Installing Cisco VTS on OpenStack

The following sections provide details about installing VTS on a Linux-OpenStack environment. Ensure that you review the Prerequisites chapter, before you begin installing VTS.

Installing Cisco VTS in a Linux—OpenStack Environment

Installing Cisco VTS in an OpenStack environment involves:

- Installing the VTC VM. See Installing the VTC VM, on page 1 for details.
- Installing the Host Agent and the Open Stack Neutron Plugin.

  See Installing Host Agent, on page 8 and Registering OpenStack VMM, on page 7

Installing the VTC VM

You can install the VTC VM using either the automatic or manual configuration option.

To install the VTC VM using an ISO file (Auto Configuration), see Installing VTC VM—Automatic Configuration Using ISO File, on page 2

To install VTC VM using the virt-manager application (Manual Configuration), see Installing VTC VM—Manual Configuration Using virt-manager Application, on page 4

To install VTC VM using VNC (Manual Configuration), see Installing VTC VM - Manual Configuration using VNC, on page 5

Note If you need to access the VTC VM's console using virt-manager, VNC, or SPICE, it may be necessary to manually switch to tty1 using the *CTRL+ALT+F1* key combination. After connecting to the VM's console, if the output shows a blank screen, then you must manually switch to tty1.

Installing VTC VM—Automatic Configuration Using ISO File

To enable configuration using ISO file, the administrator needs to create a text file with the VM settings, wrap it into an ISO file, and then attach the ISO to the VM's CD drive.

Step 1 Connect to the controller node via SSH, and copy the vtc.qcow2 file to /var/lib/libvirt/images/ folder.

Step 2 Copy the vtc.sample.xml file to your controller. A sample XML file is available at Sample XML File—VTC Installation.

Step 3 Create a file called config.txt. The contents of the file is given in the below example:

Note Note: Underlay IPv6 is not supported for VTSR in Cisco VTS 2.5.2.

```
Hostname=vtc
    ManagementIPv4Method=Static
    ManagementIPv4Address=1.1.1.2
    ManagementIPv4Netmask=255.255.255.0
    ManagementIPv4Gateway=1.1.1.1
    ManagementIPv6Method=Static
    ManagementIPv6Address=1::2
    ManagementIPv6Netmask=64
    ManagementIPv6Gateway=1::1
    UnderlayIPv4Method=Static
    UnderlayIPv4Address=2.2.2.2
    UnderlayIPv4Netmask=255.255.255.0
    UnderlayIPv4Gateway=2.2.2.1
    UnderlayIPv6Method=Static
    UnderlayIPv6Address=2::2
    UnderlayIPv6Netmask=64
    UnderlayIPv6Gateway=2::1
    DNSv4=3.3.3.3
    DNSv6=3::3
    Domain=cisco.com
    NTP=1.1.1.1
    vts-adminPassword=cisco123
    AdministrativeUser=admin
    AdministrativePassword=cisco123
```

Note

- Cisco VTS follows the restrictions on valid hostnames as specified in RFC 952 and RFC 1123, which states that the valid characters are *a* to *z*, *A* to *Z*, *0* to *9*, and -. Each label can be from 1 to 63 characters long, and the entire hostname can have a maximum of 253 ASCII characters.
- The *config.txt file* must have a blank line at the end.
- If you are using IPv6, all parameters are required. If you are not using IPv6, you need not specify the following parameters:
  - ManagementIPv6Address
  - ManagementIPv6Netmask
  - ManagementIPv6Gateway
  - UnderlayIPv6Address
  - UnderlayIPv6Netmask
  - UnderlayIPv6Gateway
  - DNSv6

In this file:

- Hostname—The hostname of the VM
- ManagementPv4Method—Whether to use DHCP, Static, or None IPv4 addressing for the management interface (eth0)
- ManagementIPv4Address—Management IPv4 address of the VM (required only for static addressing)
- ManagementIPv4Netmask—Management IPv4 netmask of the VM (required only for static addressing)
- ManagementIPv4Gateway—Management IPv4 gateway of the VM (required only for static addressing)
- ManagementPv6Method—Whether to use DHCP, Static, SLAAC, or None IPv6 addressing for the management interface (eth0)
- ManagementIPv6Address—Management IPv6 address of the VM (required only for static addressing)
- ManagementIPv6Netmask—Management IPv6 netmask of the VM (required only for static addressing)
- ManagementIPv6Gateway—Management IPv6 gateway of the VM (required only for static addressing)
- UnderlayPv4Method—Whether to use DHCP, Static, or None IPv4 addressing for the underlay interface (eth1)
- UnderlayIPv4Address—Underlay IPv4 address of the VM (required only for static addressing)
- UnderlayIPv4Netmask—Underlay IPv4 netmask of the VM (required only for static addressing)
- UnderlayIPv4Gateway—Underlay IPv4 gateway of the VM (required only for static addressing)
- UnderlayPv6Method—Whether to use DHCP, Static, SLAAC, or None IPv6 addressing for the underlay interface (eth1)
- UnderlayIPv6Address—Underlay IPv6 address of the VM (required only for static addressing)
- UnderlayIPv6Netmask—Underlay IPv6 netmask of the VM (required only for static addressing)
- UnderlayIPv6Gateway—Underlay IPv6 gateway of the VM (required only for static addressing)

- DNSv4—DNS IPv4 address (required only for static addressing or if DHCP does not send the option) and may contain multiple entries if enclosed in double quotes (")
- DNSv6—DNS IPv6 address (required only for static and SLAAC addressing or if DHCP does not send the option) and may contain multiple entries if enclosed in double quotes (")
- Domain—DNS search domain (required only for static addressing or if DHCP does not send the option)
- NTP—NTP IPv4 address, IPv6 address, or FQDN (required only for static addressing or if DHCP does not send the option)
- vts-adminPassword—Password for the vts-admin user
- AdministrativeUser—New administrative user for login via SSH
- AdministrativePassword—Password for the new administrative user

Step 4 Use mkisofs to create an ISO file. For example:
```
mkisofs -o config.iso config.txt
```
Step 5 Create the VTC VM using following command:
```
virsh create vtc.sample.xml
```

Installing VTC VM—Manual Configuration Using virt-manager Application

To install the VTC VM, configuring the VM, manually, using the virt-manager application:

Step 1 Connect to the controller node via SSH, and copy the vtc.qcow2 file to /var/lib/libvirt/images/ folder.
Step 2 Copy the vtc.sample.xml file to your controller. Modify it as per your setup.
Step 3 Create the VTC VM using following command:
```
virsh create vtc.sample.xml
```
Step 4 Run the command:
```
virsh list --all
```
It should display:
```
Id     Name     State
----------------------------------------------------
2 VTC running
```
Step 5 Start virt-manager. Run:
```
virt-manager
```
Step 6 Once virt-manager window opens, click on the VTC VM to open up the VTC VM console.
In the console you get the installation wizard which takes you through the steps to configure VTC VM for the first time.
Step 7 Enter the following:
Note For items that take multiple values, such as DNS and NTP, each value must be separated by a space.

- VTS Hostname

- DHCP/Static IP configuration for static IP
- Management IP address for VTC—This is the management IP address.
- Management IP Netmask
- Management Gateway address
- DNS Address
- DNS Search domain
- Underlay IP address—This is the IP address for internal network.
- Underlay IP Netmask
- Underlay IP Gateway
- NTP address—Can be same as gateway IP address.
- Password change for user vts-admin—Enter the default user vts-admin password. The vts-admin user is used for password recovery and to revisit a configuration screen if you make a mistake or need to change the information. If you log in to the VTC VM using vts-admin username and password again, you will get the same dialog to go through the VTC VM setup again.
- Administrator User—Enter administrative username and password. This username and password are used to login to the VM via SSH.
- Password for administrator user

VTC VM reboots at this time. Wait for two minutes for the VTC VM to be up. You can ping the IP address given for VTC VM in the setup process to verify whether the VTC VM is up.

