Deploying Hyper-Converged Ultra M Models Using UAS

This chapter provides information on the following topics:

Virtual Infrastructure Manager Installation Automation

Introduction

Leveraging RedHat and OpenStack's TripleO project concepts, UAS supports the ability to automate the deployment of both the virtual infrastructure manager (VIM, the Triple O Overcloud) and the VIM Orchestrator (the TripleO Undercloud).

Installing the VIM Orchestrator and the VIM involves deploying the following components as VMs on a RedHat Enterprise Linux (RHEL) server:

- AutoIT-NVFI
- AutoDeploy
- OpenStack Platform Director (OSP-D)

VIM Orchestrator and VIM settings are maintained in configuration files which are used by AutoDeploy.

AutoDeploy processes the VIM Orchestrator configuration and works with AutoIT-NFVI to automate the deployment of a VM running OSP-D which serves as the Undercloud. Once this operation is successful, AutoDeploy processes the VIM configuration and works with AutoIT-NFVI to deploy the OpenStack Overcloud.

Notes:

- This functionality is supported only with Ultra M deployments based on OSP 10 and that leverage the Hyper-Converged architecture.

- Before the N5.5 Ultra Service Platform (USP) release, OSP-D was deployed on its own physical server. With this release, this feature leverages this same physical server for the installation of VMs for AutoIT-NFVI, AutoDeploy, and OSP-D. This server is now referred to as the Ultra M Manager Node.
- Refer to Pre-Virtual Infrastructure Manager Installation Verification, on page 4 for pre-requisites pertaining to this feature.

VIM Installation Automation Overview

Figure 1: NFVI Deployment Automation Workflow, on page 2 provides an overview of the deployment automation process. Details are provided in Table 1: Virtual Infrastructure Manager Installation Automation Workflow Descriptions, on page 2. This information assumes that all prerequisite hardware has been installed, cabled, and configured.

Note The workflow information in this section assumes a new deployment scenario. If you are using this feature in relation with an upgrade process, please contact your support representative for complete details.

Figure 1: NFVI Deployment Automation Workflow

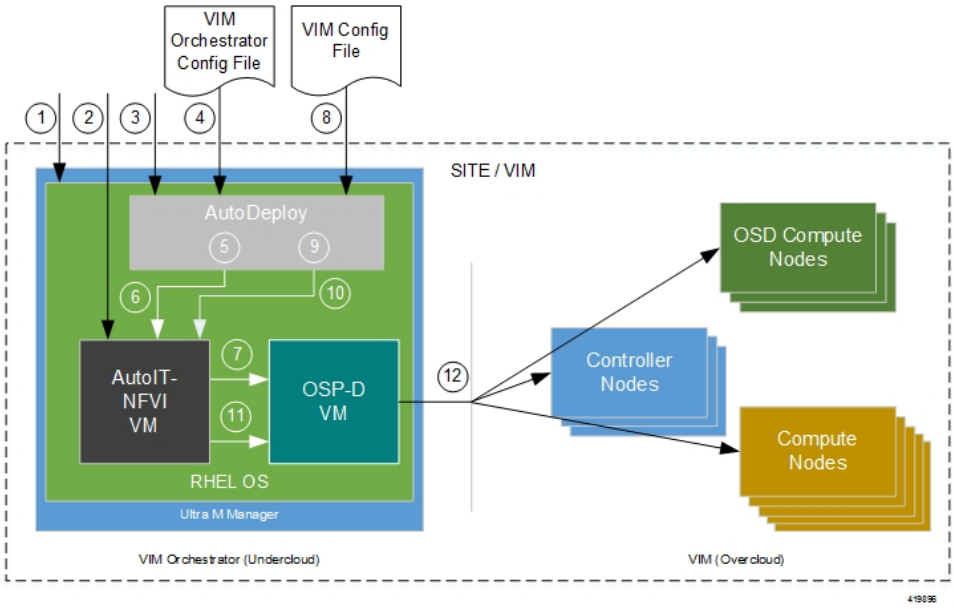


Table 1: Virtual Infrastructure Manager Installation Automation Workflow Descriptions

Callout	Description
1	Install RedHat Enterprise Linux (RHEL) operating system on bare metal hardware (Ultra M Manager Node).
2	Deploy the AutoIT-NFVI VM.

Callout	Description
3	Deploy the AutoDeploy VM.
4	Prepare the VIM Orchestrator configuration file that is used by AutoDeploy to initiate the OSP-D VM deployment process. This file includes all of the configuration information required to deploy OSP-D VM including configurations for constructs such as secure tokens, package images, NFVI point-of-presence descriptors (nfvi-popd), and the VIM Orchestrator descriptor (vim-orchd). Refer to Sample VIM Orchestrator Configuration File for more information. **Note** If you are using UWS to deploy the USP-VNF process, refer to Sample UWS VIM Orchestrator and VIM Configuration File. **Note** The VIM Orchestrator and VIM configuration information can be prepared and maintained in separate files or combined into a single file. Figure 1: NFVI Deployment Automation Workflow, on page 2 depicts the configurations in separate files for illustration purposes. Note, however, that UWS requires that both the the VIM Orchestrator and VIM configuration be provided in a single file.
5	On the AutoDeploy VM, load, commit, and then activate the configuration file prepared in previous step. Once commited, activate the previously loaded VIM Orchestrator configuration file.
6	AutoDeploy passes data from the activated configuration to AutoIT-NFVI requesting that it deploy the OSP-D VM for the Undercloud. Refer to Activate the VIM Orchestrator Deployment for more information.
7	AutoIT-NFVI deploys the OSP-D VM which serves as the Undercloud.
8	Prepare the VIM configuration file that is used by AutoDeploy to initiate the VIM (Overcloud) installation process. This file includes all of the configuration information required to deploy the VIM components including configurations for constructs such as secure tokens, node parameters, NFVI point-of-presence descriptors (nfvi-popd), and the VIM descriptor (vimd). **Note** The VIM Orchestrator and VIM configuration information can be prepared and maintained in separate files or combined into a single file. Figure 1: NFVI Deployment Automation Workflow, on page 2 depicts the configurations in separate files for illustration purposes. Note, however, that UWS requires that both the VIM Orchestrator and VIM configuration be provided in a single file.
9	On the AutoDeploy VM, load, commit, and then activate the configuration file prepared in the previous step.
10	AutoDeploy passes data from the activated configuration to AutoIT-NFVI for delivery to the OSP-D VM responsible for installing the VIM.
11	AutoIT-NFVI initiates the VIM installation by passing parameters received from AutoDeploy to the OSP-D VM.
12	The OSP-D VM installs the VIM per the configuration requirements.

Once all of the VIM servers have been successfully deployed, the process of deploying the VNF can begin as described in VNF Deployment Automation, on page 22.

Pre-Virtual Infrastructure Manager Installation Verification

Prior to installing the virtual infrastructure manager (VIM) and the VIM Orchestrator, please ensure that the following is true:

- Ensure that all required hardware is installed, powered on, cabled and configured according to the information and instructions in the *Ultra M Solutions Guide*. Refer to the following sections in that document:
  - *Hardware Specifications*
  - *Install and Cable the Hardware*
  - *Configure the Switches*
  - *Prepare the UCS C-Series Hardware*
- Ensure that all required software is available and that you have access to the Cisco-provided USP ISO image. Refer to the *Software Specifications* section of the *Ultra M Solutions Guide* for more details.

Install the VIM Orchestrator

The initial part of the Virtual Infrastructure Manager installation automation process is to install the VIM Orchestrator. You cannot install the VIM until after the VIM Orchestration installation is successful.

Note Before proceeding, ensure that all of the items in Pre-Virtual Infrastructure Manager Installation Verification, on page 4 have been verified.

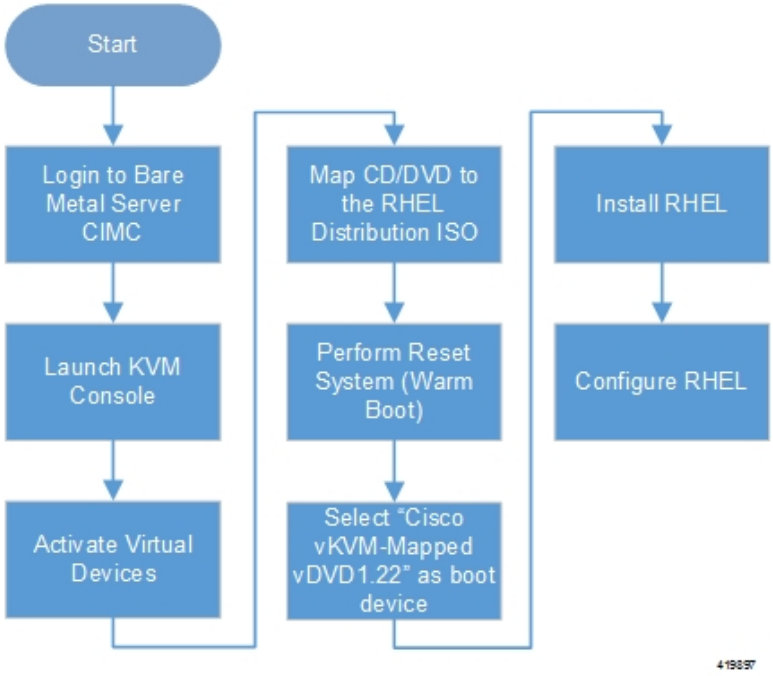
To install the VIM Orchestrator:

1. Install and Configure RHEL, on page 5.
2. Onboard the USP ISO, on page 10.
3. Extract the UAS Bundle, on page 11.
4. Deploy AutoIT-NFVI, on page 12.
5. Deploy AutoDeploy, on page 13.
6. Prepare the UWS VIM Orchestrator and VIM Configuration File, on page 15, OR Prepare the VIM Orchestrator Configuration File, on page 16 based on your deployment requirements.
7. Activate the VIM Orchestrator Deployment, on page 16.

