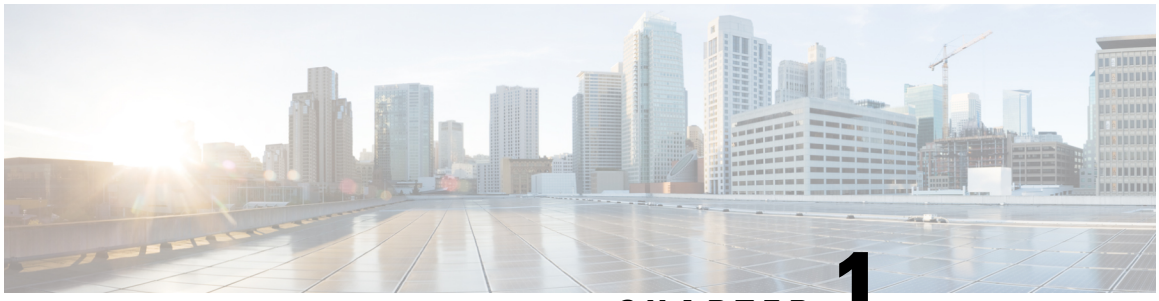
Cisco Virtual Topology System (VTS) 2.6.2 Installation Guide

First Published: 2019-04-02

Last Modified: 2019-04-02

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

CONTENTS



CHAPTER 1

Introduction

The Cisco Virtual Topology System (VTS) is a standards-based, open, overlay management and provisioning system for data center networks.

This document describes how to install the different components of Cisco Virtual Topology System (VTS).

- For information about installing Cisco VTS on an OpenStack environment, see Installing Cisco VTS in a Linux—OpenStack Environment, on page 9.
- For information about installing Cisco VTS on a VMware ESXi environment, see Installing Cisco VTS on a VMware Environment, on page 39.

For information about the prerequisites to install Cisco VTS, see Prerequisites , on page 3.

For information about installing Cisco VTS in High Availability mode, see Installing VTS in High Availability Mode, on page 59

You can also install Cisco VTS without a Virtual Machine Manager (VMM). See the *Cisco VTS Developer Guide* for details.

For more information about Cisco VTS, see the product documentation available on Cisco.com.

- Introduction, on page 1

Introduction

The Cisco Virtual Topology System (VTS) is a standards-based, open, overlay management and provisioning system for data center networks.

This document describes how to install the different components of Cisco Virtual Topology System (VTS).

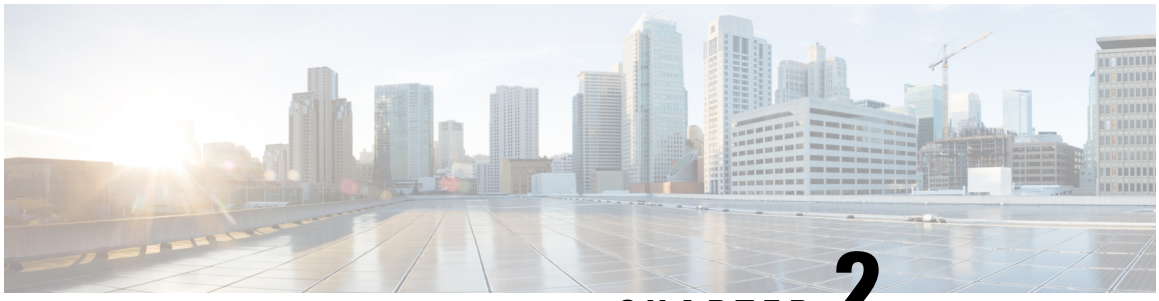
- For information about installing Cisco VTS on an OpenStack environment, see Installing Cisco VTS in a Linux—OpenStack Environment, on page 9.
- For information about installing Cisco VTS on a VMware ESXi environment, see Installing Cisco VTS on a VMware Environment, on page 39.

For information about the prerequisites to install Cisco VTS, see Prerequisites , on page 3.

For information about installing Cisco VTS in High Availability mode, see Installing VTS in High Availability Mode, on page 59

You can also install Cisco VTS without a Virtual Machine Manager (VMM). See the *Cisco VTS Developer Guide* for details.

For more information about Cisco VTS, see the product documentation available on Cisco.com.



CHAPTER 2

Prerequisites

This chapter provides information about the prerequisites for installing VTS components. It provides details about the system requirements, supported Virtual Machine Manager (VMM) and supported platforms.

System Requirements for VTC VM

The following table provides information about the minimum system requirements for the VTC virtual machine:

Requirement	Details
Disk space	48 GB
Logical CPUs	8
Memory	32 GB
Computing Host	Certified with Cisco UCS B-series, Cisco UCS C-series Rack Servers

System Requirements for VTSR

The following table gives details about the minimum system requirements for VTSR:

Note VTSR serves two purposes. It is required to enable VTS High Availability. It also acts as the control plane for the VTF. You need to install VTSR only if you consider enabling High Availability or if you plan to have a VTF in your set up.

Requirement	Details
Disk Space	77GB
Logical CPUs	14
Memory	48 GB RAM
Computing Host	Certified with Cisco UCS B-series, Cisco UCS C-series Rack Servers

System Requirements for VTF

The following table gives details about the minimum system requirements for the VTF virtual machine:

Requirement	Details
Disk Space	8 GB
CPUs	2
Memory	16 GB RAM
Server network interface card (NIC)	Intel DPDK-supported NIC

See OpenStack VTF vhost Mode Considerations, on page 103 for details about vhost Mode requirements.

Supported Virtual Machine Managers

Cisco VTS can be installed on the following supported versions of VMMs:

- OpenStack:
  - OpenStack Queens (17.0.x) is supported

	OpenStack Newton	OpenStack Queens
On RHEL	14.0.3	17.x

- vCenter:
  - vCenter/VMware ESXi 6.0 Update 2
  - vCenter/VMware ESXi 6.5 Update 1

Supported Platforms for VXLAN

The following tables provide information about the platforms that Cisco VTS support, and their roles.

Note Cisco VTS supports VXLAN overlays using the BGP EVPN control plane.

Role	Platform Supported
Top-of-rack (ToR) leaf switch	• Cisco Nexus 9300TX and 9300PX platform switches • Cisco Nexus 9332PQ and 93128TX switches • Cisco Nexus 9200 platform switches • Cisco Nexus 5600 platform switches • Cisco Nexus 9500 platform switches • Cisco Nexus 7x00 platform switches • Cisco Nexus 3100-V platform switches
Data center spine	• Cisco Nexus 9300TX and 9300PX platform switches • Cisco Nexus 9500 platform switches • Cisco Nexus 9200 platform switches • Cisco Nexus 7x00 Series switches • Cisco Nexus 5600 platform switches
Border leaf	• Cisco Nexus 9300TX and 9300PX platform switches • Cisco Nexus 9500 platform switches • Cisco Nexus 9200 platform switches • Cisco Nexus 5600 platform switches • Cisco Nexus 7x00 platform switches • Cisco Nexus 3100-V Platform switches
Data Center Interconnect (DCI)	• Cisco ASR 9000 Series Aggregation Services routers • Cisco Nexus 7x00 Series switches • Cisco Nexus 9300 platform switches

Fabric Extenders (FEX)	• Cisco Nexus C2248TP-E9500 • Cisco Nexus C2232PP FEX support is available for Cisco Nexus 9300, Cisco Nexus 5600, Cisco Nexus 9500 and Cisco Nexus 7x00 switches.
Hypervisor	• vCenter/VMware ESXi 6.0 Update 2 and vCenter/VMware ESXi 6.5 Update 1 • Red Hat Enterprise Linux 7.3 with KVM • Red Hat Enterprise Linux 7.5

Note Cisco Nexus 5672 does not interoperate with Cisco Nexus 93xx or 95xx.

The following table lists the software images supported for the different devices.

Table 1: Software Images Supported

Cisco Nexus 93xx	NX-OS Release 7.0(3)I5(1) and later.
Cisco Nexus 95xx	NX-OS Release 7.0(3)I5(1) and later.

Cisco Nexus 7x00	• Data center spine—NX-OS Release 8.1.(1) and later.
	• Data center interconnect (DCI):
	  • VRF Peering mode—NX-OS Release 7.3.1 and later.
	  • Integrated DCI mode—NX-OS Release 7.3.1 and later.

Cisco Nexus 5600	NX-OS Release 7.3(0)N1(1) and later.
Cisco ASR 9000	Cisco IOS XR Software Release 5.3.2 and later.

The following table lists the vPC modes supported for the different devices.

Note If Cisco Nexus 9000 series ToR is not configured with vPC related configuration, including peer-link, also known as a multichassis etherChannel trunk (MCT), you must not configure “feature vpc” on the ToR. This may bring loopback interface used for NVE to “admin down” state.

Table 2: vPC Modes Supported

Cisco Nexus 93xx	Server vPC
Cisco Nexus 95xx	Server vPC
Cisco Nexus 5600	Server vPC, FEX vPC, Enhanced vPC
Cisco Nexus 7000	Host vPC and single-homed host in port channel mode.

Supported Platforms for MPLS SR

The following tables provide information about the platforms that Cisco VTS support, and their roles.

Role	Platform Supported
Tor Leaf Switch	NCS5500
Spine	NCS5500
Hypervisor	RHEL 7.3 with KVM 7.5

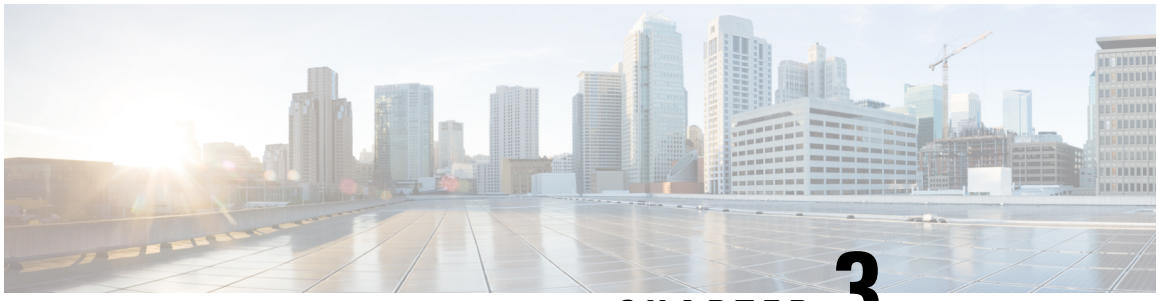
Table 3: Software Images Supported

Cisco NCS 5500	Cisco IOS XR Software Release 6.5.1

Supported Browsers

Cisco VTS supports the following browsers:

- Mozilla Firefox, version 47 and later.
- Google Chrome



CHAPTER 3

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 9 for details.
- Installing the Host Agent and the Open Stack Neutron Plugin.

  See Installing Host Agent, on page 16 and Registering OpenStack VMM, on page 15

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 10

To install VTC VM using the virt-manager application (Manual Configuration), see Installing VTC VM—Manual Configuration Using virt-manager Application, on page 12

To install VTC VM using VNC (Manual Configuration), see Installing VTC VM - Manual Configuration using VNC, on page 13

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, on page 105.

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, on page 105.

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 15, 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.

Note This procedure is supported only for OpenStack Newton.

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, on page 4 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

Note For OSPD 13, username and passphrase are not mandatory. Skip the warning message and continue.

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.

If there are VMM plugins and host-agent installation issues, please check the following files respectively on the VTC VM for further troubleshooting.

```
/opt/vts/log/{HOST_IP_ADDRESS}_ansible_logger.log
/opt/vts/log/hostagent-ansible/{HOST_IP_ADDRESS}_host_agent_install.log
```

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.

Installing Cisco VTS 262 Components in OSPD

For more information, see Cisco VTS 2.6.2 Components in OSPD

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 18, 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 21 or Deploying VTSR on VMware, on page 45, 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, on page 97.

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 -`.

In VTC L3HA scenario, cluster installation will configure loop back and ospf/isis configs on VTSRs based on the information provided in the cluster.conf file. No need to run these templates again for VTEP mode.

- vtsr-underlay-loopback-template. See Applying Loopback Template, on page 26
- vtsr-underlay-ospf-template. See Applying OSPF Template, on page 27
- vtsr-underlay-isis-template. See Applying IS-IS Template, on page 27

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

This section contains the following topics:

- Inband Installation
- Out of Band Installation

Inband Installation of VTF on OpenStack

Note VTF-Vm mode is deprecated or no longer supported in any OpenStack or vCENTER deployments from VTS 2.6.3 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 18, 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, on page 103 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 35
- Verifying VTSR Installation, on page 35
- Verifying VTF Installation, on page 36

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 13

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 13
- 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.

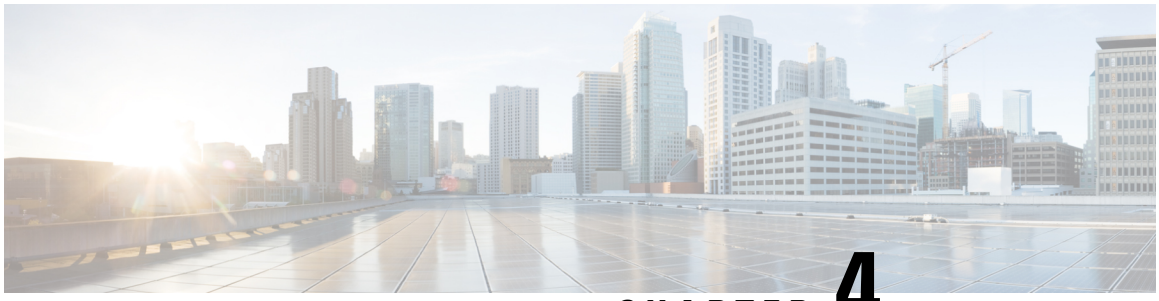
Encrypting the Password

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.



CHAPTER 4

Installing Cisco VTS on VMware

Installing Cisco VTS on a VMware Environment

Installing Cisco VTS on a VMware environment involves:

- Installing VTC VM on ESXi, on page 39
- Installing vCenter Plugin

Installing VTC VM on ESXi

To install VTC VM on an ESXi host:

Step 1 Connect to the ESXi host using the VMware vSphere Client.

Step 2 In the vSphere Client, select **File** > **Deploy OVF Template**. The Deploy OVF Template wizard appears.

Step 3 Specify the name and source location, and click Next.

Note You may place vtc.ovf and vtc.vmdk in different directories.

Step 4 Select the appropriate host to spawn the VTC VM.

Step 5 For VM disk format, use the default disk format settings (that is Thick Provision Lazy Zeroed).

Step 6 Map VTC network connectivity to appropriate port-groups on vSwitch/DVS.

- vNIC1—Used for VTC network management
- vNIC2—Used for VTC connectivity to VTF, VTSR

Step 7 Enter the following properties:

- Hostname—VTS Hostname.
- Management IPv4 Address—Management IP address for VTC. This IP address is used for VTC network management.
- Management IPv4Gateway—Management Gateway address
- Management IPv4 Netmask—Management IP Netmask
- Management IPv4 Method—DHCP / Static IP configuration for static IP .
- Management IPv6 Address—Management IP address for VTC. This IP address is used for VTC network management.
- Management IPv6 Gateway—Management Gateway address
- Management IPv6 Method—DHCP / Static IP configuration for static IP, or none.
- Management IPv6 Netmask—Management IP Netmask
- Underlay IPv4 Address—Underlay IP address. This is the IP address for internal network.
- Underlay IPv4 Gateway—Underlay Gateway IP
- Underlay IPv4 Method—DHCP / Static IP configuration for static IP.
- Underlay IPv4 Netmask—Underlay IP Netmask.
- Underlay IPv6 Address—Underlay IP address. This is the IP address for internal network.
- Underlay IPv6 Gateway—Underlay Gateway IP
- Underlay IPv6 Method—DHCP / Static IP configuration for static IP.

  Note Cisco VTS does not support IPv6 Underlay configuration. You must specify the Underlay IP6 Method value as **None** to avoid errors.

- Underlay IPv6 Netmask—Underlay IP Netmask.
- DNSv4—IP address of the DNS server.
- Domain—The DNS Search domain.
- NTPv4—NTP address. Can be same as gateway IP address.
- vts-adminPassword—Password for the vts-admin user. Password used to access VTC via SSH for vts-admin account.
- AdministrativeUser—The Administrator User. Enter administrative username.
- AdministrativePassword—Password for administrator user.

Note admin/admin is used to log into GUI for 1st time. The password will be changed during first time login into GUI

While creating admin domain with large number of devices per L2GWgroup or L3GWGroup, you must add devices in smaller batches.

Note that locally we have tried batch of 40 devices and the admin domain creation gets completed within 11 minutes.

Installing vCenter Plugin

The vCenter plugin gets installed when you register the VMM from the Cisco VTS GUI.

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.
- Mode—Whether the VMM has been registered as Trusted or Untrusted.
- API Endpoint Details. This is optional.
  - API Endpoint Details:
    - API Protocol:IP Address:Port—VMM service endpoint's IPv4/IP6 address and port.
    - Datacenter—The name of the datacenter for which Cisco VTS acts as the controller.
    - Admin User Name—Username of the vCenter VMM.
    - Admin Passphrase —Password of the vCenter VMM.

Step 4 Click **Register**.

After the VMM is registered successfully, the Plugin sections opens up.

Step 5 Enter the following in the Plugin details section:

- IP Address : Port
- Admin User Name
- Admin Passphrase

Note If you had entered the API endpoint details, the Plugin details will get populated automatically.

Notes Regarding VMware vSphere Distributed Switch

The following points need to be taken care of while you create a vDS.

Note

- All the ToRs in the inventory should be part of the vDS.
- One vDS can represent one or more ToRs.
- All the hosts that are connected to a particular ToR should be part of the same vDS.

For Non-vPC Specific Configuration

If you are not using vPC on the leaves:

- Associate one or more leafs per vDS.
- Attach the hosts data interface to the vDS uplinks.

Note See VMware documentation for the detailed procedure.

For vPC Specific Configuration

If you are using vPC on the leaves:

Step 1 Create one vDS switch for one or more vPC pairs.

Step 2 Enable enhanced LACP.

See VMware documentation for the detailed procedure.

Step 3 Create a Link Aggregation Group for each vDS.

See VMware documentation for the detailed procedure.

Step 4 You may remove the default port group that gets created as it will not be used.

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 18, 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 21 or Deploying VTSR on VMware, on page 45, 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, on page 97.

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 VMware

Deploying the VTSR.ova is similar to XRNC.

Step 1 Generate an ISO file for the VTSR VM. See Generating an ISO for VTSR, on page 18 .

