# Cisco Catalyst 8000V Edge Software Installation And Configuration Guide

**Cisco Catalyst 8000V Edge Software**
Updated April 24, 2026

# Topics included

**Full Cisco Trademarks with Software License.....viii**

**1 Overview of Cisco Catalyst 8000V Virtual Routers.....9**

Cisco Catalyst 8000V Virtual Routers.....10

Operating modes.....10

Router interfaces.....11

**2 Installation Overview.....13**

Installation Overview.....14

Download the installation files.....14

Supported hypervisors.....15

Guidelines and limitations.....16

Pre-installation checklist and related resources.....17

**3 Compatibility Matrix for Cisco Catalyst 8000V in Public and Sovereign IaaS Clouds.....19**

Compatibility Matrix for Public and Sovereign IaaS Clouds.....20

Supported instance types for Amazon Web Services.....20

Supported instance types for Microsoft Azure.....22

Supported instance types for Google Cloud Platform.....26

Supported instance types for Oracle Cloud Interface.....27

Supported instances for sovereign clouds.....28

**4 Install Cisco Catalyst 8000V in VMware ESXi Environment.....31**

Install Cisco Catalyst 8000V in VMware ESXi Environment.....32

Supported features and operations.....32

Installation requirements.....35

Restrictions for deploying in ESXi environment.....37

Deploy the OVA file Using vSphere.....38

Deploy the OVA to the VM.....38

Configure the bootstrap properties.....39

Deploy the VM Using COT.....41

Edit the basic properties of the VM.....42

Edit the custom properties of the VM.....43

Command references for deploying OVA using COT.....44

Create the VM Using the iso File.....46

Performance Enhancements on ESXi Configurations.....48

**5 Install Cisco Catalyst 8000V in KVM Environment.....51**
Install Cisco Catalyst 8000V in KVM Environment.....52
Installation requirements.....52
Create the VM Using the GUI Tool.....54
Create the VM.....55
Add a serial console.....55
Add a host USB device.....55
Customize the VM configuration.....56
Create the VM Using CLI.....57
Performance Enhancements in KVM Environment.....58
KVM halt polling.....62
Troubleshoot KVM Installation Issues.....63

**6 Install Cisco Catalyst 8000V on Nutanix AHV.....65**
Install Cisco Catalyst 8000V on Nutanix AHV.....66
Installation requirements.....66
Supported configurations.....66
Restrictions.....67
Upload the Cisco Catalyst 8000V image file.....67
Install the Virtual Machine Using Prism Central.....68
Configure VM properties.....68
Configure the disk and network settings.....69
Provide custom configuration and create the VM.....69
Verify successful VM installation.....70
Install the Virtual Machine Using Prism Element.....70
Configure disk properties.....70
Create a vNIC.....71
Create the VM.....71
Install the VM using command line interface.....72
Install the VM using REST API.....73
Verify VM creation.....74
Edit the VM properties.....75
Clone a VM.....75
Verify VM template cloning.....76

**7 Install Cisco Catalyst 8000V in Azure Local.....77**
Install Cisco Catalyst 8000V in Azure Local hyperconverged infrastructure.....78
Prerequisites in Azure Local.....78
Installation requirements.....78
Restrictions for deploying Cisco Catalyst 8000V in Azure Local.....79
Deploy the Cisco Catalyst 8000V VM using PowerShell.....79
VM deployment workflow.....80

Configure network settings....85

**8 Install Cisco Catalyst 8000V in NFVIS Environment....87**

Install Cisco Catalyst 8000V in NFVIS Environment....88
Supported hardware....88
Installing the virtual machines....90
Install Cisco Catalyst 8000V in NFVIS 4.5.1 or later versions....90
Install Cisco Catalyst 8000V in NFVIS 4.5.0 or earlier versions....93
Monitor the virtual machine....96
Upgrade and downgrade scenarios....97

**9 Install Cisco Catalyst 8000V in OpenStack Environment....99**

Install Cisco Catalyst 8000V in OpenStack Environment....100
Installation requirements....100
Restrictions for installing in OpenStack....101
Installing Cisco Catalyst 8000V in OpenStack....101

**10 Day 0 Configuration....105**

Day 0 Configuration....106
Bootstrap support across hypervisors and clouds....106
Feature support for Day 0 configuration....106
Prerequisites for Day 0 configuration....107
Limitations for Day 0 configuration....107
Select the bootstrapping mechanism....108
Day 0 Configuration Using .txt or .xml Files....108
Create the bootstrap file....108
Bootstrap properties....109
Sample iosxe_config.txt file....111
Sample ovf-env.xml file....111
Day 0 Configuration for OVF Deployments....113
Day 0 Configuration Using config-drive....113
Day 0 Configuration Using Custom Data....114
Configure the IOS configuration property....115
Configure scripts property....115
Configure the script credentials property....116
Configure Python package property....117
Configure the license property....118
Provide the Day 0 bootstrap file....118
Verify the custom data configuration in Microsoft Azure....119
Verify the custom data configuration in Google Cloud Platform....123
Day 0 Configuration in Controller Mode....123
Verify Router Configuration....124
Frequently Asked Questions....124

**11 Security-Enhanced Linux.....127**
Security-Enhanced Linux.....128
Prerequisites.....128
Restrictions.....128
Configure Security-Enhanced Linux in EXEC Mode.....128
Configure Security-Enhanced Linux in CONFIG Mode.....129
SYSLOG message reference.....129
Verify Security-Enhanced Linux enablement.....130
Troubleshoot Security-Enhanced Linux.....130

**12 Map Network Interfaces to VM Network Interfaces.....133**
Map Network Interfaces to VM Network Interfaces.....134
Supported network interface speeds.....134
Interface mapping limitations and configurations.....134
View interface mapping.....135
Add network interfaces.....135
Remove a network interface.....136
Configure interface mapping for cloned virtual machines.....136

**13 Enable VNF Secure Boot.....139**
VNF Secure Boot.....140
Guidelines.....140
Setup secure boot in ESXi environment.....140
Setup secure boot in KVM environment.....141
Setup secure boot for NFVIS.....141

**14 Configure VRF Route Sharing.....143**
Configure VRF Route Sharing.....144
Prerequisites for VRF Route Sharing.....144
Restrictions for VRF Route Sharing.....144
Sample topology and use cases.....145
Configure VRF Route Sharing.....147
Verify VRF Route Sharing.....148

**15 Configure Console Access.....151**
Configure Console Access.....152
Boot the Cisco Catalyst 8000V as the VM.....152
Post configuration guidelines.....153
Access the VM through the virtual VGA console.....153
Access the VM through the virtual serial port.....154
Open a Telnet session to the console.....155

Modify the console port access after installation....156

**16 Upgrade the Cisco IOS XE Software....159**

Upgrade the Cisco IOS XE Software....160

Prerequisites before upgrading the Cisco IOS XE software....160

Restrictions when upgrading the Cisco IOS XE software....161

Install mode process flow....162

Frequently asked questions....173

Troubleshooting software upgrade issues....174

**17 Configure vCPU Distribution....175**

Configure vCPU Distribution....176

vCPU templates....176

Manage vCPU distribution....178

**18 Manage the device using Web User Interface....181**

Manage the device using Web User Interface....182

Use Basic or Advanced Mode Setup Wizard....182

Configure LAN Settings....182

Configure Primary WAN Settings....182

Configure Secondary WAN Settings....183

Configure Secondary WAN Settings....183

Using Web User Interface for day one setup....184

Monitor and Troubleshoot Device Plug and Play (PnP) Onboarding using WebUI....185

**19 Access and Use the GRUB Mode....187**

Access and Use the GRUB Mode....188

Access the GRUB mode....188

Access the GRUB menu....189

Load a new system image from GRUB menu....189

Modify the configuration register....191

Modify the configuration register settings....193

Verify the configuration register settings....194

**20 Perform a Factory Reset....197**

Factory reset in Cisco Catalyst 8000V....198

Prerequisites for factory reset....199

Restrictions for factory reset....199

Perform a Factory Reset....199

**21 Bridge Domain Interfaces....203**

Information about bridge domain interface....204

Restrictions for bridge domain interfaces....204
Ethernet virtual circuit overview....205
Bridge domain interface encapsulation....205
Assign a MAC address....205
Support for IP protocols....206
IP forward support....206
Packet forwarding....206
Link states of a bridge domain and a bridge domain interface....207
Bridge domain interface statistics....208
Create or delete a bridge domain interface....208
Bridge domain virtual IP interface....208
How to configure a bridge domain interface....209
Display and verify bridge domain interface configuration....211
Configure bridge domain virtual IP interface....212
Associate VIF interface with a bridge domain....213
Verify bridge domain virtual IP interface....213
Example configuration bridge domain virtual IP interface....213

**22 Support for software media termination point....215**
Support for Software Media Termination Point....216
Information about support for software media termination point....216
How to configure support for software media termination point....216
Prerequisites....216
Restrictions....216
Configure support for software media termination point....216

**23 Radio Aware Routing....223**
Radio Aware Routing....224
Benefits....224
Restrictions....224
License requirements....224
System components....224
QoS provisioning on PPPoE extension session....225
Configuration examples for the RAR feature in Bypass Mode....225
Configuration examples for the RAR Feature in Aggregate Mode....227
Verify radio aware routing session details....229
Troubleshoot radio aware routing....235

© 2020-2026 Cisco Systems, Inc. All rights reserved.

# Full Cisco Trademarks with Software License

Full Cisco Trademarks with Software License

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED “AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

All printed copies and duplicate soft copies of this document are considered uncontrolled. See the current online version for the latest version.

Cisco has more than 200 offices worldwide. Addresses and phone numbers are listed on the Cisco website at www.cisco.com/go/offices.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: https://www.cisco.com/c/en/us/about/legal/trademarks.html. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1721R)

# 1 Overview of Cisco Catalyst 8000V Virtual Routers

**Topics:**


- Cisco Catalyst 8000V Virtual Routers


Describes the capabilities and benefits of Cisco Catalyst 8000V. Explains the operating modes and router interfaces for Cisco Catalyst 8000V.

# Cisco Catalyst 8000V Virtual Routers

Provides an overview of the Cisco Catalyst 8000V virtual router.

Cisco Catalyst 8000V Edge Software or Cisco Catalyst 8000V, is a virtual, form-factor router deployed on a virtual machine (VM) running on an x86 server hardware.

Cisco Catalyst 8000V includes a virtual Route Processor and a virtual Forwarding Processor (FP) as part of its architecture, and provides secure connectivity from an enterprise location such as a branch office or a data center, to a public or a private cloud.

### Benefits

Cisco Catalyst 8000V offers these virtualization benefits that are listed in this section:

- **Hardware independence**: The Cisco Catalyst 8000V router uses the benefits of virtualization in the cloud to provide hardware independence. Since Cisco Catalyst 8000V runs on a virtual machine, you can use this router on any x86 hardware that the virtualization platform supports.
- **Sharing of resources**: The resources used by Cisco Catalyst 8000V are managed by the hypervisor, and these resources can be shared amongst the VMs. The amount of hardware resources that the VM server allocates to a specific VM can be regulated and resources can also be reallocated to other VMs on the server.
- **Flexibility in deployment**: VMs can be moved from one server to another. A Cisco Catalyst 8000V instance can be moved from a server in one physical location to a server in another physical location without moving any hardware resources.
- **Enhanced software security through Secure Object Store**: In Cisco Catalyst 8000V, storage partitions for NVRAM, licensing, and other data are created as Object stores. The individual Object stores are encrypted to ensure data security, and this product is Cisco Secure Development lifecycle (CSDL) compliant. Further, Cisco Catalyst 8000V supports a 16G disk profile.

## Operating modes

Provides information on the modes in which the Cisco Catalyst 8000V router operates.

Cisco Catalyst 8000V virtual router runs on Cisco IOS XE and Cisco IOS XE SD-WAN, and supports these modes:

- **Autonomous mode**: Offers a subset of the Cisco IOS XE software features and technologies, and provides Cisco IOS XE security and switching features on a virtualization platform.
- **Controller mode**: Delivers comprehensive SD-WAN, WAN gateway, and network services functions in the virtual and cloud environments.
- **SD-Routing mode**: Offers an operating environment where a Cisco Catalyst 8000V instance operating in autonomous routing mode can be managed by the Cisco Catalyst SD-WAN manager.

This guide provides the overview, installation, and configuration information for Cisco Catalyst 8000V running on Cisco IOS XE or autonomous mode.

### Cisco Catalyst 8000V command line interface

Before you install or configure this router, set up access to the command line interface (CLI). The CLI is the primary interface for configuring and using Cisco Catalyst 8000V in autonomous mode. You can accomplish this in these ways:

- Provision a serial port in the VM to connect and access the Cisco IOS XE CLI commands.
- Use a serial port to manage a Cisco Catalyst 8000V VM only if the underlying hypervisor supports associating a serial port with a VM. For more details, see the respective hypervisor installation information that is available in this document.

- Use the remote SSH/Telnet to access the Cisco IOS XE CLI commands. By default, Telnet is disabled for security reasons. In an on-premises deployment, SSH is also disabled. Although SSH is preferred for remote user management, you must enable SSH manually in an on-premises deployment.

  In cloud deployments, SSH is enabled by default. To access SSH, ensure that your cloud security settings allow SSH connectivity for both inbound and outbound traffic.

The software for Cisco Catalyst 8000V uses the standard Cisco IOS XE CLI commands and conventions. The commands are not case sensitive, and you can abbreviate the commands and parameters if the abbreviations contain enough letters to be different from any other currently available commands or parameters. To access all the features of Cisco IOS XE CLI and how to use them, see the System Management Configuration Guide.

## Router interfaces

Router Gigabit Ethernet interfaces, interface numbering and mapping in Cisco Catalyst 8000V.

The Cisco Catalyst 8000V router interfaces function in the same way as those on hardware-based Cisco routers. The Cisco Catalyst 8000V interfaces function in these ways:

- The interfaces are logically named as the Gigabit Ethernet (GE) interfaces.
- The available interface numbering depends on the Cisco Catalyst 8000V version.

### Interface numbering

- Interface port numbering starts at 1 and continues up to the total number of interfaces supported. For more information about supported vNICs and the minimum and maximum number of vNICs supported for each VM instance, see Installation requirements on page 35.
- Gigabit Ethernet interface 0 is not supported.
- Designate any interface as the management interface. You can designate a management interface by performing the appropriate Day0 bootstrapping mechanisms available for your target environment. For more details, see Day 0 Configuration on page 106.

### Router interface mapping

When you first boot the device, the Cisco Catalyst 8000V router interfaces are mapped to the vNIC interfaces on the VM based on the vNIC enumeration to the Cisco Catalyst 8000V. On subsequent boot, the Cisco Catalyst 8000V router interfaces are mapped to the vNIC MAC addresses.

For more information, see Mapping the Cisco Catalyst 8000V Network Interfaces to the VM Network Interfaces.

# 2 Installation Overview

**Topics:**


- Installation Overview


Provides an overview of the Cisco Catalyst 8000V Edge Software installation process, covering supported hypervisors and public cloud platforms, various operating modes like Autonomous and SD-WAN, and the general workflow required for successful deployment.

# Installation Overview

Provides high-level information on installing Cisco Catalyst 8000V.

This chapter provides high-level information about installing Cisco Catalyst 8000V. Cisco hardware routers are usually shipped with the Cisco IOS XE software pre-installed. Because Cisco Catalyst 8000V is not a hardware-based router, you need to download the Cisco IOS XE software from Cisco.com and install the virtual router directly on the virtual machine.

Before you proceed to the installation, provision the VM attributes so that the Cisco Catalyst 8000V software can install and boot.

Refer to the sections in this chapter for details about the installation files and options, which depend on your chosen hypervisor.

## Download the installation files

Details the steps to download the installation files for Cisco Catalyst 8000V.

Perform these steps to download the installation files for Cisco Catalyst 8000V.

### Procedure

1. Go to the Cisco Software Download page.
2. From the **Select a Product** field at the bottom of the page, search for Cisco Catalyst 8000V.
3. Click the Cisco Catalyst 8000V link and go to the **Download** page.
4. From the left pane, select the appropriate release, for example, *Bengaluru 17.4.1*.
5. From the list of available images, click **Download** or **Add to Cart**. Then, follow the instructions to download the software.

> **Note**
>
> To know which installation file you want to download, see Installation Files.

### Installation files

This section specifies the software images that are available for installing Cisco Catalyst 8000V on the supported hypervisors.

This section specifies the software images that are available for installing Cisco Catalyst 8000V on the supported hypervisors.

| Image type | Hypervisor | Mode | Secure boot | Sample filename |
|---|---|---|---|---|
| bin | ESXi, KVM, AWS, Microsoft Azure, GCP, Azure Local | Upgrade (bundle mode)<br>Upgrade (install mode) | No | c8000v-universalk9.17.04.01a.SPA.bin |

| Image type | Hypervisor | Mode | Secure boot | Sample filename |
|---|---|---|---|---|
| iso - Used to install the software image on the VM | ESXi, KVM | New installation | No | c8000v-universalk9.17.04.01a.iso |
| ova - Used to deploy the OVA template on the VM (in TAR format) | ESXi | New installation | Yes | c8000v-universalk9.17.04.01a.ova |
| qcow2 - Used to install the software image in KVM environments. | KVM | New installation | No | c8000v-universalk9.17.04.01a.qcow2 |
| serial.qcow2 | KVM | New installation | No | c8000v-universalk9.17.04.01a.efi.qcow2 |
| efi.qcow2 | KVM | New installation | Yes | c8000v-universalk9.17.04.01a.efi.qcow2 |
| serial.efi.qcow2 | KVM | New installation | Yes | c8000v-universalk9.17.04.01a-serial.efi.qcow2 |
| tar.gz | NFVIS | New installation | Yes | c8000v-universalk9.17.04.01a-tar.gz |
| iso | Azure Local | New installation | No | c8000v-universalk9.17.04.01a.iso |

## Supported hypervisors

This section lists all the hypervisors that are supported for installing Cisco Catalyst 8000V.

This section lists all the hypervisors that are supported for installing Cisco Catalyst 8000V.

A hypervisor enables multiple operating systems to share a single hardware host machine. Each operating system could use the host's processor, memory, and other resources, but the hypervisor controls and allocates only the required resources to each one. This ensures that the operating systems (VMs) do not disrupt each other. The supported hypervisors for Cisco Catalyst 8000V are:

- **VMware ESXi**: Cisco Catalyst 8000V runs on the VMware ESXi hypervisor, which runs on a x86 hardware containing virtualization extension. To see the VMware requirements and to learn how to install Cisco Catalyst 8000V in the ESXi environment, see Installing in VMware ESXi Environment.
- **Red Hat KVM**: Cisco Catalyst 8000V also runs on the Red Hat Enterprise Linux (RHEL). For more information on Supported RHEL versions (Linux KVM based on Red Hat Enterprise Linux), see Install Cisco Catalyst 8000V in KVM Environment.
- **Public Clouds**: Apart from the listed hypervisors, you can also deploy and use Cisco Catalyst 8000V in Amazon Web Services, Microsoft Azure, Google Cloud Platform, Alibaba Cloud, and Oracle Cloud Infrastructure. See the respective public cloud deployment guides for detailed information.
- **Nutanix AHV**: Cisco Catalyst 8000V also runs on the Nutanix AHV Hypervisor. For more information, see Install Cisco Catalyst 8000V in Nutanix AHV Hypervisor.

### Virtual machine processing resources

The Cisco Catalyst 8000V is a low-latency application. It not function properly if processing resources on the host are oversubscribed. By default, most hypervisors support overcommitting the processing resources. However, for Cisco Catalyst 8000V, if you oversubscribe and do not schedule the virtual CPUs (vCPUs) reliably, you could experience packet processing drops, error messages, or system outages.

The Cisco Catalyst 8000V vCPUs must be scheduled by the host hypervisor to run on real physical cores. Each hypervisor has various controls that influence the scheduling of the vCPUs to the physical cores. As a best practice, Cisco recommends that you to use a ratio of 1:1 for the vCPUs to real physical cores.

For detailed information on virtual machine processing resources, see the respective hypervisor tuning guides provided by the hypervisor. Additionally, refer to the appropriate hypervisor sections in this guide that provide all the information on installing and using Cisco Catalyst 8000V in the hypervisor.

## Guidelines and limitations

This section specifies the general guidelines and limitations you should consider before installing a Cisco Catalyst 8000V router.

This section specifies the general guidelines you should be aware of before installing a Cisco Catalyst 8000V router.

- You can access the Cisco IOS XE CLI either through the virtual VGA console or the console on the virtual serial port. Select the console from the GRUB mode during the first-time installation or change the console using the Cisco IOS XE **platform console** command after the router boots. For more information, see Access and Use the GRUB Mode on page 188.
- Some hypervisors might not support serial console access. Verify support using your hypervisor documentation.
- For .qcow2 files, the image selected during installation determines the type of console you can select.
- If the hypervisor does not support vNIC Hot Add or Hot Remove, do not make any changes to the VM hardware (memory, CPUs, hard drive size, and so on) while the VM is powered on.
- Although Cisco Catalyst 8000V instances in controller mode support the 4GB memory configuration, Cisco recommends using 8GB of memory for feature-rich and high-scale environments to prevent high memory utilization issues.
- From Cisco IOS XE 17.6.1, you can enable the FIPS mode if the host and VM supports RDRAND or RDSEED, or both instructions. Otherwise, an error message is displayed.

> **Note**
>
> Some hypervisors have configuration options or runtime options to block the use of RDSEED or RDRAND, or both in a VM. These options must not be enabled. That is, RDSEED or RDRAND, or both must not be blocked by the hypervisor if you want to enable the FIPS mode.

### Limitations

This list provides the limitations or restrictions that are applicable while installing Cisco Catalyst 8000V.

- Cisco Catalyst 8000V within a nested VM has not been tested and is not recommended for this reason.
- Gigabit Ethernet0 interface is no longer available. You can designate any interface as the management interface.
- If you are running a virtual function on an I350 device, redundancy protocols like HSRP/VRRP are not supported.
- vNICs do not support duplex settings in an interface.

- vNICs do not support auto-negotiations.
- From Cisco IOS XE 17.9.1, the **show license udi** command is no longer supported in Cisco Catalyst 8000V.
- Cisco Catalyst 8000V does not support L2TP functionality including L2TP client and L2TP Network Server (LNS).

## Pre-installation checklist and related resources

See this section to view the information that you should be aware of before you proceed with the Cisco Catalyst 8000V installation.

Before you install Cisco Catalyst 8000V, verify:

- The server hardware is supported by the hypervisor vendor.

  If you're using VMware, verify that the server is listed on the VMware Hardware Compatibility List. See the VMware documentation for more information.
- The I/O devices (for example, FC, iSCSI, SAS) being used are supported by the VM vendor.
- Sufficient RAM is allocated on the server for the VMs and the hypervisor host. If you're using VMware, ensure that the server has enough RAM to support both the VMs and VMware ESXi.
- The hypervisor version is supported by Cisco Catalyst 8000V.
- The correct VM settings for the amount of memory, number of CPUs, and disk size are configured.
- The vNICs are configured using a supported network driver.

### Resources for installing Cisco Catalyst 8000V

After you download the installation file, the deployment procedures depend on the hypervisor or public cloud you select. See this table to learn how to deploy Cisco Catalyst 8000V.

| Hypervisor/cloud | Guide or chapter |
|---|---|
| Installing Cisco Catalyst 8000V in VMware | Installing in VMware ESXi Environment |
| Installing in Cisco Catalyst 8000V in KVM | Installing in Kernel Virtual Machine Support (KVM) Environments |
| Installing in Cisco Catalyst 8000V in Amazon Web Services | Deploying Cisco Catalyst 8000V Edge Software in Amazon Web Services |
| Installing in Cisco Catalyst 8000V in Microsoft Azure | Deploying Cisco Catalyst 8000V in Microsoft Azure |
| Installing in Cisco Catalyst 8000V in Google Cloud Platform | Deploying Cisco Catalyst 8000V in Google Cloud Platform |
| Installing in Cisco Catalyst 8000V in Alibaba Cloud | Deploying Cisco Catalyst 8000V in Alibaba Cloud |
| Installing in Cisco Catalyst 8000V in Oracle Cloud Infrastructure | Deploying Cisco Catalyst 8000V in Oracle Cloud Infrastructure |
| Installing in Cisco Catalyst 8000V in Nutanix AHV Hypervisor | Installing in Nutanix AHV Hypervisor |
| Installing in Cisco Catalyst 8000V in Azure Local | Installing in Azure Local |

# 3 Compatibility Matrix for Cisco Catalyst 8000V in Public and Sovereign IaaS Clouds

**Topics:**


- Compatibility Matrix for Public and Sovereign IaaS Clouds


Provides the hardware and software compatibility requirements for the Cisco Catalyst 8000V, including supported hypervisors, public cloud platforms, and the specific virtual machine resource allocations (vCPU, RAM, and disk) required for various performance tiers and network interfaces.

# Compatibility Matrix for Public and Sovereign IaaS Clouds

Specifies the supported public cloud compute instance types and sizes for Cisco Catalyst 8000V public cloud installation.

This chapter explains the supported public cloud compute instance sizes for Cisco Catalyst 8000V routers. Each of this section describes the compute instance details for every supported public cloud and sovereign IaaS cloud.

## Supported instance types for Amazon Web Services

This topic specifies the supported instance types for Cisco Catalyst 8000V running on AWS.

Cisco Catalyst 8000V running on Amazon Web Services (AWS) supports the instance types that are specified in this table. The instance type determines the size of the instance and the required amount of memory.

| Release number | Supported instance types |
|---|---|
| Cisco IOS XE 26.1.1 | • t3.medium<br>• c5.9xlarge, c5.2xlarge, c5.xlarge, c5.large<br>• c5n.18xlarge, c5n.4xlarge<br>• c6in.8xlarge, c6in.2xlarge, c6in.xlarge, c6in.large |
| Cisco IOS XE 17.18.1a | • t3.medium<br>• c5.9xlarge, c5.2xlarge, c5.xlarge, c5.large<br>• c5n.18xlarge, c5n.4xlarge<br>• c6in.8xlarge, c6in.2xlarge, c6in.xlarge, c6in.large |
| Cisco IOS XE 17.16.1a | • t3.medium<br>• c5.9xlarge, c5.2xlarge, c5.xlarge, c5.large<br>• c5n.18xlarge, c5n.4xlarge<br>• c6in.8xlarge, c6in.2xlarge, c6in.xlarge, c6in.large |
| Cisco IOS XE 17.15.5<br>Cisco IOS XE 17.15.4c<br>Cisco IOS XE 17.15.4<br>Cisco IOS XE 17.15.3a<br>Cisco IOS XE 17.15.2a<br>Cisco IOS XE 17.15.1a | • t3.medium<br>• c5.9xlarge, c5.2xlarge, c5.xlarge, c5.large<br>• c5n.18xlarge, c5n.4xlarge<br>• c6in.8xlarge, c6in.2xlarge, c6in.xlarge, c6in.large |
| Cisco IOS XE 17.14.1a | • t3.medium<br>• c5.9xlarge, c5.2xlarge, c5.xlarge, c5.large<br>• c5n.18xlarge, c5n.4xlarge<br>• c6in.8xlarge, c6in.2xlarge, c6in.xlarge, c6in.large |

| Release number | Supported instance types |
|---|---|
| Cisco IOS XE 17.13.1a | • t3.medium<br>• c5.9xlarge, c5.2xlarge, c5.xlarge, c5.large<br>• c5n.18xlarge, c5n.4xlarge<br>• c6in.8xlarge, c6in.2xlarge, c6in.xlarge, c6in.large |
| Cisco IOS XE 17.12.7<br>Cisco IOS XE 17.12.6<br>Cisco IOS XE 17.12.5d<br>Cisco IOS XE 17.12.5c<br>Cisco IOS XE 17.12.5b<br>Cisco IOS XE 17.12.5a<br>Cisco IOS XE 17.12.4b<br>Cisco IOS XE 17.12.4a<br>Cisco IOS XE 17.12.2<br>Cisco IOS XE 17.12.1 | • t3.medium<br>• c5.9xlarge, c5.2xlarge, c5.xlarge, c5.large<br>• c5n.18xlarge, c5n.9xlarge, c5n.4xlarge, c5n.2xlarge, c5n.xlarge, c5n.large |
| Cisco IOS XE 17.11.1a | • t3.medium<br>• c5.9xlarge, c5.2xlarge, c5.xlarge, c5.large<br>• c5n.18xlarge, c5n.9xlarge, c5n.4xlarge, c5n.2xlarge, c5n.xlarge, c5n.large |
| Cisco IOS XE 17.10.1a | • t3.medium<br>• c5.9xlarge, c5.2xlarge, c5.xlarge, c5.large<br>• c5n.18xlarge, c5n.9xlarge, c5n.4xlarge, c5n.2xlarge, c5n.xlarge, c5n.large |
| Cisco IOS XE 17.9.9<br>Cisco IOS XE 17.9.8<br>Cisco IOS XE 17.9.7b<br>Cisco IOS XE 17.9.7a<br>Cisco IOS XE 17.9.6<br>Cisco IOS XE 17.9.5f<br>Cisco IOS XE 17.9.5e<br>Cisco IOS XE 17.9.4a<br>Cisco IOS XE 17.9.4<br>Cisco IOS XE 17.9.3a<br>Cisco IOS XE 17.9.2a<br>Cisco IOS XE 17.9.1a | • t3.medium<br>• c5.9xlarge, c5.4xlarge, c5.2xlarge, c5.xlarge, c5.large<br>• c5n.18xlarge, c5n.9xlarge, c5n.4xlarge, c5n.2xlarge, c5n.xlarge, c5n.large |

| Release number | Supported instance types |
|---|---|
| Cisco IOS XE 17.8.1a | • t3.medium<br>• c5.9xlarge, c5.4xlarge, c5.2xlarge, c5.xlarge, c5.large<br>• c5n.9xlarge, c5n.4xlarge, c5n.2xlarge, c5n.xlarge, c5n.large |
| Cisco IOS XE 17.7.2<br>Cisco IOS XE 17.7.1a | • t3.medium<br>• c5.9xlarge, c5.4xlarge, c5.2xlarge, c5.xlarge, c5.large<br>• c5n.9xlarge, c5n.4xlarge, c5n.2xlarge, c5n.xlarge, c5n.large |
| Cisco IOS XE 17.6.8a<br>Cisco IOS XE 17.6.7<br>Cisco IOS XE 17.6.6a<br>Cisco IOS XE 17.6.6<br>Cisco IOS XE 17.6.5a<br>Cisco IOS XE 17.6.5<br>Cisco IOS XE 17.6.4<br>Cisco IOS XE 17.6.3a<br>Cisco IOS XE 17.6.2<br>Cisco IOS XE 17.6.1a | • t3.medium<br>• c5.9xlarge, c5.4xlarge, c5.2xlarge, c5.xlarge, c5.large<br>• c5n.9xlarge, c5n.4xlarge, c5n.2xlarge, c5n.xlarge, c5n.large |
| Cisco IOS XE 17.5.1a | • t3.medium, t2.medium<br>• c4.8xlarge, c4.4xlarge, c4.2xlarge, c4.xlarge, c4.large<br>• c5.9xlarge, c5.4xlarge, c5.2xlarge, c5.xlarge, c5.large<br>• c5n.9xlarge, c5n.4xlarge, c5n.2xlarge, c5n.xlarge, c5n.large |
| Cisco IOS XE 17.4.2<br>Cisco IOS XE 17.4.1b<br>Cisco IOS XE 17.4.1a | • t3.medium, t2.medium<br>• c4.8xlarge, c4.4xlarge, c4.2xlarge, c4.xlarge, c4.large<br>• c5.9xlarge, c5.4xlarge, c5.2xlarge, c5.xlarge, c5.large<br>• c5n.9xlarge, c5n.4xlarge, c5n.2xlarge, c5n.xlarge, c5n.large |

For more information about the instance types, see Amazon EC2 Instance Types.

## Supported instance types for Microsoft Azure

This topic specifies the instance types that are supported for Cisco Catalyst 8000V deployment in Microsoft Azure.

This topic specifies the instance types that are supported for Cisco Catalyst 8000V deployment in Microsoft Azure. The 2, 4 and 8 NIC solution templates mentioned in this table are currently offered in the Microsoft Azure marketplace in the public cloud.

| Cisco IOS XE release | Supported instance types/max NICs supported |
|---|---|
| Cisco IOS XE 26.1.1 | • Dsv5 series: Standard_D16s_v5, Standard_D8s_v5, Standard_D4s_v5<br>• Dv5 series: Standard_D16_v5, Standard_D8_v5, Standard_D4_v5<br>• Dsv2 series: Standard_DS4_v2, Standard_DS3_v2<br>• Dv2 series: Standard_D4_v2, Standard_D3_v2<br>• Fsv2 series: Standard_F32s_v2, Standard_F16s_v2 |
| Cisco IOS XE 17.18.1a | • Dsv5 series: Standard_D16s_v5, Standard_D8s_v5, Standard_D4s_v5<br>• Dv5 series: Standard_D16_v5, Standard_D8_v5, Standard_D4_v5<br>• Dsv2 series: Standard_DS4_v2, Standard_DS3_v2<br>• Dv2 series: Standard_D4_v2, Standard_D3_v2<br>• Fsv2 series: Standard_F32s_v2, Standard_F16s_v2 |
| Cisco IOS XE 17.16.1a | • Dsv5 series: Standard_D16s_v5<br>• Dsv2 series: Standard_DS4_v2, Standard_DS3_v2, Standard_DS2_v2<br>• Dv2 series: Standard_D4_v2, Standard_D3_v2, Standard_D2_v2<br>• Fsv2 series: Standard_F32s_v2, Standard_F16s_v2 |
| Cisco IOS XE 17.15.5<br>Cisco IOS XE 17.15.4c<br>Cisco IOS XE 17.15.4<br>Cisco IOS XE 17.15.3a<br>Cisco IOS XE 17.15.2a<br>Cisco IOS XE 17.15.1a | • Dsv5 series: Standard_D16s_v5, Standard_D8s_v5, Standard_D4s_v5<br>• Dv5 series: Standard_D16_v5, Standard_D8_v5, Standard_D4_v5<br>• Dsv2 series: Standard_DS4_v2, Standard_DS3_v2, Standard_DS2_v2<br>• Dv2 series: Standard_D4_v2, Standard_D3_v2, Standard_D2_v2<br>• Fsv2 series: Standard_F32s_v2, Standard_F16s_v2 |
| Cisco IOS XE 17.14.1a | • Dv5 series: Standard_D16_v5<br>• Dsv2 series: Standard_DS4_v2, Standard_DS3_v2, Standard_DS2_v2<br>• Dv2 series: Standard_D4_v2, Standard_D3_v2, Standard_D2_v2<br>• Fsv2 series: Standard_F32s_v2, Standard_F16s_v2 |

| Cisco IOS XE release | Supported instance types/max NICs supported |
|---|---|
| Cisco IOS XE 17.13.1a | • Dv5 series: Standard_D16_v5<br>• Dsv2 series: Standard_DS4_v2, Standard_DS3_v2, Standard_DS2_v2<br>• Dv2 series: Standard_D4_v2, Standard_D3_v2, Standard_D2_v2<br>• Fsv2 series: Standard_F32s_v2, Standard_F16s_v2 |
| Cisco IOS XE 17.12.7<br>Cisco IOS XE 17.12.6<br>Cisco IOS XE 17.12.5d<br>Cisco IOS XE 17.12.5c<br>Cisco IOS XE 17.12.5b<br>Cisco IOS XE 17.12.5a<br>Cisco IOS XE 17.12.4b<br>Cisco IOS XE 17.12.4a | • Dsv5 series: Standard_D16s_v5, Standard_D8s_v5, Standard_D4s_v5<br>• Dv5 series: Standard_D16_v5, Standard_D8_v5, Standard_D4_v5<br>• Dsv2 series: Standard_DS4_v2, Standard_DS3_v2, Standard_DS2_v2<br>• Dv2 series: Standard_D4_v2, Standard_D3_v2, Standard_D2_v2<br>• Fsv2 series: Standard_F32s_v2, Standard_F16s_v2 |
| Cisco IOS XE 17.12.2<br>Cisco IOS XE 17.12.1 | • Dv5 series: Standard_D16_v5<br>• Dsv2 series: Standard_DS4_v2, Standard_DS3_v2, Standard_DS2_v2<br>• Dv2 series: Standard_D4_v2, Standard_D3_v2, Standard_D2_v2<br>• Fsv2 series: Standard_F32s_v2, Standard_F16s_v2 |
| Cisco IOS XE 17.11.1a | • Dsv2 series: Standard_DS4_v2, Standard_DS3_v2, Standard_DS2_v2<br>• Dv2 series: Standard_D4_v2, Standard_D3_v2, Standard_D2_v2<br>• Fsv2 series: Standard_F32s_v2, Standard_F16s_v2 |
| Cisco IOS XE 17.10.1a | • Dsv2 series: Standard_DS4_v2, Standard_DS3_v2, Standard_DS2_v2<br>• Dv2 series: Standard_D4_v2, Standard_D3_v2, Standard_D2_v2<br>• Fsv2 series: Standard_F32s_v2, Standard_F16s_v2 |

| Cisco IOS XE release | Supported instance types/max NICs supported |
|---|---|
| Cisco IOS XE 17.9.9<br>Cisco IOS XE 17.9.8<br>Cisco IOS XE 17.9.7b<br>Cisco IOS XE 17.9.7a<br>Cisco IOS XE 17.9.6<br>Cisco IOS XE 17.9.5f<br>Cisco IOS XE 17.9.5e<br>Cisco IOS XE 17.9.4a<br>Cisco IOS XE 17.9.3a<br>Cisco IOS XE 17.9.2a<br>Cisco IOS XE 17.9.1a | • Dsv2 series: Standard_DS4_v2, Standard_DS3_v2, Standard_DS2_v2<br>• Dv2 series: Standard_D4_v2, Standard_D3_v2, Standard_D2_v2<br>• Fsv2 series: Standard_F32s_v2, Standard_F16s_v2 |
| Cisco IOS XE 17.8.1a | • Dsv2 series: Standard_DS4_v2, Standard_DS3_v2, Standard_DS2_v2<br>• Dv2 series: Standard_D4_v2, Standard_D3_v2, Standard_D2_v2<br>• Fsv2 series: Standard_F32s_v2, Standard_F16s_v2 |
| Cisco IOS XE 17.7.2<br>Cisco IOS XE 17.7.1a | • Dsv2 series: Standard_DS4_v2, Standard_DS3_v2, Standard_DS2_v2<br>• Dv2 series: Standard_D4_v2, Standard_D3_v2, Standard_D2_v2<br>• Fsv2 series: Standard_F32s_v2, Standard_F16s_v2 |
| Cisco IOS XE 17.6.8a<br>Cisco IOS XE 17.6.7<br>Cisco IOS XE 17.6.6a<br>Cisco IOS XE 17.6.5a,<br>Cisco IOS XE 17.6.4a<br>Cisco IOS XE 17.6.3a<br>Cisco IOS XE 17.6.2a<br>Cisco IOS XE 17.6.1a | • Dsv2 series: Standard_DS4_v2, Standard_DS3_v2, Standard_DS2_v2<br>• Dv2 series: Standard_D4_v2, Standard_D3_v2, Standard_D2_v2<br>• Fsv2 series: Standard_F32s_v2, Standard_F16s_v2 |
| Cisco IOS XE 17.5.1a | • Dsv2 series: Standard_DS4_v2, Standard_DS3_v2, Standard_DS2_v2<br>• Dv2 series: Standard_D4_v2, Standard_D3_v2, Standard_D2_v2<br>• Fsv2 series: Standard_F32s_v2, Standard_F16s_v2 |

| Cisco IOS XE release | Supported instance types/max NICs supported |
|---|---|
| Cisco IOS XE 17.4.2<br>Cisco IOS XE 17.4.1b<br>Cisco IOS XE 17.4.1a | • Dsv2 series: Standard_DS4_v2, Standard_DS3_v2, Standard_DS2_v2<br>• Dv2 series: Standard_D4_v2, Standard_D3_v2, Standard_D2_v2<br>• Fsv2 series: Standard_F32s_v2, Standard_F16s_v2 |

## Supported instance types for Google Cloud Platform

This topic specifies the supported instance types for Cisco Catalyst 8000V running on GCP.