Install and Configure RHEL

As described in VIM Installation Automation Overview, on page 2, the VIM Orchestrator (OSP-D) is deployed as a VM on top of RHEL. Figure 2: Installation Process for RHEL Bare Metal Server, on page 5 illustrates the process for installing RHEL.

Figure 2: Installation Process for RHEL Bare Metal Server



General RHEL installation information and procedures are located in the product documentation:

- https://access.redhat.com/documentation/en/red-hat-enterprise-linux/

Prior to installing RHEL, refer to Table 2: Red Hat Installation Settings, on page 5 for settings required for the VIM Orchestrator installation in Ultra M.

Note Table 2: Red Hat Installation Settings, on page 5 assumes that you are using the product's graphical user interface (GUI) for Red Hat installation.

Table 2: Red Hat Installation Settings

Parameters and Settings	Description
Installation Summary > Language Support	
English > English (United States)	Sets the language to English and the region to United States.

Parameters and Settings	Description
Installation Summary > Software Selection	
Base Environment = Virtualization Host Add-Ons for Selected Environment = Virtualization Platform	
Installation Summary > Network & Host Name	
Host name	Configure the desired host name.
Installation Summary > Network & Host Name > Ethernet (enp6s0f0) > Configure > IPv4 Setting	
IP Address Netmask Gateway DNS Server Search Domain	Configure and save settings for the network interface by which the server can be accessed externally. **Note** The first, or top-most interface shown in the list in the **Network & Host Name** screen should be used as the external interface for the server. In the above GUI path example, it is identified as "enp6s0f0". Post installation, the external interface is identified as "eno1" in RHEL.
Installation Summary > Installation Destination > CiscoUCSC-MRAID12G (sda) > I will configure partitioning > Click here to create them automatically	
Select all partitions, then click "-" / = 100GB /var = 500GB /swap = 100GB /home = remaining space /boot = 1GB	Removes any previously configured partitions and creates partitions with the required sizes. **Note** You must use LVM-based partitioning.
Installation Summary > KDUMP	
kdump = disabled	It is recommended that kdump be disabled.
Installation Summary > Begin Installation > User Settings	
Root Password	Configure and confirm the root user password.
Create user "nfvi"	Creates a new user account. This account is used during the VIM Orchestration installation to log onto the Ultra M Manager Node. **Note** Ensure that a strong password is used. It must be a minimum of 8 alpha and/or numeric characters and must contain at least 1 uppercase letter, 1 lowercase letter, 1 number, and 1 special character (e.g. @, #, $, etc.).

To install and configure RHEL:

1. Follow the CIMC processes on the bare metal server as identified in Figure 2: Installation Process for RHEL Bare Metal Server, on page 5.
2. Select the option to install Red Hat Enterprise Linux to begin the installation.
3. Configure the settings identified in Table 2: Red Hat Installation Settings, on page 5.
4. Begin the installation and configure the User Setting identified in Table 2: Red Hat Installation Settings, on page 5.
5. Click **Reboot** once the installation is complete.
6. Log in to RedHat as the **nfvi** user.
7. Set password-less sudo access for **nfvi**.

```
echo "nfvi ALL=(root) NOPASSWD:ALL" | tee -a /etc/sudoers.d/nfvi
chmod 0440 /etc/sudoers.d/nfvi
```

8. Configure the network interfaces and network bridges.

Note If any of the network interface or bridge configuration files do not exist, create the related configuration files. Example configuration files are provided in Example RedHat Network Interface and Bridge Configuration Files.

a. Configure the eno2 interface by appending the following parameters to the */etc/sysconfig/network-scripts/ifcfg-eno2* file.

```
<--SNIP-->
DEVICE=eno2
ONBOOT=yes
BRIDGE=br-ex
NM_CONTROLLED=no
NETMASK=<netmask>
GATEWAY=<gateway_address>
```

b. Configure the eno1 interface by appending the following parameters to the */etc/sysconfig/network-scripts/ifcfg-eno1* file.

```
<--SNIP-->
DEVICE=eno1
ONBOOT=yes
BRIDGE=br-ctlplane
NM_CONTROLLED=no
```

c. Configure the br-ex network bridge by adding the following parameters to the */etc/sysconfig/network-scripts/ifcfg-br-ex* file.

```
<--SNIP-->
DEVICE=br-ex
DEFROUTE=yes
TYPE=Bridge
ONBOOT=yes
BOOTPROTO=static
NM_CONTROLLED=no
DELAY=0
IPADDR=<external_ip_address>
NETMASK=<netmask>
```

GATEWAY=<*gateway_address*>
PREFIX="24"
DNS1="<*DNS_server_address*>"
DOMAIN="<*domain_name*>"
IPV4_FAILURE_FATAL="yes"

d Configure the br-ctlplane bridge by adding the following parameters to the */etc/sysconfig/network-scripts/ifcfg-br-ctlplane* file.

```
<--SNIP-->
DEFROUTE=yes
TYPE=Bridge
ONBOOT=yes
BOOTPROTO=static
NM_CONTROLLED=no
DELAY=0
DEVICE=br-ctlplane
```

Caution Once configured, it is recommended that you do not make any changes to the network interface or bridge configuration. Doing so will require that you redeploy AutoIT-NFVI and AutoDeploy.

9 Create and prepare the directories required for installing the UAS components.

```
sudo mkdir -p /var/cisco/isos
sudo mkdir -p /var/cisco/disks
sudo chmod 777 -R /var/cisco
```

10 Reboot the bare metal server.

```
sudo reboot
```

11 Login as a root user upon reboot.

Note If the server is not accessible via the configured IP address, login into the server's KVM console and troubleshoot the configuration.

12 Validate the network configuration.

```
ifconfig | more
```

Example output:

```
br-ctlplane: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet6 fe80::22c:c8ff:fed9:f176  prefixlen 64  scopeid 0x20<link>
        ether 00:2c:c8:d9:f1:76  txqueuelen 1000  (Ethernet)
        RX packets 52  bytes 7044 (6.8 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 8  bytes 648 (648.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

br-ex: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 172.25.22.59  netmask 255.255.255.0  broadcast 172.25.22.255
        inet6 fe80::22c:c8ff:fed9:f177  prefixlen 64  scopeid 0x20<link>
        ether 00:2c:c8:d9:f1:77  txqueuelen 1000  (Ethernet)
        RX packets 1394  bytes 122906 (120.0 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 717  bytes 71762 (70.0 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

eno1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet6 fe80::22c:c8ff:fed9:f176  prefixlen 64  scopeid 0x20<link>
        ether 00:2c:c8:d9:f1:76  txqueuelen 1000  (Ethernet)
```

```
        RX packets 57  bytes 8072 (7.8 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 16  bytes 1296 (1.2 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
        device memory 0xc7000000-c70fffff

eno2: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet6 fe80::22c:c8ff:fed9:f177  prefixlen 64  scopeid 0x20<link>
        ether 00:2c:c8:d9:f1:77  txqueuelen 1000  (Ethernet)
        RX packets 1497  bytes 148860 (145.3 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 726  bytes 72476 (70.7 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
        device memory 0xc6f00000-c6ffffff

enp6s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        ether 00:2c:c8:68:3b:ec  txqueuelen 1000  (Ethernet)
        RX packets 1  bytes 68 (68.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

enp7s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        ether 00:2c:c8:68:3b:ed  txqueuelen 1000  (Ethernet)
        RX packets 1  bytes 68 (68.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1  (Local Loopback)
        RX packets 84  bytes 6946 (6.7 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 84  bytes 6946 (6.7 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

virbr0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        inet 192.168.122.1  netmask 255.255.255.0  broadcast 192.168.122.255
[root@rhel-baremetal nfvi]# brctl show
bridge name bridge id  STP enabled interfaces
br-ctlplane  8000.002cc8d9f176 no  eno1
br-ex  8000.002cc8d9f177 no  eno2
virbr0  8000.5254003d7549 yes  virbr0-nic
```

