Deploying VNFs Using AutoVNF

This chapter describes the following topics:

Introduction

USP-based VNFs can be deployed using a stand-alone AutoVNF instance in environments with a pre-installed network function virtualization orchestrator (NFVO). In this scenario, a single AutoVNF VM is deployed on the VIM and communicates with a pre-existing VNFM installation to deploy the VNF(s). The VNFM can be installed in a tenant other than the one in which AutoVNF is installed.

Important Cisco Elastic Services Controller (ESC) is the only VNFM supported in this release.

A single AutoVNF VM can deploy one or more VNFs in one or more tenants within the same VIM.

VNF Deployment Automation Overview

Figure 1: AutoVNF Deployment Automation Workflow for a Single VNF, on page 2 and Figure 2: AutoVNF Deployment Automation Workflow for a Multi-VNF, on page 3 provide an overview of the VNF deployment automation process for when using a stand-alone AutoVNF instance. Details are provided in Table 1: VNF Deployment Automation Workflow Descriptions, on page 3.

NOTES:

- The workflow described in this section is supported only with VNF deployments performed through AutoVNF and that are based on OSP 10 or OSP 13.
- This information assumes that you have deployed the NFVI, VIM, and VNFM.
- This information assumes that all artifacts required during configuration must be pre-created in OpenStack.