Step 2 In the vSphere Client, select **File** > **Deploy OVF Template**. The Deploy OVF Template wizard appears.

Step 3 Select VTSR.ova from the source location, and click **Next**.The OVF template details are displayed.

Step 4 Click **Next** to specify the destination. Enter the following details:

- Name for the VM
- Folder or datacenter where the VM will reside

Step 5 Click **Next** to select the storage location to store the files for the template. The default values for virtual disk format and VM Storage Policy need not be changed.

Step 6 Click **Next** to set up the networks. Specify the first network as the Underlay Network and the second network as the Management Network.

Step 7 Click **Next**. Review the settings selections.

Step 8 Click **Finish** to start the deployment.

Step 9 After the deployment is complete, edit the VM settings. Add a CD/DVD Drive selecting Datastore ISO file and point to the vtsr.iso file which was generated and uploaded to the host.

Step 10 Power on the VM.

Step 11 To ensure VTSR is configred with the proper Day Zero configuration, SSH to VTSR and then run:

```
RP/0/RP0/CPU0:vtsr01-vcenter#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 12 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 Step11.

An output 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
```

Do not press or Enter key when the VTSR is loading or getting registered with VTC. For vCenter, VTSR may take approximately 30-45 minutes to come up.

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 -`.

In VTC L3HA scenario, cluster installation will configure loop back and ospf/isis configs on VTSRs based on the information provided in the cluster.conf file. No need to run these templates again for VTEP mode.

- vtsr-underlay-loopback-template. See Applying Loopback Template, on page 26
- vtsr-underlay-ospf-template. See Applying OSPF Template, on page 27
- vtsr-underlay-isis-template. See Applying IS-IS Template, on page 27

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
```

Installing VTF on vCenter

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 18, 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 vCenter supports VTF in VTEP mode only.

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 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 VTEP from the drop down.

Step 2 Go to **Host Inventory** and edit the host on which VTF (VTEP mode) installation needs to be done.

Step 3 On Host Details, fill in all fields.

Ensure that you review the tooltips for important information about the entries.

Step 4 Select the Virtual Switch. You have the following options:

- Not Defined
- DVS
- vtf-vtep

To install VTF, select vtf-vtep

Step 5 Enter the VTF details.

- Underlay VLAN ID
- Underlay bridge/portgroup on DVS—This is the port group towards the fabric to which the VTF underlay interface will be connected. This needs to be created in advance on vCenter.
- Internal Bridge/Portgroup—This is the DvS portgroup towards Virtual Machines and should be also created in advance on vCenter. This portgroup should be setup as trunk, and security policy should allow Promiscuous mode, Mac address changes and Forged transmits (Set to Accept).

- Datastore—This is the datastore where the vmdk of the VTF VM will be stored, specify the datastore on the VTF host that you want to use.

Set up of the underlay ToR and the corresponding port-group on the DVS has to be done manually on vCenter.

Step 6 Verify the interfaces information.

Step 7 Check the **Install VTF on Save** check box, and click Save.

Step 8 Check the installation status in the Host Inventory page.

Step 9 Check the VTF registration status on **Inventory > Virtual Forwarding Groups** page.

Uninstalling VTF in a vCenter Environment

Before you VTF uninstall, go to **Inventory > Virtual Forwarding Groups** to verify that VTF is shown in Virtual Forwarding Groups page.

To uninstall VTF

Step 1 Go to Host Inventory, and edit the host to change Virtual Switch type from vtf-vtep to not-defined.

Step 2 Click **Save**.

Step 3 Check the uninstallation status on the Host Inventory page to verify whether Installation status is unchecked and Virtual Switch is not-defined.

Step 4 Go to the **Inventory > Virtual Forwarding Groups** page, to verify that it does not show VTF that you uninstalled.

Step 5 Go to vCenter using vSphere Web Client.

Step 6 Go to Hosts and Clusters, click the VTF VM that got uninstalled from VTS GUI.

Step 7 Power off the VTF VM.

Step 8 Delete the VTF from disk

Verifying VTS Installation

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

- Verifying VTC VM Installation, on page 35
- Verifying VTSR Installation, on page 35
- Verifying VTF Installation, on page 36

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 13

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 13
- 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.

Troubleshooting VTS Login Issues

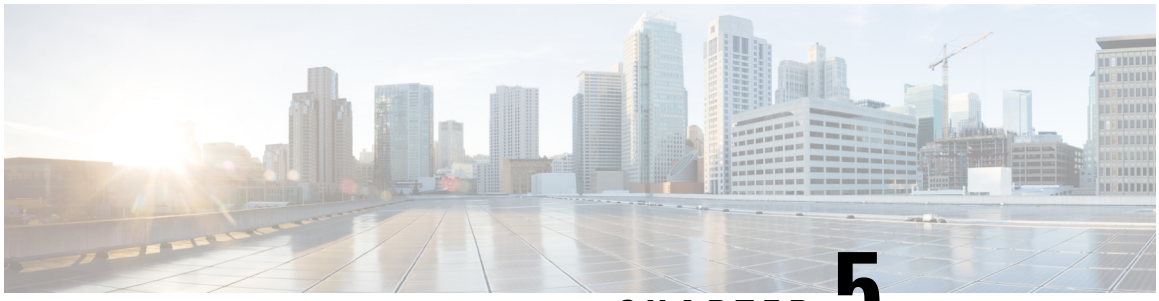
When you are unable to log in into VTS via CLI, check for the following points:

As a part of security compliance, it is recommended that you should remember your admin password. If you forget the admin password, then you should use another user account and change the password as a root user.

VTS installs an SSH Guard application on the system by default.

When you enter a wrong password consecutively for 4 times, the SSH Guard application temporarily blocks the IP that has authentication failures for approximately 7 to 10.5 minutes. If you are trying to enter the wrong password again, the blocking time doubles for each set of 4 failed logins.

For example, a host with IP of 192.0.2.1 is blocked when you enter a wrong password for 4 times within a period of 20 minutes interval. The IP is unblocked from the first set of login failures within 7 to 10.5 minutes due to periodic interval checks. When you try to enter the wrong password again for 4 consecutive times, the blocking time doubles upto 14 minutes.

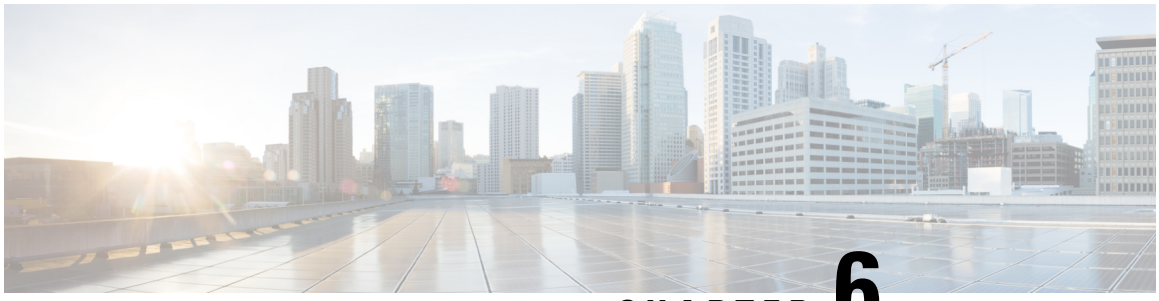


CHAPTER 5

Post-Installation Tasks

Post Installation of VTS

See the *Getting Started with Cisco Virtual Topology System* chapter in the *Cisco VTS User Guide* for details about the tasks that you need to perform after you install Cisco VTS.



CHAPTER 6

Installing VTS in High Availability Mode

This chapter provides detailed information about installing VTS in high availability (HA) mode. It details the procedure to enable VTS L2 and VTS L3.

See Enabling VTS L2 High Availability, on page 60 for the detailed procedure to enable VTS L2 HA.

See Enabling VTS L3 High Availability for Underlay and Management Networks for the detailed procedure to enable VTS L3 HA.

Important Notes regarding updating the cluster.conf file:

- master_name and slave_name can not be the same
- master_network_interface and slave_network_interface are interface names of VTC1 and VTC2 where the real IP resides. They should be the same.
- If you are using VTF's, fill in vip_private and private_network_interface fields. Otherwise, leave these two fields blank.
- Private_network_interface is the secondary interface names of VTC1 and VTC2 on the private network that VTF is also on.
- vip_private is the vip for the VTS master's private interface.
- private_gateway is the gateway for the private network of your vip_private.

This chapter has the following sections.

Enabling VTS L2 High Availability

To enable VTC L2 HA, VTC1 and VTC2 must be on the same subnet.

Spawn two VTC VMs. At a minimum, you would need to have three IP addresses for VTC. One for VTC1, One for VTC2, one for the public Virtual IP (VIP). If you are using VTFs, you will also need one for the private VIP, which other devices on the private network such as the VTF can reach.

Note Cisco VTS supports dual stack clusters for L2 HA. Have both the VTCs (vts01 and vts02) installed and configured with IPv6 & IPv4 address for dual stack to be supported. Both of the VTCs should be reachable by any means with IPv6 address or IPv4 address.

Note Before enabling HA, make sure that both VTC 1 and VTC 2 have the same password. If not, go to the VTC GUI and do a change password on newly brought up VTC, to make the password identical with that of the other VTC . When you upgrade a VTC / bring up a new VTC / do a hardware upgrade of VTC host, you should make sure that password is the same.

Enabling VTS L2 HA involves:

- Setting up the VTC Environment, on page 60
- Enabling VTC High Availability, on page 61
- Registering vCenter to VTC, on page 62
- Enabling VTSR High Availability, on page 63

Setting up the VTC Environment

You need to set up the VTC environment before you run the high availability script.

Step 1 Create a copy of cluster.conf file from cluster.conf.tmpl, which is under the /opt/vts/etc directory. For example:

```
admin@vts01:~$ cd /opt/vts/etc

admin@vts01:~$ sudo copy cluster.conf.tmpl cluster.conf
```

Step 2 Specify the VIP address and the details of the two nodes in cluster.conf file . For example:

```
admin@vts01:/var/# cd /opt/vts/etc/

admin@vts01/etc# sudo vi cluster.conf


###Virtual Ip of VTC Master on the public interface. Must fill in at least 1
vip_public=172.23.92.202
vip_public_ipv6=2001:420:10e:2015:c00::202
```

```
###VTC1 Information. Must fill in at least 1 ip address
master_name=vts01
master_ip=172.23.92.200
master_ipv6=2001:420:10e:2015:c00::200



###VTC2 Information. Must fill in at least 1 ip address
slave_name=vts02
slave_ip=172.23.92.201
slave_ipv6=2001:420:10e:2015:c00:201



###In the event that a network failure occurs evenly between the two routers, the cluster needs an
outside ip to determine where the failure lies
###This can be any external ip such as your vmm ip or a dns but it is recommended to be a stable ip
 within your environment
###Must fill in at least 1 ip address
external_ip=171.70.168.183
external_ipv6=2001:420:200:1::a
```

Note The two nodes communicate each other using VIP IP, and user can use VIP address to login to Cisco VTS UI. You will be directly logged in to the master node, when you use VIP IP address. Make sure that you specify the correct host name, IP Address, and interface type.

Enabling VTC High Availability

You must run the cluster_install.sh script on both VTCs to enable high availability.

Step 1 Run the cluster installer script /opt/vts/bin/cluster_install.sh on both VTC1 and VTC2 . For example:

```
admin@vts02:/opt/vts/etc$ sudo su  -

[sudo] password for admin:

root@vts02:/opt/vts/etc$ cd ../bin

root@vts02:/opt/vts/bin# ./cluster_install.sh
172.23.92.200 vts01
172.23.92.201 vts02
2001:420:10e:2015:c00::200 vts01
2001:420:10e:2015:c00::201 vts02

Change made to ncs.conf file. Need to restart ncs

Created symlink from /etc/systemd/system/multi-user.target.wants/pacemaker.service to
/lib/systemd/system/pacemaker.service.

Created symlink from /etc/systemd/system/multi-user.target.wants/corosync.service to
/lib/systemd/system/corosync.service.

Both nodes are online. Configuring master

Configuring Pacemaker resources

Master node configuration finished
```

```
HA cluster is installed
```

Step 2 Check the status on both the nodes to verify whether both nodes online, and node which got installed first is the master, and the other, slave. For example:

```
admin@vts02:/opt/vts/log/nso$ sudo crm status

[sudo] password for admin:

Last updated: Mon Apr 10 18:43:52 2017          Last change: Mon Apr 10 17:15:21 2017 by root via
crm_attribute on vts01

Stack: corosync

Current DC: vts01 (version 1.1.14-70404b0) - partition with quorum

2 nodes and 4 resources configured


Online: [ vts01 vts02 ]


Full list of resources:


Master/Slave Set: ms_vtc_ha [vtc_ha]

     Masters: [ vts02 ]

     Slaves: [ vts01 ]

ClusterIP      (ocf::heartbeat:IPaddr2):       Started vts02

ClusterIPV6    (ocf::heartbeat:IPaddr2):       Started vts02
```

Registering vCenter to VTC

To do this:

Step 1 Log in to VCSA.

Step 2 Go to **Networking > Distributed Virtual Switch > Manage > VTS**.

Note For vCenter 6.5, the VTS comes under Configure tab.

Step 3 Click on System Configuration

Step 4 Enter the following:

- VTS IP—This is the Virtual public IP address.
- VTS GUI Username
- VTS GUI Password

Step 5 Click Update.

Enabling VTSR High Availability

You need to enable VTSR HA, if you have VTFs in your setup. For information about enabling VTSR HA, see Installing VTSR in High Availability Mode, on page 88.

Enabling VTS L3 High Availability for Underlay and Management Networks

This section describes the procedure to enable VTC L3 HA. In L3 HA environments, VTCs are located on separate overlay networks.

Before enabling VTC L3 HA, make sure that the following requirements are met:

Step 1 Both VTC 1 and VTC 2 must have the same password. If they are not the same, change the password via the VTC GUI to make it identical with that of the other VTC. When you upgrade a VTC, bring up a new VTC, or do a hardware upgrade of a VTC host, remember to make sure that the passwords match.

Step 2 VTC1 and VTC2 are deployed in two different networks VTC1 is able to ping VTC 2 and wise versa via both underlay and overlay networks.

We recommend that you add static routes in /etc/network/interfaces on the VTCs. For example:

```
post-up route add -net 44.44.44.0/24 gw 10.10.10.1 dev eth1
post-up route add -net 10.10.10.0/24 gw 44.44.44.1 dev eth1
```

Step 3 VTSR installation is mandatory for VTC L3 HA to work. Although the VTSR(s) will not be able to register to the master VTC's Virtual IP Address (VIP) initially, the master VTC must still be able to reach all VTSRs in order to configure them to support L3HA.

Step 4 A site-id is required and need to be updated in VTSR_template file before generating the VTSR iso file and is generated by either:

1. Creating a site directly from the VTC1 GUI.
2. Using the uuidgen linux command to generate a 32 character site id.

Note In Dual stack scenario VTC supports dual stack but VTSR can only be either IPV4 or IPv6.

In both cases, note down the site id for use in the below section Deploying VTSR on VMware, on page 45 and Installing VTSR, on page 18. In the second case the site id will also be required to update the VTC master site configuration and associate the UUID to the site after the HA process is complete.

The implementation of L3 HA is done for both management and underlay networks or management network only. For VIP ip reachability across networks the master VTC will configure BGP and route policies between the VTSRs and their respective directly connected TORs based on the user network settings provided in cluster.conf. Routing policy including tagging helps to migrate VIP IPs across the VTSRs during VTC HA.

Setting up the VTC Environment

You need to set up the VTC environment before you run the high availability script.

Step 1 Modify the /opt/vts/etc/cluster.conf_tmpl file on both the VTCs. And rename to cluster.conf .A sample modified file is given below:

Both the VTCs must have the identical information in the cluster.conf file.

