



Cisco Virtual Topology System (VTS) 2.1.5 Installation Guide

First Published:

Last Modified:

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](#) 1

CHAPTER 2

[Prerequisites](#) 3

- [System Requirements for VTC VM](#) 3
- [System Requirements for IOS XRv VM](#) 4
- [System Requirements for VTF VM](#) 4
- [Supported Virtual Machine Managers](#) 5
- [Supported Platforms](#) 5

CHAPTER 3

[Installing Cisco VTS](#) 9

- [Installing Cisco VTS in a Linux - OpenStack Environment](#) 9
 - [Installing the VTC VM](#) 9
 - [Installing VTC VM - Automatic Configuration Using ISO File](#) 10
 - [Installing VTC VM - Manual Configuration Using virt-manager Application](#) 11
 - [Installing VTC VM - Manual Configuration using VNC](#) 12
 - [Installing OpenStack Plugin](#) 13
 - [Installing OpenStack Host Agent](#) 14
- [Installing Cisco VTS on a VMware Environment](#) 15
 - [Installing VTC VM on ESXi](#) 15
 - [Installing vCenter VTC Plugin](#) 16
 - [Important Notes Regarding VMware vSphere Distributed Switch \(VDS\)](#) 17
 - [For Non-VPC Specific Configuration](#) 17
 - [For VPC Specific Configuration](#) 17
- [Installing the Virtual Topology Forwarder](#) 18
 - [Creating an IOS XRv VM](#) 18
 - [Setting up Nested VM in RedHat](#) 18
 - [Setting up Nested VM in VMware](#) 19
 - [Bringing up the KVM-based IOS XRv VM](#) 19

Deploying the vCenter-based IOS XRv VM	19
Running the Setup Script	20
Creating an ISO for IOS XRv	20
Installing VTF on VMWare	22
Installing VTF on OpenStack	23
Verifying VTS Installation	24
Verifying VTC VM Installation	24
Verifying IOS XRv VM Installation	24
Verifying VTF VM Installation	25

CHAPTER 4**Installing VTS in High Availability Mode 27**

Instantiating VTC VMs and IOS XRv VMs	27
Setting up the VTC Environment	28
Enabling VTC High Availability	30
Enabling IOS XRv High Availability	32
Modifying OpenStack Plugin and Host Agent Configuration to use VIP	32
Verifying the VTC High Availability	33
Uninstalling VTC High Availability	34

CHAPTER 5**Post-Installation Tasks 35**

APPENDIX A**Sample XML Files 37**

Sample XML File—VTC Installation	37
Sample XML File—IOS XRv Installation	39



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) 2.1.5

- 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 15](#).

Cisco VTS can be deployed with a Virtual Topology Forwarder (VTF), a software data plane designed for packet processing on x86 server. For information about installing the VTF, see [Installing the Virtual Topology Forwarder, on page 18](#).

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 27](#)

You can also install Cisco VTS without a Virtual Machine Manager (VMM). See the *Cisco VTS 2.1.5 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, page 3](#)
- [System Requirements for IOS XRv VM, page 4](#)
- [System Requirements for VTF VM, page 4](#)
- [Supported Virtual Machine Managers, page 5](#)
- [Supported Platforms, page 5](#)

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
CPU	8
Memory	16 GB
Computing Host	Certified with Cisco UCS B-series, Cisco UCS C220 and C240 Rack Servers
Hypervisor	<ul style="list-style-type: none">• VMware ESXi 5.5 or VMware ESXi 6.0• Red Hat Enterprise Linux 7.1 with KVM

System Requirements for IOS XRv VM

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


Note

The IOS XRv VM 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 IOS XRv only if you consider enabling High Availability or if you plan to have a VTF in your set up.

Requirement	Details
Disk Space	Primary disk must be 2 GB; secondary disk of arbitrary size can be added
CPUs	6
Memory	32 GB RAM
Computing Host	Certified with Cisco UCS B-series, Cisco UCS C220 and C240 Rack Servers
Hypervisor	<ul style="list-style-type: none"> • VMware ESXi 5.5 or VMware ESXi 6.0 • Red Hat Enterprise Linux 7.1 with KVM

System Requirements for VTF VM

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

Requirement	Details
Disk Space	8 GB
CPU Cores	2
Memory	16 GB RAM

Hypervisor	<ul style="list-style-type: none"> VMware ESXi 5.5 or VMware ESXi 6.0 Red Hat Enterprise Linux 7.1 with KVM
Server network interface card (NIC)	Intel DPDK-supported NIC

Supported Virtual Machine Managers

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

- OpenStack:
 - OpenStack Icehouse
 - OpenStack Juno
 - OpenStack Liberty on CentOS
- VMware vCenter:
 - VMware vCenter 5.5 Update 2
 - VMware vCenter 6.0 Update 1 Server Appliance

Supported Platforms

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



Note

VTS supports VXLAN overlays using the BGP EVPN control plane.