Figure 1: AutoVNF Deployment Automation Workflow for a Single VNF

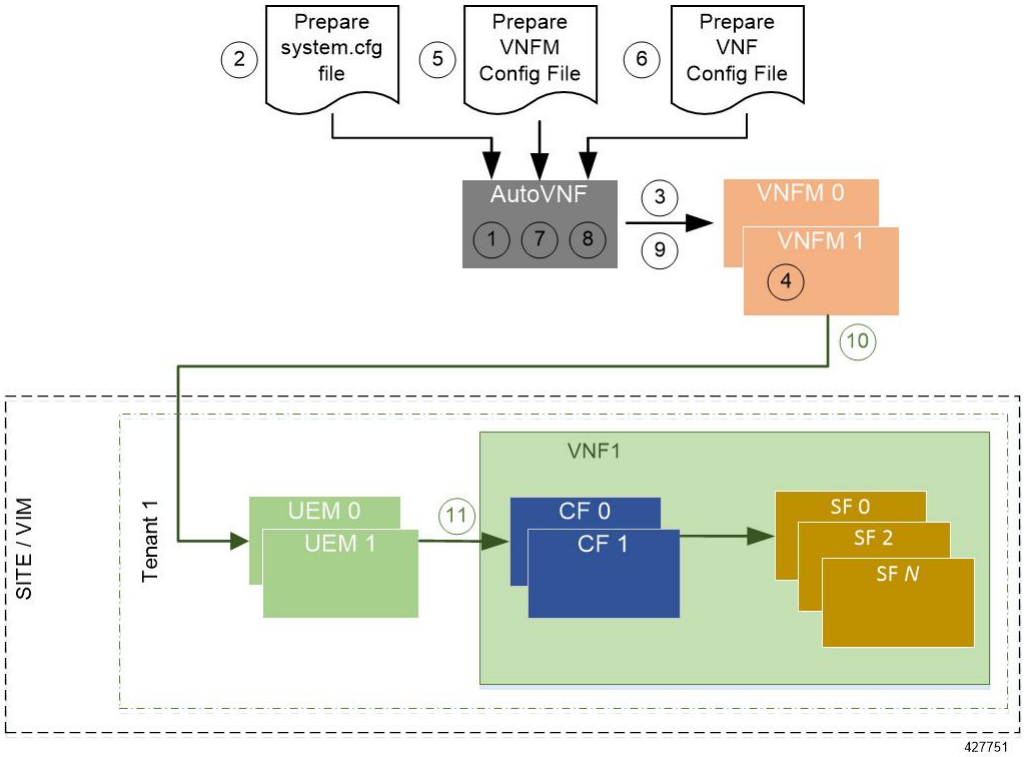

Figure 2: AutoVNF Deployment Automation Workflow for a Multi-VNF

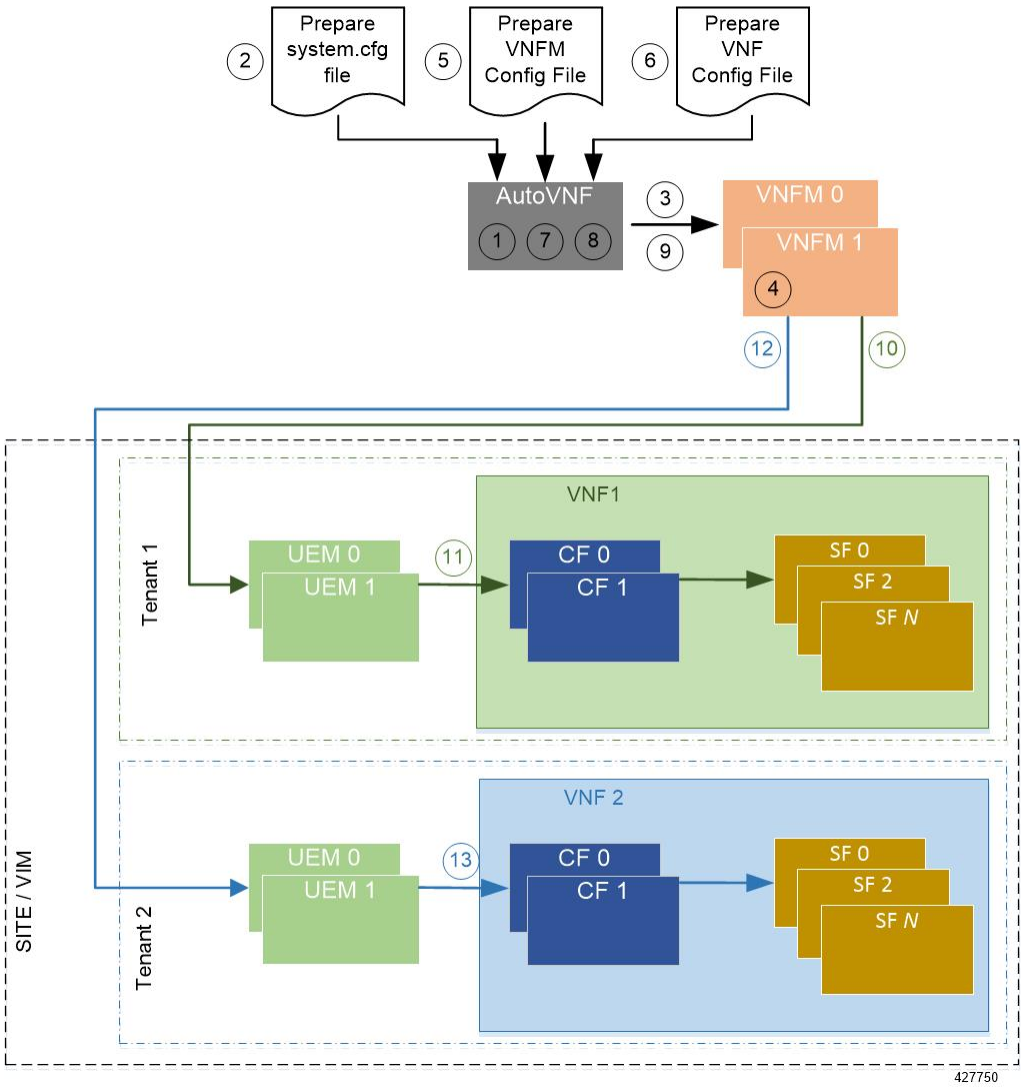

Table 1: VNF Deployment Automation Workflow Descriptions

Callout	Description
1	On the Onboarding Server, deploy AutoVNF using the *boot_uas.py* script provided as part of the release ISO. Refer to Deploy the AutoVNF VM, on page 10 for more information. The release ISO also includes the software images for the VNFM and VNFCs.
2	Prepare the system.cfg file to the AutoVNF VM. This file provides the VNF's Day-0 configuration. Refer to Sample system.cfg File for an example configuration file.

