

# Backup and Restore Procedures for Various Ultra-M Components

## Contents

[Introduction](#)

[Background Information](#)

[Abbreviations](#)

[Backup Procedures](#)

[OSPD Backup](#)

[AutoDeploy Backup](#)

[AutoIT-VNF Backup](#)

[AutoVNF Backup](#)

[ESC Backup](#)

[EM Backup](#)

[StarOS Backup](#)

[Restore Procedure](#)

[OSPD Recovery](#)

[AutoDeploy Recovery](#)

[AutoIT-VNF Recovery](#)

[AutoVNF Recovery](#)

[ESC Recovery](#)

[EM Recovery](#)

[StarOS Recovery](#)

## Introduction

This document describes the steps required in order to backup and restore a Virtual Machine (VM) 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 designed to simplify the deployment of VNFs. Ultra-M solution consists of the following Virtual Machine (VM) types:

- Auto-IT
- Auto-Deploy
- Ultra Automation Services (UAS)
- Element Manager (EM)
- Elastic Services Controller (ESC)
- Control Function (CF)
- Session Function (SF)

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



This document is intended for the Cisco personnel who are familiar with Cisco Ultra-M platform.

**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

## Backup Procedures

# OSPD Backup

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

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

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

```
[stack@director ~]$ systemctl list-units "openstack*" "neutron*" "openvswitch*"
```

UNIT	LOAD	ACTIVE	SUB	DESCRIPTION
neutron-dhcp-agent.service	loaded	active	running	OpenStack Neutron DHCP Agent
neutron-openvswitch-agent.service	loaded	active	running	OpenStack Neutron Open vSwitch Agent
neutron-ovs-cleanup.service	loaded	active	exited	OpenStack Neutron Open vSwitch Cleanup Utility
neutron-server.service	loaded	active	running	OpenStack Neutron Server
openstack-aodh-evaluator.service	loaded	active	running	OpenStack Alarm evaluator service
openstack-aodh-listener.service	loaded	active	running	OpenStack Alarm listener service
openstack-aodh-notifier.service	loaded	active	running	OpenStack Alarm notifier service
openstack-ceilometer-central.service	loaded	active	running	OpenStack ceilometer central agent
openstack-ceilometer-collector.service	loaded	active	running	OpenStack ceilometer collection service
openstack-ceilometer-notification.service	loaded	active	running	OpenStack ceilometer notification agent
openstack-glance-api.service	loaded	active	running	OpenStack Image Service (code-named Glance) API server
openstack-glance-registry.service	loaded	active	running	OpenStack Image Service (code-named Glance) Registry server
openstack-heat-api-cfn.service	loaded	active	running	Openstack Heat CFN-compatible API Service
openstack-heat-api.service	loaded	active	running	OpenStack Heat API Service
openstack-heat-engine.service	loaded	active	running	Openstack Heat Engine Service
openstack-ironic-api.service	loaded	active	running	OpenStack Ironic API service
openstack-ironic-conductor.service	loaded	active	running	OpenStack Ironic Conductor service
openstack-ironic-inspector-dnsmasq.service	loaded	active	running	PXE boot dnsmasq service for Ironic Inspector
openstack-ironic-inspector.service	loaded	active	running	Hardware introspection service for OpenStack Ironic
openstack-mistral-api.service	loaded	active	running	Mistral API Server
openstack-mistral-engine.service	loaded	active	running	Mistral Engine Server
openstack-mistral-executor.service	loaded	active	running	Mistral Executor Server
openstack-nova-api.service	loaded	active	running	OpenStack Nova API Server
openstack-nova-cert.service	loaded	active	running	OpenStack Nova Cert Server
openstack-nova-compute.service	loaded	active	running	OpenStack Nova Compute Server
openstack-nova-conductor.service	loaded	active	running	OpenStack Nova Conductor Server
openstack-nova-scheduler.service	loaded	active	running	OpenStack Nova Scheduler Server
openstack-swift-account-reaper.service	loaded	active	running	OpenStack Object Storage (swift) - Account Reaper
openstack-swift-account.service	loaded	active	running	OpenStack Object Storage

```

(swift) - Account Server
openstack-swift-container-updater.service loaded active running OpenStack Object Storage
(swift) - Container Updater
openstack-swift-container.service loaded active running OpenStack Object Storage
(swift) - Container Server
openstack-swift-object-updater.service loaded active running OpenStack Object Storage
(swift) - Object Updater
openstack-swift-object.service loaded active running OpenStack Object Storage
(swift) - Object Server
openstack-swift-proxy.service loaded active running OpenStack Object Storage
(swift) - Proxy Server
openstack-zaqar.service loaded active running OpenStack Message Queuing
Service (code-named Zaqar) Server
openstack-zaqar@1.service loaded active running OpenStack Message Queuing
Service (code-named Zaqar) Server Instance 1
openvswitch.service loaded active exited Open vSwitch

```

LOAD = Reflects whether the unit definition was properly loaded.

ACTIVE = The high-level unit activation state, i.e. generalization of SUB.

SUB = The low-level unit activation state, values depend on unit type.

37 loaded units listed. Pass --all to see loaded but inactive units, too.

To show all installed unit files use 'systemctl list-unit-files'.

3. Confirm that you have sufficient disk space available before you perform the backup process. This tarball is expected to be at least 3.5 GB.

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

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

```

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

```

## AutoDeploy Backup

1. AutoDeploy requires this data to be backed up:

- AutoDeploy Confd Configuration Database (CDB)
- AutoDeploy Configuration
- AutoDeploy Logs
- Syslog Configuration

2. Backup of the AutoDeploy Confd CDB data and running configuration is required after every activation/deactivation and ensure that the data is transferred to a backup server.

3. AutoDeploy executes in standalone mode and if this data is lost, you will not be able to gracefully deactivate the deployment. Hence, it is mandatory to take the backup of configuration and CDB data.

```

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:/home/ubuntu# service autodeploy status
autodeploy start/running, process 1313
```

```
root@auto-deploy:/home/ubuntu# service autodeploy stop
autodeploy 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
```

4. Copy **autodeploy\_cdb\_backup.tar** to the backup server.

5. Take a backup of the running configuration in the AutoDeploy and transfer it to a backup server.

```
root@auto-deploy:/home/ubuntu# confd_cli -u admin -C
```

```
Welcome to the ConfD CLI
```

```
admin connected from 127.0.0.1 using console on auto-deploy
```

```
auto-deploy#show running-config | save backup-config-$(date +%Y%m%d).cfg
auto-deploy#
```

6. Start the AutoDeploy ConfD service.

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

```
root@auto-deploy:/home/ubuntu# service autodeploy start
autodeploy start/running, process 8835
```

