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 6

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 any linux server that is reachable to all the controller/compute nodes as well as the fabric devices 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 6, for details.

This is applicable when Admin has selected **Yes** to the Question "Do you want VTS to install VMM plugin 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. Hostname entered should be of FQDN format, that is, <hostname>.<domain>.
- 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
```

Note If compute has server-type “virtual-server” has to be associated with a VMM prior to the upgrade. If there is no VMM associated, “virtual-server” will be converted to a “baremetal” type during the upgrade. This is because VMM association is mandatory for a virtual server starting VTS v2.6.2.

OSPD 10 Integration

With VTS262 a new enhanced and significantly simpler method of installing the VTS components has been introduced, effective obsoleting the VTS260 procedure. The procedure is also available in VTS261.

This document provides the main install and configuration steps, including the configuration of the multi site feature (introduced in 262).

Brief Overview:

In contrast to the overcloud package install procedure in VTS260, the new overcloud package install procedure does not rely on the modification of the overcloud image. It operates by using the native package manager

on the overcloud nodes to access the package repository and install the packages via the "NodeExtraConfig" hook. The install happens thus transparently, and can be also initiated on already deployed overcloud nodes. The overclud nodes require thus access to the yum package repository, and also RH package registration.

The configuration of the components follows on - this operation is unchanged from VTS260, with the exception of new features.

Install Packages on the undercloud director:

Step 1 On the undercloud director node Install the cisco262 newton repo (Edit the credentials accordingly)

```
sudo cat > /etc/yum.repos.d/262.repo <<EOL
[cisco2.6.2.vts262-os-newton]
name=cisco2.6.1.vts262-os-newton
baseurl=https://devhub.cisco.com/artifactory/vts-yum-dev/2.6.2.vts262-os-newton
username=<username>
password=<passwd>
enabled = 1
gpgcheck = 0
metadata_expire = 86400

EOL
```

Step 2 Install the THT extra RPM:

```
sudo yum install cisco-vts-tripleo-heat-templates-extra --enablerepo cisco2.6.2.vts262-os-newton
```

Step 3 (optional) Install the VTS tools

```
sudo yum install cisco-vts-os-util --enablerepo cisco2.6.2.vts262-os-newton
```

Edit the Cisco VTS Environment Template

Step 1 Copy the vts environment template:

```
cp /usr/share/openstack-tripleo-heat-templates/environments/neutron-cisco-vts.yaml ~/templates
```

Step 2 Edit the neutron-cisco-vts.yaml template.

Key items requiring user edits highlighted in RED.

Refer https://cisco.jiveon.com/docs/DOC-1882605 for configuration parameters.

```
cat /home/stack/templates/neutron-cisco-vts.yaml
```

```
## A Heat environment file which can be used to enable Cisco VTS extensions, configured via puppet
# vts 2.6.1

# By default the configuration has items required to deploy VPP/VPFA on all nodes + the cisco ML2
VTS driver

resource_registry:

  ## Base Neutron ML2 definitions for VTS
  OS::TripleO::Services::NeutronCorePluginVTS:
/usr/share/openstack-tripleo-heat-templates/puppet/services/neutron-plugin-ml2-cisco-vts.yaml
  OS::TripleO::Services::NeutronCorePlugin: OS::TripleO::Services::NeutronCorePluginVTS
```

```
  ## Comment out below line when deploying VTS Agent on compute nodes instead of VPP/VPFA
  OS::TripleO::Services::ComputeNeutronCorePlugin: OS::TripleO::Services::NeutronCorePluginVTS

  ## Disable Neutron L3 agent that conflict with VPFA
  OS::TripleO::Services::NeutronL3Agent: OS::Heat::None

  ## OVS and VTS Agent sub-section ##

  ## Disable/enable the default OVS Agent for compute and controller
  OS::TripleO::Services::ComputeNeutronOvsAgent: OS::Heat::None
  OS::TripleO::Services::NeutronOvsAgent: OS::Heat::None

  ## Disable/enable VTS agent service. VTS agent and OVS agent are mutually exclusive
  ## NOTE: The OS::TripleO::Services::VTSAgent needs to be added to the deployment role file
  OS::TripleO::Services::VTSAgent:
/usr/share/openstack-tripleo-heat-templates/puppet/services/neutron-cisco-vts-agent.yaml
## Package install and VPFA Configuration Hook scripts with RH registration wrapper
  OS::TripleO::NodeExtraConfig:
/usr/share/openstack-tripleo-heat-templates/puppet/extraconfig/pre_deploy/cisco_vts_rh_reg_wrapper.yaml
## Rsyslog client
 OS::TripleO::Services::RSyslogClient:
/usr/share/openstack-tripleo-heat-templates/puppet/services/rsyslog-client.yaml

 ## VPP Service(s)
 OS::TripleO::Services::Vpp: OS::Heat::None
 OS::TripleO::Services::VppCompute:
/usr/share/openstack-tripleo-heat-templates/puppet/services/vpp-compute.yaml
 OS::TripleO::Services::VppController:
/usr/share/openstack-tripleo-heat-templates/puppet/services/vpp-controller.yaml
 OS::TripleO::Services::CiscoVpfaCompute:
/usr/share/openstack-tripleo-heat-templates/puppet/services/cisco-vpfa-compute.yaml
 OS::TripleO::Services::CiscoVpfaController:
/usr/share/openstack-tripleo-heat-templates/puppet/services/cisco-vpfa-controller.yaml