Callout	Description
3	Copy the VNFM scripts supplied in the UAS ISO from the AutoVNF VM to the VNFM VMs.
4	Confirm that the VNFM has been configured with the VIM connectors for the VNF tenants. A connector is needed for each tenant. Refer to Sample ESC VIM Connector Configuration for an example configuration.
5	Prepare the VNFM configuration file that provides AutoVNF with the necessary information for communicating with a pre-existing VNFM installation. Refer to Sample AutoVNF VNFM Configuration File for more information.
6	Prepare the VNF configuration file that is used by AutoVNF to initiate the VNF deployment process. This file includes all of the configuration information required to deploy all of the VNF components (VNFCs) such as secure tokens, network catalogs, VDU catalogs, and VDUs. Refer to Sample AutoVNF VNF Configuration File for more information.
7	On the AutoVNF VM, load and commit the VNFM configuration file prepared in the previous step. Once commited, activate the loaded AutoVNF VNFM configuration file. AutoVNF processes this data to deploy the VNFCs. Refer to Activate the AutoVNF Configuration Files, on page 13 for more information.
8	On the AutoVNF VM, load and commit the VNF configuration file prepared in the previous step. Once committed, activate the loaded AutoVNF VNF configuration file. AutoVNF processes this data to deploy the VNFCs. Refer to Activate the AutoVNF Configuration Files, on page 13 for more information.
9	AutoVNF passes the VNF configuration to the pre-created VNFM VM instance. It ensures that the various VM catalogs pertaining to other VNFCs are on-boarded by the VNFM. It accomplishes this through a number of YANG-based definitions which are then used to configure various aspects of the virtualized environment using REST and NETCONF APIs. That VNFM mounts the VNFC catalogs and works with AutoVNF to deploy the various components that comprise the desired VNF use-case (e.g. UGP or USF).
10, 12	The VNFM leverages the VNFC information to deploy the UEM VMs cluster. Though the USP architecture represents a single VNF to other network elements, it is comprised of multiple VM types each having their own separate catalogs. The UEM component of the USP works with the VNFM to deploy these catalogs based on the intended VNF use case (e.g. UGP, USF, etc.).

Callout	Description
11, 13	The UEM processes the Day-0 configuration information it received from the VNFM and deploys the Control Function (CF) and Service Function (SF) VNFC VMs. Once all of the VNF components (VNFCs) have been successfully deployed, AutoVNF notifies AutoDeploy. **Important** In multi-VNF deployments, AutoVNF waits until it receives confirmation that all of the VNFCs have been on-boarded successfully for the current VNF deployment before it initiates the deployment of the next VNF.

Pre-VNF Installation Verification

Prior to installing the USP, please ensure that the following is true:

- The prerequisite hardware is installed and operational with network connectivity.
- The prerequisite software is installed and configured and functioning properly:
  - You have administrative rights to the operating system.
  - VIM Orchestrator is properly installed and operational.
  - VIM components are properly installed and operational. This configuration includes networks, flavors, and sufficient quota allocations to the tenant.

    Note Supported and/or required flavors and quota allocations are based on deployment models. Contact your Cisco representative for more information.

  - You have administrative rights to the OpenStack setup.
  - The VNFM software is in properly installed and operational.

    Note Cisco's Elastic Services Controller (ESC) is the only VNFM supported in this release.

- The Cisco USP software ISO has been downloaded and is accessible by you.

Deploy the USP-based VNF

The AutoVNF software roles within the Ultra Automation Services (UAS) is used to automate the USP-based VNF deployment. The automated deployment process through AutoVNF is described in VNF Deployment Automation Overview, on page 1.

To deploy the USP-based VNF using AutoDeploy:

1. Onboard the USP ISO, on page 6.
2. Extract the UAS Bundle, on page 7.
3. Extract the UEM VM Image, on page 8.
4. Extract the UGP VM Image, on page 9.
5. Upload the USP VM Images to Glance, on page 10.
6. Deploy the AutoVNF VM, on page 10.
7. Activate the AutoVNF Configuration Files, on page 13.