7. Navigate to the scripts directory and collect the logs from AutoDeploy VM.

```
cd /opt/cisco/usp/uas/scripts
```

8. Launch the **collect-uas-logs.sh** script in order to collect the logs.

```
sudo ./collect-uas-logs.sh
```

9. Take the ISO image backup from the AutoDeploy and transfer to the backup server.

```
root@POD1-5-1-7-2034-auto-deploy-uas-0:/home/ubuntu# /home/ubuntu/isos
root@POD1-5-1-7-2034-auto-deploy-uas-0:/home/ubuntu/isos# ll
```

```
total 4430888
drwxr-xr-x 2 root root 4096 Dec 20 01:17 ./
drwxr-xr-x 5 ubuntu ubuntu 4096 Dec 20 02:31 ../
-rwxr-xr-x 1 ubuntu ubuntu 4537214976 Oct 12 03:34 usp-5_1_7-2034.iso*
```

10. Collect syslog configuration and save it in the backup server.

```
ubuntu@auto-deploy-vnf-iso-5-1-5-1196-uas-0:~$sudo su
root@auto-deploy-vnf-iso-5-1-5-1196-uas-0:/home/ubuntu#ls /etc/rsyslog.d/00-autodeploy.conf
00-autodeploy.conf
root@auto-deploy-vnf-iso-5-1-5-1196-uas-0:/home/ubuntu#ls /etc/rsyslog.conf
rsyslog.conf
```

## AutoIT-VNF Backup

AutoIT-VNF is a stateless VM so there is no database (DB) which needs to be backed up. AutoIT-VNF is responsible to do the package management along with the configuration management repository for Ultra-M, hence, it is essential that those backups be taken.

1. Take the backup of day-0 StarOS configurations and transfer them to the backup server.

```
root@auto-it-vnf-iso-5-8-uas-0:/home/ubuntu# cd /opt/cisco/usp/uploads/
root@auto-it-vnf-iso-5-8-uas-0:/opt/cisco/usp/uploads# ll
total 12
drwxrwxr-x  2 uspadmin usp-data 4096 Nov  8 23:28 ./
drwxr-xr-x 15 root      root    4096 Nov  8 23:53 ../
-rw-rw-r--  1 ubuntu   ubuntu   985  Nov  8 23:28 system.cfg
```

2. Navigate to the scripts directory and collect the logs from AutoIT VM.

```
cd /opt/cisco/usp/uas/scripts
```

3. Launch the **collect-uas-logs.sh** script in order to collect the logs.

```
sudo ./collect-uas-logs.sh
```

4. Collect syslog configuration backup and save it in the backup server.

```
ubuntu@auto-it-vnf-iso-5-1-5-1196-uas-0:~$sudo su
root@auto-it-vnf-iso-5-1-5-1196-uas-0:/home/ubuntu#ls /etc/rsyslog.d/00-autoit-vnf.conf
00-autoit-vnf.conf
root@auto-it-vnf-iso-5-1-5-1196-uas-0:ls /etc/rsyslog.conf
rsyslog.conf
```

## AutoVNF Backup

AutoVNF is responsible in order to bring up individual VNFM and VNF. AutoDeploy sends the configuration required to instantiate both VNFM. and VNF to AutoVNF and AutoVNF does this operation. In order to bring up VNFM, AutoVNF will talk directly to VIM/OpenStack and after the VNFM is up, AutoVNF uses VNFM to bring up VNF.

AutoVNF has 1:N redundancy and in an Ultra-M setup, there are three AutoVNF VMs running. Single AutoVNF failure is supported in Ultra-M and recovery is feasible.

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

## AutoVNF backup details:

- Running Configuration
- ConfD CDB DB
- AutoVNF logs (from each instance of AutoVNF)
- Syslog Configuration

It is recommended to take backups before any activation/deactivation on the given site and uploaded to the backup server.

### 1. Log in to master AutoVNF and validate that it is **confd-master**.

```
root@auto-testautovnf1-uas-1:/home/ubuntu# confd_cli -u admin -C
```

```
Welcome to the ConfD CLI
admin connected from 127.0.0.1 using console on auto-testautovnf1-uas-1
auto-testautovnf1-uas-1#show uas
uas version 1.0.1-1
uas state ha-active
uas ha-vip 172.57.11.101
```

```
INSTANCE IP    STATE  ROLE
-----
172.57.12.6   alive  CONFD-SLAVE
172.57.12.7   alive  CONFD-MASTER
172.57.12.13  alive  NA
```

```
auto-testautovnf1-uas-1#exit
```

```
root@auto-testautovnf1-uas-1:/home/ubuntu# ip a
```

```
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever

2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen
1000
    link/ether fa:16:3e:c7:dc:89 brd ff:ff:ff:ff:ff:ff
    inet 172.57.12.7/24 brd 172.57.12.255 scope global eth0
        valid_lft forever preferred_lft forever
    inet6 fe80::f816:3eff:fec7:dc89/64 scope link
        valid_lft forever preferred_lft forever

3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen
1000
    link/ether fa:16:3e:10:29:1b brd ff:ff:ff:ff:ff:ff
    inet 172.57.11.101/24 brd 172.57.11.255 scope global eth1
        valid_lft forever preferred_lft forever
    inet6 fe80::f816:3eff:fe10:291b/64 scope link
        valid_lft forever preferred_lft forever
```

### 2. Take a backup of the running configuration and transfer the file to the backup server.

```
root@auto-testautovnf1-uas-1:/home/ubuntu# confd_cli -u admin -C
```

```
Welcome to the ConfD CLI
admin connected from 127.0.0.1 using console on auto-testautovnf1-uas-1
auto-testautovnf1-uas-1#show running-config | save running-autovnf-12202017.cfg
auto-testautovnf1-uas-1#exit
root@auto-testautovnf1-uas-1:/home/ubuntu# ll running-autovnf-12202017.cfg
-rw-r--r-- 1 root root 18181 Dec 20 19:03 running-autovnf-12202017.cfg
```

### 3. Take a backup of CDB and transfer the file to the backup server.

```
root@auto-testautovnf1-uas-1:/opt/cisco/usp/uas/confd-6.3.1/var/confd# tar cvf
autovnf_cdb_backup.tar cdb/
cdb/
cdb/O.cdb
cdb/C.cdb
cdb/aaa_init.xml
cdb/vpc.xml
cdb/A.cdb
cdb/gilan.xml

root@auto-testautovnf1-uas-1:/opt/cisco/usp/uas/confd-6.3.1/var/confd#
root@auto-testautovnf1-uas-1:/opt/cisco/usp/uas/confd-6.3.1/var/confd# ll autovnf_cdb_backup.tar
-rw-r--r-- 1 root root 1198080 Dec 20 19:08 autovnf_cdb_backup.tar
```

### 4. Navigate to the scripts directory, collect the logs and transfer to the backup server.

```
cd /opt/cisco/usp/uas/scripts
```

```
sudo ./collect-uas-logs.sh
```

### 5. Log in to the standby instance of AutoVNF and perform these steps in order to collect the logs and transfer them to the backup server.