13 Perform the RHEL subscription-manager registration.

From Content Delivery Network (CDN) servers:

sudo subscription-manager config --server.proxy_hostname=*<proxy_url>* **--server.proxy_port=80**

subscription-manager register --username *<username>* **--password** *<password>*

subscription-manager attach -auto

sudo subscription-manager status

From Satellite Servers:

rpm -Uvh http://*<satellite_server_domain>***/pub/katello-ca-consumer-latest.noarch.rpm**

subscription-manager register --org="*<organization>***" --activationkey="***<activation_key>***"**

Example output:

```
+----------------------------------------+
   System Status Details
+----------------------------------------+
Overall Status: Current
```

14 Install the virtualization packages.

yum install virt-install -y

Example output:

```
Loaded plugins: langpacks, product-id, search-disabled-repos, subscription-manager
rhel-7-server-rpms                                                       | 3.5 kB
00:00:00
(1/3): rhel-7-server-rpms/7Server/x86_64/group                           | 709 kB
00:00:01
(2/3): rhel-7-server-rpms/7Server/x86_64/updateinfo                      | 2.3 MB
00:00:02
(3/3): rhel-7-server-rpms/7Server/x86_64/primary_db                      |  42 MB
00:00:16
Resolving Dependencies
Loaded plugins: langpacks, product-id, search-disabled-repos, subscription-manager
rhel-7-server-rpms | 3.5 kB 00:00:00
(1/3): rhel-7-server-rpms/7Server/x86_64/group | 709 kB 00:00:01
(2/3): rhel-7-server-rpms/7Server/x86_64/updateinfo | 2.3 MB 00:00:02
(3/3): rhel-7-server-rpms/7Server/x86_64/primary_db | 42 MB 00:00:16
Resolving Dependencies
yum install virt-viewer -y
Loaded plugins: langpacks, product-id, search-disabled-repos, subscription-manager
Resolving Dependencies
--> Running transaction check
---> Package virt-viewer.x86_64 0:5.0-7.el7 will be installed
```

15 Proceed to Onboard the USP ISO, on page 10.

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 Ultra M Manager Node. 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-VNF, 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.

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

Note This procedure is not necessary if you are deploying a VNF on a Hyper-Converged Ultra M mode and have already deployed the VIM Orchestrator and the VIM using the information and instructions in Virtual Infrastructure Manager Installation Automation, on page 1.

Note Before attempting to deploy the Ultra M Manager Node, ensure that the USP Installation Prerequisites have been met.

To onboard the ISO package:

1. Log on to the Ultra M Manager Node.
2. Download the USP ISO bundle and related files pertaining to the release.
3. Create a mount point on the Ultra M Manager Node 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:
   ```
   -r--r--r--. 1 root root  3371 Aug 22 23:57 release-manifest.json
   dr-xr-xr-x. 2 root root  2048 Aug 22 23:57 repo
   -r--r--r--. 1 root root  2603 Aug 22 23:57 rpm-import-dev-gpg.README
   -r-xr-xr-x. 1 root root 13736 Aug 22 23:51 setup.sh
   dr-xr-xr-x. 2 root root  2048 Aug 22 23:57 tools
   ```
6. Proceed to Extract the UAS Bundle, on page 11.

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.

Note These instructions assume you are already logged on to the server on which AutoIT-NFVI, AutoDeploy, and VIM-Orchestrator VMs are to be installed and that the USP ISO has been mounted.

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-installertotal 20
drwxr-xr-x 5 root root 4096 Aug 18 23:42 ./
drwxr-xr-x 6 root root 4096 Aug 18 23:42 ../
drwxr-xr-x 5 root root 4096 Aug 18 23:42 common/
drwxr-xr-x 2 root root 4096 Aug 18 23:42 images/
drwxr-xr-x 2 root root 4096 Aug 18 23:42 scripts/
ll /opt/cisco/usp/uas-installer/images/
total 711940
drwxr-xr-x 2 root root      4096 Aug 18 23:42 ./
drwxr-xr-x 5 root root      4096 Aug 18 23:42 ../
-rw-r--r-- 1 root root 729010688 Aug 17 23:29 usp-uas-1.0.0-1074.qcow2
ll /opt/cisco/usp/uas-installer/scripts/
total 80
-rwxr-xr-x. 1 root root   806 Aug 29 18:14 auto-deploy-booting.sh
-rwxr-xr-x. 1 root root  5460 Aug 29 18:14 autoit-user.py
-rwxr-xr-x. 1 root root   811 Aug 29 18:14 auto-it-vnf-staging.sh
-rwxr-xr-x. 1 root root  4762 Aug 29 18:14 encrypt_account.sh
-rwxr-xr-x. 1 root root  3945 Aug 29 18:14 encrypt_credentials.sh
-rwxr-xr-x. 1 root root 14031 Aug 29 18:14 start-ultram-vm.py
-rwxr-xr-x. 1 root root 14605 Aug 29 18:14 uas-boot.py
-rwxr-xr-x. 1 root root  5384 Aug 29 18:14 uas-check.py
-rwxr-xr-x. 1 root root 11283 Aug 29 18:14 usp-tenant.py
```

4 Proceed to Deploy AutoIT-NFVI, on page 12.

Deploy AutoIT-NFVI

Deployment of the AutoIT-NFVI VM is facilitated through a script. The script relies on user inputs to perform pre-requisite configurations and account encryptions. Additionally, the script removes existing AutoIT-NFVI deployments that may already exist.

The following information is required to execute the script:

- Image (QCOW2): The path and file name for the UAS qcow2 file. For example:

  /opt/cisco/usp/uas-installer/images/usp-uas-1.0.0-1074.qcow2

- **IP Address:** The IP address to be assigned to AutoIT-NFVI's external network interface.
- **Gateway:** The gateway assigned to AutoIT-NFVI's external network interface.
- **Netmask:** The mask to be assigned to AutoIT-NFVI's external network interface.
- **Provisioning Network IP Address:** The IP address to be assigned to the provisioning network interface. Within Hyper-Converged Ultra M models, this interface is used by the Ultra M Health Monitoring function.
- **Provisioning Network Gateway:** The IP address of the provisioning network gateway.
- **Provisioning Network Netmask:** The netmask to be assigned to the provisioning network interface.
- **Login Password:** The password for the default user account, which is named ubuntu.
- **ConfD Admin Password:** The password for the ConfD administrator user, which is named admin.
- **ConfD Oper Password:** The password for the ConfD operator user, which is named oper.
- **ConfD Security Password:** The password for the ConfD security administrator user, which is named security-admin.

Note All of the above passwords must be a minimum of 8 alpha and/or numeric characters and must contain at least 1 uppercase letter, 1 lowercase letter, 1 number, and 1 special character (e.g. @, #, $, etc.).

The script allocates the following resources to the AutoIT-NFVI VM:

- 2 VCPUs
- 4 GB RAM
- 80 GB Root Disk

Note These instructions assume you are already logged on to the server on which AutoIT-NFVI, AutoDeploy, and VIM-Orchestrator VMs are to be installed and that the USP ISO has been mounted.

To deploy the AutoIT-NFVI VM:

1. Navigate to the */opt/cisco/usp/uas-installer/scripts* directory:

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

2. Execute the *start_ultram_vm.py* script:

   ```
   start-ultram-vm.py --auto-nfvi
   ```

3. Enter the information requested by the script for your deployment.

   The script displays progress information. For example:

   ```
   2017-08-23 16:25:01,070 - Removing old deployment if exists
   2017-08-23 16:25:01,126 - Preparing root disk /var/cisco/autoit-nfvi/autoit-nfvi.qcow2
   2017-08-23 16:25:01,582 - Resizing disk to 80GB
   2017-08-23 16:25:05,742 - NFVI Deployment started
   2017-08-23 16:25:07,031 - NFVI VM started successfully
   ```

4. Verify that the AutoIT-NFVI VM is running.

   ```
   virsh list -all
   ```

   Example command output:

   ```
   Id    Name                           State
   ----------------------------------------------------
    2    nfvi                           running
   ```

5. Proceed to Deploy AutoDeploy, on page 13.

Deploy AutoDeploy

Note The information and instructions provided here are only applicable when AutoDeploy is used in the VIM Orchestrator installation process.

Deployment of the AutoDeploy VM is facilitated through a script. The script relies on user inputs to perform pre-requisite configurations and account encryptions. Additionally, the script removes existing AutoDeploy deployments that may already exist.

The following information is required to execute the script:

- **Image (QCOW2):** The path and file name for the UAS qcow2 file. For example:

/opt/cisco/usp/uas-installer/images/usp-uas-1.0.0-1074.qcow2

- **IP Address:** The IP address to be assigned to AutoDeploy's external network interface.
- **Gateway:** The gateway assigned to AutoDeploy's external network interface.
- **Netmask:** The mask to be assigned to AutoDeploy's external network interface.
- **Login Password:** The password for the default user account, which is named ubuntu.
- **ConfD Admin Password:** The password for the ConfD administrator user, which is named admin.
- **ConfD Oper Password:** The password for the ConfD operator user, which is named oper.
- **ConfD Security Password:** The password for the ConfD security administrator user, which is named security-admin.

Note All of the above passwords must be a minimum of 8 alpha and/or numeric characters and must contain at least 1 uppercase letter, 1 lowercase letter, 1 number, and 1 special character (e.g. @, #, $, etc.).

The script allocates the following resources to the AutoDeploy VM:

- 2 VCPUs
- 4 GB RAM
- 80 GB Root Disk

Note These instructions assume that you are logged into the AutoIT-NFVI server and that the USP ISO has been mounted.

To deploy the AutoDeploy VM:

1. Navigate to the */opt/cisco/usp/uas-installer/scripts* directory:

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

2. Execute the *start_ultram_vm.py* script:

   ```
   start-ultram-vm.py --auto-deploy
   ```

3. Enter the information requested by the script for your deployment.

   The script displays progress information. For example:

   ```
   2017-08-23 16:30:31,163 - Removing old deployment if exists
   2017-08-23 16:30:31,179 - Preparing root disk /var/cisco/auto-deploy/auto_deploy.qcow2
   2017-08-23 16:30:31,606 - Copying config disk /var/cisco/auto-deploy/cfg.iso
   2017-08-23 16:30:31,609 - Resizing disk to 80GB
   2017-08-23 16:30:35,731 - Deployment started
   2017-08-23 16:30:36,459 - Auto-deploy started successfully
   ```

4. Verify that the AutoDeploy VM is running.

   ```
   virsh list -all