Cisco Catalyst 8000V running on Google Cloud Platform (GCP) supports the instance types that are specified in this table.

| Cisco IOS XE release | Supported instance types (BYOL only) |
|---|---|
| Cisco IOS XE 26.1.1 | N4: n4-standard-32, n4-standard-4<br>N2: n2-standard-8, n2-standard-4 |
| Cisco IOS XE 17.18.1a | N4: n4-standard-32, n4-standard-4<br>N2: n2-standard-8, n2-standard-4 |
| Cisco IOS XE 17.16.1a | N1: n1-standard-8, n1-standard-4 |
| Cisco IOS XE 17.15.5<br>Cisco IOS XE 17.15.4c<br>Cisco IOS XE 17.15.4<br>Cisco IOS XE 17.15.3a<br>Cisco IOS XE 17.15.2a<br>Cisco IOS XE 17.15.1a | N1: n1-standard-8, n1-standard-4<br>N2: n2-standard-8, n2-standard-4 |
| Cisco IOS XE 17.14.1a | N1: n1-standard-8, n1-standard-4 |
| Cisco IOS XE 17.13.1a | N1: n1-standard-8, n1-standard-4 |
| Cisco IOS XE 17.12.7<br>Cisco IOS XE 17.12.6<br>Cisco IOS XE 17.12.5d<br>Cisco IOS XE 17.12.5c<br>Cisco IOS XE 17.12.5b<br>Cisco IOS XE 17.12.5a<br>Cisco IOS XE 17.12.4b<br>Cisco IOS XE 17.12.4a | N1: n1-standard-8, n1-standard-4<br>N2: n2-standard-8, n2-standard-4 |
| Cisco IOS XE 17.12.2<br>Cisco IOS XE 17.12.1a | N1: n1-standard-8, n1-standard-4 |
| Cisco IOS XE 17.11.1a | N1: n1-standard-8, n1-standard-4, n1-standard-2 |

| Cisco IOS XE release | Supported instance types (BYOL only) |
|---|---|
| Cisco IOS XE 17.10.1a | N1: n1-standard-8, n1-standard-4, n1-standard-2 |
| Cisco IOS XE 17.9.9 | N1: n1-standard-8, n1-standard-4, n1-standard-2 |
| Cisco IOS XE 17.9.8 | |
| Cisco IOS XE 17.9.7b | |
| Cisco IOS XE 17.9.7a | |
| Cisco IOS XE 17.9.6 | |
| Cisco IOS XE 17.9.5f | |
| Cisco IOS XE 17.9.5e | |
| Cisco IOS XE 17.9.4a | |
| Cisco IOS XE 17.9.3a | |
| Cisco IOS XE 17.9.2a | |
| Cisco IOS XE 17.9.1a | |
| Cisco IOS XE 17.8.1a | N1: n1-standard-8, n1-standard-4, n1-standard-2 |
| Cisco IOS XE 17.7.2 | N1: n1-standard-8, n1-standard-4, n1-standard-2 |
| Cisco IOS XE 17.7.1a | |
| Cisco IOS XE 17.6.8a | N1: n1-standard-8, n1-standard-4, n1-standard-2 |
| Cisco IOS XE 17.6.7 | |
| Cisco IOS XE 17.6.6a | |
| Cisco IOS XE 17.6.5a | |
| Cisco IOS XE 17.6.5 | |
| Cisco IOS XE 17.6.4 | |
| Cisco IOS XE 17.6.3a | |
| Cisco IOS XE 17.6.2 | |
| Cisco IOS XE 17.6.1a | |
| Cisco IOS XE 17.5.1a | N1: n1-standard-8, n1-standard-4, n1-standard-2 |
| Cisco IOS XE 17.4.2 | N1: n1-standard-8, n1-standard-4, n1-standard-2, n1-standard-1 |
| Cisco IOS XE 17.4.1b | |
| Cisco IOS XE 17.4.1a | |

## Supported instance types for Oracle Cloud Interface

This section specifies the instance types that are supported for OCI deployment.

This section specifies the instance types supported for Oracle Cloud Infrastructure (OCI) deployments in Cisco Catalyst 8000V.

| Cisco IOS XE release | Supported instance types (BYOL) |
|---|---|
| Cisco IOS XE 26.1.1 | VM.Standard.E5.Flex with 4vCPU/8GB<br>VM.Standard.E5.Flex with 16vCPU/16GB |
| Cisco IOS XE 17.18.x and later | VM.Standard.E5.Flex with 4vCPU/8GB<br>VM.Standard.E5.Flex with 16vCPU/16GB |

## Supported instances for sovereign clouds

This topic specifies the supported instances types for the sovereign clouds (China)

Cisco Catalyst 8000V running on sovereign clouds support the instance types that are specified in the tables in this section.

**Table 1: Supported Instances for Alibaba Cloud (China)**

| Cisco IOS XE release | Supported instance types (BYOL only) |
|---|---|
| Cisco IOS XE 26.1.1 | • ecs.g5.4xlarge<br>• ecs.g5.large |
| Cisco IOS XE 17.18.x<br>Cisco IOS XE 17.16.x<br>Cisco IOS XE 17.15.x<br>Cisco IOS XE 17.14.x<br>Cisco IOS XE 17.13.x<br>Cisco IOS XE 17.12.x<br>Cisco IOS XE 17.11.x<br>Cisco IOS XE 17.10.x<br>Cisco IOS XE 17.9.x<br>Cisco IOS XE 17.8.x<br>Cisco IOS XE 17.7.x<br>Cisco IOS XE 17.6.x | • ecs.g5.4xlarge<br>• ecs.g5.large |

**Table 2: Supported Instances for AWS (China)**

| Cisco IOS XE release | Supported instance types (BYOL only) |
|---|---|
| Cisco IOS XE 17.18.x | • t3.medium<br>• c4.large<br>• c5.9xlarge, c5.4xlarge, c5.2xlarge, c5.xlarge, c5.large |

| Cisco IOS XE release | Supported instance types (BYOL only) |
|---|---|
| Cisco IOS XE 17.16.x | • t3.medium<br>• c4.large<br>• c5.9xlarge, c5.4xlarge, c5.2xlarge, c5.xlarge, c5.large |
| Cisco IOS XE 17.15.x | • t3.medium<br>• c4.large<br>• c5.9xlarge, c5.4xlarge, c5.2xlarge, c5.xlarge, c5.large |
| Cisco IOS XE 17.14.x | • t3.medium<br>• c4.large<br>• c5.9xlarge, c5.4xlarge, c5.2xlarge, c5.xlarge, c5.large |
| Cisco IOS XE 17.13.x | • t3.medium<br>• c4.large<br>• c5.9xlarge, c5.4xlarge, c5.2xlarge, c5.xlarge, c5.large |
| Cisco IOS XE 17.12.x | • t3.medium<br>• c4.large<br>• c5.9xlarge, c5.4xlarge, c5.2xlarge, c5.xlarge, c5.large |
| Cisco IOS XE 17.11.x | • t3.medium<br>• c4.large<br>• c5.9xlarge, c5.4xlarge, c5.2xlarge, c5.xlarge, c5.large |
| Cisco IOS XE 17.10.x | • t3.medium<br>• c4.large<br>• c5.9xlarge, c5.4xlarge, c5.2xlarge, c5.xlarge, c5.large |
| Cisco IOS XE 17.9.x | • t3.medium<br>• c4.large<br>• c5.9xlarge, c5.4xlarge, c5.2xlarge, c5.xlarge, c5.large |
| Cisco IOS XE 17.8.x | • t3.medium<br>• c4.large<br>• c5.9xlarge, c5.4xlarge, c5.2xlarge, c5.xlarge, c5.large |

| Cisco IOS XE release | Supported instance types (BYOL only) |
| --- | --- |
| Cisco IOS XE 17.7.x | • t3.medium<br>• c4.large<br>• c5.9xlarge, c5.4xlarge, c5.2xlarge, c5.xlarge, c5.large |
| Cisco IOS XE 17.6.x | • t3.medium<br>• c4.large<br>• c5.9xlarge, c5.4xlarge, c5.2xlarge, c5.xlarge, c5.large |

# 4 Install Cisco Catalyst 8000V in VMware ESXi Environment

**Topics:**


- Install Cisco Catalyst 8000V in VMware ESXi Environment
- Deploy the OVA file Using vSphere
- Deploy the VM Using COT
- Create the VM Using the iso File
- Performance Enhancements on ESXi Configurations


Provides detailed instructions for deploying the Cisco Catalyst 8000V on VMware ESXi, covering various installation methods such as using OVA templates or manual setup via ISO/BIN images, along with the necessary virtual machine hardware and network configuration steps.

# Install Cisco Catalyst 8000V in VMware ESXi Environment

Provides information about how to deploy Cisco Catalyst 8000V in ESXi environments, the requirements for a successful deployment, and the deployment methods that are supported.

VMware ESXi is a hypervisor that allows the basic creation and management of virtual machines, and is one of the hypervisors supported by Cisco Catalyst 8000V. This hypervisor runs on x86 hardware with virtualization extensions, and you can use it to run several VMs simultaneously.

This chapter provides information about how to deploy Cisco Catalyst 8000V in ESXi environments, the requirements for a successful deployment, and the deployment methods that are supported.

Before you proceed, familiarize yourself with VMware vSphere information by referring to the official VMWare product documentation.

**Note**

Oversubscription of host resources can reduce performance, and your instance could become unstable. We recommend following the guidelines and best practices for your host hypervisor.

## Supported features and operations

Supported features and operations for Cisco Catalyst 8000V deployed in ESXi environment

VMware supports various features and operations that allow you to manage your virtual applications and perform operations such as cloning, migration, shutdown and resume.

Some of these operations save the current runtime state of the VM. When you restart the VM, the system restores this state. If the runtime state includes traffic-related state, on resumption or replaying the runtime state, additional errors, statistics, or messages are displayed on the user console. If the saved state is only configuration driven, you can use these features and operations without any issues.

See the tables to view all the supported features and operations for Cisco Catalyst 8000V instances deployed in ESXi environments.

**Table 3: Supported VMware features for vCenter Server**

| Supported entities | Description |
|---|---|
| Cloning | Enables cloning a virtual machine or template, or cloning a virtual machine to a template. |
| Migration | The entire state of the virtual machine as well as its configuration file, if necessary, is moved to the new host even while the data storage remains in the same location on shared storage. |
| vMotion | Enables moving the VM from one physical server to another while the VM remains active. |
| Template | Uses templates to create new virtual machines by cloning the template as a virtual machine. |

**Table 4: Supported VMware operations (for vCenter server and vSphere client)**

| Supported entities | Description |
|---|---|
| Power on | Powers on the virtual machine and boots the guest operating system if the guest operating system is installed. |
| Power off | Stops the virtual machine until it is powered back. The power off option performs a “hard” power off, which is analogous to pulling the power cable on a physical machine and always works. |
| Shutdown | Shut Down, or soft power off uses VMware Tools to perform a graceful shutdown of a guest operating system. In certain situations, such as when VMware Tools is not installed or the guest operating system is unresponsive, the shut down might not complete, and you must use the power off option instead |
| Suspend | Suspends the virtual machine. |
| Reset or restart | Stops the virtual machine and restarts it. |
| OVF creation | An OVF package consisting of several files in a directory captures the state of a virtual machine including disk files that are stored in a compressed format. You can export an OVF package to your local computer. |
| OVA creation | You can create a single OVA package file from the OVF package/template. The OVA can then be distributed more easily; for example, it may be downloaded from a website or moved via a USB key. |

**Table 5: Supported networking features**

| Supported entities | Description |
|---|---|
| Custom MAC address | You can set up the MAC address manually for a virtual network adapter from both vCenter Server and vSphere Client. |
| Distributed VSwitch | A vSphere distributed switch on a vCenter Server data center can manage networking traffic for all associated hosts on the data center. This feature is available only from vCenter Server. |
| Distributed resources scheduler | Provides automatic load balancing across hosts. |
| NIC load balancing | Load balancing and failover policies allow you to determine how network traffic is distributed between adapters and how to reroute traffic if an adapter fails.This feature is available in both vCenter Server and vSphere Client. |

| Supported entities | Description |
|---|---|
| NIC teaming | This feature is available in both vCenter Server and vSphere Client and allows you to set up an environment where each virtual switch connects to two uplink adapters that form a NIC team. The NIC teams can then either share the load of traffic between physical and virtual networks among some or all of its members, or provide passive failover in the event of a hardware failure or a network outage.<br><br>**Note**<br><br>NIC Teaming can cause a large number of ARP packets to flood the Cisco Catalyst 8000V and overload the CPU. To avoid this situation, reduce the number of ARP packets and implement NIC Teaming as Active-Standby rather than Active-Active. |
| vSwitch | A vSwitch is a virtualized version of a Layer 2 physical switch. A vSwitch can route traffic internally between virtual machines and link to external networks. You can use vSwitches to combine the bandwidth of multiple network adapters and balance communications traffic among them. You can also configure a vSwitch to handle a physical NIC fail-over. This feature is available in both vCenter Server and vSphere Client. |

**Table 6: High availability features**

| Supported entities | Description |
|---|---|
| VM-level high availability | To monitor operating system failures, VM-Level High Availability monitors heartbeat information in the VMware High Availability cluster. Failures are detected when no heartbeat is received from a given virtual machine within a user-specified time interval. VM-Level High Availability is enabled by creating a resource pool of VMs using VMware vCenter Server. |
| Host-level high availability | To monitor physical servers, an agent on each server maintains a heartbeat with the other servers in the resource pool such that a loss of heartbeat automatically initiates the restart of all affected virtual machines on other servers in the resource pool. Host-Level High Availability is enabled by creating a resource pool of servers or hosts, and enabling high availability in vSphere. |
| Fault tolerance | Using high availability, fault tolerance is enabled on the ESXi host. When you enable fault tolerance on the VM running the Cisco Catalyst 8000V instance, a secondary VM on another host in the cluster is created. If the primary host goes down, then the VM on the secondary host will take over as the primary VM for the Cisco Catalyst 8000V. |

**Note**

Cisco IOS-based High Availability is not supported by the Cisco Catalyst 8000V instance. High Availability is supported on the VM host only.

**Table 7: Storage options (for vCenter Server and vSphere Web Client)**

| Supported entities | Description |
|---|---|
| Local storage | Local storage is in the internal hard disks located inside your ESXi host. Local storage devices do not support sharing across multiple hosts. A datastore on a local storage device can be accessed by only one host. |
| External storage target | You can deploy the Cisco Catalyst 8000V instance on external storage, such as a Storage Area Network (SAN). |
| Mount or pass through USB storage | You can connect USB sticks to the Cisco Catalyst 8000V instance and use them as storage devices. In ESXi, you need to add a USB controller and then assign the disk devices to the Cisco Catalyst 8000V instance.<br>Cisco Catalyst 8000V supports USB disk hot-plug. However, you can use only two USB disk hot-plug devices at a time.<br>USB hub is not supported. |

## Installation requirements

This section specifies the requirements for you to install Cisco Catalyst 8000V in ESXi environment.

This section specifies the requirements for you to install Cisco Catalyst 8000V in ESXi environment. These requirements have been fully tested and meet the performance benchmarks.

**ESXi hypervisor requirements**

| Cisco IOS XE release | vSphere Web Client | vCenter Server |
|---|---|---|
| Cisco IOS XE 26.1 release | VMware vSphere Web Client versions 8.0 and 7.0 | VMware ESXi 8.0 and ESXi 7.0 |
| Cisco IOS XE 17.18.x releases<br>Cisco IOS XE 17.16.x releases<br>Cisco IOS XE 17.15.x releases<br>Cisco IOS XE 17.14.x releases<br>Cisco IOS XE 17.13.x releases<br>Cisco IOS XE 17.12.x releases | VMware vSphere Web Client versions 8.0 and 7.0 | VMware ESXi 8.0 and ESXi 7.0 |

| Cisco IOS XE release | vSphere Web Client | vCenter Server |
|---|---|---|
| Cisco IOS XE 17.11.x releases<br>Cisco IOS XE 17.10.x releases<br>Cisco IOS XE 17.9.x releases<br>Cisco IOS XE 17.8.x releases<br>Cisco IOS XE 17.7.x releases<br>Cisco IOS XE 17.6.x releases | VMware vSphere Web Client versions 7.0 and 6.7 | VMware ESXi 7.0 and ESXi 6.7 |
| Cisco IOS XE 17.5.x releases | VMware vSphere Web Client versions 6.7 and 6.5 | VMware ESXi 6.7 and ESXi 6.5 |
| Cisco IOS XE 17.4.x releases | VMware vSphere Web Client versions 6.7 and 6.5 | VMware ESXi 6.7 and ESXi 6.5 |

**Note**

Do not use a standalone vSphere client to manage the ESXi server. Starting ESXi 6.0, it is no longer possible to directly deploy Cisco Catalyst 8000V in ESXi in the case of an ova deployment. You must have a vMware vCenter server and a vSphere client to deploy a .ova file.

### vCPU requirements

These are the supported vCPU configurations for the installation.

- 1 vCPU: requires a minimum of 4 GB RAM allocation
- 2 vCPUs: requires a minimum of 4 GB RAM allocation
- 4 vCPUs: requires a minimum of 4 GB RAM allocation
- 8 vCPUs: requires a minimum of 4 GB RAM allocation
- 16 vCPUS: requires a minimum of 8 GB RAM allocation (supported from Cisco IOS XE 17.11.1a)

**Note**

The required vCPU configuration depends on the throughput license and technology package installed. For more information, see the data sheet for your release.

### vNIC requirements

The Virtual Network Interface Cards (VNICs) listed in this section are supported for the ESXi installation. A maximum of 8 vNICs is supported.

- ConnectX-6 - Supported from Cisco IOS XE 17.18.1a
- ixgbe - Supported from Cisco IOS XE 17.10.1
- ConnectX-5VF - Supported from Cisco IOS XE 17.9.1

- iavf - Supported from Cisco IOS XE 17.9.1
- i40eVF - Supported from Cisco IOS XE 17.4.1 to Cisco IOS XE 17.8.x
- VMXNET3 - Supported from Cisco IOS XE 17.4.1
- iXGBeVF - Supported from Cisco IOS XE 17.4.1

> **Note**
>
> The supported version of the NIC driver and the firmware version are the default versions that are included with the hypervisor package.

#### Other requirements

- VMware vCenter - installation tool
- VMware vSwitch - standard or distributed vSwitches are supported
- Hard Drive - only a single hard disk drive is supported. Multiple hard disk drives on a VM are not supported
- Virtual Disk - both 16 GB and 8 GB virtual disks are supported
- Virtual CPU core - one virtual CPU core is required. This needs a 64-bit processor with Virtualization Technology (VT) enabled in the BIOS setup of the host machine.
- Virtual hard disk space - a minimum size of 8 GB is required
- A default video controller, an SCSI controller set, and an installed virtual CD/DVD drive are also required for this installation.

#### What to do next

Familiarize yourself with the secure boot configuration before proceeding with the installation. For more information, see VNF Secure Boot on page 140.

## Restrictions for deploying in ESXi environment

Restrictions for deploying in ESXi environment

The VMware features and operations listed here are not supported in Cisco Catalyst 8000V. Using or performing these unsupported versions might result in dropped packets, dropped connections, and other error statistics.

- Distributed Resource Scheduling (DRS)
- Fault tolerance
- Resume
- Snapshot
- Suspend

# Deploy the OVA file Using vSphere

Provides information about deploying the OVA file using vSphere, guidelines for the deployment, and configuring the bootstrap settings for the deployment.

The VMware vSphere Web Client is a web application that runs on x86 hardware with virtualization extensions and provides access to the VMware vCenter Server.

You can use the VMware vSphere Client or the vSphere Web Client to deploy the .ova installation file. First, download the .ova file from the Cisco Software Download page and then use this file for deployment.

Use the VMware vSphere Web Client software to:

- Create, configure, and manage VMs on the vCenter Server.
- Start or stop a Cisco Catalyst 8000V instance.

### Guidelines when deploying the OVA file

These restrictions apply when you deploy the .ova file by using vSphere.

- If you change the virtual CPU configuration, you must reboot the Cisco Catalyst 8000V instance. However, changing the RAM allocation does not require a reboot.
- When you deploy the OVA, the VM requires two virtual CD/DVD drives, one for the OVF environment file and one for the .iso file.
- You can use the OVA package to select the virtual CPU configuration.

## Deploy the OVA to the VM

Perform these steps to deploy a .ova file to the VM.

### Procedure

1. Log in to the VMware vSphere client and choose **File** > **Deploy OVF Template**.
2. In the OVA Wizard, point the source to the Cisco Catalyst 8000V OVA to be deployed and click **Next** .
3. The system displays the OVF Template Details, which contain information about the OVA. Review this information, and click **Next**.
4. In the **Name and Inventory Location** field, specify the name for the VM and click **Next** .
5. From the **Deployment Configuration** drop-down list, choose the desired hardware configuration profile and click **Next**.
6. In the **Storage** field, choose the Datastore to use for the VM and click **Next**.
7. In the **Disk Format** field, choose one of these disk format options and click **Next**.
   - Thick Provision Lazy Zeroed
   - Thick Provision Eager Zeroed

> **Note**
>
> The Thin Provision option is not supported. The Thick Provision Eager Zeroed option takes longer to install but provides better performance.

8. In the **Network Mapping** area, allocate one or more virtual network interface card (vNIC) on the destination network using the drop-down list.

   Select the network mappings for the 3 default vNICs created during the OVA deployment. You can choose which vNIC will map to the router's management interface when setting the bootstrap properties.

9. Select the vNIC to connect at **Power On** and click **Next**.

   When the Cisco Catalyst 8000V installation using the OVA is complete, two additional vNICS are allocated. Cisco Catalyst 8000V supports up to ten vNICs. You must manually create additional vNICs on the VM.

#### What to do next

Proceed to configure the bootstrap properties for the VM. After you change the bootstrap properties, the system treats the VM as new. When the VM restarts, it removes all pre-existing network configuration.

## Configure the bootstrap properties

Perform this task to configure the bootstrap properties for the OVA file deployment and to complete the deployment.

### Before you begin

Perform the Deploy the OVA to the VM on page 38 task.

Perform this task to configure the bootstrap properties for the OVA file deployment and to complete the deployment. The bootstrap properties are optional when you create the VM. You can set these properties to provision the VM before starting it.

#### Procedure

1. In the VMWare vSphere client, navigate to the VM properties page.
2. Configure these bootstrap properties for the VM.

**Table 8: OVA Bootstrap Properties**

| Property | Description |
|---|---|
| Bootstrap Properties | |
| Console | Configures the console mode. Possible values are **virtual** and **serial**. |
| Login Username | Sets the login username for the router. |
| Login Password | Sets the login password for the router. |

| Property | Description |
|---|---|
| Management Interface | Designates the management interface for the Cisco Catalyst 8000V instance. The format must be GigabitEthernetx or GigabitEthernetx.xxx.<br><br>**Note**<br>The GigabitEthernet0 interface is no longer supported. |
| Management vLAN | Configures the dot1Q VLAN interface. The management interface must be configured using the GigabitEthernetx.xxx format. |
| Management Interface IPv4 Address/Mask | Configures the IPv4 address and subnet mask for the management interface. |
| Management IPv4 Default Gateway | Configures the IPv4 management default gateway address. If using DHCP, enter “dhcp” in the field. |
| Management IPv4 Gateway | Configures the IPv4 management default gateway address. If using DHCP, enter “dhcp” in the field. |
| Management IPv4 Network | Configures the IPv4 network (such as “192.168.2.0/24” or “192.168.2.0 255.255.255.0”) that the management gateway should route to. If a default route (0.0.0.0/0) is desired, you can leave this field blank. |
| PNSC IPv4 Address | Configures the IP address of the Cisco Prime Network Services Controller.<br><br>This setting is used if you plan to remotely manage the Cisco Catalyst 8000V instance using the Cisco Prime Network Services Controller. |
| Router name | Configures the hostname of the router. |
| Resource Template | Configures the Resource Template.<br><br>Possible values: default, service_plane_medium, service_plane_heavy |
| Features | |
| Enable SCP Server | Enables the IOS SCP feature. |
| Enable SSH Login and Disable Telnet Login | Enables remote login using SSH and disables remote login via Telnet. Requires that the login username and password are set. |
| Additional Configuration Properties | |
| Enable Password | Configures the password for privileged (enable) access. |
| Domain Name | Configures the network domain name. |

| Property | Description |
|---|---|
| License Boot Level | Configures the license technology level that is available when the Cisco Catalyst 8000V instance boots. The available license levels are:<br>• network-essentials<br>• network-advantage<br>• network-premier<br><br>**Note**<br>For details on Cisco DNA licenses, see Cisco DNA Software for SD-WAN and Routing. |

**Note**

After you make any change to the bootstrap properties, the system assumes that you are starting with a fresh VM. So when the VM restarts, all pre-existing networking configuration is removed.

You can also configure advanced properties after the router boots.

3. After you configure the router properties, click **Next**. The system displays the **Ready to Complete** page with the settings to be used when the OVA is deployed.
4. Select **Power On After Deployment** to automatically power on the VM.
5. Click **Finish** to deploy the OVA.

   The OVA deploys the .iso file. If you select **Power on After Deployment**, the VM is automatically powered on. After the VM is powered on, the Cisco Catalyst 8000V instance begins the installation and boot process. If a bootstrap configuration file is included in the OVA, the router configuration is automatically enabled.

---

**What to do next**

To learn about the next steps, see Configure Console Access on page 151.

## Deploy the VM Using COT

Specifies the steps to deploy the Cisco Catalyst 8000V VM by using the Common OVF Tool.

To deploy the Cisco Catalyst 8000V VM, use the **cotdeploy...esxi** command. Note that the command description provided in this task offers a general guidance. The exact steps may vary depending on your VMware environment and setup.

**Procedure**

Run the **cotdeploy...esxi** command to deploy the Cisco Catalyst 8000V. The script options are described at: http://cot.readthedocs.io/en/latest/usage_deploy_esxi.html.

> **Note**
>
> The default values may vary depending on the Cisco Catalyst 8000V version.

## Edit the basic properties of the VM

Perform these steps to edit the basic properties of the VM using COT.

Perform these steps to edit the basic properties of the Cisco Catalyst 8000V VM using COT.

### Before you begin

Download and install the COT libraries and script. For instructions, refer to the GitHub site at http://cot.readthedocs.io/en/latest/installation.html.

**Procedure**

1. To edit the basic properties of the OVA, use the **cot edit-properties** command.

   **Example**

   **cot edit-properties**

   **-p** *key1=value1*, **--properties** *key1=value1*

   This command sets properties using key value pairs. For Example, **-p "login-username=cisco"** sets the login username using a key value pair.

   Here, **-o** *output* specifies the name or the path to a new OVA package if you are creating a new OVA instead of updating the existing OVA.

2. Add additional arguments to the **cot edit-properties** command, if required.

This table specifies the **cot edit-properties**arguments that can be used to edit the basic properties of the VM.

| Script step | Description |
|---|---|
| `cot edit propertie s c8000v-universalk9.ova` | Edits the basic environment properties of the OVA file. |
| `-p "login-username=cisco"` | Sets the bootstrap login username. |
| `-p "login-password=cisco"` | Sets the bootstrap login password. |
| `-o "c8000v-universalk9-customized.ova"` | Saves a modified OVA that contains configuration commands from the text file. |

For more information on the **cot edit-properties** command, see: http://cot.readthedocs.io/en/latest/usage_edit_properties.html.

```
cot edit-properties c8000v-universalk9.ova
-p "login-username=cisco"

-p "login-password=cisco"
-o c8000v-universalk9-customized.ova
\# save modifications to a new OVA
cot info c8000v-universalk9-customized.ova
  # verify the new values of properties in the OVA
(...)
Properties:
  <config-version>                                    "1.0"
  Router Name                                         ""
  Login Username                                      "cisco"
  Login Password                                      "cisco"
  Management Interface                                "GigabitEthernet1"
  Management VLAN                                     ""
  Management Interface IPv4 Address/Mask              ""
```

## Edit the custom properties of the VM

You can add custom properties to your Cisco Catalyst 8000V instance based on the Cisco IOS XE CLI commands using the vSphere GUI. You can add these properties either before or after you boot the Cisco Catalyst 8000V instance. If you set these custom properties after you boot the Cisco Catalyst 8000V instance, you must reload the router or power-cycle the VM for the properties settings to take effect.

To edit the vApp options to add the custom Cisco Catalyst 8000V properties, perform these steps.

**Procedure**

1. In the vSphere GUI, choose the **Options** tab.
2. Choose **vApp Options** > **Advanced**.

3. In the **Advanced Property Configuration** window, click **Properties**.
4. Click **New** to add a property.
5. In the **Edit Property Settings** window, enter the required information to create a new custom property based on the appropriate Cisco IOS XE CLI command.

   **Note**

   Before adding a custom property, make sure that the Cisco IOS XE command upon which it is based is supported for your Cisco Catalyst 8000V version.

   a) Enter the label. This is a descriptive string for the property.
   b) Enter the `com.cisco.c8000v` option in the **Class ID** field.
   c) Assign the property an ID of "ios-config-xxxx." Here, xxxx is a sequence number from 0001 to 9999 that determines the order in which the custom properties are applied.
   d) Enter a description for the property.
   e) Enter the property type as “string”. This is the only type supported.
   f) Enter the default value as the Cisco IOS XE CLI command the custom property is based on.
6. Click **OK**.
7. In the **Advanced Property Configuration** window, click **OK**.
8. Reboot the Cisco Catalyst 8000V instance.

   You must reboot the router for the new or edited properties to take effect.

## Command references for deploying OVA using COT

Command references for deploying OVA using COT

This section describes the COT commands and scripts that are commonly used for deploying Cisco Catalyst 8000V using COT.

### cot edit-properties

This command is part of COT and lets you configure or modify the properties within the OVA file. Use the **cot edit-properties** command to pre-apply a small number of configuration commands to the OVA.

**cot edit-properties** *ova-filename*

Here, **-o** *output* specifies the name or path to a new OVA package, if you are creating a new OVA instead of updating the existing OVA.

**-c** *config-file* specifies the name of a text file containing IOS XE commands to be added to the OVA.

In this example, a previously created text file, `iosxe_config.txt`, containing the appropriate IOS XE configuration commands, is added to the OVA deployment using the **cot edit-properties** command.

```
$ cat iosxe_config.txt

interface GigabitEthernet1
no shutdown
ip address 192.168.100.10 255.255.255.0
!
ip route 0.0.0.0 0.0.0.0 GigabitEthernet1 192.168.100.1
```

```
$ cot edit-properties c8000v-universalk9.ova \
     -o c8000v-universalk9-customized.ova \
     -c iosxe_config.txt
$ cot info c8000v-universalk9-customized.ova
```

The **cot info** command is used to show the modified OVA.

```
Properties:
  <config-version>              "1.0"
  Router Name                   ""
```

...

```
  Intercloud Tunnel Interface Gateway IPv4 Address   ""
  <ios-config-0001>             "interface GigabitEthernet1"
  <ios-config-0002>             "no shutdown"
  <ios-config-0003>             "ip address 192.168.100.10 255.255.255.0"
  <ios-config-0004>             "ip route 0.0.0.0 0.0.0.0 GigabitEthernet1
192.168.100.1"
```

This table specifies the command and arguments used in the example.

| Script step | Description |
|---|---|
| `cot edit properties c8000v-universalk9.ova` | Edits the custom environment properties of the OVA file. |
| `-o "c8000v-universalk9-customized.ova"` | Creates a new OVA containing configuration commands from the text file. |
| `-c iosxe_config.txt` | The text file that contains IOS XE configuration commands. Each line of configuration in this file results in an entry such as com.cisco.productname.ios-config-xxxx in the XML of the OVA. |

**cot inject-config**

The `cot inject-config` command is used in COT to embed one or more bootstrap configuration files into the OVA package.

Use the **cot inject-config** commandto pre-apply a large set of configuration commands to the OVA, for example, a complete running configuration. This method is efficient for file size and loading time, because it uses plain text for configuration commands rather than XML.

For more information about the **cot inject-config** command, see http://cot.readthedocs.io/en/latest/usage_inject_config.html

cot inject-config ova-filename

Here, **-o** *output* specifies the name or path to a new OVA package if you are creating a new OVA instead of updating the existing OVA.

**-c** *config-file* specifies the name of a text file, such as iosxe_config.txt to be embedded in the OVA.

In this example, the **cot inject-config** command adds Cisco IOS XE commands in text file iosxe_config.txt to the OVA.

```
$ cat iosxe_config.txt
interface GigabitEthernet1
no shutdown
ip address 192.168.100.10 255.255.255.0
!
ip route 0.0.0.0 0.0.0.0 GigabitEthernet1 192.168.100.1
$ cot inject-config c8000v-universalk9.ova \

 -o c8000v-universalk9-customized.ova \
      -c iosxe_config.txt
$ cot info c8000v-universalk9-customized.ova
```

<.. other output snipped for brevity ..>

```
Files and Disks:                        File Size  Capacity Device
                                       --------- --------- --------------------

  c8000v_harddisk.vmdk                  71.50 kB   8.00 GB harddisk @ SCSI 0:0
  bdeo.sh                                 52.42 kB
  README-OVF.txt                           8.53 kB
  README-BDEO.txt                          6.75 kB
  cot.tgz                                116.78 kB
  c8000v-universalk9.iso                484.80 MB           cdrom @ IDE 1:0
  config.iso                              350.00 kB           cdrom @ IDE 1:1
```

This table specifies the **cot inject-config** command and arguments used in the example.

| Script step | Description |
|---|---|
| `cot inject-config c8000v-universalk9.ova` | Edits the custom environment properties of the OVA file. |
| `-o "c8000v-universalk9-customized.ova"` | Specifies the name of the new or modified OVA that contains the configuration commands from the text file. |
| `-c iosxe_config.txt` | The name of the text file that contains the IOS XE configuration commands. |

## Create the VM Using the iso File

Specifies the steps to install the .iso file on the VMware ESXi host manually and create the VM using the vSphere GUI.

Perform this task to install the .iso file on the VMware ESXi host manually and create the VM using the vSphere GUI. This procedure provides general guidance for deploying Cisco Catalyst 8000V, but the steps may vary depending on your VMware environment and setup. The instructions in this procedure are based on VMware ESXi 5.0.

**Procedure**

1. Download the iso file from the Cisco Catalyst 8000V software installation image package and copy it onto the VM Datastore.
2. In the vSphere client, choose **Create a New Virtual Machine**.
3. From **Configuration**, choose the option to create a custom configuration, and click **Next**.
4. In the **Name and Location** field, specify the name for the VM and click **Next**.
5. In the **Storage** field, choose the datastore to use for the VM and click **Next** .
6. From the **Virtual Machine Version** drop-down field,choose **Virtual Machine Version 15** or a higher version that is available and click **Next**.

   **Note**

   Cisco Catalyst 8000V is not compatible with ESXi Server versions prior to 6.5 Update 2.

7. In **Guest Operating System** area, choose **Linux** and the **Other 3.x Linux (64-bit)**options and click **Next**.
8. In the **CPUs** area, configure these settings and click **Next**:
   a) Number of virtual sockets (virtual CPUs)
   b) Number of cores per socket: The number of cores per socket should always be set to 1, regardless of the number of virtual sockets selected. For example, a Cisco Catalyst 8000V with a 4 vCPU configuration should be configured as 4 sockets and 1 core per socket.
9. In the **Memory** field, configure the supported memory size for your **Cisco Catalyst 8000V** release and click **Next**.
10. In the **Network** field, allocate at least three virtual network interface cards (vNICs).
    a) Select the number of vNICs that you want to connect from the drop-down menu.

       **Note**

       The VMware ESXi interface allows you to create 4 vNICs during the initial VM setup. You can add more vNICs after creating the VM and booting Cisco Catalyst 8000V for the first time.

    b) Add the vNICs.

       Select a different network for each vNIC.

       Select the adapter type from the drop-down menu. See the requirements sections in this guide for the supported adapter type in your release.

    c) Select all the vNICs to connect at power-on and click **Next**. You can add vNICs into the VM using vSphere while the Cisco Catalyst 8000V is running. For more information about adding vNICS to an existing VM, see the vSphere documentation.
11. From the **SCSI Controller** drop-down field, choose **VMware Paravirtual** and click **Next**.
12. From the **Select a Disk** drop-down field, choose **Create a New Virtual Disk** .
13. In the **Create a Disk** area, configure these settings and click **Next**:
    a) **Capacity: Disk Size**: The virtual hard disk size required in your release. See the requirements section in this chapter for the supported disk size.
    b) **Disk Provisioning**: Select either **Thick Provision Lazy Zeroed** or **Thick Provision Eager Zeroed**.

> **Note**
>
> The Thin Provision option is not supported. Choosing **Thick Provision Eager Zeroed** takes longer to install, but it provides better performance.

c) **Location**: Store with the Virtual Machine

14. From the **Advanced Options** field, choose **SCSI (0:0)** for the virtual device node.
15. On the **Ready to Complete** window, click the **Edit the Virtual Machine** settings before completion.
16. Select the **Continue** checkbox.
17. In the **Hardware** tab, click **New CD/DVD Drive**.
    a) Choose the **Device Type** that the VM will boot from:

    Choose the **Datastore ISO file** option to boot from the .iso file. Browse to the location of the .iso file on the datastore set in step 1.

    b) In the **Device Status** field, select the **Connect at Power On** checkbox.
    c) Select the **Virtual Device Node CD/DVD** drive on the host that the VM will boot from.
18. In the **Resources** tab, under **CPU**, set the **Resource Allocation** setting to **Unlimited** and click **OK**.
19. Click **Finish**.

The VM is now configured for the Cisco Catalyst 8000V and is ready to boot. The Cisco Catalyst 8000V is booted when the VM is powered on.

> **Note**
>
> To configure the Day 0 settings of a manually installed Cisco Catalyst 8000V, attach a second CD/DVD drive pointing to an ISO that contains the said bootstrap configuration. For further details on the supported bootstrap ISO contents, see Day 0 Configuration on page 106.

> **Note**
>
> To access and configure the Cisco Catalyst 8000V from the serial port on the ESXi host instead of the virtual VGA console, provision the VM to use this setting before powering on the VM and starting the router.

---

**What to do next**

For the next steps, see Booting the Cisco Catalyst 8000V VM and Accessing the Console sections.

## Performance Enhancements on ESXi Configurations

Provides information on how to improve the performance of Cisco Catalyst 8000V running in an ESXi environment by adjusting the settings on the host and the virtual machine.

Improve the performance of Cisco Catalyst 8000V running in an ESXi environment by adjusting these settings on the host and the virtual machine.

- Enable the hypervisor performance settings on the hypervisor.
- Limit the overhead of vSwitch by enabling SR-IOV for supported physical NICs.
- Ensure that the vCPUs of the VM run on the same NUMA node as the physical NICs.
- Set the **VM Latency Sensitivity** setting to **High**.

#### Host configuration settings

To improve VMware ESXi performance, use the host configuration settings described here.

- Choose **High Performance** from the **Power Management** field.
- Disable **Hyperthreading**.
- Enable SR-IOV for supported physical adapters.

#### Virtual machine configuration settings

To improve the performance of the VMware ESXi configuration, configure these settings on the host:

- Ensure that the ESXi version is compatible with your Cisco Catalyst 8000V version.
- Set the **Virtual Hardware: CPU reservation** setting to **Maximum**.
- Reserve all the guest memory in **Virtual Hardware: Memory**.
- Choose the **VMware Paravirtual** option from **Virtual Hardware: SCSI Controller**.
- From the **Virtual Hardware: Network Adapter: Adapter Type** option, choose SR-IOV for supported NICs.
- Configure the **General Guest OS Version** > **VM Options** option to **Other 3.x or later Linux (64-bit)**.
- Configure the **VM Options** option under **Advanced Latency Sensitivity** to **High**.
- In **VM Options** > **Advanced Edit Configuration**, add `numa.nodeAffinity` to the same NUMA node as the SRIOV NIC.

# 5 Install Cisco Catalyst 8000V in KVM Environment

**Topics:**


- Install Cisco Catalyst 8000V in KVM Environment
- Create the VM Using the GUI Tool
- Create the VM Using CLI
- Performance Enhancements in KVM Environment
- Troubleshoot KVM Installation Issues


Provides comprehensive instructions for installing Cisco Catalyst 8000V in KVM environments, including prerequisites, supported Linux distributions, and step-by-step procedures for deploying the software using tools like virt-install and virt-manager.

# Install Cisco Catalyst 8000V in KVM Environment

This chapter provides information on how to install Cisco Catalyst 8000V in KVM environments.

Red Hat Enterprise Linux (RHEL) is based on Kernel-based Virtual Machine (KVM), an open source, full virtualization solution for Linux on x86 hardware that contains virtualization extensions.

This chapter provides information on how to install Cisco Catalyst 8000V in KVM environments.

You can install the Cisco Catalyst 8000V virtual router as a virtual machine on Red Hat KVM virtualization. First, manually create the VM. Then, install the software using either an .iso file or a qcow2 file. Install Cisco Catalyst 8000V in a KVM environment by using the:

- GUI Tool: Download and install the virt-manager RPM package on the KVM server. Virt-manager is a desktop user interface for managing virtual machines. Installation by using the GUI is the recommended installation method.
- Command Line Interface (CLI): Use the CLI to install the Cisco Catalyst 8000V VM.

## Installation requirements

This section lists the supported host Linux operating system versions on which Cisco Catalyst 8000V has been validated.

This section lists the supported host Linux operating system versions on which Cisco Catalyst 8000V has been validated. All other deployment combinations not mentioned in this section will fall under Cisco Catalyst 8000V non-certified support model. Ask your local Cisco Sales representative for more details.