### 6. Backup the syslog configuration on the master and standby AutoVNF VMs and transfer them to the backup server.

```
ubuntu@auto-testautovnf1-uas-1:~$sudo su
root@auto-testautovnf1-uas-1:/home/ubuntu#ls /etc/rsyslog.d/00-autovnf.conf
00-autovnf.conf
```

```
root@auto-testautovnf1-uas-1:/home/ubuntu#ls /etc/rsyslog.conf
rsyslog.conf
```

## ESC Backup

1. AutoVNF is responsible in order to bring up ESC in an Ultra-M solution by interacting with VIM directly. AutoVNF/EM passes VNF specific configuration to ESC and ESC, in turn, will bring up VNF by interacting to VIM.

2. ESC has 1:1 redundancy in Ultra-M Solution. There are two ESC VMs deployed and that support single failure in Ultra-M. i.e. you can recover the system if there is a single failure in the system.

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



## ESC backup details:

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

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

4. Verify that the health of ESC is good with the use of **health.sh** script.

```
[root@auto-test-vnfm1-esc-0 admin]# escadm status
0 ESC status=0 ESC Master Healthy

[root@auto-test-vnfm1-esc-0 admin]# health.sh
esc ui is disabled -- skipping status check
esc_monitor start/running, process 836
esc_mona is up and running ...
vimmanager start/running, process 2741
vimmanager start/running, process 2741
esc_confd is started
tomcat6 (pid 2907) is running...           [ OK ]
postgresql-9.4 (pid 2660) is running...
ESC service is running...
Active VIM = OPENSTACK
ESC Operation Mode=OPERATION

/opt/cisco/esc/esc_database is a mountpoint

===== ESC HA (MASTER) with DRBD =====

DRBD_ROLE_CHECK=0
MNT_ESC_DATABASE_CHECK=0
VIMMANAGER_RET=0
ESC_CHECK=0
STORAGE_CHECK=0
ESC_SERVICE_RET=0
MONA_RET=0
ESC_MONITOR_RET=0

=====

ESC HEALTH PASSED
```

5. Take a backup of the running configuration and transfer the file to the backup server.

```
[root@auto-test-vnfm1-esc-0 admin]# /opt/cisco/esc/confd/bin/confd_cli -u admin -C
admin connected from 127.0.0.1 using console on auto-test-vnfm1-esc-0.novalocal
auto-test-vnfm1-esc-0# show running-config | save /tmp/running-esc-12202017.cfg
auto-test-vnfm1-esc-0#exit

[root@auto-test-vnfm1-esc-0 admin]# ll /tmp/running-esc-12202017.cfg
-rw-----. 1 tomcat tomcat 25569 Dec 20 21:37 /tmp/running-esc-12202017.cfg
```

## Backup Database

1. Set ESC to maintenance mode.

2. Log in to ESC VM and execute this command before you take the backup.

```
[admin@auto-test-vnfm1-esc-0 admin]# sudo bash
[root@auto-test-vnfm1-esc-0 admin]# cp /opt/cisco/esc/esc-scripts/esc_dbtool.py
/opt/cisco/esc/esc-scripts/esc_dbtool.py.bkup
[root@auto-test-vnfm1-esc-0 admin]# sudo sed -i "s,'pg_dump','usr/pgsql-9.4/bin/pg_dump,'"
/opt/cisco/esc/esc-scripts/esc_dbtool.py
```

#Set ESC to mainenance mode

```
[root@auto-test-vnfm1-esc-0 admin]# escadm op_mode set --mode=maintenance
```

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

```
[root@auto-test-vnfm1-esc-0 admin]# escadm op_mode show
```

4. Backup DB with the use of DB backup restore tool available in ESC.

```
[root@auto-test-vnfm1-esc-0 admin]# sudo /opt/cisco/esc/esc-scripts/esc_dbtool.py backup --file
scp://<username>:<password>@<backup_vm_ip>:<filename>
```

5. Set ESC back to Operation Mode and confirm the mode.

```
[root@auto-test-vnfm1-esc-0 admin]# escadm op_mode set --mode=operation
```

```
[root@auto-test-vnfm1-esc-0 admin]# escadm op_mode show
```

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

```
[root@auto-test-vnfm1-esc-0 admin]# /opt/cisco/esc/esc-scripts
```

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

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

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

```
[admin@auto-test-vnfm2-esc-1 ~]$ cd /etc/rsyslog.d
[admin@auto-test-vnfm2-esc-1 rsyslog.d]$ls /etc/rsyslog.d/00-escmanager.conf
00-escmanager.conf
```

```
[admin@auto-test-vnfm2-esc-1 rsyslog.d]$ls /etc/rsyslog.d/01-messages.conf
01-messages.conf
```

```
[admin@auto-test-vnfm2-esc-1 rsyslog.d]$ls /etc/rsyslog.d/02-mona.conf
02-mona.conf
```

```
[admin@auto-test-vnfm2-esc-1 rsyslog.d]$ls /etc/rsyslog.conf
rsyslog.conf
```

## EM Backup

1. After VNF/ESC is up, AutoVNF uses ESC in order to bring up the EM cluster. Once the EM cluster is up, EM will interact with ESC to bring up VNF (VPC/StarOS).

2. EM has 1:N redundancy in Ultra-M Solution. There is a cluster of three EM VMs and Ultra-M

supports the recovery of single VM failure.

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

EM backup details:

- Running configuration
- NCS DB
- EM Logs (from both NCS active/standby VMs)
- Syslog configuration

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

4. Take EM running configuration back up and transfer the file to the backup server.

```
ubuntu@vnfd1deploymentem-0:~$ sudo -i
root@vnfd1deploymentem-0:~# ncs_cli -u admin -C

admin connected from 127.0.0.1 using console on vnfd1deploymentem-0
admin@scm# show running-config | save em-running-12202017.cfg

root@vnfd1deploymentem-0:~# ll em-running-12202017.cfg
-rw-r--r-- 1 root root 19957 Dec 20 23:01 em-running-12202017.cfg
```

5. Take EM NCS DB backup and transfer the file to the backup server.

```
ubuntu@vnfd1deploymentem-0:~$ sudo -i
root@vnfd1deploymentem-0:~# cd /opt/cisco/em/git/em-scm/ncs-cdb
root@vnfd1deploymentem-0:/opt/cisco/em/git/em-scm/ncs-cdb# ll

total 472716
drwxrwxr-x 2 root root    4096 Dec 20 02:53 ./
drwxr-xr-x 9 root root    4096 Dec 20 19:22 ../
-rw-r--r-- 1 root root     770 Dec 20 02:48 aaa_users.xml
-rw-r--r-- 1 root root   70447 Dec 20 02:53 A.cdb
-rw-r--r-- 1 root root 483927031 Dec 20 02:48 C.cdb
-rw-rw-r-- 1 root root     47 Jul 27 05:53 .gitignore
-rw-rw-r-- 1 root root    332 Jul 27 05:53 global-settings.xml
-rw-rw-r-- 1 root root    621 Jul 27 05:53 jvm-defaults.xml
-rw-rw-r-- 1 root root   3392 Jul 27 05:53 nacm.xml
-rw-r--r-- 1 root root   6156 Dec 20 02:53 O.cdb
-rw-r--r-- 1 root root  13041 Dec 20 02:48 startup-vnfd.xml

root@vnfd1deploymentem-0:/opt/cisco/em/git/em-scm/ncs-cdb#
root@vnfd1deploymentem-0:/opt/cisco/em/git/em-scm# tar cvf em_cdb_backup.tar ncs-cdb

ncs-cdb/
ncs-cdb/O.cdb
ncs-cdb/C.cdb
ncs-cdb/nacm.xml
ncs-cdb/jvm-defaults.xml
ncs-cdb/A.cdb
ncs-cdb/aaa_users.xml
ncs-cdb/global-settings.xml
```

```
ncs-cdb/.gitignore
ncs-cdb/startup-vnfd.xml
```

```
root@vnfd1deploymentem-0:/opt/cisco/em/git/em-scm# ll em_cdb_backup.tar
-rw-r--r-- 1 root root 484034560 Dec 20 23:06 em_cdb_backup.tar
```