   Id    Name                           State
   ----------------------------------------------------
    2     nfvi                           running
    3     auto-deploy                    running
   ```

Note It is recommended that you do not make any changes to the AutoIT-NFVI network interface or bridge configuration. Doing so will require that you redeploy AutoDeploy.

5 Choose the desired method by which to continue the deployment process:

- Use Ultra Web Services (UWS) to continue the deployment process. To use this method, proceed to Prepare the UWS VIM Orchestrator and VIM Configuration File, on page 15.
- Use the ConfD CLI/APIs to continue the deployment process. To use this method, proceed to Prepare the VIM Orchestrator Configuration File, on page 16.

Note You will need access to both the OpenStack GUI and CLI to complete the configuration procedures.

Prepare the UWS VIM Orchestrator and VIM Configuration File

If you will be using the UWS GUI to complete the deployment process, you will need to prepare the configuration file specifying the parameters for deploying the VIM Orchestrator and the VIM. This is an XML file which is uploaded to the UWS in order to initiate the deployment process.

Unlike the deployment process using the ConfD CLI, UWS requires that the VIM Orchestrator and VIM parameters be defined in the same configuration file.

This file includes all of the configuration information required to deploy OSP-D VM including configurations for constructs such as secure tokens, package images, NFVI point-of-presence descriptors (nfvi-popd), and the VIM Orchestrator descriptor (vim-orchd), node parameters, and the VIM descriptor (vimd).

A sample configuration file is provided for reference in Sample UWS VIM Orchestrator and VIM Configuration File.

Refer to AutoDeploy Configuration File Constructs for more information on the constructs used in this file.

Note These instructions assume you are already logged on to the AutoDeploy VM as the root user.

To prepare the service deployment configuration file:

1 Create and edit your service deployment configuration file using the XML editing tool of your choice according to your VNF deployment requirements. Use the sample provided in Sample UWS VIM Orchestrator and VIM Configuration File as a reference.

2 Copy your configuration file to the AutoDeploy VM.

3 Enter the following URL in a supported web browser:

https://<auto-deploy-vm-floating-ip-address>:8443/

Note Refer Ultra Web Services for a list of supported web browsers.

4 Refer to the UWS online help for information on how to upload the service deployment configuration file and for further instructions on how to activate it.

Prepare the VIM Orchestrator Configuration File

As described in VIM Installation Automation Overview, on page 2, the VIM Orchestrator configuration file is used by AutoDeploy to activate the OSP-D VM deployment process.

This file includes all of the configuration information required to deploy OSP-D VM including configurations for constructs such as secure tokens, package images, NFVI point-of-presence descriptors (nfvi-popd), and the VIM Orchestrator descriptor (vim-orchd).

Note As stated in Table 1: Virtual Infrastructure Manager Installation Automation Workflow Descriptions, the VIM Orchestrator and VIM configuration information can be prepared and maintatined in separate files or combined into a single file. Figure 3: UAS Deployment Automation Workflow for a Single VNF, on page 23 depicts the configurations in separate files for illustration purposes.

A sample configuration file is provided for reference in Sample VIM Orchestrator Configuration File.

Refer to AutoDeploy Configuration File Constructs for more Information on the constructs used in this file. You can also refer to RedHat user documentation for information on how to install the satellite server if your deployment requires:

https://access.redhat.com/documentation/en-US/Red_Hat_Network_Satellite/5.0/html/Installation_Guide/s1-intro-sat.html

Note These instructions assume you are already logged on to the AutoDeploy VM as the root user.

To prepare the VIM Orchestrator configuration file:

1. Create and edit your VIM Orchestrator configuration file according to your deployment requirements. Use the sample provided in Sample VIM Orchestrator Configuration File as a reference.
2. Save the VIM Orchestrator configuration file you have created to your home directory.
3. Proceed to Activate the VIM Orchestrator Deployment, on page 16.

Activate the VIM Orchestrator Deployment

Once you have completed preparing your VIM Orchestrator configuration file, you must load the configuration and activate the deployment in order to bring up the OSP-D VM and the Undercloud.

Note These instructions assume you are already logged on to the AutoDeploy VM as the root user and that your VIM Orchestrator configuration file has been prepared for your deployment as per the information and instructions in Prepare the VIM Orchestrator Configuration File, on page 16.

To activate the OSP-D VM deployment using AutoDeploy:

1. Login to the ConfD CLI as the *admin* user.
   ```
   confd_cli -u admin -C
   ```
2. Enter the ConfD configuration mode.
   ```
   config
   ```

3 Load the VIM Orchestrator configuration file to provide the deployment artifacts to the VIM.

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

Note If changes are made to the VIM Orchestrator configuration file after it was committed, you can apply the changes using the **load replace** command instead of the **load merge** command. You will also need to **commit** your changes.

4 Activate the deployment to begin creating the VMs.

```
activate-vim-orch-deployment service-deployment-id <deployment-id>
```

Note 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.

5 Monitor the progress of the deployment by:

a Viewing transaction logs:

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

transaction_id is the ID displayed as a result of the **activate-deployment** command in step 4, on page 17.

```
config xmlns="http://tail-f.com/ns/config/1.0">
  <log xmlns="http://www.cisco.com/usp/nfv/usp-autodeploy-oper">
    <tx-id>1503523153766</tx-id>
   <log>Wed Aug 23 21:19:15 UTC 2017 [Task: 1503523153766] Started VIM Orch deployment
 VimOrchDeploymentRequest [transType=ACTIVATE, serviceDeploymentId=north-east,
siteList=[]]
Wed Aug 23 21:20:50 UTC 2017 [Task: 1503523153766/sjc-vim-orch] Starting VimOrch
deployment
Wed Aug 23 21:20:50 UTC 2017 [Task: 1503523153766/sjc-vim-orch] Current VIM Orch
deployment status is deployment-unknown
Wed Aug 23 21:20:50 UTC 2017 [Task: 1503523153766/sjc-vim-orch] Deploying VIM Orch
Wed Aug 23 21:56:51 UTC 2017 [Task: 1503523153766/sjc-vim-orch] VIM Orch deployed
successfully
Wed Aug 23 21:56:51 UTC 2017 [Task: 1503523153766] Success
</log>
  </log>
</config>
```

b Checking the service deployment:

```
show service-deployment
```

Example command output:

Add to output

```
service-deploymentr north-east
siter auto-test-sjc
  nfvi-popr nfvi-deployment-status "Required Undercloud services are UP"
  nfvi-popr vim-orch status deployment-success
  nfvi-popr vim-orch steps-total 84
  nfvi-popr vim-orch steps-completed 84
  nfvi-popr vim-orch version "Red Hat OpenStack Platform release 10.0 (Newton)"
                                                    FIRMWARE  IP       BIOS
     IS           PHYSNET
NFVI NODE ID               UUID  STATUS  ROLE      VERSION   ADDRESS  VERSION  ID
SIZE  JOURNAL  ID  ID
-------------------------------------------------------------------------------------------
autoit-nfvi-physical-node  -     up      vim-orch  -         -        -
```

The deployment-status in the above output changes based on the current progress. The command can be re-issued multiple times to refresh the status.

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

6 *Optional.* Monitor the VIM Orchestrator deployment from within the AutoIT-NFVI ConfD configuration mode.

a Login to the ConfD CLI as the admin user.

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

b Enter the ConfD configuration mode.