 ## Monit agent service(s)
 OS::TripleO::Services::MonitAgent:
/usr/share/openstack-tripleo-heat-templates/puppet/services/monit-agent.yaml
 OS::TripleO::Services::MonitVpfaAgent:
/usr/share/openstack-tripleo-heat-templates/puppet/services/monit-agent-vpfa.yaml

 ## Collectd agent service
 OS::TripleO::Services::CollectDAgent:
/usr/share/openstack-tripleo-heat-templates/puppet/services/collectd-agent.yaml



parameter_defaults:

 ## Current VTS version
 VTSversion: "2.6.2"

 ####################
 ### VTS General ###
 ####################

 VTSUsername: 'admin'
 VTSPassword:
 VTSServer: ''
 VTSVMMID: ''
 VTSSiteId: ''

 ####################
 ### Neutron ML2 ###
 ####################
```

```
 NeutronCorePlugin: 'neutron.plugins.ml2.plugin.Ml2Plugin'
 NeutronMechanismDrivers: 'sriovnicswitch,cisco_vts'
 NeutronTypeDrivers: 'vxlan,vlan,flat'
 NeutronServicePlugins: 'cisco_vts_router,trunk'

 ## DHCP Agent interface driver. Uncomment ONLY if/when deploying VPP on the controller node(s).
 #NeutronInterfaceDriver: 'cisco_controller.drivers.agent.linux.interface.NamespaceDriver'

 ########################
 ### VTS Agent Config ###
 ########################

 VTSPhysicalNet: ''
 #VTSRetries: 15
 #VTSTimeout:
 #VTSPollingInterval: 6

 ###################
 ### VPFA Config ###
 ###################

 UnderlayIpNewtorksList: ''
 VTSR_u_IpAddressList: ''
 #NetworkNameServerIP: ''

 ## Set a common VTS Network Gateway address OR set/override it using the PerNodeData parameter
further-on
 #VTSNetworkIPv4Gateway: '10.0.0.1'

 # VPFA Configuration requires the assignment of an underlay IP address for the VPFA per node.
 # This needs to be specified against the UUID of the target node in a JSON data blob.
 # To derive the UUID, after node introspection execute the following CLI command steps:
 #
 # 1. 'ironic node-list'. Note Openstack ID of the target node
 # 2. 'openstack baremetal introspection data save <Openstack ID from step1> | jq
.extra.system.product.uuid
 # 3. Note the Node UUID and use it in the json configuration blob below. Multiple nodes can be
specified.
 #
 # The per-node data can be used to set/override any of the cisco_vpfa:: module configuration parameters

 #

 PerNodeData: |
 {
 "< Node1 UUID >": {
 "cisco_vpfa::vtf_underlay_ip_v4": "10.0.0.2",
 "cisco_vpfa::vtf_underlay_mask_v4": "24",
 "cisco_vpfa::network_ipv4_gateway": "10.0.0.1"},
 "< Node2 UUID >": {
 "cisco_vpfa::vtf_underlay_ip_v4": "10.0.0.3",
 "cisco_vpfa::vtf_underlay_mask_v4": "24",
 "cisco_vpfa::network_ipv4_gateway": "10.0.0.1"}
 }

 ## Enable/Disable VPFA collection of VPP Stats (defaults to true when not set)
 #VppStats: True

 ####################################
 ### VPP Configuration Parameters ###
 ####################################

 ## MTU for Tun/tap interfaces
```

```
#VppTunTapMtu: '9000'

## The CPUs listed below need to be part of the grub isol CPU list (configured elsewhere)
#VppCpuMainCoreController: '0'
#VppCpuMainCoreCompute: '0'

## Comma delimited workers list
#VppCpuCorelistWorkersCompute: ''
#VppCpuCorelistWorkersController: ''

## Avoid dumping vhost-user shared memory segments to core files
#VppVhostUserDontDumpMem: false

########################
### VTS Update Info ###
########################

VTSUpdate: 'true'

## VTS Yum Repo settings
VTSyumRepos: |
[cisco2.6.2.vts262]
name=cisco2.6.2.vts262
baseurl=http://devhub.cisco.com/artifactory/vts-yum-dev/2.6.2.vts262
username=
password=
enabled = 1
gpgcheck = 0
metadata_expire = 86400

[cisco2.6.2.vts262-os-newton]
name=cisco2.6.2.vts261-os-newton
baseurl=http://devhub.cisco.com/artifactory/vts-yum-dev/2.6.2.vts262-os-newton
username=
password=
enabled = 1
gpgcheck = 0
metadata_expire = 86400

## Repository Proxy Settings
RepoProxy: 'http://proxy.esl.cisco.com:8080/'

########################################
### VPFA rSyslog Client Configuration ###
########################################

# IMPORTANT: Add OS::TripleO::Services::RSyslogClient to the role data catalogue for the service
config to come into
# effect

# ****** EDIT the syslog server <IP ADDRESS> and <PORT> in ClientLogFilters and add/remove entries
as needed! *******
# The default template below configures UDP servers on port 514. UDP is denoted by a single @ sign.
To add a TCP
# server, add an extra stanza prefixing with @@ the server's IP address