### Supported RHEL versions (Linux KVM based on Red Hat Enterprise Linux)

This table lists the supported RHEL versions for Cisco Catalyst 8000V.

| Cisco IOS XE release | Red Hat Enterprise Linux version |
|---|---|
| Cisco IOS XE 26.1 release | Red Hat Enterprise Linux versions 9.6, and 8.10 |
| Cisco IOS XE 17.18.x releases | Red Hat Enterprise Linux versions 9.6, 9.2, and 8.10 |
| Cisco IOS XE 17.16.x releases | Red Hat Enterprise Linux versions 9.2 and 8.10 |
| Cisco IOS XE 17.15.5 release | Red Hat Enterprise Linux versions 9.6, 9.2, 8.10, and 8.4 |
| Cisco IOS XE 17.15.1 to 17.15.4 releases | Red Hat Enterprise Linux versions 9.2, 8.10, and 8.4 |
| Cisco IOS XE 17.14.x releases | Red Hat Enterprise Linux versions 9.2, 8.4, and 7.7 |
| Cisco IOS XE 17.13.x releases | Red Hat Enterprise Linux versions 9.2, 8.4, and 7.7 |
| Cisco IOS XE 17.12.x releases | Red Hat Enterprise Linux versions 9.2, 8.4, and 7.7 |
| Cisco IOS XE 17.11.x releases | Red Hat Enterprise Linux versions 8.4 and 7.7 |
| Cisco IOS XE 17.10.x releases | Red Hat Enterprise Linux versions 8.4 and 7.7 |
| Cisco IOS XE 17.9.x releases | Red Hat Enterprise Linux versions 8.4 and 7.7 |
| Cisco IOS XE 17.8.x releases | Red Hat Enterprise Linux versions 8.4 and 7.7 |
| Cisco IOS XE 17.7.x releases | Red Hat Enterprise Linux versions 8.4 and 7.7 |
| Cisco IOS XE 17.6.x releases | Red Hat Enterprise Linux versions 7.7 and 7.5 |

| Cisco IOS XE release | Red Hat Enterprise Linux version |
|---|---|
| Cisco IOS XE 17.5.x releases | Red Hat Enterprise Linux versions 7.7 and 7.5 |
| Cisco IOS XE 17.4.x releases | Red Hat Enterprise Linux versions 7.7 and 7.5 |

From Cisco IOS XE 17.12.1 release, Cisco Catalyst 8000V is also supported on Intel Atom® C3000 processor (Denverton) CPU-based servers with Intel x550 NIC on RHEL 8.4 KVM hypervisor. You can operate Cisco Catalyst 8000V on other x86 CPUs with various hypervisor operating system versions. However, support is provided only for the versions listed in the Supported RHEL versions table.

**Supported SUSE Linux® Enterprise Server (SLES) versions**

This table lists the supported SLES versions for Cisco Catalyst 8000V.

| Cisco IOS XE release | SUSE Linux® Enterprise Server version |
|---|---|
| Cisco IOS XE 26.1.1 release | SUSE Linux Enterprise Server version 15 SP7 |
| Cisco IOS XE 17.18.x releases | SUSE Linux Enterprise Server version 15 SP5 |
| Cisco IOS XE 17.16.x releases | SUSE Linux Enterprise Server version 15 SP5 |
| Cisco IOS XE 17.15.x releases | SUSE Linux Enterprise Server version 15 SP5 |
| Cisco IOS XE 17.14.x releases | SUSE Linux Enterprise Server version 15 SP3 |
| Cisco IOS XE 17.13.x releases | SUSE Linux Enterprise Server version 15 SP3 |
| Cisco IOS XE 17.12.x releases | SUSE Linux Enterprise Server version 15 SP3 |
| Cisco IOS XE 17.11.x releases | SUSE Linux Enterprise Server version 15 SP3 |
| Cisco IOS XE 17.10.x releases | SUSE Linux Enterprise Server version 15 SP3 |
| Cisco IOS XE 17.9.x releases | SUSE Linux Enterprise Server version 15 SP3 |
| Cisco IOS XE 17.6.3 releases | SUSE Linux Enterprise Server version 15 SP3 |

**Supported Ubuntu versions**

This table lists the supported Ubuntu versions for Cisco Catalyst 8000V.

| Supported IOS XE release | Ubuntu version |
|---|---|
| Cisco IOS XE 17.18.1a and later | Ubuntu 22.04 LTS |

**Supported vNICs**

This table lists the supported vNIC versions for Cisco Catalyst 8000V.

| Supported IOS XE release | Supported VNIC |
|---|---|
| Cisco IOS XE Release 17.18.1a and later | ConnectX-6 |
| Cisco IOS XE Release 17.10.1 and later | Ixgbe |

| Supported IOS XE release | Supported VNIC |
|---|---|
| Cisco IOS XE Release 17.9.1 and later | iavf |
| Cisco IOS XE Release 17.9.1 and later | ConnectX-5VF |
| Cisco IOS XE Release 17.4.1 to Cisco IOS XE 17.8.x releases | i40evf |
| Cisco IOS XE Release 17.4.1 and later | Virtio |
| Cisco IOS XE Release 17.4.1 and later | ixgbevf |

**Supported vCPU configurations**

This table lists the supported vCPU configurations for installing Cisco Catalyst 8000V in KVM environments.

| Supported IOS XE release | Supported vCPU configuration |
|---|---|
| Cisco IOS XE 17.4.1 and later | 1 vCPU: requires minimum 4 GB RAM allocation |
| Cisco IOS XE 17.4.1 and later | 2 vCPUs: requires minimum 4 GB RAM allocation |
| Cisco IOS XE 17.4.1 and later | 4 vCPUs: requires minimum 4 GB RAM allocation |
| Cisco IOS XE 17.4.1 and later | 8 vCPUs: requires minimum 8 GB RAM allocation |
| Cisco IOS XE 17.11.1a and later | 16 vCPUs: requires minimum 8 GB RAM allocation |

## Create the VM Using the GUI Tool

Explains how to create the Cisco Catalyst 8000V VM in KVM environments using the GUI tool.

You can install the Cisco Catalyst 8000V virtual router as a virtual machine on Red Hat KVM virtualization by using the GUI tool. While you can use the CLI to perform this installation, Cisco recommends using the GUI tool.

### Summary

Download these key components that you require for installing Cisco Catalyst 8000V in KVM environments by using the GUI tool.

- Download and install the virt-manager RPM package on the KVM server.
- Download either the .qcow2 image or the .iso image from the Cisco Software Download page. Copy the file onto a local device or a network device.

### Workflow

Perform these tasks to install Cisco Catalyst 8000V in KVM environments.

1. Create the virtual machine and configure the virtual machine settings.
2. Add a serial console.
3. Add a host USB device.
4. Customize the virtual machine configuration.

   See the respective sections on this page for the detailed steps.

## Create the VM

Perform these steps if you want to create a new VM in a KVM environment by using the GUI tool.

### Procedure

1. Launch the virt-manager GUI.
2. Click **Create a New Virtual Machine**.
3. Select the Cisco Catalyst 8000V qcow2 or iso file location after performing one of these two actions:
   - If you have downloaded the .qcow2 file, select **Import Existing Disk Image**.
   - If you have downloaded the .iso file, select **Local Install Media (ISO Image or CDROM)**.
4. Configure the memory and the CPU parameters.
5. Configure the virtual machine storage settings.
6. If you want to add additional hardware before creating the VM, click **Customize configuration before install**. The system displays the **Add Hardware** button. Click this button to add various hardware options, such as additional disks or a serial port interface.
7. Add a serial console.

   This is an optional step. To view the detailed steps to add a serial console, see Add a serial console on page 55.
8. Customize your virtual machine configuration.

   This is an optional step. To view the detailed steps to customize your configuration, see Customize the VM configuration on page 56.
9. Click **Finish**.

### What to do next

Access the Cisco Catalyst 8000V console by performing one of these actions:

- If you are using a virtual console, double-click the VM instance to access the VM console.
- If you are using a serial console, see Booting the Cisco Catalyst 8000V and Accessing the Console .

## Add a serial console

Perform these steps to add a serial console to the hardware and configure the kernel console.

### Procedure

1. In the virt-manager GUI, click **Add Hardware**.
2. From the menu, choose the **Serial** option.
3. From the **Device Type** drop-down field, select **TCP net console (tcp)**.
4. Specify the port number, and select the **Use Telnet** checkbox.
5. Click **Finish**.
6. After adding all necessary hardware, click **Begin Installation**.

## Add a host USB device

To add the USB host device using virt-manager, complete these steps.

If you use a Cisco IOS XE 17.9.x version, a 17.12.x version, or a later version, Cisco Catalyst 8000V on KVM supports USB device pass-through which allows a USB device connected to the host machine to be passed directly to your VM. To add a USB host device, you must modify the VM configuration by using tools like virt-manager, virsh, or by directly editing the XML configuration of the VM.

**Procedure**

1. In the virt-manager GUI, click **Add Hardware**.
2. Select the USB host device.
3. From the **USB Device** drop-down field, choose the USB device that you want to attach to the Cisco Catalyst 8000V.
4. Click **Finish**.

> **Note**
>
> Cisco Catalyst 8000V supports USB disk hot-plug and allows up to two USB disk hot-plug devices to be attached.. When you add the devices, ensure you add one device at a time, by using the virt-manager.

## Customize the VM configuration

### Before you begin

Perform the Create the VM on page 55 workflow by using a .qcow2 or an .iso image. If you want to customize your configuration before creating the instance, follow this procedure. Ensure you perform this task before you apply the Day 0 configuration to your device.

**Procedure**

1. In the creating VM workflow, when you choose the **Customize configuration before install** option, the **Add Hardware** button is displayed. Click this button.
2. Choose the **Storage** option.
3. Select the **Select Managed Or Other Existing Storage** check box.
4. Click **Browse** and navigate to the location where you have the **c8000v_config.iso** file.

   This step is applicable only when you want to add a Day 0 or bootstrap configuration.
5. From the **Device-type** drop-down field, choose **IDE CDROM**.
6. Click **Finish**.
7. To adjust the VM boot options in the virtual manager, including the boot device order, select **IDE CDROM** or the **c8000v_config.iso**option. Use the Up Arrow to move the device to the first position in the boot order.
8. Click **Apply** to apply the changes you made to the boot order.
   The image disk device option is pushed down as the second option in the boot device order.
9. After adding all the necessary hardware, click **Begin Installation** to start the new VM deployment.

**What to do next**

Perform the Day 0 configuration for your device. To learn how to do this, see Day 0 Configuration on page 106.

## Create the VM Using CLI

Specifies the steps to create the Cisco Catalyst 8000V VM using command line interface.

### Before you begin

- Download and install the virt-install RPM package on the KVM server.
- Download the .qcow2 image from the Cisco Catalyst 8000V software installation image package. Then copy the file to a local device or network device.

**Procedure**

1. To create the VM for a .qcow2 image, use the **virt-install** command.

   Creates a new VM instance and starts the instance.

   **Example**

```
                        virt-install                      \
                        --connect=qemu:///system   \
                        --name=my_c8kv_vm           \
                        --os-type=linux            \
                        --os-variant=rhel4         \
                        --arch=x86_64              \
                        --cpu host                 \
                        --vcpus=1,sockets=1,cores=1,threads=1   \
                        --hvm                      \
                        --ram=4096                 \
                        --import                   \
                     --disk path=<path_to_c8000v_qcow2>,bus=ide,format=qcow2
  \
                        --network bridge=virbr0,model=virtio
       \
                        --noreboot
```

2. To create the VM, for a .iso image, perform these steps:
   a) Run the **qemu-img** command.

      Creates an 8G disk image in the **.qcow2** format.

      **Example**

```
                                qemu-img create -f qcow2 c8000v_disk.qcow2
 8G
```

   b) Run the **virt-install** command.

      Creates a new VM instance. Cisco Catalyst 8000V installs the image onto the specified disk file after this command is executed. This command requires the correct permissions for the VM to be created.

**Example**

```
                                virt-install                \
                                --connect=qemu:///system  \
                                --name=my_c8000v_vm          \
                                --description "Test VM"   \
                                --os-type=linux           \
                                --os-variant=rhel4        \
                                --arch=x86_64             \
                                --cpu host                \
                                --vcpus=1,sockets=1,cores=1,threads=1
\
                                --hvm                     \
                                --ram=4096                \
                                --cdrom=<path_to_c8000v_iso>          \
                                --disk
path=c8000v_disk.qcow2,bus=virtio,size=8,sparse=false,cache=none,format=qcow2
    \
                                --network bridge=virbr0,model=virtio
\
                                --noreboot
```

This sample configuration creates a 1 vCPU VM with 4G of RAM having one network interface and one serial port.

3. To provide the Day 0 configuration through the `c8000v_config.iso` disk image, add an additional parameter to the **virt-install** command.

   **Example**

   ```
   virt-install --disk path=/my/path/c8000v_config.iso,device=cdrom,bus=ide
   ```

   For more information, see Day 0 Configuration on page 106.

4. After the installation is complete, the Cisco Catalyst 8000V VM shuts down. Run the **virshstart** command to start the VM.

   **Note**

   Due to this issue specific to RHEL, when you launch Cisco Catalyst 8000V in a RHEL environment by using the **virt-install** command, set the host mode based on the RHEL version.

   - For Red Hat Enterprise Linux 6, use `--cpu host .`
   - For Red Hat Enterprise Linux 7, use `--cpu host-model .`

## Performance Enhancements in KVM Environment

Provides all the settings and configurations you need to enhance the performance for Cisco Catalyst 8000V running in KVM environments.

Increase the performance of a Cisco Catalyst 8000V instance running in a KVM environment by modifying certain settings on the KVM host. These settings are independent of the IOS XE configuration settings on the Cisco Catalyst 8000V instance.

- These settings might affect the number of VMs you can instantiate on a server.
- Tuning steps are most impactful for a small number of VMs that you instantiate on a host.

To improve the KVM configuration performance, Cisco recommends that you do the actions listed in this table.

| Action | Sample configuration (if applicable) |
|---|---|
| Enable vCPU pinning | `<cputune>`<br>`<vcpupin vcpu='0' cpuset='3'/>`<br>`<emulatorpin cpuset='3'/>`<br>`</cputune>` |
| Enable emulator pinning | `<cputune>`<br>`<vcpupin vcpu='0' cpuset='3'/>`<br>`<emulatorpin cpuset='3'/>`<br>`</cputune>` |
| Enable numa tuning.<br>Ensure that all vCPUs are pinned to the physical cores on the same socket. | `<numatune>`<br>`<memory mode='strict' nodeset='0'/>`<br>`</numatune>` |
| Set hugepage memory backing | `<currentMemory unit='KiB'>4194304</currentMemory>`<br><br>`<memoryBacking>`<br>`<hugepages>`<br>`<page size='1048576' unit='KiB' nodeset='0'/>`<br><br>`</hugepages>`<br>`<nosharepages/>`<br>`</memoryBacking>` |
| Use virtio instead of IDE | `<devices>`<br>`<emulator>/usr/libexec/qemu-kvm</emulator>`<br>`<disk type='file' device='disk'>`<br>`<driver name='qemu' type='qcow2'/>`<br>`<source file='/var/lib/libvirt/images/rhel7.0.qcow2'/>`<br>`<backingStore/>`<br>`<target dev='vda' bus='virtio'/>`<br>`<boot order='1'/>`<br>`<address type='pci' domain='0x0000' bus='0x00' slot='0x05' function='0x0'/>`<br>`</disk>` |
| Use graphics VNC instead of SPICE. | `<graphics type='vnc' port='5900' autoport='yes' listen='127.0.0.1' keymap='en-us'>`<br>`<listen type='address' address='127.0.0.1'/>`<br><br>`</graphics>` |
| Remove unused devices such as USB, tablet devices | |

| Action | Sample configuration (if applicable) |
|---|---|
| Disable memballoon | `<memballon model='none'>` |

### CPU pinning

Improve the performance of KVM environments by using the KVM CPU Affinity option to assign a virtual machine to a specific processor. To use this option, configure CPU pinning on the KVM host.

In the KVM host environment, use these commands:

| Command | Action |
|---|---|
| **virsh nodeinfo** | Verifies the host topology to find out how many vCPUs are available for pinning by using the following command. |
| **virsh capabilities** | Verifies the available vCPU numbers. |
| **virsh vcpupin<vmname> <vcpu#> <host core#>** | Pins the virtual CPUs to sets of processor cores.<br>Execute this KVM command for each vCPU on your Cisco Catalyst 8000V instance. |

- This example pins the virtual CPU 1 to host core 3.

  **virsh vcpupin c8000v 1 3**

- This example shows the KVM commands needed if you have a Cisco Catalyst 8000V configuration with four vCPUs and the host has eight cores:

  **virsh vcpupin c8000v 0 2**

  **virsh vcpupin c8000v 1 3**

  **virsh vcpupin c8000v 2 4**

  **virsh vcpupin c8000v 3 5**

  The host core number can be any value from 0 to 7. For more information, refer to the official KVM documentation.

**Note**

When you configure CPU pinning, consider the CPU topology of the host server. If you are using a Cisco Catalyst 8000V instance with multiple cores, do not configure CPU pinning across multiple sockets.

### BIOS settings

Optimize the performance of the KVM configuration by applying the recommended BIOS settings listed in this table.

| Configuration | Recommended setting |
|---|---|
| Intel Hyper-Threading Technology | Disabled |
| Number of Enable Cores | ALL |
| Execute Disable | Enabled |

| Configuration | Recommended setting |
|---|---|
| Intel VT | Enabled |
| Intel VT-D | Enabled |
| Intel VT-D coherency support | Enabled |
| Intel VT-D ATS support | Enabled |
| CPU Performance | High throughput |
| Hardware Prefetcher | Disabled |
| Adjacent Cache Line Prefetcher | Disabled |
| DCU Streamer Prefetch | Disable |
| Power Technology | Custom |
| Enhanced Intel Speedstep Technology | Disabled |
| Intel Turbo Boost Technology | Enabled |
| Processor Power State C6 | Disabled |
| Processor Power State C1 Enhanced | Disabled |
| Frequency Poor Override | Enabled |
| P-State Coordination | HW_ALL |
| Energy Performance | Performance |

**Host OS settings**

Cisco recommends that you use hugepages and enable emulator pinning on the host. These actions are recommended on the host.

- Enable IOMMU=pt
- Enable intel_iommu=on
- Enable hugepages
- Use SR-IOV if your system supports it for higher networking performance. Check the SR-IOV limitations your system might have.

In addition to enabling hugepages and emulator pinning, configure: nmi_watchdog=0 elevator=cfq transparent_hugepage=never

> **Note**
>
> If you use Virtio VHOST USER with VPP or OVS-DPDK, you can increase the buffer size to 1024 (rx_queue_size='1024' ) provided the version of your QEMU supports it.

#### IO Settings

You can use SR-IOV to enhance performance. However, this may introduce limitations such as the number of virtual functions (VF), OpenStack limitations for SR-IOV like QoS support, live migration, and security group support.

If you use a modern vSwitch such as fd.io VPP or OVS-DPDK, reserve at least two cores for the VPP worker threads or the OVS-DPDK PMD threads.

Configure these parameters to run the VPP through command line.

| Parameter | Description and notes |
|---|---|
| -cpu host | This parameter causes the VM to inherit the host OS flags. You need libvirt version 0.9.11 or later for this feature to be available in the XML configuration. |
| -m 8192 | Requires 8GB RAM for optimal zero packet drop rates. |
| rombar=0 | Disables PXE boot delays. Set rombar=0 to the end of each device option list or add "<rom bar=off />" to the device xml configuration. |

## KVM halt polling

This section describes the KVM halt polling parameter with respect to Cisco Catalyst 8000V.

`halt_poll_ns` is a KVM module parameter that allows you to alter the behaviour of how idle KVM guest virtual CPUs (vcpus) are handled.

When a virtual CPU in a KVM guest has no threads to run, QEMU traditionally halts the idle CPU. By default, this setting specifies a period of 400 nanoseconds during which a virtual CPU waits and polls before entering a CPU Idle state.

When new work arrives during the polling period, before the vCPU is halted, the vCPU is immediately ready to execute the work. If the vCPU has already entered the idle state when new work arrives, it must be brought out of the idle state before starting the new task. The transition from idle to running state causes additional latency, which can negatively affect latency-sensitive workloads.

With the default kernel parameters, the guest Cisco Catalyst 8000V router CPU consumes all the available resources on the host CPU.

#### Configure KVM halt polling

Configure the halt_poll_ns parameter in KVM to improve performance.

You can configure halt_poll_ns in two ways:

- Large halt_poll_ns: n this case, more CPU time is spent busy-spinning for events that wake the virtual CPU, and fewer ACPI deep sleep cycles occur. This means more power is consumed. However, there are fewer wakeups from deep sleep states. Depending on the configured state, this can cause issues such as cache misses.

- Small halt_poll_ns: In this case, less CPU time is spent busy-spinning for events that wake the CPU, and more ACPI deep sleep cycles occur. This results in lower power consumption, but more wakeups from deep sleep states are required. Increased wakeups can lead to many deep sleep instances. Depending on the configuration, this can cause many cache misses and longer wakeup times.

**Procedure**

1. At run time, run `echo 0 > /sys/module/kvm/parameters/halt_poll_ns`.
2. To load the module, perform this configuration.

```
# rmmod kvm_intel
# rmmod kvm
# modprobe kvm halt_poll_ns=0
# mpdprobe kvm_intel
```

3. When you boot the device, add `kvm.halt_poll_ns=<specify value>` in the parameters section of grub2.

## Troubleshoot KVM Installation Issues

Provides information to troubleshoot common issues while installing Cisco Catalyst 8000V in KVM environments.

See this section to troubleshoot common issues while installing Cisco Catalyst 8000V in KVM environments.

**Issue**

In a KVM environment, when you clone a Cisco Catalyst 8000V virtual machine using the **virt-manager** virtual machine manager, it results in a Cisco Catalyst 8000V virtual machine that you might not be able to boot. The issue occurs because **virt-manager** creates a cloned image that is larger than the original Cisco Catalyst 8000V VM image. The additional bytes, typically in the kilobyte range, prevent the virtual machine from booting.

**Workaround**

Use one of these workarounds to resolve this issue.

- Use the **virt-clone** command to clone the Cisco Catalyst 8000V VM image.
- For a cloned Cisco Catalyst 8000V VM image created by **virt-manager** during the bootup, select the GOLDEN image to boot instead of packages.conf.
- In the Create a new virtual machine window, deselect **Allocate Entire Disk Now** before creating the new Cisco Catalyst 8000V VM. This action allows the cloned Cisco Catalyst 8000V VM image to boot. Do not use this workaround for nested cloning. Apply it only to the first cloned Cisco Catalyst 8000V VM image.

# 6 Install Cisco Catalyst 8000V on Nutanix AHV

**Topics:**


- Install Cisco Catalyst 8000V on Nutanix AHV
- Install the Virtual Machine Using Prism Central
- Install the Virtual Machine Using Prism Element
- Install the VM using command line interface
- Install the VM using REST API
- Edit the VM properties
- Clone a VM

# Install Cisco Catalyst 8000V on Nutanix AHV

Provides an overview about Nutanix AHV hypervisor and specifies the requirements, installation methods, and verification of Cisco Catalyst 8000V deployment on Nutanix AHV.

Nutanix AHV (Acropolis Hypervisor) is a secure, type-1, native hypervisor based on Linux's Kernel-based Virtual Machine (KVM) technology. This hypervisor is built directly into the Nutanix Cloud Infrastructure (NCI) platform. The platform provides compute, storage, and networking capabilities across public and private clouds.

#### Benefits

Cisco Catalyst 8000V running on Nutanix AHV offers these benefits.

- Management of Cisco Catalyst 8000V within this environment is performed through Nutanix Prism, which provides a unified console for virtualization, storage, and networking.
- This hypervisor provides one-click upgrades, integrated full-stack lifecycle management, and an intuitive UI for your VM management.
- Configuration settings, such as virtual networking and high availability, are managed at the cluster level, ensuring consistency and reduced operational risks.

#### Supported Nutanix AHV versions

This table lists the supported Nutanix AHV hypervisor bundled in Nutanix AOS operating system versions.

| Supported IOS XE release | Nutanix AHV version |
|---|---|
| Cisco IOS XE 17.18.1a and later | Nutanix AOS version 7.3 |
| Cisco IOS XE 17.15.5 | Nutanix AOS version 7.3 |

## Installation requirements

Provides the requirements for installing Cisco Catalyst 8000V on Nutanix AHV.

This list provides the requirements for installing Cisco Catalyst 8000V on Nutanix AHV.

- Hypervisor Version: Nutanix AOS 7.3 with AHV
- Network Interface Cards (NICs)
  - SR-IOV support is limited to Nvidia ConnectX-6 NICs.
  - Non-SR-IOV interfaces are treated as Virtio interfaces.
- Machine Type: Q35
- .qcow2 and .iso image formats are supported
- UEFI boot must be enabled
- Ensure access to a REST client, such as Postman, to install Cisco Catalyst 8000V using the REST API.

## Supported configurations

Specifies the supported configurations for deploying Cisco Catalyst 8000V in Nutanix AHV.

This list specifies the supported configurations for this deployment:

- 1 vCPU: requires a minimum of 4 GB RAM allocation.
- 2 vCPUs: requires a minimum of 4 GB RAM allocation.
- 4 vCPUs: requires a minimum of 4 GB RAM allocation.
- 8 vCPUs: requires a minimum of 4 GB RAM allocation.
- 16 vCPUs: requires a minimum of 8 GB RAM allocation.

## Restrictions

Specifies the restrictions or limitations for deploying Cisco Catalyst 8000V in Nutanix AHV.

This list specifies the limitations for deploying Cisco Catalyst 8000V in Nutanix AHV

- Hot add and hot delete operations for NICs are not supported.
- In Cisco IOS XE 26.1.1, SR-IOV is supported only with NVIDIA ConnectX-6 NICs.

## Upload the Cisco Catalyst 8000V image file

Provides information on how to upload a .ova file or a .qcow2 file for Cisco Catalyst 8000V installation on Nutanix AHV hypervisor.

Before you install Cisco Catalyst 8000V on Nutanix AHV, upload the .ova or the .qcow2 file to Prism Central.

- For instructions on uploading a .ova file, refer to the Upload the OVA file section in this document.
- For instructions on uploading a .qcow2 file, refer to the Upload the qcow2 file section in this document.

### Upload the OVA file

Specifies the steps to upload the OVA file while deploying Cisco Catalyst 8000V in Nutanix AHV.

Complete these steps to upload the OVA file before installing Cisco Catalyst 8000V on Nutanix Hypervisor. If you have a qcow2 file, refer to the Upload the qcow2 file section in this document.

#### Procedure

1. Log in to the Prism Central Dashboard.
2. Click **Compute** > **OVAs** and click **Upload OVA**.
3. In the **Upload OVA** window, select the **OVA File** radio button to upload the file from your device, or select the **URL** radio button to upload the file from a URL.
4. From the **Select AHV Cluster** drop-down list, select the cluster.
5. In the **Name** field, specify a name for your file.
6. In the **Checksum** field, choose either **SHA-256** or **SHA-1**, and enter the appropriate value.
7. Perform one of these actions:
   - If you chose the OVA radio button, in the **Choose OVA File** field, click **Upload** to upload the file.
   - If you chose the **URL** option to provide the file, specify the link to the file in the **URL** field.
8. Click **Upload**.

The system displays your OVA file in the dashboard. You can use the file for the VM deployment.

---

**Upload the QCOW2 file**

Lists the steps to upload a .qcow2 file for installing Cisco Catalyst 8000V in Nutanix AHV.

This procedure lists the steps you need to perform to upload the .qcow2 file for installing Cisco Catalyst 8000V in Nutanix AHV. If you have uploaded an iso file already, ignore this section and proceed to the installation steps.

**Procedure**

---

1. Log in to the Prism Central GUI.
2. From the navigation, choose **Compute & Storage** > **Images**.
3. Click **Add Image** and under **Image Source**, choose one of these options:
   - **Upload a File**: Click **Choose File** and select the .qcow2 file from your system.
   - **URL**: Specify the URL where the QCOW2 file is hosted.
4. Specify these details:
   a) **Name**: Provide a unique image name for the file, for example, c8000v-universalk9_17.12.06a_qcow2.
   b) **Image Type**: Select the **Disk** option. Ensure you do not select ISO in this field. Choosing ISO would result in VM boot failure.
   c) **Description**: Optionally provide a description for your file.
5. Select the appropriate **Cluster or Storage Container** based on your Prism Central version.
6. Click **Save** to upload the file.

---

## Install the Virtual Machine Using Prism Central

Lists the possible installation methods for installing Cisco Catalyst 8000V in Nutanix AHV hypervisor.

Prism Central offers a centralized management platform for multiple clusters from a single interface. Perform the procedures in this document to install Cisco Catalyst 8000V on Nutanix AHV hypervisor using Prism Central.

### Configure VM properties

Lists the steps to configure the properties of the VM when installing Cisco Catalyst 8000V in Nutanix AHV via Prism Central.

Perform these steps to configure the properties of the VM when installing Cisco Catalyst 8000V in Nutanix AHV via Prism Central.

**Procedure**

---

1. Log in to the Prism Central dashboard.
2. In the left navigation, choose **Compute** > **VMs** > **Create VM**.
3. In the **Name** field, specify a name for the VM. If desired, provide a description for the VM in the **Description** field.
4. From the **Project** drop-down list, choose the project.
5. From the **Cluster** drop-down list, choose the cluster to which you want to add the VM.

6. Select the **Number of VMs**.
7. In the **VM Properties** field, configure these settings:
   a) **CPUs**: Enter the number of virtual CPUs to allocate to this VM.
   b) **Cores per CPU**: Enter the number of cores assigned to each virtual CPU.
   c) **Memory**: Enter the amount of memory (in GiB) to allocate to this VM.
8. Click **Next** to proceed to disk and network settings.

## Configure the disk and network settings

Lists the steps to configure the disk and network settings while installing Cisco Catalyst 8000V on Nutanix AHV hypervisor by using Prism Central.

**Procedure**

1. In the **Resources** page, click **Attach Disk** to attach a disk.
2. In the **Attach Disk** pop-up window, configure these disk settings:
   a) **Type**: Select the type of storage device, **DISK** or **CD-ROM**, from the drop-down list. Choose Disk for hard disk, CDROM for .iso files, and CD-ROM for bootstrap configuration files.
   b) **Operation**: Specify whether to create a blank CD-ROM device, allocate space without specifying an image, or clone from an image using this drop-down list.
   c) **Storage Container**: Choose either the default container or a self-service container.
   d) **Capacity**: Choose the disk capacity from this drop-down list.
   e) **Bus Type**: Select the Bus Type based on the **Type** option you chose. For virtIO, use **SCSI** as the Bus Type.
3. Click **Save** to save the Disk settings.
4. Click the **Attach to Subnet** button to attach the VM to a subnet.
5. In the **Attach to Subnet** pop-up window, configure these settings:
   a) From the **Subnet Name** drop-down list, choose the target vLAN.
   b) From the **Network Connection State** drop-down list, choose the state for the network that you want it to operate in after VM creation. The available options are **Connected** and **Disconnected**.
6. Click **Save** to save these settings.

## Provide custom configuration and create the VM

Lists the steps to configure custom settings while creating the VM and powering on the VM after successful creation.

Perform these steps to configure custom settings while creating the VM during Cisco Catalyst 8000V installation on Nutanix AHV hypervisor using Prism Central.

**Procedure**

1. In the **Management** page, you can assign a category to your VM.
2. Optionally, choose the **Timezone** for your VM. If needed, set this VM as the Agent VM to be powered on first.
3. From the **Guest Customization** script drop-down field, choose a **Script Type** for your Startup Script.
4. From the **Configuration Method** drop-down field, choose **Custom Script** if you wish to provide a customized configuration.
5. Upload or paste your startup configuration in the **Startup Script** field.
6. Click **Next** to go to the **Review** page and review all your settings.

7. Click **Create VM** to create the virtual machine.
8. To power on the VM, in the web console, choose **VM** > **More Actions** > **Power Operations** > **Power On**.

   After the VM is created and powered on successfully, you can access the VM from the console by navigating to **VM** > **Launch Console**.

### Verify successful VM installation

Specifies the verification steps after the Cisco Catalyst 8000V instance is successfully deployed on Nutanix AHV and is up and running.

To verify that the Cisco Catalyst 8000V instance is successfully deployed on Nutanix AHV and is up and running, perform these steps.

**Procedure**

1. Log in to the Prism Central GUI or the Prism Element GUI.
2. Choose **VM** > **Launch Console** to access the VM from the console.
3. To access the VM through the Cisco Catalyst 8000V command line, run the **show platform software system hypervisor** command.

```
C8000v# show platform software system hypervisor
Hypervisor: KVM
Manufacturer: Nutanix
Product Name: AHV
Serial Number: xxxxx
UUID: xxxxx
Image Variant: None
```

   The output should display **Nutanix** against the **Manufacturer** field and **AHV** against the **Product Name** field.

## Install the Virtual Machine Using Prism Element

Provides information on installing Cisco Catalyst 8000V on Nutanix AHV hypervisor using Prism Element, a built-in, localized management interface for individual Nutanix clusters.

Nutanix Prism Element (PE) is the built-in, localized management interface for individual Nutanix clusters, running on every node. This interface provides configuration, monitoring, and VM management capabilities.

This document provides information on installing Cisco Catalyst 8000V on Nutanix AHV hypervisor using Prism Element. This installation method is suitable for single clusters with a localized management of hardware, storage, and networking within the cluster.

### Configure disk properties

Explains the steps to configure the disk properties for a VM when installing Cisco Catalyst 8000V on Nutanix AHV using Prism Element.

Perform these steps to configure the disk properties for a VM when installing Cisco Catalyst 8000V on Nutanix AHV using Prism Element.

**Procedure**

1. Log in to the **Controller VM Dashboard**.
2. Click **Settings** > **Image Configuration** and upload the Cisco Catalyst 8000V .iso image file and the bootstrap configuration file.
3. In the VM dashboard, click the **Create VM** button.
4. In the **Add Disk** pop-up window, configure these settings:
   a) **Type**: Select the type of storage device, **DISK** or **CD-ROM**, from the drop-down list. Choose Disk for harddisk, choose CDROM for .iso files, and choose CD-ROM for bootstrap configuration files.
   b) **Operation**: Specify the option to create a blank CD-ROM device or allocate space without specifying an image, or to clone from an image, from this drop-down list.
   c) **Bus Type**: Select the Bus Type from the drop-down list based on the **Type** option you chose.
   d) **Image**: Select the image that you have created by using the image service feature. This field is applicable only when **Clone from Image Service** is selected in Operation. This field specifies the image to copy.
   e) **Logical Size (GIB)**: Enter the disk size in GiB.
5. Click **Add** and proceed to the next task to create a vNIC.

## Create a vNIC

Explains how to create a vNIC when installing Cisco Catalyst 8000V on Nutanix AHV hypervisor using Prism Element.

Perform these steps to create a vNIC when installing Cisco Catalyst 8000V on Nutanix AHV hypervisor using Prism Element.

**Before you begin**

Ensure you complete the Add Disk procedure before performing this task.

**Procedure**

1. Navigate to the **Create vNIC** pop-up window and configure the settings specified here.
2. From the **Subnet Name** drop-down list, choose the target VLAN.
3. From the **Network Connection State** drop-down list, choose the state for the network that you want it to operate in after VM creation. The available options are **Connected** and **Disconnected**.
4. Review all the settings. Then, click **Add** to create the network interface for the VM.

   Repeat this task to create additional interfaces, if required.

## Create the VM

Provides instructions for creating a VM during the installation of Cisco Catalyst 8000V on Nutanix AHV hypervisor using Prism Element.

Provides instructions for creating a VM during the installation of Cisco Catalyst 8000V on Nutanix AHV hypervisor using Prism Element.

After you add the network interfaces, the system displays the **Create VM** pop-up window. Use the displayed options to create the VM.

**Procedure**

1. Review the Network Adapters (NIC). Add, edit, or delete a NIC, if required.
2. To configure affinity policy for this VM, click **Set Affinity**.
3. In the **Set VM Host Affinity** pop-up window, select the host or hosts on which you want to configure the affinity for this VM.
4. To customize the VM by using a custom script, select the **Custom Script** check box. Paste the script in this field.
5. Upload the user data to a storage system. Upload it from your local device, or copy and paste the content of the script in the provided text box.
6. Click **Save** to create the **VM**.
7. In the web console, choose **VM** > **Power On** to boot the VM.

   After the VM is created and powered on successfully, you can access the VM from the console by navigating to **VM** > **Launch Console**.

## Install the VM using command line interface

Provides the instructions to install Cisco Catalyst 8000V on Nutanix AHV hypervisor using command line interface.

Perform these steps to install Cisco Catalyst 8000V on Nutanix AHV hypervisor using command line interface.

**Procedure**

1. Run this command to create the VM.

```
acli vm.create c8kv-test memory=8G num_vcpus=4 num_cores_per_vcpu=1
```

2. Create the system disk and attach it to the virtual machine.

```
acli vm.disk_create c8kv-test create_size=20G container=default-container
```

3. Attach the ISO files.

```
acli vm.disk_create c8kv-test cdrom=true clone_from_image=c8kv-iso-image
acli vm.disk_create c8kv-test cdrom=true clone_from_image=c8kv-bootstrap-config
```

4. Create the network interface.

```
acli vm.nic_create c8kv-test network=labnet
```

5. Power on the VM.

```
acli vm.on c8kv-test
```

## Install the VM using REST API

Provides the steps to install the Cisco Catalyst 8000V VM using the Nutanix REST API.

Provides the steps to install the Cisco Catalyst 8000V VM using the Nutanix REST API.

This installation method involves installation using a REST client like Postman. To create a VM using the Nutanix REST API, send a POST request to the Prism Central VM endpoint with a JSON that defines the VM specifications. Follow these steps to perform this installation.

### Before you begin

You must have access to a REST client such as Postman.

### Procedure

1. Log in to the Prism Central GUI.
2. On the top right corner of the page, click the drop-down arrow.
3. Choose **REST API Explorer**.
4. Configure these parameters in a REST client like Postman.
   a) **REST Endpoint**: Provide the details for the REST endpoint.
   b) **HTTP Method**: Specify the method, for example, POST.
   c) **Authentication**: Use **Basic Auth** with your Prism Central username and password.
   d) **Credentials**: Provide the credentials used to log in to the Prism Central GUI.
5. Provide the JSON body for creating a VM using the details specified in the table.

| Parameter | Specification |
|---|---|
| Disks | Hard disk with 20GB capacity<br>CDROM for the ISO image<br>CDROM for the bootstrap configuration |
| Networks | To get the list of networks to use for creating interfaces for the VM, use this REST API endpoint with a GET call: https://{Nutanix-Cluster-Id}/api/nutanix/v2.0/networks |

### Sample JSON

```
{
  "description": "VM with CD-ROM referencing image",
  "memory_mb": 8196,
  "name": "c8kv-test",
  "num_vcpus": 4,
  "num_cores_per_vcpu": 1,
  "machine_type": "Q35",
  "boot": {
    "uefi_boot": true
  },
  "vm_nics": [
    {
      "network_uuid": "77979a04-861b-48bf-a790-4306953e1b17",
      "connect": true
```

```
    },
    {
      "network_uuid": "d5eee64d-8510-4485-814d-a479959e48f3",
      "connect": true
    }
  ],
  "vm_disks": [
    {
      "disk_address": {
        "device_bus": "SCSI",
        "device_index": 0
      },
      "is_cdrom": false,
      "vm_disk_create": {
        "size": 21474836480,
        "storage_container_uuid": "61a44dc6-ca9c-4507-ae89-82705046e3e7"
      }
    },
    {
      "disk_address": {
        "device_bus": "SATA",
        "device_index": 0
      },
      "is_cdrom": true,
      "vm_disk_clone": {
        "disk_address": {
          "device_bus": "SATA",
          "device_index": 0,
          "vmdisk_uuid": "624b05b2-3e97-48a4-8d63-f47f8d9c56bc"
        }
      }
    },
    {
      "disk_address": {
        "device_bus": "SATA",
        "device_index": 1
      },
      "is_cdrom": true,
      "vm_disk_clone": {
        "disk_address": {
          "device_bus": "SATA",
          "device_index": 1,
          "vmdisk_uuid": "35df40b3-f6a3-48f1-bb93-c22e4cbf9144"
        }
      }
    }
  ]
}
```

## Verify VM creation

Specifies how to verify whether the VM was created successfully using REST API.

To check the status of the REST API call, ensure that:

- the HTTP response status is 201 created, and
- the response body has a task_uuid returned.

Verify the task status by sending a GET request to this REST API endpoint:
https://{Nutanix-Cluster-URL}/api/nutanix/v3/tasks/{task_uuid}

## Edit the VM properties

Specifies the steps to configure additional settings or edit your current settings after the VM is up.

After you install and bring up your VM, if you want to configure additional settings or if you want to edit your current settings, follow these steps.

**Procedure**

1. Log in to the Prism Central or the Prism Element GUI.
2. Choose **Virtual Infrastructure** > **VMs**.
3. On the **Virtual Machines** page, select the VM whose properties you want to modify.
4. On the **VM Properties** page, choose **Actions** > **Update**.
5. On the **VM Instance** page, navigate through the various tabs to modify or customize the settings for your VM.
6. After reviewing the changes, click **Save** to apply the changes.