```
config
```

c View the transaction log.

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

The transaction_id can be found by executing the show transactions command.

Example command output:

```
<config xmlns="http://tail-f.com/ns/config/1.0">
  <log xmlns="http://www.cisco.com/usp/nfv/usp-nfvi-oper">
    <tx-id>1503523250-177517</tx-id>
   <log>2017-08-23 21:20:50,242 - RPC triggered for VIM-ORCH deployment: sjc-vim-orch,
 deactivate: False
2017-08-23 21:20:50,251 - Notify deployment of type 'deployment-started'
<--- SNIP --->
<--- SNIP --->
2017-08-23 21:56:46,710 - Notify deployment progress with message 'Verifying Undercloud
 services' tx_type 'vim-orch-deployment' (83/84)
2017-08-23 21:56:46,745 - Verifying if undercloud services are active
2017-08-23 21:56:46,745 - Executing command
[stack@172.25.22.56:/tmp/undercloud/osp10_post_install.sh -s verify_services]
2017-08-23 21:56:47,054 - Notify deployment progress with message 'Required Undercloud
 services are UP' tx_type 'vim-orch-deployment' (84/84)
2017-08-23 21:56:47,099 - Verified all required undercloud services are active
2017-08-23 21:56:47,100 - Undercloud packages installed successfully and servics are
 UP in VIM Orchestrator VM
2017-08-23 21:56:47,122 - VIM Orch Transaction: 1503523250-177517 for deployment:
sjc-vim-orch completed successfully.
2017-08-23 21:56:47,150 - Notify deployment of type 'deployment-completed'
</log>
  </log>
</config>
```

If there are any issues with your VIM deployment, refer to Monitoring AutoDeploy Operations and Monitoring AutoIT-VNF Operations for information on collecting logs.

7 Proceed to Install the VIM, on page 18.

Install the VIM

Upon successful installation of the VIM Orchestrator, the OSP-D VM is operational. You must now install the VIM.

Note The information and instructions in this section assume that the VIM Orchestrator have been installed according

To install the VIM Orchestrator:

1. Prepare the VIM Configuration File, on page 19.

Note If you are using UWS for your deployment, VIM parameters would have already been defined in the same configuration file as those for the VIM Orchestrator. No further steps are necessary. Refer to Prepare the UWS VIM Orchestrator and VIM Configuration File, on page 15 for more information.

2. Activate the VIM Deployment, on page 20.

Prepare the VIM Configuration File

As described in VIM Installation Automation Overview, on page 2, the VIM configuration file is used by AutoDeploy to activate the VIM installation automation process.

This file includes all of the configuration information required to deploy the VIM components including configurations for constructs such as secure tokens, node parameters, NFVI point-of-presence descriptors (nfvi-popd), and the VIM descriptor (vimd).

Note As stated in the Table 1: Virtual Infrastructure Manager Installation Automation Workflow Descriptions, the VIM Orchestrator and VIM configuration information can be prepared and maintained in separate files or combined into a single file. Figure 3: UAS Deployment Automation Workflow for a Single VNF, on page 23 depicts the configurations in separate files for illustration purposes.

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

Refer to AutoDeploy Configuration File Constructs for more Information on the constructs used in this file. You can also refer to RedHat user documentation for information on how to install the satellite server if your deployment requires:

https://access.redhat.com/documentation/en-US/Red_Hat_Network_Satellite/5.0/html/Installation_Guide/s1-intro-sat.html

Note These instructions assume you are already logged on to the AutoDeploy VM as the root user.

To prepare the VIM configuration file:

1. Create and edit your VIM Orchestrator configuration file according to your deployment requirements. Use the sample provided in as a reference.
2. Save the VIM configuration file, you have created to your home directory.
3. Proceed to Activate the VIM Deployment, on page 20.

Activate the VIM Deployment

Once you have completed preparing your VIM configuration file, you must load the configuration and activate the deployment in order to bring up the VIM and the Overcloud.

Note These instructions assume you are already logged on to the AutoDeploy VM as the root user and that your VIM configuration file has been prepared for your deployment as per the information and instructions in Prepare the VIM Configuration File, on page 19.

To activate the VIM installation using AutoDeploy:

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

   confd_cli -u admin -C

2. Enter the ConfD configuration mode.
   config

3. Load the VIM configuration file to provide the deployment artifacts.

   load merge <*your_vim_file_name*>.**cfg**
   commit
   end

Note If changes are made to the VIM Orchestrator configuration file after it was committed, you can apply the changes using the **load replace** command instead of the **load merge** command. You will also need to **commit** your changes.

4. Activate the deployment to begin creating the VMs on the VIM infrastructure.

   activate-vim-deployment service-deployment-id <*deployment-id*>

Note 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.

5. Monitor the progress of the deployment by:

   a. Viewing transaction logs:

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

      transaction_id is the ID displayed as a result of the **activate-deployment** command in step 4, on page 20.

      The logs display status messages for deployment, for example:

```
<config xmlns="http://tail-f.com/ns/config/1.0">
  <log xmlns="http://www.cisco.com/usp/nfv/usp-autodeploy-oper">
    <tx-id>1503545683619</tx-id>
    <log>Thu Aug 24 03:34:44 UTC 2017 [Task: 1503545683619] Started VIM deployment
VimDeploymentRequest [transType=ACTIVATE, serviceDeploymentId=north-east, siteList=[]]
Thu Aug 24 03:34:45 UTC 2017 [Task: 1503545683619/vnf1_vim] Starting Vim deployment
Thu Aug 24 03:34:45 UTC 2017 [Task: 1503545683619/vnf1_vim] Current VIM deployment
status is deployment-unknown
Thu Aug 24 03:34:45 UTC 2017 [Task: 1503545683619/vnf1_vim] Deploying VIM
Thu Aug 24 05:03:46 UTC 2017 [Task: 1503545683619/vnf1_vim] VIM deployed successfully
Thu Aug 24 05:03:46 UTC 2017 [Task: 1503545683619] Success
</log>
```

```
  </log>
</config>
```

b Checking the service deployment:

show service-deployment

Example command output:

```
siter auto-test-sjc
nfvi-popr nfvi-deployment-status "Stack vnf1_vim create completed"
nfvi-popr vim-orch status deployment-success
nfvi-popr vim-orch steps-total 84
nfvi-popr vim-orch steps-completed 84
nfvi-popr vim-orch version "Red Hat OpenStack Platform release 10.0 (Newton)"
nfvi-popr vim status deployment-success
nfvi-popr vim steps-total 16
nfvi-popr vim steps-completed 16
nfvi-popr vim version "Red Hat OpenStack Platform release 10.0 (Newton)"
                                                        FIRMWARE
                                              IS
NFVI NODE ID              UUID  STATUS  ROLE            VERSION   IP ADDRESS    BIOS
 VERSION                 ID        SIZE     JOURNAL  ID          PHYSNET ID
-------------------------------------------------------------------------------------
autoit-nfvi-physical-node  -     up      vim-orch        -         -             -

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

The deployment-status in the above output changes based on the current progress. The command can be re-issued multiple times to refresh the status.

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

6 *Optional.* Monitor the VIM Orchestrator deployment from within the AutoIT-NFVI ConfD configuration mode.

a Login to the ConfD CLI as the admin user.

confd_cli -u admin -C

b Enter the ConfD configuration mode.

config

c View the transaction log.

show log <*transaction_id*>**| display xml**
The transaction_id can be found by executing the **show transactions** command.

```
<config xmlns="http://tail-f.com/ns/config/1.0">
  <log xmlns="http://www.cisco.com/usp/nfv/usp-nfvi-oper">
    <tx-id>1503545685-984486</tx-id>
    <log>2017-08-24 03:34:46,017 - RPC triggered for VIM deployment: vnf1_vim,
deactivate: 0
2017-08-24 03:34:46,059 - Notify deployment of type 'deployment-started'
<--- SNIP --->
<--- SNIP --->
2017-08-24 05:02:57,935 - Overcloud deployed successfully.
2017-08-24 05:02:57,957 - VIM Transaction: 1503545685-984486 for deployment: vnf1_vim
 completed successfully.
2017-08-24 05:02:57,986 - Notify deployment of type 'deployment-completed'
</log>
  </log>
</config>
```

If there are any issues with your VIM deployment, refer to Monitoring AutoDeploy Operations and Monitoring AutoIT-VNF Operations for information on collecting logs.

Upon successful completion of the VIM deployment, proceed to VNF Deployment Automation, on page 22 for information and instructions on deploying your USP-based VNF.

VNF Deployment Automation

VNF Deployment Automation Overview

USP-based VNF deployment automation is performed through UAS.

Note This information in these sections assume that all of the USP Installation Prerequisites and/or that all requirements identified in the *Ultra M Solutions Guide* have been met.

Figure 3: UAS Deployment Automation Workflow for a Single VNF, on page 23 and Figure 4: UAS Deployment Automation Workflow for a Multi-VNF , on page 23 provide an overview of the VNF deployment automation process for Hyper-Converged Ultra M deployments. Details are provided in Table 3: VNF Deployment Automation Workflow Descriptions, on page 24.

Notes:

- The workflow described in this section is supported only with Ultra M deployments based on OSP 10 and that leverage the Hyper-Converged architecture.

- This information assumes that you have deployed the VIM Orchestrator and the VIM using the information and instructions in Virtual Infrastructure Manager Installation Automation, on page 1.