Step 8 SSH into VTC VM using the IP address, administrative username/password given in the setup process (not vts-admin user).

Installing VTC VM - Manual Configuration using VNC

If the server where VTC is to be installed resides on a remote location with network latency or low bandwidth, you may want to opt for the use of VNC in order to gain graphical console access to the VTC VM, and manually configure the VM. To do this:

Step 1 Connect to the controller node via SSH, and copy the vtc.qcow2 file to /var/lib/libvirt/images/ folder.

Step 2 Copy the vtc.sample.xml file to your controller. Modify it as per your setup. A sample XML file is available at Sample XML File—VTC Installation.

Step 3 Replace the following sections of the vtc.sample.xml file:

```
<graphics type='spice' port='5900' autoport='yes' listen='127.0.0.1'>
     <listen type='address' address='127.0.0.1'/>
   </graphics>
```

with the following:

```
<graphics type='vnc' port='5900' autoport='yes' listen='0.0.0.0'>
      <listen type='address' address='0.0.0.0'/>
    </graphics>
```

Note Setting the listen address to 0.0.0.0 allows external clients to connect to the VNC port (5900). You will also need to make sure that iptables configuration (if any) allows inbound TCP port 5900 connections.

Step 4 Create the VTC VM using following command:

```
virsh create vtc.sample.xml
```

You should now be able to use a VNC client to connect to the graphics console of the VTC VM to continue with the setup process.

Step 5 Enter the following:

Note For items that take multiple values, such as DNS and NTP, each value must be separated by a space.

- VTS Hostname
- DHCP / Static IP configuration for static IP
- Management IP address for VTC—This is the management IP address.
- Management IP Netmask
- Management Gateway address
- DNS Address
- DNS Search domain
- Underlay IP address—This is the IP address for internal network.
- Underlay IP Netmask
- Underlay IP Gateway
- NTP address—Can be same as gateway IP address.
- Password change for user vts-admin—Enter the default user vts-admin password. The vts-admin user is used for password recovery and to revisit a configuration screen if you make a mistake or need to change the information. If you log in to the VTC VM using vts-admin username and password again, you will get the same dialog to go through the VTC VM setup again.
- Administrator User—Enter administrative username and password. This username and password are used to login to the VM via SSH.
- Password for administrator user

VTC VM reboots at this time. Wait for two minutes for the VTC VM to be up. You can ping the IP address given for VTC VM in the setup process to verify whether the VTC VM is up.

Step 6 SSH into VTC VM using the IP address, administrative username/password given in the setup process (not vts-admin user).

Installing OpenStack Plugin

The OpenStack plugin gets installed when you register the VMM from the Cisco VTS GUI. See Registering OpenStack VMM, on page 7, for details.

This is applicable when Admin has selected **Yes** to the Question "Do you want VTS to install components?", in VMM Page of Cisco VTS UI. If the admin selected **No** then plugin is not installed, and the installation of plugin needs to be done manually on OpenStack Controllers.

Registering OpenStack VMM

You can register the OpenStack VMM using the Cisco VTS GUI.

If you opt for the guided set up using the Setup wizard, VMM registration is done as part of the wizard flow. See the *Using the Setup Wizard* section in the *Getting Started with Cisco Virtual Topology System* chapter in the *Cisco VTS User Guide* for details.

If you are not using the Setup wizard, you can register the VMM using the **Administration > Virtual Machine Manager** UI.

Note If you install an unsupported OpenStack plugin version, you might encounter errors after installation. We recommend that you review the Supported Virtual Machine Managers section before you install the OpenStack plugin.

Step 1 Go to **Administration** > **Virtual Machine Manager**.

Step 2 Click the **Add (+)** button.
The Register VMM page is displayed.

Step 3 Enter the VMM Details:

- Name—Name of the VMM.
- Version —Specify the version from the drop-down. If you choose openstack-newton as the Version in the **Version** drop-down, it displays a question "Do you want VTS to install VMM plugin components?".

  If you choose **No**, enter the VMM ID. You can enter the VMM ID present in the file */etc/neutron/plugins/ml2/ml2_conf.ini* in the controller machine. By default, **Yes** is chosen.
- Mode—Whether the VMM has been registered as Trusted or Untrusted.
- API Endpoint Details—The fields differ based on the VMM you choose.
  - API Endpoint Details for OpenStack
    - API Protocol:IP Address:Port—VMM service endpoint's IPv4/IP6 address and port. Make sure you use the same IP address format (IPv4/IPv6) for all IP address fields. Mixed mode is not supported.
    - Keystone Protocol:IP Address:Port—Keystone protocol, IP address and port for OpenStack.

- Openstack Admin Project—Tenant with Administrator privileges in OpenStack. This can be any tenant with Administrator privileges. Any change to this tenant name, username, and passphrase needs to be updated in Cisco VTS for Multi-VMM operations to work properly.
- Admin User Name—admin user for the admin project in OpenStack.
- Admin Passphrase—Password of the admin user.

Step 4 Click **Register**.
After the VMM is registered successfully, the Plugin sections open up.

Step 5 **For OpenStack:**

Note If you choose **No** for the question 'Do you want VTS to install VMM plugin components?' in VMM Details, the radio button mentioned in **a**) is not displayed. It has only the Neutron Server section. The Add Neutron Server popup has the username and password as optional entries. You can choose not to give those. In that case Cisco VTS only saves the IP address. If you enter the Neutron server details you get an option to Save and Validate the plugin installation.

a) Select the desired radio button to specify whether you want to Install plug in with Red Hat OSP Director or not. If you select Yes, enter the following details:
   - OSP Director IP Address
   - OSP Director User name
   - OSP Director Passphrase
b) Click **Save**. The Neutron Servers section opens up.
c) Click **Add (+)** to add a Neutron Server. The Add Neutron Server popup is displayed.
d) Enter the Server IP Address and the Server User Name.
e) Click **Save** and Install Plugin. You may add more Neutron Servers using the **Add (+)** option, if you have multiple controllers (HA Mode). The Server Plugin Installation status shows whether the installation was a success.
   Note If you had opted not to use OSP Director, you need to enter the password for the Neutron servers while adding the servers.
   In case the Plugin Installation Status in the Virtual Machine Manager page shows the failure icon, you may choose to edit the VMM using the Edit option and rectify the error. Click the **Server Plugin Status** icon to view details of the error.

Installing Host Agent

You can use the Host Agent while specifying the Virtual Switch type, in Host Inventory.

Note After the installation of the Host Agent if neutron-vts-agent service is down on the compute host, check whether the compute host has Python module pycrypto installed. If it does not exist, install this module and restart the neutron-vts-agent.

Step 1 Go to **Inventory** > **Host Inventory**. The Inventory / Host Inventory page appears. The Host Inventory page has two tabs—**Virtual Servers** and **Baremetals**. By default, the page displays Virtual Server details.

Step 2 To view host details on Virtual Servers, select the VMM from the Select VMM drop-down, and select the device from the Select Device drop-down list. The following details are displayed:

- Host Name
- IP Address
- Host Type
- Associated VMM
- Virtual Switch
- Interfaces
- Installation Status—Shows the installation status.
- VTF Mode—Displayed on the top left of the table shows the VTF mode you have chosen in the Administration > System Settings window.

Step 3 Enter the following host details, while adding a new host or while editing the host:

- Host Name—This is mandatory. Only letters, numbers, underscore and dashes are allowed. Requires at lease one letter or number.
- Host Interface—IPv4/IPv6 address of the host. This is mandatory.
- Host IP Address
- Device Port Name
- User Name
- Passphrase
- Host Configuration
  - VMM ID—The VMM ID of the VMM to which you want to associate the host to.
  - Virtual Switch—Select **ovs**, then check the **Install VTS agent on save** check box.