## Clone a VM

Describes the steps to clone a Cisco Catalyst 8000V VM in Nutanix AHV hypervisor.

**Procedure**

1. Log in to the Prism Central GUI.
2. In the navigation pane, choose **Compute & Storage** > **VMs**.
3. Perform the Clone action using one of these steps:
   - In the **List** tab, choose the target VM and then choose **Clone** from the **Actions** drop-down menu.
   - In the **List** tab, right-click the target VM and select **Clone** from the drop-down list.
4. Click **Clone** in the **Summary** page of an individual VM.

   The system displays the **Clone VM** window, which includes the same fields as the **Create VM** window.
5. For the clone VM, configure these settings:
   a) Enter a unique **Name** for the cloned VM.
   b) Verify and select the appropriate **Project**.
   c) Select the target **Cluster**.
   d) Specify the **Number of Clusters**.
   e) Under **Guest Customization**, select the customization method if required.
6. Click **Next**.
7. Review the settings, and click **Create/Clone**.

> **Note**
>
> If the source VM uses static IP addressing, ensure you update the **Network Settings** after cloning to prevent IP conflicts.

## Verify VM template cloning

Provides instructions to verify whether the VM cloning in Nutanix AHV hypervisor is successful.

Take these actions to verify whether the VM cloning in Nutanix AHV hypervisor succeeded.

**Procedure**

1. Click the **Activity/Tasks (bell icon)** in the top-right corner.
2. Verify whether the task status displays **Succeeded**.
3. After the cloning is successful, check the power status of the VM. The VM should be powered on.
4. Optionally, update the hostname and the IP address of the VM. Then, confirm the network connectivity status.

# 7 Install Cisco Catalyst 8000V in Azure Local

**Topics:**


- Install Cisco Catalyst 8000V in Azure Local hyperconverged infrastructure
- Prerequisites in Azure Local
- Installation requirements
- Restrictions for deploying Cisco Catalyst 8000V in Azure Local
- Deploy the Cisco Catalyst 8000V VM using PowerShell


Provides a comprehensive guide for deploying and managing Cisco Catalyst 8000V virtual machines in Azure Local. This document covers overview, prerequisites, and deployment method.

# Install Cisco Catalyst 8000V in Azure Local hyperconverged infrastructure

Provides an overview about Azure Local hyperconverged infrastructure and specifies the requirements, installation methods, and verification of Cisco Catalyst 8000V deployment in Azure Local.

Azure Local hyperconverged infrastructure (HCI), formerly Azure Stack HCI, is a Microsoft-delivered set of integrated hardware platforms and software that delivers Microsoft Azure public cloud services in a local data center to enable organizations to construct hybrid clouds.

Azure Local is also a subscription-based, hyper-converged infrastructure operating system that is delivered as an Azure service. This solution combines Windows and Linux virtualization and is one of the variants of Azure Stack that provides on-prem, hyper-converged infrastructure intended for data centers and remote branches.

### Benefits

Cisco Catalyst 8000V running on Azure Local offers these benefits:

- Azure Local solutions offer virtualized compute, storage, and networking capabilities on industry-standard x86 servers.
- Azure Local allows you to combine resources in the same cluster, making it easier to deploy, manage, and scale.

# Prerequisites in Azure Local

Provides the requirements for installing Cisco Catalyst 8000V on Azure Local.

This list provides the requirements for installing Cisco Catalyst 8000V on Azure Local. Before you deploy the Cisco Catalyst 8000V virtual machine, verify that:

- You have the configuration information for the VM as an .iso file.
- You have a valid user account or subscription with Microsoft Azure.
- You have an active login for Microsoft Azure PowerShell.
- You are familiar with the basics of Azure Local, including the necessary Microsoft Azure PowerShell commands for creating and managing a virtual machine (VM).
- You deploy the necessary Microsoft Azure resources for the deployment.

# Installation requirements

This section lists the supported Azure Local versions on which Cisco Catalyst 8000V has been validated.

This section lists the supported Azure Local versions on which Cisco Catalyst 8000V has been validated.

### Supported Azure Local versions

This table lists the supported Azure Local versions for Cisco Catalyst 8000V.

| Cisco IOS XE release | Azure Local version |
|---|---|
| Cisco IOS XE 26.2.x release | Azure Local version 23H2 |
| Cisco IOS XE 26.1.x release | Azure Local version 23H2 |
| Cisco IOS XE 17.18.2 release and later | Azure Local version 23H2 |

| Cisco IOS XE release | Azure Local version |
|---|---|
| Cisco IOS XE 17.15.5 release and later | Azure Local version 23H2 |

## Restrictions for deploying Cisco Catalyst 8000V in Azure Local

Specifies the restrictions or limitations for deploying Cisco Catalyst 8000V in Azure Local.

This list specifies the limitations for deploying Cisco Catalyst 8000V in Azure Local.

- Authorization permissions must be correctly configured for the Cisco Catalyst 8000V to access Azure resources such as route tables. Lack of permissions can cause failures in route table updates and high availability functions.
- Ensure that the identifiers in the redundancy node match the values in the Azure Portal to avoid synchronization issues.
- When you upload the configuration file, the first attempt could fail. Reupload the file to configure the Day 0 settings.

## Deploy the Cisco Catalyst 8000V VM using PowerShell

Provides all the details you need to deploy the Cisco Catalyst 8000V VM using PowerShell.

This section describes how to create and manage VMs on Azure Local using Windows PowerShell. The following table outlines the high-level steps involved to create and manage the VM. It also provides the link to the respective Azure Local documentation. To learn about the PowerShell commands used in detail. refer to this PowerShell Command page.

- Review the detailed Azure Local documentation for each step.
- Refer to the PowerShell command reference for syntax and usage examples.

| Action | PowerShell command and description | Link to the Azure PowerShell documentation |
|---|---|---|
| Create the VM | `New-VM`<br>Creates a new VM. | https://learn.microsoft.com/en-us/azure/azure-local/manage/vm-powershell?view=azloc-2604#create-a-vm |
| Create a disk | `New-VHD`<br>Creates one or more new virtual hard disks. | https://learn.microsoft.com/en-us/powershell/module/hyper-v/new-vhd?view=windowsserver2025-ps |
| Add a hard disk | `Add-VMHardDiskDrive`<br>Adds a hard disk drive to a virtual machine. | https://learn.microsoft.com/en-us/powershell/module/hyper-v/add-vmharddiskdrive?view=windowsserver2025-ps |
| Configure the VM settings | `Set-VMProcessor`<br>Configures settings for the virtual processors of a virtual machine. Settings are applied uniformly to all virtual processors belonging to the virtual machine. | https://learn.microsoft.com/en-us/powershell/module/hyper-v/set-vmprocessor?view=windowsserver2025-ps |
| Configure the memory | `Set-VMMemory`<br>Configures the memory of a virtual machine. | https://learn.microsoft.com/en-us/powershell/module/hyper-v/set-vmmemory?view=windowsserver2025-ps |
| Connect the switch and the VLAN | `Connect-VMNetworkAdapter`<br>Connects a virtual network adapter to a virtual switch. | https://learn.microsoft.com/en-us/powershell/module/hyper-v/connect-vmnetworkadapter?view=windowsserver2025-ps |
| Configure the firmware of the VM | `Set-VMFirmware`<br>Sets the firmware configuration of a virtual machine. | https://learn.microsoft.com/en-us/powershell/module/hyper-v/set-vmfirmware?view=windowsserver2025-ps |
| Configure a virtual DVD Drive | `Set-VMDvdDrive`<br>Configures a virtual DVD drive. | https://learn.microsoft.com/en-us/powershell/module/hyper-v/set-vmdvddrive?view=windowsserver2025-ps |
| Start the VM | `Start-VM`<br>Starts the virtual machine. | https://learn.microsoft.com/en-us/powershell/module/hyper-v/start-vm?view=windowsserver2025-ps |

## VM deployment workflow

This section provides a sample end-to-end workflow to create a Cisco Catalyst 8000V VM and deploy the instance in

Azure Local using PowerShell commands.

```
PS C:\Users\azs-admin>
```

**Procedure**

1. Set the required parameters.

   **Example**

```
$HostName    = "hostname.com"
$VMName      = "vm-name"

$VMPath      = "C:\ProgramData\Microsoft\Windows\Hyper-V"
$CpuCount    = 4
$MemoryGB    = 8

$SwitchName  = "ConvergedSwitch(compute_management)"
$VlanId      = 3309

$NewDiskGB   = 16

$OsIsoPath   =
"C:\ClusterStorage\UserStorage_6\c8000v-universalk9_vga.2026-01-05_14.36_xyz.iso"

$Day0IsoPath = "C:\ClusterStorage\UserStorage_6\day0_2ctrl.iso"

Invoke-Command -ComputerName $HostName -ScriptBlock {
param(
    [string]$VMName,
    [string]$VMPath,
    [int]$CpuCount,
    [int]$MemoryGB,
    [string]$SwitchName,
    [int]$VlanId,
    [int]$NewDiskGB,
    [string]$OsIsoPath,
    [string]$Day0IsoPath
)

$vmFolder = Join-Path $VMPath $VMName
$vhdPath  = Join-Path $vmFolder "$VMName.vhdx"

# Ensure folder exists (on the HOST, not your local machine)
New-Item -ItemType Directory -Path $vmFolder -Force | Out-Null
```

2. Create a disk.

   **Example**

```
if (-not (Test-Path $vhdPath)) {
    New-VHD -Path $vhdPath -SizeBytes ($NewDiskGB * 1GB) -Dynamic | Out-Null

}
```

3. Create the VM (Gen2).

**Example**

```
if (-not (Get-VM -Name $VMName -ErrorAction SilentlyContinue)) {
    New-VM -Name $VMName -Generation 2 -Path $vmFolder -MemoryStartupBytes
($MemoryGB * 1GB) | Out-Null
}
```

4. Attach a disk (if not already attached).

**Example**

```
$hasDisk = (Get-VMHardDiskDrive -VMName $VMName -ErrorAction SilentlyContinue
 | Where-Object { $_.Path -eq $vhdPath })
if (-not $hasDisk) {
    Add-VMHardDiskDrive -VMName $VMName -Path $vhdPath | Out-Null
}
```

5. Set a CPU.

**Example**

```
Set-VMProcessor -VMName $VMName -Count $CpuCount | Out-Null
```

6. Set memory as force static (disable dynamic).

**Example**

```
Set-VMMemory -VMName $VMName -DynamicMemoryEnabled $false -StartupBytes
($MemoryGB * 1GB) | Out-Null
```

7. Configure switch and VLAN settings.

**Example**

```
# (If NIC exists but not connected, connect it; otherwise just set VLAN)
Connect-VMNetworkAdapter -VMName $VMName -SwitchName $SwitchName -ErrorAction
 SilentlyContinue | Out-Null
Set-VMNetworkAdapterVlan -VMName $VMName -Access -VlanId $VlanId | Out-Null

$nic = Get-VMNetworkAdapter -VMName $VMName
# 1. Bandwidth Settings $bf = Get-VMSystemSwitchExtensionPortFeature `
-FeatureName "Ethernet Switch Port Bandwidth Settings"
Add-VMSwitchExtensionPortFeature -VMNetworkAdapter $nic `
-VMSwitchExtensionFeature $bf | Out-Null

# 2. Security Settings $sf = Get-VMSystemSwitchExtensionPortFeature `
-FeatureName "Ethernet Switch Port Security Settings"
Add-VMSwitchExtensionPortFeature -VMNetworkAdapter $nic `
-VMSwitchExtensionFeature $sf | Out-Null

# 3. Profile Settings $pf = Get-VMSystemSwitchExtensionPortFeature `
-FeatureName "Ethernet Switch Port Profile Settings" $pf.SettingData.ProfileId
 = "{00000000-0000-0000-0000-000000000000}" $pf.SettingData.ProfileData = 2
$pf.SettingData.ProfileName = "Testprofile" $pf.SettingData.VendorId =
"{1FA41B39-B444-4E43-B35A-E1F7985FD548}" $pf.SettingData.VendorName =
"NetworkController" $pf.SettingData.NetCfgInstanceId =
"{56785678-a0e5-4a26-bc9b-c0cba27311a3}" $pf.SettingData.CdnLabelId = 1111
```

```
$pf.SettingData.CdnLabelString = "TestCdn" Add-VMSwitchExtensionPortFeature
-VMNetworkAdapter $nic ` -VMSwitchExtensionFeature $pf | Out-Null
```

8. Enable Secure Boot off.

   **Example**

   Set-VMFirmware -VMName $VMName -EnableSecureBoot Off | Out-Null

   > **Note**
   >
   > Secure Boot is disabled here for simplicity. Cisco Catalyst 8000V also supports Secure Boot with the Microsoft UEFI Certificate Authority template if your security policy requires it:
   >
   > ```
   > Set-VMFirmware -VMName $VMName -EnableSecureBoot On -SecureBootTemplate
   > "MicrosoftUEFICertificateAuthority"
   > ```

   > **Note**
   >
   > Do not use the default "MicrosoftWindows" Secure Boot template. Doing so will prevent the Cisco Catalyst 8000V from booting.

9. Configure DVD 1: OS ISO

   **Example**

   ```
   $dvdDrives = Get-VMDvdDrive -VMName $VMName -ErrorAction SilentlyContinue
   if (-not $dvdDrives) {
       Add-VMDvdDrive -VMName $VMName -Path $OsIsoPath | Out-Null
   } else {
       Set-VMDvdDrive -VMName $VMName -ControllerNumber
   $dvdDrives[0].ControllerNumber -ControllerLocation
   $dvdDrives[0].ControllerLocation -Path $OsIsoPath | Out-Null
   }
   ```

10. Configure DVD 2: Day0 ISO

    **Example**

    ```
    $dvdDrives = Get-VMDvdDrive -VMName $VMName
    if ($dvdDrives.Count -lt 2) {
        Add-VMDvdDrive -VMName $VMName -Path $Day0IsoPath | Out-Null
    } else {
        Set-VMDvdDrive -VMName $VMName -ControllerNumber
    $dvdDrives[1].ControllerNumber -ControllerLocation
    $dvdDrives[1].ControllerLocation -Path $Day0IsoPath | Out-Null
    }
    ```

11. Set the first boot from DVD.

**Example**

```
$dvd = (Get-VMDvdDrive -VMName $VMName | Select-Object -First 1)
Set-VMFirmware -VMName $VMName -FirstBootDevice $dvd | Out-Null
```

12. Start the VM.

**Example**

```
Start-VM -Name $VMName -ErrorAction SilentlyContinue | Out-Null
```

13. Set VHDX as first boot device.

**Example**

```
$VMName   = "vm-name”
$HVHost   = "HOST1"

$hdd = Get-VMHardDiskDrive -ComputerName $HVHost -VMName $VMName |
Select-Object -First 1
Set-VMFirmware -ComputerName $HVHost -VMName $VMName -FirstBootDevice $hdd
```

14. Connect to VM.

**Example**

```
PS C:\Users\azs-admin> & "C:\Windows\System32\vmconnect.exe" HOST "c8000v-name"
```

15. Delete the VM using PowerShell.

**Example**

```
# Delete VM including VHD files
Invoke-Command -ComputerName $HostName -Credential $cred -ScriptBlock {
param($VMName)

# Stop VM
Stop-VM -Name $VMName -Force -TurnOff -ErrorAction SilentlyContinue
Start-Sleep -Seconds 3

# Get VHD paths before removing VM
$vhds = Get-VMHardDiskDrive -VMName $VMName | Select-Object -ExpandProperty
Path
$vmPath = (Get-VM -Name $VMName).Path

# Remove VM
Remove-VM -Name $VMName -Force

# Delete VHD files
foreach ($vhd in $vhds) {
if (Test-Path $vhd) {
Remove-Item $vhd -Force
Write-Host "Deleted VHD: $vhd" -ForegroundColor Green
}
}
```

```
# Delete VM folder
if ($vmPath -and (Test-Path $vmPath)) {
Remove-Item $vmPath -Recurse -Force
Write-Host "Deleted VM folder: $vmPath" -ForegroundColor Green
}

Write-Host "VM '$VMName' deleted successfully." -ForegroundColor Cyan
} -ArgumentList $VMName
```

**Note**

Make sure to note the VHD paths before running the `Remove-VM` command. After the VM is deleted, its disk paths can no longer be queried.

16. Update the VM settings.

**Example**

**Note**

For some update operations, you must stop the VM first.

To update VM processor, use these commands.

**Example**

```
$NewCpu    = 8
Set-VMProcessor -ComputerName $HostName -VMName $VMName -Count $NewCpu
```

## Configure network settings

This section outlines the network settings configuration.

**Procedure**

1. To get the VM, use this command:

**Example**

```
Get-VMNetworkAdapter -ComputerName $HostName -VMName $VMName |
Select Name, SwitchName, MacAddress, DynamicMacAddressEnabled

Name            SwitchName                          MacAddress
DynamicMacAddressEnabled
----            ----------                          ----------
------------------------
```

```
Network Adapter ConvergedSwitch(compute_management) 00155D1F0931
        False
```

2. To set the existing NIC to static MAC, use this command:

   **Example**

```
$HostName  = "hostname.com"
$VMName    = "vm-name"
$Mac       = "00155D1F0931"

Set-VMNetworkAdapter `
  -ComputerName $HostName `
  -VMName $VMName `
  -Name "Network Adapter" `
  -StaticMacAddress $Mac
```

3. To add a network adapter, use this command:

   **Example**

```
 $HostName   = " hostname.com"
$VMName     = " vm-name"
$SwitchName = "ConvergedSwitch(compute_management)"
$NicName    = "mgmtnicname"

Add-VMNetworkAdapter `
  -ComputerName $HostName `
  -VMName $VMName `
  -SwitchName $SwitchName `
  -Name $NicName
```

4. To assign a VLAN, use this command:

   **Example**

```
 VLAN 3311 (access mode)
Set-VMNetworkAdapterVlan `
  -ComputerName $HostName `
  -VMName $VMName `
  -VMNetworkAdapterName $NicName `
  -Access `
  -VlanId 3311
```

   **Example**

> **Note**
>
> Additional NICs may require the same extension port feature sequence as the primary NIC.

# 8 Install Cisco Catalyst 8000V in NFVIS Environment

**Topics:**


- Install Cisco Catalyst 8000V in NFVIS Environment


Outlines the procedures for deploying the Cisco Catalyst 8000V within a Cisco NFVIS environment, including image registration, resource provisioning, and step-by-step installation on platforms such as the Cisco Enterprise Network Compute System (ENCS).

# Install Cisco Catalyst 8000V in NFVIS Environment

Specifies the installation workflow for installing Cisco Catalyst 8000V in Cisco Enterprise NFVIS solution and provides information on upgrading a Cisco Integrated Services Virtual Router (ISRv) running on NFVIS to Cisco Catalyst 8000V.

Cisco Enterprise Network Function Virtualization Infrastructure Software (Cisco Enterprise NFVIS) is a Linux-based infrastructure software. It helps service providers and enterprises dynamically deploy virtualized network functions, including a virtual router, firewall, and WAN acceleration, on a supported Cisco device.

The Cisco Enterprise NFVIS solution helps service providers and enterprises convert critical network functions into software, making it possible to rapidly deploy and manage network services across diverse and distributed locations.

Cisco NFVIS version 4.4 or later is required for Cisco Catalyst 8000V installation.

This chapter specifies the installation workflow for Cisco Catalyst 8000V in NFVIS. This chapter also provides information on upgrading a Cisco Integrated Services Virtual Router (ISRv) running on NFVIS to Cisco Catalyst 8000V.

## Supported hardware

This section lists all the hardware requirements for installing Cisco Catalyst 8000V in NFVIS.

The tables in this section list all hardware requirements for installing Cisco Catalyst 8000V in NFVIS.

### Supported hardware platforms running NFVIS

See this list for all hardware platforms supported for this installation.

- Cisco 5400 Series Enterprise Network Compute System (ENCS)
- Cloud Services Platform 5000 Series (CSP)
- Cisco 8200 UCPE Series

### Supported network interface modules

See this list of all supported network interface modules (NIMs) for this installation.

- NIM-4G-LTE-VZ
- NIM-4G-LTE-ST
- NIM-4G-LTE-NA
- NIM-4G-LTE-GA
- NIM-4G-LTE-LA
- NIM-LTEA-EA
- NIM-LTEA-LA
- NIM-1MFT-T1/E1
- NIM-2MFT-T1/E1
- NIM-4MFT-T1/E1
- NIM-8MFT-T1/E1
- NIM-1CE1T1-PRI
- NIM-2CE1T1-PRI

- NIM-8CE1T1-PRI
- NIM-16A
- NIM-24A
- NIM-VA-B
- NIM-VAB-A
- NIM-VAB-M
- NIM-4SHDSL-EA
- NIM-1GE-CU-SFP
- NIM-2GE-CU-SFP
- NIM-ES2-8-P
- NIM-ES2-8 NIM-ES2-4

**Supported network interface cards**

The table lists all network interface cards (NICs) supported for this installation.

| Hardware | VNIC |
|---|---|
| ENCS | virtio, igbvf and i40evf |
| UCPE | virtio, igbvf and ixgbevf |
| CSP | • virtio, igbvf - supported from Cisco IOS XE 17.4.1<br>• i40evf - supported from Cisco IOS XE 17.4.1 to 17.8.x<br>• ConnectX-5VF and iavf - supported from Cisco IOS XE 17.9.1<br>• Ixgbe - supported from Cisco IOS XE 17.10.1 |

**Supported profiles**

See the list of profiles supported for this installation.

- Mini – 1vCPU
- Small – 2vCPU
- Medium – 4vCPU
- Large – 4vCPU

> **Note**
>
> Cisco Catalyst 8000V works as a low latency VM and performs as expected with dedicated vCPU cores.

## Installing the virtual machines

Follow the stages mentioned in this workflow to perform Cisco Catalyst 8000V installation in NFVIS.

### Summary

From the Cisco IOS XE 17.4.1 release, you can either freshly install a Cisco Catalyst 8000V VM in NFVIS, or you can upgrade from a Cisco ISRv to Cisco Catalyst 8000V.

### Workflow

To perform Cisco Catalyst 8000V installtion on NFVIS, proceed through each stage in this workflow.

1. Register a VM image: Copy or download the VM image to the NFVIS server. Alternatively, host the image on an HTTP or HTTPS server. After downloading the file, register the image using the registration API. This API allows you to specify the file path on the HTTP or HTTPS server where the tar.gz file is hosted. Registering the image is a one-time activity. After registration is active, multiple VMs can be deployed by using the registered image.
2. Create a custom profile: Optionally create a custom profile for the VM image. This action is beneficial if the profiles defined in the image file do not match the requirements. Custom profiles provide specific profiling details for a VM image, such as the number of virtual CPUs on which the VM runs, the amount of virtual memory the VM consumes. Depending on the topology, additional networks and bridges can be created during the deployment.
3. Deploy the VM: Use the deployment API to deploy the VM. This API provides values to the parameters that are passed to the system during deployment. Depending on the VM that is deployed, some parameters are mandatory and others are optional. For more details about the APIs, see VM Lifecycle Management APIs .
4. Manage and monitor the VM: Use APIs and commands to access the VM status and debug logs. Using the VM management APIs, you can start, stop, or reboot a VM, and view the statistics for a VM, such as CPU usage. You can change or update a VM profile to one of the existing profiles in the image file, or create a new custom profile for the VM. You can also add or update vNICs on a VM.

## Install Cisco Catalyst 8000V in NFVIS 4.5.1 or later versions

Follow the procedures mentioned under this section to install Cisco Catalyst 8000V in NFVIS 4.5.1 or later versions.

If you want to install Cisco Catalyst 8000V on NFVIS version 4.5.1 or later, follow the procedures in this section.

### Upload the installation file in NFVIS

This task specifies how to upload the installation file in NFVIS.

Perform these steps to upload the installation file in NFVIS.

#### Procedure

1. Log in to the NFVIS Portal.
2. In the portal, **Configuration** > **Virtual Machine** > **Images** > **Image Repository**.
3. Upload the installation file by performing one of these two options:
   - Selct **Local** > **Select File**. From your device, locate and select the installation file. The upload completes.

- Select **Remote**.

4. If you chose **Remote**, provide these details:
   a) **Image Name**: Specify the name of the image file in this field.
   b) **Protocol**: Choose the protocol from this drop-down list.
   c) **IP Address**: Specify the IP address for the remote location in this field.
   d) **Port**: Specify the port for the remote location in this field.
   e) **Image File Path**: Specify the file path to the installation file in this field.

### Create a network

Perform these steps to create a network when you install Cisco Catalyst 8000V in NFVIS version 4.5.1 or later.

#### Procedure

1. In the NFVIS Portal, choose **Configuration** > **Virtual Machine** > **Networking** > **Networks**.
2. To create a new network, click the **+** icon.
3. In the **Add Network** area, enter these details:
   a) **Network**: Choose the network from this drop-down list.
   b) **Mode**: Enter the mode in which the VNF will boot.
   c) **VLAN**: Choose the VLAN for the VM.
   d) **VLAN-Range**: Enter the range of VLAN for your VM.
   e) **Native VLAN**: Choose the native VLAN for your VM from this field.
   f) **Bridge**: The Layer 2 domain between virtual network interface controllers (vNICs) of VMs. Choose either the **Existing** or the **Create New** radio button.
   g) **Interface**: Choose the interface from this field for your VM.

   **Note**

   Single Root Input/Output Virtualization (SR-IOV) is not supported in this installation.

4. Click **Submit** .

### Create a VM package

Follow these steps to create a VM package when installing Cisco Catalyst 8000V in NFVIS 4.5.1 and later versions.

#### Procedure

1. In the NFVIS portal, choose **Configuration** > **Virtual Machine** > **Images** > **Image Packaging**.
2. To create a VM package, click the **+** icon.
3. In the **Image Packaging** area, configure these details.

| Field | Description |
|---|---|
| **Name** | The name associated with the VM packaging. |

| Field | Description |
|---|---|
| **Version** | The version of the package. |
| **VM Type** | The type of the VM for which you're creating the package. |
| **Dedicate Cores (Optimize)** | The number of dedicated cores a container requires. By default, the value is **False** . |
| **Serial Console** | Use this field to enable or disable access through the serial console. By default, the value is **Disable** . |
| **SRIOV Driver** | The SRIOV driver supported by the VM interfaces. |
| **Local** | Use this option if the image you want to bundle is available in the intdatastore. |
| **Upload Raw Images** | Use this option to upload an image from your local machine for packaging. |
| **Raw Disk File Bus** | Choose the root disk image bus from this drop-down list. |
| **Thick Disk Provisioning** | Choose **true** from the drop-down list to enable thick provisioning. By default, the value is **false**. |

4. To upload the bootstrap file, perform one of these options:
   - Choose **Local** and choose the **Add Local File** option to add a locally available bootstrap file.
   - Choose the **Upload Bootstrap Files** option to browse to the bootstrap configuration file from your computer.
   - Choose the track state of the VM from the **Monitored** drop-down field. By default, the value is **False** .
5. Click **Submit** to generate the VM Package.

### Deploy the VM

Follow these steps to deploy the VM in NFVIS.

#### Procedure

1. In the NFVIS Portal, choose **Configuration** > **Deploy**.
2. In the **VM Deployment** window, choose the **Router** icon.
3. Click the VM. Four drag handlers appears around the VM. Drag from one of those handlers to any of the networks and provide the details.
4. In the **VM Details** area, configure these deployment settings.

| Field | Description |
|---|---|
| **VM Name** | Specify the name for your VM. |
| **Image** | Choose the appropriate value from the drop-down list. |
| **Profile** | Choose a profile from the drop-down list. The default profile is used if no profile is specified during deployment. |

| Field | Description |
|---|---|
| **Group Name** | Choose a group if you want this VM to be associated with a specific group. |
| **VNC Password** | Enter the VNC password in this field. |
| **Controller** | Choose **non-vManage** to deploy in autonomous mode and choose **vManage** to deploy the VM in controller mode. |
| **Tech Package** | Choose the desired tech package from the drop-down field. The available options are network essentials, network advantage, and network premier. |
| **NGIO** | The Next Generation Input/Output (NGIO) option that determines the NIM enablement capability for the VM. Choose **ENABLE** from the NGIO drop-down list to enable NGIO. |
| **SSH Username** | The username to remotely log in to the Cisco Catalyst 8000V VM. |
| **SSH Password** | The SSH password to access the VM. |
| **Port Number** | The port number that is required for the SSH connectivity to the VM. |
| **External Port Number** | The external port number that is required for the SSH connectivity to the VM. |

5. Click **Deploy**.

## Install Cisco Catalyst 8000V in NFVIS 4.5.0 or earlier versions

Follow the procedures mentioned under this section to install Cisco Catalyst 8000V in NFVIS 4.5.0 or earlier versions.

If you want to install Cisco Catalyst 8000V on NFVIS version 4.5.0 or earlier, follow the procedures in this section.

### Deploy the virtual machine in NFVIS

Follow these steps to deploy the VM in NFVIS.

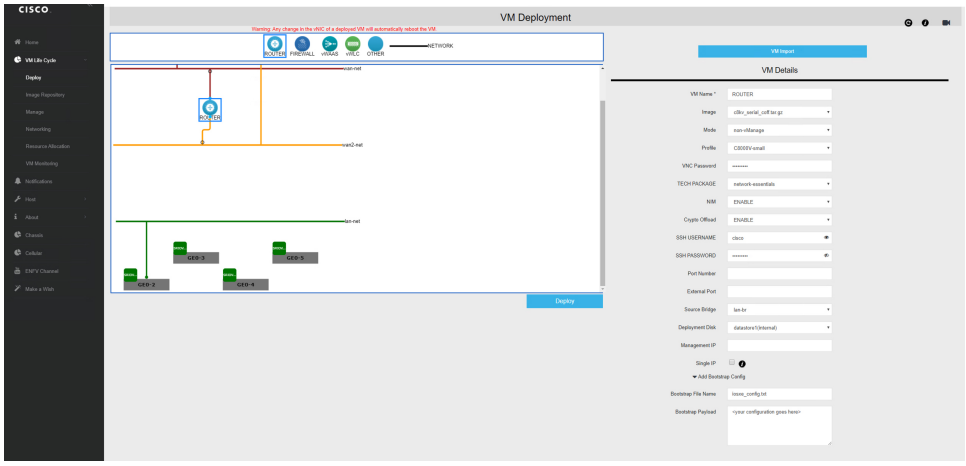
#### Procedure

1. In the NFVIS Portal, choose **VM Lifecycle** > **Deploy**.
2. In the **VM Deployment** window, drag and drop the **Router** icon to the pane and map it to the desired networks, as required.
3. In the **VM Details** section, configure these settings:

| Field | Description |
|---|---|
| **VM Name** | Enter the name for the virtual machine. |
| **Image** | Select the image from this drop-down field. |

| Field | Description |
|---|---|
| **Mode** | From this drop-down field, select the **non-vManage** option. |
| **Profile** | From this drop-down field, choose the profile that has to be applied to the VM. |
| **Tech Package** | From this drop-down field, choose the desired tech package. |
| **NIM** | Choose the **ENABLE** option from this drop-down field if a specific network function physical hardware is installed. Selecting the ENABLE option allows you to pass the hardware through into the VM. |
| **Crypto Offload** | Choose the **ENABLE** option from this drop-down field to offload the crypto processing to a hardware chip. |

4. Enter the username and password for the ssh login for Cisco Catalyst 8000V.
5. Configure the other VM details, if needed. These details include **VNC Password**, **Port Number**, **External Port**, **Source Bridge**, **Deployment Disk**, and **Management IP**.
6. Select the **Add Bootstrap Config** option to provide the bootstrap configuration file before deploying the VM. Ensure that you use the filename `iosxe_config.txt` for the bootstrap configuration file.




**Note**

The Gigabit Ethernet 1 interface is reserved for management communications with the NFVIS host.

7. Click **Deploy** to deploy the VM.

### What to do next

After deploying the VM instance, check the Instance details through the **Manage** tab. This tab lists the summary of the VM instances.

To access the console, click the Console symbol next to the VM. You can also connect to the serial console of the VM using this NFVIS command:

```
vmConsole <ROUTER-NAME>
```

### Download the Cisco Catalyst 8000V image

Perform these steps to download the Cisco Catalyst 8000V image for installation in NFVIS.

#### Procedure

1. Go to https://software.cisco.com/download/home.
2. In the **Search** bar at the bottom of the page, search Cisco Catalyst 8000V.
3. Select the **Software Type** from the list. For example, IOS XE Software.
4. From the list of files, download the Cisco Catalyst 8000V image file with the tar.gz extension.

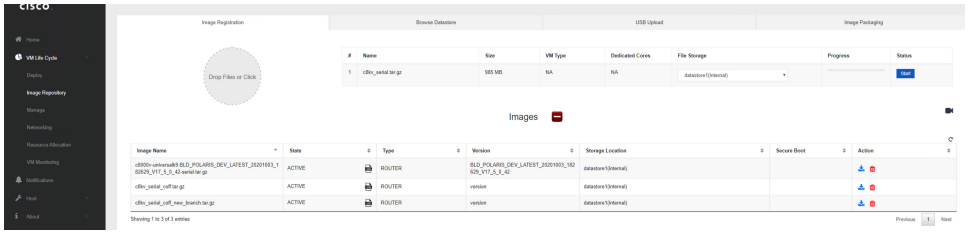
> **Note**
>
> To deploy a Cisco Catalyst 8000V image in NFVIS, the image must be packaged with the image properties file.

### Upload the image on NFVIS

Follow these steps to upload the image on NFVIS.

#### Procedure

1. Log in to the NFVIS Portal.
2. In the portal, choose **VM Lifecycle** > **Image Repository**.
3. Click the **Image Registration** tab, and click the upload arrow adjacent to the **Images** option.
4. From the **Drop Files or Click** option on top of the page, select the appropriate file.




5. Click **Start** to upload the image.

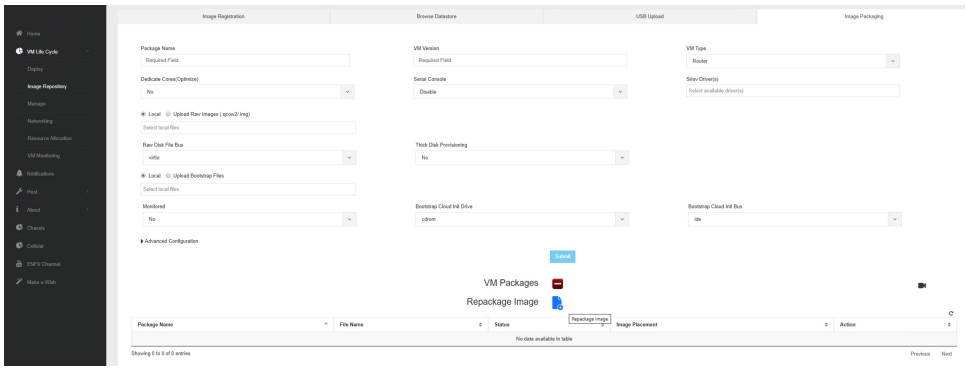
   After the image is uploaded, NFVIS creates the respective profiles and registers the image. You can find your file listed in the **Images** section on the same page.

### Create a VM Package using the web interface

Follow these steps to create a VM package when installing Cisco Catalyst 8000V using the web interface.

### Procedure

1. In the NFVIS Portal, choose **Image Repository** > **Image Packaging**.
2. Click the **Create** icon.
3. Click **VM Packages**.
4. In the **Image Packaging** tab, enter the appropriate details.




5. From the **Dedicated Code** drop-down list, select **Yes**.
6. Click **Submit** to upload the bootstrap files.
7. After the image is created, register the image to populate the profiles in NFVIS. Select the image you created, and click **Register**.

### Create a network

This procedure outlines the steps to create a network when installing Cisco Catalyst 8000V in NFVIS.

### Procedure

1. In the NFVIS Portal, select **VM Lifecycle** > **Networking**.
2. In the **Networks & Bridges** page, click the **Create** icon next to **Networks & Bridges**.
3. Enter the appropriate values for the **Network**, **Mode**, **VLAN**, **Bridge**, and **Interface** fields.

> **Note**
>
> Single Root Input/Output Virtualization (SRIOV) is not supported.

4. Click **Submit** to create the network.

## Monitor the virtual machine

Perform these steps to view the VM Resource Allocation and VM Statistics.

This section provides the steps to view the VM Resource Allocation and VM Statistics.

**Procedure**

1. To view the VM Resource Allocation follow these steps:
   a) In the NFVIS Portal, choose **VM Life Cycle** > **Resource Allocation**.

      The system displays the VM CPU Allocation tab which displays the overall CPU allocation.

   b) Click **VM Memory Allocation** to view the overall memory allocations.
   c) Click **VM Disk Allocation** to view the overall disk allocations.
2. To view the VM Stastics, perform these steps:
   a) In the NFVIS Portal select **VM Life Cycle** > **Resource Allocation**.

      The system displays the **VM CPU Utilization** tab which displays the overall CPU utilization per VM.

   b) Click **Memory Allocation** to view the memory utilization per VM.
   c) Click **VNC Utilization** tab to view the VNIC utilization per VM.
   d) Click the **Disk Utilization** tab to view the disk utilization per VM.

   The first interface on Cisco Catalyst 8000V is always reserved for Cisco NFVIS management network (generally Gigabit Ethernet 1). Cisco NFVIS assigns the IP address to this interface and it periodically monitors the VM by using ICMP pings via the interface.

   **Warning**

   Shutting down the interface or changing the IP address might result in the recovery and reload of the NFVIS VM.

## Upgrade and downgrade scenarios

This section describes the upgrade and downgrade between Cisco ISRV and Cisco Catalyst 8000V.

From the Cisco IOS XE 17.4.x release onwards, Cisco Catalyst 8000V replaces Cisco Integrated Services Virtual Router (Cisco ISRv). You can upgrade an existing Cisco ISRv router to Cisco Catalyst 8000V. For additional details, refer to Upgrade the Cisco IOS XE Software on page 160.

For Cisco Catalyst 8000V software package support and guest VM compatibility with Cisco NFVIS, see Release Notes for Cisco Enterprise Network Function Virtualization Infrastructure Software, Release 26.1.x.

**Note**

Support for the NFVIS environment in Cisco Catalyst 8000V ends with the Cisco IOS XE 26.1 release.

### Supported upgrade paths

See these lists to view all the supported upgrade paths for:

- Autonomous mode
  - 16.12.x > 17.4 C8000V
  - 17.2.x > 17.4 C8000V

    - 17.3.x > 17.4 C8000V

- Controller mode
    - 16.12.2 ISRV > 16.12.4 ISRV > 17.4 C8000V
    - 16.12.3 ISRV > 16.12.4 ISRV > 17.4 C8000V
    - 16.12.4 ISRV > 17.4 C8000V
    - 17.1.1 ISRV > 17.3.x ISRV > 17.4 C8000V
    - 17.2.1 ISRV > 17.2.2 ISRV > 17.4 C8000V
    - 17.2.2 ISRV > 17.4 C8000V
    - 17.3.x ISRV > 17.4 C8000V

**Guidelines and notes**

- You cannot downgrade from Cisco Catalyst 8000V to Cisco ISRV.
- To upgrade from Cisco ISRV to Cisco Catalyst 8000V, you must use Cisco ISRV version 16.12.4, 17.2.3, or 17.3.2. Devices running other versions cannot be upgraded to Cisco Catalyst 8000V.
- The **install add file bootflash:c8000v-universalk9.XXX.bin activate commit** command is not supported. when upgrading from Cisco IOS XE 17.1.x or an earlier release to Cisco IOS XE 17.4.x. To upgrade Cisco ISRV, copy the `c8000v-universalk9.XXX.bin` file to `bootflash:`, in the Configuration folder. Then, use the **write memory** command to copy the configuration and start the upgrade process.
- If you are an existing Cisco CSR1000V user running Cisco IOS XE 16.12.3 release or earlier, and want to upgrade to Cisco Catalyst 8000V, you cannot upgrade by using the Web UI. You must first upgrade to Cisco CSR1000V releases 16.12.4, 17.2.3, or 17.3.2 before you upgrade to Cisco Catalyst 8000V.
- When you upgrade a Cisco ISRv instance to Cisco Catalyst 8000V in the controller mode, first upgrade to Cisco IOS XE 17.3.1 and later releases or 16.12.4 and later releases.
- All licensing information is retained after you upgrade to Cisco Catalyst 8000V.

# 9 Install Cisco Catalyst 8000V in OpenStack Environment

**Topics:**


- Install Cisco Catalyst 8000V in OpenStack Environment


Explains Cisco Catalyst 8000V deployment in an OpenStack environment, detailing the necessary prerequisites, image registration, flavor creation, and the specific steps required to launch and configure the virtual router instance.

# Install Cisco Catalyst 8000V in OpenStack Environment

Provides information on Cisco Catalyst 8000V installation in OpenStack through the OpenStack dashboard GUI, a Heat template, or the OpenStack command-line interface.

An OpenStack environment is a cloud computing infrastructure platform that

- virtualizes and manages physical data center resources, such as compute, storage, and networking,
- supports deployment of private and public clouds to provide users on-demand access to resources, and
- enables management, scaling, and automation of virtual machines and services.