Onboard the USP ISO

The files required to deploy the USP components are distributed as RPMs (called "bundles") in a single ISO package. They are maintained using YUM on the Onboarding Server. The following bundles are part of the ISO:

USP Bundle Name	Description
usp-em-bundle	The Element Manager (EM) Bundle RPM containing images and metadata for the Ultra Element Manager (UEM) module.
usp-uas-bundle	The Ultra Automation Services Bundle RPM containing AutoIT, AutoDeploy, AutoVNF, Ultra Web Services (UWS), and other automation packages.
usp-ugp-bundle	The Ultra Gateway Platform (UGP) Bundle RPM containing images for Ultra Packet core (VPC-DI). This bundle contains non-trusted images.
usp-vnfm-bundle	The VNFM Bundle RPM containing an image and a boot-up script for ESC (Elastic Service Controller).
usp-yang-bundle	The Yang Bundle RPM containing YANG data models including the VNFD and VNFR.
usp-auto-it-bundle	The bundle containing the AutoIT packages required to deploy the UAS.
ultram-manager	This package contains the script and relevant files needed to deploy the Ultra Health Service.

Important Release 6.4 will not be backward compatible with previous releases, i.e., you cannot deploy a 6.4 ISO from an AutoDeploy/AutoIT/AutoVNF running a pre-6.4 release, and vice-versa.

In addition to the bundles, the ISO bundle also includes scripts used to deploy the bundles including UAS.

Before proceeding with these instructions, ensure that the perquisites identified in USP Installation Prerequisites have been met.

To onboard the ISO package:

1. Log on to the Onboarding Server.
2. Download the USP ISO bundle and related files pertaining to the release.
3. Create a mount point on the Onboarding Server and mount the ISO package:

   mkdir /var/usp-iso
4. Mount the USP ISO.

   sudo mount -t iso9660 -o loop *<ISO_download_directory>/<ISO_package_name>* **/var/usp-iso**

   Example: The following command mounts the ISO bundle called *usp-5_5_0-1255.iso* located in a directory called *5_5_0-1283* to */var/usp-iso*:

   sudo mount -t iso9660 -o loop 5_5_0-1064/usp-5_5_0-1064.iso /var/usp-iso

   ```
   mount: /dev/loop1 is write-protected, mounting read-only
   ```
5. Verify the mount configuration.

   df –h

   Example output:

   ```
   Filesystem      Size  Used Avail Use% Mounted on
   /dev/sda2       187G  178G  316M 100% /
   devtmpfs         63G     0   63G   0% /dev
   tmpfs            63G  4.0K   63G   1% /dev/shm
   tmpfs            63G  1.4M   63G   1% /run
   tmpfs            63G     0   63G   0% /sys/fs/cgroup
   /dev/sda1       477M  112M  336M  25% /boot
   tmpfs            13G     0   13G   0% /run/user/0
   /dev/loop1      4.2G  4.2G     0 100% /var/usp-iso
   ```
6. Proceed to Extract the UAS Bundle, on page 7.

Extract the UAS Bundle

Once the USP ISO has been mounted, the UAS bundle must be extracted from the ISO in order to prepare the configuration files required for deployment.

These instructions assume you are already logged on to the Onboarding Server.

To extract the UAS bundle:

1. Navigate to the tools directory within the ISO mount.

   cd /var/usp-iso/tools/
2. Launch the *usp-uas-installer.sh* script.

   sudo ./usp-uas-installer.sh

   The script extracts the files that comprise the UAS bundle to */opt/cisco/usp/uas-installer*.
3. Verify that files have been extracted.

   Example output:

   ll /opt/cisco/usp/uas-installer

```
total 12
drwxr-xr-x. 5 root root 4096 May 11 08:04 common
drwxr-xr-x. 2 root root 4096 May 11 08:04 images
drwxr-xr-x. 2 root root 4096 May 11 08:04 scripts
```

ll /opt/cisco/usp/uas-installer/images/

```
total 707580
-rw-r--r--. 1 root root 723898880 May 10 15:40 usp-uas-1.0.0-601.qcow2
```

ll /opt/cisco/usp/uas-installer/scripts/

```
total 56
-rwxr-xr-x. 1 root root  5460 May 11 08:04 autoit-user.py
-rwxr-xr-x. 1 root root  4762 May 11 08:04 encrypt_account.sh
-rwxr-xr-x. 1 root root  3945 May 11 08:04 encrypt_credentials.sh
-rwxr-xr-x. 1 root root 13846 May 11 08:04 uas-boot.py
-rwxr-xr-x. 1 root root  5383 May 11 08:04 uas-check.py
-rwxr-xr-x. 1 root root 10385 May 11 08:04 usp-tenant.py
```