6. Navigate to the scripts directory, collect the logs and transfer it to the backup server.

```
/opt/cisco/em-scripts
```

```
sudo ./collect-em-logs.sh
```

- Repeat the same procedure to collect the log from NCS standby VM
- Backup the syslog configuration on all the EM VMs and transfer them to backup server

```
root@vnfd1deploymentem-0:/etc/rsyslog.d# pwd
/etc/rsyslog.d
```

```
root@vnfd1deploymentem-0:/etc/rsyslog.d# ll
total 28
```

```
drwxr-xr-x  2 root root 4096 Jun  7 18:38 ./
drwxr-xr-x 86 root root 4096 Jun  6 20:33 ../
-rw-r--r--  1 root root  319 Jun  7 18:36 00-vnmf-proxy.conf
-rw-r--r--  1 root root  317 Jun  7 18:38 01-ncs-java.conf
-rw-r--r--  1 root root  311 Mar 17  2012 20-ufw.conf
-rw-r--r--  1 root root  252 Nov 23  2015 21-cloudinit.conf
-rw-r--r--  1 root root 1655 Apr 18  2013 50-default.conf
```

```
root@vnfd1deploymentem-0:/etc/rsyslog.d# ls /etc/rsyslog.conf
rsyslog.conf
```

## StarOS Backup

For StarOS, this information needs to be backed up.

- Charging Data Records (CDRs) needs to be backed up at remote server at regular intervals
- Day-0 configuration applied at the time of the deployment of the system
- Day-N configuration
- SSDs
- Syslog configuration

## Restore Procedure

### OSPD Recovery

OSPD recovery procedure is performed based on these assumptions

- OSPD backup is available from old OSPD server
- OSPD Recovery will be done on the new server which is the replacement of the old OSPD server in the system.

# AutoDeploy Recovery

1. AutoDeploy VM is recoverable when the VM is in error or shutdown state, perform a hard reboot to bring up the impacted VM. Execute these checks in order to see if this helps recover AutoDeploy.

## Checking AutoDeploy Processes

Verify that key processes are running on the AutoDeploy VM:

```
root@auto-deploy-iso-2007-uas-0:~# initctl status autodeploy
autodeploy start/running, process 1771
```

```
root@auto-deploy-iso-2007-uas-0:~# ps -ef | grep java
root      1788  1771  0 May24 ?          00:00:41 /usr/bin/java -jar
/opt/cisco/usp/apps/autodeploy/autodeploy-1.0.jar com.cisco.usp.autodeploy.Application --
autodeploy.transaction-log-store=/var/log/cisco-uas/autodeploy/transactions
Stopping/Restarting AutoDeploy Processes
```

## #To start the AutoDeploy process:

```
root@auto-deploy-iso-2007-uas-0:~# initctl start autodeploy
autodeploy start/running, process 11094
```

## #To stop the AutoDeploy process:

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

## #To restart the AutoDeploy process:

```
root@auto-deploy-iso-2007-uas-0:~# initctl restart autodeploy
autodeploy start/running, process 11049
```

## #If the VM is in ERROR or shutdown state, hard-reboot the AutoDeploy VM

```
[stack@pod1-ospd ~]$ nova list |grep auto-deploy
| 9b55270a-2dcd-4ac1-aba3-bf041733a0c9 | auto-deploy-ISO-2007-uas-
0 | ACTIVE | - | running | mgmt=172.16.181.12,
10.84.123.39
```

```
[stack@pod1-ospd ~]$ nova reboot -hard 9b55270a-2dcd-4ac1-aba3-bf041733a0c9
```

2. If AutoDeploy is unrecoverable, follow these procedures in order to restore it to the state it was before in. Use the backup taken earlier.

```
[stack@pod1-ospd ~]$ nova list |grep auto-deploy
| 9b55270a-2dcd-4ac1-aba3-bf041733a0c9 | auto-deploy-ISO-2007-uas-
0 | ACTIVE | - | running | mgmt=172.16.181.12,
10.84.123.39 [stack@pod1-ospd ~]$ cd /opt/cisco/usp/uas-installer/scripts
```

```
[stack@pod1-ospd ~]$ ./auto-deploy-booting.sh --floating-ip 10.1.1.2 --delete
```

3. After AutoDeploy is deleted, create it again with the same floatingip address.

```
[stack@pod1-ospd ~]$ cd /opt/cisco/usp/uas-installer/scripts
```

```
[stack@pod1-ospd scripts]$ ./auto-deploy-booting.sh --floating-ip 10.1.1.2
```

```
2017-11-17 07:05:03,038 - INFO: Creating AutoDeploy deployment (1 instance(s)) on
'http://10.1.1.2:5000/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 '10.1.1.2' 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: 10.1.1.2]

```

4. Copy the **Autodeploy.cfg** file, ISO and the **confd\_backup tar** file from your backup server to the AutoDeploy VM.

5. Restore confd cdb files from the 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

```

#Restart AutoDeploy process

```

root@auto-deploy-iso-2007-uas-0~# service autodeploy restart
autodeploy start/running, process 2144

```

#Check that confd was loaded properly by checking earlier transactions.

```

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      ID      ID      TX ID
-----
1512571978613  service-deployment  tb5bxb      2017-12-06T14:52:59.412+00:00  deployment-success

```

6. If VM is successfully restored and running; ensure all the syslog specific configuration is restored from the previous successful known backup.