```
###Virtual Ip of VTC Master on the public interface. Must fill in at least 1
vip_public= 192.168.10.254
vip_public_ipv6=
###VTC1 Information. Must fill in at least 1 ip address
master_name=Onion-VTC1
master_ip= 60.60.60.10
master_ipv6=
###VTC2 Information. Must fill in at least 1 ip address
slave_name= Onion-VTC2
slave_ip= 70.70.70.10
slave_ipv6=
###In the event that a network failure occurs evenly between the two routers, the cluster needs an
outside ip to determine where the failure lies
###This can be any external ip such as your vmm ip or a dns but it is recommended to be a stable ip
 within your environment
###Must fill in at least 1 ip address
external_ip= 81.81.81.1
external_ipv6=
###If you intend to use a virtual topology forwarder (VTF) in your environment, please fill in the
vip for the underlay as well as the underlay gateway. Otherwise leave blank.
###Virtual Ip of VTC Master on the private interface. You can fill in ipv4 configuration, ipv6, or
both if you use both
vip_private= 45.45.45.10
private_gateway=10.10.10.1
vip_private_ipv6=
private_gateway_ipv6=
###If you have your vtc's in different subnets, vtsr will need to be configured to route traffic and
 the below section needs to be filled in
###If you have your vtc's on the same subnet, the below section should be skipped
###Name of your vrf. Example: VTS_VIP
vrf_name= mgmt-vrf
###Ip of your first Vtsr. Example: 11.1.1.5
vtsr1_mgmt_ip=60.60.60.15
vtsr1_mgmt_ipv6=
###List of neighbors for vtsr1, separated by comma. Example: 11.1.1.1,11.1.1.2
vtsr1_bgp_neighbors=  11.11.11.11
vtsr1_bgp_neighbors_ipv6=
###Ip of your second Vtsr. Example: 12.1.1.5
vtsr2_mgmt_ip= 70.70.70.15
vtsr2_mgmt_ipv6=
###List of neighbors for vtsr2, separated by comma. Example: 12.1.1.1,12.1.1.2
vtsr2_bgp_neighbors= 12.12.12.12
vtsr2_bgp_neighbors_ipv6=
###Username for Vtsr
vtsr_user= admin
###Vtsr ASN information
remote_ASN= 6500
local_ASN= 6501
###Vtsr BGP information
bgp_keepalive= 10
bgp_hold= 30
###Update source for Vtsr1 (i.e. loopback)
vtsr1_update_source=loopback0
```

```
###Update source for Vtsr2 (i.e. loopback)
vtsr2_update_source=loopback0
###Router BGP Id for Vtsr1
vtsr1_router_id= 21.21.21.21
###Router BGP Id for Vtsr2
vtsr2_router_id= 31.31.31.31
###Ipv4 Route Distinguisher Loopback for IPv6 Vtsr1 (if VTSR has a defined IPv6 management address,
 an ipv4 loopback address will be needed for the route distinguisher)
vtsr1_rd_loopback_name=
vtsr1_rd_loopback_address=
###Ipv4 Route Distinguisher Loopback for IPv6 Vtsr2
vtsr2_rd_loopback_name=
vtsr2_rd_loopback_address=

###XRVR1 name
vtsr1_name= vtsr01
###XRVR2 name
vtsr2_name= vtsr02
###If you plan on having your VTC's on different subnets and intend to use a virtual topology forwarder
 (VTF) in your environment,
### please fill out the following fields. Otherwise, leave blank
###List of neighbors for vtsr1, separated by comma. Example: 2.2.2.2,2.2.2.3
vtsr1_underlay_neighbors= 2.2.2.2
vtsr1_underlay_neighbors_ipv6=
###List of neighbors for vtsr2, separated by comma. Example: 3.3.3.2,3.3.3.3
vtsr2_underlay_neighbors= 6.6.6.6,9.9.9.9
vtsr2_underlay_neighbors_ipv6=
###OSPF Parameters
ospf_id_v4=100
ospf_id_v6=
area=0.0.0.0
default_cost=10
###ISIS Parameters
isis_id=
is_type=
lsp_mtu=
key_chain_id=
key_id=
cryptographic_algorithm=
#Network name (consist of an even number of octets and be of the form 01.2345.6789.abcd.ef)
vtsr1_network_entity=
vtsr2_network_entity=

###Directly connected Tor information for Vtsr1
vtsr1_directly_connected_device_ip= 10.10.50.3
vtsr1_directly_connected_device_ipv6=
vtsr1_directly_connected_device_user= admin
vtsr1_directly_connected_device_neighbors= 21.21.21.21
vtsr1_directly_connected_device_neighbors_ipv6=
vtsr1_directly_connected_ospf=
vtsr1_directly_connected_router_id=2.2.2.2
vtsr1_directly_connected_update_source=loopback0
###Directly connected Tor information for Vtsr2
vtsr2_directly_connected_device_ip= 10.10.50.7
vtsr2_directly_connected_device_ipv6=
vtsr2_directly_connected_device_user=admin
vtsr2_directly_connected_device_neighbors=31.31.31.31
vtsr2_directly_connected_device_neighbors_ipv6=
vtsr2_directly_connected_ospf=
vtsr2_directly_connected_router_id=6.6.6.6
vtsr2_directly_connected_update_source=loopback0
###VPC Peer information if any. Otherwise leave blank
vtsr1_vpc_peer_ip=
vtsr1_vpc_peer_ipv6=
vtsr1_vpc_peer_user=
```

```
vtsr1_vpc_peer_ospf=
vtsr1_vpc_peer_router_id=
vtsr1_vpc_peer_update_source=
vtsr2_vpc_peer_ip=
vtsr2_vpc_peer_ipv6=
vtsr2_vpc_peer_user=
vtsr2_vpc_peer_ospf=
vtsr2_vpc_peer_router_id=
vtsr2_vpc_peer_update_source=
###VTC Underlay Addresses
vtc1_underlay= 10.10.10.10
vtc2_underlay= 44.44.44.44
vtc1_underlay_ipv6=
vtc2_underlay_ipv6=
##Gateway of secondary L3 underlay
vtc2_private_gateway=44.44.44.1
#vtc2_private_gateway_ipv6=
```

Step 2 Make sure VTSR configuration ISOs are up to date with above configurations. For example:

Note # The VTS_REGISTRATION_PASSWORD and VTS_SITE_UUID values are VTC UI password and SITE ID those are created on VTC above respectively

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="192.168.10.254"
VTS_IPV6_ADDRESS=
# 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"

# Mandatory Management-VRF name for VTSR.
VTS_MANAGEMENT_VRF="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="60.60.60.15"
```

```
NODE1_MGMT_NETWORK_IP_NETMASK="255.255.255.0"
NODE1_MGMT_NETWORK_IP_GATEWAY="60.60.60.1"
NODE1_MGMT_NETWORK_IPV6_ADDRESS=
NODE1_MGMT_NETWORK_IPV6_NETMASK=
NODE1_MGMT_NETWORK_IPV6_GATEWAY=
NODE1_UNDERLAY_NETWORK_IP_ADDRESS="10.10.10.33"
NODE1_UNDERLAY_NETWORK_IP_NETMASK="255.255.255.0"
NODE1_UNDERLAY_NETWORK_IP_GATEWAY="10.10.10.1"
# 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="admin"
# Password needs an encrypted value
# Example : "openssl passwd -1 -salt <salt-string> <password>"
VTSR_OPER_PASSWORD="$1$cisco$Qv2TLtPNI3jqwXMOA3M3f0/"

# 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="70.70.70.15"
NODE2_MGMT_NETWORK_IP_NETMASK="255.255.255.0"
NODE2_MGMT_NETWORK_IP_GATEWAY="70.70.70.1"
NODE2_MGMT_NETWORK_IPV6_ADDRESS=
NODE2_MGMT_NETWORK_IPV6_NETMASK=
NODE2_MGMT_NETWORK_IPV6_GATEWAY=
NODE2_UNDERLAY_NETWORK_IP_ADDRESS="44.44.44.15"
NODE2_UNDERLAY_NETWORK_IP_NETMASK="255.255.255.0"
NODE2_UNDERLAY_NETWORK_IP_GATEWAY="44.44.44.1"
#  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"
```

Enabling VTC High Availability

To enable VTC high availability, perform the following steps.

Note Step 1 to 3 have to be run on both VTCs. Step 4 must be run only on the node that you want to make the active VTC.

Step 1 SSH to VTC 1 and VTC 2

Step 2 Go to the following directory:

```
cd /opt/vts/etc/
```

Step 3 Copy the cluster.conf file to /opt/vts/etc on both VTC 1 and VTC 2.

Step 4 Go to the following directory:

```
cd /opt/vts/bin
```

a) Run the sudo ./cluster_install.sh command. For example:

```
admin@Onion-VTC1:/opt/vts/bin#sudo  ./cluster_install.sh
```

You will be asked to provide the vtsr password. vtsr password is the password for VTSR1 and VTSR2. In addition to this, you will also be asked for the passwords for the switches directly connected to VTSR1 and VTSR2. And you will be prompted to run cluster_install.sh on VTC2 as well A message similar to below:

```
Please run cluster_install.sh on Onion-VTC2. Will wait until finished ==> At this point on VTC2
run the cluster install script
```

b) Run the sudo ./cluster_install.sh command. For example:

```
admin@Onion-VTC2:/opt/vts/bin#sudo  ./cluster_install.sh
```

You will be asked to provide the vtsr password. vtsr password is the password for VTSR1 and VTSR2. In addition to this, you will also be asked for the passwords for the switches directly connected to VTSR1 and VTSR2.An output similar to what is given below is displayed:

```
Change made to ncs.conf file. Need to restart ncs
Finding running docker container ID
263f311dbdff
Created symlink from /etc/systemd/system/multi-user.target.wants/pacemaker.service to
/lib/systemd/system/pacemaker.service.
Created symlink from /etc/systemd/system/multi-user.target.wants/corosync.service to
/lib/systemd/system/corosync.service.
Retrieving and storing node certificates
Retrieving master certificate
Retrieving slave certificate
Importing master certificate
Importing slave certificate
```

```
HA cluster is installed
admin@Onion-VTC2:/opt/vts/bin#
```

c) Once this is over on VTC2 ,Automatically on VTC1 the process will resume and you will see an output similar to what is given below :

```
2018-08-25 19:33:31,358 - INFO - Setup finished at 19:33:31
Configuring Pacemaker resources
Master node configuration finished
Retrieving and storing node certificates
Retrieving master certificate
Retrieving slave certificate
Importing master certificate
Importing slave certificate
HA cluster is installed.
admin@Onion-VTC1:/opt/vts/bin#
```

Enabling VTS L3 High Availablity Management Network Only

This section describes the procedure to enable VTC L3 HA. In L3 HA environments, VTCs are located on separate overlay networks.

Before enabling VTC L3 HA, make sure that the following requirements are met

:

Step 1 Both VTC 1 and VTC 2 must have the same password. If they are not the same, change the password via the VTC GUI to make it identical with that of the other VTC. When you upgrade a VTC, bring up a new VTC, or do a hardware upgrade of a VTC host, remember to make sure that the passwords match.

Step 2 VTC1 and VTC2 are deployed in two different networks VTC1 is able to ping VTC 2 and wise versa via both underlay and overlay networks.

We recommend that you add static routes in /etc/network/interfaces on the VTCs. For example:

```
sudo vi /etc/network/interfaces
post-up route add -net 44.44.44.0/24 gw 10.10.10.1 dev eth1
post-up route add -net 10.10.10.0/24 gw 44.44.44.1 dev eth1
```

Step 3 VTSR installation is mandatory for VTC L3 HA to work. Although the VTSR(s) will not be able to register to the master VTC's Virtual IP Address (VIP) initially, the master VTC must still be able to reach all VTSRs in order to configure them to support L3HA.

Step 4 A site-id is required and need to be updated in VTSR_template file before generating the VTSR iso file and is generated by either

1. Creating a site directly from the VTC1 GUI.
2. Using the uuidgen linux command to generate a 32 character site id.

   In both cases, note down the site id for use in the below section. In the second case the site id will also be required to update the VTC master site configuration and associate the UUID to the site after the HA process is complete.

   The implementation of L3 HA is done for both management and underlay networks or management networks only. For VIP ip reachability across networks the master VTC will configure BGP and route policies between the VTSRs

and their respective directly connected TORs based on the user network settings provided in cluster.conf. Routing policy including tagging helps to migrate VIP IPs across the VTSRs during VTC HA.

Enabling IOS XRv High Availability

You need to enable IOS XRv HA, if you have VTFs in your setup. For information about enabling IOS XRv HA, see Installing VTSR in High Availability Mode, on page 88.

Registering vCenter to VTC

To do this:

Step 1 Log in to VCSA.

Step 2 Go to **Networking > Distributed Virtual Switch > Manage > VTS**.

Note For vCenter 6.5, the VTS comes under Configure tab.

Step 3 Click on System Configuration

Step 4 Enter the following:

- VTS IP—This is the Virtual public IP address.
- VTS GUI Username
- VTS GUI Password

Step 5 Click Update.

Enabling VTS L3 High Availability for Management Network only

Setting up the VTC Environment for L3 High Availability Management

Use tthis procedure only to set up the VTC environment for L3 Hig Availabitiy (HA) management.

Step 1 Copy the /opt/vts/etc/cluster.conf_tmpl file to /opt/vts/etc/cluster.conf on both VTCs. The cluster.conf files must be identical on both VTCs. A sample modified file is given below:

a)

```
###Virtual Ip of VTC Master on the public interface. Must fill in at least 1
vip_public= 192.168.10.254
```

```
vip_public_ipv6=

###VTC1 Information. Must fill in at least 1 ip address
master_name=Onion-VTC1
master_ip=60.60.60.10
master_ipv6=

###VTC2 Information. Must fill in at least 1 ip address
slave_name= Onion-VTC2
slave_ip=70.70.70.10
slave_ipv6=

###In the event that a network failure occurs evenly between the two routers, the cluster needs an
outside ip to determine where the failure lies
###This can be any external ip such as your vmm ip or a dns but it is recommended to be a stable ip
 within your environment
###Must fill in at least 1 ip address
external_ip=81.81.81.1
external_ipv6=

###If you intend to use a virtual topology forwarder (VTF) in your environment, please fill in the
vip for the underlay as well as the underlay gateway. Otherwise leave blank.
###Virtual Ip of VTC Master on the private interface. You can fill in ipv4 configuration, ipv6, or
both if you use both
vip_private=
private_gateway=

vip_private_ipv6=
private_gateway_ipv6
###If you have your vtc's in different subnets, vtsr will need to be configured to route traffic and
 the below section needs to be filled in
###If you have your vtc's on the same subnet, the below section should be skipped

###Name of your vrf. Example: VTS_VIP
vrf_name=mgmt-vrf

###Ip of your first Vtsr. Example: 11.1.1.5
vtsr1_mgmt_ip=60.60.60.15
vtsr1_mgmt_ipv6=

###List of neighbors for vtsr1, separated by comma. Example: 11.1.1.1,11.1.1.2
vtsr1_bgp_neighbors= 11.11.11.11
vtsr1_bgp_neighbors_ipv6=

###Ip of your second Vtsr. Example: 12.1.1.5
vtsr2_mgmt_ip=70.70.70.15
vtsr2_mgmt_ipv6=

###List of neighbors for vtsr2, separated by comma. Example: 12.1.1.1,12.1.1.2
vtsr2_bgp_neighbors= 12.12.12.12
vtsr2_bgp_neighbors_ipv6=

###Username for Vtsr
vtsr_user= admin

###Vtsr ASN information
remote_ASN= 6500
local_ASN= 6501

###Vtsr BGP information
bgp_keepalive= 10
bgp_hold= 30

###Update source for Vtsr1 (i.e. loopback)
vtsr1_update_source=loopback0
```

```
###Update source for Vtsr2 (i.e. loopback)
vtsr2_update_source=loopback0

###Router BGP Id for Vtsr1
vtsr1_router_id= 21.21.21.21

###Router BGP Id for Vtsr2
vtsr2_router_id= 31.31.31.31

###Ipv4 Route Distinguisher Loopback for IPv6 Vtsr1 (if VTSR has a defined IPv6 management address,
 an ipv4 loopback address will be needed for the route distinguisher)
vtsr1_rd_loopback_name=loopback5
vtsr1_rd_loopback_address=41.41.41.41

###Ipv4 Route Distinguisher Loopback for IPv6 Vtsr2
vtsr2_rd_loopback_name=loopback5
vtsr2_rd_loopback_address=51.51.51.51


###XRVR1 name
vtsr1_name= vtsr01

###XRVR2 name
vtsr2_name= vtsr02

###If you plan on having your VTC's on different subnets and intend to use a virtual topology forwarder
 (VTF) in your environment,
### please fill out the following fields. Otherwise, leave blank

###List of neighbors for vtsr1, separated by comma. Example: 2.2.2.2,2.2.2.3
vtsr1_underlay_neighbors=
vtsr1_underlay_neighbors_ipv6=

###List of neighbors for vtsr2, separated by comma. Example: 3.3.3.2,3.3.3.3
vtsr2_underlay_neighbors=
vtsr2_underlay_neighbors_ipv6=

###OSPF Parameters
ospf_id_v4=
ospf_id_v6=
area=
default_cost=

###ISIS Parameters
isis_id=
is_type=
lsp_mtu=
key_chain_id=
key_id=
cryptographic_algorithm=
Network name (consist of an even number of octets and be of the form 01.2345.6789.abcd.ef)
vtsr1_network_entity=
vtsr2_network_entity=

###Directly connected Tor information for Vtsr1
vtsr1_directly_connected_device_ip=
vtsr1_directly_connected_device_ipv6=
vtsr1_directly_connected_device_user=
vtsr1_directly_connected_device_neighbors=
vtsr1_directly_connected_device_neighbors_ipv6=
vtsr1_directly_connected_ospf=
vtsr1_directly_connected_router_id=
vtsr1_directly_connected_update_source=
```

```
###Directly connected Tor information for Vtsr2
vtsr2_directly_connected_device_ip=
vtsr2_directly_connected_device_ipv6=
vtsr2_directly_connected_device_user=
vtsr2_directly_connected_device_neighbors=
vtsr2_directly_connected_device_neighbors_ipv6=
vtsr2_directly_connected_ospf=
vtsr2_directly_connected_router_id=
vtsr2_directly_connected_update_source=

###VPC Peer information if any. Otherwise leave blank
vtsr1_vpc_peer_ip=
vtsr1_vpc_peer_ipv6=
vtsr1_vpc_peer_user=
vtsr1_vpc_peer_ospf=
vtsr1_vpc_peer_router_id=
vtsr1_vpc_peer_update_source=

vtsr2_vpc_peer_ip=
vtsr2_vpc_peer_ipv6=
vtsr2_vpc_peer_user=
vtsr2_vpc_peer_ospf=
vtsr2_vpc_peer_router_id=
vtsr2_vpc_peer_update_source=

###VTC Underlay Addresses
vtc1_underlay=
vtc2_underlay=
vtc1_underlay_ipv6=
vtc2_underlay_ipv6=

##Gateway of secondary L3 underlay
vtc2_private_gateway=
vtc2_private_gateway_ipv6=
```

Step 2 Make sure VTSR configuration ISOs are up to date with above configurations. For example:

Note The VTSR values VTS_REGISTRATION_PASSWORD and VTS_SITE_UUID map to the VTC UI password and SITE ID on the VTCs, respectively.

```
# 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="192.168.10.254"
VTS_IPV6_ADDRESS=
# 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"

# Mandatory Management-VRF name for VTSR.
VTS_MANAGEMENT_VRF="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="60.60.60.15"
NODE1_MGMT_NETWORK_IP_NETMASK="255.255.255.0"
NODE1_MGMT_NETWORK_IP_GATEWAY="60.60.60.1"
NODE1_MGMT_NETWORK_IPV6_ADDRESS=
NODE1_MGMT_NETWORK_IPV6_NETMASK=
NODE1_MGMT_NETWORK_IPV6_GATEWAY=
NODE1_UNDERLAY_NETWORK_IP_ADDRESS="10.10.10.33"
NODE1_UNDERLAY_NETWORK_IP_NETMASK="255.255.255.0"
NODE1_UNDERLAY_NETWORK_IP_GATEWAY="10.10.10.1"
# 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="admin"
# Password needs an encrypted value
# Example : "openssl passwd -1 -salt <salt-string> <password>"
VTSR_OPER_PASSWORD="$1$cisco$Qv2TLtPNI3jqwXMOA3M3f0/"

# 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="70.70.70.15"
NODE2_MGMT_NETWORK_IP_NETMASK="255.255.255.0"
NODE2_MGMT_NETWORK_IP_GATEWAY="70.70.70.1"
NODE2_MGMT_NETWORK_IPV6_ADDRESS=
NODE2_MGMT_NETWORK_IPV6_NETMASK=
NODE2_MGMT_NETWORK_IPV6_GATEWAY=
NODE2_UNDERLAY_NETWORK_IP_ADDRESS="44.44.44.15"
NODE2_UNDERLAY_NETWORK_IP_NETMASK="255.255.255.0"
NODE2_UNDERLAY_NETWORK_IP_GATEWAY="44.44.44.1"
#  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"
```

Deploying VTSR VMs

To deploy VTSR 1 and VTSR 2, follow the below steps:

Step 1 Generate an ISO file for the VTSR VM. See Generating an ISO for VTSR, on page 18.