Step 4 Click **Save**.
After the installation is complete you can see the green check button under Installation Status.

Note This is applicable when Admin has selected **Yes** to the Question "Do you want VTS to install components?", in VMM Page of VTS UI. If the admin had selected **No** then host agent is not installed, and the installation of host agent needs to be done manually on computes.

Step 5 Specify the physnet type. This is mandatory. You can find this using ovs bridge #sudo ovs-vsctl show | more . By default, it is *tenant*.

Step 6 Log in to the compute and check the service is up and running.

```
# sudo service neutron-vts-agent status
```

Installing Host Agent on Newton OSPD using CLI

Step 1 Log in to VTC

Step 2 Go to *cd /opt/vts/lib/ansible/playbooks*

Step 3 Create SAMPLE_INVENTORY_OVS.

Step 4 Install SSH keys. See Setting Up Ansible Install Through an SSH Proxy (for RHEL OpenStack Platform Director), on page 13.

```
sudo ansible-playbook ssh_proxy.yaml -i SAMPLE_INVENTORY_OVS -e ACTION=install -l proxy
```

Step 5 Install OVS Host Agent on compute.

```
sudo ansible-playbook neutron-compute.yaml -i SAMPLE_INVENTORY_OVS -e ACTION=install
```

SAMPLE_INVENTORY_OVS for Newton OSPD with IPV4

```
[proxy]

director ansible_ssh_host=172:20:200:18



# SSH Proxy access parameters. Do not modify the group name

[proxy:vars]

ansible_connection=ssh

ansible_port=22

ansible_ssh_user=stack

ansible_ssh_pass=<password>



[proxied_hosts:children]

vts_p_hosts_ovs



# Host specific variable for P hosts with VTS OVS agent

[vts_p_hosts_ovs]
```

```
overcloud-controller-0 ansible_ssh_host=172.20.200.5

overcloud-controller-1 ansible_ssh_host=172.20.200.4

overcloud-compute-2 ansible_ssh_host=172:20:200:27


# Group variables for P hosts with VTS OVS agent

[vts_p_hosts_ovs:vars]

ansible_connection=ssh

ansible_port=22

ansible_ssh_user=heat-admin

ansible_ssh_pass=<password>

VMM_NAME=OSPD_Newton


# Common group variables

[all:vars]

VTS_IP=172:20:200:20

VTS_USERNAME=admin

VTS_PASSWORD=<password>


[defaults]

timeout=60


[ssh_connection]

ssh_args="-C -o ControlMaster=auto -o ControlPersist=600s -o GSSAPIAuthentication=no -o
UserKnownHostsFile=/dev/null -o StrictHostKeyChecking=no"
```

SAMPLE_INVENTORY_OVS for Newton OSPD with IPV6

```
[proxy]

director ansible_ssh_host=2001:420:10e:2010:172:20:100:18

# SSH Proxy access parameters. Do not modify the group name
```

```
[proxy:vars]

ansible_connection=ssh

ansible_port=22

ansible_ssh_user=stack

ansible_ssh_pass=cisco123

[proxied_hosts:children]

vts_p_hosts_ovs

# Host specific variable for P hosts with VTS OVS agent

[vts_p_hosts_ovs]

fd41:4c47:94d7:c790:172:23:92:18 ansible_ssh_host=fd41:4c47:94d7:c790:172:23:92:18
ansible_ssh_user=admin
fd41:4c47:94d7:c790:172:23:92:17 ansible_ssh_host=fd41:4c47:94d7:c790:172:23:92:17
ansible_ssh_user=admin
fd41:4c47:94d7:c790:172:23:92:43 ansible_ssh_host=fd41:4c47:94d7:c790:172:23:92:43
ansible_ssh_user=admin
fd41:4c47:94d7:c790:172:23:92:44 ansible_ssh_host=fd41:4c47:94d7:c790:172:23:92:44
ansible_ssh_user=admin
fd41:4c47:94d7:c790:172:23:92:45 ansible_ssh_host=fd41:4c47:94d7:c790:172:23:92:45
ansible_ssh_user=admin


# Group variables for P hosts with VTS OVS agent

[vts_p_hosts_ovs:vars]

ansible_connection=ssh

ansible_port=22

ansible_ssh_user=heat-admin

ansible_ssh_pass=cisco123

VMM_NAME=OSPD_Newton


# Common group variables

[all:vars]

VTS_IP=[2001:420:10e:2010:172:20:100:20]

VTS_USERNAME=admin
```

```
VTS_PASSWORD=Cisco123!


[defaults]

timeout=60


[ssh_connection]

ssh_args="-C -o ControlMaster=auto -o ControlPersist=600s -o GSSAPIAuthentication=no -o
UserKnownHostsFile=/dev/null -o StrictHostKeyChecking=no"
```

Uninstalling Host Agent

To uninstall OVS host agent on Newton OSPD using CLI:

Step 1 Log in to VTC.

Step 2 Go to *cd /opt/vts/lib/ansible/playbooks*

Step 3 Create SAMPLE_INVENTORY_OVS.

Step 4 Uninstall SSH keys.

```
sudo ansible-playbook ssh_proxy.yaml -i SAMPLE_INVENTORY_OVS -e ACTION=uninstall -l proxy
```

Step 5 Uninstall the OVS Host Agent on compute.

```
sudo ansible-playbook neutron-compute.yaml -i SAMPLE_INVENTORY_OVS -e ACTION=uninstall
```

Setting Up Ansible Install Through an SSH Proxy (for RHEL OpenStack Platform Director)

Running ansible playbooks on hosts situated behind an SSH proxy is supported by installing the SSH public key of the user running the ansible script on the target hosts' *ssh allowed_hosts* file. This is necessary to run the ansible scripts on nodes deployed by the Red Hat OpenStack Platform director. The proxy host and its parameters need to be defined in a separate group in the inventory, named "proxy". Currently only one proxy per OpenStack Platform director domain is allowed, where the OpenStack Platform director domain can be composed of an arbitrary group of nodes.
An example of a Sample Inventory file:

```
[proxy]
undercloud1 ansible_ssh_host=10.194.132.62


# SSH Proxy access parameters. Do not modify the group name
[proxy:vars]
ansible_connection=ssh
ansible_ssh_user=stack
ansible_ssh_pass=cisco123
```

```
[proxied_hosts:children]
neutron_servers
vts_v_hosts


[neutron_servers]
overcloud2 ansible_ssh_host="NAME/IP of target Neutron server"


[neutron_servers:vars]
ansible_ssh_user=heat-admin


[vts_v_hosts]
rhel1 ansible_ssh_host="name/IP of target host" host_ip="20.0.87.203"
host_netmask_len="24" net_gw="20.0.87.1" underlay_if="ens224" interfaces='["eno16777984",
"eno33557248", "eno50336512"]' u_addresses='["11.0.0.0/8"]' vif_type="vhostuser"


[vts_v_hosts:vars]
ansible_ssh_user=heat-admin
```

The following command will execute the ssh key install on the two nodes in the "neutron_servers" and "vts_v_hosts" groups.

```
sudo ANSIBLE_HOST_KEY_CHECKING=False ansible-playbook ssh_proxy.yaml  -i SAMPLE_INVENTORY
-e ACTION=install
```

Following the SSH key installation, you may run the ansible playbook. For example:

```
ansible-playbook vpp.yaml -i SAMPLE_INVENTORY -e ACTION=install -l vts_v_hosts
```

Note It is important to consider variable precedence and edit or comment out the respective SSH username and password fields from the inventory file. The proxy will always be accessed on the username and password specified under the "proxy" group, while the proxied hosts will be accessed using the credentials defined in their individual group or host settings.

The inventory file and proxy settings covers only one OpenStack domain. To manage multiple domains, it is necessary to create multiple inventory files, one per domain, reusing the pattern and definitions above.