4. Proceed to Extract the UEM VM Image, on page 8.

Extract the UEM VM Image

The image files required to deploy the UEM are distributed as part of an RPM bundle. The bundle is called "usp-em-bundle-<*version*>-1.x86_64.rpm" and it is distributed as part of the USP ISO image.

The UEM image file is called "em-<*version*>.qcow2". Prior to installing the UGP VNF, you must extract this file from the ISO.

Important These instructions assume that you have already mounted the USP ISO.

To extract the files:

1. Navigate to the directory containing the rpm bundles.

   cd /var/usp-iso/repo

2. View the contents of, and information about the bundle.

   rpm -qilp usp-em-bundle-*<version>*-1.x86_64.rpm

   Example output:

   ```
   <---SNIP--->
   /opt/cisco/usp/bundles/em-bundle/em-5_7_0_1481.qcow
   /opt/cisco/usp/bundles/em-bundle/usp-build-info.json
   /opt/cisco/usp/bundles/em-bundle/usp-bundle-manifest.yml
   <---SNIP--->
   ```

3. Extract the required artifacts from the bundle.

   a. Make a directory and ensure that it is empty.

      mkdir -p /tmp/artifacts; rm -rf /tmp/artifacts/*

   b. Navigate to the directory just created.

      cd /tmp/artifacts/

c. Extract the UEM image file.

```
rpm2cpio /var/usp-iso/repo/usp-em-bundle-<version>-1.x86_64.rpm | cpio
 -idmv ./opt/cisco/usp/bundles/em-bundle/em-<version>.qcow2
```

d. Ensure that the image has been extracted.

```
ls -l ./opt/cisco/usp/bundles/em-bundle/em-<version>.qcow2
```

4. Proceed to Extract the UGP VM Image, on page 9.

Extract the UGP VM Image

The image files required to deploy the UGP are distributed as part of an RPM bundle. The bundle is called "usp-ugp-bundle-<*version*>.x86_64.rpm" and it is distributed as part of the USP ISO image.

The UGP image file is called "qvpc-di-<*version*>.qcow2.tgz". Prior to installing the UGP VNF, you must extract the image files from the ISO.

Important These instructions assume that you have already mounted the USP ISO.

To extract the files:

1. Navigate to the directory containing the rpm bundles.

```
cd /var/usp-iso/repo
```

2. View the contents of, and information about the bundle.

```
rpm -qilp usp-ugp-bundle-<version>-1.x86_64.rpm
```

Example output:

```
<---SNIP--->
/opt/cisco/usp/bundles/ugp-bundle/qvpc-di-21.4.v0.qcow2.tgz
/opt/cisco/usp/bundles/ugp-bundle/qvpc-di-21.4.v0.qcow2.tgz.md5
/opt/cisco/usp/bundles/ugp-bundle/qvpc-di-21.4.v0.qcow2.tgz.sha1
/opt/cisco/usp/bundles/ugp-bundle/qvpc-di-21.4.v0.qcow2.tgz.sha512

<---SNIP--->
```

3. Extract the required artifacts from the bundle.

a. Make a directory and ensure that it is empty.

```
mkdir -p /tmp/artifacts; rm -rf /tmp/artifacts/*
```

b. Navigate to the directory just created.

```
cd /tmp/artifacts/
```

c. Extract the UGP image files.

```
rpm2cpio /var/usp-iso/repo/usp-ugp-bundle-<version>-1.x86_64.rpm |
cpio -idmv
./opt/cisco/usp/bundles/ugp-bundle/qvpc-di-<version>.qcow2.tgz
```

d. Ensure that the images have been extracted.

```
ls -l ./opt/cisco/usp/bundles/ugp-bundle/qvpc-di-<version>.qcow2.tgz
```