ClientLogFilters: |
[
{
 "expression": "$syslogfacility-text == 'local3' and $syslogseverity-text == 'crit'",
"action": "@[<IP ADDRESS>]:<PORT>;forwardFormat"
},
{
"expression": "$syslogfacility-text == 'local3' and $syslogseverity-text == 'err'",
```

```
"action": "@[<IP ADDRESS>]:<PORT>;forwardFormat"
},
{
"expression": "$syslogfacility-text == 'local3' and $syslogseverity-text == 'warning'",
"action": "@[<IP ADDRESS>]:<PORT>;forwardFormat"
},
{
"expression": "$syslogfacility-text == 'local3' and $syslogseverity-text == 'info'",
"action": "@[<IP ADDRESS>]:<PORT>;forwardFormat"
}
]

# Cisco VPFA default log and priority settings
ImFiles: |
{
"10-vpfa_error_log": {
"file_name": "/var/log/vpfa/vpfa_server_errors.log",
"file_tag": "vpfa",
"file_severity": "err",
"file_facility": "local3"
},
"10-vpfa_warning_log": {
"file_name": "/var/log/vpfa/vpfa_server_warning.log",
"file_tag": "vpfa",
"file_severity": "warning",
"file_facility": "local3"
},
"10-vpfa_critical_log": {
"file_name": "/var/log/vpfa/vpfa_server_critical.log",
"file_tag": "vpfa",
"file_severity": "critical",
"file_facility": "local3"
},
"10-vpfa_info_log": {
"file_name": "/var/log/vpfa/vpfa_server.log",
"file_tag": "vpfa",
"file_severity": "info",
"file_facility": "local3"
}
}

ClientLogTemplates: |
[
{
"name": "forwardFormat",
"template": "<%PRI%>%TIMESTAMP:::date-rfc3339% %HOSTNAME% %syslogtag:1:32%%msg:::sp-if-no-1st-sp%%msg%"

}
]


###################################
### Monit Agent Configuration ###
###################################

# IMPORTANT: To enable the Monit Agent config, add the VPFA specific
"OS::TripleO::Services::MonitVpfaAgent"
# or generic "OS::TripleO::Services::MonitAgent" to the corresponding nodes role data configuration.



## General settings. Applied to all Monit Agents
## Credentials
MonitUser: ''
```

```
 MonitPassword:
 MonitSSLPemFile: '/etc/ssl/certs/monit.pem'

 ## VPFA Monit node bind IP address - when unset, defaults to use underlay IP of the VPFA
 #MonitVpfaBindAddress:
 ## Generic node's monit server bind IP address - when unset, defaults to the management IP of the
node.
 #MonitBindAddress:

 ## Monit server port
 #MonitHttpServerPort: 2812

 ## Monit check interval
 # MonitCheckInterval: 30

 ## Monit Check config applied on nodes enabled with the OS::TripleO::Services::MonitVpfaAgent role.

 MonitVpfaChecks: |
 {
 "vpp":
 {
 "type": "process",
 "config":
 {
 "matching": "vpp",
 "program_start": "/sbin/service vpp start",
 "program_stop": "/sbin/service vpp stop"
 }
 },
 "vpfa":
 {
 "type": "process",
 "config":
 {
 "matching": "vpfa_restconf_server",
 "program_start": "/sbin/service vpfa start",
 "program_stop": "/sbin/service vpfa stop"
 }
 }
 }

 ## Raw config added to nodes enabled with the OS::TripleO::Services::MonitVpfaAgent role.
 ## Used in to add configuration not supported by the puppet module types.
 MonitVpfaRawConfig: |
 'check network underlay interface vnet'

 ## Check config applied on nodes enabled with the OS::TripleO::Services::MonitAgent role.
 MonitChecks: |
 {
 }

 ## Used in to add configuration not supported by the puppet module types.
 MonitRawConfig: |
 ''

 ######################################
 ### Collectd Agent Configuration ###
 ######################################

 # IMPORTANT: To enable the Collectd Agent config, add the "OS::TripleO::Services::CollectDAgent"


 ##Enable or disable collectd (default is true)
 # CollectDEnable: true
```

```
## Purge default/previous configurations
CollectDPurge: true

## CollectD Plugin configurations
## Each named plugin should have its own named dict entry, followed by a "content" element containing
the
## plugin's XML configuration stanza, in JSON list format.
## The configuration content is the native collectd configuration for the plugin
CollectDPluginConfigs: |
{
"memory":
{
"content":
[
"<Plugin memory>",
"ValuesAbsolute true",
"ValuesPercentage false",
"</Plugin>"
]
},
"cpu":
{
"content":
[
"<Plugin cpu>",
"ReportByCpu true",
"ReportByState true",
"ValuesPercentage false",
"ReportNumCpu false",
"ReportGuestState false",
"SubtractGuestState true",
"</Plugin>"
]
},
"python":
{
"content":
[
"<Plugin python>",
"ModulePath \"/opt/cisco/vpe/collectd/\"",
"LogTraces true",
"Import \"cisco-vpfa-collectd-plugin\"",
"</Plugin>"
]
},
"write_log":
{
"content":
[
"<Plugin write_log>",
"Format JSON",
"</Plugin>"
]
},
"interface":
{
"content":
[
"<Plugin interface>",
"Interface \"br-ctlplane\"",
"Interface \"br-ex\"",
"Interface \"br-tenant\"",
"Interface \"lo\"",
```

```
 "IgnoreSelected false",
 "ReportInactive true",
 "UniqueName false",
 "</Plugin>"
 ]
 },
 "disk":
 {
 "content":
 []
 },
 "load":
 {
 "content":
 [
 "<Plugin load>",
 "ReportRelative true",
 "</Plugin>"
 ]
 }
 }
```

Step 3 Edit the RH Registration template.