```

ubuntu@auto-deploy-vnf-iso-5-1-5-1196-uas-0:~$sudo su

```

```
root@auto-deploy-vnf-iso-5-1-5-1196-uas-0:/home/ubuntu#ls /etc/rsyslog.d/00-autodeploy.conf
00-autodeploy.conf
```

```
root@auto-deploy-vnf-iso-5-1-5-1196-uas-0:/home/ubuntu#ls /etc/rsyslog.conf
rsyslog.conf
```

## AutoIT-VNF Recovery

1. AutoIT-VNF VM is recoverable, if the VM is in error or shutdown state, perform a hard reboot in order to bring up the impacted VM. Execute these steps in order to recover AutoIT-VNF.

### Checking AutoIT-VNF Processes

Verify that key processes are running on the AutoIT-VNF VM:

```
root@auto-it-vnf-iso-5-1-5-1196-uas-0:~# service autoit status
AutoIT-VNF is running.
```

### #Stopping/Restarting AutoIT-VNF Processes

```
root@auto-it-vnf-iso-5-1-5-1196-uas-0:~# service autoit stop
AutoIT-VNF API server stopped.
```

### #To restart the AutoIT-VNF processes:

```
root@auto-it-vnf-iso-5-1-5-1196-uas-0:~# service autoit restart
AutoIT-VNF API server stopped.
Starting AutoIT-VNF
```

```
/opt/cisco/usp/apps/auto-it/vnf
AutoIT API server started.
```

### #If the VM is in ERROR or shutdown state, hard-reboot the AutoDeploy VM

```
[stack@pod1-ospd ~]$ nova list |grep auto-it
| 1c45270a-2dcd-4ac1-aba3-bf041733d1a1 | auto-it-vnf-ISO-2007-uas-
0 | ACTIVE | - | running | mgmt=172.16.181.13,
10.84.123.40
```

```
[stack@pod1-ospd ~]$ nova reboot -hard 1c45270a-2dcd-4ac1-aba3-bf041733d1a1
```

2. If AutoIT-VNF is unrecoverable, follow these procedures in order to restore it to the state it was before in. Use the backup file.

```
[stack@pod1-ospd ~]$ nova list |grep auto-it
| 580faf80-1d8c-463b-9354-781ea0c0b352 | auto-it-vnf-ISO-2007-uas-
0 | ACTIVE | - | running | mgmt=172.16.181.3,
10.84.123.42 [stack@pod1-ospd ~]$ cd /opt/cisco/usp/uas-installer/scripts
```

```
[stack@pod1-ospd ~]$ ./ auto-it-vnf-staging.sh --floating-ip 10.1.1.3 --delete
```

3. Recreate Auto-IT by running **auto-it-vnf staging** script and ensure to use the same floating ip that was used earlier.

```
[stack@pod1-ospd ~]$ cd /opt/cisco/usp/uas-installer/scripts
```

```
[stack@pod1-ospd scripts]$ ./auto-it-vnf-staging.sh --floating-ip 10.1.1.3
```

```
2017-11-16 12:54:31,381 - INFO: Creating StagingServer deployment (1 instance(s)) on
'http://10.1.1.3:5000/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 '10.1.1.3' 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: 10.1.1.3

```

#### 4. ISO image(s) used in the POD needs to be reloaded on AutoIT-VNF.

```
[stack@pod1-ospd ~]$ cd images/5_1_7-2007/isos
```

```
[stack@pod1-ospd isos]$ curl -F file=@usp-5_1_7-2007.iso http://10.1.1.3:5001/isos
```

```
{
  "iso-id": "5.1.7-2007"
}
```

Note: 10.1.1.3 is AutoIT-VNF IP in the above command.

#Validate that ISO is correctly loaded.

```
[stack@pod1-ospd isos]$ curl http://10.1.1.3:5001/isos
```

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

#### 5. Copy the VNF **system.cfg** files from the remote server to the AutoIT-VNF VM. In this example, it is copied from AutoDeploy to AutoIT-VNF VM.

```
[stack@pod1-ospd autodeploy]$ scp system-vnf* ubuntu@10.1.1.3:.
ubuntu@10.1.1.3'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
```

#### 6. Copy the files in an appropriate location on AutoIT-VNF as referenced in AutoDeploy configuration. See here;

```
ubuntu@auto-it-vnf-iso-2007-uas-0:~$ sudo -i
root@auto-it-vnf-iso-2007-uas-0:~$ cp -rp system-vnf1.cfg system-vnf2.cfg
/opt/cisco/usp/uploads/
root@auto-it-vnf-iso-2007-uas-0:~$ls /opt/cisco/usp/uploads/
system-vnf1.cfg  system-vnf2.cfg
```

#### 7. If VM is successfully restored and running, ensure all the syslog specific configuration is



restored from the previous successful known backup.

```
root@auto-deploy-vnf-iso-5-1-5-1196-uas-0:/home/ubuntu#ls /etc/rsyslog.d/00-autoit-vnf.conf
00-autoit-vnf.conf
```

```
root@auto-deploy-vnf-iso-5-1-5-1196-uas-0:ls /etc/rsyslog.conf
rsyslog.conf
```

## AutoVNF Recovery

1. AutoVNF VM is recoverable if the VM is in error or shutdown state. Perform a hard reboot in order to bring up the impacted VM. Execute these steps in order to recover AutoVNF.

2. Identify the VM which is in Error or Shutdown state. Hard-reboot the AutoVNF VM.

In this example, reboot **auto-testautovnf1-uas-2**.

```
[root@tb1-baremetal scripts]# nova list | grep "auto-testautovnf1-uas-[0-2]"

| 3834a3e4-96c5-49de-a067-68b3846fba6b | auto-testautovnf1-uas-
0 | ACTIVE | - | running | auto-
testautovnf1-uas-orchestration=172.57.12.6; auto-testautovnf1-uas-
management=172.57.11.8
|
| 0fbfec0c-f4b0-4551-807b-50c5fe9d3ea7 | auto-testautovnf1-uas-
1 | ACTIVE | - | running | auto-
testautovnf1-uas-orchestration=172.57.12.7; auto-testautovnf1-uas-management=172.57.11.12
|
| 432e1a57-00e9-4e58-8bef-2a20652df5bf | auto-testautovnf1-uas-
2 | ACTIVE | - | running | auto-
testautovnf1-uas-orchestration=172.57.12.13; auto-testautovnf1-uas-
management=172.57.11.4
|
```

```
[root@tb1-baremetal scripts]# nova reboot --hard 432e1a57-00e9-4e58-8bef-2a20652df5bf
Request to reboot server <Server: auto-testautovnf1-uas-2> has been accepted.
```

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

3. Once the VM comes up, validate that it joins the cluster back.

```
root@auto-testautovnf1-uas-1:/opt/cisco/usp/uas/scripts# confd_cli -u admin -C
Welcome to the ConfD CLI
admin connected from 127.0.0.1 using console on auto-testautovnf1-uas-1
```

```
auto-testautovnf1-uas-1#show uas
```

```
uas version 1.0.1-1
uas state ha-active
uas ha-vip 172.57.11.101
```

```
INSTANCE IP STATE ROLE
-----
172.57.12.6 alive CONFD-SLAVE
172.57.12.7 alive CONFD-MASTER
172.57.12.13 alive NA
```

4. If AutoVNF VM cannot be recovered by the mentioned procedure, you need to recover it with

the help of these steps.

```
[stack@pod1-ospd ~]$ nova list | grep vnf1-UAS-uas-0
| 307a704c-a17c-4cdc-8e7a-3d6e7e4332fa | vnf1-UAS-uas-0
| ACTIVE | - | running | vnf1-UAS-uas-orchestration=172.168.11.10; vnf1-UAS-uas-management=172.168.10.3
```

```
[stack@pod1-ospd ~]$ nova delete vnf1-UAS-uas-0
Request to delete server vnf1-UAS-uas-0 has been accepted.
```

5. In order to recover the **autovnf-uas** VM, execute **uas-check** script to check state. It must report an Error. Then execute again with **--fix** option in order to recreate the missing UAS VM.