4. Extract the CF qcow2 image.

   ```
   tar -zxvf qvpc-di-<version>.qcow2.tgz qvpc-di-cf-<version>.qcow2
   ```

5. Extract the SF qcow2 image.

   ```
   tar -zxvf qvpc-di-<version>.qcow2.tgz qvpc-di-sf-<version>.qcow2
   ```

6. Proceed to Upload the USP VM Images to Glance, on page 10.

Upload the USP VM Images to Glance

The UAS, UEM, and UGP VM images extracted from the USP ISO must be uploaded into OpenStack Glance.

To upload the images to Glance:

1. Login to OSP-D and make sure to “su - stack” and “source stackrc”. Determine the name of the heat *stack_name*.

   ```
   heat stack-list
   ```

2. Source the rc file for the stack.

   ```
   source ~/  <stack_name> rc
   ```

3. Upload the UAS image.

   ```
   glance image-create --file usp-uas- <version> .qcow2 --container-format
    bare --disk-format qcow2 --name ultra-autovnf- <version>
   ```

4. Upload the UEM image.

   ```
   glance image-create --file em- <version> .qcow2 --container-format bare
    --disk-format qcow2 --name ultra-em- <version>
   ```

5. Upload the CF image.

   ```
   glance image-create --file qvpc-di-cf- <version> .qcow2 --container-format
    bare --disk-format qcow2 --name ultra-cf- <version>
   ```

6. Upload the SF image.

   ```
   glance image-create --file qvpc-di-xf- <version> .qcow2 --container-format
    bare --disk-format qcow2 --name ultra-sf- <version>
   ```

7. Proceed to Deploy the AutoVNF VM, on page 10.

Deploy the AutoVNF VM

The VM for AutoVNF is deployed using *boot_uas.py* script provided with the UAS bundle. The script is located in the following directory:

/opt/cisco/usp/bundles/uas-bundle/tools

This script includes a number of deployment parameters for the VM. These parameters are described in the help information pertaining to the script which can be accessed by executing the following command:

```
./boot_uas.py -h
```

The help information is provided as an appendix in this document. Refer to boot_uas.py Help.

Important These instructions assume you are already logged on to the Onboarding Server.

To deploy the AutoVNF VM:

1. Navigate to the directory containing the *boot_uas.py* file.

   ```
   cd /opt/cisco/usp/bundles/uas-bundle/tools
   ```

2. Deploy the AutoVNF VM.

   ```
   ./boot_uas.py  --autovnf --openstack --image <image_name> --flavor <flavor_name> --net <network_name>
   ```

   There are additional arguments that can be executed with this script based on your deployment scenario. Refer to boot_uas.py Help for details.

Important Both version 2 and 3 of OpenStack Keystone APIs are supported. You can specify the desired version using the **--os_identity_api_version** argument with this script. For example to specify the use of version 3, add the argument **--os_identity_api_version 3**. The default is version 2.

   Upon executing the script, you are prompted to enter user crendentials for performing operations within the AutoVNF VM.

3. Provide the requested information.

   - **AutoVNF VM Login Password**: The password for the default user account, which is named *ubuntu*.
   - **AutoVNF API Access password for "admin"**: The password for the ConfD administrator user, which is named admin.
   - **AutoVNF API Access password for "oper"**: The password for the ConfD operator user, which is named oper.
   - **AutoVNF API Access password for "security"**: The password for the ConfD security administrator user, which is named security-admin.

Important Ensure that all passwords meet the requirements specified in Password Requirements and Login Security.

4. Log on to the AutoVNF VM as *ubuntu*. Use the password that was created earlier for this user.
5. Become the root user.

   ```
   sudo -i
   ```

6. Prepare the *system.cfg* file. This will serve as the Day-0 config for the VNF. Refer to Sample system.cfg File for an example configuration file.