```
cat /home/stack/templates/rhel-registration/environment-rhel-registration.yaml
```

```
# Note this can be specified either in the call
# to heat stack-create via an additional -e option
# or via the global environment on the seed in
# /etc/heat/environment.d/default.yaml
parameter_defaults:
  rhel_reg_activation_key: ""
  rhel_reg_auto_attach: "auto"
  rhel_reg_base_url: ""
  rhel_reg_environment: ""
  rhel_reg_force: "true"
  rhel_reg_machine_name: ""
  rhel_reg_org: ""
  rhel_reg_password: ""
  rhel_reg_pool_id: ""
  rhel_reg_release: ""
  rhel_reg_repos:
"rhel-7-server-rpms,rhel-7-server-extras-rpms,rhel-7-server-rh-common-rpms,rhel-ha-for-rhel-7-server-rpms,rhel-7-server-openstack-10-rpms"

  rhel_reg_sat_url: ""
  rhel_reg_server_url: ""
  rhel_reg_service_level: ""
  rhel_reg_user: ""
  rhel_reg_type: ""
  rhel_reg_method: "portal"
  rhel_reg_sat_repo: ""
  rhel_reg_http_proxy_host: ""
  rhel_reg_http_proxy_port: ""
  rhel_reg_http_proxy_username: ""
  rhel_reg_http_proxy_password: ""
```

Step 4 When deploying VPP: Edit the NIC templates

```
snipped output of cat ~/templates/nic-configs/compute.yaml
```

```
....


config:
```

```
  os_net_config:
    network_config:

          …existing settings…..

            -
              type: vpp_bond
              name: net_bonding0
              bonding_options: "mode=2,xmit_policy=l34"
              members:
                -
                  type: vpp_interface
                  name: enp4s0f2
                  uio_driver: uio_pci_generic
                  options: vlan-strip-offload off
                -
                  type: vpp_interface
                  name: enp4s0f1
                  uio_driver: uio_pci_generic
                  options: vlan-strip-offload off

#For a single interface, the list item would be:
            -
              type: vpp_interface
              name: enp4s0f3
              uio_driver: uio_pci_generic
```

Configure the Service Roles

Add the necessary services to the defined node roles.

```
sudo vi /usr/share/openstack-tripleo-heat-templates/roles_data.yaml
```

Then add the following roles to the compute(s) or controllers where the VPFA and VPP are to run:

For compute nodes add either VTSAgent or the combination of Vpp and Vpfa:

```
OS::TripleO::Services::VTSAgent

    - OS::TripleO::Services::VppCompute

    - OS::TripleO::Services::CiscoVpfaCompute

    - OS::TripleO::Services::RSyslogClient
    - OS::TripleO::Services::MonitVpfaAgent
    - OS::TripleO::Services::CollectDAgent
```

If running VTS agent or VPP/VPFA on controller nodes add either VTSAgent or the combination of Vpp and Vpfa:

```
OS::TripleO::Services::VTSAgent

    - OS::TripleO::Services::VppController

    - OS::TripleO::Services::CiscoVpfaController

    - OS::TripleO::Services::RSyslogClient
    - OS::TripleO::Services::MonitVpfaAgent
    - OS::TripleO::Services::CollectDAgent
```

IMPORTANT: Whenever running VTS-agent or VPP/VPFA either remove the OS::TripleO::Services::NeutronOvsAgent and OS::TripleO::Services::OvsAgent services from the node role definitions or include the following in your environment file:

```
OS::TripleO::Services::ComputeNeutronOvsAgent: OS::Heat::None

OS::TripleO::Services::NeutronOvsAgent: OS::Heat::None
```

Deploy the Overcloud

Include the edited environment files with the deploy command