Role	Platform Supported
Top-of-rack (ToR) leaf switch	<ul style="list-style-type: none"> Cisco Nexus 9300TX and 9300PX platform switches Cisco Nexus 9332PQ and 93128TX switches Cisco Nexus 5600 platform switches Cisco Nexus 9500 platform switches

Data center spine	<ul style="list-style-type: none"> • Cisco Nexus 9300TX and 9300PX platform switches • Cisco Nexus 9500 platform switches • Cisco Nexus 7x00 Series switches • Cisco Nexus 5600 platform switches
Border leaf and border spine	<ul style="list-style-type: none"> • Cisco Nexus 9300TX and 9300PX platform switches • Cisco Nexus 9500 platform switches • Cisco Nexus 5600 platform switches
Data center interconnect (DCI)	<ul style="list-style-type: none"> • Cisco ASR 9000 Series Aggregation Services Routers • Cisco Nexus 7x00 Series Switches
Fabric Extenders (FEX)	<ul style="list-style-type: none"> • Cisco Nexus C2248TP-E9500 • Cisco Nexus C2232PP <p>FEX support is available for Cisco Nexus 9300, Cisco Nexus 5600, Cisco Nexus 9500 and Cisco Nexus 7x00 switches.</p>
Virtual machine manager (VMM)	<ul style="list-style-type: none"> • OpenStack Icehouse, Juno, and Liberty release • VMware vCenter 5.5 Update 2 • VMware vCenter 6.0 Update 1 Server Appliance
Hypervisor	<ul style="list-style-type: none"> • VMware ESXi 5.5; VMware ESXi 6.0 • Red Hat Enterprise Linux 7.1 with KVM
Virtual forwarders	Cisco Virtual Topology Forwarder (VTF)

**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)I1(2) and later.
Cisco Nexus 95xx	NX OS Release 7.0(3)I1(2) and later.
Cisco Nexus 7x00	<ul style="list-style-type: none"> • Data center spine —NX OS Release 7.3 and later. • Data center interconnect (DCI): <ul style="list-style-type: none"> ◦ VRF Peering mode—NX OS Release 7.3 and later. ◦ Integrated DCI mode—TBD.
Cisco Nexus 5600	NX OS Release 7.3 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.

Table 2: VPC Modes Supported

Cisco Nexus 93xx	Server VPC
Cisco Nexus 95xx	Server VPC
Cisco Nexus 5600	Server VPC, FEX VPC, Enhanced VPC



Installing Cisco VTS

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

- [Installing Cisco VTS in a Linux - OpenStack Environment, page 9](#)
- [Installing Cisco VTS on a VMware Environment, page 15](#)
- [Installing the Virtual Topology Forwarder, page 18](#)
- [Verifying VTS Installation, page 24](#)

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 VTF on OpenStack, on page 23](#)

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 11](#)

To install VTC VM using VNC (Manual Configuration), see [Installing VTC VM - Manual Configuration using VNC, on page 12](#)

Installing VTC VM - Automatic Configuration Using ISO File

For a VTC VM to configure itself on OpenStack, the administrator needs to create a text file with the VM settings, wrap it into an ISO file, and then attach the ISO to the VM's CD drive.

-
- Step 1** Connect to the controller node via SSH, and copy the `vtc.qcow2` file to `/var/lib/libvirt/images/` folder.
- Step 2** Copy the `vtc.sample.xml` file to your controller. A sample XML file is available at [Sample XML Files](#), on page 37.
- Step 3** Create a file called `config.txt`. The contents of the file is given in the below example:

```

Hostname=vtc
ManagementIPv4Method=Static
ManagementIPv4Address=1.1.1.2
ManagementIPv4Netmask=255.255.255.0
ManagementIPv4Gateway=1.1.1.1
UnderlayIPv4Method=Static
UnderlayIPv4Address=2.2.2.2
UnderlayIPv4Netmask=255.255.255.0
DNSv4=3.3.3.3
Domain=cisco.com
NTPv4=1.1.1.1
vts-adminPassword=cisco123
AdministrativeUser=admin
AdministrativePassword=cisco123

```

Note The `config.txt` file must have a blank line at the end.

In this file:

- **Hostname**—The hostname of the VM
- **ManagementPv4Method**—Whether to use DHCP or static 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).
- **UnderlayIPv4Method**—Whether to use DHCP or static 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).
- **DNSv4**—DNS IPv4 address (required only for static addressing).
- **Domain**—DNS search domain (required only for static addressing).
- **NTPv4**—NTP IPv4 address or FQDN (required only for static addressing).
- **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
- 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 Files](#), on page 37.
- 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
- 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 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 2.1.5 User Guide* for details.