Important Though administrative user credentials can be specified in clear text in the system.cfg file, it is not recommended. For security purposes, it is recommended that you configure a secure token for the user account in the VNF configuration file and reference that file as part of the VDU catalog pertaining to the CF using the **login-credential** parameter. In the *system.cfg* file, use the *$CF_LOGIN_USER* and *$CF_LOGIN_PASSWORD* variables as follows to call the values configured for the secure token:

```
configure
  context local
    administrator $CF_LOGIN_USER password $CF_LOGIN_PASSWORD ftp
```

7. Upload the *system.cfg* to the */opt/cisco/usp/uploads/* directory on the AutoVNF VM.
8. Copy the ESC scripts from the */opt/cisco/usp/uas/autovnf/vnfms/esc-scripts* directory on the AutoVNF VM to the VNFM (ESC) VMs.

   These are custom scripts which aid in the VNF instantiation.

   a. Connect to the master VNFM (ESC) VM and copy the scripts.

   ```
   cd /opt/cisco/usp/uas/autovnf/vnfms/esc-scripts
   scp esc-vpc-di-internal-keys.sh <esc_user>@<master_esc_vm_address>:
   opt/cisco/esc/esc-scripts/esc-vpc-di-internal-keys.sh
   scp esc_vpc_chassis_id.py<esc_user>@<master_esc_vm_address>:
   opt/cisco/esc/esc-scripts/esc_vpc_chassis_id.py
   scp esc_volume_em_staging.sh <esc_user>@<master_esc_vm_address>:
   /opt/cisco/esc/esc-scripts/esc_volume_em_staging.sh
   ```

   b. Connect to the standby VNFM (ESC) VM and copy the scripts.

   ```
   scp esc-vpc-di-internal-keys.sh <esc_user>@<standby_esc_vm_address>:
   opt/cisco/esc/esc-scripts/esc-vpc-di-internal-keys.sh
   scp esc_vpc_chassis_id.py<esc_user>@<standby_esc_vm_address>:
   opt/cisco/esc/esc-scripts/esc_vpc_chassis_id.py
   scp esc_volume_em_staging.sh <esc_user>@<standby_esc_vm_address>:
   /opt/cisco/esc/esc-scripts/esc_volume_em_staging.sh
   ```

9. Confirm that the VNFM has been configured with the VIM connectors for the VNF tenants. A connector is needed for each tenant. Refer to Sample ESC VIM Connector Configuration for an example configuration.

   a. Connect to the master VNFM (ESC) VM.

   b. Log on to the ConfD command line.

   ```
   /opt/cisco/esc/confd/bin/confd_cli -C
   ```

   c. Confirm the VIM connector configuration.

   ```
   show running-config esc_system_config vim_connectors vim_connector
    <vim_connector_name>
   ```

   If the connectors have not been configured, refer to the documentation for the appropriate version of ESC software. ESC product documentation is available here: https://www.cisco.com/c/en/us/support/cloud-systems-management/elastic-services-controller-esc/tsd-products-support-series-home.html

Important The OpenStack Keystone configuration version specified for the authentication URL in the connector must match the version used when deploying AutoVNF and the version specified in the AutoVNF configuration file.

d. Repeat step 9.c, on page 12 for each VIM connector.

Important If the ESC VMs are upgraded or redeployed at any time, ensure that you reload the VIM connectors on the new or upgraded ESC VM deployment.

10. Prepare the AutoVNF VNFM configuration file.

This file provides the information necessary to allow AutoVNF to communicate with the VNFM (ESC).

A sample configuration file is provided for reference in Sample AutoVNF VNFM Configuration File.

Important The OpenStack Keystone configuration version specified in the VNFM configuration file used by AutoVNF must match the version used when deploying AutoVNF and the version specified in the ESC VIM connector(s). Set the **api-version** parameter to the appropriate version type.

Important If the ESC VMs are upgraded or redeployed at any time after the AutoVNF is deployed, you may need to change the ESC endpoint details in the AutoVNF VNFM configuration file and reload it.

11. Save the AutoVNF VNFM configuration file to your home directory on the AutoVNF VM.

12. Prepare the AutoVNF VNF configuration file.

This file provides the VNF configuration information used by AutoVNF during the deployment process. A sample configuration file is provided for reference in Sample AutoVNF VNF Configuration File.