```
[stack@pod1-ospd ~]$ cd /opt/cisco/usp/uas-installer/scripts/
[stack@pod1-ospd scripts]$ ./uas-check.py auto-vnf vnf1-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@tb3-ospd scripts]$ ./uas-check.py auto-vnf vnf1-UAS --fix
```

```
2017-11-22 14:01:07,215 - INFO: Check of AutoVNF cluster started
2017-11-22 14:01:09,575 - INFO: Instance vnf1-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 vnf1-UAS-uas-0'
2017-11-22 14:01:13,568 - INFO: Removed instance vnf1-UAS-uas-0'
2017-11-22 14:01:13,568 - INFO: Creating instance vnf1-UAS-uas-0' and attaching volume 'vnf1-UAS-uas-vol-0'
2017-11-22 14:01:49,525 - INFO: Created instance 'vnf1-UAS-uas-0'
```

```
[stack@tb3-ospd scripts]$ ./uas-check.py auto-vnf vnf1-UAS
```

```
2017-11-16 13:11:07,472 - INFO: Check of AutoVNF cluster started
2017-11-16 13:11:09,510 - INFO: Found 3 ACTIVE AutoVNF instances
2017-11-16 13:11:09,511 - INFO: Check completed, AutoVNF cluster is fine
```

6. Log in to master AutoVNF VM. Within a few minutes after the recovery, newly created instance must join the cluster and in alive state.

```
tb3-bxb-vnf1-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 CONFD-SLAVE
172.17.180.7 alive CONFD-MASTER
172.17.180.9 alive NA
```

#if uas-check.py --fix fails, you may need to copy this file and execute again.

```
[stack@tb3-ospd]$ mkdir -p /opt/cisco/usp/apps/auto-it/common/uas-deploy/
[stack@tb3-ospd]$ cp /opt/cisco/usp/uas-installer/common/uas-deploy/userdata-uas.txt
/opt/cisco/usp/apps/auto-it/common/uas-deploy/
```

7. If the VM is successfully restored and running, ensure all the syslog specific configuration is restored from the previous successful known backup. Ensure it is restored in all the AutoVNF VMs.

```
ubuntu@auto-testautovnf1-uas-1:~$sudo su
root@auto-testautovnf1-uas-1:/home/ubuntu#ls /etc/rsyslog.d/00-autovnf.conf
00-autovnf.conf
```

```
root@auto-testautovnf1-uas-1:/home/ubuntu#ls /etc/rsyslog.conf
rsyslog.conf
```

## ESC Recovery

1. ESC VM is recoverable if the VM is in the Error or shutdown state. Perform a hard reboot in order to bring up the impacted VM. Execute these steps to recover ESC.

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

```
[root@tbl-baremetal scripts]# nova list | grep auto-test-vnfm1-ESC-
| f03e3cac-a78a-439f-952b-045aea5b0d2c | auto-test-vnfm1-ESC-
0 | ACTIVE | - | running | auto-testautovnf1-
uas-orchestration=172.57.12.11; auto-testautovnf1-uas-
management=172.57.11.3
|
| 79498e0d-0569-4854-a902-012276740bce | auto-test-vnfm1-ESC-
1 | ACTIVE | - | running | auto-testautovnf1-
uas-orchestration=172.57.12.15; auto-testautovnf1-uas-
management=172.57.11.5
|
```

```
[root@tbl-baremetal scripts]# [root@tbl-baremetal scripts]# nova reboot --hard f03e3cac-a78a-
439f-952b-045aea5b0d2c\
Request to reboot server <Server: auto-test-vnfm1-ESC-0> has been accepted.
```

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

3. If ESC VM is deleted and needs to be brought up again, follow this sequence of steps.

```
[stack@pod1-ospd scripts]$ nova list |grep ESC-1
| c566efbf-1274-4588-a2d8-0682e17b0d41 | vnfm1-ESC-ESC-
1 | ACTIVE | - | running | vnfm1-
UAS-uas-orchestration=172.168.11.14; vnfm1-UAS-uas-
management=172.168.10.4
|
```

```
[stack@pod1-ospd scripts]$ nova delete vnfm1-ESC-ESC-1
Request to delete server vnfm1-ESC-ESC-1 has been
accepted.
```

4. 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@vnfm1-uas-uas-0:~$ sudo -i
root@vnfm1-uas-uas-0:~# confd_cli -u admin -C
Welcome to the ConfD CLI
```

```
admin connected from 127.0.0.1 using console on vnfm1-uas-uas-0
vnfm1-uas-uas-0#show transaction
```

TX ID

TX TYPE

DEPLOYMENT ID

TIMESTAMP	STATUS
35eefc4a-d4a9-11e7-bb72-fa163ef8df2b 29T02:01:27.750692-00:00	vnf-deployment vnf1-DEPLOYMENT 2017-11-29 deployment-success
73d9c540-d4a8-11e7-bb72-fa163ef8df2b 29T01:56:02.133663-00:00	vnfm-deployment vnf1-ESC 2017-11-29 deployment-success

vnf1-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 - VNFM Deployment RPC triggered for deployment: vnf1-ESC,
deactivate: 0

2017-11-29 01:56:02,179 - Notify deployment
..
2017-11-29 01:57:30,385 - Creating VNFM 'vnf1-ESC-ESC-1' with [python //opt/cisco/vnf-
staging/bootvm.py vnf1-ESC-ESC-1 --flavor vnf1-ESC-ESC-flavor --image 3fe6b197-961b-4651-af22-
dfd910436689 --net vnf1-UAS-uas-management --gateway_ip 172.168.10.1 --net vnf1-UAS-uas-
orchestration --os_auth_url http://10.1.1.5:5000/v2.0 --os_tenant_name core --os_username *****
--os_password ***** --bs_os_auth_url http://10.1.1.5:5000/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_confd_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]...
```

5. 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/Cisco@123). You need to remove the **-encrypt\_key** option as well. For **user\_pass** and **user\_confd\_pass**, you need to use the format **-user\_passwd username:password** (example - admin:Cisco@123).

Now, find the URL to **bootvm.py** from running-config and get the **bootvm.py** file to the autovnf-uas VM. 10.1.1.3 is the Auto-IT in this case.

```
root@vnf1-uas-uas-0:~# confd_cli -u admin -C
Welcome to the ConfD CLI
admin connected from 127.0.0.1 using console on vnf1-uas-uas-0

vnf1-uas-uas-0#show running-config autovnf-vnfm:vnfm
...
configs bootvm
  value http://10.1.1.3:80/bundles/5.1.7-2007/vnfm-bundle/bootvm-2\_3\_2\_155.py
!

root@vnf1-uas-uas-0:~# wget http://10.1.1.3:80/bundles/5.1.7-2007/vnfm-bundle/bootvm-
2_3_2_155.py

--2017-12-01 20:25:52-- http://10.1.1.3/bundles/5.1.7-2007/vnfm-bundle/bootvm-2\_3\_2\_155.py
Connecting to 10.1.1.3:80... 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@vnf1-uas-uas-0:~# echo "openstack.endpoint=publicURL" > /tmp/esc_params.cfg
```

### 6. Execute the shell script that executes **bootvm.py** python script with options.