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

---

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

**Step 2** Click the Add (+) button.  
The Add Virtual Machine Manager popup is displayed.

**Step 3** Enter the following details:

- VMM Type—Specify the VMM type. Choose openstack from the drop-down list.
- Version Name—Specify the version details.

- Description—Enter a description for the VMM.
- IP Address-Port—Enter the IP address.
- User Name—Enter the VMM username.
- Password—Enter the VMM password.

**Step 4** Click Add  
The VMM you added is listed in the Virtual Machine Manager screen.  
You can check the status of VMM registration in the Status column.  
To delete a VMM, select the VMM and click **X** (delete).

---

## Installing OpenStack Host Agent

You can use the Install Capabilities button in the Host Inventory page to install the OpenStack Host Agent.

---

- Step 1** Go to **Inventory > Host Inventory**. The Inventory / Host Inventory window appears
- Step 2** Click + to add a host. You may also edit a host and modify the parameters to enable installation of physical or virtual capabilities.
- If you click + (Add) the Add Host popup is displayed. Enter the following details.
  - If you choose to edit an existing host, the following windows are displayed depending upon the host type:
    - Host Details:
      - Host Name
      - Host Type
      - Host Interface
      - Host IP Address
      - Host Interface
      - Device Port Name
      - Capability—Specify whether it is a virtual-switch or not.
      - Username
      - Password
    - Common Parameters—These are displayed only if the host capability is virtual-switch.
    - VTF IP—The IP address of the VTF.
    - Subnet Mask

- Gateway
- Tenant Bridge—Name of the tenant network port group/bridge on the binding-host to which VTF is attached.
- Underlay Bridge—Name of the underlay network portgroup/bridge on the binding-host to which VTF is attached.
- Datastore
- Username
- Password

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

- Step 3** Click **Install Capabilities**. Based on the host type, it installs the host agent on the host. See the Status column for the installation status. The VMM type is also shown once the capabilities get installed.
- 

## Installing Cisco VTS on a VMware Environment

Installing Cisco VTS on a VMware environment involves:

- [Installing VTC VM on ESXi, on page 15](#)
- [Installing vCenter VTC Plugin, on page 16](#)
- [Initializing 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 source location, and click Next.  
**Note** Ensure that you have placed the vtc.ovf and vtc.vmdk file in the same directory.
- Step 4** Map VTC network connectivity to appropriate port-groups on vSwitch/DVS.
- vNIC1—Used for VTC network management
  - vNIC2—Used for VTC connectivity to VTF, IOS XRv
- Step 5** Enter the following properties:
- Hostname—VTS Hostname.

- ManagementIPv4Method—DHCP / Static IP configuration for static IP .
- ManagementIPv4Address—Management IP address for VTC. This IP address is used for VTC network management.
- ManagementIPv4Netmask—Management IP Netmask
- ManagementIPv4Gateway—Management Gateway address
- UnderlayIPv4Method—DHCP / Static IP configuration for static IP.
- UnderlayIPv4Address—Underlay IP address. This is the IP address for internal network.
- UnderlayIPv4Netmask—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

## Installing vCenter VTC Plugin

The vCenter plugin gets installed when you register the vCenter 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 2.1.5 User Guide* for details.

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



**Note** You need to restart the vCenter webserver process before you log in to the vCenter web UI.

- Step 1** Go to **Administration > Virtual Machine Manager**.
- Step 2** Click the Add (+) button.  
The Add Virtual Machine Manager popup is displayed.
- Step 3** Enter the following details:
- VMM Type—Specify the VMM type. Choose vcenter from the drop-down list.

- Version Name—Specify the version details.
- Description—Enter a description for the VMM.
- IP Address-Port—Enter the IP address and the port. The default port is 443.
- User Name—Enter the VMM username.
- Password—Enter the VMM password.

**Step 4**

Click Add

The VMM you added is listed in the Virtual Machine Manager screen.

You can check the status of VMM registration in the Status column.

To delete a VMM, select the VMM and click X (delete).

## Important Notes Regarding VMware vSphere Distributed Switch (VDS)

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 VDSs.
- 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 the Virtual Topology Forwarder

You can install VTF using the Cisco VTS GUI. See [Installing OpenStack Host Agent and VTF using GUI](#) for details.

Before you install VTF, you must install the IOS XRv VM and register it to VTS. IOS XRv VM is the control plane VM.

Installing and registering IOS XRv involves:

- [Creating an IOS XRv VM](#) , on page 18
- [Creating an ISO for IOS XRv](#), on page 20

## Creating an IOS XRv VM

The IOS XRv VM is an essential part of the Virtual VTEP topology. The IOS XRv VM contains a nested VM so IOS XRv must enable nesting capabilities.

### Setting up Nested VM in RedHat

This has been verified with RedHat 7.1 OSP.