Caution Ensure that the network service descriptor (NSD) identified in the AutoVNF VNF configuration file is identical to the one specified in the AutoVNF VNFM configuration file.

13. Save the AutoVNF VNF configuration file to your home directory on the AutoVNF VM.

14. Proceed to Activate the AutoVNF Configuration Files, on page 13.

Activate the AutoVNF Configuration Files

Once you have completed preparing your AutoVNF VNFM and VNF configuration files, you must load the configuration and activate the deployment.

Important User credentials are configured through Secure Tokens specified in the configuration file. Ensure that passwords configured with Secure Token meet the requirements specified in Password Requirements and Login Security.

Once activated, AutoVNF proceeds with the deployment automation workflow as described in VNF Deployment Automation Overview, on page 1.

Important These instructions assume you are already logged on to the AutoVNF VM as the *root* user and that your configuration files have been prepared for your deployment as per the information and instructions in Deploy the AutoVNF VM, on page 10. These instructions also assume that AutoVNF has access to the VNFC image files (either locally or on a remote server) provided with the USP ISO.

To activate the USP deployment using AutoVNF:

1. Login to the ConfD CLI as the admin user.

   ```
   confd_cli -u admin -C
   ```

2. Enter the *admin* user password when prompted.
3. Enter the ConfD configuration mode.

   ```
   config
   ```

4. Load the AutoVNF VNFM configuration file to load the VNFM information into the AutoVNF database.

   ```
   load merge <your_vnfm_file_name> .cfg
   commit
   end
   ```

Important If you are performing this process as a result of an upgrade or redeployment, you must use the load replace variant of this command:

```
load replace <your_vnfm_file_name> .cfg
commit
end
```

5. Activate the AutoVNF VNFM configuration file.

   ```
   activate nsd <nsd_name>
   ```

Important The output of this command is a transaction-id which can be used to monitor the deployment progress. If need be, the VIM deployment can be deactivated using the **deactivate** variant of this command.

6. Load the AutoVNF VNF configuration file to load the deployment name and its attributes in the AutoVNF database.

```
load merge <your_vnf_file_name> .cfg
commit
end
```

Important If you are performing this process as a result of an upgrade or redeployment, you must use the load replace variant of this command:

```
load replace <your_vnf_file_name> .cfg
commit
end
```

7. Activate the AutoVNF VNF configuration file.

   ```
   activate nsd <nsd_name>
   ```

Important The output of this command is a transaction-id which can be used to monitor the deployment progress. If need be, the VIM deployment can be deactivated using the **deactivate** variant of this command.

8. Monitor the progress of the deployment by viewing transaction logs:

   ```
   show log <transaction_id> | display xml
   ```

   transaction_id is the ID displayed as a result of the **activate-deployment** command.

   The logs display status messages for each node in each VNF that the configuration file defines. Example success messages for the different components deployed through AutoVNF are shown below:

   - VNF:

     ```
     Fri May 12 21:44:35 UTC 2017 [Task: 1494624612779/tb1vnfd2] Successfully completed
     all Vnf Deployments.
     ```

   - Entire Deployment:

     ```
     Fri May 12 21:57:38 UTC 2017 [Task: 1494624612779] Success
     ```

Important If there are any issues seen when executing the above commands, refer to Monitoring and Troubleshooting the Deployment.

Upgrading/Redeploying the Stand-alone AutoVNF VM Instance

Use the following procedure to upgrade or redeploy the AutoVNF software image in scenarios where AutoVNF was brought up as stand-alone instance.

Important These instructions assume you are already logged on to the Onboarding Server.

1. Delete the AutoVNF VM instance.

   ./boot_uas.py --openstack --autovnf --delete *<transaction_id>*

2. *Optional*. If required remove the OpenStack artifacts which were created manually to bring up AutoVNF.
3. Follow the procedures in Deploy the USP-based VNF, on page 5 to redeploy AutoVNF with the new software version.

Important Upgrading or redeploying the VNF can be performed as part of this process or it can be performed separately. Refer to Upgrading/Redeploying VNFs Deployed Through a Stand-alone AutoVNF Instance for details and instructions.