```
openstack overcloud deploy \
--templates \
-e /usr/share/openstack-tripleo-heat-templates/environments/network-isolation.yaml \
-e /home/stack/templates/network-environment.yaml \
-e /home/stack/templates/rhel-registration/environment-rhel-registration.yaml \
-e /home/stack/templates/neutron-cisco-vts.yaml \
--control-scale 1 \
--compute-scale 1 \
--control-flavor control \
--compute-flavor compute \
--log-file oclogs/overcloudDeploy_$(date +%m_%d_%y__%H_%M_%S).log \
--ntp-server ntp.esl.cisco.com \
--verbose --timeout 100
```

OSPD 13 and VTS262 ML2 Integration

VTS integration with OSPD13 relies on a VTS Tripleo Heat Templates package and VTS specific (eg ML2) containers. This section documents the installation and configuration of the system for ML2 integration.

Before You Begin:

- If you are using a docker registry other than **Undercloud**, you must modify the configuration according to the RH OSPD13 documentation.

- Ensure that RH Undercloud is installed, and that the standard RH container images are downloaded and set up as per the RH OSPD13 documentation. For more information, see https://access.redhat.com/documentation/en-us/red_hat_openstack_platform/13/html/director_installation_and_usage/

- Ensure that the RH or satellite registration environment template is complete as per https://access.redhat.com/documentation/en-us/red_hat_openstack_platform/13/html/advanced_overcloud_customization/sect-registering_the_overcloud#registering_the_overcloud_with_an_environment_file. The RH registration environment template file is by default available at the following location:

  /home/stack/templates/rhel-registration/environment-rhel-registration.yaml

Procedure

	Command or Action	Purpose
Step 1	Set up your HTTP proxy configuration by configuring the HTTP_PROXY and HTTPS_PROXY environment variables. This step is optional.	
Step 2	Install the cisco262 queens repo.	**Note** You must edit the credentials accordingly. `sudo cat > /etc/yum.repos.d/262.repo <<EOL` `[cisco2.6.2.vts262]` `name=cisco2.6.2.vts262` `baseurl=https://devhub.cisco.com/artifactory/vts-yum-dev/2.6.2.vts262` `username=<username>` `password=<password>` `enabled = 0` `gpgcheck = 0` `metadata_expire = 86400` `[cisco2.6.2.vts262-os-queens]` `name=cisco2.6.1.vts262-os-queens` `baseurl=https://devhub.cisco.com/artifactory/vts-yum-dev/2.6.2.vts262-os-queens` `username=<username>` `password=<passwd>` `enabled = 0` `gpgcheck = 0` `metadata_expire = 86400` `EOL` `sudo yum install cisco-vts-tripleo-heat-templates-extra --enablerepo cisco2.6.2.vts262-os-queens`
Step 3	Perform these steps to download the Neutron ML2 container:	1. Log in to the RH repository using the RH SSO credentials. `sudo docker login -u <username> registry.connect.redhat.com` `Password: <password>` 2. Pull the ML2 container. `docker pull registry.connect.redhat.com/cisco/cisco-vts262` 3. Tag the container. `docker tag registry.connect.redhat.com/cisco/cisco-vts262`

	Command or Action	Purpose
		192.168.126.1:8787/rhosp13/neutron-cisco-vts-ml2 4. Push the container into the repository. docker push 192.168.126.1:8787/rhosp13/neutron-cisco-vts-ml2
Step 4	Perform these steps to set up the **neutron-cisco-vts.yaml** environment file.	1. Copy the environment file template from its default location to your templates directory. cp /usr/share/openstack-tripleo-heat-templates/environments/neutron-ml2-cisco-vts-262.yaml ~/templates 2. Complete the highlighted configuration items in the template. In the ~/templates/neutron-ml2-cisco-vts-262.yaml environment file, complete the highlighted configuration items: # A docker enabled Heat environment file which can be used to enable Cisco VTS , configured via puppet # By default the configuration has items required to deploy the cisco ML2 VTS driver + LLDP on all nodes resource_registry: OS::TripleO::Services::NeutronCorePluginVTS: ../../docker/services/neutron-plugin-ml2-cisco-vts-262.yaml OS::TripleO::Services::NeutronCorePlugin: OS::TripleO::Services::NeutronCorePluginVTS OS::TripleO::Services::ComputeNeutronCorePlugin: OS::TripleO::Services::NeutronCorePluginVTS OS::TripleO::NodeExtraConfig: /usr/share/openstack-tripleo-heat-templates/puppet/extraconfig/pre_deploy/cisco_vts_hiera_override.yaml resource_registry: OS::TripleO::Services::NeutronCorePluginVTS: ../../docker/services/neutron-plugin-ml2-cisco-vts-262.yaml OS::TripleO::Services::NeutronCorePlugin: OS::TripleO::Services::NeutronCorePluginVTS OS::TripleO::Services::ComputeNeutronCorePlugin: OS::TripleO::Services::NeutronCorePluginVTS OS::TripleO::NodeExtraConfig: /usr/share/openstack-tripleo-heat-templates/puppet/extraconfig/pre_deploy/cisco_vts_hiera_override.yaml ## ### Docker Cisco VTS Neutron images ### ## DockerNeutronApiImage: 'repo/neutron-cisco-vts-ml2:latest' DockerNeutronConfigImage: 'repo/neutron-cisco-vts-ml2:latest' ##################