---

**Step 1** Run `cat /sys/module/kvm_intel/parameters/nested`.

**Step 2** If the output is N, enable nested KVM feature after shutting down all active VMs.

```
echo "options kvm-intel nested=1" | sudo tee /etc/modprobe.d/kvm-intel.conf
 rmmmod kvm_intel
 modprobe kvm_intel
```

**Step 3** Run `cat /sys/module/kvm_intel/parameters/nested` and verify that it gives Y.

---

## Setting up Nested VM in VMware

This has been verified for ESXI 5.5 based hosts.

- 
- Step 1** Bring up the IOS XRv VM.
- Step 2** Edit the VMFS (VM file system in host ESXi OS where the VM is hosted) to enable VHV.
- Step 3** On the ESXi host, find the VMX file for the VM.
- ```
vim-cmd vmsvc/getallvms
```
- Step 4** After you locate the VMX file for the IOS XRv VM, enable the VHV flag.
- ```
vi /vmfs/volumes/5575f8e2-194cc001-df6a-885a92889f0a/XRVR2/XRNC.vmx
 vhw.enable= "TRUE"
```
- Step 5** Power on the VM.
- 

## Bringing up the KVM-based IOS XRv VM

- 
- Step 1** Create IOS XRv VM XML referring the sample XML (XRNC.XML).
- Step 2** Generate an ISO file for the IOS XRv. See [Creating an ISO for IOS XRv, on page 20](#).
- Step 3** Create the VM using the XML.
- ```
virsh create XRNC.xml
```
-

Deploying the vCenter-based IOS XRv VM

-
- Step 1** Generate an ISO file for the IOS XRv VM. See [Creating an ISO for IOS XRv, on page 20](#).
- Step 2** In the vSphere Client, select **File > Deploy OVF Template**. The Deploy OVF Template wizard appears.
- Step 3** Select XRNC.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 XRNC.iso file which was generated and uploaded to the host.
- Step 10** Power on the VM.

Running the Setup Script

You must run the setup script on the IOS XRv to complete the configuration.

SSH into your IOS XRv.

- If you do not want to run in High Availability mode, run the setup script as in the below example:

```
cisco@XRVR-DL1:~$ sudo /opt/cisco/package/sr/bin/setupXRNC_HA.sh 0.0.0.0
```

- If you do want to run in HA mode, run the setup script by providing the IP address of the other IOSXRv DL. For example:

```
cisco@XRVR-DL1:~$ sudo /opt/cisco/package/sr/bin/setupXRNC_HA.sh 11.1.1.17
```

Creating an ISO for IOS XRv

To create an ISO file for IOS XRv:

- Step 1** Create the system.cfg file based on the below sample.

Note Ensure that there are no spaces or extra characters in the configuration file.

```
# This is a sample day0 configuration file
```

```
# Copyright (c) 2015 cisco Systems
```

```
# VTS Information
```

```
VTS_ADDRESS="172.29.128.12"
```

```
VTS_REGISTRATION_USERNAME="admin"
```

```
VTS_REGISTRATION_PASSWORD="Cisco123!"
```

```
# VTC/VTF Network Configuration
```

```

HOSTNAME="DL-XRVR"
NTP_SERVER="172.29.128.1"
NETWORK_CONFIG_METHOD="static"
#MGMT_NETWORK_CONFIG_METHOD="dhcp"
NETWORK_NAMESERVER_IP="172.29.128.1"
UNDERLAY_NETWORK_CONFIG_METHOD="static"
UNDERLAY_NETWORK_IP_ADDRESS="10.29.128.82"
UNDERLAY_NETWORK_IP_NETMASK="255.255.255.0"
#NETWORK_IP_NETMASK=24
#UNDERLAY_NETWORK_IP_GATEWAY="10.168.94.1"

MGMT_NETWORK_CONFIG_METHOD="static"
MGMT_NETWORK_IP_ADDRESS="172.29.128.44"
MGMT_NETWORK_IP_NETMASK="255.255.255.0"
MGMT_NETWORK_IP_GATEWAY="172.29.128.1"

# VTC/VTF Admin user/password hash
USERNAME='cisco'
# Generate with openssl passwd -1 -salt <salt> <password>
# cisco/cisco123
PASSWORD_HASH='$1$xxx$J3aa90XAPYg6HSNUUD2o1'

# XRVR Specific Settings (VTC only)
XRVR_USERNAME="admin"
XRVR_PASSWORD="cisco123"
XRVR_STATIC_MGMT_IP="172.29.128.45/24"
XRVR_STATIC_UNDERLAY_IP="10.29.128.83/24"
XRVR_NAME="XRVR1"

```

Note The IOS XRv login/password is hardcoded to admin/cisco123.