Figure 3: UAS Deployment Automation Workflow for a Single VNF

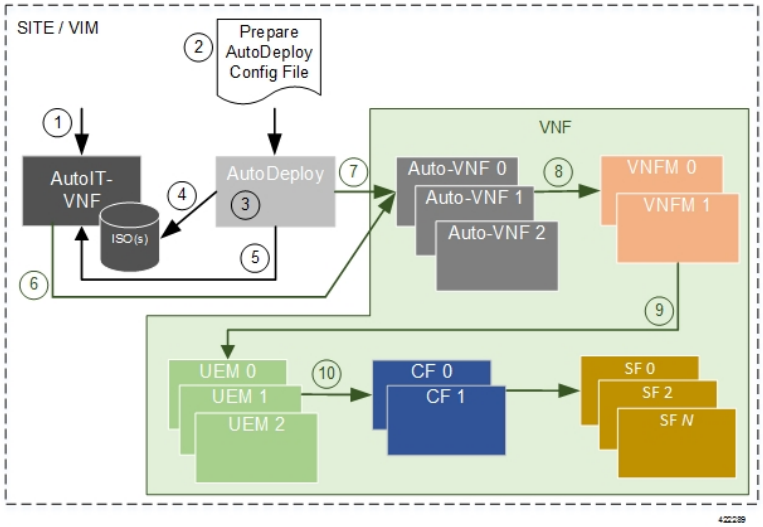


Figure 4: UAS Deployment Automation Workflow for a Multi-VNF

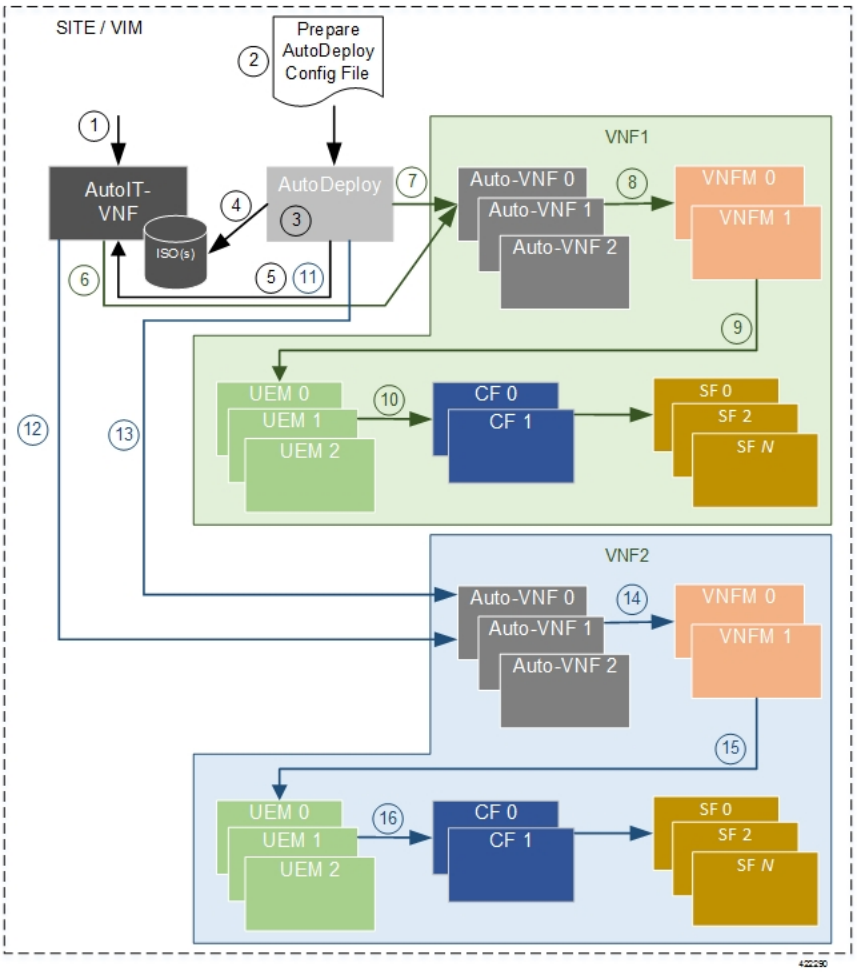


Table 3: VNF Deployment Automation Workflow Descriptions

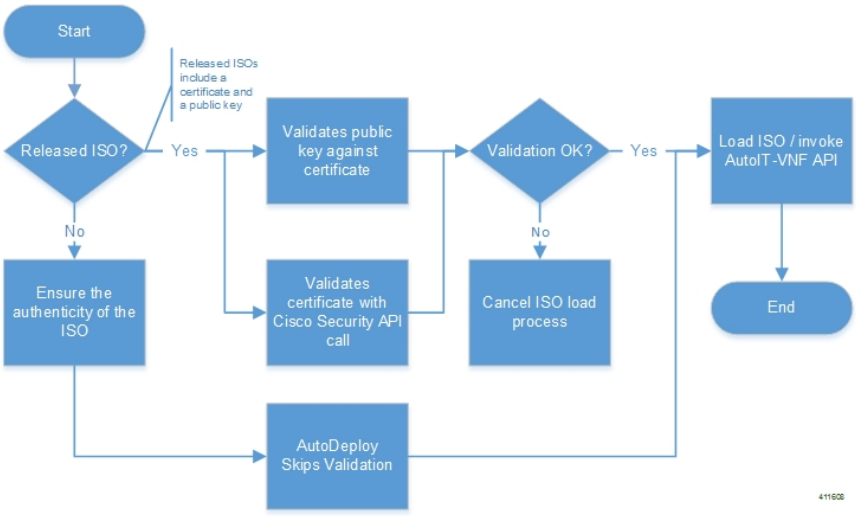
Callout	Description
1	From the Ultra M Manager Node, deploy AutoIT-VNF using the *auto-it-vnf-staging.sh* script provided as part of the release ISO. Refer to Deploy the AutoIT-VNF VM, on page 29 for more information.
2	Prepare the AutoDeploy configuration file that is used by AutoDeploy to activate the Site and initiate the VNF deployment process. This file includes all of the configuration information required to deploy not only the AutoVNF software module, but all of the VNF components (VNFCs) as well. It includes configurations for constructs such as secure tokens, VNF package images, host aggregates, VNFM descriptors (VNFMDs), and the VNF descriptors (VNFDs). Refer to Sample Ultra M AutoDeploy Configuration File **Note** If you're using UWS to deploy the USP-VNF process, refer to Sample Ultra M UWS Service Deployment Configuration File

Callout	Description
3	On the AutoDeploy VM, load and commit the configuration file prepared in previous step. Once commited, activate the previously loaded AutoDeploy configuration file. AutoDeploy processes this data to activate the Site and to deploy the functions needed to orchestrate the VNF deployment. Refer to Activate the USP-based VNF, on page 32 for more information.
4	AutoDeploy validates the ISO and then loads it on to AutoIT-VNF. Refer to Release Image Signing Validation Workflow, on page 26 for more information on ISO validation.
5	AutoDeploy passes data from the activated configuration to AutoIT-VNF requesting that it deploy the AutoVNF VM cluster for the initial VNF.
6, 12	AutoIT-VNF deploys the AutoVNF VM cluster for the VNF.
7, 13	Once the AutoVNF VM cluster is successfully deployed, AutoDeploy passes a configuration file to the AutoVNF which it uses to orchestrate the VNFM deployment.
8, 14	The leader (master) AutoVNF software module leverages the VNFMD information to work with the VIM to deploy the VNFM VMs. Once the VNFM VMs are successfully deployed, AutoVNF also 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. The 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).
9, 15	The VNFM leverages the VNFD 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.).
10, 16	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.
11	In multi-VNF deployments, AutoDeploy waits until it receives confirmation that all of the VNFCs have been onboarded successfully before it requests AutoIT-VNF to deploy the AutoVNF cluster for the next VNF.

Release Image Signing Validation Workflow

AutoDeploy and AutoIT-VNF validate image signatures using the certificate and public key provided as part of the release ISO. Figure 5: AutoDeploy Release Image Signature Validation Workflow, on page 26 illustrates the AutoDeploy release signing validation workflow. Figure 6: AutoIT-VNF Release Image Signature Validation Workflow, on page 27 depicts the workflow for AutoIT-VNF.

Figure 5: AutoDeploy Release Image Signature Validation Workflow



As shown in Figure 5: AutoDeploy Release Image Signature Validation Workflow, on page 26, certificate validation is performed through an API call to Cisco Security severs. As such, the Domain Name Service (DNS) must be configured on the Auto Deploy VM enabling it to connect to the Internet.

Status messages for the AutoDeploy validation workflow can be viewed by executing the **show log** *<transaction-id>* **| display xml** command.