```
root@vnf1-uas-uas-0:~# /bin/sh esc.sh
+ python ./bootvm.py vnf1-ESC-ESC-1 --flavor vnf1-ESC-ESC-flavor --image 3fe6b197-961b-4651-af22-dfd910436689 --net vnf1-UAS-uas-management --gateway_ip 172.168.10.1 --net vnf1-UAS-uas-orchestration --os_auth_url http://10.1.1.5:5000/v2.0 --os_tenant_name core --os_username core --os_password Cisco@123 --bs_os_auth_url http://10.1.1.5:5000/v2.0 --bs_os_tenant_name core --bs_os_username core --bs_os_password Cisco@123 --esc_ui_startup false --esc_params_file /tmp/esc_params.cfg --user_pass admin:Cisco@123 --user_confd_pass admin:Cisco@123 --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
```

```
+-----+-----+
| Property                               |                               |
| Value                                  |                               |
|                                         |                               |
|                                         |                               |
+-----+-----+
| OS-DCF:diskConfig                       |                               |
| MANUAL                                  |                               |
|                                         |                               |
| OS-EXT-AZ:availability_zone             |                               |
| mgmt                                    |                               |
|                                         |                               |
| OS-EXT-SRV-ATTR:host                    | tb5-ultram-osd-compute-     |
| 1.localdomain                          |                               |
|                                         |                               |
| OS-EXT-SRV-ATTR:hypervisor_hostname    | tb5-ultram-osd-compute-     |
| 1.localdomain                          |                               |
|                                         |                               |
| OS-EXT-SRV-ATTR:instance_name          | instance-                   |
| 000001eb                                |                               |
|                                         |                               |
| OS-EXT-STS:power_state                  |                               |
|                                         |                               |
+-----+-----+
```

OS-EXT-STS:task_state	-
OS-EXT-STS:vm_state	active
OS-SRV-USG:launched_at	2017-12-02T13:28:32.000000
OS-SRV-USG:terminated_at	-
accessIPv4	
accessIPv6	
addresses	{ "vnf1-UAS-uas-orchestration": [{ "OS-EXT-IPS-MAC:mac_addr": "fa:16:3e:d7:c6:19", "version": 4, "addr": "172.168.11.14", "OS-EXT-IPS:type": "fixed" }], "vnf1-UAS-uas-management": [{ "OS-EXT-IPS-MAC:mac_addr": "fa:16:3e:31:ee:cd", "version": 4, "addr": "172.168.10.6", "OS-EXT-IPS:type": "fixed" }] }
config_drive	True
created	2017-12-02T13:27:49Z
flavor	{ "id": "457623b6-05d5-403c-b2e4-aa3b6a0c9d32", "links": [{ "href": " <a href="http://10.1.1.5:8774/flavors/457623b6-05d5-403c-b2e4-aa3b6a0c9d32">http://10.1.1.5:8774/flavors/457623b6-05d5-403c-b2e4-aa3b6a0c9d32</a> ", "rel": "bookmark" }] }
hostId	f5d2bbf0c5a7df34cf2e6f62ae0702ef120ff82f81c3f7664ffb35e9
id	2601b8ec-8ff8-4285-810a-e859f6642ab6
image	{ "id": "3fe6b197-961b-4651-af22-dfd910436689", "links": [{ "href": " <a href="http://10.1.1.5:8774/images/3fe6b197-961b-4651-af22-dfd910436689">http://10.1.1.5:8774/images/3fe6b197-961b-4651-af22-dfd910436689</a> ", "rel": "bookmark" }] }
key_name	-

```

| metadata          |
| {}                |
|
| name              | vnf1-esc-esc-
1
|
| os-extended-volumes:volumes_attached | []
|
| progress          |
0
|
| security_groups   | [{"name": "default"}, {"name":
"default"}]
|
| status            |
ACTIVE
|
| tenant_id         |
fd4b15df46c6469cbacf5b80dcc98a5c
|
| updated           | 2017-12-
02T13:28:32Z
|
| user_id           |
d3b51d6f705f4826b22817f27505c6cd

```

## 7. From OSPD, check that the new ESC VM is ACTIVE/running.

```

[stack@pod1-ospd ~]$ nova list|grep -i esc
| 934519a4-d634-40c0-a51e-fc8d55ec7144 | vnf1-ESC-ESC-
0 | ACTIVE | - | running | vnf1-
UAS-uas-orchestration=172.168.11.13; vnf1-UAS-uas-
management=172.168.10.3
|
| 2601b8ec-8ff8-4285-810a-e859f6642ab6 | vnf1-ESC-ESC-
1 | ACTIVE | - | running | vnf1-
UAS-uas-orchestration=172.168.11.14; vnf1-UAS-uas-
management=172.168.10.6

```

#Log in to new ESC and verify Backup state. You may execute health.sh on ESC Master too.

```
ubuntu@vnf1-uas-uas-0:~$ ssh admin@172.168.11.14
```

```
...  
#####  
#   ESC on vnf1-esc-esc-1.novalocal is in BACKUP state.  
#####
```

```
[admin@vnf1-esc-esc-1 ~]$ escadm status  
0 ESC status=0 ESC Backup Healthy
```

```
[admin@vnf1-esc-esc-1 ~]$ health.sh  
===== ESC HA (BACKUP) =====  
=====
```

```
ESC HEALTH PASSED
```

```
[admin@vnf1-esc-esc-1 ~]$ cat /proc/drbd  
version: 8.4.7-1 (api:1/proto:86-101)  
GIT-hash: 3a6a769340ef93b1ba2792c6461250790795db49 build by mockbuild@Build64R6, 2016-01-12  
13:27:11
```

```
1: cs:Connected ro:Secondary/Primary ds:UpToDate/UpToDate C r-----  
ns:0 nr:504720 dw:3650316 dr:0 al:8 bm:0 lo:0 pe:0 ua:0 ap:0 ep:1 wo:f oos:0
```

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

9. For ESC database restore, ensure that the ESC service is stopped before restoring the database; For ESC HA, execute in secondary VM first and then the primary VM.

```
# service keepalived stop
```

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

```
# escadm status
```

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

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

12. Restart ESC service to complete the database restore.

13. For HA execute in both VMs, restart the keepalived service.