Step 2 After a VTSR is up and running, ssh to it's management ip address and make sure that the routes are added and both VTC 1 and VTC 2 can reach VTSR 1 and VTSR 2, and vice versa.

Day Zero Configuration for High Availability

The following example shos Day zero configuration for L3 High Availability:

```
VTSR1 #

vrf mgmt-vrf
```

```
address-family ipv4 unicast
!
address-family ipv6 unicast
!
!
tpa
vrf default
  address-family ipv4
   update-source dataports GigabitEthernet0/0/0/0
  !
!
vrf mgmt-vrf
  address-family ipv4
   update-source dataports GigabitEthernet0/0/0/1
  !
  address-family ipv6
   update-source dataports GigabitEthernet0/0/0/1
  !
!
!
interface Loopback0
ipv4 address 128.0.0.10 255.255.255.255
!
interface Loopback1
ipv4 address 169.254.10.1 255.255.255.255
!
interface MgmtEth0/RP0/CPU0/0
shutdown
!
interface GigabitEthernet0/0/0/0
ipv4 address 10.10.10.33 255.255.255.0
!
interface GigabitEthernet0/0/0/1
vrf mgmt-vrf
ip address 60.60.60.15/24
!
interface GigabitEthernet0/0/0/2
shutdown
!
router static
address-family ipv4 unicast
   0.0.0.0/0  60.60.60.1
  !
!
!

VTSR2 #

vrf mgmt-vrf
address-family ipv4 unicast
!
address-family ipv6 unicast
!
!
tpa
vrf default
  address-family ipv4
   update-source dataports GigabitEthernet0/0/0/0
  !
!
vrf mgmt-vrf
  address-family ipv4
   update-source dataports GigabitEthernet0/0/0/1
  !
```

```
  address-family ipv6
   update-source dataports GigabitEthernet0/0/0/1
  !
!
!
call-home
service active
contact smart-licensing
profile CiscoTAC-1
  active
  destination transport-method http
!
!
interface Loopback0
ipv4 address 128.0.0.10 255.255.255.255
!
interface Loopback1
ipv4 address 169.254.10.1 255.255.255.255
!
interface MgmtEth0/RP0/CPU0/0
shutdown
!
interface GigabitEthernet0/0/0/0
ipv4 address 44.44.44.15 255.255.255.0
!
interface GigabitEthernet0/0/0/1
vrf mgmt-vrf
ipv4 address 70.70.70.15/24
!
interface GigabitEthernet0/0/0/2
shutdown
!
router static
address-family ipv4 unicast
  0.0.0.0/0 44.44.44.1
!
vrf mgmt-vrf
  address-family ipv4 unicast
   0.0.0.0/0 70.70.70.1
  !
!
!
```

Verifying VTSR High Availability Setup

Check the HA status with command sudo crm status . For example:

root@vtsr01:/opt/cisco/package# crm status

Last updated: Thu Aug 30 16:27:25 2018 Last change: Thu Aug 30 14:52:01 2018 by root via cibadmin on vtsr01

Stack: corosync

Current DC: vtsr01 (version 1.1.14-70404b0) - partition with quorum

2 nodes and 24 resources configured

Online: [vtsr01 vtsr02]

Full list of resources:

```
dl_server       (ocf::heartbeat:anything):      Started vtsr01
 redis_server       (ocf::heartbeat:anything):      Started vtsr01
 Clone Set: cfg_dl_clone [cfg_dl]
    Started: [ vtsr01 vtsr02 ]
 Clone Set: rc_clone [rc]
    Started: [ vtsr01 vtsr02 ]
 Clone Set: sm_clone [sm]
    Started: [ vtsr01 vtsr02 ]
 Clone Set: tunnel_clone [tunnel]
    Started: [ vtsr01 vtsr02 ]
 Clone Set: confd_clone [confd]
    Started: [ vtsr01 vtsr02 ]
 Clone Set: mping_clone [mgmt_ping]
    Started: [ vtsr01 vtsr02 ]
 Clone Set: uping_clone [underlay_ping]
    Started: [ vtsr01 vtsr02 ]
 Clone Set: monit_clone [monit]
    Started: [ vtsr01 vtsr02 ]
 Clone Set: socat_confd_clone [socat-confd]
    Started: [ vtsr01 vtsr02 ]
 Clone Set: socat_monit_clone [socat-monit]
    Started: [ vtsr01 vtsr02 ]
 Clone Set: mate_tunnel_clone [mate_tunnel]
    Started: [ vtsr01 vtsr02 ]
root@vtsr01:/opt/cisco/package#
```

root@vtsr02:/opt/cisco/package# crm status

Last updated: Thu Aug 30 16:32:06 2018 Last change: Thu Aug 30 14:52:01 2018 by root via cibadmin on vtsr01

Stack: corosync

Current DC: vtsr01 (version 1.1.14-70404b0) - partition with quorum

2 nodes and 24 resources configured

Online: [vtsr01 vtsr02]

Full list of resources:

```
dl_server       (ocf::heartbeat:anything):      Started vtsr01
 redis_server       (ocf::heartbeat:anything):      Started vtsr01
 Clone Set: cfg_dl_clone [cfg_dl]
    Started: [ vtsr01 vtsr02 ]
 Clone Set: rc_clone [rc]
    Started: [ vtsr01 vtsr02 ]
 Clone Set: sm_clone [sm]
    Started: [ vtsr01 vtsr02 ]
 Clone Set: tunnel_clone [tunnel]
    Started: [ vtsr01 vtsr02 ]
 Clone Set: confd_clone [confd]
    Started: [ vtsr01 vtsr02 ]
 Clone Set: mping_clone [mgmt_ping]
    Started: [ vtsr01 vtsr02 ]
 Clone Set: uping_clone [underlay_ping]
    Started: [ vtsr01 vtsr02 ]
 Clone Set: monit_clone [monit]
```

```
    Started: [ vtsr01 vtsr02 ]
 Clone Set: socat_confd_clone [socat-confd]
    Started: [ vtsr01 vtsr02 ]
 Clone Set: socat_monit_clone [socat-monit]
    Started: [ vtsr01 vtsr02 ]
 Clone Set: mate_tunnel_clone [mate_tunnel]
    Started: [ vtsr01 vtsr02 ]
root@vtsr02:/opt/cisco/package#
```

Enabling VTC High Availability

To enable VTC high availability, do the following steps:

Note Step 1 to 3 should be run on both VTCs. Step 4 must be run only on the node that you want to make the active VTC.

Step 1 SSH to VTC 1 and VTC 2 .

Step 2 Go to the following directory:

cd /opt/vts/etc/

Step 3 Copy the cluster.conf file to /opt/vts/etc on both VTC 1 and VTC 2.

Step 4 Go to the following directory::

cd /opt/vts/bin

Run the **sudo ./cluster_install.sh** command. For example: admin@Onion-VTC1:/opt/vts/bin#sudo ./cluster_install.sh You will be asked to provide the vtsr password. vtsr password is the password for VTSR1 and VTSR2. You will be prompted to run cluster_install.sh on VTC2 as well . A message is displyaed similar to below example:

Please run cluster_install.sh on Onion-VTC2. Will wait until finished ==> At this point on VTC2 run the cluster install script

Run the sudo ./cluster_install.sh command. For example:

admin@Onion-VTC2:/opt/vts/bin#sudo ./cluster_install.sh

You will be asked to provide the vtsr password. vtsr password is the password for VTSR1 and VTSR2.

An output similar to what is given below is displayed:

Change made to ncs.conf file. Need to restart ncs

Finding running docker container ID

263f311dbdff

Created symlink from /etc/systemd/system/multi-user.target.wants/pacemaker.service to /lib/systemd/system/pacemaker.service.

Created symlink from /etc/systemd/system/multi-user.target.wants/corosync.service to /lib/systemd/system/corosync.service.

Retrieving and storing node certificates

Retrieving master certificate

Retrieving slave certificate

Importing master certificate

Importing slave certificate

HA cluster is installed

admin@Onion-VTC2:/opt/vts/bin#

Once this is over on VTC2 ,Automatically on VTC1 the process will resume and you will see an output similar to what is given below :

```
2018-08-25 19:33:31,358 - INFO - Setup finished at 19:33:31
Configuring Pacemaker resources
Master node configuration finished
Retrieving and storing node certificates
Retrieving master certificate
Retrieving slave certificate
Importing master certificate
Importing slave certificate
HA cluster is installed.
admin@Onion-VTC1:/opt/vts/bin#
```

Verifying the VTC High Availabilty

To verify the

Step 1 Run the sudo crm status command. For example

admin@Onion-VTC1:~$ sudo crm status

[sudo] password for admin:

Last updated: Mon Aug 27 17:34:47 2018 Last change: Sat Aug 25 20:20:11 2018 by root via crm_attribute on Onion-VTC2

Stack: corosync

Current DC: Onion-VTC1 (version 1.1.14-70404b0) - partition with quorum

2 nodes and 5 resources configured

Online: [Onion-VTC1 Onion-VTC2]

Full list of resources:

Master/Slave Set: ms_vtc_ha [vtc_ha]

Masters: [Onion-VTC1]

Slaves: [Onion-VTC2]

ClusterIP (ocf::heartbeat:IPaddr2): Started Onion-VTC1

ClusterIP2 (ocf::heartbeat:IPaddr2): Started Onion-VTC1

admin@Onion-VTC1:~$

Step 2 Verify on which VTC the virtual IP is configured. For example:

root@Onion-VTC1:/home/admin# ip addr s

2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000

link/ether 52:54:00:c7:92:c2 brd ff:ff:ff:ff:ff:ff

inet 60.60.60.10/24 brd 60.60.60.255 scope global eth0

valid_lft forever preferred_lft forever

inet 192.168.10.254/32 scope global eth0

valid_lft forever preferred_lft forever

inet6 fe80::5054:ff:fec7:92c2/64 scope link

valid_lft forever preferred_lft forever

3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000

link/ether 52:54:00:0e:17:59 brd ff:ff:ff:ff:ff:ff

inet 10.10.10.10/24 brd 10.10.10.255 scope global eth1

valid_lft forever preferred_lft forever

inet6 fe80::5054:ff:fe0e:1759/64 scope link

valid_lft forever preferred_lft forever

Verifying VTSR High Availability

Before you begin

Step 1 Verify the following on VTSR 1 and VTSR 2.

```
!
route-policy BGP_HA_VTC1_MGMT_IPV4
  if destination in (192.168.10.254) and community matches-every (6501:1001) then
    delete community all
    set next-hop 60.60.60.10
  else
    drop
  endif
end-policy
!
route-policy BGP_HA_VTC2_MGMT_IPV4
  if destination in (192.168.10.254) and community matches-every (6501:1002) then
    delete community all
    set next-hop 70.70.70.10
  else
    drop
  endif
end-policy
!
route-policy REDISTRIBUTE_TO_BGP_HA_MGMT_IPV4
  if destination in (192.168.10.254) and tag is 1001 then
```

```
    set community (6501:1001)
    done
  elseif destination in (192.168.10.254) and tag is 1002 then
    set community (6501:1002)
    done
  else
    drop
  endif
end-policy
!
router static
address-family ipv4 unicast
  0.0.0.0/0 10.10.10.1
!
vrf mgmt-vrf
  address-family ipv4 unicast
   0.0.0.0/0 60.60.60.1
   192.168.10.254/32 60.60.60.10 tag 1002
  !
!
!
router ospf 100
router-id 21.21.21.21
address-family ipv4 unicast
area 0.0.0.0
  default-cost 10
  interface Loopback0
  !
  interface GigabitEthernet0/0/0/0
  !
!
!
router bgp 6501
bgp router-id 21.21.21.21
address-family ipv4 unicast
 !
 address-family vpnv4 unicast
!
 address-family ipv6 unicast
!
 address-family vpnv6 unicast
!
vrf mgmt-vrf
  rd 60.60.60.15:1
  bgp router-id 60.60.60.15
  address-family ipv4 unicast
  network 192.168.10.254/32 route-policy   REDISTRIBUTE_TO_BGP_HA_MGMT_IPV4
  !
  neighbor 11.11.11.11
   remote-as 6500
   ebgp-multihop 255
  timers 10 30
   description ***MGMT IPV4 Network Directly connected BGP Peer of Vtsr1
   update-source GigabitEthernet0/0/0/1
   address-family ipv4 unicast
    route-policy BGP_HA_VTC1_MGMT_IPV4 out
   !
  !
  neighbor 12.12.12.12
   remote-as 6500
   ebgp-multihop 255
   timers 10 30
   description ***MGMT IPV4 Network Directly connected BGP Peer of Vtsr2
   update-source GigabitEthernet0/0/0/1
```

```
   address-family ipv4 unicast
    route-policy BGP_HA_VTC2_MGMT_IPV4 out
   !
  !
!
!
 RP/0/RP0/CPU0:vtsr01#show bgp sessions
Wed Aug 29 10:06:15.706 UTC
```

Neighbor VRF Spk AS InQ OutQ NBRState NSRState

11.11.11.11 mgmt-vrf 0 6500 0 0 Established None

12.12.12.12 mgmt-vrf 0 6500 0 0 Established None

RP/0/RP0/CPU0:vtsr01#

RP/0/RP0/CPU0:vtsr01#show bgp all all

Thu Aug 30 15:56:22.350 UTC

Address Family: VPNv4 Unicast

Address Family: VPNv6 Unicast

BGP router identifier 21.21.21.21, local AS number 6501

BGP generic scan interval 60 secs

Non-stop routing is enabled

BGP table state: Active

Table ID: 0x0 RD version: 0

BGP main routing table version 4

BGP NSR Initial initsync version 3 (Not Reached)

BGP NSR/ISSU Sync-Group versions 0/0

BGP scan interval 60 secs

Status codes: s suppressed, d damped, h history, * valid, > best

i - internal, r RIB-failure, S stale, N Nexthop-discard

Origin codes: i - IGP, e - EGP, ? - incomplete

Network Next Hop Metric LocPrf Weight Path

Route Distinguisher: 41.41.41.41:1 (default for vrf mgmt-vrf)

*> 192.168.10.254/32

60.60.60.10 0 32768 i

Processed 1 prefixes, 1 paths

Address Family: IPv4 Unicast

Address Family: IPv6 Unicast

```
---------------------------
```

Step 2 Verify whether the configuration is pushed on VTSR 2.

```
route-policy BGP_HA_VTC1_MGMT_IPV4
  if destination in (192.168.10.254) and community matches-every (6501:1001) then
    delete community all
    set next-hop 60.60.60.10
  else
    drop
  endif
end-policy
!
route-policy BGP_HA_VTC2_MGMT_IPV4
  if destination in (192.168.10.254) and community matches-every (6501:1002) then
    delete community all
    set next-hop 70.70.70.10
  else
    drop
  endif
end-policy
!
route-policy REDISTRIBUTE_TO_BGP_HA_MGMT_IPV4
  if destination in (192.168.10.254) and tag is 1001 then
    set community (6501:1001)
    done
  elseif destination in (192.168.10.254) and tag is 1002 then
    set community (6501:1002)
    done
  else
    drop
  endif
end-policy
!
router static
address-family ipv4 unicast
  0.0.0.0/0 44.44.44.1
!
vrf mgmt-vrf
  address-family ipv4 unicast
   0.0.0.0/0 70.70.70.1
   192.168.10.254/32 60.60.60.10 tag 1002
  !
!
!
router ospf 100
router-id 31.31.31.31
address-family ipv4 unicast
area 0.0.0.0
  default-cost 10
  interface Loopback0
  !
  interface GigabitEthernet0/0/0/0
  !
!
!
router bgp 6501
bgp router-id 31.31.31.31
address-family ipv4 unicast
!
address-family vpnv4 unicast
!
address-family ipv6 unicast
!
```

```
address-family vpnv6 unicast
!
vrf mgmt-vrf
  rd 70.70.70.15:1
  bgp router-id 70.70.70.15
  address-family ipv4 unicast
  network 192.168.10.254/32 route-policy REDISTRIBUTE_TO_BGP_HA_MGMT_IPV4
  !
  neighbor 11.11.11.11
   remote-as 6500
   ebgp-multihop 255
   timers 10 30
   description ***MGMT IPV4 Network Directly connected BGP Peer of Vtsr1
   update-source GigabitEthernet0/0/0/1
   address-family ipv4 unicast
    route-policy BGP_HA_VTC1_MGMT_IPV4 out
   !
  !
  neighbor 12.12.12.12
   remote-as 6500
   ebgp-multihop 255
   timers 10 30
   description ***MGMT IPV4 Network Directly connected BGP Peer of Vtsr2
   update-source GigabitEthernet0/0/0/1
   address-family ipv4 unicast
    route-policy BGP_HA_VTC2_MGMT_IPV4 out
   !
  !
!
!
```

RP/0/RP0/CPU0:vtsr02#show bgp sessions

Wed Aug 29 10:13:34.909 UTC

Neighbor VRF Spk AS InQ OutQ NBRState NSRState

11.11.11.11 mgmt-vrf 0 6500 0 0 Established None

12.12.12.12 mgmt-vrf 0 6500 0 0 Established None

RP/0/RP0/CPU0:vtsr02#

Switching Over Between Master and Slave Nodes

There are two of ways to switch over from Master to Slave node.

- Restart the nso service on the Master. The switchover happens automatically. For example:

```
admin@vts02:/opt/vts/log/nso$ sudo service nso restart


admin@vts02:/opt/vts/log/nso$ sudo crm status

[sudo] password for admin:

Last updated: Mon Apr 10 18:43:52 2017          Last change: Mon Apr 10 17:15:21 2017
by root via crm_attribute on vts01

Stack: corosync
```

```
Current DC: vts01 (version 1.1.14-70404b0) - partition with quorum

2 nodes and 4 resources configured


Online: [ vts01 vts02 ]


Full list of resources:


Master/Slave Set: ms_vtc_ha [vtc_ha]

     Masters: [ vts01 ]

     Slaves: [ vts02 ]

ClusterIP      (ocf::heartbeat:IPaddr2):       Started vts01

ClusterIPV6    (ocf::heartbeat:IPaddr2):       Started vts01
```

Or,

- Set the Master node to standby, and then bring it online.