Installing VTSR

The VTSR VM acts as the control plane for the VTF. You need to install VTSR only if you plan to have a VTF in your set up.

Installing VTSR involves:

- Generating an ISO file. See Generating an ISO for VTSR, on page 15, for details.
- Deploying the VTSR on the VMM. See Deploying VTSR on OpenStack, on page 17 or Deploying VTSR on VMWare, for details.

Generating an ISO for VTSR

To create an ISO for VTSR:

Note For an HA installation, you need to create two ISOs and deploy them separately.

If you are upgrading from 2.5.2 or 2.5.2.1 to 2.6.0. you need to generate VTSR ISO again.

Step 1 Go to */opt/cisco/package/vts/share*.

Step 2 Make a copy of the vtsr_template.cfg template and edit for your VTSR instance. A sample vtsr_template.cfg file is given below:

```
# This is a sample  VTSR configuration file
# Copyright (c) 2015 cisco Systems

# Please protect the generated ISO, as it contains authentication data
# in plain text.

# VTS Registration Information:
#  VTS_ADDRESS should be the IP for VTS. The value must be either an ip or a mask.
#  VTS_ADDRESS is mandatory. If only the V4 version is specified,
#  The V4 management interface for the VTSR (NODE1_MGMT_NETWORK_IP_ADDRESS)
#  will be used. If the V6 version is specified, the V6 management interface
#  for the VTSR (NODE1_MGMT_NETWORK_IPV6_ADDRESS) must be specified and will be used.
#VTS_ADDRESS="172.23.209.17"
VTS_IPV6_ADDRESS="fded:1bc1:fc3e:96d0::1000:17"
# VTS_REGISTRATION_USERNAME used to login to VTS.
VTS_REGISTRATION_USERNAME="admin"
# VTS_REGISTRATION_PASSWORD is in plaintext.
VTS_REGISTRATION_PASSWORD="Cisco123!"
# VTSR VM Admin user/password
USERNAME="admin"
PASSWORD="cisco123"

# VTSR VM Network Configuration for Node 1:
#  NETWORK_IP_ADDRESS, NETWORK_IP_NETMASK, and NETWORK_IP_GATEWAY
#     are required to complete the setup. Netmask can be in the form of
#     "24" or "255.255.255.0"
#  The first network interface configured with the VTC VM will be used for
#  underlay connectivity; the second will be used for the management network.
#  For both the MGMT and UNDERLAY networks, a <net-name>_NETWORK_IP_GATEWAY
#  variable is mandatory; they are used for monitoring purposes.
#
#  V6 is only supported on the mgmt network and dual stack is
#  currently not supported, so if both are specified V6 will take priority (and
#  requires VTS_IPV6_ADDRESS to be set).
#  The *V6* parameters for the mgmt network are optional. Note that if V6 is used for mgmt
#  it must be V6 on both nodes. Netmask must be the prefix length for V6.
#NODE1_MGMT_NETWORK_IP_ADDRESS="172.23.209.19"
#NODE1_MGMT_NETWORK_IP_NETMASK="255.255.255.192"
```

```
#NODE1_MGMT_NETWORK_IP_GATEWAY="172.23.209.1"
NODE1_MGMT_NETWORK_IPV6_ADDRESS="fded:1bc1:fc3e:96d0::1000:19"
NODE1_MGMT_NETWORK_IPV6_NETMASK="64"
NODE1_MGMT_NETWORK_IPV6_GATEWAY="fded:1bc1:fc3e:96d0::1"
NODE1_UNDERLAY_NETWORK_IP_ADDRESS="82.82.82.19"
NODE1_UNDERLAY_NETWORK_IP_NETMASK="255.255.255.0"
NODE1_UNDERLAY_NETWORK_IP_GATEWAY="82.82.82.1"
# AUX network is optional
#NODE1_AUX_NETWORK_IP_ADDRESS="169.254.20.100"
#NODE1_AUX_NETWORK_IP_NETMASK="255.255.255.0"
#NODE1_AUX_NETWORK_IP_GATEWAY="169.254.20.1"
# XR Hostname
NODE1_XR_HOSTNAME="vtsr01"
# Loopback IP and netmask
NODE1_LOOPBACK_IP_ADDRESS="128.0.0.10"
NODE1_LOOPBACK_IP_NETMASK="255.255.255.255"

# VTSR VM Network Configuration for Node 2:
#  If there is no HA then the following Node 2 configurations will remain commented and
#  will not be used and Node 1 configurations alone will be applied
#  For HA , the following Node 2 configurations has to be uncommented
#  VTSR VM Network Configuration for Node 2
#  NETWORK_IP_ADDRESS, NETWORK_IP_NETMASK, and NETWORK_IP_GATEWAY
#     are required to complete the setup. Netmask can be in the form of
#     "24" or "255.255.255.0"
#  The first network interface configured with the VTC VM will be used for
#  underlay connectivity; the second will be used for the management network.
#  For both the MGMT and UNDERLAY networks, a <net-name>_NETWORK_IP_GATEWAY
#  variable is mandatory; they are used for monitoring purposes.
#
#  V6 is only supported on the mgmt network and dual stack is
#  currently not supported, so if both are specified V6 will take priority (and
#  requires VTS_IPV6_ADDRESS to be set).
#  The *V6* parameters for the mgmt network are optional. Note that if V6 is used for mgmt
#  it must be V6 on both nodes. Netmask must be the prefix length for V6.
#NODE2_MGMT_NETWORK_IP_ADDRESS="172.23.209.20"
#NODE2_MGMT_NETWORK_IP_NETMASK="255.255.255.192"
#NODE2_MGMT_NETWORK_IP_GATEWAY="172.23.209.1"
NODE2_MGMT_NETWORK_IPV6_ADDRESS="fded:1bc1:fc3e:96d0::1000:20"
NODE2_MGMT_NETWORK_IPV6_NETMASK="64"
NODE2_MGMT_NETWORK_IPV6_GATEWAY="fded:1bc1:fc3e:96d0::1"
NODE2_UNDERLAY_NETWORK_IP_ADDRESS="82.82.82.20"
NODE2_UNDERLAY_NETWORK_IP_NETMASK="255.255.255.0"
NODE2_UNDERLAY_NETWORK_IP_GATEWAY="82.82.82.1"
#  AUX network is optional
#  Although Aux network is optional it should be either present in both nodes
#  or not present in both nodes.
#  It cannot be present on Node1 and not present on Node2 and vice versa
#NODE2_AUX_NETWORK_IP_ADDRESS="179.254.20.200"
#NODE2_AUX_NETWORK_IP_NETMASK="255.255.255.0"
#NODE2_AUX_NETWORK_IP_GATEWAY="179.254.20.1"
# XR Hostname
NODE2_XR_HOSTNAME="vtsr02"
# Loopback IP and netmask
```

```
NODE2_LOOPBACK_IP_ADDRESS="130.0.0.1"
NODE2_LOOPBACK_IP_NETMASK="255.255.255.255"
```

Step 3 Update the following on *vtsr_template.cfg* for your deployment.

Note To deploy VTSR in HA mode, you need to create two ISOs. To create two ISOs, comment out the parameters starting NODE2_ in the sample file, and provide the appropriate values.

- VTS_ADDRESS - VTS IP address
- NODE1_MGMT_NETWORK_IP_ADDRESS - VTSR IP address
- NODE1_MGMT_NETWORK_IP_GATEWAY - VTSR gateway address
- NODE1_UNDERLAY_NETWORK_IP_ADDRESS - This is the place where TOR is connected directly
- NODE1_UNDERLAY_NETWORK_IP_GATEWAY - Underlay network IP address and Underlay network IP gateway should be brought where the VTS underlay network is configured.