Step 2 Copy your system.cfg files for IOS XRv at the same path where the script resides. For example:

```

admin:/opt/cisco/package/vts/bin$ ls -l
total 1432
-rwxr-xr-x 1 vts-admin vts-admin 4767 Sep 29 16:40 build_vts_config_iso.sh
-rw-r--r-- 1 root root 1242 Sep 29 23:54 system.cfg

```

Step 3 Create the ISO file as shown below (you need to log in as root).

```

root:/opt/cisco/package/vts/bin# ./build_vts_config_iso.sh xrnc system.cfg
Validating input.
Generating ISO File.
Done!

```

Step 4 Spawn the IOS XRv VM with ISO connected to it.

Step 5 Power on the VM.

Installing VTF on VMWare

You can use the Install Capabilities button in the Host Inventory page to install VTF.

-
- Step 1** Go to **Inventory > Host Inventory**. The Inventory / Host Inventory window appears
- Step 2** Click + to add a host. You may also edit a host and modify the parameters to enable installation of physical or virtual capabilities.
- If you click + (Add) the Add Host popup is displayed. Enter the following details.
 - If you choose to edit an existing host, the following windows are displayed depending upon the host type:
 - Host Details:
 - Host Name
 - Host Type
 - Host Interface
 - Host IP Address
 - Host Interface
 - Device Port Name
 - Capability—Specify whether it is a virtual-switch or not.
 - Username
 - Password
 - Common Parameters—These are displayed only if the host capability is virtual-switch.
 - VTF IP—The IP address of the VTF.
 - Subnet Mask
 - Gateway
 - Tenant Bridge—Name of the tenant network port group/bridge on the binding-host to which VTF is attached.
 - Underlay Bridge—Name of the underlay network portgroup/bridge on the binding-host to which VTF is attached.
 - Datastore
 - Username
 - Password
- Ensure that you review the tooltips for important information about the entries.
- Step 3** Click **Install Capabilities**. It installs VTF on the host. See the Status column for the installation status. The VMM type is also shown once the capabilities get installed.

Installing VTF on OpenStack

You can use the Install Capabilities button in the Host Inventory page to install VTF.

Step 1

Go to **Inventory > Host Inventory**. The Inventory / Host Inventory window appears

Step 2

Click + to add a host. You may also edit a host and modify the parameters to enable installation of physical or virtual capabilities.

- If you click + (Add) the Add Host popup is displayed. Enter the following details.
- If you choose to edit an existing host, the following windows are displayed depending upon the host type:
 - Host Details:
 - Host Name
 - Host Type
 - Host Interface
 - Host IP Address
 - Host Interface
 - Device Port Name
 - Capability—Specify whether it is a virtual-switch or not.
 - Username
 - Password
 - Common Parameters—These are displayed only if the host capability is virtual-switch.
 - VTF IP—The IP address of the VTF.
 - Subnet Mask
 - Gateway
 - Tenant Bridge—Name of the tenant network port group/bridge on the binding-host to which VTF is attached.
 - Underlay Bridge—Name of the underlay network portgroup/bridge on the binding-host to which VTF is attached.
 - Datastore
 - Username
 - Password

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

- Step 3** Click **Install Capabilities**. It installs the VTF on the host. See the Status column for the installation status. The VMM type is also shown once the capabilities get installed.
-

Verifying VTS Installation

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

- [Verifying VTC VM Installation, on page 24](#)
- [Verifying IOS XRv VM Installation, on page 24](#)
- [Verifying VTF VM Installation, on page 25](#)

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.
Incase the ping fails, verify VM networking to the underlay network.
- Note** Underlay network gateway is the switched virtual interface (SVI) created for IOSXRv and VTF on the leaf where the controller is connected.
- Step 4** After a few minutes, verify whether the VTS UI is reachable at the below URL:
`https://<vts-management-ip>:8443/VTS`
-

Verifying IOS XRv VM Installation

To verify ISO XRv VM installation:

-
- Step 1** Log in to the IOS XRv VM 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, use virt-manager or VNC based console method to login into the VM. See [Installing VTC VM - Manual Configuration using VNC, on page 12](#)

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

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

Step 4 Verify whether the nested IOS XRv is booting up. To do this, run:

```
sudo telnet 0 5087
```

If the o/p command fails, verify whether nested virtualization on the host where IOSXRv is booted is turned on.

Also, verify whether another telnet session is not using up this session.

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

Step 6 On the XRv shell, run the setup command.

```
sudo /opt/cisco/package/sr/bin/setupXRNC_HA.sh 0.0.0.0
```

0.0.0.0 should be replaced by the IP address of the second IOSXRv in case of HA installation.

Verifying VTF VM Installation

To verify VTF VM installation:

Step 1 Log in to the VTF VM 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, use virt-manager or VNC based console method to login into the VM. See [Installing VTC VM - Manual Configuration using VNC, on page 12](#)

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

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

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

```
sudo telnet 0 5002
```

If the o/p command fails, run the following command:

```
sudo service vpfa restart
```

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



Installing VTS in High Availability Mode

See the following sections for detailed information about installing VTS in high availability mode.