	Command or Action	Purpose
		``### VTS General ### #################### VTSUsername: 'admin' VTSPassword: VTSServer: VTSVMMID: VTSSiteId: #################### ### Neutron ML2 ### #################### NeutronCorePlugin: 'neutron.plugins.ml2.plugin.Ml2Plugin' NeutronMechanismDrivers: 'sriovnicswitch,cisco_vts' NeutronTypeDrivers: 'vxlan,vlan,flat' NeutronServicePlugins: 'cisco_vts_router,trunk' ############################ ### Install VTS packages ### ############################ VTSUpdate: 'true' VTSyumRepos: | [cisco2.6.2.vts262] name=cisco2.6.2.vts262 baseurl=https://devhub.cisco.com/artifactory/vts-yum-dev/2.6.2.vts262 username=<user> password=<pass> enabled = 1 gpgcheck = 0 metadata_expire = 86400 proxy = [cisco2.6.2.vts262-os-queens] name=cisco2.6.2.vts262-os-queens baseurl=https://devhub.cisco.com/artifactory/vts-yum-dev/2.6.2.vts262-os-queens username=<user> password=<pass> enabled = 1 gpgcheck = 0 metadata_expire = 86400 proxy = VTSUpgradeNewPackages: ' "cisco-vts-puppet-tripleo", "lldpd", "vts-lldpd-configure", "cisco-vts-puppet-neutron"```
Step 5	Deploy the overcloud by including the following items to the deploy command line:	The environment file that is shown in Step 4. The relevant RH registration environment file. For example: ```cat deploy-overcloud-vts-ml2.sh #!/bin/bash openstack overcloud deploy \ --templates \ -e /home/stack/templates/node-info.yaml \ -e /home/stack/templates/overcloud_images.yaml \```

	Command or Action	Purpose
		-e /usr/share/openstack-tripleo-heat-templates/environments/network-isolation.yaml \ -e /usr/share/openstack-tripleo-heat-templates/environments/host-config-and-reboot.yaml \ -e /usr/share/openstack-tripleo-heat-templates/environments/docker.yaml \ -e /usr/share/openstack-tripleo-heat-templates/environments/docker-ha.yaml \ -e /home/stack/templates/rhel-registration/environment-rhel-registration.yaml \ -e /home/stack/templates/rhel-registration/rhel-registration-resource-registry.yaml \ -e /home/stack/templates/network-environment.yaml \ -e /home/stack/templates/neutron-ml2-cisco-vts-262.yaml \ --stack vts \ --debug \ --log-file oclogs/overcloudDeploy_$(date +%m_%d_%y__%H_%M_%S).log \ --ntp-server ntp.esl.cisco.com \ --verbose --timeout 100

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 23, for details.

  To generate VTSR day0 config, we need to create the site on VTC GUI first and use the generated site-id in vtsr day0 config file to generate the vtsr day0 iso file.

- Deploying the VTSR on the VMM. See Deploying VTSR on OpenStack, on page 26 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 2.6, you need to generate the VTSR ISO again with Monit details in the vtsr_template.cfg file. See also, Upgrading VTSR.

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

Step 2 Make a copy of the new 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="10.85.88.152"
#VTS_IPV6_ADDRESS="a1::10"
# 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="cisco"
PASSWORD="cisco123"

# Mandatory Management-VRF name for VTSR.
VTS_MANAGEMENT_VRF="vtsr-mgmt-vrf"

# 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="19.1.0.20"
NODE1_MGMT_NETWORK_IP_NETMASK="255.255.255.0"
NODE1_MGMT_NETWORK_IP_GATEWAY="19.1.0.1"
#NODE1_MGMT_NETWORK_IPV6_ADDRESS="a1::20"
#NODE1_MGMT_NETWORK_IPV6_NETMASK="64"
#NODE1_MGMT_NETWORK_IPV6_GATEWAY="a1::1"
NODE1_UNDERLAY_NETWORK_IP_ADDRESS="19.0.128.20"
NODE1_UNDERLAY_NETWORK_IP_NETMASK="255.255.255.0"
NODE1_UNDERLAY_NETWORK_IP_GATEWAY="19.0.128.1"
# 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"
```

```
# Operational username and password - optional
# These need to be configured to start monit on VTSR

#VTSR_OPER_USERNAME="monit-ro-oper"
# Password needs an encrypted value
# Example : "openssl passwd -1 -salt <salt-string> <password>"
#VTSR_OPER_PASSWORD="$1$cisco$b88M8bkCN2ZpXgEEc2sG9/"

# VTSR monit interval - optional - default is 30 seconds
#VTSR_MONIT_INTERVAL="30"

# 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="19.1.0.21"
#NODE2_MGMT_NETWORK_IP_NETMASK="255.255.255.0"
#NODE2_MGMT_NETWORK_IP_GATEWAY="19.1.0.1"
##NODE2_MGMT_NETWORK_IPV6_ADDRESS="a1::21"
##NODE2_MGMT_NETWORK_IPV6_NETMASK="64"
##NODE2_MGMT_NETWORK_IPV6_GATEWAY="a1::1"
#NODE2_UNDERLAY_NETWORK_IP_ADDRESS="19.0.128.21"
#NODE2_UNDERLAY_NETWORK_IP_NETMASK="255.255.255.0"
#NODE2_UNDERLAY_NETWORK_IP_GATEWAY="19.0.128.1"
#  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"

# VTS site uuid
VTS_SITE_UUID="abcdefab-abcd-abcd-abcd-abcdefabcdef"
```