Step 4 Run the *build_vts_config_iso.sh* vtsr script: This will generate the ISO file that you need to attach to the VM before booting it.
For example:

```
 admin@dev:~$ /opt/cisco/package/vts/bin/build_vts_config_iso.sh vtsr
/opt/cisco/package/vts/share/vtsr_template.cfg
               Validating input.
               validating
               Generating ISO File.
               Done!
               admin@dev:~$ ls -l
               -rw-r--r-- 1 admin vts-admin 360448 Jan 4 18:16 vtsr_node1_cfg.iso
```

Note In case you had entered the parameters for the second ISO, for HA deployment, running the script generates two ISOs.

Deploying VTSR on OpenStack

To deploy VTSR on OpenStack:

Step 1 Create VTSR.XML referring the sample XML file. For example:

```
<domain type='kvm' id='20'>
  <name>SAMPLE-VTSR-1</name>
  <memory unit='GiB'>48</memory>
  <cpu mode='host-passthrough'/>
  <vcpu placement='static'>14</vcpu>
  <resource>
    <partition>/machine</partition>
  </resource>

  <os>
    <type arch='x86_64' machine='pc-i440fx-rhel7.0.0'>hvm</type>
    <boot dev='hd'/>
    <boot dev='cdrom'/>
```

```
</os>
<features>
  <acpi/>
  <apic/>
  <pae/>
</features>
<clock offset='localtime'/>
<on_poweroff>destroy</on_poweroff>
<on_reboot>restart</on_reboot>
<on_crash>restart</on_crash>
<devices>
  <emulator>/usr/libexec/qemu-kvm</emulator>

  <disk type='file' device='cdrom'>
    <driver name='qemu'/>
    <source file='/home/admin/VTS20/images/vtsr_node1_cfg.iso'/>
    <target dev='hda' bus='ide'/>
    <readonly/>
  </disk>

  <disk type='file' device='disk'>
    <driver name='qemu' type='qcow2'/>
    <source file='/home/admin/VTS20/images/vtsr.qcow2'/>
    <target dev='vda' bus='virtio'/>
    <alias name='virtio-disk0'/>
    <address type='pci' domain='0x0000' bus='0x00' slot='0x09' function='0x0'/>
  </disk>

  <controller type='usb' index='0'>
    <alias name='usb0'/>
    <address type='pci' domain='0x0000' bus='0x00' slot='0x01' function='0x2'/>
  </controller>
  <controller type='ide' index='0'>
    <alias name='ide0'/>
    <address type='pci' domain='0x0000' bus='0x00' slot='0x01' function='0x1'/>
  </controller>
  <controller type='pci' index='0' model='pci-root'>
    <alias name='pci.0'/>
  </controller>

  <interface type='bridge'>
    <source bridge='br-ex'/>
    <virtualport type='openvswitch'>
      <parameters interfaceid='4ffa64df-0d57-4d63-b85c-78b17fcac60a'/>
    </virtualport>
    <target dev='vtsr-dummy-mgmt'/>
    <model type='virtio'/>
    <alias name='vnet1'/>
    <address type='pci' domain='0x0000' bus='0x00' slot='0x02' function='0x0'/>
  </interface>


  <interface type='bridge'>
    <source bridge='br-inst'/>
```

```
  <virtualport type='openvswitch'>
    <parameters interfaceid='4ffa64df-0d67-4d63-b85c-68b17fcac60a'/>
  </virtualport>
  <target dev='vtsr-dummy-2'/>
  <model type='virtio'/>
  <alias name='vnet1'/>
  <address type='pci' domain='0x0000' bus='0x00' slot='0x03' function='0x0'/>
</interface>


<interface type='bridge'>
  <source bridge='br-inst'/>
  <virtualport type='openvswitch'>
    <parameters interfaceid='4ffa64df-0f47-4d63-b85c-68b17fcac70a'/>
  </virtualport>
  <target dev='vtsr-dummy-3'/>
  <model type='virtio'/>
  <alias name='vnet1'/>
  <address type='pci' domain='0x0000' bus='0x00' slot='0x04' function='0x0'/>
</interface>

<interface type='bridge'>
  <source bridge='br-inst'/>
  <virtualport type='openvswitch'>
    <parameters interfaceid='4ffa64df-0d47-4d63-b85c-58b17fcac60a'/>
  </virtualport>
  <vlan>
    <tag id='800'/>
  </vlan>
  <target dev='vtsr-gig-0'/>
  <model type='virtio'/>
  <alias name='vnet1'/>
  <address type='pci' domain='0x0000' bus='0x00' slot='0x05' function='0x0'/>
</interface>

<interface type='bridge'>
  <source bridge='br-ex'/>
  <virtualport type='openvswitch'>
    <parameters interfaceid='3ffa64df-0d47-4d63-b85c-58b17fcac60a'/>
  </virtualport>
  <target dev='vtsr-gig-1'/>
  <model type='virtio'/>
  <alias name='vnet1'/>
  <address type='pci' domain='0x0000' bus='0x00' slot='0x06' function='0x0'/>
</interface>

<interface type='bridge'>
  <source bridge='br-inst'/>
  <virtualport type='openvswitch'>
    <parameters interfaceid='a2f3e85a-4de3-4ca9-b3df-3277136c4054'/>
  </virtualport>
  <vlan>
    <tag id='800'/>
  </vlan>
```

```
      <target dev='vtsr-gig-2'/>
      <model type='virtio'/>
      <alias name='vnet3'/>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x07' function='0x0'/>
    </interface>

    <serial type='pty'>
      <source path='/dev/pts/0'/>
      <target port='0'/>
      <alias name='serial0'/>
    </serial>
    <console type='pty' tty='/dev/pts/0'>
      <source path='/dev/pts/0'/>
      <target type='serial' port='0'/>
      <alias name='serial0'/>
    </console>
    <input type='tablet' bus='usb'>
      <alias name='input0'/>
    </input>
    <input type='mouse' bus='ps2'/>
    <graphics type='vnc' port='5900' autoport='yes' listen='0.0.0.0' keymap='en-us'>
      <listen type='address' address='0.0.0.0'/>
    </graphics>
    <video>
      <model type='cirrus' vram='9216' heads='1'/>
      <alias name='video0'/>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x08' function='0x0'/>
    </video>
    <memballoon model='virtio'>
      <alias name='balloon0'/>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x0a' function='0x0'/>
    </memballoon>
  </devices>
</domain>
```

Step 2 Create the VM using the XML and pointing the correct qcow2 and ISO.

```
virsh create VTSR.xml
```

Step 3 To ensure VTSR is configured with the proper Day Zero configuration, SSH to VTSR and then run:

```
RP/0/RP0/CPU0:vtsr01#bash
[xr-vm_node0_RP0_CPU0:~]$docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
31f6cbe6a048 vtsr:dev "/usr/bin/supervisord" 3 weeks ago Up 7 days vtsr
```

Step 4 Run either of the following commands:

- `[xr-vm_node0_RP0_CPU0:~]$docker exec -it vtsr bash`

Or,

- `[xr-vm_node0_RP0_CPU0:~]$docker exec -it 31 bash`

In the second option, 31 is the process ID, which you can get from Step 3.

an out put similar to the below example is displayed:

```
connecting to confd_cli
root@host:/opt/cisco/package# confd_cli -u admin -C
```

```
Welcome to the ConfD CLI
admin connected from 127.0.0.1 using console on host
host> en
host# show running-config vtsr-?
Possible completions:
vtsr-config vtsr-day0-config
host(config)# vtsr-config ?
Possible completions:
dhcp-relays global-config interfaces ip-routes l2-networks vm-macs vrfs vtfs
host(config)# vtsr-config
```

Applying VTSR Device Templates Using vts-cli.sh Script

The Day Zero configuration (OSPF, loopback0) has to be configured on VTSR using the *vts-cli.sh* script. You can apply the following templates:

Note This procedure is not required in case you have VTF in L2 switch mode.