```
# service keepalived start
```

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

```
[admin@auto-test-vnfm2-esc-1 ~]$  
[admin@auto-test-vnfm2-esc-1 ~]$ cd /etc/rsyslog.d  
[admin@auto-test-vnfm2-esc-1 rsyslog.d]$ls /etc/rsyslog.d/00-escmanager.conf  
00-escmanager.conf
```

```
[admin@auto-test-vnfm2-esc-1 rsyslog.d]$ls /etc/rsyslog.d/01-messages.conf  
01-messages.conf
```



```
[admin@auto-test-vnfm2-esc-1 rsyslog.d]$ls /etc/rsyslog.d/02-mona.conf
02-mona.conf
```

```
[admin@auto-test-vnfm2-esc-1 rsyslog.d]$ls /etc/rsyslog.conf
rsyslog.conf
```

## EM Recovery

1. If EM VM is in None/Error state due to one or the other conditions, the user can follow the given sequence in order to recover the impacted EM VM.

2. ESC/VNFM is the component which monitors the EM VMs so in the case where EM is in the Error state, ESC will try to auto-recover the EM VM. For any reason if ESC is unable to complete the recovery successfully, ESC will mark that VM in error state.

3. In such scenarios, the user can do manual recovery of the EM VM once the underlying infrastructure issue is fixed. It is important to execute this manual recovery only after an underlying problem is fixed.

4. Identify the VM in the errored state.

```
[stack@pod1-ospd ~]$ source corerc
[stack@pod1-ospd ~]$ nova list --field name,host,status |grep -i err
| c794207b-a51e-455e-9a53-3b8ff3520bb9 | vnfm1-DEPLOYMENT-_vnfm1-D_0_a6843886-77b4-4f38-b941-
74eb527113a8 | None | ERROR |
```

5. Log in to ESC Master, execute **recovery-vm-action** for each impacted EM and CF VM. Be patient. ESC will schedule the recovery-action and it might not happen for a few minutes.

```
ubuntu@vnfm1-uas-uas-1:~$ ssh admin@172.168.10.3
```

...

```
[admin@vnfm1-esc-esc-0 ~]$ sudo /opt/cisco/esc/esc-confd/esc-cli/esc_nc_cli recovery-vm-action DO
vnfm1-DEPLOYMENT-_vnfm1-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.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>
```

6. Monitor the **/var/log/esc/yangesc.log** until the command completes.

```
[admin@vnfm1-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 [vnfm1-DEPLOYMENT-
_vnfm1-D_0_a6843886-77b4-4f38-b941-74eb527113a8]
```

```
#Log in to new EM and verify EM state is up.
```

```
ubuntu@vnflvnfddeploymentem-1:~$ /opt/cisco/ncs/current/bin/ncs_cli -u admin -C
```

```
admin connected from 172.17.180.6 using ssh on vnflvnfddeploymentem-1
```

```
admin@scm# show ems
```

```
EM                VNFM
```

```
ID  SLA  SCM  PROXY
```

```
-----
```

```
2   up   up   up
```

```
3   up   up   up
```

## When ESC Fails to Start VM

1. In some cases, ESC will fail to start the VM due to an unexpected state. A workaround is to perform an ESC switchover by rebooting the Master ESC. The ESC switchover takes about a minute. Execute **health.sh** on the new Master ESC in order to verify if it is up. When the ESC becomes Master, ESC might fix the VM state and start the VM. Since this operation is scheduled, you must wait 5-7 minutes for it to complete.

2. You can monitor **/var/log/esc/yangesc.log** and **/var/log/esc/escmanager.log**. If you see that the VM does not get recovered after 5-7 minutes, the user needs to go and do the manual recovery of the impacted VM(s).

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

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

```
root@abautotestvnfmlem-0:/etc/rsyslog.d# ll
```

```
total 28
```

```
drwxr-xr-x  2 root root 4096 Jun  7 18:38 ./
```

```
drwxr-xr-x 86 root root 4096 Jun  6 20:33 ../]
```

```
-rw-r--r--  1 root root  319 Jun  7 18:36 00-vnmf-proxy.conf
```

```
-rw-r--r--  1 root root  317 Jun  7 18:38 01-ncs-java.conf
```

```
-rw-r--r--  1 root root   311 Mar 17  2012 20-ufw.conf
```

```
-rw-r--r--  1 root root   252 Nov 23  2015 21-cloudinit.conf
```

```
-rw-r--r--  1 root root  1655 Apr 18  2013 50-default.conf
```

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

## StarOS Recovery

1. In the event where one of the StarOS VM appears in None/Error state due to one or the other conditions, the user can follow this sequence in order to recover the impacted StarOS VM.

2. ESC/VNFM is the component which monitors the StarOS VMs so in the case where CF/SF VM is in error state, ESC will try to auto-recover the CF/SF VM. For any reason, if ESC is unable to complete the recovery successfully, ESC will mark that VM in Error state.

3. In such scenarios, the user can do manual recovery of the CF/SF VM once the underlying infrastructure issue is fixed. It is important to execute this manual recovery only after you fix an underlying problem.

4. Identify the VM in the Error state.

```
[stack@pod1-ospd ~]$ source corerc
```

```
[stack@pod1-ospd ~]$ nova list --field name,host,status |grep -i err
| c794207b-a51e-455e-9a53-3b8ff3520bb9 | vnf1-DEPLOYMENT-_s4_0_c2b19084-26b3-4c9c-8639-62428a4cb3a3 | None | ERROR |
```

5. Log in to ESC Master, execute **recovery-vm-action** for each impacted EM and CF VM. Be patient. ESC will schedule the recovery-action and it might not happen for a few minutes.

```
ubuntu@vnf1-uas-uas-1:~$ ssh admin@172.168.10.3
```

...

```
[admin@vnf1-esc-esc-0 ~]$ sudo /opt/cisco/esc/esc-confd/esc-cli/esc_nc_cli recovery-vm-action DO
vnf1-DEPLOYMENT-_s4_0_c2b19084-26b3-4c9c-8639-62428a4cb3a3
[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>
```

##Monitor the /var/log/esc/yangesc.log until command completes.

```
[admin@vnf1-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 [vnf1-DEPLOYMENT-_s4_0_c2b19084-26b3-4c9c-8639-62428a4cb3a3]
```

6. Also, validate the same by running the **show card tab** on StarOS. If the recovered VM is SF, the user might need to make it active if it is desired. Make necessary StarOS configuration changes.

```
[local]VNF1# show card tab
```

Saturday December 02 14:40:20 UTC 2017

Slot	Card Type	Oper State	SPOF	Attach
1: CFC	Control Function Virtual Card	Active	No	
<b>2: CFC</b>	<b>Control Function Virtual Card</b>	<b>Standby</b>	-	
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	Standby	-	

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	Active	No

## When ESC Fails to Start VM

In some cases, ESC will fail to start the VM due to an unexpected state. A workaround is to perform an ESC switchover by rebooting the Master ESC. The ESC switchover takes about a minute. Execute **health.sh** on the new Master ESC in order to verify it is up. When the ESC becomes Master, ESC might fix the VM state and start the VM. Since this operation is scheduled, you must wait 5-7 minutes for it to complete. You can monitor **/var/log/esc/yangesc.log** and **/var/log/esc/escmanager.log**. If you do not see that VM gets recovered after 5-7 minutes, you will need to go and do the manual recovery of the impacted VM(s).