- [Instantiating VTC VMs and IOS XRv VMs](#) , page 27
- [Setting up the VTC Environment](#), page 28
- [Enabling VTC High Availability](#), page 30
- [Enabling IOS XRv High Availability](#), page 32
- [Modifying OpenStack Plugin and Host Agent Configuration to use VIP](#), page 32
- [Verifying the VTC High Availability](#), page 33
- [Uninstalling VTC High Availability](#), page 34

Instantiating VTC VMs and IOS XRv VMs

Spawn two VTC VMs and two IOS XRv VMs. See [Creating an ISO for IOS XRv](#), on page 20 for bringing up the IOS XRv. In the IOS XRv system.cfg file, for the field 'VTS_ADDRESS', use the address you have planned for the VTS VIP.

Each VTC should have an IOS XRv that is on the same or different broadcast domain. The two VTCs can be on different L3 subnet.

Note Each IOS XRv should have 32 GB of memory and at least 6 vcpu.

At a minimum, you would need to have four IP addresses for VTC - One for VTC1, one for VTC2, one for the public Virtual IP (VIP), and one for the private VIP, which the other devices on the private network such as IOS XRv and VTF can reach.

Setting up the VTC Environment

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

Step 1

Edit `/opt/cisco/package/vtc/bin/cluster.conf` file on both the VTCs.
Both the VTCs must have the identical information in the `cluster.conf` file.

```

###Virtual IP of VTC Master on the public interface
vip_public=103.1.1.2

###Virtual IP of VTC Master on the private interface
vip_private=11.1.1.20
private_network_interface=eth1

###VTC1 Information
master_name=vtc1
master_ip=101.1.1.4
master_network_interface=eth0

###VTC2 Information
slave_name=vtc2
slave_ip=102.1.1.4
slave_network_interface=eth0

###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 in your environment
external_ip=172.20.100.40

###If you have your VTCs in different subnets, xrvr will need to be configured to route traffic
###and the below section needs to be filled in.
###If you have the VTCs on the same subnet, the below section can be skipped.

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

###Ip of your first XRVr. Example: 11.1.1.5
xrvr1_mgmt_ip=

###List of neighbors for XRv1, separated by comma. Example: 11.1.1.1,11.1.1.2
xrvr1_bgp_neighbors=

###Ip of your first XRVr. Example: 12.1.1.5
xrvr2_mgmt_ip=

###List of neighbors for XRv2, separated by comma. Example: 12.1.1.1,12.1.1.2
xrvr2_bgp_neighbors=

```

```

####Credentials for Xrvr
xrvr_user=
xrvr_pass=

####Xrvr ASN information
remote_ASN=
local_ASN=

####Xrvr BGP information
bgp_keepalive=
bgp_hold=

```

- Note**
- The Xrvr section should only be filled in if you plan on using VTC1 and VTC2 on different subnets. Otherwise leave this section blank.
 - master_name and slave_name has to match the VTC Hostname.
 - master_network_interface and slave_network_interface are interface names of VTC1 and VTC2 where the real IP resides.
 - private_network_interface is the secondary interface names of VTC1 and VTC2 on the private network that IOS XRv is also on.
 - vip_private is the VIP for the VTS master's private interface.

Step 2

After modifying the cluster.conf file with the appropriate values for each field, execute the /opt/cisco/package/vtc/bin/modify_host_vtc.sh script. This will do the following:

- Modify the hostname for VTC1 and VTC2 under /etc/hostname.

```

cisco@vtc1:~$ more /etc/hostname
vtc1

```

```

cisco@vtc2:~$ more /etc/hostname
vtc2

```

- Change the hostname using the **hostname** command.
- Modify and add additional entries for VTC1 and VTC2 in /etc/hosts.

Note /etc/hosts entries are not identical for VTC1 and VTC2. Entry for 127.0.1.1 are different on both VTC.

On VTC1:

```

127.0.0.1      localhost
127.0.1.1      vtc1
101.1.1.4      vtc1
102.1.1.4      vtc2

```

On VTC 2:

```

127.0.0.1      localhost
127.0.1.1      vtc2
101.1.1.4      vtc1
102.1.1.4      vtc2

```

Enabling VTC High Availability

You must run the cluster installer script `/opt/cisco/package/vtc/bin/cluster_install.sh` on both VTCs to enable high availability.



Note Before you run the script you must ensure that you have reviewed the sections above and performed the tasks listed in them.