  In the below example, vts02 is initially the Master, which is then switched over to the Slave role.

```
admin@vts01:~$ sudo crm node standby
[sudo] password for admin:


admin@vts01:/opt/vts/log/nso$ sudo crm status

[sudo] password for admin:

Last updated: Mon Apr 10 18:43:52 2017          Last change: Mon Apr 10 17:15:21 2017
by root via crm_attribute on vts01

Stack: corosync

Current DC: vts01 (version 1.1.14-70404b0) - partition with quorum

2 nodes and 4 resources configured


Node vts01 standby
Online: [ vts02 ]


Full list of resources:


Master/Slave Set: ms_vtc_ha [vtc_ha]
```

```
     Masters: [ vts02 ]

     Stopped: [ vts01 ]

ClusterIP      (ocf::heartbeat:IPaddr2):       Started vts02

ClusterIPV6    (ocf::heartbeat:IPaddr2):       Started vts02


admin@vts01~$ sudo crm node online


admin@vts02:/opt/vts/log/nso$ sudo crm status

[sudo] password for admin:

Last updated: Mon Apr 10 18:43:52 2017          Last change: Mon Apr 10 17:15:21 2017
by root via crm_attribute on vts01

Stack: corosync

Current DC: vts01 (version 1.1.14-70404b0) - partition with quorum

2 nodes and 4 resources configured


Online: [ vts01 vts02 ]


Full list of resources:


Master/Slave Set: ms_vtc_ha [vtc_ha]

     Masters: [ vts02 ]

     Slaves: [ vts01 ]

ClusterIP      (ocf::heartbeat:IPaddr2):       Started vts02

ClusterIPV6    (ocf::heartbeat:IPaddr2):       Started vts02
```

Uninstalling VTC High Availability

To move VTC back to its pre-High Availability state, run the following script:

Note Make sure the ncs server is active/running. Then run this script on both the active and standby nodes.

```
root@vts02:/opt/vts/bin# ./cluster_uninstall.sh
This will move HA configuration on this system back to pre-installed state. Proceed?(y/n)
y
```

Troubleshooting Password Change Issues

If a password change is performed while the VTS Active and Standby were up, and the change does not get applied to the Standby, the changed password will not get updated in the /opt/vts/etc/credentials file on the Standby. Due to this, when VTS Standby VM is brought up, it cannot connect to NCS. CRM_MON shows the state as shutdown for Standby, and it does not come online.

To troubleshoot this:

Step 1 Copy the **/opt/vts/etc/credentials** file from the VTC Active to the same location (/opt/vts/etc/credentials) on the VTC Standby node.

Step 2 Run the crm node command on VTC Standby to bring it online.

```
crm node online VTC2
```

Step 3 Run the command crm status to show both VTC1 and VTC2 online.

```
crm status
```

Installing VTSR in High Availability Mode

VTSR high availability mode needs to be enabled before you install VTF(s) in your set up. The second VTSR will not get registered to the VTC if it starts up after VTF installation .

Enabling VTSR high availability involves:

- Generating two ISOs for the Master and the Slave VMs. See Generating an ISO for VTSR, on page 18 for details.
- Deploy the two VTSR VMs using the respective ISO files generated during the process. See Deploying VTSR on OpenStack, on page 21 or Deploying VTSR on VMware, on page 45, based on your VMM type.

The system automatically detects which VM is the Master and which is the slave, based on the information you provide while generating the ISO files.

Verifying VTSR High Availability Setup

You can check the VTSR HA status using the sudo crm status. For example:

```
root@vtsr01:/opt/cisco/package# crm status
Last updated: Tue Aug 28 21:00:18 2018               Last change: Sat Aug 25 13:29:45 2018
 by root via cibadmin on vtsr01
Stack: corosync
Current DC: vtsr01 (version 1.1.14-70404b0) - partition with quorum
2 nodes and 24 resources configured

Online: [ vtsr01 vtsr02 ]

Full list of resources:
```

```
dl_server        (ocf::heartbeat:anything):        Started vtsr01
redis_server        (ocf::heartbeat:anything):        Started vtsr01
Clone Set: cfg_dl_clone [cfg_dl]
     Started: [ vtsr01 vtsr02 ]
Clone Set: rc_clone [rc]
     Started: [ vtsr01 vtsr02 ]
Clone Set: sm_clone [sm]
     Started: [ vtsr01 vtsr02 ]
Clone Set: tunnel_clone [tunnel]
     Started: [ vtsr01 vtsr02 ]
Clone Set: confd_clone [confd]
     Started: [ vtsr01 vtsr02 ]
Clone Set: mping_clone [mgmt_ping]
     Started: [ vtsr01 vtsr02 ]
Clone Set: uping_clone [underlay_ping]
     Started: [ vtsr01 vtsr02 ]
Clone Set: monit_clone [monit]
     Started: [ vtsr01 vtsr02 ]
Clone Set: socat_confd_clone [socat-confd]
     Started: [ vtsr01 vtsr02 ]
Clone Set: socat_monit_clone [socat-monit]
    Started: [ vtsr01 vtsr02 ]
Clone Set: mate_tunnel_clone [mate_tunnel]
     Started: [ vtsr01 vtsr02 ]
root@vtsr01:/opt/cisco/package#

root@vtsr02:/opt/cisco/package# crm status
Last updated: Tue Aug 28 21:04:38 2018                Last change: Sat Aug 25 13:29:45 2018
 by root via cibadmin on vtsr01
Stack: corosync
Current DC: vtsr01 (version 1.1.14-70404b0) - partition with quorum
2 nodes and 24 resources configured

Online: [ vtsr01 vtsr02 ]

Full list of resources:

dl_server        (ocf::heartbeat:anything):        Started vtsr01
redis_server        (ocf::heartbeat:anything):        Started vtsr01
Clone Set: cfg_dl_clone [cfg_dl]
     Started: [ vtsr01 vtsr02 ]
Clone Set: rc_clone [rc]
     Started: [ vtsr01 vtsr02 ]
Clone Set: sm_clone [sm]
     Started: [ vtsr01 vtsr02 ]
Clone Set: tunnel_clone [tunnel]
     Started: [ vtsr01 vtsr02 ]
Clone Set: confd_clone [confd]
     Started: [ vtsr01 vtsr02 ]
Clone Set: mping_clone [mgmt_ping]
     Started: [ vtsr01 vtsr02 ]
Clone Set: uping_clone [underlay_ping]
     Started: [ vtsr01 vtsr02 ]
Clone Set: monit_clone [monit]
     Started: [ vtsr01 vtsr02 ]
Clone Set: socat_confd_clone [socat-confd]
     Started: [ vtsr01 vtsr02 ]
Clone Set: socat_monit_clone [socat-monit]
    Started: [ vtsr01 vtsr02 ]
Clone Set: mate_tunnel_clone [mate_tunnel]
     Started: [ vtsr01 vtsr02 ]
root@vtsr02:/opt/cisco/package#
```

Hight Availability Scenarios

This section describes the various HA scenarios.

Manual Failover

To do a manual failover:

Step 1 Run **sudo crm node standby** on the current VTC Active to force a failover to the Standby node.

Step 2 Verify the other VTC to check whether it has taken over the Active role.

Step 3 On the earlier Active, run **crm node online** to bring it back to be part of the cluster again.

VTC Master Reboot

When the VTC Active reboots, much like a manual failover, the other VTC takes over as the Active. After coming up out of the reboot, the old Active VTC will automatically come up as the Standby.

Split Brain

When there is a network break and both VTCs are still up, VTC HA attempts to ascertain where the network break lies. During the network failure, the Active and Standby will lose connectivity with each other. At this point, the Active will attempt to contact the external ip (a parameter set during the initial configuration) to see if it still has outside connectivity.

If it cannot reach the external ip, VTC cannot know if the Standby node is down or if it has promoted itself to Active. As a result, it will shut itself down to avoid having two Active nodes.

The Standby, upon sensing the loss of connectivity with the Active, tries to promote itself to the Active mode. But first, it will check if it has external connectivity. If it does, it will become the Active node. However, if it also cannot reach the external ip (for instance if a common router is down), it will shut down.

At this point, the VTC that had the network break cannot tell if the other VTC is Active and receiving transactions. When the network break is resolved, it will be able to do the comparison and the VTC with the latest database will become Active.

If the other VTC also has a network break or is not available, the agent will not be able to do the comparison still, and it will wait. If the other VTC is not be available for some time, you may force the available VTC to be master:

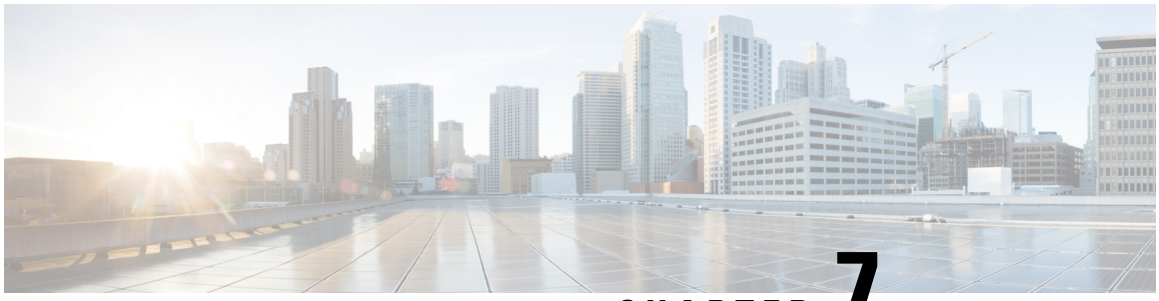
```
admin@vtc1:/home/admin# sudo /opt/vts/bin/force_master.py
```

Double Failure

When both VTC are down at the same time, a double failure scenario has occurred. After a VTC has come up, it does not immediately know the state of the other VTC's database. Consequently, before HA is resumed, an agent runs in the background trying to compare the two databases. When both systems have recovered, it will be able to do the comparison and the VTC with latest database will become the Active.

If the other VTC is not be available for some time, you may force the available VTC to be master:

```
admin@vtc1:/home/admin# sudo /opt/vts/bin/force_master.py
```



CHAPTER 7

Upgrading Cisco VTS

This chapter provides information about how to upgrade to Cisco VTS 2.6.2

Note You can directly upgrade to Cisco VTS 2.6.2 from Cisco VTS 2.6.1 or Cisco VTS 2.5.2.1 . If you are running a version other than 2.6.1 or 2.5.2.1, you have to first upgrade to either one of these 2 releases before you upgrade to version 2.6.2.

Also if you have VTSR, make sure to complete the upgrade of VTC, VTSR, reinstall VMM plugins, VTS agent and VTF before performing any CRUD operation.

This chapter has the following sections:

Upgrading VTC

Before you upgrade, ensure that:

- Cisco VTS is running version 2.5.2, 2.5.2.1, 2.6.1, or 2.6.2
- The admin has taken the backups for Day Zero and Day One configurations for all the switches managed by Cisco VTS.

  See Device documentation for the procedure about how to copy Day Zero configuration locally.

- In an HA set up, HA status is checked on both the VMs. On the Cisco VTS GUI, check HA status under **Administration > High Availability**. Or, you may use the following command:

  ```
  # sudo crm status
  ```

- In an HA setup, both VTCs are online, and one is set as Master and other is set as Slave.
- In an HA setup, *service nso status* of both VTCs is in *active (running)* state.

- In an HA setup, VTS is reachable using the VIP IP address (the IP address used to log in to the Cisco VTS GUI).
- The VTS virtual machines have enough disk-space before starting the upgrade. See Prerequisites , on page 3 chapter for details.
- All the devices in the inventory are reachable and accessible via Cisco VTS. Use the check-sync functionality to make sure all devices are in sync (**Inventory > Network Inventory** GUI).
- For devices that you want to be in *unmanaged* state, you set the devices to *unmanaged* mode:

```
 set devices device [device_name] [device_extension]:device-info device-use unmanaged
commit
```

Examples:

```
set devices device asr-dci1 asr9k-extension:device-info device-use unmanaged

set devices device n9k-leaf n9k-extension:device-info device-use unmanaged
```

When a device is specified as *unmanaged*, Cisco VTS will not sync with these devices as part of the upgrade process. Hence, if before upgrade, you use the above command to mark the devices that are not managed by Cisco VTS, then VTS will not sync with these devices and this will not cause a failure during the upgrade.

- Devices are in unlocked state (Check **Inventory > Network Inventory** GUI).
- You back up the current VTC VMs (Master and Slave) as snapshots which will need to be used to rollback if there is any problem found during the upgrade.

Note VTC and VTS are interchangeable.

"Source VTC/VTS System" can be in 2.5.2 or 2.5.2.1 or 2.6.1 or 2.6.2 Versions.

If there are service extensions or device template is configured, then follow steps mentioned in VTS Service Extension and Device Templates Migration, on page 119 section.

Step 1 Take the snapshot of the existing VTC VM. See Backing up VTC VMs as Snapshots, on page 95 for details.

Step 2 Download *vts-backup.sh* from Cisco.com, to your VTC VM (Master VTC in an HA setup).

Step 3 Go to root user using sudo su - from admin user login.

Step 4 Run the following command, on the source VTC system.

```
show_tech_support -t -a
```

This command backs up log files, including device configuration, and generates a tar file. Copy the tar file outside of the VTC host. This file will be required for troubleshooting purpose. This needs to be done on both VTC nodes in case of an HA setup.

Step 5 Run the backup script to take a backup of the database, credential files, and templates of source VTC. This copies the backup tar file to a local directory and the home directory of the remote server you specify.

```
./vts-backup.sh
Remote server IP : <remote_server>
Remote server user: <user>
Remote server password: <password>
```

Step 6 Shutdown the current VTC VM (both Master and Slave in case of HA).

Step 7 Bring up the new VTC VM with the 2.6.3 image, with the same management IP address (both Master and Slave VTC VM in case of HA).

Step 8 Copy the *vts-backup.tgz* backup file created on VTS 2.6.2 VM from a remote location to current VTC.

Step 9 Copy the upgrade ISO file from cisco.com to a local directory on VTC VM.

Step 10 Log in to VTC VM (Master VTC in case of HA) as root user using sudo su - from admin user login.

Step 11 Create a mount directory.

```
mkdir /mnt/vts-upgrade
```

Step 12 Mount the ISO which is copied to the local directory to /mnt/vts-upgrade

```
mount -o loop /tmp/VTS-docker-upgrade-2.6.3.iso /mnt/vts-upgrade
```

Step 13 Enter into the mount directory.

```
cd /mnt/vts-upgrade
```

Step 14 Run the upgrade script as *./upgrade.sh <backup tar file with path>*.

```
./upgrade.sh /tmp/vts-backup.tgz
```

- After Upgrade is done on VTC1 it will ask details for HA setup.
- The upgrade will setup VTCs in HA (no manual steps needed).

If you have out of band template configuration in Cisco VTS source system, follow the procedure detailed in section Preserving Out of Band Template Configuration, on page 96. If the upgrade fails, you need to perform a rollback to revert to source vtc version. See Performing a Rollback, on page 101 for details. You must rerun the upgrade procedure to upgrade to version 2.6.3 again.

Step 15 Run the following command, as root user. (Same as Step 3)

```
show_tech_support -t -a
```

This command backs up log files, including device configuration, and generates a tar file. Copy the tar file outside of the VTC host. This file will be required for troubleshooting purpose. This needs to be done on both VTC nodes in case of an HA setup.

Note During the upgrade process, when you do show_tech_support after you run the upgrade script, L2 High Availability gets broken. If you face this issue, follow the steps listed in the message, and reboot the Master and Slave nodes.

Step 16 Log in to VIP and do sync-to to all devices, except for VTSR.

Backing up VTC VMs as Snapshots

Saving VTC snapshots involves:

- On vCenter—Need to be done for all VTC VMs (Master and Slave):
  1. Power Off the VTC VM (recommended)
  2. Right click on the VTC VM, select **Snapshot**, and then select **Take Snapshot...**.
  3. Enter Name and Description for snapshot and click **Ok**.
  4. Power On the VTC VM.

- On OpenStack—Need to be done for all VTC VMs (Master and Slave):

1. Shutdown the VTC VMs to take snapshot using virsh save utility. VTC VMs will no longer be available in running state.

   Do virsh list, which shows the VTC domain ID, name, and status. Use Domain ID to save VTC VMs.

   ```
   root@vts-controller-110 ]# virsh list
    Id          Name             State
   ------------------------------------
   236          VTC1             running
   237          VTC2             running
   ```

   ```
   virsh save <id> <file>
   ```

   For example:

   ```
    virsh save <VTC Domain ID> <file>
   ```

   ```
   virsh save 236 vtc1.txt
                            virsh save 237 vtc2.txt
   ```

2. Take vtc.qcow2 image backup which was used to bring up Master and Slave.

   ```
    tar –cvf vtc1.qcow2.tar vtc1.qcow2
           tar –cvf vtc2.qcow2.tar vtc2.qcow2
   ```

3. Copy tar images to external drive (vtc1.qcow2.tar ,vtc2.qcow2.tar are VTC snapshots, which will be used to rollback).

4. Restore VTC VMs which will bring VTC VMs back to running state.

   ```
   virsh restore vtc1.txt
            virsh restore vtc2.txt
   ```

5. Verify if Master and Slave are up and running in HA mode. Verify GUI login using VIP IP.

Preserving Out of Band Template Configuration

If you have out of band template configuration in Cisco VTS 2.6.1 and want to upgrade to 2.6.2, do the following to ensure that the out of band template configuration is preserved after you upgrade to 2.6.2 without any interruption to the data plane.

Step 1 Upgrade to VTS 2.6.2 without doing a sync-to.

```
cd /mnt/upgrades/python
python upgrade.py upgrade -ip <vip-ip> -p <password> -b <backup dir>
```

Step 2 Run sync-to dry-run.

```
cd /mnt/upgrades/python/scripts
       ./sync_to_dry_run.script
```

Step 3 Check /opt/vts/run/upgrade/ folder with files having non-zero size.

Step 4 If there are files with non-zero size, then Southbound lock all the devices.

```
cd /mnt/upgrades/python/scripts
       ./southbound_lock_managed_devices.script
```

Step 5 Create templates that contain the out of band configuration and apply the templates. Configuration with - sign will be removed from device configuration. Configuration with + sign will be added to device configuration.

Step 6 Unlock all the devices.

```
cd /mnt/upgrades/python/scripts
      ./unlock_managed_devices.script
```

Step 7 Do a sync-to to all the devices.

```
cd /mnt/upgrades/python/scripts
      ./synch_to.script
```

Upgrading VTSR

To upgrade VTSR VM, do the following:

Step 1 Get the default site ID from the VTS GUI Home page to generate a new VTSR ISO before upgrading to new VTSR.