### Cisco Catalyst 8000V and OpenStack

OpenStack works with Cisco Catalyst 8000V by:

- providing a cloud infrastructure to host it as a virtual machine (VM), and
- managing the underlying physical hardware resources and using virtualization to create the virtual machine that the Cisco Catalyst 8000V software runs on.

Install and boot Cisco Catalyst 8000V on OpenStack Train, starting with Cisco IOS XE Release 17.7.1. OpenStack Train is the 20th version of the open-source cloud infrastructure software and acts as a hypervisor manager. Manage this deployment through the OpenStack dashboard through the GUI, a Heat template, or the OpenStack command-line interface (CLI).

See the sections in this chapter to view the installation workflows for each of these methods.

### Features supported

This list provides the features that are supported in the Cisco Catalyst 8000V installation in OpenStack:

- IPv6
- CDNA Licensing model
- vNIC hot add and delete in the autonomous mode

## Installation requirements

This section specifies the requirements for installing Cisco Catalyst 8000V in OpenStack.

The requirements for installing Cisco Catalyst 8000V in OpenStack are

- Cisco IOS XE releases: 17.6.1 through 17.18.1
- OpenStack Release: Train Release
- Red Hat Enterprise Linux (RHEL) Release 8.2 (Ootpa)
- RHEL OSP version 16.1 (Train)
- CVIM version 4.2
- Virtual disk: 8-GB and 16-GB virtual disks
- Minimum supported profile: 1 vCPU with 4-GB memory and 8-GB or 16-GB virtual disk

## Restrictions for installing in OpenStack

This section lists all the restrictions that are applicable for installing Cisco Catalyst 8000V in OpenStack.

See this list for the restrictions and limitations that are applicable when installing Cisco Catalyst 8000V in OpenStack.

- Console URLs generated by OpenStack heat-deployments are subject to a token time-to-live (TTL) with a default setting of 10 minutes. Any NoVNC URLs that you use expire after this default time. This expiration can occur more quickly under certain conditions, such as when booting up low-profile instances or when using different setups.

  To overcome this limitation, use the built-in Instance VNC console in the portal. Or, use the **virsh console** command on the compute node to access the console.

## Installing Cisco Catalyst 8000V in OpenStack

Provides information on installing Cisco Catalyst 8000V in OpenStack along with the installation requirements and supported installation methods.

When you install Cisco Catalyst 8000V in OpenStack, OpenStack provides a cloud infrastructure to host Cisco Catalyst 8000V as a virtual machine (VM).

### Summary

The components in this installation are:

- OpenStack: this open-source cloud platform manages a pool of computing, storaging, and network resources.
- Virtualization layer: OpenStack relies on virtualization technologies to create a virtual machine (VM) environment from the physical hardware.
- Cisco Catalyst 8000V: the virtual router operates when OpenStack supplies virtualized hardware resources such as CPU, memory, and network interfaces.

### Workflow

You can install Cisco Catalyst 8000V by using:

- The OpenStack GUI. To learn how to do this, see Install the VM by using the GUI on page 101.
- The heat template. To learn how to perform this installation, see Install the VM using a Heat template on page 103.
- The CLI. You can create a VM by running the **openstack server create** command in the OpenStack CLI. For more information, see https://docs.openstack.org/python-openstackclient/train/cli/command-objects/server.html#server-create.

#### Install the VM by using the GUI

Perform these steps if you want to create a VM in OpenStack by using the GUI.

Perform these steps if you want to create a VM in OpenStack by using the GUI. You can access the dashboard after deployment when you use this installation method.

#### Procedure

1. On the OpenStack portal, click **Images**, and choose the image that you want to launch.

   Alternatively, you can also click **Instances** and then **Launch Instance**.

2. In the left pane, click **Details** and specify these details:
   - **Instance Name**: Enter a name for your instance.

- **Description**: Enter a description for your instance. This field is optional.
- **Availability Zone**: This field specifies the logical partitioning of the cloud. Enter **Nova** in this field.
- **Count**: Enter the number of instances that you want to create. Increase the count to create multiple instances with the same settings.

3. Click **Next**, and click **Source** from the left pane.
4. In the left pane, click **Source** and configure these settings.
   a) From the **Select Boot Source** drop-down field, choose either **Image**, **Instance Snapshot**, **Volume**, or **Volume Snapshot**.

      The **Source** option specifies the template used to create an instance. You can use an image, a snapshot of an instance (image snapshot), a volume, or a volume snapshot (if enabled). You can also choose to use persistent storage by creating a new volume.

   a) To delete the Volume when you delete the instance, from the **Delete Volume on Instance Delete** field, choose **Yes**.
5. In the left pane, click **Flavor**. Choose an option based on your memory and storage requirements and click **Next**.
6. In the left pane, click **Networks** and choose a network to connect the Cisco Catalyst 8000V VM with the servers in that network. This option is also required if you want to set up a topology.

   **Note**

   You can use the **Network Ports** drop-down list and choose a NIC if you want to select an SRIOV port to be attached to the VM.

7. In the left pane, click **Security Groups** and choose the security groups to launch your instance in. A default security group is also available.
8. In the left pane, click **Configuration** and perform these actions:
   a) Copy and paste the user data in the **Customization Script** field.

      The following is a sample user data configuration script:

```
hostname c8kv-ios_cfg
                        license smart enable
                        username lab privilege 15 secret lab
                        ip domain name cisco.com
                        interface GigabitEthernet1
                        ip address 10.0.0.5 255.255.255.0
                        no shut
                        exit
                        ip route 0.0.0.0 0.0.0.0 10.0.0.1
                        line vty 0 4
                        login local
                        exit
```

   b) Alternatively, you can upload an XML file or the `iosxe_config.txt` file to provide user data or configuration data. Click **Choose File** and browse to your XML or .txt file.

> **Note**
>
> For detailed information on configuring day zero settings, see Day 0 Configuration on page 106.

c) Select the **Configuration Drive** check box, and click **Next**.

9. Click **Launch Instance** to launch your instance.

> **Note**
>
> You must copy the ciscosdwan_cloud_init.cfg file to bootflash when you switch from the autonomous mode to the controller mode.

### Install the VM using a Heat template

Perform this task to install the OpenStack VM using a Heat template.

Perform this task to install the OpenStack VM using a Heat template.

Heat templates in OpenStack allows you to create OpenStack resources such as instances, volumes, security groups. This template specifies the infrastructure for your cloud application in the form of text files and enables you to automate the deployment of infrastructure, services, and applications.

#### Procedure

1. Log in to the OpenStack portal.
2. From the menu options on top, click **Project**.
3. Click **Orchestration** and select **Stack**.
4. In the **Stacks** window, click **Launch Stack** .
5. From the **Template Source** drop-down list, choose **File**, **URL**, or **Direct Input**, based on how you want to provide the template and click **Next**.

   If you chose the **File** option, click the **Choose File** option, browse to the location where you have saved your template file, upload this file.
6. In the **Stack Name** field, specify the name for the stack.
7. To enable rollback, select the **Rollback on Failure** check box.
8. Enter a password for the admin in the **Password for user “admin”** field.
9. Click **Launch**.

   After the launch is complete, in the **Stacks** window, the system displays a **Create Complete** message in the **Status** column.

# 10 Day 0 Configuration

**Topics:**


- Day 0 Configuration
- Day 0 Configuration Using .txt or .xml Files
- Day 0 Configuration for OVF Deployments
- Day 0 Configuration Using config-drive
- Day 0 Configuration Using Custom Data
- Day 0 Configuration in Controller Mode
- Verify Router Configuration
- Frequently Asked Questions


Explains the methods for Day 0 bootstrap configuration of Cisco Catalyst 8000V, detailing how to automate the initial setup using configuration files, virtual CD-ROMs, or cloud-init custom data across various deployment environments.

# Day 0 Configuration

Specifies the information to configure the Day 0 settings for Cisco Catalyst 8000V in autonomous mode.

Day 0 configuration is the initial, one-time setup of a device, performed after installing the device, to make it reachable and ready for management.

Cisco Catalyst 8000V supports Cisco IOS XE, Cisco SD-Routing, and Cisco IOS XE SD-WAN functionalities. Access the Cisco IOS XE functionalities by booting the instance in the autonomous mode. To access and use the Cisco SD-Routing and Cisco SD-WAN functionalities, boot your instance in the SD-Routing and controller modes, respectively.

The autonomous mode is the default mode in which a Cisco Catalyst 8000V instance boots up. To perform the Day 0 configuration in the autonomous mode, see the sections in this document.

See Install and Upgrade for Cisco Catalyst 8000V Controller Mode to deploy the Cisco Catalyst 8000V instance in the controller mode.

**Note**

If the system is unable to detect any of the following four parameters – OTP, UUID, VBOND, ORG, the device boots in the autonomous mode.

## Bootstrap support across hypervisors and clouds

This section describes how Cisco Catalyst 8000V supports bootstrap across various hypervisors and cloud platforms in autonomous mode.

This table describes how Cisco Catalyst 8000V supports bootstrap across various hypervisors and cloud platforms in autonomous mode.

| Hypervisor | iosxe_config.txt on CD-ROM | ovf-env.xml on CD-ROM | OVA installation | Config-drive format | Custom data | User data |
|---|---|---|---|---|---|---|
| VMware | Yes | Yes | Yes | Yes | No | No |
| KVM | Yes | Yes | No | Yes | No | No |
| AWS | No | No | No | No | Yes | Yes |
| Microsoft Azure | No | No | No | No | Yes | Yes |
| GCP | No | No | No | Yes | Yes | Yes |

## Feature support for Day 0 configuration

View details in this section to learn which features are supported by each Day 0 configuration method.

View details in this section to learn which features are supported by each Day 0 configuration method.

| | iosxe_config.txt on CD-ROM | ovf-env.xml on CD-ROM | OVA installation | Config-drive format | Custom data | User data |
|---|---|---|---|---|---|---|
| Raw configuration copy and paste | Yes | Yes | No | Yes | Yes | Yes |
| Availability of specific configuration fields | No | Yes | Yes | Yes | Yes | Yes |
| GUI availability | No | No | Yes | No | No | No |
| Guestshell bootstrapping | Yes; via manual IOS configuration | Yes; via manual IOS configuration | No | Yes; via manual IOS configuration | Yes | Yes; via manual IOS configuration |

- Public clouds provide a single input mechanism for bootstrapping a VM. On the device side, each cloud supports three input formats: custom-data, user-data, and SDWAN (using the ciscosdwan_cloud_init.cfg file from vManage). For example, in AWS, you can provide the bootstrap information in any of these formats to the instance at launch via the EC2 user data text box or the File Upload option. Cisco Catalyst 8000V then determines and processes the configuration information that you provided.
- The custom-data and user-data columns in the table refer to the bootstrapping input formats and not the cloud native bootstrap input mechanisms for which they were originally named. All the public clouds support both the formats, but the custom-data format is more mature and is the recommended option for most applications.
- For private clouds, you can perform the bootstrap configuration by providing a configuration file in the `iosxe_config.txt` format or the `ovf-env.xml` format. You must upload the configuration file to the VM during Cisco Catalyst 8000V installation through an attached CD-ROM.

## Prerequisites for Day 0 configuration

This section lists all the prerequisites you must know before you perform the Day 0 configuration.

This section lists all the prerequisites you must know before you perform the Day 0 configuration.

- To deploy the Cisco Catalyst 8000V instance in controller mode, generate the bootstrap config file from vManage. Rename the generated config file to `ciscosdwan_cloud_init.cfg`. Use this same file for the device to automatically bootup in the Controller mode and register to vManage.

  Do not manually edit the automatically generated config file from vManage. If you edit the file, the controller could go out of sync, and the device might not boot up successfully on first power-on.

## Limitations for Day 0 configuration

This section provides all the limitations for Day 0 configuration.

The items in this list provides all the limitations for Day 0 configuration.

- If you use the PayG licensing model, you cannot perform a mode switch as controller mode does not support the PayG licensing model.
- Only the autonomous mode supports Dual-IOSd.
- Images without payload encryption and NO-LI images are not supported in the controller mode.

- Switching modes between controller and autonomous modes after onboarding and mode determination causes loss of configuration.
- When you switch from the autonomous mode to the controller mode or vice versa, Cisco Federal Licensing and Smart Licensing registration does not work. You must reregister for the licenses to work.
- When deploying a Cisco Catalyst 8000V VM through GUI, network interfaces may be added in an order that does not match the creation sequence. The numbering is determined by the driver name and PCI address, so Day 0 configuration may be assigned incorrectly for some interfaces. If this occurs, configure the affected network interfaces manually after deployment.

### Select the bootstrapping mechanism

Before you perform the Day 0 configuration, read this section to select the bootstrapping mechanism you want to use.

Decide which bootstrapping mechanism to use before you configure Day 0 settings. The available mechanisms for device configuration are:

- The GUI tool: If you have installed Cisco Catalyst 8000V on VMware, and you chose an OVA deployment, you can perform the configuration by using the OVA deployment wizard. The wizard supports bootstrap-specific fields, so you do not need to manually create a bootstrap configuration file.
- .txt file/.xml file: If you are in a private cloud and want to configure Day 0 settings with IOS commands, choose the iosxe_config.txt file. This method lets you collect the CLIs to apply, paste them into a file, and provide it to the VM as a CD-ROM.
- Custom data: When you deploy Cisco Catalyst 8000V on AWS, Microsoft Azure, or GCP, use the custom-data formatted bootstrap configuration. This method is more functional and flexible than using user-data. Configuration using user-data is intended for users with an established user-data deployment.

## Day 0 Configuration Using .txt or .xml Files

Provides the high-level workflow and the procedures to perform the bootstrap configuration by providing a iosxe_config.txt file or an ovf-env.xml file.

During installation of a new, out-of-box device, you can provide bootstrap configuration to boot the device in autonomous mode.

#### Summary

In a private cloud such as KVM environment, you can perform the bootstrap configuration by providing a iosxe_config.txt file or an ovf-env.xml file. You can gather the configuration commands you want to apply via the CLI, paste them into a file, and provide the file to the VM as a CD-ROM. Depending on the hypervisor environment, the data is then used for the bootstrap configuration.

#### Workflow

These are the stages of performing the Day 0 configuration using .txt or .xml files.

1. Create the bootstrap file.
2. Configure the bootstrap file properties.

### Create the bootstrap file

Complete these steps to create a bootstrap configuration file.

A bootstrap configuration file in .txt or .xml format lets you provide the Day 0 configuration for your device simply and flexibly. Perform this procedure when you create the virtual machine in hypervisors such as KVM.

**Procedure**

1. Perform one of these two steps:
   - Create the `iosxe_config.txt` file. Make sure the file includes each IOS conf t command on a separate line.
   - Create the ovf-env.xml file and configure the desired properties.

   **Note**

   To know more about the individual properties in the .xml file, see Bootsrap properties.

2. To convert the .xml or the .txt file to a consumable form for the virtual machine, create a disk image from the file using the following command:

   **Example**

   ```
   mkisofs -l -o /my/path/c8000v_config.iso <configuration_filename>
   ```

3. Mount the **c8000v_config.iso** as an additional disk during creation of the Cisco Catalyst 8000V virtual machine.

## Bootstrap properties

Describes the individual bootstrap properties used to create the **ovf-env.xml** file.

The table describes the individual bootstrap properties used to create the **ovf-env.xml** file.

**Table 9: Bootstrap Properties**

| Property | Description |
|---|---|
| console | Configures the console mode. Possible values include auto, virtual, serial. |
| domain-name | Domain name of the router. |
| enable-scp-server | Enables the IOS SCP feature. |
| enable-ssh-server | Enables remote login using SSH and disables remote login via Telnet. Requires that the login user name and password are set. |
| hostname | The host name of the router. |

| Property | Description |
|---|---|
| ios-config | Enables execution of a Cisco IOS command.<br>To execute multiple commands, use multiple instances of ios-config, with a number appended to each instance. For example, ios-config-1, ios-config-2. The commands are executed in numerical order according to the appended number.<br>Example<br>`ios-config-1="username cisco priv 15 pass ciscoxyz"`<br>`ios-config-2="ip scp server enable"`<br>`ios-config-3="ip domain lookup"`<br>`ios-config-4="ip domain name cisco.com”` |
| license | Configures the license technology level that is available when the Cisco Catalyst 8000V instance boots. |
| login-password | The login password for the router. |
| login-username | The user name for the router. |
| mgmt-interface | Designates the management interface for the Cisco Catalyst 8000V instance. The format must be GigabitEthernetx or GigabitEthernetx.xxx. |
| mgmt-ipv4-addr | The management gateway address/mask in the IPv4 format for the GigabitEthernet0 management interface. |
| mgmt-ipv4-gateway | The IPv4 management default gateway address. If you're using DHCP, enter **dhcp** in the field. |
| mgmt-ipv4-network | Configures the IPv4 Network (such as “192.168.2.0/24” or “192.168.2.0 255.255.255.0”) that the management gateway should route to. If this value is not specified, the default route (0.0.0.0/0) is used. |
| mgmt-vlan | Configures the dot1Q VLAN interface. Requires the management interface to be configured using the GigabitEthernetx.xxx format. |
| pnsc-agent-local-port | (Optional) Configures the Cisco Prime Network Services Controller service agent SSL port on the local Cisco Catalyst 8000V to receive policies from the service manager.<br>This setting is used if you plan to remotely manage the Cisco Catalyst 8000V using the Cisco Prime Network Services Controller. |
| pnsc-ipv4-addr | Configures the IP address of the Cisco Prime Network Services Controller.<br>This setting is used if you plan to remotely manage the Cisco Catalyst 8000V instance using the Cisco Prime Network Services Controller. |
| pnsc-shared-secret-key | Configures the Cisco Prime Network Services Controller shared secret key for the Cisco Prime Network Services Controller agent to set the SSL certificate from the controller.<br>This setting is used if you plan to remotely manage the Cisco Catalyst 8000V instance using the Cisco Prime Network Services Controller. |
| privilege-password | Configures the password for privileged (enable) access. |
| resource-template | Configures the Resource Template. Possible values include default, service_plane_medium, and service_plane_heavy. |

## Sample iosxe_config.txt file

This file is an example iosxe_config.txt for use with OpenStack and other environments.

This is a sample ios_xe config.txt file.

```
hostname ultra-ios_cfg
license smart enable
username lab privilege 15 password lab
ip domain name cisco.com
crypto key generate rsa modulus 3072
interface GigabitEthernet1
ip address 10.0.0.5 255.255.255.0
no shut
exit
ip route 0.0.0.0 0.0.0.0 10.0.0.1
line vty 0 4
 login local
exit
```

**Note**

From Cisco IOS XE Release 26.1.1, the RSA modulus must be at least 3072 bits.

### Sample iosxe_config.txt File for OpenStack environment

This configuration file serves as an example for use in OpenStack environments.

```
hostname c8kv-ios_cfg
license smart enable
username lab priv 15 secret lab
ip domain name cisco.com
interface GigabitEthernet1
ip address 10.0.0.5 255.255.255.0
no shut
exit
ip route 0.0.0.0 0.0.0.0 10.0.0.1
line vty 0 4
login local
exit
```

## Sample ovf-env.xml file

Sample ovf-env.xml file

Here is a sample ovf-env.xml Day 0 configuration file.

```
<?xml version="1.0" encoding="UTF-8"?>
<Environment
   xmlns:oe="http://schemas.dmtf.org/ovf/environment/1">
    <PropertySection>
        <Property oe:key="com.cisco.c8000v.license.1" oe:value="security"/>
        <Property oe:key="com.cisco.c8000v.console.1" oe:value="serial"/>
```

```
<Property oe:key="com.cisco.c8000v.config-version.1" oe:value="1.0"/>
        <Property oe:key="com.cisco.c8000v.domain-name.1" oe:value=""/>
        <Property oe:key="com.cisco.c8000v.enable-scp-server.1"
oe:value="False"/>
        <Property oe:key="com.cisco.c8000v.enable-ssh-server.1"
oe:value="False"/>
        <Property oe:key="com.cisco.c8000v.hostname.1" oe:value="lab"/>
        <Property oe:key="com.cisco.c8000v.license.1" oe:value="ax"/>
        <Property oe:key="com.cisco.c8000v.login-password.1" oe:value=""/>
        <Property oe:key="com.cisco.c8000v.login-username.1" oe:value="lab"/>

        <Property oe:key="com.cisco.c8000v.mgmt-interface.1"
oe:value="GigabitEthernet1"/>
        <Property oe:key="com.cisco.c8000v.mgmt-ipv4-addr.1"
oe:value="172.25.223.251/25"/>
        <Property oe:key="com.cisco.c8000v.mgmt-ipv4-gateway.1"
oe:value="172.25.223.129"/>
        <Property oe:key="com.cisco.c8000v.mgmt-ipv4-network.1" oe:value=""/>

        <Property oe:key="com.cisco.c8000v.mgmt-vlan.1" oe:value=""/>
        <Property oe:key="com.cisco.c8000v.pnsc-agent-local-port.1"
oe:value=""/>
        <Property oe:key="com.cisco.c8000v.pnsc-ipv4-addr.1" oe:value=""/>
        <Property oe:key="com.cisco.c8000v.pnsc-shared-secret-key.1"
oe:value=""/>
       <Property oe:key="com.cisco.c8000v.privilege-password.1" oe:value=""/>

        <Property oe:key="com.cisco.c8000v.remote-mgmt-ipv4-addr.1"
oe:value=""/>
        <Property oe:key="com.cisco.c8000v.resource-template.1"
oe:value="service_plane_medium"/>
        <Property oe:key="com.cisco.c8000v.ios-config-0001" oe:value="logging
 buffered 10000"/>
       <Property oe:key="com.cisco.c8000v.ios-config-0002" oe:value="hostname
 uut-ovf"/>
       <Property oe:key="com.cisco.c8000v.ios-config-0003" oe:value="ip domain
 name cisco.com"/>
        <Property oe:key="com.cisco.c8000v.ios-config-0004" oe:value="crypto
key generate rsa modulus 3072"/>
       <Property oe:key="com.cisco.c8000v.ios-config-0005" oe:value="interface
 GigabitEthernet2"/>
       <Property oe:key="com.cisco.c8000v.ios-config-0006" oe:value="ip address
 10.0.0.5 255.255.255.0"/>
       <Property oe:key="com.cisco.c8000v.ios-config-0007" oe:value="no shut"/>

        <Property oe:key="com.cisco.c8000v.ios-config-0008" oe:value="exit"/>

       <Property oe:key="com.cisco.c8000v.ios-config-0009" oe:value="ip route
 0.0.0.0 0.0.0.0 10.0.0.1"/>
    </PropertySection>
</Environment>
```

> **Note**
>
> From Cisco IOS XE Release 26.1.1, the RSA modulus must be at least 3072 bits.

## Day 0 Configuration for OVF Deployments

Provides the high-level information to configure the Day 0 settings for OVF deployments.

Day 0 configuration for OVF deployments is supported only for VMware platforms through the vCenter UI or the COT tool.

For more information on Day 0 configuration for Cisco Catalyst 8000V running on the ESXi hypervisor, see Deploy the OVA to the VM on page 38.

For more information on Day 0 configuration for VMware deployments using the COT tool, see Deploy the VM Using COT.

## Day 0 Configuration Using config-drive

Provides the steps to configure the Day 0 settings by using the config-drive option.

You can use the **--config-drive** option to specify that the configuration is loaded when Cisco Catalyst 8000V is booting. In this option:

- CD-ROMs and the second hard drive can be used to store the configuration information in the config-drive format.
- The configuration information is available as a file with contents that match the format of either the `iosxe_config.txt` file or the `ovf-env.xml` file.

To use the config drive option for your day zero configuration, set the **--config-drive** option to **`true`**, and specify the name of the configuration file in which you enter the router configuration to be booted.

### Xml or txt file

You can provide the configuration information as an xml file in the ovf-env.xml file format for OVF deployments or as a text file in the iosxe_config.txt file format.

> **Tip**
>
> Cisco recommends that you use only one configuration file type, either the .txt file or the .xml file, and not both.

See the sample configurations for xml and txt file configurations. Use one of the provided configurations to add your configuration file to the filesystem.

```
nova boot c8000v-vm-174 --image c8000v-174 --flavor c8000v.2vcpu.4gb --nic
port-id=6773be11-7b95-48cd-b372-fb8a3cae2b50 --config-drive=true --file
ovf-env.xml=/home/stack/conf_files/ut/ovf-env.xml
```

OR

```
nova boot c8000v-vm-174 --image c8000v-174 --flavor c8000v.2vcpu.4gb --nic
port-id=6773be11-7b95-48cd-b372-fb8a3cae2b50 --config-drive=true --file
iosxe_config.txt=/home/stack/conf_files/ut/iosxe_config.txt
```

The system requires these hard-coded file names for the config-drive settings to boot.

**Using user data**

In certain environments such as OpenStack, use the user_data option to provide the file into the filesystem with the config-drive format. Here is the sample user data for the OpenStack environment:

```
openstack server create "admin-VK-C8KISOSerial-20210917"
--config-drive true
--image c8kv-image-176
--flavor m1.large
--network mgmt-nt
--network prod-nt
--block-device-mapping id=admin-VK-EmptyVolume-SerialTest:type=volume
--user-data userdata.txt
```

# Day 0 Configuration Using Custom Data

Explains how to perform day 0 configuration using custom data which automates the configuration steps and helps connect to on-premises sites.

After you download the Cisco Catalyst 8000V installation files and deploy the image in your environment, you must manually configure the Cisco Catalyst 8000V instance before the device is fully functional.

To automate the configuration steps or to connect to on-premises sites, upload the Cisco Catalyst 8000V custom data or user data to all the supported public and private clouds.

### Summary

The Day 0 bootstrap file enables you to run Cisco IOS XE configuration commands, install Python packages in guestshell, run scripts in guestshell, and provide licensing information to boot the Cisco Catalyst 8000V instance with your chosen technology package.

By uploading the custom data for your cloud service provider or your private cloud, you can automate the Day 0 or bootstrap configuration. Upload or attach a bootstrap configuration file (iosxe_config.txt), or provide the user data to automate these processes and bring the device to a functional state with minimal intervention.

### Workflow

To launch a Cisco Catalyst 8000V instance by using custom data, complete these steps.

1. Configure the IOS configuration property.
2. Configure the scripts property.
3. Configure the scripts credential property.
4. Configure the python package property.
5. Configure the license property.
6. Provide the Day 0 configuration file.

Before you get to each of these steps in detail, refer to the sample bootstrap files at https://github.com/csr1000v/customdata-examples.

## Configure the IOS configuration property

To configure the IOS properties required for your Day 0 setup, use the IOS Configuration property.

To configure IOS properties for your Day 0 setup, use the IOS Configuration property as described in this sample configuration.

```
Section: IOS configuration
hostname C8000V1
interface  GigabitEthernet1
description “static IP address config”
ip address 10.0.0.1 255.255.255.0
interface GigabitEthernet2
description “DHCP based IP address config”
ip address dhcp
```

After the line labeled `Section: IOS configuration`, enter the Cisco IOS XE configuration commands to be run on the Cisco Catalyst 8000V router.

When you run this command, the specified IOS configuration is applied to the Cisco Catalyst 8000V router on Day 0.

> **Note**
>
> Do not use the IOS configuration property to run do-exec **license smart trust idtoken** commands for establishing CSSM trust during Day 0 bootstrap. This workflow is not supported. Configure the Smart Licensing trust token after the router is operational, or use the supported Day 0 script-based licensing workflow.

## Configure scripts property

Configure scripts property for Day 0 set up

The Scripts property helps automate your deployment and enables other automation objectives. To run a Python or Bash script on Day 0 within the guestshell context, provide the public URL and arguments for the script in the Scripts property.

A script must include a shebang (!) character as the first line. This line tells Cisco IOS-XE which script interpreter (Python or Bash) to use for parsing the script. For example, the first line of a python script can contain `#!/usr/bin/env python`, while the first line of a bash script can contain `#!/bin/bash`. This line allows the Python or Bash script to run as executable code in a Linux environment.

When you execute the script, it runs in the guestshell container of the Cisco Catalyst 8000V instance. To access the guestshell container, use the **guestshell** EXEC mode command. For more information on guestshell commands, see the Programmability Configuration Guide.

To configure the Scripts property, follow the format given here:

```
Section: scripts
public_url <arg1> <arg2>
```

In this script, the first line of the property should read `Section: Scripts`.

In the second line of the property, enter the URL of the script and its arguments. The script can be either a Python or a Bash script. The script runs in guestshell during the first boot, after you upload the bootstrap file and create the Cisco Catalyst 8000V instance.

To view more script examples, see the Scripts section in https://github.com/csr1000v/customdata-examples. Also review these two examples.

**Sample configuration**

```
Section: Script
https://raw.githubusercontent.com/csr1000v/customdata-
examples/master/scripts/smartLicensingConfigurator.py --idtoken "<token_string>"
 --throughput <throughput_value>
```

The two lines in the scripts property retrieve the `smartLicensingConfigurator.py` script from the `customdata-examples` repository at the specified URL. The script runs in the guestshell container of the Cisco Catalyst 8000V using the arguments `idtoken` and `throughput`.

**Sample configuration 2**

```
Section: Scripts
ftp://10.11.0.4/dir1/dir2/script.py -a arg1 -s arg2
```

These two lines in the Scripts property retrieve the `script.py` script from the FTP server with the IP address 10.11.0.4, and runs the script with the `./script.py -a arg1 -s arg2` bash command in the guestshell container of the Cisco Catalyst 8000V instance, using arg1 and arg2 as arguments.

**Notes**

If a script in the Scripts property requires a Python package that is not included in the standard CentOS Linux release, you must include information about the Python package in the Python package property. The standard CentOS Linux release that is used by the guestshell is CentOS Linux release 7.1.1503. For more information, see Configure scripts property.

Prior to uploading the bootstrap file and running the bash or python script, Cisco recommends that you test the URL that you intend to use in the Scripts property. You can test the `ftp://10.11.0.4/dir1/dir2/script.py -a arg1 -s arg2` URL by first running the curl software tool to download the script file. In the guestshell, enter the curl command, as illustrated here:

```
curl -m 30 --retry 5 --user username:password
ftp://10.11.0.4/dir1/dir2/script_needs_credentials.py.
```

If the curl command is successful, a copy of the Python script is downloaded, verifying whether the URL is correct.

## Configure the script credentials property

Configure the script credentials property in Day 0 configuration

If you specify an FTP server in the Script property and the server requires user name and password credentials, provide these credentials using the Script credentials property.

If the FTP server allows anonymous access, you do not need to use the Script credentials property.

Configure the Scripts property with a URL and parameters that match those used in the Script credentials property. To configure the Script credentials property, use this format

```
Section: Script credentials
public_url <username> <password>
```

**Sample configuration**

Here's a sample Script credentials configuration:

```
Section: Script credentials
ftp://10.11.0.4/dir1/dir2/script1.py userfoo foospass
```

The second line in the Script credentials property specifies the user name (`userfoo`) and password (`foospass`) credentials for the Python script `script1.py`.

“Include the name of the FTP server as specified in the Scripts property. An example line in the Scripts property is: `ftp://10.11.0.4/dir1/dir2/script1.py -a arg1 -s arg2`. See example 2 in Configuring the scripts property.

## Configure Python package property

Configure Python package property for Day 0 configuration

If a Python package is required by a script in the Scripts property and is not included in the standard CentOS Linux release 7.1.1503, specify the package information in the Python package property. When you add the Python package property to the bootstrap file, the Cisco Catalyst 8000V downloads and installs the required Python package before the script in the Scripts property runs.

> **Note**
>
> Cisco Catalyst 8000V supports only Python3 in guestshell.

### Configuring Python package property

To configure the Python package property, use the format specified here:

```
Section: Python package
package_name [ version ] [ sudo ] { [ pip_arg1 [ ..[ pip_arg9] ] ] }
```

The arguments *version*, **sudo**, and *pip_arg1* to *pip_arg9* are optional. Place the pip command arguments between the “{“ and “}” braces.

If you specify the version argument, the specific version number is downloaded.

If you specify the sudo argument, the package is downloaded as a sudo user.

### Sample configuration for Microsoft Azure

```
Section: Python package
ncclient 0.5.2
```

For example, the second line of the Python package property indicates that *package_name* is ncclient and the *version* is "0.5.2". When you upload the bootstrap file, version 0.5.2 of the ncclient package installs in the guestshell container of Cisco Catalyst 8000V.

```
Section: Python package
c8000v_azure_guestshell 1.1.2 sudo {--user}
```

In this example, the second line of the Python package property specifies that the *package_name* is "c8000v_azure_guestshell" and the *version* is "1.1.2". When the bootstrap file is uploaded, version 1.1.2 of the c8000v_azure_guestshell package is installed in the guestshell container of Cisco Catalyst 8000V. This command is executed as a sudo user: `sudo pip install c8000v_azure_guestshell==1.1.2 --user`.

If you do not specify an argument, `--user` is used as the default argument.

**Sample configuration for GCP**

```
Section: Python package
ncclient 0.5.2
```

In this example, the second line of the Python package property specifies that the *package_name* is "ncclient", and the *version* is "0.5.2". When the bootstrap file is uploaded, version 0.5.2 of the ncclient package is installed in the guestshell container of the Cisco Catalyst 8000V instance.

```
Section: Python package
c8000v_gcp_ha 3.0.0 sudo {--user}
```

In this example, the second line of the Python package property specifies that the *package_name* is "c8000v_gcp_ha", and the *version* is "3.0.0". When the bootstrap file is uploaded, version 3.0.0 of the c8000v_gcp_ha package is installed in the guestshell container of the Cisco Catalyst 8000V instance. This command is executed as a sudo user: `pip3 install c8000v_gcp_ha=3.0.0 --user`.

If you do not specify an argument, `--user` is used as the default argument.

## Configure the license property

Configure the license property to specify the license technology level for Cisco Catalyst 8000V.

Configure the license property to specify the license technology level for Cisco Catalyst 8000V.

**Procedure**

1. Enter the first line of the property: `Section: License`.
2. Enter the second line of the property: **TechPackage:***tech_level* .

   Specifies the tech level of the license.

   There must be no spaces between TechPackage: and tech_level. Valid tech_level values are ax, security, appx, and ipbase.

   Enter the tech_level value in lowercase.

   **Note**

   The License property is used only to specify the license technology package level during Day 0 bootstrap. It does not establish Cisco Smart Software Manager (CSSM) trust and does not process a Smart Licensing ID token. To establish CSSM trust, use the supported Smart Licensing configuration workflow after the router is operational, or use the documented Day 0 script-based Smart Licensing workflow if applicable.

```
Section: License
TechPackage:security
```

## Provide the Day 0 bootstrap file

Provide the Day 0 bootstrap file for Microsoft Azure deployments.

Provide the Day 0 bootstrap file. This file creates a Cisco Catalyst 8000V VM by executing the CLI commands for Microsoft Azure deployments.

**Procedure**

1. Run this command in the CLI.

```
az vm create --name C8000V-name --resource-group resource-group { [ arg1 [
..[ arg9] ] ] } --custom-data bootstrap-file
```

For further information on the **az vm create** command, see: https://docs.microsoft.com/en-us/cli/azure/vm?view=azure-cli-latest#az-vm-create.

**Example**

```
az vm create -n c8000V-VM-Name -g MyResourceGroup --image
cisco:cisco-c8000V-1000v:16_6:16.6.120170804 --data-disk-sizes-gb 8
--availability-set myAvlSet --nics nic1 nic2 nic3 nic4 --admin-username
azureuser --admin-password "+Cisco123456" --authentication-type password -l
westus --size Standard_DS4_v2 --custom-data bootstrap.txt.
```

When you execute this command, a Cisco Catalyst 8000V VM is created. The router is configured using the commands in the bootstrap.txt file.

2. Use the **Cisco C8000V Settings** option to provide the custom data bootstrap config file.

For more information on managing Linux VMs, see Tutorial: Create and Manage Linux VMs with the Azure CLI 2.0.

## Verify the custom data configuration in Microsoft Azure

Verify the custom data configuration in Microsoft Azure

After you upload the Day 0 bootstrap file, the VM is created and configuration commands are executed. Run the commands in this task to verify the configuration of each property.

**Procedure**

1. To help determine if the license property worked, in Cisco IOS XE CLI on Cisco Catalyst 8000V, run the **show version** command.

**Example**

```
Router#show version
Cisco IOS XE Software, Version
Copyright (c) 1986-2020 by Cisco Systems, Inc.

Cisco IOS-XE software, Copyright (c) 2005-2020 by cisco Systems, Inc.
All rights reserved.  Certain components of Cisco IOS-XE software are
licensed under the GNU General Public License ("GPL") Version 2.0.  The
software code licensed under GPL Version 2.0 is free software that comes
with ABSOLUTELY NO WARRANTY.  You can redistribute and/or modify such
GPL code under the terms of GPL Version 2.0.  For more details, see the
documentation or "License Notice" file accompanying the IOS-XE software,
or the applicable URL provided on the flyer accompanying the IOS-XE
software.
```

```
ROM: IOS-XE ROMMON

Router uptime is 1 minute
Uptime for this control processor is 7 minutes
System returned to ROM by reload
System image file is "bootflash:packages.conf"
Last reload reason: Unknown reason



This product contains cryptographic features and is subject to United
States and local country laws governing import, export, transfer and
use. Delivery of Cisco cryptographic products does not imply
third-party authority to import, export, distribute or use encryption.
Importers, exporters, distributors and users are responsible for
compliance with U.S. and local country laws. By using this product you
agree to comply with applicable laws and regulations. If you are unable
to comply with U.S. and local laws, return this product immediately.

A summary of U.S. laws governing Cisco cryptographic products may be found
at:
http://www.cisco.com/wwl/export/crypto/tool/stqrg.html

If you require further assistance please contact us by sending email to
export@cisco.com.

License Level: ipbase
License Type: N/A(Smart License Enabled)
Next reload license Level: ipbase

The current throughput level is 250000 kbps


Smart Licensing Status: Registration Not Applicable/Not Applicable

cisco C8000V (VXE) processor (revision VXE) with 2271486K/3075K bytes of
memory.
Processor board ID 9MUG8CATY8R
Router operating mode: Controller-Managed
1 Gigabit Ethernet interface
32768K bytes of non-volatile configuration memory.
8106756K bytes of physical memory.
11530240K bytes of virtual hard disk at bootflash:.

Configuration register is 0x2102

[guestshell@guestshell ~]$ pip3 freeze | grep  gpg==1.10.0
gpg==1.10.0
[guestshell@guestshell ~]$
```

2. To see if errors occurred after running the commands in the scripts property, look at the customdata.log file in the `/home/guestshell/customdata` directory. The *scriptname*.log file stores output sent to STDOUT by the script.
3. To check if the Python property worked, enter the **pip freeze** | **grep***package-name* command to view the currently installed python packages. Search for the package *package-name*that you are looking for.
4. To check if the Cisco IOS XE commands were successful in the IOS Configuration property, enter the **show running-configuration** command.