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.

Note VTSR_OPER_USERNAME and VTSR_OPER_PASSWORD are mandatory to start Monit on VTSR.

VTSR_MONIT_INTERVAL is optional. It is 30 seconds, by default. See *Monitoring Cisco VTS* chapter in the *Cisco VTS User Guide* for details about Monit.

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.

Note Ensure that you log in as a root user.

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 31
- vtsr-underlay-ospf-template. See Applying OSPF Template, on page 32

- vtsr-underlay-isis-template. See Applying IS-IS Template, on page 32

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

```
# 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="10.85.88.152"
#VTS_IPV6_ADDRESS="a1::10"
# 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="cisco"
PASSWORD="cisco123"

# Mandatory Management-VRF name for VTSR.
VTS_MANAGEMENT_VRF="vtsr-mgmt-vrf"

# 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="19.1.0.20"
NODE1_MGMT_NETWORK_IP_NETMASK="255.255.255.0"
NODE1_MGMT_NETWORK_IP_GATEWAY="19.1.0.1"
#NODE1_MGMT_NETWORK_IPV6_ADDRESS="a1::20"
#NODE1_MGMT_NETWORK_IPV6_NETMASK="64"
#NODE1_MGMT_NETWORK_IPV6_GATEWAY="a1::1"
NODE1_UNDERLAY_NETWORK_IP_ADDRESS="19.0.128.20"
NODE1_UNDERLAY_NETWORK_IP_NETMASK="255.255.255.0"
NODE1_UNDERLAY_NETWORK_IP_GATEWAY="19.0.128.1"
# 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"

# Operational username and password - optional
# These need to be configured to start monit on VTSR
```

```
#VTSR_OPER_USERNAME="monit-ro-oper"
# Password needs an encrypted value
# Example : "openssl passwd -1 -salt <salt-string> <password>"
#VTSR_OPER_PASSWORD="$1$cisco$b88M8bkCN2ZpXgEEc2sG9/"

# VTSR monit interval - optional - default is 30 seconds
#VTSR_MONIT_INTERVAL="30"

# 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="19.1.0.21"
#NODE2_MGMT_NETWORK_IP_NETMASK="255.255.255.0"
#NODE2_MGMT_NETWORK_IP_GATEWAY="19.1.0.1"
##NODE2_MGMT_NETWORK_IPV6_ADDRESS="a1::21"
##NODE2_MGMT_NETWORK_IPV6_NETMASK="64"
##NODE2_MGMT_NETWORK_IPV6_GATEWAY="a1::1"
#NODE2_UNDERLAY_NETWORK_IP_ADDRESS="19.0.128.21"
#NODE2_UNDERLAY_NETWORK_IP_NETMASK="255.255.255.0"
#NODE2_UNDERLAY_NETWORK_IP_GATEWAY="19.0.128.1"
#  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"

# VTS site uuid
VTS_SITE_UUID="abcdefab-abcd-abcd-abcd-abcdefabcdef"
```

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:

Note loopback 1 for VTSR device is reserved for VTSR and docker communication. We recommended that you do not use it for VTSR while executing template script.

```
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
```

Applying IS-IS Template

Cisco VTS supports IS-IS as an underlay protocol for VTSR. You can apply the IS-IS underlay template using VTS CLI, to enable this feature.

You must configure Keychain before you apply the IS-IS template.

To apply IS-IS template:

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

Step 2 Apply Keychain template. Run the following command:

```
admin@VTC1:/opt/vts/bin# ./vts-cli.sh -applyTemplate vtsr-keychain-template
```

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

Note MD5 cryptography on VTSR and Cisco Nexus 9000 devices does not match. Select HMAC-MD5, if Cisco Nexus 9000 is using MD5.

```
Enter device name: vtsr01
Enter key-chain-name: AUTH
Enter key-id: 1
Enter password (if clear text password, precede password with ! character): !cisco123
Enter accept-lifetime-start-date in yyyy/mm/dd hh:mm:ss format, hh range is 0-23): 2018/02/28 12:00:00
Enter send-lifetime-start-date in yyyy/mm/dd hh:mm:ss format, hh range is 0-23): 2018/02/28 12:00:00
Enter cryptographic-algorithm(options: alg-hmac-sha1-12 or alg-md5-16 or alg-sha1-20 or alg-hmac-md5-16
 or alg-hmac-sha1-20): alg-hmac-md5-16
```

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-keychain-template successfully applied to device vtsr01
```

Note If you want to use Keychain with end date, you can use the vtsr-keychain-enddate-template.

Step 3 Apply the IS-IS Template. Run the following command:

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

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

```
Enter device name: vtsr01
Enter instance-name: 1
Enter is-type-level (value can be 1 or 2 or 1and2): 1
Enter mtu: 4352
Enter keychain-name: AUTH
Enter Network-Entity/Net-name(consist of an even number of octets,
                and be of the form 01.2345.6789.abcd.ef etc
                up to
                01.2345.6789.abcd.ef01.2345.6789.abcd.ef01.2345.67.):
47.0004.004d.0001.0001.0c28.0104.00
Enter physical-interface-name: GigabitEthernet0/0/0/0
Enter loopback-interface-number: 0
```

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-isis-template successfully applied to device vtsr01
```

Enabling Transparent QoS

To enable Transparent QoS:

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

Step 2 Apply the Transparent QoS Template. Run the following command:

```
admin@VTC1:/opt/vts/bin# ./vts-cli.sh -applyTemplate vts-cli -vppQos <enable/disable>
```

Installing VTF on OpenStack

Note **VTF-Vm mode is deprecated or no longer supported in any OpenStack or vCENTER deployments from VTS 2.6.2 onwards.**

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 23, 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 and OpenStack Queens.

Before you install VTF, do the following:

- 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).

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-pci-generic are supported. Choose an option from the drop-down.

  Note We recommend you use the VFIO driver (on compute nodes where data traffic will be in play) as it is a more robust and secure option. It will involve a reload of the compute as compared to choosing UIO driver option. For Controller with only VTF (for DHCP purposes) we recommend using the UIO driver.

- 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 drop-down.
- 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. After an upgrade, the old inventory_file will be available at "/opt/vts/install/old_version". 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

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.

Note In case of an upgrade, post upgrade, before you uninstall VTF, you need to setup SSH access (only required for OSPD Setup), and then uninstall VTF.

- 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.