Step 2 Delete the existing VTSR VM and bring up the new VM using the new image. See Installing VTSR, on page 18 for details.

Step 3 If you opt to enable Monit feature, run the following command to configure Monit details via VTS CLI. This is required to update the VTC database with Monit details. For example:

Note This has to be done after VTSR gets registered with the VTC.

```
If VTC is being upgraded from 2.5.2 to 2.6.2.1 then change the monit configuration as below from VTC.

If it is VTSR HA then apply the same config on both the vtsr.admin@VTC1:/opt/vts/bin$ sudo ./vts-cli.sh
 -monitConfig vtsr-monit
admin@VTC-OSPD-131-MASTER:/opt/vts/bin$ sudo ./vts-cli.sh -monitConfig vtsr-monit
[sudo] password for admin:
command monitConfig executing with input vtsr-monit ...
Enter Site Name: <site name>
 Enter VTSR Monit user: <monit username>
 Enter VTSR Monitpassword:<monit password>
 Enter salt for VTSR Monit password encryption: <key for slat encryption>
 Enter VTSR Monit process monitoring interval(in seconds): <seconds>
 Applying Monit config in VTS DB for vtsr01...
 Changing device vtsr01 state to southbound-locked...
 Applying Monit credentials on vtsr01...
 Applying Monit process monitoring interval vtsr01...Enter VTSR Monit user: admin
 Changing device vtsr01 state to unlocked...
 Successfully applied Monit config on vtsr01 in VTS DB
```

Step 4 If VTC is being upgraded from 2.5.2 or 2.5.2.1 (where VTS and VTF password encryption is not supported), then do the following

a) Obtaining encrypted VTC/VTF/VTSR AUTH ENCRYPTED password from VTSR

1. ssh to any vtsr device
2. run bash
3. docker exec -it vtsr bash
4. confd_cli -u admin -C
5. show running-config vtsr-day0-config | include "vts-auth password"

vtsr-day0-config vts-auth password-hash ENCRYPTED_PASSWORD_SHOWN

For example, the output will look like below.

vtsr01# show running-config vtsr-day0-config | inc "vts-auth password"

vtsr-day0-config vts-auth password-hash YLdKnf3qSsKA2JWQT9a0Sg==

6. show running-config vtsr-day0-config | inc "vtf-auth password"

   vtsr-day0-config vtf-auth password-hash ENCRYPTED_PASSWORD_SHOWN

   For example, the output will look like below.

   vtsr01# show running-config vtsr-day0-config | inc "vtf-auth password"

   vtsr-day0-config vtf-auth password-hash YLdKnf3qSsKA2JWQT9a0Sg==

7. show running-config vtsr-day0-config | inc "vtsr-day0-config password"

   vtsr-day0-config password ENCRYPTED_PASSWORD_SHOWN

   For example, the output will look like below.

   vtsr01# show running-config vtsr-day0-config | include "vtsr-day0-config password"

   vtsr-day0-config password G+QB+Rq/HyF1a/TDErBMgA==

b) Updating VTS and VTF ENCRYPTED PASSWORD (got from above -- e.g. YLdKnf3qSsKA2JWQT9a0Sg== and ALdKnf3qBbKA2JWQT9a0Sg== above) in NCS CDB.

1. ncs_cli -u admin
2. config
3. set devices device <vtsr device name> state admin-state southbound-locked
4. set devices device <vtsr device name> config cisco-vtsr-day0:vtsr-day0-config vts-auth password-hash <encrypted password>
5. set devices device <vtsr device name> config cisco-vtsr-day0:vtsr-day0-config vtf-auth password-hash <encrypted password>
6. set devices device <vtsr device name> config cisco-vtsr-day0:vtsr-day0-config password <encrypted password>
7. commit
8. set devices device <vtsr device name> state admin-state unlocked
9. commit

Step 5 Do a sync-to operation for VTSR(s) in order to sync the configuration from the VTC.

Upgrading VTF

VTF has to be uninstalled and installed after the VTC upgrade.

See Inband Installation of VTF on OpenStack, on page 28 and Installing VTF on vCenter, on page 49 for details about VTF installation and uninstallation.

Upgrading Cisco VTS under OSPD

VTS component upgrade involves the same steps as in OSPD 10 Integration . The components will be automatically upgraded and the new configuration parameters applied, upon an overcloud update.

Post Upgrade Considerations

This section has certain important points you need to consider after you upgrade to Cisco VTS 2.6.2

- A default site will be created after upgrade to VTS 2.6.2.
- After upgrade, run `chown -R nso:vts-log /opt/vts/log/nso` once on the VTS Slave. This is required so that the ssh user has access to nso logs.
- Upgrade from Cisco VTS 2.6.0 to Cisco VTS 2.6.1—Impact on SRIOV ports:

  OpenStack behavior for SRIOV ports is similar to that of OVS ports in that SRIOV ports, by default, get associated with tenant's default Security Group.

  When SRIOV ports get migrated from Cisco VTS 2.6.0 to Cisco VTS 2.6.1, Cisco VTS removes any Security Groups associated with them as they do not serve any purpose anyways.

  You must edit these SRIOV ports and associate them with either 'no security groups' or with a security group that does not use 'remote-sg'.

  If above action is not performed, any subsequent SRIOV ports updates from OpenStack would get rejected as Cisco VTS does not allow SRIOV ports to get associated with Security Groups containing remote-sg.

- After upgrade from Cisco VTS 2.6.0 to Cisco VTS 2.6.1, fabric static routes (for an overlay Router) which are designated for selective devices will not be device specific anymore. They will be applied to all network devices that have the overlay networks associated to the Router. As such, after upgrade, many devices will be going out-of-sync because of this and you have to decide if you want these static routes on those devices.
- After upgrade you need to go to **Site Administration > Virtual Machine Manager** page, and edit each OpenStack VMM and edit each Neutron Server and save. This is required to update J-Driver plugin.
- After upgrade, a default site will be created automatically.
- After upgrade you need to go to **Inventory > Host Inventory** page, and edit each host with OVS virtual switch type to trigger reinstallation of Host Agent.
- When you upgrade to Cisco VTS 2.6.1, each rule specified in the existing Security Groups gets reprogrammed on the VTFs with the reflexive attribute turned on. This may render some of the rules within the Security Groups redundant. You need to remove any rules that are deemed redundant in the context of reflexive ACL feature.
- Cisco VTS will not be transitioning any of the BGP configuration that were part of L3 Service Extension templates when you upgrade to Cisco VTS 2.6.1, which supports BGP as a service. For migrating BGP to a service, you must:

  1. Create the appropriate Port Extension with BGP configuration and associate them respectively with all the necessary ports deployed. See *Creating Port Extensions* section in the *Cisco VTS User Guide*, for details.

2. Validate whether the configuration being pushed to the devices are similar to those that had been pushed via the service templates.
3. Disassociate the service templates from the devices and verify that there is no configuration or service loss.
4. Repeat the process for all BGP configuration that can be transitioned.

Note It is not applicable for 2.6.1 to 2.6.2 upgrade, but still valid for other upgrade path.

- **1.** If there is a port-scope static route with BFD enabled in 2.6.0 or 2.6.1 and upgrade to 2.6.2 then VTC automatically pushes “no ip redirects” or “no ipv6 redirects” in 2.6.2 VTS cdb. After post upgrade if “check sync” is done then device will show as Out of sync saying that VTS 2.6.2 cdb has “no ip redirects” or “no ipv6 redirects” and not in device. This is expected in 2.6.2 and customer has to do “sync to” (from VTS CDB to Device) to push the config to device only if below matching configuration shows in check sync output.

```
devices {
     device tor6-pod2 {
         config {
             nx:interface {
                 Vlan 1001 {
                     ip {
-                        redirects false;
                     }
                     ipv6 {
-                        redirects false;
                     }
                 }
             }
         }
     }
}
```

2. If below configuration pushed from VTC 2.5.2.1 or 2.6.1 to device (asr9k as DCI/DCGW) and do an upgrade to 2.6.2

```
vrf admin-rtr-1
address-family ipv4 unicast
  import from default-vrf route-policy data-center-vrf-import-policy advertise-as-vpn

  import route-target
   200:30002 stitching
   201:45000
  300:30002 stitching
  !
  export to default-vrf route-policy data-center-vrf-export-policy allow-imported-vpn

  export route-target
   200:30002 stitching
   201:45000
   300:30002 stitching
  !
```

- After post upgrade to 2.6.2, the DCI device will go out of sync and “check sync” on VTC will show the the following configuration:

```
devices {
     device dc3-pod2 {
         config {
             cisco-ios-xr:vrf {
                 vrf-list admin-rtr-1 {
                     address-family {
                         ipv4 {
                             unicast {
                                 import {
                                     from {
                                         default-vrf {
-                                            route-polic
```

Workaround for this issue is do the “sync from” (DCI to VTS) only if the above configuration matches in check sync output.

Performing a Rollback

The following sections describes the procedure to roll back to the Cisco VTS version from which you upgraded.

- Performing a Rollback on vCenter
- Performing a Rollback on OpenStack

Performing a Rollback on OpenStack

Do the following to rollback to the Cisco VTS version from which you upgraded. This should be done for all the VTC VMs (Master and Slave).

Step 1 On the controller, do *virsh list*.

```
root@vts-controller-110 ]# virsh list
 Id          Name             State
------------------------------------
236          VTC1             running
237          VTC2             running
```

Step 2 Virsh destroy already existing VTC VMs (Master and Slave).

```
virsh destroy <id>
```

Step 3 Copy *vtc1.qcow2.tar* and *vtc2.qcow2.tar* from external drive to the controller.

Step 4 Untar *vtc1.qcow2.tar* and *vtc2.qcow2.tar*

```
untar -xvf vtc1.qcow2.tar
         untar -xvf vtc2.qcow2.tar
```

Step 5 Create Master and Slave VTC (virsh create utility) using *vtc.xml* file which points to the location of *qcow* images that is untarred in the above step.

Note Create the Master VTC first, wait for two to three minutes, and then create the Slave VTC.

Step 6 Verify if Master and Slave are up and running in HA mode. Verify GUI log in using VIP IP.

Note Make sure that the *service nso status* of both VTCs is in *active* state.

In case nso status is in *inactive* state then kill and recreate that VTC. Then reverify if Master and Slave are up and running in HA mode. Verify GUI log in using VIP IP. Also, make sure that service nso status of both VTC is currently in *active* state.

Step 7 Manually reregister the VMM and Host Agent from VTS GUI.

Performing a Rollback on vCenter

Do the following to rollback to the Cisco VTS version from which you upgraded. This should be done for all the VTC VMs (Master and Slave).

Step 1 Power Off the VTC VM (recommended).

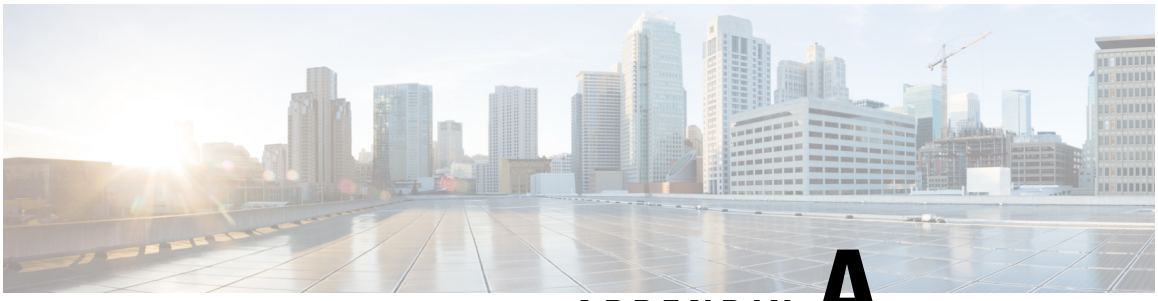
Step 2 Right click on VTC VM and select **Snapshot**, and then **Snapshot Manager...**.

Step 3 Select **Snapshot** and click **Go to**. Click **Close** to close the screen.

Step 4 Power On the VTC VM.

Step 5 Verify if HA is up and running. Verify GUI log in using VIP IP.

Step 6 Manually reregister VMM from VTS GUI.



APPENDIX A

OpenStack VTF vhost Mode Considerations

This appendix details the general considerations for deploying VTF in vhost mode on OpenStack.

Note Only RHEL nodes are currently supported as target vhost nodes.

Requirement	Details
RHEL version	7.3
QEMU	qemu-kvm-rhev-2.6.0-28.el7_3.6.x86_64
libVirt	libvirt-2.0.0-10.el7_3.4.x86_64
Kernel Drivers	uio_pci_generic: version 0.01.0 or vfio_pci: version 0.2
OpenStack	• Newton • Nova Compute
Target vhost Compute Node RAM / CPU	16Gb / 2 CPU
Hugepage	The installer takes care of this requirement.

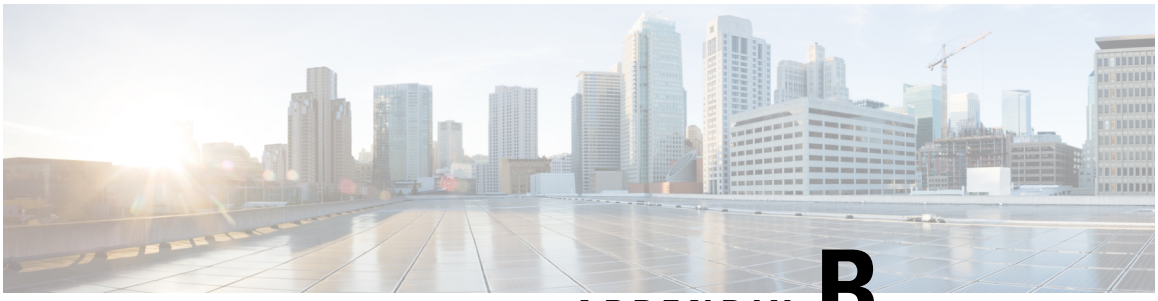
Note

- If an image is tuned to run on VTF (w mem_page_size) and the same image is used on a server that does not have VTF and huge pages created, it might fail.
- If you deploy a VM on VTF host w/o mem_page_size large, the VM might come up fine, but may not be able to ping anything.
- Using an image with mem_page_size set on a OVS host (non-VTF) fails because huge pages are not created.

- Requirements to run with Vector Packet Processing (VPP) and DPDK—See VPP and DPDK documentation for details.
- Numa node requirements

OpenStack VTF vhost Mode Considerations

- OpenStack Flavor Extra Specs details— See OpenStack Flavors documentation for details.
- NIC Support—The following are supported:
  - Normal NIC-Intel NIANTIC (x510, IXGBE 82599)
  - Cisco VIC
  - Mellanox NIC (MCX4121A-ACAT) ConnectX-4 Lx EN 25GbE dual-port SFP28, PCIe3.0 x8, tall bracket



APPENDIX B

Sample XML Files

The following sections provide sample XML files.

Sample XML File—VTC Installation

```
<domain type='kvm' id='1332'>
  <name>VTC-release2.1</name>
  <uuid>5789b2bb-df35-4154-a1d3-e38cefc856a3</uuid>
  <memory unit='KiB'>16389120</memory>
  <currentMemory unit='KiB'>16388608</currentMemory>
  <vcpu placement='static'>8</vcpu>
  <resource>
    <partition>/machine</partition>
  </resource>
  <os>
    <type arch='x86_64' machine='pc-i440fx-rhel7.0.0'>hvm</type>
    <boot dev='hd'/>
  </os>
  <features>
    <acpi/>
    <apic/>
    <pae/>
  </features>
  <cpu mode='custom' match='exact'>
    <model fallback='allow'>Westmere</model>
    <feature policy='require' name='vmx'/>
  </cpu>
  <clock offset='utc'/>
  <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='disk'>
      <driver name='qemu' type='qcow2' cache='none'/>
      <source file='/home/cisco/VTS2.1/vtc.qcow2'/>
      <target dev='vda' bus='virtio'/>
      <alias name='virtio-disk0'/>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x06' 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='pci' index='0' model='pci-root'>
  <alias name='pci.0'/>
</controller>
<controller type='virtio-serial' index='0'>
  <alias name='virtio-serial0'/>
  <address type='pci' domain='0x0000' bus='0x00' slot='0x05' function='0x0'/>
</controller>
<interface type='bridge'>
  <mac address='52:54:00:5b:12:3a'/>
  <source bridge='br-ex'/>
  <virtualport type='openvswitch'>
    <parameters interfaceid='263c1aa6-8f7d-46f0-b0a3-bdbdad40fe41'/>
  </virtualport>
  <target dev='vnet0'/>
  <model type='virtio'/>
  <alias name='net0'/>
  <address type='pci' domain='0x0000' bus='0x00' slot='0x03' function='0x0'/>
</interface>
<interface type='bridge'>
  <mac address='52:54:00:8d:75:75'/>
  <source bridge='br-control'/>
  <virtualport type='openvswitch'>
    <parameters interfaceid='d0b0020d-7898-419e-93c8-15dd7a08eebd'/>
  </virtualport>
  <target dev='vnet1'/>
  <model type='virtio'/>
  <alias name='net1'/>
  <address type='pci' domain='0x0000' bus='0x00' slot='0x0b' function='0x0'/>
</interface>
<serial type='tcp'>
  <source mode='bind' host='127.0.0.1' service='4888'/>
  <protocol type='telnet'/>
  <target port='0'/>
  <alias name='serial0'/>
</serial>
<console type='tcp'>
  <source mode='bind' host='127.0.0.1' service='4888'/>
  <protocol type='telnet'/>
  <target type='serial' port='0'/>
  <alias name='serial0'/>
</console>
<channel type='spicevmc'>
  <target type='virtio' name='com.redhat.spice.0'/>
  <alias name='channel0'/>
  <address type='virtio-serial' controller='0' bus='0' port='1'/>
</channel>
<input type='mouse' bus='ps2'/>
<graphics type='spice' port='5900' autoport='yes' listen='127.0.0.1'>
  <listen type='address' address='127.0.0.1'/>
</graphics>
<sound model='ich6'>
  <alias name='sound0'/>
  <address type='pci' domain='0x0000' bus='0x00' slot='0x04' function='0x0'/>
</sound>
<video>
  <model type='qxl' ram='65536' vram='65536' heads='1'/>
  <alias name='video0'/>
  <address type='pci' domain='0x0000' bus='0x00' slot='0x02' function='0x0'/>
</video>
<memballoon model='virtio'>
  <alias name='balloon0'/>
  <address type='pci' domain='0x0000' bus='0x00' slot='0x07' function='0x0'/>
```

```
    </memballoon>
  </devices>
  <seclabel type='dynamic' model='selinux' relabel='yes'>
    <label>system_u:system_r:svirt_t:s0:c26,c784</label>
    <imagelabel>system_u:object_r:svirt_image_t:s0:c26,c784</imagelabel>
  </seclabel>
</domain>
```

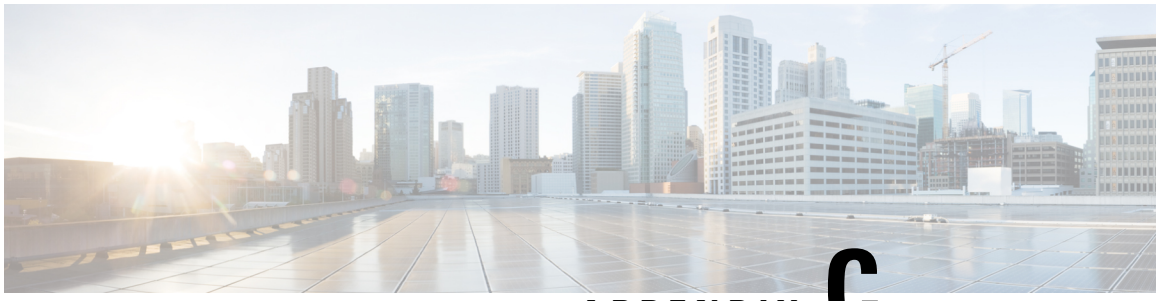
Sample XML File—VTSR Installation

```
<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>
```

Sample XML Files



APPENDIX C

Running VTC and VTSR within OpenStack as Tenant Virtual Machines

In certain deployment scenarios, it may be necessary to run VTC and/or VTSR as tenant VMs on OpenStack. This is a deviation from the recommended method of running VTC and VTSR directly on KVM. This appendix provides details on the considerations and steps required in such scenarios.