**Example**

```
Router#show running-config
Building configuration...

Current configuration : 6982 bytes
!
! Last configuration change at 14:34:36 UTC Fri Nov 6 2020 by NETCONF
!
version 17.3
service timestamps debug datetime msec
service timestamps log datetime msec
service password-encryption
! Call-home is enabled by Smart-Licensing.
service call-home
platform qfp utilization monitor load 80
platform punt-keepalive disable-kernel-core
platform console serial
!
hostname Router
!
boot-start-marker
boot-end-marker
!
vrf definition 65528
 !
 address-family ipv4
 exit-address-family
!
no logging buffered
no logging rate-limit
!
aaa new-model
!
aaa authentication login default local
aaa authentication enable default enable
aaa authorization console
aaa authorization exec default local
!
aaa session-id common
fhrp version vrrp v3
!
no ip dhcp use class
!
no ip igmp ssm-map query dns
login on-success log
ipv6 unicast-routing
!
subscriber templating
!
multilink bundle-name authenticated
!
crypto pki trustpoint TP-self-signed-2465303444
 enrollment selfsigned
 subject-name cn=IOS-Self-Signed-Certificate-2465303444
 revocation-check none
 rsakeypair TP-self-signed-2465303444
!
crypto pki trustpoint SLA-TrustPoint
 enrollment pkcs12
 revocation-check crl
!
!
```

```
crypto pki certificate chain TP-self-signed-2465303444
 certificate self-signed 01
  30820330 30820218 A0030201 02020101 300D0609 2A864886 F70D0101 05050030
  31312F30 2D060355 04031326 494F532D 53656C66 2D536967 6E65642D 43657274
  69666963 6174652D 32343635 33303334 3434301E 170D3230 31313036 31343333
  35345A17 0D333031 31303631 34333335 345A3031 312F302D 06035504 03132649
  4F532D53 656C662D 5369676E 65642D43 65727469 66696361 74652D32 34363533
  30333434 34308201 22300D06 092A8648 86F70D01 01010500 0382010F 00308201
  0A028201 0100B02F AD33A0FF 0C50D3F2 D06CFDC6 F3CB73BB 4070D649 E07D16CE
  E6271C90 34E86882 822C8D71 E4BAC29D 85285258 51E748E1 8C9FB2C5 12242A22
  7FB71551 02CB4DBC 64089D2F 8DBB6C4A D3E2F112 8E16E71F FE70D102 F59862A3
  E920E77E 52E62E02 1979F800 3D13601F 27C42F81 483BFB34 697F1C20 3952626A
  CA1F5805 26D50A39 33F264D6 1AD485A0 8EB45882 FC97DCA2 106C8FAD 8CDBC0E6
  FF609188 B4677AB0 FBBE77F2 359EA002 E1A5D37D EA895FF3 92732A2B 63465DFD
  4A2A277C 17E7F720 2007A6B6 A7C7296F D0CD2707 8C7C9690 F86B0642 1BA9F28C
  F729157B 8C472E40 78A4E6BE 70471018 4B62EE36 48193FCA 062DB09F 38BC420B
  687E5866 DFA10203 010001A3 53305130 0F060355 1D130101 FF040530 030101FF
  301F0603 551D2304 18301680 14ABBD00 3D02C6E1 7706FA96 29B037A8 583E7B2E
  69301D06 03551D0E 04160414 ABBD003D 02C6E177 06FA9629 B037A858 3E7B2E69
  300D0609 2A864886 F70D0101 05050003 82010100 40C60BF0 2184CF86 08CACB66
  73E74D63 E87A6661 DC839037 D0DB08D0 33C4993C EC326432 E3573D1B EC3B42AF
  F410BF72 2AAB6D8F 1406B352 FE6B5365 CCA7E094 96980FC7 A4B77A02 49CB8C01
  3EC87F01 58BFEE33 0DA222DB 0A1BA130 0AC01F1F FDBF2085 D41EFA45 7A4C7F5E
  2D004D04 D11433BF 69337D90 117A86ED 2CF57A49 AD7DA227 129E53DF 55E12E03
  4D8E0097 A29DC365 11E8B386 891C310E F19EDF6D D9B3EA1E E26ABDBD EF82D8E9
  B0484E26 C0FC1D71 91B19B70 221E1A1A 090F8EA1 3A5FC4FD A4EF36CD EFD2F1F4
  6056C87D 8A76ED1A 68FB76F5 956C6B50 7EFA9D8C 90EA910F 187EBD13 0BF76E5A
  0B9CE20E AA5927C4 7AD13C28 58C6E920 76E36475
       quit
crypto pki certificate chain SLA-TrustPoint
 certificate ca 01
  30820321 30820209 A0030201 02020101 300D0609 2A864886 F70D0101 0B050030
  32310E30 0C060355 040A1305 43697363 6F312030 1E060355 04031317 43697363
  6F204C69 63656E73 696E6720 526F6F74 20434130 1E170D31 33303533 30313934
  3834375A 170D3338 30353330 31393438 34375A30 32310E30 0C060355 040A1305
  43697363 6F312030 1E060355 04031317 43697363 6F204C69 63656E73 696E6720
  526F6F74 20434130 82012230 0D06092A 864886F7 0D010101 05000382 010F0030
  82010A02 82010100 A6BCBD96 131E05F7 145EA72C 2CD686E6 17222EA1 F1EFF64D
  CBB4C798 212AA147 C655D8D7 9471380D 8711441E 1AAF071A 9CAE6388 8A38E520
  1C394D78 462EF239 C659F715 B98C0A59 5BBB5CBD 0CFEBEA3 700A8BF7 D8F256EE
  4AA4E80D DB6FD1C9 60B1FD18 FFC69C96 6FA68957 A2617DE7 104FDC5F EA2956AC
  7390A3EB 2B5436AD C847A2C5 DAB553EB 69A9A535 58E9F3E3 C0BD23CF 58BD7188
  68E69491 20F320E7 948E71D7 AE3BCC84 F10684C7 4BC8E00F 539BA42B 42C68BB7
  C7479096 B4CB2D62 EA2F505D C7B062A4 6811D95B E8250FC4 5D5D5FB8 8F27D191
  C55F0D76 61F9A4CD 3D992327 A8BB03BD 4E6D7069 7CBADF8B DF5F4368 95135E44
  DFC7C6CF 04DD7FD1 02030100 01A34230 40300E06 03551D0F 0101FF04 04030201
  06300F06 03551D13 0101FF04 05300301 01FF301D 0603551D 0E041604 1449DC85
  4B3D31E5 1B3E6A17 606AF333 3D3B4C73 E8300D06 092A8648 86F70D01 010B0500
  03820101 00507F24 D3932A66 86025D9F E838AE5C 6D4DF6B0 49631C78 240DA905
  604EDCDE FF4FED2B 77FC460E CD636FDB DD44681E 3A5673AB 9093D3B1 6C9E3D8B
  D98987BF E40CBD9E 1AECA0C2 2189BB5C 8FA85686 CD98B646 5575B146 8DFC66A8
  467A3DF4 4D565700 6ADF0F0D CF835015 3C04FF7C 21E878AC 11BA9CD2 55A9232C
  7CA7B7E6 C1AF74F6 152E99B7 B1FCF9BB E973DE7F 5BDDEB86 C71E3B49 1765308B
  5FB0DA06 B92AFE7F 494E8A9E 07B85737 F3A58BE1 1A48A229 C37C1E69 39F08678
  80DDCD16 D6BACECA EEBC7CF9 8428787B 35202CDC 60E4616A B623CDBD 230E3AFB
  418616A9 4093E049 4D10AB75 27E86F73 932E35B5 8862FDAE 0275156F 719BB2F0
  D697DF7F 28
       quit
!
license udi pid C8000V sn 9MUG8CATY8R
diagnostic bootup level minimal
memory free low-watermark processor 69848
```

```
!
!
username admin privilege 15 secret 9
$14$vKLj$yfnFjRidlKJg9.$4obKgKyy4TsoUs0sJ2t3HXPnA3XjYWRBnnYKBwVeJrw
!
redundancy
!
interface Loopback65528
 vrf forwarding 65528
 ip address 192.168.1.1 255.255.255.255
!
```

### Verify the custom data configuration in Google Cloud Platform

Verify the custom data configuration in Google Cloud Platform

After you run the custom data script, the VM is created and the configuration commands are executed. To verify the same, use the commands and scripts mentioned in this task.

**Procedure**

1. Run the **show version** command.

   This command displays information about the device, including details about the license property as used in Cisco IOS XE CLI on the Cisco Catalyst 8000V instance.

2. Check the `customdata.log` file in the `/bootflash/<cloud>/` directory to determine if any errors occurred after executing commands in the scripts property. The *scriptname*.log file contains the output sent to STDOUT by the script.
3. Enter the `pip freeze | grep <package-name>` command in the Guestshell to view the currently installed Python packages and verify whether the Python property worked. In this command, *package-name* refers to the package that you want to locate.
4. To verify the Cisco IOS XE commands in the IOS Configuration property, run the **show running-configuration** command.

## Day 0 Configuration in Controller Mode

Describes Day 0 configuration for Cisco Catalyst 8000V instances running in controller mode.

This section provides information on Day 0 configuration for a Cisco Catalyst 8000V in the controller (SD-WAN) mode.

- Download the `ciscosdwan_cloud_init.cfg` file from Cisco vManage. Use the contents as provided for the Day 0 configuration.
- For a Cisco Catalyst 8000V instance running on Cisco CSP-5000 hypervisor, enter the settings in the **Day Zero Config** screen.

  Maintain the format specified here:

  - Source File Name: Enter the value for this field in the `day0_ciscosdwan_cloud_init.cfg` format.
  - Destination File Name: Enter the value for this field in the `day0-dest-filename /openstack/content/ciscosdwan_cloud_init.cfg` format.

For detailed information on switching to controller mode or bootstrapping your virtual router with Cisco SD-WAN, see Install and Upgrade for Cisco Catalyst 8000V Controller Mode.

## Verify Router Configuration

Describes the verification steps to view the mode and the Day 0 configuration status of an Cisco Catalyst 8000V instance.

To verify whether you've deployed or upgraded to the IOS XE 17.4 or later releases successfully, perform the steps in this task.

**Procedure**

Log in to the CLI and run the **show version** command.

The command displays the version of your instance. The **operating device-mode** parameter displays the mode in which your Cisco Catalyst 8000V instance is running.

This is a sample configuration of an instance running in autonomous mode.

**Example**

```
Device# show version | inc operating
Router operating mode: Autonomous
Device# show platform software device-mode
Operating device-mode: Autonomous
Device-mode bootup status:
------------------------------------
Device# show platform software chasfs r0 brief | inc device_managed_mode
/tmp/chassis/local/rp/chasfs/etc/device_managed_mode : [autonomous]
/tmp/fp/chasfs/etc/device_managed_mode : [autonomous]
Device# show version | inc Last reload
Last reload reason: Enabling autonomous-mode
```

## Frequently Asked Questions

Answers the frequently asked questions related to Day 0 configuration for Cisco Catalyst 8000V.

### I have been using Cisco IOS XE image until now. Which mode should I now choose?

If you have been using the Cisco IOS XE universalk9 image so far, deploy the IOS XE 17.4 image and enter the autonomous mode.

### If I am upgrading to the Cisco Catalyst 8000V 17.4 release, do I need to provide the bootstrap configuration?

If you are an existing non-SD WAN user and are upgrading to the IOS XE 17.4 release (autonomous mode), you can perform the upgrade directly. You do not need to perform Day 0 configuration or custom data configuration again.

If the Cisco Catalyst 8000V instance runs on Microsoft Azure or Google Cloud Platform, the device uses the custom data that you provided during the initial configuration.

If the Cisco Catalyst 8000V instance runs on AWS, the device fetches the custom data from the cloud service provider.

**What happens to my custom data configuration after switching modes?**

The existing configuration data is deleted. You must perform the bootstrap or custom data configuration just as you do for a fresh installation.

**What happens to my custom data after a factory reset?**

When you perform a factory reset, the configuration and files on the disk are erased. The router boots as if newly installed, searches for configuration files in the designated location, and determines the mode with the associated configuration.

**Can I deploy my Cisco Catalyst 8000V instance in any mode with PayG license?**

If you use the PayG licensing model, you cannot deploy the Cisco Catalyst 8000V instance in the controller mode or switch to the controller mode. This mode does not support the PayG licensing model.

# 11 Security-Enhanced Linux

**Topics:**


- Security-Enhanced Linux


Introduces the mandatory access control architecture used to safeguard system integrity and prevent unauthorized resource access. It describes how to configure different security modes and troubleshoot policy violations using system logs.

# Security-Enhanced Linux

Explains the mandatory access control architecture that protects system integrity on Cisco IOS XE platforms. Learn how this mechanism prevents tampering and limits damage from potentially malicious or flawed applications.

Security-Enhanced Linux (SELinux) is a solution that incorporates a strong, flexible Mandatory Access Control (MAC) architecture into Cisco IOS-XE platforms through a combination of a Linux kernel security module and system utilities.

SELinux provides an enhanced mechanism to enforce the separation of information based on confidentiality and integrity requirements. This approach helps prevent tampering and bypassing of application security mechanisms while limiting damage from malicious or flawed applications.

#### Security-Enhanced Linux Modes

SELinux can operate either in **Permissive mode** or **Enforcing mode** when enabled on a system.

- **Permissive mode** : SELinux does not enforce the policy and only generates system logs for any denials caused by policy violations. Operations are logged for resource access policy violations but not denied.
- **Enforcing mode** : SELinux policy is enabled and enforced. It denies resource access based on the access policy rules, and generates system logs.

  SELinux is enabled in **Enforcing mode** by default on supported Cisco IOS XE platforms. In the **Enforcing mode**, any system resource access that does not have the necessary allow policy is treated as a violation, and the operation is denied. The violating operation fails when a denial occurs, and system logs are generated. In **Enforcing mode**, the solution works in access-violation prevention mode.

## Prerequisites

There are no specific prerequisites for this feature.

## Restrictions

There are no specific restrictions for this feature.

## Configure Security-Enhanced Linux in EXEC Mode

Configuring SELinux in EXEC mode denies resource access based on the access policy rules, and generates system logs. Use this example to configure SELinux in EXEC mode:

#### Procedure

---

**set platform software selinux {default | enforcing | permissive}**

Configuring SELinux in EXEC mode denies resource access based on the access policy rules, and generates system logs.

---

#### Example

```
Device# set platform software selinux ?

default     Set SELinux mode to default
enforcing   Set SELinux mode to enforcing
permissive  Set SELinux mode to permissive
```

## Configure Security-Enhanced Linux in CONFIG Mode

Configuring SELinux in CONFIG mode means setting the policy to either **enforcing** or **permissive** state. Use this example to configure SELinux in CONFIG mode:

Use this example to configure SELinux in configuration mode:

**Procedure**

---

**platform security selinux {enforcing | permissive}**

Configuring SELinux in CONFIG mode means setting the policy to either **enforcing** or **permissive** state.

---

**Example**

```
Device(config)# platform security selinux

enforcing   Set SELinux policy to Enforcing mode
permissive  Set SELinux policy to Permissive mode

Device(config)# platform security selinux permissive

Device(config)#
*Oct 20 21:52:45.155: %IOSXE-1-PLATFORM: R0/0:
SELINUX_MODE_PROG: Platform Selinux confinement mode downgraded to permissive!

Device(config)#
```

## SYSLOG message reference

This section covers details about syslog messages for SELinux events.

| Facility-Severity-Mnemonic | %SELINUX-1-VIOLATION |
|---|---|
| Severity-Meaning | Alert Level Log |
| Message | N/A |
| Message Explanation | Resource access was made by the process for which a resource access policy does not exist. The operation was flagged, and resource access was denied. A system log was generated with information that process resource access has been denied. |
| Component | SELINUX |

| Facility-Severity-Mnemonic | %SELINUX-1-VIOLATION |
|---|---|
| Recommended Action | Contact Cisco TAC with the following relevant information as attachments:<br>• The exact message as it appears on the console or in the system<br>• Output of the **show tech-support** command (text file)<br>• Archive of Btrace files from the box using the following command:<br>**request platform software trace archive target <URL>**<br>• Output of the **show platform software selinux** command |

This example shows sample syslog messages:

Example 1:

```
*Nov 14 00:09:04.943: %SELINUX-1-VIOLATION: R0/0: audispd: type=AVC
msg=audit(1699927057.934:129): avc: denied { getattr } for pid=5899 comm="ls"

path="/root/test" dev="rootfs" ino=25839
scontext=system_u:system_r:polaris_iosd_t:s0
tcontext=system_u:object_r:admin_home_t:s0 tclass=file permissive=0
```

Example 2:

```
*Nov 14 00:09:04.947: %SELINUX-1-VIOLATION: R0/0: audispd: t type=AVC
msg=audit(1699927198.486:130): avc: denied { write } for pid=6012 comm="echo"

path="/root/test" dev="rootfs" ino=25839
scontext=system_u:system_r:polaris_iosd_t:s0
tcontext=system_u:object_r:admin_home_t:s0 tclass=file permissive= 0
```

## Verify Security-Enhanced Linux enablement

Use the **show platform software selinux** command to view the SELinux configuration mode:

```
Device# show platform software selinux
========================================
IOS-XE SELINUX STATUS
========================================
SElinux Status :    Enabled
Current Mode :      Enforcing
Config file Mode :  Enforcing
```

## Troubleshoot Security-Enhanced Linux

If there is an instance of an SELinux violation on your device or network, please reach out to Cisco TAC with these details:

- The message exactly as it appears on the console or in the system log. For example:

```
device#request platform software trace archive target
    flash:selinux_btrace_logs
```

- Output of the **show tech-support** command (text file)
- Archive of Btrace files from the box using this command:

  **request platform software trace archive target <URL>**
- Output of the **show platform software selinux** command

# 12 Map Network Interfaces to VM Network Interfaces

**Topics:**


- Map Network Interfaces to VM Network Interfaces


Explains how the Cisco Catalyst 8000V maps its internal Cisco IOS XE network interfaces to the virtual network adapters of the host virtual machine, detailing the interface ordering and mapping logic across different hypervisor environments.

# Map Network Interfaces to VM Network Interfaces

Specifies how to map the logical interface on Cisco Catalyst 8000V with the vNIC and the vNIC MAC address. Also explains how to add and delete network interfaces and the supported interface speed for Cisco Catalyst 8000V.

A network interface is a virtual port on a router that:

- provides entry and exit points for network traffic,
- is mapped by virtualization software to physical network hardware, and
- supports various functions such as WAN, LAN, and management connectivity.

#### Interface numbering

Network interfaces in Cisco Catalyst 8000V routers are functionally similar to those on hardware-based routers, and are named according to the Gigabit Ethernet (GE) numbering, which varies by version.

- The interface port numbering begins at 1 and increases according to the number of supported vNICs. See VM Requirements for more details.
- Gigabit Ethernet interface 0 is not supported.
- Any interface can be designated as a management interface using Day 0 bootstrapping. For details, see Day 0 Configuration.

## Supported network interface speeds

This section specifies all the network interface speeds that are supported for Cisco Catalyst 8000V.

This table specifies all the network interface speeds that are supported for Cisco Catalyst 8000V

| Cisco IOS XE release | Interface speed (Mbps) |
|---|---|
| Cisco IOS XE 17.15.1a and later | Up to 40,000 |
| Cisco IOS XE 17.9.1 to Cisco IOS XE 17.14.x releases | Up to 25,000 |
| Cisco IOS XE 17.4.1 to Cisco IOS XE 17.8.x releases | Up to 10,000 |

## Interface mapping limitations and configurations

See this list to learn the notes and limitations for configuring interface mapping in Cisco Catalyst 8000V.

See this list to view all the guidelines and limitations for configuring and managing interface mapping in Cisco Catalyst 8000V.

- Neither hot add nor hot delete operations for interfaces are supported in controller mode. If you wish to change interface mappings in controller mode, use the **request platform software sdwan config reset** command.
- Removing a vNIC without updating the Cisco Catalyst 8000V network interface configuration may result in configuration mismatches upon reboot. In this situation, logical vNIC names and MAC addresses can be reassigned. As a result, GigabitEthernet interfaces might map to different physical interfaces on the hypervisor.
- After adding or deleting network interfaces on the VM, verify the updated vNIC-to-interface mapping before changing network interface configurations.

**Network interface mapping options in ESXi**

Supported interface mappings in the ESXi hypervisor are:

- Each Cisco Catalyst 8000V router interface maps to a unique host Ethernet interface.
- Multiple Cisco Catalyst 8000V interfaces map to a single ESXi host Ethernet interface.
- Cisco Catalyst 8000V interfaces map directly to a trunk interface on the ESXi vSwitch.

## View interface mapping

Perform these steps to view the interface mapping in Cisco Catalyst 8000V.

Perform these steps to view the interface mapping in Cisco Catalyst 8000V.

Cisco Catalyst 8000V maps the GigabitEthernet network interfaces to the logical virtual network interface card (vNIC) name assigned by the VM. The VM in turn maps the logical vNIC name to a physical MAC address.

When you boot a Cisco Catalyst 8000V instance for the first time, the router interfaces are mapped to the logical vNIC interfaces that were added when the VM was created.

**Procedure**

To view the mapping between the interface on the router with the vNIC and the vNIC MAC address, run the **show platform software vnic-if interface-mapping** command.

**Example**

```
Router# show platform software vnic-if interface-mapping
------------------------------------------------------------------------
Interface Name         Short Name     vNIC Name                 Mac Addr
------------------------------------------------------------------------
GigabitEthernet2        Gi2         eth2 (vmxnet3)           0050.5689.0034
GigabitEthernet1        Gi1         eth1 (vmxnet3)           0050.5689.000b
------------------------------------------------------------------------
```

The vNIC name shown in the display identifies a logical interface that the Cisco Catalyst 8000V instance uses to associate with the interface on the hypervisor. his logical vNIC name does not always correspond to the NIC name configured during VM installation. For example, 'eth1' in the display might not link to 'NIC1' from the VM setup process.

GigabitEthernet0 interface is no longer supported.

## Add network interfaces

This topic provides high-level information on how to add network interfaces to a VM.

This topic provides high-level information about adding network interfaces to a VM. First, add vNICs on the VM to add GigabitEthernet interfaces on Cisco Catalyst 8000V.

To add a vNIC on the VM:

- Add the vNIC through the virtual machine settings that are supported for the hypervisor. For details on adding vNICs, see the individual hypervisor documentation.
- Interface hot add is not supported on Cisco Catalyst 8000V in controller mode. Use the **request platform software sdwan config reset** command to add an interface in this mode.

## Remove a network interface

Refer to this task to use a two-step hot remove procedure to remove a vNIC from the VM.

You might need to use a two-step hot remove procedure to remove a vNIC from the VM, depending on the hypervisor. Check if your hypervisor supports a two-step hot remove by looking for hypervisors with `vNIC Two-Step Hot Remove Support = Yes`.

### Before you begin

Power down the VM before you remove an interface.

### Procedure

1. Run the **clear platform software vnic-if interface GigabitEthernet`interface-number`** command.

   **Example**

   ```
   Router# clear platform software vnic-if interface GigabitEthernet4
   ```

   Clears the interface on the VM.

   Here, the value for interface-number can be any number between 0 and 32.

   > **Note**
   >
   > This command might not be supported on all Cisco IOS XE releases.

2. Remove the vNIC from the hypervisor configuration.

   his step depends on your hypervisor. Refer to your hypervisor's official documentation for instructions on how to remove a vNIC.

3. Reboot the router.

## Configure interface mapping for cloned virtual machines

Perform the steps mentioned in this task on the original VM first before cloning to prevent the interface mapping from becoming mismatched.

Perform the steps mentioned in this task on the original VM first before cloning to prevent the interface mapping from becoming mismatched.

When you first install a Cisco Catalyst 8000V instance, a database that maps the vNIC name to the MAC address is created. This database is used to maintain a persistent mapping between the router interfaces and the vNIC-to-MAC address mapping in case you add or delete vNICs. The interfaces are mapped to the stored Universal Unique Identification (UUID) maintained by VMware.

The mapping between router network interfaces and vNICs applies only to the current VM where the router is installed. After cloning the VM, the stored UUID does not match the new UUID, resulting in interface mappings that do not align with the router configuration.

### Before you begin

Ensure that the original VM includes the number of configured vNICs required on the cloned VM.

**Procedure**

1. Enter the **clear platform software vnic-if nvtable** command on the original VM.

   This command clears the persistent interface database on the original VM and updates the interface mapping to the hypervisor.

2. Reboot the Cisco Catalyst 8000V.
3. On the cloned VM, verify the interface mapping using the **showplatformsoftwarevnic-ifinterface-mapping** command.
4. Configure the router interfaces on the cloned VM accordingly.

   The router configuration on the cloned VM should match that of the original VM.

# 13 Enable VNF Secure Boot

**Topics:**


- VNF Secure Boot
- Guidelines
- Setup secure boot in ESXi environment
- Setup secure boot in KVM environment
- Setup secure boot for NFVIS


Defines the secure boot methodology and provides the process to enable secure boot for guest VMs that prevents loading of software which is not signed with an acceptable digital signature.

## VNF Secure Boot

Defines a secure boot methodology that prevents loading software which is not signed with an acceptable digital signature.

Secure boot is part of the Unified Extensible Firmware Interface (UEFI) standard which ensures that a device boots only using a software that is trusted by the Original Equipment Manufacturer (OEM). The UEFI specification defines a secure boot methodology that prevents loading software which is not signed with an acceptable digital signature. When the device starts, the firmware checks the signature of the boot software and the operating system. If the signatures are valid, the device boots, and the firmware gives the control to the operating system.

When you enable the secure boot feature:

- Malicious software applications and unauthorized operating systems are prevented from loading into the systems during the system startup process.
- Only authorized software applications are allowed to boot up from the device. The software applications that boot up on the device are certified by Cisco.
- A secure compute system ensures that the intended software on the system runs without malware or tampered software.

**View the system boot mode**

Before you enable and set up the secure boot feature, verify the system boot mode and the bootloader version. To do this, run the **show platform software system boot** command.

```
Router#show platform software system boot
Boot mode: EFI
Bootloader version: 2.0
```

## Guidelines

This section specifics all the guidelines and supported secure boot environments for Cisco Catalyst 8000V.

The secure boot environments listed here are supported for Cisco Catalyst 8000V.

- ESXi version 6.5 or higher
- KVM RHEL 7.5 using open stack license
- NFVIS release 3.11 or later

Here are some guidelines for the secure boot feature.

- Only EFI firmware modes support secure boot.
- GRUB2 and new disk partition layout is available.

## Setup secure boot in ESXi environment

Specifies the high-level steps to setup secure boot in ESXi environment.

This task specifies the high-level steps to setup secure boot in ESXi environment.

**Procedure**

1. Create VM using ESXi 6.5 or later version using VM version 13 through 19. To choose the EFI firmware mode, navigate to **VM Options** > **Boot Options** > **Firmware** > **EFI**.

   **Note**

   If you require VM version greater than 19, first boot the C8000V with secure boot using ESXi version 19 or earlier. After the router boots successfully, use vCenter to upgrade the VM compatibility to the required version.

2. Power down the VM after the initial boot and the IOS prompt is complete.
3. Enable the EFI secure boot by navigating to **Edit Settings** > **VM Options** > **Boot Options** > **Secure Boot**.
4. Power up the VM.

   **Important**

   You cannot modify the firmware mode (from BIOS to EFI or vice versa) after you create the VM.

The VNF boots up securely.

## Setup secure boot in KVM environment

This task specifies the high-level steps to set up secure boot in KVM environment.

This task specifies the high-level steps to set up the secure boot functionality in KVM environment.

**Procedure**

1. Create the VM.
2. Power down the VM after the VM is created and the VNF IOS prompt is complete.
3. Install the PK, KEK, and db certificates from the **EFI Firmware** menu and perform a reset.

   To create the custom keys, see Custom Keys for Secure boot. For more information about db certificates, see MicCorUEFCA2011_2011-06-27.crt and MicWinProPCA2011_2011-10-19.crt.

4. Secure boot the VM.

## Setup secure boot for NFVIS

This task specifies the high-level steps to set up the secure boot functionality for NFVIS.

This task specifies the high-level steps to set up the secure boot functionality for NFVIS.

**Procedure**

1. Upgrade to NFVIS 3.11 release or later.
2. Register an Cisco Catalyst 8000V EFI tarball with the NFVIS repository.
3. Create a VM using the registered EFI image.
4. Secure boot the VM.

# 14 Configure VRF Route Sharing

**Topics:**


- Configure VRF Route Sharing


Describes how to configure Virtual Routing and Forwarding (VRF) sharing on Cisco Catalyst 8000V, detailing the methods for route leaking and communication between different VRF instances to enable shared services and reachability.

# Configure VRF Route Sharing

Describes how to configure VRF Route Sharing across VxLAN peers to deploy shared services across the cloud.

VRF Route Sharing is a functionality that allows intentional sharing of route information between separate, isolated Virtual Routing and Forwarding (VRF) instances on a router. This functionality allows traffic to be forwarded between an On-Premise site and a public cloud site.

Configure the VRF Route Sharing functionality to deploy shared services across hybrid clouds. Endpoints on the On-Premise site can consume shared services that run on the public cloud.

#### VRF Route Sharing in Cisco Catalyst 8000V

In a hybrid cloud solution where there is an APIC layer (On-Premise) and a public cloud site, the Cisco Catalyst 8000V instance connects the data centers through Layer-3 boundaries. The Cisco Catalyst 8000V instance has a VRF instance configured with two sets of import and export route-targets. One set of import and export route targets is associated with the BGP EVPN session that uses VXLAN encapsulation and provides L3 routing information in the On-Premise router.

The other set of import and export route targets is associated with the L3VPN BGP neighbour in the service provider network. The Cisco Catalyst 8000V instance enables the L3 traffic movement across the EVPN by stitching the route between the On-Premise site and the service provider network.

The Cisco Catalyst 8000V instance forwards traffic across the EVPN even if the VRFs have the same VTEP IP (VxLAN tunnel endpoint) and RMAC (router MAC address). With this feature, the Cisco Catalyst 8000V instance uses a binding label to setup the routing and forwarding chain.

The Cisco Catalyst 8000V instance shares the L3 prefix with multiple VRFs on the On-Premise site, and the On-Premise site shares its L3 prefixes with the Cisco Catalyst 8000V instance. The APIC layer imports the addresses and the services are thus consumed in the APIC side.

## Prerequisites for VRF Route Sharing

Prerequisites before configuring VRF Route Sharing

Before you configure the VRF Route Sharing functionality to enable the traffic between the ACI and the public cloud, ensure that:

- You configure VRF1 and VRF2 on the vPC pair of ACI.
- VRF3 and VRF4 on the Cisco Catalyst 8000V instance, which peers with VGW, have two RTs for each VRF.
- The Cisco Catalyst 8000V instance imports EVPN routes of VRF1 and VRF2 from ACI into VRF3 and VRF4.
- The IP BGP on the Cisco Catalyst 8000V side redistributes the routes to the gateway in the public cloud.
- The next-hop for routes from ACI is the spine or the border leaf of the ACI.
- There are no overlaps of prefix across the route sharing VRF.
- You advertise the L3 VPN routing to forward the VRF prefixes to the EVPN neighbours. Run the **advertise l2vpn evpn** command and export the stitching RTs to push the native routes towards the EVPN.

## Restrictions for VRF Route Sharing

Limitations for configuring the VRF Route Sharing functionality

Review these restrictions before configuring the VRF Route Sharing functionality in Cisco Catalyst 8000V

- The VRF Sharing functionality supports up to 32 common VRFs and 1000 customer VRF combinations.
- This functionality does not support RT filters.

- VRF Route Sharing only supports IPv4 addresses. This functionality does not support IPv6 addresses.

## Sample topology and use cases

Sample topology and use cases that explain VRF Route Sharing for Cisco Catalyst 8000V

### Sample topology for VRF Route Sharing

Consider this sample topology which explains the VRF Route Sharing functionality in a hybrid cloud. In this sample topology, assume the Cisco Catalyst 8000V instance is deployed on the VM of the public cloud. Site A is an ACI deployment site, while Site B is the public cloud.

Leaf 1 and Leaf 2 form the Virtual Port Channel (vPC) pair for ACI. Both vPCs have different Route Distinguishers (RD). VRF1 and VRF2 are configured on the vPC pair for ACI. For example:

- VRF1 – RT:RT-EVPN-1, prefix:192.168.1.1
- VRF2 – RT:RT-EVPN-2, prefix:192.168.2.2

VRF3 and VRF4 are configured on the Cisco Catalyst 8000V instance. These two VRFs connect to the Voice Gateway (VGW). Each VRF uses a different Route Target (RT). For example:

- VRF3 – RT for EVPN: RT-EVPN-3, RT for IP BGP: RT-3, prefix:192.168.3.3
- VRF4 – RT for EVPN: RT-EVPN-4, RT for IP BGP: RT-4, prefix:192.168.4.4

In this topology, assume the BGP-EVPN fabric is present between the ACI and the Cisco Catalyst 8000V instance in the public cloud. The IP BGP protocol is used between the Cisco Catalyst 8000V instance and a cloud service provider, such as Microsoft Azure. The BGP-EVPN fabric redistributes the stitching routes between the EVPN and the IP BGP.

To enable the traffic flow between the ACI Site and the Public Cloud, both ACI and the Cisco Catalyst 8000V instance need to support VRF Route Sharing.

The Cisco Catalyst 8000V instance must be able to import the EVPN routes of VRF1 and VRF2 from ACI into VRF3 and VRF4. The IP BGP on the Cisco Catalyst 8000V side then redistributes the routes to the VGW in the public cloud.

> **Note**
>
> When the VTEP (VxLAN Tunnel Endpoint) IP and the RMAC (Route MAC addrress) are the same for two leafs, and the VNIC alone differs, theCisco Catalyst 8000V instance can forward the traffic across the tunnel.

### Use cases

Using the same sample topology, here are the use cases for configuring VRF Route Sharing in a Cisco Catalyst 8000V instance:

- When VRF1 and VRF2 can talk to VRF3, but VRF3 and VRF4 cannot talk to each other.

```
vrf definition VRF3
rd 300:1
address-family ipv4
route-target export RT-EVPN-3 stitching
route-target import RT-EVPN-1 stitching
route-target import RT-EVPN-2 stitching
vrf definition VRF4
rd 400:1
address-family ipv4
```

- When VRF1 and VRF2 can talk to VRF3&4, but VRF3 and VRF4 cannot talk to each other.

```
vrf definition VRF3
rd 300:1
address-family ipv4
route-target export RT-EVPN-3 stitching
route-target import RT-EVPN-1 stitching
route-target import RT-EVPN-2 stitching
vrf definition VRF4
rd 400:1
address-family ipv4
route-target export RT-EVPN-4 stitching
route-target import RT-EVPN-1 stitching
route-target import RT-EVPN-2 stitching
```

- When VRF1 and VRF2 can talk to VRF3, but VRF3 and VRF4 can talk to each other.

```
vrf definition VRF3
rd 300:1
address-family ipv4
route-target export RT-EVPN-3 stitching
route-target import RT-EVPN-1 stitching
route-target import RT-EVPN-2 stitching
route-target export RT-3
route-target import RT-4
vrf definition VRF4
rd 400:1
address-family ipv4
route-target import RT-3
route-target export RT-4
```

- When VRF1 and VRF2 can talk to VRF3&4, but VRF3 and VRF4 can talk to each other.

```
vrf definition VRF3
rd 300:1
address-family ipv4
route-target export RT-EVPN-3 stitching
route-target import RT-EVPN-1 stitching
route-target import RT-EVPN-2 stitching
route-target export RT-3
route-target import RT-4
vrf definition VRF4
rd 400:1
address-family ipv4
route-target export RT-EVPN-4 stitching
route-target import RT-EVPN-1 stitching
route-target import RT-EVPN-2 stitching
route-target import RT-3
route-target export RT-4
```

For this use case, the Cisco Catalyst 8000V instance must configure EVPN on both VRF3 and VRF4.

Although IP BGP imports all the routes from VRF3 and VRF4, BGP does not advertise the imported routes of the VRF to the EVPN peer.

> **Note**
>
> Use the **Stitching** keyword in the configuration only when the sharing happens across the EVPN.

## Configure VRF Route Sharing

Perform this configuration to set up VRF Route Sharing in a hybrid cloud.

Perform this configuration to set up VRF Route Sharing in a hybrid cloud.

### Procedure

1. Run the **vrf definition** command to define the VRFs.
2. Create each VRF instance with a unique route distinguisher (RD). For each VRF, specify route-targets for both import and export using the **route-target import** and **route-target export** commands.

   #### Example

   In this sample solution, VRF 1 and VRF 2 (on-premise) can talk to VRF 3 and VRF 4 (in the public cloud). However, VRF3 and VRF4 cannot talk to each other.

   ```
   Router(config)# vrf definition vrf3
   Router(config-vrf)# rd 3:3
   address-family ipv4
   Route-target export 100:3
   Route-target import 100:4
   route-target export 3:3 stitching
   route-target import 1:1 stitching
   route-target import 2:2 stitching
   exit-address-family
   !
   !
   vrf definition vrf4
   rd 4:4
   address-family ipv4
   Route-target import 100:3
   Route-target export 100:4
   route-target export 4:4 stitching
   route-target import 1:1 stitching
   route-target import 2:2 stitching
   exit-address-family
   !
   !
   ```

3. Assign interfaces to the VRFs. This configuration sets up each interface to use the correct VRF, configures the IP settings, and activates the interface.

   ```
   Router(config)# interface BDI100
   no shutdown
   vrf forwarding vrf3
   ip address 10.1.1.1 255.255.255.224
   !
   interface GigabitEthernet4.2
   encapsulation dot1Q 2
   ```

```
vrf forwarding vrf3
ip address 10.4.4.1 255.255.255.224
bridge-domain 100
member vni 10100
!
interface nve1
source-interface loopback0
host-reachability protocol bgp
member vni 10100 vrf vrf3
!
```

4. Configure BGP on the router. Redistribute routes using this configuration.

```
router bgp 100
bgp router-id 10.11.11.11
no bgp default ipv4-unicast
neighbor 192.168.22.22 remote-as 200
neighbor 198.162.22.22 update-source loopback0
neighbor 198.162.22.22 ebgp-multihop 255
address-family ipv4 vrf vrf3
redistribute connected
neighbor 10.0.0.2 remote-as 300
neighbor 10.0.0.2 activate
neighbor 10.0.0.2 send-community both
advertise l2vpn evpn
exit-address-family
!
address-family l2vpn evpn
neighbor 198.162.22.22 activate
neighbor 198.162.22.22 send-community both
exit-address-family
end
```

## Verify VRF Route Sharing

Run the commands in this task to check if the VRF Route Sharing feature is enabled.

Run the commands in this task to check if the VRF Route Sharing feature is enabled.

**Procedure**

1. **show ip bgp l2vpn evpn summary .**

   Provides the BGP summary information for the VRF default address family (L2VPN EVPN).

   **Example**

```
show ip bgp l2vpn evpn summary
BGP router identifier 10.11.11.11, local AS number 100
BGP table version is 8, main routing table version 8
7 network entries using 2408 bytes of memory
......
BGP activity 14/0 prefixes, 16/0 paths, scan interval 60 secs
7 networks peaked at 17:34:38 Aug 14 2019 CST (00:00:26.895 ago)
Neighbor        V           AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down
State/PfxRcd
198.162.22.22   4          200       6       5        4    0    0 00:01:23
```

```
      4
Device#
```

2. **show ip route vrf vrf3 bgp | in binding .**

   Displays IP routing tables for the VRF. If a binding label appears in the output, the configuration is successful and BGP uses the binding label as the next hop.

   **Example**

```
+++ 17:35:05 Minuet(default) exec +++
show ip route vrf vrf3 bgp | in binding
B     10.2.1.0/24 [20/0] via binding label: 0x3000001, 00:00:26
B     10.2.2.0/24 [20/0] via binding label: 0x3000002, 00:00:26
B     192.168.1.0/24 [20/0] via binding label: 0x3000001, 00:00:26
B     192.168.2.0/24 [20/0] via binding label: 0x3000002, 00:00:26
```

# 15 Configure Console Access

**Topics:**


- Configure Console Access


Describes the process of booting Cisco Catalyst 8000V for the first time, including instructions on how to access the console, monitor the boot sequence, and perform the initial login to verify that the virtual router is operational.

# Configure Console Access

Describes the process of booting Cisco Catalyst 8000V for the first time, including instructions on configuring the console access for a Cisco Catalyst 8000V VM.

A console is an interface for direct communication with a device such as a VM.

- In the context of a VM, a console provides managed access to a VM hosted in a cloud envioronment.
- The console provides an interface that works independently of the VM's operating system or network state. This independence makes it an essential tool for diagnosing and fixing critical issues.

Cisco Catalyst 8000V boots when the VM is powered on. Depending on your configuration, you can monitor the installation process using the virtual VGA console or the virtual serial port console.

## Boot the Cisco Catalyst 8000V as the VM

Perform these steps to access and configure Cisco Catalyst 8000V from the serial port on the hypervisor.

### Before you begin

To access and configure Cisco Catalyst 8000V from the serial port on the hypervisor instead of the virtual VGA console, you should provision the VM to use this setting before powering on the VM and booting the router.

#### Procedure

1. Power up the VM. Within 5 seconds of powering on the VM, select either the Virtual Console or the Serial Console. Use the selected console to view the router boot process and access the Cisco Catalyst 8000V CLI.
2. (Optional) Select one of these options:
   - **VirtualConsole**: Choose this option to use the virtual console. If you choose to use the virtual console, the rest of the steps in this procedure do not apply. Cisco Catalyst 8000V boots using the Virtual Console if you do not select any other option within the 5 second timeframe.
   - **SerialConsole**: Choose this option to use the virtual serial port console on the VM.

   The virtual serial port must already be present on the VM for this option to work.

   You can select the console port during the boot process only the first time Cisco Catalyst 8000V boots. For instructions to change the console port access after Cisco Catalyst 8000V has booted for the first time, see Modify the console port access after installation on page 156.

   The Cisco Catalyst 8000V starts the boot process.
3. Use one of these commands to telnet to the VM:
   - **telnet://***host-ipaddress*:*portnumber*
   - **telnet***host-ipaddressportnumber* (from a UNIX xTerm terminal)

   The system first calculates the SHA-1, which may take a few minutes. After the initial installation process completes, the system removes the .iso package file from the virtual CD-ROM and reboots the VM. This enables Cisco Catalyst 8000V to boot normally off the virtual Hard Drive.

> **Note**
>
> The system reboots during first-time installation only.

The boot time for Cisco Catalyst 8000V varies by software release and the hypervisor you use.

4. After booting, the system displays a screen with the main software image and the Golden Image. The highlighted entry will boot automatically in three seconds. Do not select the Golden Image. Wait for the main software image to boot.

> **Note**
>
> Cisco Catalyst 8000V does not include a ROMMON image that is included in many Cisco hardware-based routers. During installation, a backup copy of the installed version is stored in a backup partition. This copy can be selected to boot from in case you upgraded your boot image, deleted the original boot image, or somehow corrupted your disk. Booting from the backup copy is equivalent to booting a different image from ROMMON. For more information on changing the configuration register settings to access GRUB mode, see Access and Use the GRUB Mode on page 188.

You can now enter the router configuration environment by entering the standard commands **enable** and then **configureterminal**.

---

## Post configuration guidelines

See this list for guidelines before you install the licenses after configuring the console access.