Run vts-cli.sh, after you run `sudo su -`.

- vtsr-underlay-loopback-template. See Applying Loopback Template, on page 22
- vtsr-underlay-ospf-template. See Applying OSPF Template, on page 22

To determine the usage go to /opt/vts/bin and enter ./vts-cli.sh

```
admin@tb11-vtc:/opt/vts/bin$ ./vts-cli.sh
Usage:
        vts-cli -<command> <Name>
Valid commands are:
          vts-cli -createTemplate <templateName>
                    -- creates template structure in VTC db.
          vts-cli -applyTemplate <templateName>
                    -- collects template variables values & applies template to device.
          vts-cli -deleteTemplate <templateName>
                    -- deletes template structure from VTC db.
          vts-cli -deleteTemplateConfig <templateName>
                    -- deletes earlier applied template config from device.
          vts-cli -getTemplate <templateName>
                    -- gets template structure from VTC db.
          vts-cli -getTemplateConfig <templateName>
                    -- gets template configuration from VTS.
          vts-cli -bulkEditNtwksArp <tenantName>
                  -- collects inputs for bulk edit of arp suppression of networks associated
 with a specific Tenant.
          vts-cli -listNetworks <tenantName>
                    -- Lists L2 networks for a given Tenant.
          vts-cli -changeHostRole <host-name>
                    -- change all host connections role from managed to unmanaged (or
vice-versa)
```

If there are issues in running the commands, check the */opt/vts/bin/vts-cli.log* to get more details.

Applying Loopback Template

To apply Loopback template:

Step 1 On VTC (Master VTC in case of an HA setup), go to /opt/vts/bin.

Step 2 Run the following command:

```
admin@VTC1:/opt/vts/bin$ vts-cli.sh -applyTemplate vtsr-underlay-loopback-template
```

This will prompt you to input the parameters. For example:

```
Enter device name: vtsr01
Enter loopback-interface-number: 0
Enter ipaddress: 100.100.100.100
Enter netmask: 255.255.255.255
Template vtsr-underlay-loopback-template successfully applied to device vtsr01
```

In case you have a VTSR HA setup, apply the template on both VTSRs.

. The following message is shown if the configuration got applied correctly:

```
Template vtsr-underlay-loopback-template successfully applied to device vtsr01
```

Applying OSPF Template

To apply OSPF template:

Step 1 On VTC (Master VTC in case of an HA setup), go to /opt/vts/bin.

Step 2 Run the following command:

```
admin@VTC1:/opt/vts/bin$ vts-cli.sh -applyTemplate vtsr-underlay-ospf-template
```

This will prompt you to input the parameters. For example:

```
Enter device name: vtsr01
Enter process-name: 100
Enter router-id: 10.10.10.10
Enter area-address: 0.0.0.0
Enter physical-interface: GigabitEthernet0/0/0/0
Enter loopback-interface-number: 0
Enter default-cost: 10
```

In case you have a VTSR HA setup, apply the template on both VTSRs.

. The following message is shown if the configuration got applied correctly:

```
Template vtsr-underlay-ospf-template successfully applied to device vtsr01
```

Installing VTF on OpenStack

We recommend that you register the VMM via the VTS GUI, before you install VTF to ensure there are no errors later.

Before you install VTF, you must install VTSR and register it to VTS. See Installing VTSR, on page 14, for details.

Also, verify whether VTSR is in sync with the VTC. If not, use the sync-from operation via VTS-GUI to synchronize the VTS configuration by pulling configuration from the device. See *Synchronizing Configuration* section in the *Cisco VTS User Guide* for more information on this feature.

Note

- On all supported versions of OpenStack, Cisco VTS supports only the vhost deployment mode for VTF. Deploying VTF as a VM is not supported on OpenStack. See OpenStack VTF vhost Mode Considerations for additional details related to vhost mode installation.
- VTF as L2 switch is supported on OpenStack Newton.

Before you install VTF, do the following:

- Ensure that Stunel ver 4.56 should be as part of base compute install, before VTF vhost gets installed in OpenSack deployment.

  You can get the RPM via the URL http://mirror.centos.org/centos/7/os/x86_64/Packages/.

- Set additional routes on VTC VM(s)— You need to add routes for all underlay networks into VTC for across-the-ToR underlay communication. For example, if Switched Virtual Interface (SVI) configuration across ToR from VTC is:

```
interface Vlan100
      no shutdown
      no ip redirects
      ip address 33.33.33.1/24
      no ipv6 redirects
      ip router ospf 100 area 0.0.0.0
      ip pim sparse-mode
```

  then, below route needs to be added on VTC VM(s):

```
sudo route add -net 33.33.33.0/24 gw 2.2.2.1
```

  Where, 2.2.2.1 is the SVI IP address on the local ToR from VTC VM(s).

- Only for an OSPD setup: Set underlay IP of VTSRs via API call.

  API and Payload

```
vtsr01:

https://<VTS_IP>:8888/api/running/devices/device/vtsr01/vtsr-extension:device-info



{
  "vtsr-extension:device-info": {
    "underlay-ip": "<underlay-ip-of-vtsr01>"
  }
}
```

```
 vtsr02:
https://<VTS_IP>:8888/api/running/devices/device/vtsr02/vtsr-extension:device-info


{
  "vtsr-extension:device-info": {
    "underlay-ip": "<underlay-ip-of-vtsr02>"
  }
}
Method: PATCH


HEADERS:
Content-Type: application/vnd.yang.data+json
Accept: application/vnd.yang.data+json
```

Curl example:

```
curl -k -X PATCH -d @vtsr01.json -u <VTS_USERNAME>:<VTS_PASSWORD> -H
"Content-Type:application/vnd.yang.data+json" -H "Accept: application/vnd.yang.data+json"
 https://<VTS_IP>:8888/api/running/devices/device/vtsr01/vtsr-extension:device-info

curl -k -X PATCH -d @vtsr02.json -u <VTS_USERNAME>:<VTS_PASSWORD> -H
"Content-Type:application/vnd.yang.data+json" -H "Accept: application/vnd.yang.data+json"
 https://<VTS_IP>:8888/api/running/devices/device/vtsr02/vtsr-extension:device-info
```

vtsr01.json and vtsr02.json are files having the above payload

Step 1 Specify the VTF Mode in the System Settings. Go to **Administration > System Settings** page, select either L2 or VTEP from the drop-down, based on your requirement.

Step 2 Go to **Host Inventory > Virtual Servers**, and edit the host on which VTF-L2/VTEP installation needs to be done.

Step 3 Select the VMM Name

Step 4 Select the Virtual Switch as vtf-L2 or vtf-vtep.

Note The options that you get here are based on your selection for VTF Mode in the System Settings UI.

Step 5 Go to "VTF Details" tab and enter the required information for the VTF-L2/VTEP.

- VTF Name—Only letters, numbers, and dashes are allowed. Requires at least one letter or number.
- VTF IP—Enter Compute host underlay IPv4 address.
- Subnet Mask—Enter compute host underlay subnet mask.
- Max Huge Page Memory—Max huge page memory % that is being allocated on the host. This value is greater than 0 and less than or equal to 100. Default value is 40.
- Gateway—Enter the Compute host underlay gateway.
- PCI Driver—vfio-pci and uio-pco-generic are supported. Choose an option from the drop-down.

- Underlay Interfaces—Interface connected from compute host to the physical device (N9K/N7K/N5K). It has 2 options, Physical or Bond. Select Physical if you need to add only one interface that are connected from the compute host.
  Select Bond option if you need to add multiple interfaces that are connected from the compute host. i.e multiple entries in the Interfaces tab.
- Bond Mode—Choose required Bond mode from the dropdown.
- Bond Interfaces—Add multiple Interfaces.
- Routes to Reach Via Gateway—Routes to reach other underlay networks from this VTF host

Advanced Configurations Section:

- Multi-Threading—Set Enable Workers to true for Multithreading. By default it is set to true.
- Jumbo Frames Support—By default, it is true.
- Jumbo MTU Size—Enter Value Between Range of 1500 - 9000.