Enabling OpenStack DHCP Server on Node(s) Running VTF

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]
VTS_IP="<IP or FQDN of VTS>"
VTS_USERNAME="<vts-username>"
VTS_PASSWORD=@@@@

# The VMM_NAME needs to correspond to the VMM Name registered in the VTS,
# Alternatively the VMM_ID can be used, which overrides the name setting
VMM_NAME="<Vmm Name>"
#VMM_ID="<Vmm ID>"

# VTS Router Underlay IP address(es)
#vtsr_ips='["1.1.1.1", "2.2.2.2"]'

### VTF V-Host (VPP) specific variables ###
[vts_v_hosts]
#<nova compute name> ansible_ssh_host="DNS name/IP of target host" vhost_type="compute"
host_ip="Underlay IP address" host_netmask_len="Underlay Netmask" net_gw="Underlay Gateway"
 underlay_if="ens224" interfaces='["eth1", "eth2"]' u_addresses='["List of routes on the
underlay"]' vif_type="tap"

# V-Host Group variables
[vts_v_hosts:vars]
#If not using a sudo capable user below, then please specify "ansible_sudo_pass"
ansible_ssh_user="root"
ansible_ssh_pass=@@@@
#ansible_ssh_private_key_file="~/.ssh/id_rsa"

config_method="static"
name_server="10.0.0.1"
#VTS address on the underlay. If not set, defaults to VTS_IP
#vts_u_address="11.0.0.1"

#max_hp_memory_perc=80
enable_workers=False
#pci_driver="vfio-pci"
#l2_mode=False

monit_username=monit-ro
monit_password=@@@@

monit_ssl=True


### Neutron Control Servers ###
[neutron_servers]
#<name> ansible_ssh_host="DNS name/IP of target host"
```

```
[neutron_servers:vars]
#os_version="Newton"
#If not using a sudo capable user below, then please specify "ansible_sudo_pass"
ansible_ssh_user="admin"
ansible_ssh_pass=@@@@
#ansible_sudo_pass=@@@

# VTS_USERNAME and PASSWORD can be overriden here, or the all group setting used
#VTS_USERNAME=<VTS username>
#VTS_PASSWORD=@@@@


### Grouping of host-groups behind the SSH Proxy ###
# List the host group names that are proxied by an SSH gateway
# Comment out when NOT using an SSH proxy
[proxied_hosts:children]
#neutron_servers
#vts_v_hosts

# Access parameters to the proxied hosts. **The password is that of the ssh proxy**
[proxied_hosts:vars]
ansible_ssh_pass=@@@@
# **THE FOLLOWING LINE IS NOT USER MODIFIABLE**
ansible_ssh_common_args='-o UserKnownHostsFile=/dev/null -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\']}}"'


### SSH Proxy node definition. Do not modify the group name ###
# Maximum of one proxy is currently supported
[proxy]
undercloud1 ansible_ssh_host="<director>"

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

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 IPtables rules are programmed correctly.

Verifying VTS Installation

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

- Verifying VTC VM Installation, on page 40
- Verifying VTSR Installation, on page 40
- Verifying VTF Installation, on page 41

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.

```
On VTSR262 based on XR 651.we have Mgmt under new Mgmt VRF.So Ping of Mgmt should be within VRF :
vrf vtsr-mgmt-vrf
address-family ipv4 unicast
!
address-family ipv6 unicast
!
RP/0/RP0/CPU0:vtsr01#ping 50.1.1.251 vrf vtsr-mgmt-vrf
Thu Aug 30 13:51:25.873 UTC
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 50.1.1.251, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/2 ms
```

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.

VMM_ID : It is the VMM-ID local to the site

Site_ID : As part of Multi-site support, we have to define the Site-ID which VTSR instance is managing VFG within that site. For more information, see *Multi-site support* section of the *Cisco VTS 2.6.2 User Guide* .

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.

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
- For vhost mode, connect to the compute and checkvpfa/vpp services or RPM packages.

If registration is successful, you should see the newly registered VTF IP (underlay IP) under VTF list (**Inventory > Virtual Forwarding Groups**).

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.

In case VTC and VTF are on different subnets, then verify whether routes to VTS are configured on the compute.

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 (**Inventory > Virtual Forwarding Groups**).

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.
- If you setup two nodes with different GUI Password and try setting up L2 HA, it will fail. You need to make sure that both the nodes have same password before setting up L2 HA.
- If you already have L2 HA, you can change the GUI Password from the GUI by logging in with VIP IP. This will change the GUI Password on both Master and Slave nodes. Changing the GUI password on master and slave nodes separately is not supported.

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 example:

```
root@vts:~# passwd admin
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
```

For an HA installation you must change the password on both Master and Slave with the command ‘passwd’. In an HA setup, you can change the passwords without uninstalling HA. This password change will not impact the HA setup as HA uses the GUI Password, which needs to be same on both Master and Slave nodes.

Note You can set different admin password on both the nodes, but make sure you remember and use the correct password to log in to the respective nodes.

Encrypting the Password

The password encryption tool (encpwd) is pre-installed on the Cisco VTS. To encrypt the password:

Run the following command:

```
$ encpwd 'clearTextPassword'
```

Note Any special characters in the password need to be preceded with \. For example,Cisco123! should be entered as Cisco123\!. For security reasons, we recommend that you clear the history from the command line to avoid the clear texts to VTC are configured on compute.