The following is an example output:

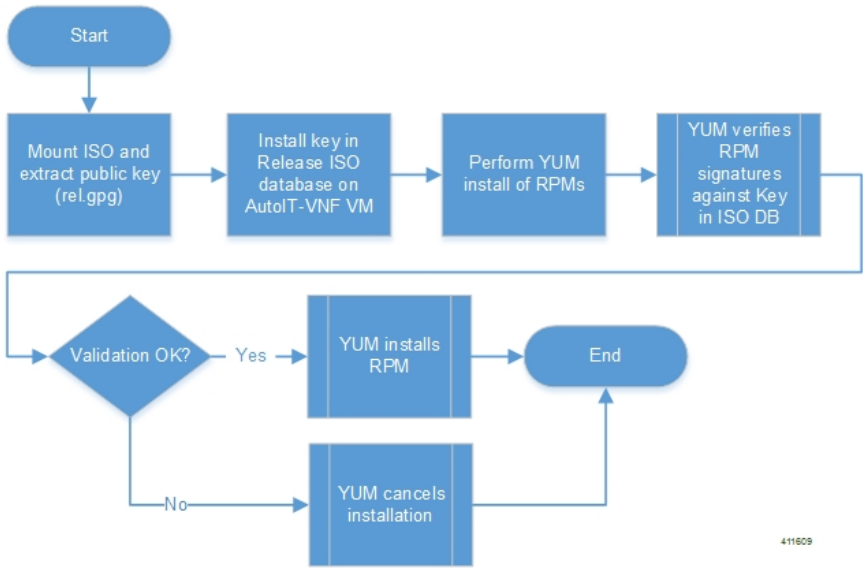
```
Fri May 19 00:31:47 UTC 2017 [Task: 1495153905253/vnf-pkg4] Uploading config file(s)
Fri May 19 00:31:47 UTC 2017 [Task: 1495153905253/vnf-pkg4] Uploading image file(s)
Fri May 19 00:31:47 UTC 2017 [Task: 1495153905253/vnf-pkg4] Validation of ISO called for
OS linux
Fri May 19 00:31:47 UTC 2017 [Task: 1495153905253/vnf-pkg4] Executing /tmp mount -t iso9660
 -o loop /home/ubuntu/isos/usp-5_1_0-631.iso /tmp/5061946078503935925
Fri May 19 00:31:47 UTC 2017 [Task: 1495153905253/vnf-pkg4] Command exited with return code:
 0
Fri May 19 00:31:47 UTC 2017 [Task: 1495153905253/vnf-pkg4] Executing . ls -lah
/tmp/5061946078503935925/repo
Fri May 19 00:31:47 UTC 2017 [Task: 1495153905253/vnf-pkg4] Command exited with return code:
 0
Fri May 19 00:31:47 UTC 2017 [Task: 1495153905253/vnf-pkg4] Executing . python
/opt/cisco/signing/cisco_openpgp_verify_release.py -e
/tmp/5061946078503935925/repo/USP_RPM_CODE_REL_KEY-CCO_RELEASE.cer -G
/tmp/5061946078503935925/repo/rel.gpg
Fri May 19 00:31:47 UTC 2017 [Task: 1495153905253/vnf-pkg4] Command exited with return code:
 0
Fri May 19 00:31:47 UTC 2017 [Task: 1495153905253/vnf-pkg4] ISO validation successful
Fri May 19 00:31:47 UTC 2017 [Task: 1495153905253/vnf-pkg4] Executing . umount
/tmp/5061946078503935925
Fri May 19 00:31:47 UTC 2017 [Task: 1495153905253/vnf-pkg4] Command exited with return code:
 0
```

```
Fri May 19 00:31:47 UTC 2017 [Task: 1495153905253/vnf-pkg4] Executing . rm -r
/tmp/5061946078503935925
Fri May 19 00:31:47 UTC 2017 [Task: 1495153905253/vnf-pkg4] Command exited with return code:
 0
Fri May 19 00:31:47 UTC 2017 [Task: 1495153905253/vnf-pkg4] Uploading ISO file
```

The status is also viewable through the AutoDeploy upstart logs */var/log/upstart/autodeploy.log*:

```
2017-05-19T00:31:47,056 DEBUG [VnfPackageTask:227] [pool-5-thread-2] This ISO has rel.gpg,
 will continue with validation
2017-05-19T00:31:47,057 DEBUG [Task:52] [pool-5-thread-2] Executing . python
/opt/cisco/signing/cisco_openpgp_verify_release.py -e
/tmp/5061946078503935925/repo/USP_RPM_CODE_REL_KEY-CCO_RELEASE.cer -G
/tmp/5061946078503935925/repo/rel.gpg
2017-05-19T00:31:47,562 DEBUG [VnfPackageTask:299] [pool-5-thread-2] Output: ^[[92mDownloading
 CA certificate from http://www.cisco.com/security/pki/certs/crcam2.cer ...^[[0m
2017-05-19T00:31:47,563 DEBUG [VnfPackageTask:299] [pool-5-thread-2] Output:
^[[92mSuccessfully downloaded crcam2.cer.^[[0m
2017-05-19T00:31:47,563 DEBUG [VnfPackageTask:299] [pool-5-thread-2] Output: ^[[92mDownloading
 SubCA certificate from http://www.cisco.com/security/pki/certs/innerspace.cer ...^[[0m
2017-05-19T00:31:47,564 DEBUG [VnfPackageTask:299] [pool-5-thread-2] Output:
^[[92mSuccessfully downloaded innerspace.cer.^[[0m
2017-05-19T00:31:47,565 DEBUG [VnfPackageTask:299] [pool-5-thread-2] Output:
^[[92mSuccessfully verified Cisco root, subca and end-entity certificate chain.^[[0m
2017-05-19T00:31:47,565 DEBUG [VnfPackageTask:299] [pool-5-thread-2] Output:
^[[92mSuccessfully fetched a public key from
/tmp/5061946078503935925/repo/USP_RPM_CODE_REL_KEY-CCO_RELEASE.cer.^[[0m
2017-05-19T00:31:47,565 DEBUG [VnfPackageTask:299] [pool-5-thread-2] Output:
^[[92mSuccessfully authenticated /tmp/5061946078503935925/repo/rel.gpg key using Cisco X.509
 certificate trust chain.^[[0m
2017-05-19T00:31:47,565 DEBUG [Task:52] [pool-5-thread-2] Command exited with return code:
 0
2017-05-19T00:31:47,566 DEBUG [Task:52] [pool-5-thread-2] ISO validation successful
2017-05-19T00:31:47,567 DEBUG [Task:52] [pool-5-thread-2] Executing . umount
/tmp/5061946078503935925
2017-05-19T00:31:47,583 DEBUG [Task:52] [pool-5-thread-2] Command exited with return code:
 0
2017-05-19T00:31:47,583 DEBUG [Task:52] [pool-5-thread-2] Executing . rm -r
/tmp/5061946078503935925
2017-05-19T00:31:47,585 DEBUG [Task:52] [pool-5-thread-2] Command exited with return code:
 0
```

Figure 6: AutoIT-VNF Release Image Signature Validation Workflow



AutoIT logs display whether or not the public key was extracted and installed into the ISO database. The logs are located in */var/log/cisco/usp/auto-it/autoit.log*.

The following is an example output:

```
2017-05-19 00:33:06,868 - INFO: Mounting ISO to /tmp/tmpRfy_rf/iso_mount
2017-05-19 00:33:06,914 - INFO: Installing GPG key '/tmp/tmpRfy_rf/iso_mount/repo/rel.gpg'
2017-05-19 00:33:07,278 - INFO: Installing ISO
2017-05-19 00:35:37,762 - INFO: Unmouting /tmp/tmpRfy_rf/iso_mount
2017-05-19 00:35:37,821 - INFO: ISO successfully loaded
```

Additionally, though the automation workflow handles the public key and RPM validation as described above, you can view the public key when logged on to the AutoIT-VNF VM by entering the following commands:

sudo su uspadmin
rpm --root /opt/cisco/usp/namespaces/5.1.0-631/.chroot_base/ -q gpg-pubkey

The above command displays the public key.

rpm --root /opt/cisco/usp/namespaces/<*version*>**/.chroot_base/ -K**
/opt/cisco/usp/isos/<*version*>**/repo/usp-auto-it-bundle-**<*bundle_version*>**.x86_64.rpm**

The above command displays whether or not the RPM signature matched the key ('OK').

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

Deploy the USP-based VNF

The software modules that comprise the Ultra Automation Services (UAS) are used to automate the USP-based VNF deployment. UAS is designed to support deployment automation for all USP-based VNF scenarios.

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

Configure VIM Tenants

VIM tenant configuration is performed by executing an API which consumes a separate configuration file called *addi-tenant.cfg*. A sample configuration file is in Sample addi-tenant.cfg File.

Note These instructions assume you are already logged on to the AutoDeploy VM as the root user.