See this list for guidelines before you install the licenses after configuring the console access.

- When you boot a Cisco Catalyst 8000V instance for the first time, the mode the router boots in depends on the release version.
- You must install the software license or enable an evaluation license to obtain the supported throughput and features. Depending on the release version, you must enable the boot level or change the maximum throughput level, and reboot Cisco Catalyst 8000V.
- The installed license technology package must match the package level configured with the **licensebootlevel** command. If the license package does not match the setting you have configured, the throughput is limited to 100 Kbps.
- (VMware ESXi only) If you manually created the VM using the .iso file, you need to configure the basic router properties. Use Cisco IOS XE CLI commands, or manually configure the properties in the vSphere GUI.

## Access the VM through the virtual VGA console

Access the VM through the virtual VGA console

When installing the Cisco Catalyst 8000V software image, to access the VM through the virtual VGA console, use the Virtual VGA console setting. You do not need to make additional configuration changes to access the Cisco Catalyst 8000V CLI through the virtual VGA console if one of the condition applies:

- You do not change the console setting during the bootup process.

- You do not add two virtual serial ports to the VM configuration. This is applicable if you are using automatic console detection.

## Access the VM through the virtual serial port

Provides information on accessing the VM through the virtual serial port.

You can access a Cisco Catalyst 8000V instance using the virtual VGA console by default. If you use automatic console detection and two virtual serial ports are present, the Cisco Catalyst 8000V CLI becomes available on the first virtual serial port.

You can also configure the VM to use the serial console, which always attempts to use the first virtual serial port for the Cisco Catalyst 8000V CLI.

- To know how to configure the console access for VMWare ESXi environment, see Configure access to serial console in VMware ESXi on page 154.
- To know how to configure the console access for KVM environments, see Configure the access to serial console in KVM on page 155.

### Configure access to serial console in VMware ESXi

This task specifies how to configure the serial console access in ESXi. Perform these steps using VMware vSphere. For detailed information about vSphere, see the official VMware vSphere documentation.

#### Procedure

1. Power-down the VM.
2. Select the VM and configure the virtual serial port settings.
   a) Choose **Edit Settings** > **Add**.
   b) Choose **Device Type** > **Serial port**.
   c) Click **Next**.
   d) Choose **Select Port Type**.
   e) Select **Connect via Network**, and click **Next**.
3. Select **Select Network Backing** > **Server (VM listens for connection)**.
4. Enter the **Port URI** using the **telnet** ://: *portnumber* syntax.

   Here, *portnumber* is the port number of the virtual serial port.
5. Under the I/O mode, choose the **Yield CPU on poll** option, and click **Next**.
6. Power on the VM and access the virtual serial port console.
7. Configure these security settings for the virtual serial port.
   a) Choose the ESXi host for the virtual serial port.
   b) Click the **Configuration** tab and choose **Security Profile**.
   c) In the **Firewall** area, click **Properties** , and then select the **VM serial port connected over Network** value.

      You can now access the Cisco IOS XE console using the Telnet port URI. When you configure the virtual serial port, the Cisco Catalyst 8000V is no longer accessible from the VM's virtual console.

> **Note**
>
> To use these settings, either the **AutoConsole** option or the **SerialConsole** option in the GRUB menu should be selected during the Cisco Catalyst 8000V bootup. If you have already installed the Cisco Catalyst 8000V software using the virtual VGA console, you must configure either the Cisco IOS XE **platformconsoleauto** command or the Cisco IOS XE **platformconsoleserialcommand** and reload the VM for the console access through the virtual serial port to work.

#### Configure the access to serial console in KVM

Configure the serial console access in a KVM environment by using the KVM console on your server. Perform the steps using the KVM console on your server.

#### Procedure

1. Power off the VM.
2. Click the default **Serial1** device (if it exists) and click **Remove**.

   This removes the default pty-based virtual serial port which would otherwise count as the first virtual serial port.
3. Click **Add Hardware** and select **Serial** to add a serial device.
4. In **Character Device**, choose the **TCP Net Console (tcp)** device type from the drop-down list.
5. In **Device Parameters**, choose the mode from the drop-down list.
6. In the **Host** field, enter 0.0.0.0.

   With this setting, the server accepts telnet connections on any interface.
7. Choose the port from the drop-down list.
8. Choose the **Use Telnet** option.
9. Click **Finish**.

   You can now access the Cisco IOS XE console using the Telnet port URI. For more information, see Open a Telnet session to the console on page 155.

   > **Note**
   >
   > To use these settings, select either the **AutoConsole** option or the **SerialConsole** option in the GRUB menu when you boot the Cisco Catalyst 8000V instance. If you have already installed the Cisco Catalyst 8000V software using the virtual VGA console, you must configure either the Cisco IOS XE **platformconsoleauto** command or the **platformconsoleserial** command. Reload the VM in order for the console access through the virtual serial port to work.

## Open a Telnet session to the console

Perform these steps and use Cisco IOS XE CLI commands to open a Telnet session to the Cisco Catalyst 8000V console on the virtual serial port.

Use Cisco IOS XE CLI commands to open a Telnet session to the Cisco Catalyst 8000V console on the virtual serial port

**Procedure**

1. Telnet to the VM by performing one of these steps:
   - Run the **telnet://***host-ipaddress***:***portnumber* command.
   - From a UNIX terminal, run the **telnet***host-ipaddressportnumber* command.
2. At the Cisco Catalyst 8000V IOS XE password prompt, enter your credentials.

   **Example**

   ```
   User Access Verification
   Password: mypass
   ```

   **Note**

   If no password has been configured, press **Return** .

3. From the user EXEC mode, enter the **enable** command.

   **Example**

   ```
   Router> enable
   ```

4. At the password prompt, enter your system password.

   **Example**

   ```
   Password: enablepass
   ```

5. When the enable password is accepted, the system displays the privileged EXEC mode prompt.

   **Example**

   ```
   Router#
   ```

6. You now have access to the CLI in the privileged EXEC mode. Enter the commands you require to complete your tasks.
7. To exit the Telnet session, use the **exit** or **logout** command.

   **Example**

   ```
   Router# logout
   ```

## Modify the console port access after installation

Modify the console port access after installation by performing these steps.

After the Cisco Catalyst 8000V instance has booted successfully, change the console port access to the router using Cisco IOS XE commands. After you change the console port access, you must reload the router for the change to take effect.

**Procedure**

1. **enable**

   Enables the privileged EXEC mode. Enter your password, if prompted.

   **Example**

   ```
   Router> enable
   ```

2. **configureterminal**

   Enters the global configuration mode.

   **Example**

   ```
   Router# configure terminal
   ```

3. Perform one of these actions:

   - **platformconsolevirtual**: Specifies that the Cisco Catalyst 8000V is accessed through the hypervisor virtual VGA console.
   - **platformconsoleserial**: Specifies that the Cisco Catalyst 8000V is accessed through the serial port on the VM. Use this option only if your hypervisor supports serial port console access.

   **Example**

   ```
   Router(config)# platform console virtual
   ```

   **Example**

   ```
   Router(config)# platform console serial
   ```

4. **end**

   Exits the configuration mode.

   **Example**

   ```
   Router(config)# end
   ```

5. **copysystem:running-confignvram:startup-config**

   Copies the running configuration to the NVRAM startup configuration.

   **Example**

   ```
   Router# copy system:running-config nvram:startup-config
   ```

6. **reload**

Reloads the operating system.

**Example**

```
Router# reload
```

---

### What to do next

Install the Cisco Catalyst 8000V licenses. For instructions about installing and using licenses, refer to Configure Licenses and Throughput for Cisco Catalyst 8000V Edge Software.

# 16 Upgrade the Cisco IOS XE Software

**Topics:**


- Upgrade the Cisco IOS XE Software


Outlines the procedures for upgrading the Cisco Catalyst 8000V software, covering the steps for both Autonomous and Controller modes, image acquisition, and the CLI commands required to perform a successful software update.

# Upgrade the Cisco IOS XE Software

Explains the workflow and steps to perform the Cisco Catalyst 8000V upgrade process by using the installation methods described in this document.

The Cisco Catalyst 8000V virtual router runs on the Cisco IOS XE platform, the same platform that has powered Cisco CSR1000V or Cisco ISRV.

To use the Cisco Catalyst 8000V router, first obtain the software image from the Cisco Software Download page. Obtain the installation files, and begin the installation or upgrade. For additional information about the installation files, see Installation files on page 14.

If you are an existing Cisco CSR1000V or a Cisco ISRV user, download the latest installation file from the Cisco Software Download page. Begin the upgrade process by by using the installation methods described in this chapter.

### Software packaging for Cisco Catalyst 8000V

The software image for Cisco Catalyst 8000V is available as a consolidated package and as optional subpackages. Each consolidated package contains a collection of software subpackages, and each software subpackage is an individual software file that controls a different element or elements of the virtual router. Using a consolidated package, you can upgrade all the individual subpackages with a single software image download.

You can upgrade an individual software subpackage, or upgrade all the software subpackages for a specific consolidated package during a complete upgrade. To run the router using individual subpackages from a consolidated package, download the image from Cisco.com and extract the subpackages.

Upgrading using subpackage consumes less memory than upgrading through a consolidated package. For this reason, upgrading through subpackages is the recommended method, especially for deployments with small footprints.

## Prerequisites before upgrading the Cisco IOS XE software

Prerequisites before upgrading the Cisco IOS XE software

Prerequisites before you upgrade the Cisco IOS XE software:

- Obtain the Cisco Catalyst 8000V software image from the Cisco Software Download page. For instructions on obtaining the installation files, see Download the installation files on page 14.
- Check the version of your hypervisor before performing the upgrade. The upgrade fails if your hypervisor version is not supported by the current version of Cisco IOS XE on Cisco Catalyst 8000V.
- Verify that the VM meets the memory requirements for the Cisco Catalyst 8000V software image. If the upgraded version requires more memory than the previous version, increase the VM's memory allocation before starting the upgrade process.

### Prerequisites with HSECK9 license

If you are upgrading a Cisco CSR1000V or Cisco ISRV router where throughput is greater than 250 Mbps, to Cisco Catalyst 8000V Cisco IOS XE 17.4.1 and later, a High Security (HSECK9) license is required.

If you were running a throughput level greater than 250 Mbps prior to the upgrade, you must purchase an HSECK9 license for service continuity after the upgrade. If an HSECK9 license is not available after upgrade, throughput is restricted to 250 Mbps. If you want to switch to Cisco DNA subscription-based licensing model, you must perform a fresh Cisco Catalyst 8000V deployment.

Depending on your pre-upgrade setup, ensure that you meet the corresponding HSECK9 license requirements before you upgrade.

- If the Cisco CSR1000V or Cisco ISRV is connected to CSSM, ensure that all these requirements are met.
  - Throughput greater than 250 Mbps is part of start-up configuration.

To check start-up configuration, enter the **show running-config** command in privileged EXEC mode. For example:

```
Device# show running-config | include throughput
platform hardware throughput level MB 500
```

- There is a positive balance of the required number of HSECK9 licenses (DNA_HSECK9) in the corresponding Smart Account and Virtual Account in CSSM. No further pre-upgrade action is required. As long as the device is connected to CSSM, on upgrade, the device automatically triggers the HSECK9 request and installs the required Smart Licensing Authorization Code (SLAC).

If throughput is less than or equal to 250 Mbps, an HSECK9 license is not required.

#### Prerequisites with Specific Licensing Reservation

If the Cisco CSR1000V or Cisco ISRV is using Specific License Reservation (SLR), update the SLR authorization code to include an HSECK9 license (DNA_HSECK9) and only then upgrade the device. This ensures uninterrupted throughput after upgrade.

See this example on how to update the SLR authorization code: Example: Smart Licensing (SLR With Throughput >250 Mbps, Without Export-Controlled License) to Smart Licensing Using Policy.

## Restrictions when upgrading the Cisco IOS XE software

Restrictions when upgrading the Cisco IOS XE software

Restrictions when upgrading the Cisco IOS XE software:

- Only a .bin file is applicable for upgrading or downgrading your software. The .iso, .qcow2, and .ova files are used for first-time installations only.
- You can upgrade to a new software version on the same VM only. The procedures do not describe how to install or rehost an existing router running the same or upgraded software version on a different VM.
- If you are upgrading to Cisco Catalyst 8000V, your licenses will continue to function as is. However, if you wish to switch to the CDNA licensing model, you must perform a fresh installation.
- The Cisco Catalyst 8000V router does not support In-Service Software Upgrade (ISSU).
- Cisco Catalyst 8000V does not support L2TP functionality including L2TP client and L2TP Network Server (LNS).
- The system requirements for the x86 hardware might differ from those of the hardware currently running on the router.
- If you have freshly installed Cisco Catalyst 8000V, you cannot downgrade to Cisco ISRV or Cisco CSR1000V. If your previous installation was Cisco CSR1000V and you upgraded to Cisco Catalyst 8000V, you can downgrade to Cisco CSR1000V, but you cannot downgrade to Cisco ISRV.

#### Restrictions when upgrading from Cisco CSR1000V or Cisco ISRV

- In the case of an upgrade from Cisco CSR1000V or Cisco ISRV, the disk partition structure remains the same as the previous version, and the secure object storage functionality is not available.
- You cannot upgrade a Cisco CSR1000V running PCI pass-through to Cisco Catalyst 8000V as Cisco Catalyst 8000V does not support PCI pass-through.
- If you want to upgrade to Cisco Catalyst 8000V from a Cisco CSR1000V or a Cisco ISRV prior to 16.12.x, you must first upgrade your current version to 16.12.x. After completing this step, upgrade to the latest version of Cisco Catalyst 8000V.

- You can upgrade from Cisco CSR1000V to Cisco Catalyst 8000V, or from an older to a newer version of Cisco Catalyst 8000V, only by using N-2 or N-1 to N release upgrade paths. N-1 and N-2 are defined as extended maintenance releases. For example, to upgrade a CSR1000V 17.3.x instance to a Cisco Catalyst 8000V 17.11.1a release, first upgrade to version 17.6.x.
- Upgrading a Cisco ISRV or a Cisco CSR1000V to Cisco Catalyst 8000V does not alter the file system layout nor provide any of the new features such as the Secure Object Store which rely on the file system. You must perform a fresh installation to activate these features.

## Install mode process flow

Install mode process flow for installing and upgrading IOS XE software

The install mode process flow comprises three commands to perform installation and upgrade of software on platforms-**install add**, **install activate**, and **install commit**.

**Table 10: List of install Commands**

| Command | Syntax | Purpose |
|---|---|---|
| **install add** | **install add file**`location:filename.bin` | Copies the contents of the image, package, and SMUs to the software repository. File location may be local or remote. This command does the following:<br>• Validates the file-checksum, platform compatibility checks, and so on.<br>• Extracts individual components of the package into subpackages and packages.conf<br>• Copies the image into the local inventory and makes it available for the next steps. |
| **install activate** | **install activate** | Activates the package added using the **install add** command.<br>• Use the **show install summary** command to see which image is inactive. This image will get activated.<br>• System reloads on executing this command. Confirm if you want to proceed with the activation. Use this command with the **prompt-level none** keyword to automatically ignore any confirmation prompts. |

| Command | Syntax | Purpose |
|---|---|---|
| **install activate auto abort-timer** | **install activate auto-abort timer** `<30-1200>` | The **auto-abort timer** starts automatically, with a default value of 120 minutes. If the **install commit** command is not executed within the time provided, the activation process is terminated, and the system returns to the last-committed state.<br>• You can change the time value while executing the **install activate** command.<br>• The **install commit** command stops the timer, and continues the installation process.<br>• The **install activate auto-abort timer stop** command stops the timer without committing the package.<br>• Use this command with the **prompt-level none** keyword to automatically ignore any confirmation prompts.<br>• This command is valid only in the three-step install variant. |
| **install commit** | **install commit** | Commits the package activated using the **install activate** command, and makes it persistent over reloads.<br>• Use the **show install summary** command to see which image is uncommitted. This image will get committed. |
| **install abort** | **install abort** | Terminates the installation and returns the system to the last-committed state.<br>• This command is applicable only when the package is in activated status (uncommitted state).<br>• If you have already committed the image using the **install commit** command, use the **install rollback to** command to return to the preferred version. |

| Command | Syntax | Purpose |
|---|---|---|
| **install remove** | **install remove {file `<filename>` \| inactive}** | Deletes inactive packages from the platform repository. Use this command to free up space.<br>• file: Removes specified files.<br>• inactive: Removes all the inactive files. |
| **install rollback to** | **install rollback to {base \| label \| committed \| id}** | Rolls back the software set to a saved installation point or to the last-committed installation point. The following are the characteristics of this command:<br>• Requires reload.<br>• Is applicable only when the package is in committed state.<br>• Use this command with the **prompt-level none** keyword to automatically ignore any confirmation prompts.<br>**Note**<br>If you are performing install rollback to a previous image, the previous image must be installed in install mode. Only SMU rollback is possible in bundle mode. |
| **install deactivate** | **install deactivate file <filename>** | Removes a package from the platform repository. This command is supported only for SMUs.<br>• Use this command with the **prompt-level none** keyword to automatically ignore any confirmation prompts. |

**Table 11: List of show Commands**

| Command | Syntax | Purpose |
|---|---|---|
| **show install log** | **show install log** | Provides the history and details of all install operations that have been performed since the platform was booted. |

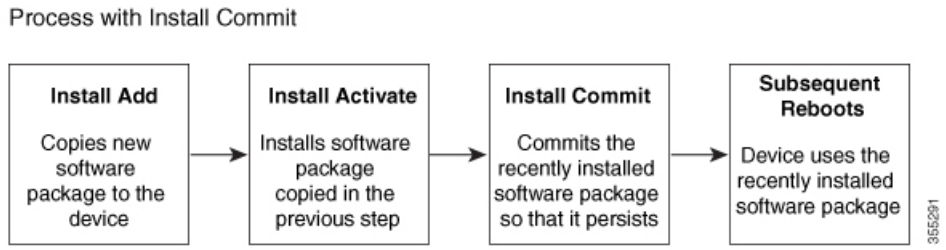
| Command | Syntax | Purpose |
|---|---|---|
| **show install package** | **show install package <filename>** | Provides details about the .pkg/.bin file that is specified. |
| **show install summary** | **show install summary** | Provides an overview of the image versions and their corresponding install states for all the FRUs.<br>• The table that is displayed will state for which FRUs this information is applicable.<br>• If all the FRUs are in sync in terms of the images present and their state, only one table is displayed.<br>• If, however, there is a difference in the image or state information among the FRUs, each FRU that differs from the rest of the stack is listed in a separate table. |
| **show install active** | **show install active** | Provides information about the active packages for all the FRUs.<br>If there is a difference in the information among the FRUs, each FRU that differs from the rest of the stack is listed in a separate table. |
| **show install inactive** | **show install inactive** | Provides information about the inactive packages, if any, for all the FRUs.<br>If there is a difference in the information among the FRUs, each FRU that differs from the rest of the stack is listed in a separate table. |
| **show install committed** | **show install committed** | Provides information about the committed packages for all the FRUs.<br>If there is a difference in the information among the FRUs, each FRU that differs from the rest of the stack is listed in a separate table. |
| **show install uncommitted** | **show install uncommitted** | Provides information about uncommitted packages, if any, for all the FRUs.<br>If there is a difference in the information among the FRUs, each FRU that differs from the rest of the stack is listed in a separate table. |
| **show install rollback** | **show install rollback {point-id \| label}** | Displays the package associated with a saved installation point. |

| Command | Syntax | Purpose |
|---|---|---|
| **show version** | **show version [rp-slot] [installed [user-interface] \| provisioned \| running]** | Displays information about the current package, along with hardware and platform information. |

The install mode process flow comprises three commands to perform installation and upgrade of software on platforms-**install add**, **install activate** , and **install commit**.

This flow chart explains the install process with install commands:

**Figure 1: Process with install commit**




The **install add** command copies the software package from a local or remote location to the platform. The location can be FTP, HTTP, HTTPs, or TFTP. The command extracts individual components of the .package file into subpackages and packages.conf files. It also validates the file to ensure that the image file is specific to the platform on which it is being installed.

The **install activate** command performs the required validations and provisions the packages previously added using the **install add** command. It also triggers a system reload.

The **install commit** command confirms the packages previously activated using the **install activate** command, and makes the updates persistent over reloads.

> **Note**
>
> Installing an update replaces any previously installed software image. At any time, only one image can be installed in a device.

**Boot in install mode**

Install, activate, and commit a software package using a single command (one-step install) or multiple separate commands (three-step install).

You can install, activate, and commit a software package using a single command (one-step install) or multiple separate commands (three-step install).

If the platform is working in bundle mode, the one-step install procedure must be used to initially convert the platform from bundle mode to install mode. Subsequent installs and upgrades on the platform can be done with either one-step or three-step variants.

### One-step Installation or convert from bundle mode to install mode

> **Note**
>
> - All the CLI actions (for example, add, activate, and so on) are executed on all the available FRUs.
> - The configuration save prompt will appear if an unsaved configuration is detected.
> - The reload prompt will appear after the second step in this workflow. Use the **prompt-level none** keyword to automatically ignore the confirmation prompts.
> - If the prompt-level is set to None, and there is an unsaved configuration, the install fails. You must save the configuration before reissuing the command.

Use the one-step install procedure described below to convert a platform running in bundle boot mode to install mode. After the command is executed, the platform reboots in install boot mode.

Later, the one-step install procedure can also be used to upgrade the platform.

This procedure uses the **install add file activate commit** command in privileged EXEC mode to install a software package, and to upgrade the platform to a new version.

#### Procedure

1. **enable**

   **Example**

   ```
   Device>enable
   ```

   Enables privileged EXEC mode. Enter your password, if prompted.

2. **installadd filelocation:***filename* [ **activate commit** ]

   **Example**

   ```
   Device#install add file
   bootflash:c8000be-universalk9.BLD_V177_THROTTLE_LATEST_20211021_031123_V17_7_0_117.SSA.bin
    activate commit
   ```

   Copies the software install package from a local or remote location (through FTP, HTTP, HTTPs, or TFTP) to the platform and extracts the individual components of the .package file into subpackages and packages.conf files. It also performs a validation and compatibility check for the platform and image versions, activates the package, and commits the package to make it persistent across reloads.

   The platform reloads after this command is run.

3. **exit**

   **Example**

   ```
   Device#exit
   ```

Exits privileged EXEC mode and returns to user EXEC mode.

---

Platform is upgraded to new version.

**Three-step install**

> **Note**
>
> - All the CLI actions (for example, add, activate, and so on) are executed on all the available FRUs.
> - The configuration save prompt will appear if an unsaved configuration is detected.
> - The reload prompt will appear after the install activate step in this workflow. Use the **prompt-level none** keyword to automatically ignore the confirmation prompts.

The three-step installation procedure can be used only after the platform is in install mode. This option provides more flexibility and control to the customer during installation.

This procedure uses individual **install add** , **install activate** , and **install commit** commands for installing a software package, and to upgrade the platform to a new version.

**Procedure**

---

1. **enable**

   **Example**

   ```
   Device>enable
   ```

   Enables privileged EXEC mode. Enter your password, if prompted.

2. **installadd filelocation:***filename*

   **Example**

   ```
   Device#install add file
   bootflash:c8000be-universalk9.BLD_V177_THROTTLE_LATEST_20211027_030841_V17_7_0_120.SSA.bin
   ```

   Copies the software install package from a remote location (through FTP, HTTP, HTTPs, or TFTP) to the platform, and extracts the individual components of the .package file into subpackages and packages.conf files.

3. **showinstall summary**

   **Example**

   ```
   Device#show install summary
   ```

   (Optional) Provides an overview of the image versions and their corresponding install state for all the FRUs.

4. **installactivate** [ **auto-abort-timer** *<time>* ]

**Example**

```
Device# install activate auto-abort-timer 120
```

Activates the previously added package and reloads the platform.

- When doing a full software install, do not provide a package filename.
- In the three-step variant, **auto-abort-timer** starts automatically with the **install activate** command; the default for the timer is 120 minutes. If the **install commit** command is not run before the timer expires, the install process is automatically terminated. The platform reloads and boots up with the last committed version.

5. **installabort**

**Example**

```
Device#install abort
```

(Optional) Terminates the software install activation and returns the platform to the last committed version.

- Use this command only when the image is in activated state, and not when the image is in committed state.

6. **installcommit**

**Example**

```
Device#install commit
```

Commits the new package installation and makes the changes persistent over reloads.

7. **installrollback tocommitted**

**Example**

```
Device#install rollback to committed
```

(Optional) Rolls back the platform to the last committed state.

8. **installremove { file***filesystem: filename* **| inactive }**

**Example**

```
Device#install remove inactive
```

(Optional) Deletes software installation files.

- **file** : Deletes a specific file
- **inactive** : Deletes all the unused and inactive installation files.

9. **showinstall summary**

**Example**

```
Device#show install summary
```

(Optional) Displays information about the current state of the system. The output of this command varies according to the **install** commands run prior to this command.

10. **exit**

**Example**

```
Device#exit
```

Exits privileged EXEC mode and returns to user EXEC mode.

---

**Sample upgrade output**

Sample upgrade output from release 17.06.02 to Release 17.07.01

This sample configuration output demonstrates an upgrade from Cisco Catalyst 8000V release 17.06.02 to Release 17.07.01.

```
====================================
Upgrade steps
install add file bootflash:/ c8000v-universalk9.17.07.01a.SPA.bin
install activate
install commit
====================================

Router#show version | inc IOS XE
Cisco IOS XE Software, Version 17.06.02
Router#show version | inc mode
Router operating mode: Autonomous

Router# dir bootflash:*bin*
Directory of bootflash:/*bin*

Directory of bootflash:/

   31  -rw-   832807301   Mar 7 2022 02:07:28 +00:00
c8000v-universalk9.17.07.01a.SPA.bin
5183766528 bytes total (2348220416 bytes free)

Router#install add file bootflash:/c8000v-universalk9.17.07.01a.SPA.bin
install_add: START Mon Mar 7 02:16:30 UTC 2022
install_add: Adding PACKAGE
install_add: Checking whether new add is allowed ....

--- Starting Add ---
Performing Add on Active/Standby
  [1] Add package(s) on R0
  [1] Finished Add on R0
Checking status of Add on [R0]
Add: Passed on [R0]
Finished Add

Image added. Version: 17.07.01a.0.1883
SUCCESS: install_add  Mon Mar  7 02:20:07 UTC 2022
VK5-C8K-8G-1762-1#

Router# show install summary
[ R0 ] Installed Package(s) Information:
State (St): I - Inactive, U - Activated & Uncommitted,
            C - Activated & Committed, D - Deactivated & Uncommitted
```

```
--------------------------------------------------------------------------------
Type  St   Filename/Version
--------------------------------------------------------------------------------
IMG   C    17.06.02.0.2786
IMG   I    17.07.01a.0.1883

--------------------------------------------------------------------------------
Auto abort timer: inactive
--------------------------------------------------------------------------------

====================
install activate
====================

Router# show install summary
[ R0 ] Installed Package(s) Information:
State (St): I - Inactive, U - Activated & Uncommitted,
            C - Activated & Committed, D - Deactivated & Uncommitted
--------------------------------------------------------------------------------
Type  St   Filename/Version
--------------------------------------------------------------------------------
IMG   C    17.06.02.0.2786
IMG   I    17.07.01a.0.1883

Router# install activate
install_activate: START Mon Mar  7 02:50:00 UTC 2022
install_activate: Activating PACKAGE
Following packages shall be activated:
/bootflash/c8000v-rpboot.17.07.01a.SPA.pkg
/bootflash/c8000v-mono-universalk9.17.07.01a.SPA.pkg
/bootflash/c8000v-firmware_nim_xdsl.17.07.01a.SPA.pkg
/bootflash/c8000v-firmware_nim_shdsl.17.07.01a.SPA.pkg
/bootflash/c8000v-firmware_nim_ge.17.07.01a.SPA.pkg
/bootflash/c8000v-firmware_nim_cwan.17.07.01a.SPA.pkg
/bootflash/c8000v-firmware_nim_async.17.07.01a.SPA.pkg
/bootflash/c8000v-firmware_ngwic_t1e1.17.07.01a.SPA.pkg
/bootflash/c8000v-firmware_dsp_sp2700.17.07.01a.SPA.pkg
/bootflash/c8000v-firmware_dreamliner.17.07.01a.SPA.pkg

This operation may require a reload of the system. Do you want to proceed?
[y/n]y
--- Starting Activate ---
Performing Activate on Active/Standby

  [1] Activate package(s) on R0
    --- Starting list of software package changes ---
    Old files list:
      Modified c8000v-firmware_dreamliner.17.06.02.SPA.pkg
      Modified c8000v-firmware_dsp_sp2700.17.06.02.SPA.pkg
      Modified c8000v-firmware_ngwic_t1e1.17.06.02.SPA.pkg
      Modified c8000v-firmware_nim_async.17.06.02.SPA.pkg
      Modified c8000v-firmware_nim_cwan.17.06.02.SPA.pkg
      Modified c8000v-firmware_nim_ge.17.06.02.SPA.pkg
      Modified c8000v-firmware_nim_shdsl.17.06.02.SPA.pkg
      Modified c8000v-firmware_nim_xdsl.17.06.02.SPA.pkg
      Modified c8000v-mono-universalk9.17.06.02.SPA.pkg
      Modified c8000v-rpboot.17.06.02.SPA.pkg
    New files list:
      Added c8000v-firmware_dreamliner.17.07.01a.SPA.pkg
      Added c8000v-firmware_dsp_sp2700.17.07.01a.SPA.pkg
      Added c8000v-firmware_ngwic_t1e1.17.07.01a.SPA.pkg
      Added c8000v-firmware_nim_async.17.07.01a.SPA.pkg
```

```
      Added c8000v-firmware_nim_cwan.17.07.01a.SPA.pkg
      Added c8000v-firmware_nim_ge.17.07.01a.SPA.pkg
      Added c8000v-firmware_nim_shdsl.17.07.01a.SPA.pkg
      Added c8000v-firmware_nim_xdsl.17.07.01a.SPA.pkg
      Added c8000v-mono-universalk9.17.07.01a.SPA.pkg
      Added c8000v-rpboot.17.07.01a.SPA.pkg
    Finished list of software package changes
  [1] Finished Activate on R0
Checking status of Activate on [R0]
Activate: Passed on [R0]
Finished Activate

Send model notification for install_activate before reload
Install will reload the system now!
SUCCESS: install_activate  Mon Mar  7 02:57:34 UTC 2022


==================================
install commit
==================================

Router# show version | inc IOS XE
Cisco IOS XE Software, Version 17.07.01a
Router# show version | inc mode
Router operating mode: Autonomous
Router# show license udi
UDI: PID:C8000V,SN:9JM01Z7G2JH
```

**Upgrade in install mode**

Use either the one-step installation or the three-step installation to upgrade the platform in install mode.

Use either the one-step installation or the three-step installation to upgrade the platform in install mode.

**Downgrade in install mode**

Use the information in this section to downgrade the platform to a previous version.

Use the **install rollback** command to downgrade the platform to a previous version by pointing it to the appropriate image, provided the image you are downgrading to was installed in install mode.

The **install rollback** command reloads the platform and boots it with the previous image.

> **Note**
>
> The **install rollback** command succeeds only if you have not removed the previous file using the **install remove inactive** command.

Alternatively, you can downgrade by installing the older image using the install commands.

**Terminate a software installation**

Terminate the activation of a software package by using the information in this section.

You can terminate the activation of a software package in these ways:

- When the platform reloads after activating a new image, the auto-abort-timer is triggered (in the three-step install variant). If the timer expires before you issue the **install commit** command, the installation process terminates. The platform then reloads and boots with the last committed version of the software image.

Alternatively, use the **install auto-abort-timer stop** command to stop this timer, without using the **install commit** command. The new image remains uncommitted in this process.

- Using the **install abort** command returns the platform to the version that was running before installing the new software. Use this command before issuing the **install commit** command.

## Frequently asked questions

Frequently asked questions for upgrading Cisco Catalyst 8000V from Cisco CSR1000V and Cisco ISRV.

### Can I downgrade from Cisco Catalyst 8000V to Cisco CSR1000V or Cisco ISRv?

You can downgrade from Cisco Catalyst 8000V only if you've upgraded to Cisco Catalyst 8000V from a Cisco CSR1000V or a Cisco ISRv 17.3.x or a later version.

> **Note**
>
> You cannot downgrade to Cisco CSR1000V or Cisco ISRv if you have freshly installed Cisco Catalyst 8000V .

### When I upgrade from a Cisco CSR1000V 16.12.x version or below, will Secure Object Storage be supported?

No, Secure Object Storage is not carried over through upgrades. You must perform a fresh installation or reinstall the VM to enable Secure Object Storage support.

### Will my license need to change when I upgrade from a Cisco CSR1000V or a Cisco ISRv to Cisco Catalyst 8000V ?

When you upgrade to Cisco Catalyst 8000V , the licenses remain the same. However, the licenses move from SL to SLE after the upgrade. If the throughput was <=250M before the upgrade, it is retained as is after the upgrade.

If the throughput was >250M and the device was registered to CSSM, the connection stays intact and the throughput automatically triggers the SLAC installation on the device. The corresponding throughput is set once SLAC is installed.

If the device was not connected to CSSM and throughput was >250M, you must manually install SLAC in the offline mode or configure SLE commands to establish trust with CSSM. Then, configure the throughput to trigger the SLAC installation.

> **Note**
>
> If SLAC is not installed, the throughput remains at 250M.

### Is automation available for the upgrade process?

No, automation is currently not supported for the migration.

### What is the failure mode handling when I perform a downgrade?

When a Cisco CSR1000V image is booting up as a result of a downgrade, the system checks for the partition format. If the partition format does not match the requirements, the bootup is halted. If a Cisco Catalyst 8000V image is booting up as a result of an upgrade or a downgrade, it continues to boot using the existing partition format.

### What is the memory and performance impact after the upgrade?

The size of the Cisco Catalyst 8000V image might be slightly larger, which could affect the overall memory footprint. However, this does not change the overall memory requirements. The minimum required RAM for this image is 4GB. This feature does not impact performance.

## Troubleshooting software upgrade issues

Troubleshooting software upgrade issues

Troubleshooting common software installation upgrade issues.

Run show commands to view installation summaries, logs, and software versions. Use this information to help identify issues with the upgrade.

- **show install summary**
- **show install log**
- **show version**
- **show version running**

If you are unable to troubleshoot using the show commands, run these commands:

- **dir <install directory>**
- **more location: packages.conf**
- **show tech-support install**: This command automatically runs the **show** commands that display information specific to installation.
- **request platform software trace archive target bootflash** < **location**>: This command archives all the trace logs relevant to all the processes running on the system since the last reload, and saves this information in the specified location.

If your issue persists, contact the Cisco TAC to resolve the issue.

# 17 Configure vCPU Distribution

**Topics:**


- Configure vCPU Distribution


Explains how to configure and optimize virtual CPUs (vCPUs) for Cisco Catalyst 8000V, detailing the relationship between vCPU allocation and router performance as well as the steps to modify CPU resources within the hypervisor environment.

# Configure vCPU Distribution

Provides details about the vCPU distribution and templates and which one to choose for your network.

The distribution and allocation of virtual CPUs in Cisco Catalyst 8000V can be configured based on the requirements of the network environment and the deployment needs. You can allocate a specific number of vCPUs to the Cisco Catalyst 8000V virtual machine, depending on the network's throughput and other requirements.

Allocating vCPUs maps the Cisco IOS XE processing threads in the Guest OS to vCPU threads. These threads are then assigned to physical cores. Allocating more vCPUs results in higher throughput and better performance.

Cisco offers several templates with pre-allocated vCPUs. Select the template that best suits your needs, depending on the vCPU allocation for each template.

## vCPU templates

This section specifies the types of vCPU templates that are available and guidelines to choose a template.

To ease the allocation and distribution of vCPUs, Cisco offers several templates. These templates provide a predefined profile that determines how a VM allocates and distributes its vCPU resources. Based on the vCPU allocation defined for each of the templates, you can choose the template best suited for your needs. This list specifies the high-level categorization of the templates that Cisco Catalyst 8000V offers.

- Service Plane Heavy: This template is recommended if you configure Cisco Catalyst 8000V as a service node for AppQoE. This profile is recommended when deploying the New Gen Firewall (NGFW) capabilities such as running Unified Threat Defense (UTD) containers on Cisco Catalyst 8000V. The minimum requirement for the Service Plane Heavy template is 8 vCPUs and 16 GB of memory, and this template includes containers running SNORT. This applies to both SD-WAN and autonomous mode deployments.
- Control Plane Heavy: This template is recommended for use cases such as route reflector deployment and scaling FlexVPN to 10,000 deployments.
- Data Plane Heavy: This is the default template for Cisco Catalyst 8000V. Use this template to achieve maximum forwarding performance.

For detailed allocation information for each template, see the tables in this chapter.

### Guidelines to choose a template

This list provides a few guidelines before choosing a vCPU template.

- Do not create a VM larger than the total number of physical cores of your host. For example, if you have a host system with six cores per NUMA node, it is recommended you size your virtual machines with no more than 6 vCPUs.
- Using a Control Plane Extra heavy template, a service plane app can obtain 1.5 full cores for its operation in certain cases, for example, Wide Area Application Services (WAAS).
- The Control Plane heavy template allocates an extra core to the Control Plane and Service Plane services compared to the Data Plane heavy template. The Control Plane receives one core, and the Service Plane receives another. If there is no Service Plane application, the Control Plane uses both the cores.
- If the configuration output for the Template option reads 'None', the Data Plane heavy template is applied by default. However, this is not applicable for Cisco Catalyst 8000V instances running in the controller mode.
- By default, the Cisco Catalyst 8000V core allocation favors a larger data plane for performance. If there is no Service Plane application, the Control Plane also utilizes the Service Plane's resources.
- When using a Service Plane Heavy template, a Service Plane application such as Snort IPS can use up to 2 full cores for its operation.

### Control Plane Extra Heavy template

This section specifies the vCPU distribution for the Control Plane Extra heavy template.

This table specifies the vCPU distribution for the Control Plane Extra heavy template.

**Table 12: vCPU distribution for Control Plane Extra Heavy template**

| Number of vCPUs | 1 | 2 | 4 | 8 | 16 |
|---|---|---|---|---|---|
| Control Plane | 1/3 | 1/2 | 1 1/2 | 1 1/2 | 6 |
| Service Plane | 1/3 | 1/2 | 1 1/2 | 1 1/2 | 6 |
| Data Plane | 1/3 | 1 | 1 | 5 | 10 |

### Control Plane Heavy template

This section shows the vCPU distribution for the Control Plane Heavy template.

This table shows the vCPU distribution for the Control Plane Heavy template.

**Table 13: vCPU distribution for Control Plane Heavy template**

| Number of vCPUs | 1 | 2 | 4 | 8 | 16 |
|---|---|---|---|---|---|
| Control Plane | 1/3 | 1/2 | 1 | 1 | 4 |
| Service Plane | 1/3 | 1/2 | 1 | 1 | 4 |
| Data Plane | 1/3 | 1 | 2 | 6 | 12 |

### Data Plane Heavy template

This table shows the vCPU distribution for the Data Plane Heavy template.

This table shows the vCPU distribution for the Data Plane Heavy template.

**Table 14: vCPU distribution for Data Plane Heavy template**

| Number of vCPUs | 1 | 2 | 4 | 8 | 16 |
|---|---|---|---|---|---|
| Control Plane | 1/3 | 1/2 | 1/2 | 1/2 | 2 |
| Service Plane | 1/3 | 1/2 | 1/2 | 1/2 | 2 |
| Data Plane | 1/3 | 1 | 3 | 7 | 14 |

### Data Plane Normal template

This section provides information on Data Plane Normal template for vCPU distribution.

Data Plane Normal is a Cisco Catalyst 8000V resource profile which enables Cisco Catalyst 8000V to behave as it did prior to applying a vCPU distribution template. The Data Plane Normal template provides these features:

- This template offers a balanced distribution of resources between the data, control, and service planes.
- You can use the CLI to set this template for an instance. If you create a Cisco Catalyst 8000V VM with the Data Plane Heavy template for vCPU distribution as specified in the ovf-env.xml file, you can override these settings by using the CLI commands in the Data Plane Normal template.

#### Service Plane Heavy template

This table shows the vCPU distribution for the Service Plane Heavy template.

The table shows the vCPU distribution for the Service Plane Heavy template.

**Table 15: vCPU distribution for Service Plane Heavy template**

| Number of vCPUs | 1 | 2 | 4 | 8 | 16 |
|---|---|---|---|---|---|
| Control Plane | 1/3 | 1/2 | 1 | 2 | 8 |
| Service Plane | 1/3 | 1/2 | 1 | 2 | 8 |
| Data Plane | 1/3 | 1 | 2 | 4 | 8 |

#### Service Plane Medium template

This section shows the vCPU distribution for the Service Plane Medium template.

The table shows the vCPU distribution for the Service Plane Medium template.