This appendix has the following section:

Running VTC and VTSR within OpenStack as Tenant VMs

Note If VTC and/or VTSR are running as tenant VMs, the management and underlay networks which they are attached to must be independent of the tenant networks which they are designed to manage later on.

To run VTC/VTSR as a tenant VM, the following consideration needs to be made:

- The nova flavor should match VTC/VTSR's requirements.
- The VTC/VTSR VM should use persistent instead of ephemeral storage. This is achieved by using a cinder volume as the persistent drive.
- There must be a way to auto-configure VTC parameters using a config drive. This is achieved by using a 2nd cinder volume, mounted as CDROM.

Note After VTC is launched, its default password needs to be changed from the Web UI before VTSR registers correctly.

Prerequisites:

- VTC and VTSR software image have been downloaded from cisco.com to OpenStack controller node.
- Config ISO images for VTC and VTSR have been created.

- Cinder volume should have at least 130G of space available. For example: VTC requires 48G and VTSR requires 80G based on 2.5.0.
- Neutron networks for attaching VTC (2x NICs) and VTSR's (6x NICs) have been created.
- Openstack only allows traffic from the IP address of the VM that OpenStack assigns during the installation.

  VIP address is not something that OpenStack assigns. So the normal behavior for OpenStack is to drop the traffic for VIP IP, due to security reasons.

  If you need to access VIP, you may use the allowed-address-pair option.

  While creating a port allowed-address-pairs can be passed, as an additional parameter, to specify the additional IP that should be allowed. This is the neutron port create API.

For VTC

The following section details the steps specific to VTC.

Step 1 Glance VTC image into OpenStack. For example:

```
glance image-create --file vtc.qcow2 --progress --visibility public --disk-format qcow2 --name vtc250
 --container-format bare
[=============================>] 100%
+------------------+--------------------------------------+
| Property         | Value                                |
+------------------+--------------------------------------+
| checksum         | e195df17122ec8bdaa771b3d148546e4     |
| container_format | bare                                 |
| created_at       | 2017-08-03T13:42:39Z                 |
| disk_format      | qcow2                                |
| id               | 52a10029-91ef-44f6-9f78-159cead8da9c |
| min_disk         | 0                                    |
| min_ram          | 0                                    |
| name             | vtc250                               |
| owner            | ea71291e36e94fa1b5745779b1d456cc     |
| protected        | False                                |
| size             | 10529538048                          |
| status           | active                               |
| tags             | []                                   |
| updated_at       | 2017-08-03T13:44:10Z                 |
| virtual_size     | None                                 |
| visibility       | public                               |
+------------------+--------------------------------------+
```

Step 2 Create a (persistent) cinder volume for booting up VTC, based on VTC image. For example:

```
openstack volume create --image vtc250 --size 48 vtc_vol
+---------------------+--------------------------------------+
| Field               | Value                                |
+---------------------+--------------------------------------+
| attachments         | []                                   |
| availability_zone   | nova                                 |
| bootable            | false                                |
| consistencygroup_id | None                                 |
| created_at          | 2017-08-03T13:45:05.573850           |
| description         | None                                 |
| encrypted           | False                                |
| id                  | e4fb13fb-a23a-45ce-a2b4-0a3cfe4916af |
| migration_status    | None                                 |
```

```
| multiattach         | False                                |
| name                | vtc_vol                              |
| properties          |                                      |
| replication_status  | disabled                             |
| size                | 48                                   |
| snapshot_id         | None                                 |
| source_volid        | None                                 |
| status              | creating                             |
| type                | None                                 |
| user_id             | 3b5684ca7fd2418084090b48904a9237     |
+---------------------+--------------------------------------+
```

Step 3 Create VTC config image based on VTC config drive (vtc_config_250.iso). For example:

```
openstack image create vtc_config --file vtc_config_250.iso --disk-format iso --container-format bare
+------------------+------------------------------------------------------+
| Field            | Value                                                |
+------------------+------------------------------------------------------+
| checksum         | c020985f6de566b3b8b6bad02e440f93                     |
| container_format | bare                                                 |
| created_at       | 2017-08-03T13:46:40Z                                 |
| disk_format      | iso                                                  |
| file             | /v2/images/0d74a180-9af4-4dfb-bc81-1f31b11f5a4e/file |
| id               | 0d74a180-9af4-4dfb-bc81-1f31b11f5a4e                 |
| min_disk         | 0                                                    |
| min_ram          | 0                                                    |
| name             | vtc_config                                           |
| owner            | ea71291e36e94fa1b5745779b1d456cc                     |
| protected        | False                                                |
| schema           | /v2/schemas/image                                    |
| size             | 358400                                               |
| status           | active                                               |
| tags             |                                                      |
| updated_at       | 2017-08-03T13:46:41Z                                 |
| virtual_size     | None                                                 |
| visibility       | private                                              |
+------------------+------------------------------------------------------+
```

Step 4 Set VTC config image properties. For example:

```
openstack image set --property hw_cdrom_bus=ide --property hw-disk_bus=ide vtc_config
```

Step 5 Create VTC config cinder volume, based on VTC config image. For example:

```
openstack volume create vtc_config_vol --image vtc_config --size 1
+---------------------+--------------------------------------+
| Field               | Value                                |
+---------------------+--------------------------------------+
| attachments         | []                                   |
| availability_zone   | nova                                 |
| bootable            | false                                |
| consistencygroup_id | None                                 |
| created_at          | 2017-08-03T13:48:37.932104           |
| description         | None                                 |
| encrypted           | False                                |
| id                  | 32c93acf-0e35-4a67-89b9-44ae190ac76a |
| migration_status    | None                                 |
| multiattach         | False                                |
| name                | vtc_config_vol                       |
| properties          |                                      |
| replication_status  | disabled                             |
| size                | 1                                    |
| snapshot_id         | None                                 |
| source_volid        | None                                 |
| status              | creating                             |
```

```
| type                | None                                 |
| user_id             | 3b5684ca7fd2418084090b48904a9237     |
+---------------------+--------------------------------------+
```

Step 6 Boot VTC volume with attached config drive (volume). For Example:

```
nova boot --flavor m1.large \
--nic net-id=f12b2a45-aa80-42b3-8007-57730a1325fd \
--nic net-id=ec6e25c2-48e5-4f1a-9f09-774cc4ae0750 \
--block-device
id=e4fb13fb-a23a-45ce-a2b4-0a3cfe4916af,source=volume,dest=volume,device=/dev/vda,bootindex=0  \
--block-device
id=32c93acf-0e35-4a67-89b9-44ae190ac76a,source=volume,dest=volume,bus=ide,device=/dev/vdb,type=cdrom
 \
vtc
+------------------------------+-------------------------------------------------------------------------------------+
| Property                     | Value
                               |
+------------------------------+-------------------------------------------------------------------------------------+
| OS-DCF:diskConfig            | MANUAL
                               |
| OS-EXT-AZ:availability_zone  |
                               |
| OS-EXT-SRV-ATTR:host         | -
                               |
| OS-EXT-SRV-ATTR:hypervisor_hostname  | -
                               |
| OS-EXT-SRV-ATTR:instance_name        | instance-00000096
                               |
| OS-EXT-STS:power_state       | 0
                               |
| OS-EXT-STS:task_state        | scheduling
                               |
| OS-EXT-STS:vm_state          | building
                               |
| OS-SRV-USG:launched_at       | -
                               |
| OS-SRV-USG:terminated_at     | -
                               |
| accessIPv4                   |
                               |
| accessIPv6                   |
                               |
| adminPass                    | KCvjE9aZQ7Td
                               |
| config_drive                 |
                               |
| created                      | 2017-08-03T13:52:01Z
                               |
| flavor                       | m1.large (4)
                               |
| hostId                       |
                               |
| id                           | dd38f88b-95a8-40c7-8670-538be76e91ce
                               |
| image                        | Attempt to boot from volume - no image supplied
                               |
| key_name                     | -
                               |
| metadata                     | {}
                               |
| name                         | vtc
                               |
| os-extended-volumes:volumes_attached | [{"id": "e4fb13fb-a23a-45ce-a2b4-0a3cfe4916af"}, {"id":
```

```
"32c93acf-0e35-4a67-89b9-44ae190ac76a"}] |
| progress                             | 0                                                                                |
|                                      |
| security_groups                      | default                                                                          |
|                                      |
| status                               | BUILD                                                                            |
|                                      |
| tenant_id                            | ea71291e36e94fa1b5745779b1d456cc                                                 |
|                                      |
| updated                              | 2017-08-03T13:52:02Z                                                             |
|                                      |
| user_id                              | 3b5684ca7fd2418084090b48904a9237                                                 |
|                                      |
+--------------------------------------+----------------------------------------------------------------------------------+
```

For VTSR

The following section details the steps specific to VTSR:

Step 1 Glance VTSR Image into OpenStack. For Example:

```
glance image-create --file vtsr.qcow2 --progress --visibility public --disk-format qcow2 --name
vtsr250 --container-format bare
[==============================>] 100%
+------------------+--------------------------------------+
| Property         | Value                                |
+------------------+--------------------------------------+
| checksum         | 0e44a2f2d5266670e1f0664928d6f726     |
| container_format | bare                                 |
| created_at       | 2017-08-03T13:58:47Z                 |
| disk_format      | qcow2                                |
| id               | c6a80651-686f-485c-9336-1176f8338387 |
| min_disk         | 0                                    |
| min_ram          | 0                                    |
| name             | vtsr250                              |
| owner            | ea71291e36e94fa1b5745779b1d456cc     |
| protected        | False                                |
| size             | 2921594880                           |
| status           | active                               |
| tags             | []                                   |
| updated_at       | 2017-08-03T13:59:13Z                 |
| virtual_size     | None                                 |
| visibility       | public                               |
+------------------+--------------------------------------+
```

Step 2 Create Cinder Volume based on VTSR Image. For Example:

```
openstack volume create --image vtsr250 --size 80 vtsr_vol
+---------------------+--------------------------------------+
| Field               | Value                                |
+---------------------+--------------------------------------+
| attachments         | []                                   |
| availability_zone   | nova                                 |
| bootable            | false                                |
| consistencygroup_id | None                                 |
| created_at          | 2017-08-03T14:00:14.317952           |
| description         | None                                 |
| encrypted           | False                                |
| id                  | 53b919b7-56a2-4a05-93bd-5f81ba762dc1 |
```

```
| migration_status    | None                                 |
| multiattach         | False                                |
| name                | vtsr_vol                             |
| properties          |                                      |
| replication_status  | disabled                             |
| size                | 80                                   |
| snapshot_id         | None                                 |
| source_volid        | None                                 |
| status              | creating                             |
| type                | None                                 |
| user_id             | 3b5684ca7fd2418084090b48904a9237     |
+---------------------+--------------------------------------+
```

Step 3 Create VTSR Config Image based on VTSR Config ISO (vtsr_node1_cfg.iso). For Example:

```
openstack image create vtsr_config --file vtsr_node1_cfg.iso --disk-format iso --container-format
bare
+------------------+------------------------------------------------------+
| Field            | Value                                                |
+------------------+------------------------------------------------------+
| checksum         | 960a23f61e73cdcf24295e3182f4f663                     |
| container_format | bare                                                 |
| created_at       | 2017-08-03T14:01:26Z                                 |
| disk_format      | iso                                                  |
| file             | /v2/images/7e5cbbb8-e092-4ebc-9249-8a13ab0a7335/file |
| id               | 7e5cbbb8-e092-4ebc-9249-8a13ab0a7335                 |
| min_disk         | 0                                                    |
| min_ram          | 0                                                    |
| name             | vtsr_config                                          |
| owner            | ea71291e36e94fa1b5745779b1d456cc                     |
| protected        | False                                                |
| schema           | /v2/schemas/image                                    |
| size             | 360448                                               |
| status           | active                                               |
| tags             |                                                      |
| updated_at       | 2017-08-03T14:01:26Z                                 |
| virtual_size     | None                                                 |
| visibility       | private                                              |
+------------------+------------------------------------------------------+
```

Step 4 Set VTSR Config Image properties. For Example:

```
openstack image set --property hw_cdrom_bus=ide --property hw-disk_bus=ide vtsr_config
```

Step 5 Create VTSR Config Image cinder volume, based on VTSR Config Image. For Example:

```
openstack volume create vtsr_config_vol --image vtsr_config --size 1
+---------------------+--------------------------------------+
| Field               | Value                                |
+---------------------+--------------------------------------+
| attachments         | []                                   |
| availability_zone   | nova                                 |
| bootable            | false                                |
| consistencygroup_id | None                                 |
| created_at          | 2017-08-03T14:02:56.332067           |
| description         | None                                 |
| encrypted           | False                                |
| id                  | 3813f48c-10ce-4d03-9587-09d3cb6b1af1 |
| migration_status    | None                                 |
| multiattach         | False                                |
| name                | vtsr_config_vol                      |
| properties          |                                      |
| replication_status  | disabled                             |
| size                | 1                                    |
| snapshot_id         | None                                 |
```

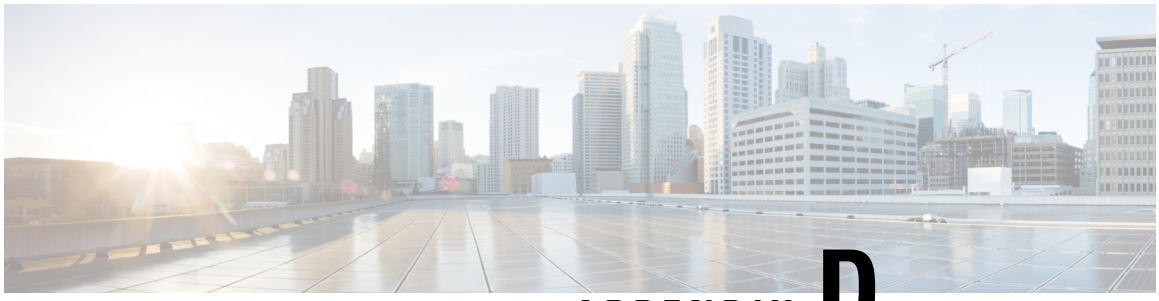
```
| source_volid        | None                                 |
| status              | creating                             |
| type                | None                                 |
| user_id             | 3b5684ca7fd2418084090b48904a9237     |
+---------------------+--------------------------------------+
```

Step 6 Boot VTSR volume with attached config drive (volume). For Example:

```
nova boot --flavor m1.xlarge  \
--nic net-id=29ddb641-aa7a-4473-a0bd-b6d6bd029240 \
--nic net-id=6c13f4a0-2871-41da-a20a-9063c2535269 \
--nic net-id=51b2c511-0341-4921-abb6-9b9f9f5d345a \
--nic net-id=ec6e25c2-48e5-4f1a-9f09-774cc4ae0750 \
--nic net-id=f12b2a45-aa80-42b3-8007-57730a1325fd \
--nic net-id=b1d841d4-257b-4dd7-bda8-fed5f3c8bef4 \
--block-device
id=53b919b7-56a2-4a05-93bd-5f81ba762dc1,source=volume,dest=volume,device=/dev/vda,bootindex=0  \
--block-device
id=3813f48c-10ce-4d03-9587-09d3cb6b1af1,source=volume,dest=volume,bus=ide,device=/dev/vdb,type=cdrom
 \
vtsr
+------------------------------+-------------------------------------------------------------------------------------+
| Property                     | Value
                               |
+------------------------------+-------------------------------------------------------------------------------------+
| OS-DCF:diskConfig            | MANUAL
                               |
| OS-EXT-AZ:availability_zone  |
                               |
| OS-EXT-SRV-ATTR:host         | -
                               |
| OS-EXT-SRV-ATTR:hypervisor_hostname  | -
                               |
| OS-EXT-SRV-ATTR:instance_name        | instance-00000097
                               |
| OS-EXT-STS:power_state       | 0
                               |
| OS-EXT-STS:task_state        | scheduling
                               |
| OS-EXT-STS:vm_state          | building
                               |
| OS-SRV-USG:launched_at       | -
                               |
| OS-SRV-USG:terminated_at     | -
                               |
| accessIPv4                   |
                               |
| accessIPv6                   |
                               |
| adminPass                    | A52TRbkcQyrn
                               |
| config_drive                 |
                               |
| created                      | 2017-08-03T14:06:42Z
                               |
| flavor                       | m1.xlarge (5)
                               |
| hostId                       |
                               |
| id                           | a3bd937a-78ab-47c4-91ca-d0f106b31f2a
                               |
| image                        | Attempt to boot from volume - no image supplied
                               |
| key_name                     | -
```

Running VTC and VTSR within OpenStack as Tenant Virtual Machines

```
                                     |
| metadata                           | {}
                                     |
| name                               | vtsr
                                     |
| os-extended-volumes:volumes_attached | [{"id": "53b919b7-56a2-4a05-93bd-5f81ba762dc1"}, {"id":
"3813f48c-10ce-4d03-9587-09d3cb6b1af1"}] |
| progress                           | 0
                                     |
| security_groups                    | default
                                     |
| status                             | BUILD
                                     |
| tenant_id                          | ea71291e36e94fa1b5745779b1d456cc
                                     |
| updated                            | 2017-08-03T14:06:42Z
                                     |
| user_id                            | 3b5684ca7fd2418084090b48904a9237
                                     |
+------------------------------------+------------------------------------------------------------------------------------------+
```



APPENDIX D

VTS Service Extension and Device Templates Migration

This appendix provides details on the considerations and pre-upgrade procedural steps required to identify the impacted service extension templates and device templates because of NED upgrade. Before your begin to pre-upgrade, make sure that the impacted templates to be fixed and migrated to the new NED version.