Step 1

Run the cluster installer script `/opt/cisco/package/vtc/bin/cluster_install.sh` on both VTC1 and VTC2 .

```
$ cd /opt/cisco/package/vtc/bin
```

```
$ sudo ./cluster_install.sh
```

Output from executing the `cluster_install.sh` would be similar to:

```
rm -rf ../load-dir/*
```

```
rm -f ../jar/*.jar
```

```
cd java && ant -q clean || true
```

```
BUILD SUCCESSFUL
```

```
Total time: 0 seconds
```

```
rm -f java/src/com/tailf/ns/tcm/namespaces/*.java
```

```
mkdir -p ../load-dir
```

```
mkdir -p java/src/com/tailf/ns/tcm/namespaces
```

```
/opt/ncs/current/bin/ncsc `ls tcm-ann.yang` > /dev/null 2>&1 && echo "-a tcm-ann.yang" ` \
```

```
--yangpath ./yang -c -o ../load-dir/tcm.fxs yang/tcm.yang
```

```
/opt/ncs/current/bin/ncsc --java-disable-prefix --exclude-enums --fail-on-warnings --java-package  
com.tailf.ns.tcm.namespaces --emit-java java/src/com/tailf/ns/tcm/namespaces/tcm.java  
../load-dir/tcm.fxs
```

```
cd java && ant -q all
```

```
BUILD SUCCESSFUL
```

```
Total time: 1 second
```

```

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

Stopping ncs: .

Starting ncs: .

Adding system startup for /etc/init.d/pacemaker ...

    /etc/rc0.d/K20pacemaker -> ../init.d/pacemaker
    /etc/rc1.d/K20pacemaker -> ../init.d/pacemaker
    /etc/rc6.d/K20pacemaker -> ../init.d/pacemaker
    /etc/rc2.d/S20pacemaker -> ../init.d/pacemaker
    /etc/rc3.d/S20pacemaker -> ../init.d/pacemaker
    /etc/rc4.d/S20pacemaker -> ../init.d/pacemaker
    /etc/rc5.d/S20pacemaker -> ../init.d/pacemaker

System start/stop links for /etc/init.d/corosync already exist.

* Stopping corosync daemon corosync
[ OK ]

Pacemaker Cluster Manager is already stopped[ OK ]

* Starting corosync daemon corosync

Apr 10 21:08:48 notice [MAIN ] Corosync Cluster Engine ('2.3.3'): started and ready to provide
service.

Apr 10 21:08:48 info [MAIN ] Corosync built-in features: dbus testagents rdma watchdog augeas
pie relro bindnow

[ OK ]

Starting Pacemaker Cluster Manager: [ OK ]

Starting Pacemaker Cluster Manager: [ OK ]

```

HA cluster is installed

Note The following step should be run only on the Master VTC, and not on the Slave VTC. Do not execute the following step more than once.

Before you run this script, ensure that the Master VTC can telnet to both XRv1 and XRv2 using the IP address provided in the `/opt/cisco/package/vtc/bin/cluster.conf` file.

Step 2

On the Master VTC, run `/opt/cisco/package/vtc/bin/master_node_install.sh`.

When the `master_node_install` script is finished, you can see both the public and private VIP using 'ip addr'. Now that the VIP is up, both IOS XRvs will automatically complete their auto-registration.

Enabling IOS XRv High Availability

To enable IOS XRv HA:

-
- Step 1** Connect to each IOS XRv via SSH, and go to `/opt/cisco/package/sr/bin`.
- Step 2** Run `setupXRNC_HA.sh DL_IP`, where `DL_IP` is the IP address of the other IOS XRv in the cluster.
-

Modifying OpenStack Plugin and Host Agent Configuration to use VIP

-
- Step 1** Modify the `/etc/neutron/plugins/openvswitch/ovs_neutron_plugin.ini` and `/etc/neutron/plugin.ini` files in the OpenStack Controller
- Change the URL to point to VTC using the Virtual IP.

Edit these files:

```
/etc/neutron/plugins/openvswitch/ovs_neutron_plugin.ini
/etc/neutron/plugin.ini
```

Edit this section in each file:

```
[ml2_ncs]
url = https://<Virtual IP>:8888/api/running/openstack
```

- Step 2** Restart the Neutron Server service as well as the Neutron VTS Agent service.
- ```
service neutron-server restart
```

```
service neutron-vts-agent restart
```

- Step 3** Make the same configuration changes for all the Compute Nodes in the `/etc/neutron/plugins/openvswitch/ovs_neutron_plugin.ini` file.

```
[ml2_ncs]
url = https://<Virtual IP>:8888/api/running/openstack
```

- Step 4** Restart the Neutron VTS Agent service.

```
service neutron-vts-agent restart
```

**Note** Changing the password and the VIP for Cisco VTS Plugin and VTS Host Agent can also be done using the Host Agent Installer.