If you want to install VTF on the compute select the checkbox 'Install VTF on Save'. Depending on the type of VMM Name chosen in the Host Details tab, either you can 'Save' or 'Save and Validate'.

Step 6 Check the Install VTF on Save checkbox, and click **Save**. After VTF is successfully installed the Installation status is changed to "Successfully installed".

Note VTF installation from Cisco VTS GUI takes care of generating the inventory_file required by ansible-playbook in order to carry out the actual installation. This inventory_file is generated and saved on VTC at "/opt/vts/install/<Host IP>/inventory_file". Preserve this file. It can be obtained from the same path during uninstallation of VTF. A sample file is given below:

```
[all:vars]
VTS_IP=2.2.2.20
VTS_USERNAME=admin
VTS_PASSWORD=@@@@

vtsr_ips="['2.2.2.23', '2.2.2.24']"

[vts_v_hosts]
2001:420:10e:2010:172:20:100:25 ansible_ssh_host=2001:420:10e:2010:172:20:100:25
host_ip=2.2.2.25 host_netmask_len=24 net_gw=2.2.2.1 vhost_type=compute vif_type=vhostus er
underlay_if=enp12s0 interfaces="" u_addresses="['2.2.2.0/24', '33.33.33.0/24']"
vtf_name=VTF-Comp0
[vts_v_hosts:vars]
ansible_ssh_user=heat-admin

#ansible_ssh_private_key_file="~/.ssh/id_rsa"
config_method="static"
#name_server=<IP of NameServer>

vts_u_address=2.2.2.20

vm_2M_nr_hugepages=1024
vm_1G_nr_hugepages=1
enable_workers=True
pci_driver=uio_pci_generic

ENABLE_JUMBO_FRAMES=False
JUMBO_MTU_SIZE=None
DEFAULT_MTU_SIZE=1500
HEADERS_FOR_VPP=64
MAX_HP_MEMORY_PERC=40


[proxy]
2001:420:10e:2010:172:20:100:18 ansible_ssh_host=2001:420:10e:2010:172:20:100:18

[proxy:vars]
ansible_connection = ssh
ansible_port = 22
ansible_ssh_user=stack
ansible_ssh_pass=@@@@

[proxied_hosts:vars]
ansible_ssh_pass=@@@@
ansible_ssh_common_args='-o "ProxyCommand=ssh -C -o UserKnownHostsFile=/dev/null -o
StrictHostKeyChecking=no -o ControlMaster=auto -o ControlPersist=300s -o GSSAPIAuthe
ntication=no -W [%h]:%p -q stack@2001:420:10e:2010:172:20:100:18"'

[proxied_hosts:children]
```

```
vts_v_hosts
```

Out of Band Installation of VTF

Note On an OSPD setup, make sure to add < vtsr_ips="['2.2.2.23', '2.2.2.24']" > to inventory file else the required iptables rules for VTSR to communicate with VTF will not be added on the host at the time of VTF installation. Where 2.2.2.23 and 2.2.2.24 are IPaddresses of vtsr01 and vtsr02, respectively.

Step 1 Specify the VTF Mode in the System Settings. Go to **Administration > System Settings** page, select either L2 or VTEP from the drop-down, based on your requirement.

Step 2 Go to **Host Inventory > Virtual Servers**, and edit the host on which VTF-L2/VTEP installation needs to be done.

Step 3 Select the VMM Name

Step 4 Select the Virtual Switch as vtf-L2 or vtf-vtep.

Note The options that you get here are based on your selection for VTF Mode in the System Settings UI.

Step 5 SSH to the VTC VM (Master VTC in case of HA), switch to super user, and go to /opt/vts/lib/ansible/playbooks.

Step 6 Use inventory file and run below command on VTC command line to install VTF on the desired host.

- Setup SSH access (only required for OSPD setup):
  ```
  root@# ansible-playbook -i vtf_comp0_inventory ssh_proxy.yaml -e ACTION=install -l proxy
  ```
- Install VTF
  ```
  root@# ansible-playbook -i vtf_comp0_inventory vpp.yaml -e ACTION=install -vvvvv
  ```

Step 7 After the installation is complete, should see below message:

```
TASK [conditional_reload : Waiting for system to boot] ************************
task path: /opt/vts/lib/ansible/playbooks/conditional_reload/tasks/main.yaml:12
skipping: [2001:420:10e:2010:172:20:100:25] => {"changed": false, "skip_reason": "Conditional check
 failed", "skipped": true}

PLAY RECAP *********************************************************************
2001:420:10e:2010:172:20:100:25 : ok=27   changed=17   unreachable=0    failed=0

root@VTC1-TB1:/opt/vts/lib/ansible/playbooks#
```

Step 8 Check Host Inventory UI. VTF details such as VTF-IP and Gateway should be auto-populated.

Step 9 Click **Save** for installation status to get updated. Installation status of VTF should be appropriately updated.

Deleting VTF in an OpenStack Environment

Step 1 Using the same inventory file sample used/generated while you had installed VTF, run the following command from VTC command line to uninstall VTF from the host:

```
root@# ansible-playbook -i vtf_comp0_inventory vpp.yaml -e ACTION=uninstall -vvvvv
```

Once uninstallation is complete, you should see the below output:

```
TASK [conditional_reload : Waiting for system to boot] ************************
task path: /opt/vts/lib/ansible/playbooks/conditional_reload/tasks/main.yaml:12
skipping: [2001:420:10e:2010:172:20:100:25] => {"changed": false, "skip_reason": "Conditional check
 failed", "skipped": true}

PLAY RECAP *********************************************************************
2001:420:10e:2010:172:20:100:25 : ok=27   changed=17   unreachable=0    failed=0

root@VTC1-TB1:/opt/vts/lib/ansible/playbooks#
```

Step 2 Go to Host Inventory and edit the host to change the Virtual Switch mode to *not-defined* and click **Save**.

Step 3 Verify whether the Installation status has disappeared.

Step 4 Verify whether the VTF is removed from Inventory > Virtual Forwarding Groups UI.

Running VTF on Controller node(s) to enable DHCP

This is enabled via an ansible-based installation. The sample inventory file to be used for this is given below:

```
### Common group variables ###

[all:vars]

# When using IPv6 literals enclose the address in square brackets []

VTS_IP="<VTS IP>"

VTS_USERNAME="admin"

VTS_PASSWORD="Cisco123!"

VMM_NAME="OSPD-Newton"

vtc_username="admin"

vtsr_ips='["1.1.1.1", "2.2.2.2"]'


[vts_v_hosts]

overcloud-controller-1.localdomain ansible_ssh_host="192.168.126.101" vhost_type="compute"
 host_ip="114.1.1.20" host_netmask_len="255.255.255.0" net_gw="114.1.1.1"
underlay_if="enp129s0f0" u_addresses='["114.1.1.0/24"]' vif_type="tap"


[vts_v_hosts:vars]

ansible_ssh_user=heat-admin
```

```
config_method="static"

name_server="171.70.168.183"

#VTS address on the underlay. If not set, defaults to VTS_IP

vts_u_address="114.1.1.101"


# For DHCP so don't need to allocate large memory to huge pages

max_hp_memory_perc=30

enable_workers=True

pci_driver="uio_pci_generic"

# If needed enable jumbo

#ENABLE_JUMBO_FRAMES="True"

#JUMBO_MTU_SIZE=9000

DEFAULT_MTU_SIZE=1500

HEADERS_FOR_VPP=64


### Neutron Control Servers ###

[neutron_servers]

overcloud-controller-1.localdomain ansible_ssh_host="192.168.126.101"


[neutron_servers:vars]

ansible_connection=ssh

ansible_ssh_user=heat-admin


[proxied_hosts:children]

neutron_servers

vts_v_hosts


[proxied_hosts:vars]

ansible_ssh_pass=<DIRECTOR LOGIN PASSWORD>

ansible_ssh_common_args='-o ProxyCommand="ssh -C -o UserKnownHostsFile=/dev/null -o
StrictHostKeyChecking=no -o ControlMaster=auto -o ControlPersist=300s -o
GSSAPIAuthentication=no -W %h:%p -q
{{hostvars[groups[\'proxy\'][0]][\'ansible_ssh_user\']}}@{{hostvars[groups[\'proxy\'][0]][\'ansible_ssh_host\']}}"'



 [proxy]

undercloud1 ansible_ssh_host="<DIRECTOR IP>"
```

```
[proxy:vars]
ansible_ssh_user=<DIRECTOR LOGIN NAME>
ansible_ssh_pass=<DIRECTOR LOGIN PASSWORD>
```

Step 1 On the Cisco VTS GUI enter VTF details (Inventory > Host Inventory), but do not trigger installation.