Pre-upgrade Considerations

The following steps provide pre-upgrade procedure to identify the impacted service extension templates and device templates migration.

Step 1 **To identify the impacted templates:**

Download the template-migration.tar file from the CCO based on the Current and Target version of the VTS.

a) After downloading the .tar file, transfer it to the VTC machine that has the customer CDB installed.

b) Extract the .tar file through "*tar -xvf <filename>*", it will create the directory "*vts-launcher*" "*cd vts-launcher*" and "*ls -lt*" to get the list of files.

c) Identify the file "*find-impacted-templates-<version1>-to-<version2>*" according to the migration path you are doing and login as an "*admin*" user to execute it.

Note If you have impacted templates you will have a directory created called "*templates*" that will contain a file per impacted template named "*<template name>.impacted.keys*". In each file there is a list of key paths that indicates an impacted key in that template.

d) Enter "cli" mode through "ncs_cli -u admin -C" and for each template that was impacted (For example, has a file in the "*templates*" directory) execute the following CLI to export the template as json:

```
show running-config templates template <templatename> | display json | save ./templates/<template
 name>.json
```

Make sure that the templates directory has the permission set to 777 before running this command.

e) Transfer the "*templates*" directory to your laptop or remote server and fix the templates by following next steps either manually (2a) or by using a dummy target VM (2b).

Step 2 **Fix the templates:**

VTS Service Extension and Device Templates Migration

2a. Fix the templates manually:

You can fix templates manually. With the available list of impacted template files you can fix each one of the templates to be compatible with the target version. For example, after executing the script "**./find-impacted-templates-252-262.sh**" you can find a file named "sample1.impacted.keys" in *my templates* directory along with an already exported template called/named "sample1.json".

Note The "sample1.json" will contain old content of the template and exported using the "show" command.

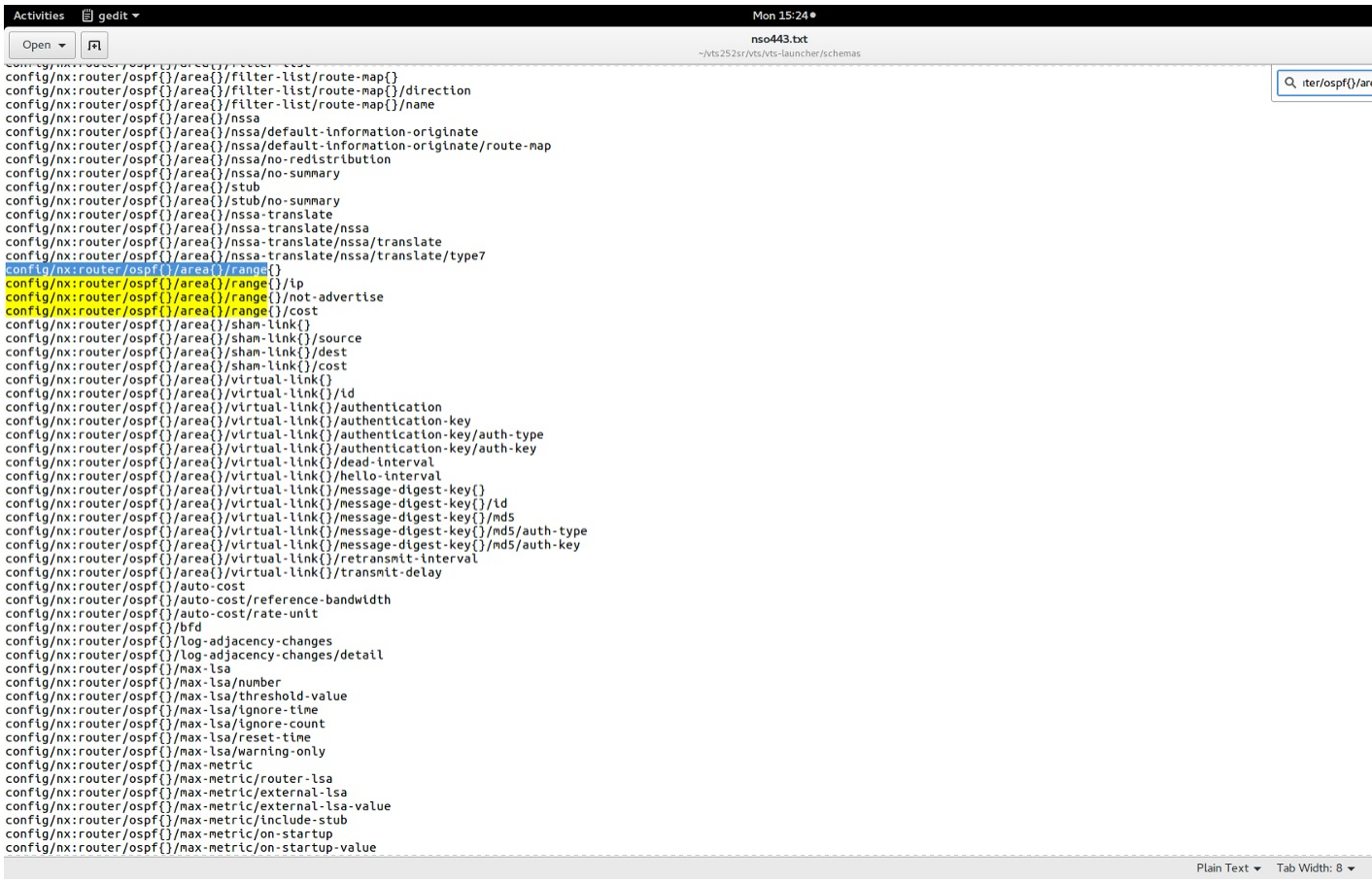
When you open the "sample1.impacted.Keys" file, you can view the following entry:

```
config/nx:router/ospf{}/area{}/range{}/mask
```

The above entry indicates that in the upgrade-to version the "**mask**" attribute was either altered or deleted. To figure out the exact changes, open the traget *<schema name>.txt* under the *vts-launchers/schemas* directory and seek the path up until the change. In this example, we seek the string "*config/nx:router/ospf{}/area{}/range*".

Note For more examples of 2.6.2 upgrade, refer to the schema file - *nso4612.txt file*.

Figure 1: Schema

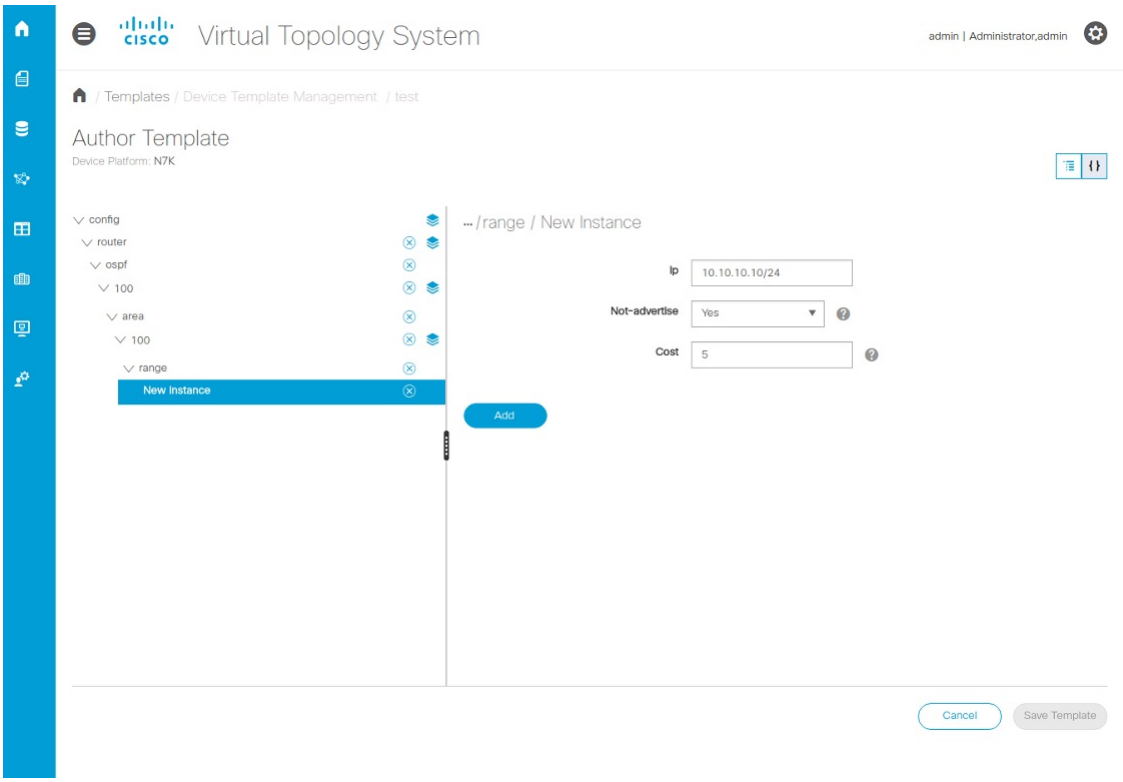


Since there is no "*/mask*" it is clear that the attribute was deleted. To identify the attribute that was added,

- Use a target VTC machine with the target version.
- Create that path as part of a new template.

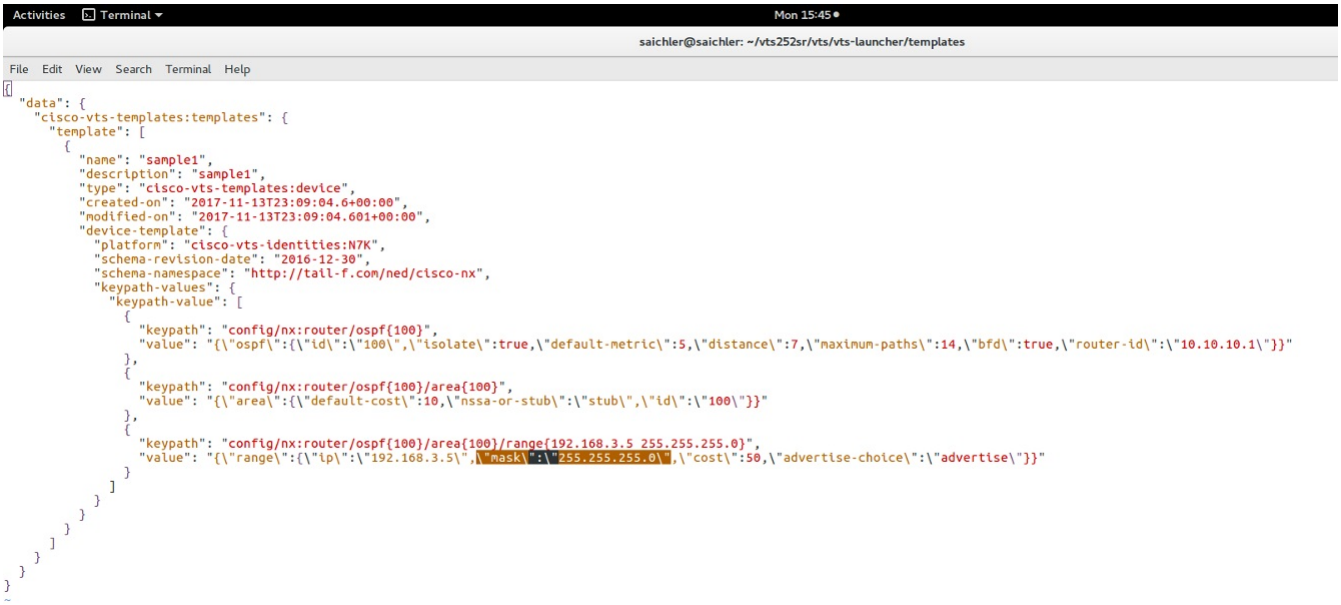
You can view the targeted VTC version in the VTS GUI as shown in below figure:

Figure 2: GUI of the targeted VTC version:



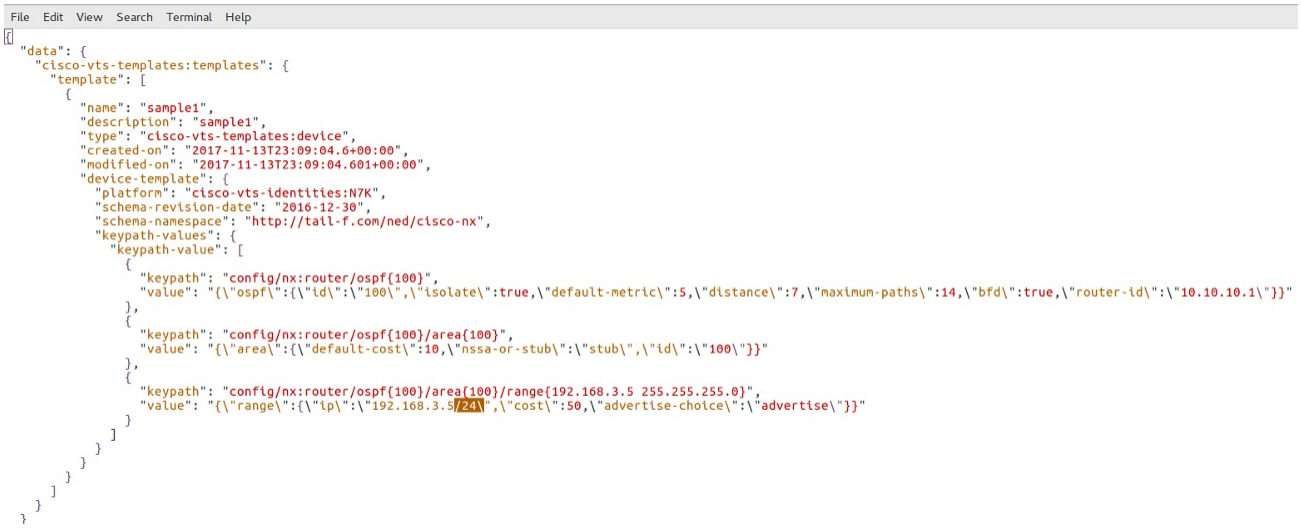
As shown in the below image, remove the "**mask**" attribute and add it as "**./xx**" to the ip address and do that fix in the json file:

Figure 3: Before the fix json file:



VTS Service Extension and Device Templates Migration

Figure 4: After the fix json file:



After the fix as shown in the above image, the sample1.jason file is ready to be used during the migration process.

Follow the above steps and migrate all the impacted templates in the *templates* directory according to the new model.

To automate the 2a section, you can refer to the sample conversion script which will convert the impacted templates to the format compatible to VTS262.

Note This conversation script is not supported and you need to identify the impacted Keypaths and modify and fix/update the script accordingly.

Steps to be performed:

- Step 1: Once you identify the templates which are impacted using template migration script, you can copy this 'Sample_Conversion2521_262.py' script to the *vts-launcher* directory.
- Step 2: Download both the packages pexpect and ptyprocess and install them as listed below.
  - Copy the below two packages to /home/admin in VTS252 VM.

    - ptyprocess-0.5.2.tar.gz

    - pexpect-4.4.0.tar.gz

- Step 3: Login to VTS252 VM as admin, switch to root by running sudo su -
  - Extract the packages:

    #tar -zxvf ptyprocess-0.5.2.tar.gz

    #tar -zxvf pexpect-4.4.0.tar.gz

  - Install ptyprocess first:

    Navigate to /home/admin/ptyprocess-0.5.2 directory, do 'chmod 777 *' and run

    #python3 setup.py install

  - Install pexpect:

    Navigate to */home/admin/pexpect-4.4.0* directory, do 'chmod 777 *' and run

#python3 setup.py install

Once done, go back to the */vts-launcher* directory and run the conversion script. At the Username/Password prompt, enter UI login credentials.

```
VTS-252-To-262/vts-launcher# python3 Sample_Conversion2521_262.py

Enter VTS Username: admin
Enter VTS Password:
```

Here impacted templates will get converted to the vts262 compatible format and are stored in the '*templates-for-migration/*' directory, which is located in *cd /vts-launcher/../templates-for-migration* directory.

- Step 4: Copy *templates-for-migration* directory to */home/admin* and follow the upgrade process steps by providing proper permissions and proceeding with the upgrade process.

2b. To fix the templates by creating a dummy target VM:

If there is a target VTC VM available, recreate the impacted templates with the same name and export as json files by running the following command from the NCS CLI:

```
- show running-config templates template <template name> | display json | save ./tmp/<template
name>.json
```

For VTS2.6.2 as it supports multi sites, when you create a template in VTS 2.6.2 you can view this template under the specific site.

If you are exporting the templates from VTS-2.6.2, use the command:

```
% show running-config vts-service sites site 111C3493-DFC6-4D54-8E0C-02470CA25111 templates template
 N7K-RS | display json | save /tmp/N7K-RS.json
```

where default-site-id is *111C3493-DFC6-4D54-8E0C-02470CA25111*

These json files can be used for template migration in the next step by copying to "/*home/admin/templates-for-migration*" in the VTC VM instead of fixing the existing templates.

Step 3 **Copy the migrated templates to the VTC 2.5.2 machine**

a) Create a directory "*/home/admin/templates-for-migration*" in the VTC VM and copy the migrated json files to the directory. During VTS upgrade, templates in the json files will get updated in the CDB. Make sure that "*templates-for-migration*" directory with the contents have the correct ownership and permission before starting the upgrade.

Note Template migration has to be completed and the migrated templates needs to be copied back to the VTC VM prior to taking the backup.

b) Run the following commands to set the right permission:

```
chown -R admin:vts-admin /home/admin/templates-for-migration
chmod -R 777 /home/admin/templates-for-migration
```

Only migrated json files which need be imported back to the CDB should be present in the "*templates-for-migration*" directory.

VTS Service Extension and Device Templates Migration