```

```
my-ipmilan-for-controller-1 (stonith:fence_ipmilan): Started pod1-controller-1
```

```
my-ipmilan-for-controller-2 (stonith:fence_ipmilan): Started pod1-controller-2
```

```
Daemon Status:
```

```
corosync: active/enabled
```

```
pacemaker: active/enabled
```

```
pcsd: active/enabled
```