To configure VIM tenants:

1. Prepare the *addi-tenant.py* according to your deployment scenario.
2. Login to the ConfD CLI as the *admin* user.

   confd_cli -u admin -C
3. Enter the ConfD configuration mode.

   config
4. Load the *addi-tenant.cfg* configuration file.

   load merge *<your_tenant_file_name>***.cfg**
   commit
   end
5. Create the tenants.

   create-tenant service-deployment-id *<service_deployment_id>* **site { tenant { tenant-name** *<tenant_attribute_name>* **} site-id** *<site_id>* **}**
6. Monitor the progress of the tenant creation by viewing transaction logs:

   show logs *<transaction_id>* **log |display xml**
   transaction_id is the ID displayed as a result of the **create-tenant** command executed in step 5, on page 29.
7. Verify that the tenants have been created properly through OpenStack Horizon.
8. Proceed to Deploy the AutoIT-VNF VM, on page 29.

Deploy the AutoIT-VNF VM

The VM for AutoIT-VNF is deployed using *uas-boot.py* script provided with the UAS bundle. The script is located in the following directory:

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

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:

./uas-boot.py -h
Additionally, the help information is provided as an appendix in this document. Refer to uas-boot.py Help.

The VM deployment parameters and their settings are stored in the auto-it-vnf-staging.sh file which executes *uas-boot.py* with the appropriate arguments.

This script contains the settings for the AutoIT-VNF VM deployment. Two deployment options are available based on how you would like to allocate an IP address to the AutoIT-VNF VM:

- Dynamically assigned floating IP address from OpenStack. An example of this file is below:

  cat auto-it-vnf-staging.sh

```
#!/bin/sh
/opt/cisco/usp/uas-installer/scripts/uas-boot.py \
    staging-server auto-it-vnf-ISO-590 \
    --image-loc /opt/cisco/usp/uas-installer/images/usp-uas-1.0.0-601.qcow2 \
    --mgmt-net cf-mgmt \
    --credential-file account.cfg \
    --availability-zone mgmt\
    --flavor auto-deploy-flavor \
    --public-net public \
    --floating-ip \
    $*
```

- Statically assigned floating IP address via the deploy script providing a fixed IP address for the VM. An example of this file is below:

  cat auto-it-vnf-staging.sh

```
#!/bin/sh
/opt/cisco/usp/uas-installer/scripts/uas-boot.py \
    staging-server auto-it-vnf-ISO-590 \
    --image-loc /opt/cisco/usp/uas-installer/images/usp-uas-1.0.0-601.qcow2 \
    --mgmt-net cf-mgmt \
    --credential-file account.cfg \
    --availability-zone mgmt\
    --flavor auto-deploy-flavor \
    --public-net public \
    --floating-ip <desired_ip_address> \
    $*
```

Sample files are provided for convenience. They are located at /opt/cisco/usp/uas-installer/scripts and can be edited to your requirements as per the instructions below.

Note These instructions assume you are already logged on to the Ultra M Manager Node.

To deploy the AutoIT-VNF VM:

1. Navigate to the directory containing the *auto-it-vnf-staging.sh* file.

   cd /opt/cisco/usp/uas-installer/scripts
2. Make edits to the *auto-it-vnf-staging.sh* file as needed for your deployment scenario.

   vi auto-it-vnf-staging.sh
   Minimally, you should verify that the image name and location match your deployment scenario.
3. Deploy the AutoIT-VNF VM.

   ./auto-it-vnf-staging.sh
4. Log on to the AutoDeploy VM as *ubuntu*. Use the password that was created earlier for this user.
5. Become the root user.

   sudo -i
6. Configure the Domain Name Service (DNS) to enable Internet connectivity to Cisco Security servers in order to validate the ISO/package certificate as described in Release Image Signing Validation Workflow, on page 26.

Note If AutoDeploy was previously deployed as part of the NFVI deployment automation process as documented in the *Ultra M Solutions Guide*, this step is not required.

To configure DNS:

a Edit the following parameters in /etc/resolve.conf:

vi /etc/resolve.conf
nameserver <*DNS-IP-ADDRESS*>
search <*DOMAIN-NAME*>

b Edit the following parameters in /etc/network/interfaces.d/eth0.cfg:

vi /etc/network/interfaces.d/eth0.cfg
auto eth0
iface eth0 inet dhcp
dns-nameservers <*DNS-IP-ADDRESS*>
dns-search <*DOMAIN-NAME*>

7 Download the USP ISO bundle and related files pertaining to the release.

You will need to record the location of the ISO as it is required when preparing the AutoDeploy configuration file.

8 Choose the desired method by which to continue the deployment process:

- Use Ultra Web Services (UWS) to continue the deployment process. To use this method, proceed to Prepare the UWS Service Deployment Configuration File, on page 31.
- Use the ConfD CLI/APIs to continue the deployment process. To use this method, proceed to Prepare the AutoDeploy Configuration File, on page 32.

Prepare the UWS Service Deployment Configuration File

If you will be using the UWS GUI to complete the deployment process, you will need to prepare the service deployment configuration file. This is an XML file which is uploaded to the UWS in order to initiate the VNF deployment process.

This file includes all of the configuration information required to deploy not only the AutoVNF software module, but all of the VNF components (VNFCs) as well. It includes configurations for constructs such as secure tokens, VNF package images, host aggregates, VNFM descriptors (VNFMDs), and the VNF descriptors (VNFDs).

A sample configuration file is provided for reference in Sample Ultra M UWS Service Deployment Configuration File.

Information on the constructs used in this file are provided in AutoDeploy Configuration File Constructs.

Note These instructions assume you are already logged on to the AutoDeploy VM as the *root* user.

To prepare the service deployment configuration file:

1 Create and edit your service deployment configuration file using the XML editing tool of your choice according to your VNF deployment requirements. Use the sample provided in Sample Ultra M UWS Service Deployment Configuration File as a reference.

2 Copy your configuration file to the AutoDeploy VM.

3 Enter the following URL in a supported web browser:

https://<auto-deploy-vm-floating-ip-address>:8443/

Note Refer to Ultra Web Services for a list of supported web browsers.

4 Refer to the UWS online help for information on how to upload the service deployment configuration file and for further instructions on how to activate it.

Prepare the AutoDeploy Configuration File

As described in VNF Deployment Automation Overview, on page 22, the AutoDeploy configuration file is used by AutoDeploy to activate the Site (VIM instance) and initiate the VNF deployment process.

This file includes all of the configuration information required to deploy not only the AutoVNF software module, but all of the VNF components (VNFCs) as well. It includes configurations for constructs such as secure tokens, VNF package images, host aggregates, VNFM descriptors (VNFMDs), and the VNF descriptors (VNFDs).

A sample configuration file is provided for reference in Sample Ultra M AutoDeploy Configuration File.

Information on the constructs used in this file are provided in AutoDeploy Configuration File Constructs.

Note These instructions assume you are already logged on to the AutoDeploy VM as the *root* user.

To prepare the AutoDeploy configuration file:

1 Create and edit your AutoDeploy configuration file according to your VNF deployment requirements. Use the sample provided in Sample Ultra M AutoDeploy Configuration File as a reference.

2 Save the AutoDeploy configuration file you have created to your home directory.

3 Proceed to Activate the USP-based VNF, on page 32.

Activate the USP-based VNF

Once you have completed preparing your AutoDeploy configuration file, you must load the configuration and activate the deployment. Once activated, AutoDeploy proceeds with the deployment automation workflow as described in VNF Deployment Automation Overview, on page 22.

Note These instructions assume you are already logged on to the AutoDeploy VM as the *root* user and that your AutoDeploy configuration file has been prepared for your deployment as per the information and instructions in Prepare the AutoDeploy Configuration File, on page 32. These instructions also assume that AutoDeploy has access to the USP ISO either locally or on a remote server.

To activate the USP deployment using AutoDeploy:

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

confd_cli -u admin -C

2 Enter the ConfD configuration mode.

config

3 Load the AutoDeploy configuration file to provide the deployment artifacts to the VIM.

load merge <*your_ad_file_name*>**.cfg**
commit
end

Note 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_ad_file_name*>**.cfg**
commit
end

4 Activate the deployment to begin creating the VMs on the VIM infrastructure:

activate-deployment service-deployment-id <*deployment-id*>

Note The above command syntax assumes a first-time deployment scenario. Other syntax options are available for use during upgrade/redeployment operations. Refer to Deactivate/Activate Operations for more information.

The output of this command is a transaction-id which can be used to monitor the deployment progress.

5 Monitor the progress of the deployment by viewing transaction logs:

show logs <*transaction_id*> **log |display xml**
transaction_id is the ID displayed as a result of the **activate-deployment** command in step 4, on page 33 .

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 AutoDeploy are shown below:

- Host Aggregate:

```
Fri May 12 21:31:44 UTC 2017 [Task:
1494624612779/tb1-sjc-vnf2-rack-tb1-sjc-cf-esc-mgmt2] Created Host Aggregate
successfully.
```

- AutoVNF:

```
Fri May 12 21:34:08 UTC 2017 [Task: 1494624612779/tb1autovnf2] Successfully deployed
 AutoVnf tb1autovnf2 with floating-ip 172.21.201.250.
```

- VNFM:

```
Fri May 12 21:39:09 UTC 2017 [Task: 1494624612779/ab-tb1-vnfm2] VNFM deployed
successfully
```

- 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
```

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