---

## Verifying the VTC High Availability

In a stable HA cluster:

- Corosync and Pacemaker should be running.
- There should be a VTC Master/Slave in the Pacemaker Layer.
- There should be a NCS Master/Slave (in the NCS Layer) and it must match the Pacemaker status.
- The virtual IPs should be residing in the VTC Master.

To verifying whether the HA cluster is stable:

---

**Step 1** Check if Corosync is running. Run the command `service corosync status` (run with root privileges)

```
$ sudo service corosync status
```

```
* corosync is running
```

**Step 2** Check if Pacemaker is running. Run the command `service pacemaker status` (run with root privileges).

```
$ sudo service pacemaker status
```

```
pacemakerd (pid 10316) is running...
```

**Step 3** Check the HA status in corosync/pacemaker layer. Run the command `sudo crm_mon -l`

- Verify that both VTC are online.
- Verify that one VTC is Master and the other is Slave.

A sample output is given below:

```
$ sudo crm_mon -l
```

```
Last updated: Wed Apr 1 22:53:37 2015
```

```
Last change: Wed Apr 1 19:56:02 2015 via cibadmin on vtc1
```

```
Stack: corosync
```

```
Current DC: vtc2 (1711341828) - partition with quorum
```

```
Version: 1.1.10-42f2063
```

```
2 Nodes configured
```

```
3 Resources configured
```

```
Online: [vtc1 vtc2]
```

```
ClusterIP (ocf::heartbeat:IPaddr2): Started vtc1
```

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

```
Masters: [vtc1]
```

```
Slaves: [vtc2]
```

- Step 4** Check the HA status in NCS level. Run the command `show ha-cluster`. A sample output is given below:

```
admin@ncs> show ha-cluster
```

```
NAME STATUS

vtc1 master
vtc2 slave
```

```
[ok][2015-04-01 22:56:16]
```

- Step 5** Check if the Virtual IPs are assigned to the Master VTC. Run the command `ip addr`.
- 

## Uninstalling VTC High Availability



**Note**

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

---

Run this script on both the active and standby nodes.

---

```
/opt/cisco/package/vtc/bin/uninstallHA.sh
```



## CHAPTER 5

# Post-Installation Tasks

---

See the *Getting Started with Cisco Virtual Topology System* chapter in the *Cisco Virtual Topology System 2.1.5 User Guide* for details about the tasks that you need to perform after you install Cisco VTS.





## Sample XML Files

The following sections provide sample XML files.

- [Sample XML File—VTC Installation, page 37](#)
- [Sample XML File—IOS XRv Installation, page 39](#)

### 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 />
 <paе />
 </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>
 </devices>
</domain>
```

```

</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—IOS XRv Installation

```

<domain type='kvm' id='1334'>
 <name>XRVR</name>
 <uuid>cdddbfdd-9b0a-4276-8ded-7d03c329d781</uuid>
 <memory unit='KiB'>32389120</memory>
 <currentMemory unit='KiB'>32388608</currentMemory>
 <vcpu placement='static'>6</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/xrnc.qcow2'>/>
 <target dev='vda' bus='virtio'>/>
 <alias name='virtio-disk0'>/>
 <address type='pci' domain='0x0000' bus='0x00' slot='0x06' function='0x0'>/>
 </disk>
 <disk type='file' device='cdrom'>
 <driver name='qemu' type='raw'>/>
 <source file='/home/cisco/xrnc_cfg.iso'>/>
 <target dev='hdc' bus='ide'>/>
 <readonly/>
 <alias name='ide0-1-0'>/>
 <address type='drive' controller='0' bus='1' target='0' unit='0'>/>
 </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>
 <controller type='ide' index='0'>
 <alias name='ide0'>/>
 <address type='pci' domain='0x0000' bus='0x00' slot='0x01' function='0x1'>/>
 </controller>
 <interface type='bridge'>
 <source bridge='br-ex'>/>
 <virtualport type='openvswitch'>
 </virtualport>
 <model type='virtio'>/>
 <address type='pci' domain='0x0000' bus='0x00' slot='0x0c' function='0x0'>/>
 </interface>
 <interface type='bridge'>
 <source bridge='br-inst'>/>

```

```

 <virtualport type='openvswitch'>
 </virtualport>
 <model type='virtio' />
 <address type='pci' domain='0x0000' bus='0x00' slot='0x03' function='0x0' />
 </interface>
 <serial type='tcp'>
 <source mode='bind' host='127.0.0.1' service='9005' />
 <protocol type='telnet' />
 <target port='0' />
 <alias name='serial0' />
 </serial>
 <console type='tcp'>
 <source mode='bind' host='127.0.0.1' service='9005' />
 <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='vnc' port='5901' 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='vga' vram='16384' 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:c197,c493</label>
 <imagelabel>system_u:object_r:svirt_image_t:s0:c197,c493</imagelabel>
</seclabel>
</domain>

```