Step 2 Select uio_pci_generic for PCI Driver to avoid reboot of Controller nodes.

Step 3 Run ansible ssh_proxy. Go to *cd /opt/vts/lib/ansible/playbooks*, run:

```
sudo ANSIBLE_HOST_KEY_CHECKING=False ansible-playbook ssh_proxy.yaml  -i SAMPLE_INVENTORY -e
ACTION=install -vvvv
```

Step 4 Run *vpp.yaml* to install VTF.

```
sudo ANSIBLE_HOST_KEY_CHECKING=False  ansible-playbook vpp.yaml -i SAMPLE_INVENTORY -e ACTION=install
 -vvvv
```

Step 5 Run *neutron-ctrl.yaml* to configure DHCP configuration file on Controller.

```
sudo ANSIBLE_HOST_KEY_CHECKING=False ansible-playbook neutron-ctrl.yaml -i SAMPLE_INVENTORY -e
ACTION=configure -vvvv
```

Step 6 Check whether this file has the correct interface driver (interface_driver = cisco_controller.drivers.agent.linux.interface.NamespaceDriver).

```
less /etc/neutron/dhcp_agent.ini
```

Step 7 Make sure that the VTF is able to reach underlay gateway, VTC/VTSR, and IP Tables.

Verifying VTS Installation

The following sections provide information about how to verify the VTS installation:

- Verifying VTC VM Installation, on page 30
- Verifying VTSR Installation, on page 31
- Verifying VTF Installation, on page 32

Verifying VTC VM Installation

To verify VTC VM installation:

Step 1 Log in to the VTC VM just created using the VTC VM console.

- If you have installed the VTC VM in a VMware environment, use the VM console.

- If you have installed the VTC VM in an RedHat KVM based-OpenStack environment, - telnet 0 <*console-port*> (The console port is telnet port in the VTC.xml file.)

Step 2 Ping the management gateway.
In case ping fails, verify the VM networking to the management network.

Step 3 For the VTC VM CLI, ping the underlay gateway.
In case the ping fails, verify VM networking to the underlay network.

Note Underlay network gateway is the switched virtual interface (SVI) created for VTSR and VTF on the leaf where the controller is connected.

Step 4 Verify whether the VTS UI is reachable, by typing in the VTS management IP in the browser.

Verifying VTSR Installation

To verify VTSR installation:

Step 1 Log in to the VTSR.

- If you have installed the VTC VM in a VMware environment, use the VM console.
- If you have installed the VTC VM in an RedHat KVM based-OpenStack environment, use virt-manager or VNC based console method to login into the VM. See Installing VTC VM - Manual Configuration using VNC, on page 5

Step 2 Ping the underlay gateway IP address.
In case ping fails, verify underlay networking.

Step 3 Ping the VTC VM.
In case ping fails, verify underlay networking.

Note You should be able to ping the gateway IP address for both management and underlay networks, as VTSR registers to the VTC using the management IP address.

Step 4 Run **virsh list** to make sure the nested VM is running.

Step 5 Verify whether the Virtual Forwarding Group (VFG) group is created on VTS GUI, and VTSR is part of the VFG group.

Note This is not available if you are running VTF in L2 mode (Administration > System Settings > VTF Mode set to L2).

Verifying VTF Installation

To verify VTF installation:

Step 1 Log in to the VTF VM / vhost.

- If you have installed the VTC VM in a VMware environment, use the VM console.
- If you have installed the VTC VM in an RedHat KVM based-OpenStack environment, use virt-manager or VNC based console method to login into the VM. See Installing VTC VM - Manual Configuration using VNC, on page 5

Step 2 Ping the underlay gateway IP address.
In case ping fails, verify underlay networking.

Step 3 Ping the VTC VM underlay IP address.
In case ping fails, verify underlay networking.

Step 4 Verify whether the VTF CLI is available . To do this, run:

```
'sudo vppctl
```

If the o/p command fails, run the following command to identify whether vpfa service is up and running:

```
sudo service vpfa status
```

If there are errors, try restarting the service.

```
sudo service vpfa restart
```

Step 5 Verify whether the VTF is part of the VFG, on VTS GUI.

Note This is not applicable is you have VTF is L2 mode (Administration > System Settings > VTF Mode is L2).

Changing Password for Cisco VTS from VTS GUI

The GUI password change will trigger the updating of password on all host agents which are running on the Physical computes. And if there are VTFs in your setup, then the VTSR and VTF passwords will also get updated.

Important

- Traffic disruption will happen only if you have VTFs installed (Virtual deployment) and it happens because of the vpfa process restart.

  In case of a Physical deployment there will not be any traffic disruption.
- For Baremetal ports there is no impact.
- The password change from the GUI will change only the host agent password. Not the Linux password. So, we cannot use the command 'passwd'
- If you are changing the Linux password of a Physical or Virtual host then you should also update the VTC host inventory with correct password. Changing the Linux password will not impact any traffic.

Step 1 Log in to VTS GUI and click on settings icon on the top-right corner and click **Change Passphrase**.

Step 2 Enter the current password, new password, then click **Change Passphrase**.

Step 3 Click **OK** in the Confirm Change Passphrase popup, to confirm.

Note The message in the Confirm Change Passphrase window is just a generic message. See important notes above for details about possible traffic disruption.

Changing Password for Cisco VTS Linux VM

You can use the Linux command 'passwd' to change the VTC VM password. After changing the password, you should use the new password for the subsequent SSH session to the VTC VM.

For an HA installation you must change the password on both Master and Slave with the command 'passwd'.

Changing Password for OSPD-integrated VTFs and VTSRs

Step 1 Change the password from the Cisco VTS GUI.

Step 2 Download the password encryption tool from https://devhub.cisco.com/artifactory/list/vts-yum/2.6.0/salt/encrypt-pass-2.6.0.vts260-10.tar.gz.

Step 3 From the OpenStack director, open the file neutron-cisco-vts.yaml and update the below field with newly encrypted password with the tool 'encrypt-password'.

```
VTSPassword: ''
```

Step 4 Redeploy the overcloud.

Running the Password Encryption Script

Ensure that the System has:

- Python with version 2.7 or greater with the standard libraries.
- Python module pycrypto (pip install pycrypto)

Step 1 Download the password encryption tool from https://devhub.cisco.com/artifactory/list/vts-yum/2.6.0/salt/encrypt-pass-2.6.0.vts260-10.tar.gz

Step 2 Untar the file using the below command:

```
$tar -xvf encrypt-pass.tar
```

Step 3 Go to <encrypt_pass> directory and run the below command:

```
$./encrypt-password <clearTextPassword>
```

Note Any special characters in the password need to be preceded with \. For example, Cisco123! should be entered as Cisco123\!