**Table 16: vCPU distribution for Service Plane Medium template**

| Number of vCPUs | 1 | 2 | 4 | 8 | 16 |
|---|---|---|---|---|---|
| Control Plane | 1/3 | 1/2 | 1 | 1 | 4 |
| Service Plane | 1/3 | 1/2 | 1 | 1 | 4 |
| Data Plane | 1/3 | 1 | 2 | 6 | 12 |

## Manage vCPU distribution

Perform these steps to manage the vCPU distribution in Cisco Catalyst 8000V.

Perform these steps to manage the vCPU distribution in Cisco Catalyst 8000V.

**Procedure**

1. **configure template platform resource** *template*

**Example**

```
Router# configure template
Enter configuration commands, one per line.  End with CNTL/Z.
```

```
Router(config)# platform resource ?
  control-plane-extra-heavy Use Control Plane Extra Heavy template
  control-plane-heavy   Use Control Plane Heavy template
  data-plane-heavy      Use Data Plane Heavy template
  data-plane-normal     Use Data Plane Normal template
  service-plane-heavy   Use Service Plane Heavy template
  service-plane-medium  Use Service Plane Medium template
Router(config)# platform resource service-plane-heavy
```

Allows you to select a resource template for Cisco Catalyst 8000V.

After entering the `platform resource` command, you must reboot the Cisco Catalyst 8000V instance to activate the template.

2. **show platform software cpu alloc**

**Example**

```
Router# show platform software cpu alloc
CPU alloc information:
Control plane cpu alloc: 0-1
Data plane cpu alloc: 2-3
Service plane cpu alloc: 0-1
Template used: CLI-service_plane_heavy
```

Determines which template is being used for vCPU distribution.

> **Note**
>
> The Control plane and the Service plane share cores 0 and 1.

# 18 Manage the device using Web User Interface

**Topics:**


- Manage the device using Web User Interface
- Using Web User Interface for day one setup
- Monitor and Troubleshoot Device Plug and Play (PnP) Onboarding using WebUI


Describes the configuration and functionality of management interfaces on the router detailing the use of the dedicated management port, the Mgmt-intf VRF, and how to isolate management traffic from data traffic.

# Manage the device using Web User Interface

Describes how to use the WebUI for day-one setup, including wizards for configuring LAN and WAN settings, and how to monitor or troubleshoot Plug and Play (PnP) onboarding.

The Web User Interface (Web UI) is an embedded GUI-based device-management tool that provides the ability to provision the device, to simplify device deployment and manageability, and to enhance the user experience. It comes with the default image, so there is no need to enable anything or install any license on the device. You can use WebUI to build configurations, and to monitor and troubleshoot the device without having CLI expertise. This chapter includes these sections:

## Use Basic or Advanced Mode Setup Wizard

To configure the router using the basic or advanced mode setup:

**Procedure**

1. Choose the **Basic Mode** or **Advanced Mode** and click **Go To Account Creation Page** .
2. Enter the username and password. Reenter the password to confirm.
3. Click **Create and Launch Wizard** .
4. Enter the device name and domain name.
5. Select the appropriate time zone from the **Time Zone** drop-down list.
6. Select the appropriate date and time mode from the **Date and Time** drop-down list.
7. Click **LAN Settings** .

## Configure LAN Settings

LAN settings are configured for network devices primarily to manage and control the local network environment effectively. This process provides details on how to configure LAN settings for a device.

**Procedure**

1. Choose the **Web DHCP Pool/DHCP Pool** name or the **Create and Associate Access VLAN** option.
   a) If you choose the Web DHCP Pool, specify the following:
      **Pool Name** –Enter the DHCP Pool Name.
      **Network** –Enter network address and the subnet mask.
   b) If you choose the Create and Associate Access VLAN option, specify the following:
      **Access VLAN** –Enter the Access VLAN identification number. The range is from 1 to 4094.
      **Network** –Enter the IP address of the VLAN.
      **Management Interfaces** –Select the interface and move to the selected list box using the right and left arrows. You can also double click or drag and drop to move the interface to the selected list box.
2. Click **Primary WAN Settings** .

## Configure Primary WAN Settings

The purpose of configuring WAN (Wide Area Network) settings on network devices is to establish and manage the connection between a local network and external networks.

**Procedure**

1. Select the primary WAN type. You con configure Serial, 3G/4G, Ethernet, or Broadband (xDSL) as primary WAN depending on the WAN types supported by the router.
2. Select the interface from the drop-down list.
3. Check the **Get DNS Server info directly from ISP** check box to get the DNS server information directly from the service provider. You can also manually enter the Primary DNS and Secondary DNS.
4. Check the **Get IP automatically from ISP** check box to get the IP address information directly from the service provider. You can also manually enter the IP address and subnet mask.
5. Check the **Enable NAT** check box to enable NAT. It is recommended to enable NAT.
6. Check the **Enable PPPOE** check box to enable PPPoE. If you have enabled PPPoE, select the required authentication mode. The options are: **PAP** and **CHAP** .
7. Enter the user name and password provided by the service provider.
8. Click **Security / APP Visibility WAN Settings** .

## Configure Secondary WAN Settings

The purpose of configuring secondary WAN settings on network devices is to provide an additional WAN connection that can serve as a backup or load-sharing link to the primary WAN. This secondary WAN connection enhances network reliability and availability by allowing failover in case the primary WAN link fails or experiences issues.

**Procedure**

1. Select the secondary WAN type. You con configure Serial, 3G/4G, Ethernet, or Broadband (xDSL) as a secondary WAN depending on the WAN types supported by the router.
2. Select the interface from the drop-down list.
3. Check the **Get DNS Server info directly from ISP** check box to get the DNS server information directly from the service provider. You can also manually enter the Primary DNS and Secondary DNS.
4. Check the **Get IP automatically from ISP** check box to get the IP address information directly from the service provider. You can also manually enter the IP address and subnet mask.
5. Check the **Enable NAT** check box to enable NAT. It is recommended to enable NAT.
6. Check the **Enable PPPOE** check box to enable PPPoE. If you have enabled PPPoE, select the required authentication mode. The options are **PAP** and **CHAP** .
7. Enter the user name and password provided by the service provider.
8. Click **Security / APP Visibility WAN Settings** .

## Configure Secondary WAN Settings

The purpose of configuring secondary WAN settings on network devices is to provide an additional WAN connection that can serve as a backup or load-sharing link to the primary WAN. This secondary WAN connection enhances network reliability and availability by allowing failover in case the primary WAN link fails or experiences issues.

**Procedure**

1. Select the secondary WAN type. You con configure Serial, 3G/4G, Ethernet, or Broadband (xDSL) as a secondary WAN depending on the WAN types supported by the router.
2. Select the interface from the drop-down list.

3. Check the **Get DNS Server info directly from ISP** check box to get the DNS server information directly from the service provider. You can also manually enter the Primary DNS and Secondary DNS.
4. Check the **Get IP automatically from ISP** check box to get the IP address information directly from the service provider. You can also manually enter the IP address and subnet mask.
5. Check the **Enable NAT** check box to enable NAT. It is recommended to enable NAT.
6. Check the **Enable PPPOE** check box to enable PPPoE. If you have enabled PPPoE, select the required authentication mode. The options are **PAP** and **CHAP** .
7. Enter the user name and password provided by the service provider.
8. Click **Security / APP Visibility WAN Settings** .

## Using Web User Interface for day one setup

Perform these steps to use Web User Interface for day one setup.

The purpose of the web user interface (Web UI) for devices is to provide an embedded, graphical user interface that simplifies device management by allowing users to provision, configure, monitor, and troubleshoot the device without requiring command-line interface (CLI) expertise. It enhances the user experience by offering an intuitive, GUI-based tool to build configurations and manage the device efficiently.

**Procedure**

1. Configure the HTTP server. By default, the HTTP server configuration should be present on the device. Ensure the configuration by checking if the **ip http server** and **ip http secure-server** commands are present in the running configuration.

```
Device #
configure terminal
Device (config)#
ip http server
Device (config)#ip http secure-server
```

2. Set up the authentication options to log into Web UI. You can use one of these methods to authenticate:
   a) You can authenicate using local database. To use a local database for Web UI authentication, ensure to have the **ip http authentication local** command in the running configuration. This command is preconfigured on the device. If the command is not present, configure the device as shown in this example:

```
Device #
configure terminal
Device (config)#
ip http authentication local
```

> **Note**
>
> You need a user with privilege 15 to access the configuration screens on Web UI. If the privilege is less than 15, you can access only the Dashboard and Monitoring screens on Web UI.

b) Authenticate using AAA options. To use AAA authentication for Web UI, ensure to configure 'ip http authentication aaa' on the device. Also, ensure that the required AAA server configuration is present on the device.

```
Device #
configure terminal
Device (config)#
ip http authentication local
```

3. Launch the browser. In the address bar, type the IP address of the device. For a secure connection, type https://ip-address.
4. Enter the default username (cisco) and password provided with the device.
5. Click **Log In** .

## Monitor and Troubleshoot Device Plug and Play (PnP) Onboarding using WebUI

This section provides information on monitoring and troubleshooting Device Plug and Play (PnP) Onboarding using WebUI.

A device can be automatically onboarded to Cisco vManage through either Zero Touch Provisioning (ZTP) or the Plug and Play (PnP) process. This section describes the procedure to monitor and troubleshoot device onboarding through the PnP method. This feature on WebUI enables you to monitor and troubleshoot the PnP onboarding process, and also see its real-time status. If this onboarding is stuck or fails, you can terminate the process and onboard your device manually.

- Your device (a computer that can run a web browser) running the WebUI and the device you are onboarding must be connected through an L2 switch port (NIM) on the device.
- The DHCP client-identifier on your device must be set to string "webui".
- Your device must support Cisco SD-WAN Day-0 device onboarding on WebUI.

**Procedure**

1. Enter the controller mode in WebUI

> **Note**
>
> If the device does not have start-up configuration at the time of PnP onboarding, the WebUI is enabled by default on supported devices.

2. On the **Welcome to Cisco SDWAN Onboarding Wizard** page, click **Reset Default Password.**

> **Note**
>
> The default password of your Day-0 device is weak. Therefore, for a secure log in, you must reset the password when you first log in to the device on WebUI. The WebUI configuration is automatically deleted after the device is onboarded successfully. In rare cases where the template configuration for your device on Cisco vManage has the WebUI configuration, it is not deleted even after a successful device onboarding.

3. You are redirected to the Device hardware and software details page. Enter your password and click **Submit.**

   The next page displays the onboarding progress and lists statuses of different components of the PnP Connect Portal and Cisco SD-WAN controllers. If the PnP IPv4 component fails, it indicates that the device PnP onboarding has failed.

   To view and download logs for the onboarding process, click the information icon on the right hand side of the SDWAN Onboarding Progress bar.

4. The next page displays the onboarding progress and lists statuses of different components of the PnP Connect Portal and Cisco SD-WAN controllers. If the PnP IPv4 component fails, it indicates that the device PnP onboarding has failed.
5. If the automated PnP onboarding fails, click **Terminate Automated Onboarding.** This allows you to onboard your device manually.

   The dialogue box appears. To continue with the termination, click **Yes** . It might take a few minutes for the termination to complete.

6. On the Bootstrap Configuration page click **Select File** and choose the bootstrap file for your device. This file can be either a generic bootstrap file (common platform-specific file) or a full configuration bootstrap file that you can download from Cisco Catalyst SD-WAN Manager. This file must contain details such as the vBond number, UUID, WAN interface, root CA and configuration.
7. Click **Upload** .
8. After your file is successfully uploaded, click **Submit.**
9. You can see the SDWAN Onboarding Progress page again with statuses of the Cisco SD-WAN controllers. To open the Controller Connection History table click the information icon on the right hand side of the SDWAN Control Connections bar. In this table you can see the state of your onboarded device. After the onboarding is complete, the state of your device changes to **connect** .

# 19 Access and Use the GRUB Mode

**Topics:**


- Access and Use the GRUB Mode


Explains how to access the GRUB bootloader menu on Cisco Catalyst 8000V, detailing the steps to interrupt the boot sequence for troubleshooting, password recovery, or modifying boot parameters.

# Access and Use the GRUB Mode

Provides information on accessing and using the GRUB mode in Cisco Catalyst 8000V.

The GNU GRand Unified Boot loader (GRUB) is a program which enables you to select the installed operating system or kernel to load at system boot time. The GRUB mode in Cisco Catalyst 8000V supports a subset of configuration register options, similar to the ROM Monitor (ROMMON) options on other Cisco routers. These configuration register options allow you to:

- Force the router to boot into GRUB mode (bootstrap program)
- Select a boot source and the default boot filename
- Recover a lost password

### Configuration register in Cisco Catalyst 8000V

Cisco Catalyst 8000V has a 16-bit configuration register in NVRAM. Each bit can be set to a value of 1 (on or set) or 0 (off or clear). The setting configured for each bit determines the router's behavior at the next power cycle.

This table describes the configuration register bits.

**Table 17: Configuration Register Bit Descriptions**

| Bit number | Hexadecimal | Description |
|---|---|---|
| 00-03 | 0x0000-0x000F | Boot field. The boot field setting determines whether the router loads an operating system and where it obtains the system image. |
| 06 | 0x0040 | Causes the system software to ignore the contents of NVRAM. This setting is useful for password recovery. |

### GRUB mode usage guidelines

Entering the GRUB mode for Cisco Catalyst 8000V running on cloud solutions depends on the console access capabilities of the cloud provider. If the cloud provider offers limited access to the console, you cannot access the GRUB mode for password recovery.

To configure the router to automatically enter GRUB mode upon reboot, use the **0x000** setting.

## Access the GRUB mode

Perform this task to access the GRUB mode.

Perform this task to access the GRUB mode.

### Procedure

1. **enable**

**Example**

```
Router> enable
```

Enables the privileged EXEC mode. Enter your password, if prompted.

2. **config-register 0x0000**

   Access the GRUB mode by setting the configuration register to "0000" (0x0).

   **Example**

```
Router# config-register 0x0000
```

This a sample configuration to access the GRUB mode.

```
Router(config)# config-register 0x0000
GNU GRUB  version 2.02
Minimal BASH-like line editing is supported. For the first word, TAB
lists possible command completions. Anywhere else TAB lists possible
device or file completions. ESC at any time exits.
grub> confreg 0x2102
```

If you enter a question mark at the `grub>` prompt, the system displays the two options available; for viewing the system help or for entering the **config register** command.

## Access the GRUB menu

Access the GRUB menu to select which image to boot from.

The GRUB menu displays the software images loaded on the router. Use the GRUB menu to select which image to boot from.

**Procedure**

1. To access the GRUB menu, enter **ESC** at the GRUB prompt.

   The system displays the GRUB menu with all the images that are loaded on the router.

2. Select the image to boot the router by using the up and down arrow keys.
3. To return to the GRUB prompt, enter the letter **c**.

## Load a new system image from GRUB menu

Perform this task to load a new system image from the GRUB mode.

Perform this task to load a new system image from the GRUB mode.

**Procedure**

1. **dirbootflash:**

   Displays a list of all the files and directories in bootflash memory.

**Example**

```
Router# dir bootflash:
Directory of bootflash:/
3      -rw-   6458388    Dec 18 2020 00:00:58 c8000v.tmp
1580   -rw-   6462268    Dec 18 2020 06:14:02 c8000v-ata
63930368 bytes total (51007488 bytes free)
```

2. **configureterminal**

   Enters the global configuration mode.

   **Example**

   ```
   Router# configure terminal
   Router(config)#
   ```

3. **bootsystembootflash:***system-image-filename* .bin

   Loads the new system image after the next system reload or power cycle.

   **Example**

   ```
   Router(config)# boot system bootflash: c8000v-universalk9.17.04.01a.SPA.bin
   ```

   > **Note**
   >
   > If the new system image is the first file or the only file displayed in the **dirbootflash:** command output, you do not need to perform this step.

4. **dowrite**

   or

   **dowritememory**

   **Example**

   ```
   Router(config)# do write memory
   ```

   > **Note**
   >
   > Entering the **dowrite** or **dowritememory** command updates the list of images available on the bootflash disk.

5. **config-register0x0000**

   Enters the GRUB mode.

   **Example**

   ```
   GNU GRUB  version 2.02
   Minimal BASH-like line editing is supported. For the first word, TAB
   ```

```
lists possible command completions. Anywhere else TAB lists possible
device or file completions. ESC at any time exits.
grub> confreg 0x2102
```

**Example**

> **Note**
>
> If you set the config-register to 0x0000, you should reset it back to the default of 0x2102 for the system to autoboot. If the value is 0x0, the system stops in the GRUB mode.

6. At the **grub>** prompt, enter **ESC** to access the GRUB menu.

   The system displays the GRUB menu with the images that are available to boot.

   **Example**

```
Cisco IOS XE Software, Version 2020-09-17_09.24_kamitch
Cisco IOS Software [Bengaluru], Virtual XE Software
(X86_64_LINUX_IOSD-UNIVERSALK9-M), Experimental Version 17.5.20200916:194029
 [HEAD-/scratch/kamitch/git/polaris-work/boottime1 106]
Copyright (c) 1986-2020 by Cisco Systems, Inc.
Compiled Wed 16-Sep-20 15:45 by kamitch

Cisco IOS-XE software, Copyright (c) 2005-2020 by cisco Systems, Inc.
All rights reserved.  Certain components of Cisco IOS-XE software are
licensed under the GNU General Public License ("GPL") Version 2.0.  The
software code licensed under GPL Version 2.0 is free software that comes
with ABSOLUTELY NO WARRANTY.  You can redistribute and/or modify such
GPL code under the terms of GPL Version 2.0.  For more details, see the
documentation or "License Notice" file accompanying the IOS-XE software,
or the applicable URL provided on the flyer accompanying the IOS-XE
software.

ROM: IOS-XE ROMMON

Router uptime is 18 minutes
Uptime for this control processor is 21 minutes
System returned to ROM by reload
System image file is "bootflash:packages.conf"
```

7. Select the image to boot the router by using the up and down arrow key. To return to the GRUB prompt, enter the letter **c** .
8. Select the .bin file to upgrade the software image on the router to the new version.
9. Press **Enter** to boot the selected image.

   The upgrade process begins.

## Modify the configuration register

This section describes how to modify the configuration register by using the confreg GRUB command.

This section describes how to modify the configuration register by using the **confreg** GRUB command. This command is similar to the **confreg** ROMMON command used on other Cisco hardware routers. Because the router does not include a ROMMON mode, tGRUB command mode provides the equivalent functionality.

You can also modify the configuration register setting from the Cisco IOS XE CLI by using the **config-register** command in global configuration mode.

> **Note**
>
> The modified configuration register value is automatically written into NVRAM. To apply the new value, reset or power-cycle the router.

**Procedure**

---

**confreg** [ *value* ]

**Example**

```
grub> confreg 0x2102
```

Changes the configuration register settings while in GRUB command mode.

- Optionally, enter the new hexadecimal value for the configuration register. Specify a value between 0x0 and 0xFFFF.
- If you do not enter the value, the router prompts you for each bit of the 16-bit configuration register.

---

This sample configuration specifies entering the GRUB mode and using the configuration register. You access the GRUB mode by entering the Cisco IOS XE **config-register** command and specifying the value as “0000”.

```
        Router(config)#
        config-register 0x0000
        GNU GRUB  version 0.97  (638K lower / 3143616K upper memory)
        [ Minimal BASH-like line editing is supported.  For the first word,
TAB
        lists possible command completions.  Anywhere else TAB lists the
possible
        completions of a device/filename.  ESC at any time exits to menu. ]
        grub> help
        [ Minimal BASH-like line editing is supported.  For the first word,
TAB
        lists possible command completions.  Anywhere else TAB lists the
possible
        completions of a device/filename.  ESC at any time exits to menu. ]
        confreg [VALUE]                         help [--all] [PATTERN ...]
        grub>
        confreg
        Configuration Summary
        (Virtual Configuration Register: 0x0)
        enabled are:
        boot: the boot loader
        do you wish to change the configuration? y/n [
        n
        ]:
        ignore system config info? y/n [
```

```
       n
       ]:
       automatically boot default system image? y/n [
       n
       ]:
       Configuration Register: 0x0
       grub> confreg
       Configuration Summary
       (Virtual Configuration Register: 0x0)
       enabled are:
       boot: the boot loader
       do you wish to change the configuration? y/n [
       n
       ]:
       ignore system config info? y/n [n]:
       automatically boot default system image? y/n [n]:
       Configuration Register: 0x42
       grub> confreg 0x2102
       Configuration Register: 0x2102
       grub>
       confreg
       Configuration Summary
       (Virtual Configuration Register: 0x2102)
       enabled are:
       boot: default image
       do you wish to change the configuration? y/n [
       n
       ]:
       grub>
       grub>
       GNU GRUB  version 2.02  (638K lower / 3143616K upper memory)
       ----------------------------------------------------------------------
       0: C8000v - packages.conf
       1: C8000v - c800v-packages-universalk9
       2: C8000v - GOLDEN IMAGE
       ----------------------------------------------------------------------
       Use the ^ and v keys to select which entry is highlighted.
       Press enter to boot the selected OS, or 'c' for a command-line.
       Highlighted entry is 0:
       Booting 'C8000v - packages.conf'
       root (hd0,0)
       Filesystem type is ext2fs, partition type 0x83
       kernel /packages.conf rw root=/dev/ram console=ttyS1,9600 max_loop=64
HARDWARE=
       virtual SR_BOOT=harddisk:packages.conf
       Calculating SHA-1 hash...done
       SHA-1 hash:
       calculated   817e1716:e8e62778:7dd0b806:32db2bdd:13e51407
       expected     817e1716:e8e62778:7dd0b806:32db2bdd:13e51407
       package header rev 1 structure detected
       Calculating SHA-1 hash...done
       SHA-1 hash:
       calculated   d4eaba99:34cbda63:26151233:9d0e9aa4:9c625302
       expected     d4eaba99:34cbda63:26151233:9d0e9aa4:9c625302
       Package type:0x7531, flags:0x0
       [Linux-bzImage, setup=0x2e00, size=0x2c18c00]
       [isord @ 0x7e6d0000, 0x191f000 bytes]
```

## Modify the configuration register settings

This section explains how to modify the configuration register settings using the Cisco IOS XE CLI.

You can change the configuration register settings using either the GRUB or the Cisco IOS XE CLI. This section explains how to modify these settings using the Cisco IOS XE CLI.

**Procedure**

1. Power on the router.
2. If you are asked whether you would like to enter the initial dialog, answer no.

   **Example**

   ```
   Would you like to enter the initial dialog? [yes/no]: no
   ```

   After a few seconds, the system displays the user EXEC prompt (Router>).

3. **enable**

   Enters the privileged EXEC mode. Enter your password, if prompted.

   **Example**

   ```
   Router> enable
   Password:password
   Router#
   ```

4. **configure terminal**

   Enters the global configuration mode.

   **Example**

   ```
   Router# configure terminal
   Enter configuration commands, one per line.
   Edit with DELETE, CTRL/W, and CTRL/U; end with CTRL/Z
   ```

5. **config-register***value* command, where *value* is a hexadecimal number preceded by **0x** :

   Changes the configuration register settings.

   **Example**

   ```
   Router(config)# config-register 0x value
   ```

6. **end**

   Exits the global configuration mode.

7. **copyrunning-configstartup-config**

   Saves the configuration changes to NVRAM.

   The new configuration register settings are saved to NVRAM, but they do not take effect until the next router reload or power cycle.

## Verify the configuration register settings

Verify the configuration register settings

Perform this task to display the configuration register settings that are currently in effect and the settings that will be used at the next router reload.

**Procedure**

Run the **show version** command in privileged EXEC mode.

This command displays the configuration register settings that are currently in effect.

**Example**

```
Configuration register is 0x142 (will be 0x142 at next reload)
```

The configuration register settings are displayed in the last line of the command output.

# 20 Perform a Factory Reset

**Topics:**


- Factory reset in Cisco Catalyst 8000V


Outlines the process for restoring a device to its original functional state by clearing configurations and personal data. Specifies the commands required for secure resets and explains which data is retained or erased during the procedure.

## Factory reset in Cisco Catalyst 8000V

Defines factory reset, which is the process of clearing the current running and start up configuration information on a router, and resetting the router to an earlier, fully functional state.

Factory reset is a process of clearing the current running and start up configuration information on a router, and resetting the router to an earlier, fully functional state.

- The factory reset process uses the **factory-reset all** command.
- The time required for a factory reset on a Cisco Catalyst 8000V instance depends on factors such as the type of storage and the devices present on the router.

**Data erased or retained during factory reset**

This table provides the information retained and erased during a factory reset:

| Command name | Data erased | Data retained |
|---|---|---|
| **factory-reset all** | • Licenses – user installed, and manufacturer provided<br>• Non-volatile random-access memory data<br>• User credentials<br>• Startup configuration<br>• All writable file systems and personal data<br>• ROMMON variable<br>• Persistent storage devices<br>• Any containers running on bootflash | • Critical information, including files that provide access to the router after the reset is complete<br>• The software packages installed before you perform a factory reset<br>• UDI and Smart Licensing files |
| **factory-reset keep-licensing-info** | • License Boot level configuration<br>• Throughput level configuration<br>• Smart license transport type<br>• Smart license URL data | • Real User Monitoring (RUM) Reports (open/unacknowledged license usage report)<br>• Usage reporting details (last ACK received, next ACK scheduled, last/next report push)<br>• Unique Device Identification (UDI) trust codes<br>• Customer policy received from CSSM<br>• SLAC, SLR authorization codes return codes<br>• Factory installed purchase information |

#### Supported scenarios

Perform a factory reset in these scenarios:

- When you want to delete a Cisco Catalyst 8000V instance in a secure manner.
- If the router data is compromised due to a malicious attack, you must reset the router to factory configuration and then reconfigure it for further use.

Factory reset is supported on a Cisco Catalyst 8000V instance running on various platforms including Amazon Web Services, Microsoft Azure, Google Cloud Platform cloud, VMware ESXi, and Hyper-V.

## Prerequisites for factory reset

Lists the prerequisites before performing a factory reset in Cisco Catalyst 8000V.

Before performing a factory reset, ensure:

- A backup of all the software images, configurations and personal data is available before performing the factory reset operation.
- Uninterrupted power supply is available when the factory reset process is in progress.
- The instance has at least 8 GB of memory in the bootflash.

## Restrictions for factory reset

Lists the restrictions for performing a factory reset in Cisco Catalyst 8000V.

This list specifies the limitations while performing a factory reset.

- Any software patches that are installed on the router are not restored after the factory reset operation.
- Restarting the Cisco Catalyst 8000V instance during the factory reset process results in failure.
- If the factory reset command is issued through a Virtual Teletype (VTY) session, the session is not restored after the completion of the factory reset process.

## Perform a Factory Reset

Describes the steps to perform a factory reset in Cisco Catalyst 8000V.

Complete the steps in this section to perform a factory reset.

#### Procedure

1. Log in to a Cisco Catalyst 8000V instance.
2. At the command prompt, run the **factory-reset all** command.

```
factoryreset# factory-reset all
                    The factory reset operation is irreversible for all
operations. Are you sure? [confirm]
                   The following will be deleted as a part of factory reset:

                    1: All writable file systems and personal data
                    2: Licenses
                    3: Configuration
                    4: User Credentials
                    The system will reload to perform a factory reset.
                    Note that any day0 configuration will be applied after
```

```
reload
                    DO NOT STOP OR INTERRUPT THE POWER DURING RESET
                    Are you sure you want to continue? [confirm]Connection
to 172.18.25.29 closed by remote host.
                    Connection to 172.18.25.29 closed.
```

3. Enter confirm to proceed with the factory reset.

   A log file in the bootflash indicates whether the process succeeded.

   > **Note**
   >
   > The duration of the factory reset process depends on the storage type and the cloud service used to deploy the instances.

4. To quit the factory reset process, press the **Escape** key.

---

**What to do next**

After the reset, Smart Licensing configuration is also deleted. You must reconfigure Smart Licensing on the router by using the token ID.

In the connected mode, register your instance for Smart Licensing and use the force option. Use the **license smart register idtoken <token> force** command. The registration process begins.

If you do not use the force option and configure Smart Licensing directly, the license registration fails. See this sample failed registration output.

```
router# show license status
router# show license status
Smart Licensing is ENABLED

Utility:
  Status: DISABLED

Data Privacy:
  Sending Hostname: yes
    Callhome hostname privacy: DISABLED
    Smart Licensing hostname privacy: DISABLED
  Version privacy: DISABLED

Transport:
  Type: Callhome

Registration:
  Status: UNREGISTERED - REGISTRATION FAILED
  Export-Controlled Functionality: NOT ALLOWED
  Initial Registration: FAILED on Feb 15 22:03:29 2019 UTC
    Failure reason: The product
regid.2013-08.com.cisco.C8KV,1.0_1562da96-9176-4f99-a6cb-14b4dd0fa135 and sudi
 containing udiSerialNumber:9XIVK9PIVPK,udiPid:C8000V has already been
registered.

License Authorization:
```

```
  Status: No Licenses in Use

Export Authorization Key:
  Features Authorized:
```

### What happens after factory reset

Describes the processes and verifications that happen after a factory reset in Cisco Catalyst 8000V.

After the factory reset is successfully completed, the router boots up. If the configuration register was set to manually boot from ROMMON before the factory reset process began, the router stops at ROMMON after the reset.

**Important**

If the current boot image is a remote image or is stored in a USB or a NIM-SSD, ensure that you take a backup of the image before starting the factory reset process.

### Verify device details

Factory reset does not change the UDI of the Cisco Catalyst 8000V instance. To verify whether the UDI is the same after the factory reset, run the **factoryreset# show license udi** command before and after the factory reset process.

To check whether Smart Licensing is enabled for your instance, after you configure Smart Licensing, run the **#show license status** command.

**Note**

If you had SLR enabled before you performed the factory reset, use the same license and enter the same license key that you received from the smart agent.

# 21 Bridge Domain Interfaces

**Topics:**


- Information about bridge domain interface
- Configure bridge domain virtual IP interface

# Information about bridge domain interface

Introduces bridge domain interface architecture, detailing restrictions, Ethernet virtual circuits, encapsulation, MAC address assignment, IP protocol support, forwarding mechanisms between Layer 2 and Layer 3, link states, interface statistics, and providing step-by-step configuration and verification instructions including virtual IP interface overview.

A bridge domain interface is a logical interface that

- allows bidirectional flow of traffic between a Layer 2 bridged network and a Layer 3 routed network,
- is identified by the same index as its associated bridge domain,
- represents a Layer 2 broadcast domain through each bridge domain, and
- can be associated only once per bridge domain, allowing just one bridge domain interface per domain.

Bridge domain interfaces support these features:

- IPv4 Multicast
- QoS marking and policing. Shaping and queuing are not supported
- IPv4 VRF
- IPv6 unicast forwarding
- Dynamic routing such as BGP, OSPF, EIGRP, RIP, IS-IS, and STATIC
- Hot Standby Router Protocol (HSRP)
- Virtual Router Redundancy Protocol (VRRP) from IOS XE 3.8.0 onwards.
- IP termination
- Layer 3 VPN termination
- Address Resolution Protocol (ARP), G-ARP, and P-ARP handling
- MAC address assignment

## Restrictions for bridge domain interfaces

These are the restrictions pertaining to bridge domain interfaces.

These are restrictions for bridge domain interfaces

- Only 4096 bridge domain interfaces are supported per system.
- For a bridge domain interface, the maximum transmission unit (MTU) size can be configured between 1500 and 9216 bytes.

### Unsupported Features

Bridge domain interfaces do not support these features:

- PPP over Ethernet (PPPoE)
- Bidirectional Forwarding Detection (BFD) protocol
- QoS
- Network-Based Application Recognition (NBAR) or Advanced Video Coding (AVC)

## Ethernet virtual circuit overview

An Ethernet Virtual Circuit (EVC) is an end-to-end representation of a single instance of a Layer 2 service that

- is offered by a provider and defines the service parameters,
- consists of one or more Layer 2 interfaces called service instances within the Cisco EVC Framework,
- instantiates a service instance on a specific router port, and
- associates each service instance with a bridge domain based on configuration.

An incoming frame can be classified as a service instance based on these criteria:

- Single 802.1Q VLAN tag, priority-tagged, or 802.1ad VLAN tag
- Both QinQ (inner and outer) VLAN tags, or both 802.1ad S-VLAN and C-VLAN tags
- Outer 802.1p CoS bits, inner 802.1p CoS bits, or both
- Payload Ethernet type (IPv4, IPv6, PPPoE-all, PPPoE-discovery, and PPPoE-session)

Service instance also supports alternative mapping criteria:

- Untagged–Maps to all frames lacking an 802.1Q or 802.1ad header
- Default–Maps to all frames

## Bridge domain interface encapsulation

Security Group classification includes both Source and Destination Group, which is specified by source SGT and DGT. SGT Based PBR feature provides the PBR route-map match clause for SGT/DGT based packet classification. SGT Based PBR feature supports configuration of an unlimited number of tags. Configure the tags according to the memory available on the platform.

An EVC provides the ability to employ different encapsulations on each Ethernet flow point (EFP) present in a bridge domain. A BDI egress point may not be aware of the encapsulation of an egress packet. The packet may have exited from one or more EFPs with different encapsulations.

Encapsulation in a bridge domain is a configuration method that dictates how traffic is tagged and untagged within the domain, depending on the encapsulation settings of its Ethernet Flow Points (EFPs).

- If all EFPs have different encapsulations, the Bridge Domain Interface (BDI) must be untagged (using the no 802.1Q tag), and you must configure traffic encapsulation (popped or pushed) and rewriting at each EFP.
- If all EFPs have the same encapsulation, configure the encapsulations directly on the BDI using the encapsulation command.
- Enabling encapsulation at the BDI when EFPs have the same encapsulation ensures effective pushing or popping of tags. This eliminates the need to configure the rewrite command at the EFPs.

## Assign a MAC address

Important information about assigning a MAC Address includes:

- All bridge domain interfaces on Cisco Catalyst 8500 Series Edge Platforms share one MAC address. The first interface within a bridge domain receives a MAC address, and each additional bridge domain interface in the same domain uses that address.
- You can configure a static MAC address on a bridge domain interface using the **mac-address** command.

## Support for IP protocols

Bridge domain interfaces allow the Cisco 8500 Series Catalyst Edge Platform to function as a Layer 3 endpoint on a Layer 2 bridge domain for these IP-related protocols:

- ARP
- DHCP
- HTTP
- ICMP
- NTP
- RARP
- SNMP
- TCP
- Telnet
- TFTP
- UDP

## IP forward support

Bridge domain interface supports these IP forwarding features:

- IPv4 input and output access control lists (ACL)
- IPv4 input and output QoS policies. The operations supported for the input and output service policies on a bridge domain interface are:
  - Classification
  - Marking
  - Policing
- IPv4 L3 VRFs

## Packet forwarding

A bridge domain interface provides bridging and forwarding services between the Layer 2 and Layer 3 network infrastructure.

### Layer 2 to Layer 3

During a packet flow from a Layer 2 network to a Layer 3 network, the packet or a copy of the packet is forwarded to the bridge domain interface in these cases.

- If the destination MAC address matches the bridge domain interface MAC address, the packet is forwarded to the bridge domain interface.
- If the destination MAC address is a multicast address, a copy of the packet is forwarded to the bridge domain interface.

> **Note**
>
> MAC address learning cannot not be performed on the bridge domain interface.

#### Layer 3 to Layer 2

When a packet arrives at a Layer 3 physical interface of a router, a route lookup action is performed. If the route lookup points to a bridge domain interface, the bridge domain interface adds layer 2 encapsulation and forwards the frame to the correct bridge domain.

During a Layer 2 lookup on a bridge domain to which the bridge domain interface belongs, the bridge domain forwards the packets to the correct service instance based on the destination MAC address.

## Link states of a bridge domain and a bridge domain interface

Bridge domain interface acts as a routable IOS interface on Layer 3 and as a port on a bridge domain. Both bridge domain interfaces and bridge domains operate with individual administrative states.

Shutting down a bridge domain interface stops the Layer 3 data service, but does not override or impact the state of the associated bridge domain.

Shutting down a bridge domain has several impacts on its associated components

- It stops Layer 2 forwarding across all associated members, including service instances and bridge domain interfaces.
- The operational state of a bridge domain is influenced by its associated service instances.
- A bridge domain interface can only become operational if at least one of its associated service instances is active.

Because a bridge domain interface is an internal interface, the operational state of bridge domain interface does not affect the bridge domain operational state.

#### Bridge domain interface-initial state

The initial administrative state of a BDI depends on how the BDI is created. When you create a BDI at boot time in the startup configuration, the default administrative state for the BDI is up. It will remain in this state unless the startup configuration includes the shutdown command.

This behavior is consistent with all the other interfaces. When you create a BDI dynamically at command prompt, the default administrative state is down.

#### Bridge domain interface-link state

A BDI maintains a link state that comprises of three states: administratively down, operationally down, and up. The link state of a BDI is derived from two independent inputs: the BDI administrative state set by the corresponding users and the fault indication state from the lower levels of the interface states. It defines a BDI link state based on the state of the two inputs.

| Fault Indication State | BDI Admin State | Resulting BDI Link State |
|---|---|---|
| No faults asserted | Shutdown | Admin-down |
| No faults asserted | No Shutdown | Up |
| At least one fault | Shutdown | Admin-down |
| At least one fault | No Shutdown | Operationally Down |

## Bridge domain interface statistics

For virtual interfaces, such as the bridge domain interface, protocol counters are periodically queried from the QFP.

When packets flow from a Layer 2 bridge domain network to a Layer 3 routing network through the bridge domain interface, the packets are treated as bridge domain interface input packets and bytes.

When packets arrive at a Layer 3 interface and are forwarded through the bridge domain interface to a Layer 2 bridge domain, the packets are treated as output packets and bytes, and the counters are updated accordingly.

A BDI maintains a standard set of Layer 3 packet counters, as is the case with all Cisco IOS interfaces.

The convention of the counters is relative to the Layer 3 cloud. For example, input refers to the traffic entry to the Layer 3 cloud from the Layer 2 BD, while output refers to the traffic exit from the Layer 3 cloud to the Layer 2 BD.

Use the **show interfaces accounting** command to display the statistics for the BDI status. Use the **show interface** *<if-name>* command to display the overall count of the packets and bytes that are transmitted and received.

## Create or delete a bridge domain interface

When you define an interface or subinterface for a Cisco IOS router, you name it and specify how it is assigned an IP address. You can create a bridge domain interface before adding a bridge domain to the system. The new bridge domain interface is activated after the associated bridge domain is configured.

When you create the bridge domain interface and the bridge domain, the system maintains the required associations for mapping the bridge domain-bridge domain interface pair.

The system maintains the mapping between bridge domains and bridge domain interfaces. The bridge domain interface uses the index of the associated bridge domain to show the association.

> **Note**
>
> When a bridge domain interface is created, a bridge domain is automatically created.

## Bridge domain virtual IP interface

The Virtual IP Interface (VIF) feature helps to associate multiple BDI interfaces with a BD instance. The BD-VIF interface inherits all the existing L3 features of IOS logical IP interface.

The Virtual IP Interface (VIF) feature has these limitations:

- BD-VIF interface does not support IP multicast.
- The number of BD-VIF interfaces with automatically generated MAC addresses varies depending on the platform.
- BD-VIF Interface does not support MPLS.
- The maximum number of BD-VIF interfaces per bridge-domain and the total number of BD-VIF interface for per system vary based on the type of platforms.

> **Note**
>
> You must configure every BD-VIF interface with a unique MAC address and it should belong to a different VRF.

The maximum number of BD-VIF supported on Cisco Catalyst 8500 Series Edge Platforms are:

- C8500-12X4QC supports a maximum of 100 BD-VIF interfaces for a bridge domain.
- C8500-12X supports a maximum of 16 BD-VIFs for a bridge domain.

> **Note**
>
> From Cisco IOS XE 17.7 release, BD-VIF supports Flexible Netflow (FNF).

## How to configure a bridge domain interface

To configure a bridge domain interface, perform the these steps:

**Procedure**

1. **enable**

   **Example**

   ```
   Router> enable
   ```

   Enables privileged EXEC mode. Enter your password if prompted.

2. **configureterminal**

   **Example**

   ```
   Router# configure terminal
   ```

   Enters global configuration mode.

3. **interface BDI***{interfacenumber}*

   **Example**

   ```
   Router(config-if)# interface BDI3
   ```

   Specifies a bridge domain interface on a Cisco 8500 Series Catalyst Edge Platform.

4. **encapsulation***encapsulationdot1q<first-tag>[second-dot1q<second-tag>]*

   **Example**

   ```
   Router(config-if)# encapsulation dot1Q 1 second-dot1q 2
   ```

   Defines the encapsulation type.

   The example shows how to define dot1q as the encapsulation type.

5. Do one of these steps:

**Example**

```
ipaddress ip-address mask
```

**Example**

```
ipv6address {X:X:X:X::X link-local | X:X:X:X::X/prefix [anycast | eui-64] | autoconfig [default]}
```

**Example**

```
Router(config-if)# ip address 2.2.2.1 255.255.255.0
```

**Example**

```
Router(config-if)# ipv6 address AB01:CD1:123:C::/64 eui-64
```

Specifies either the IPv4 or IPv6 address for the bridge domain interface.

6. **match security-group destination tag** *sgt-number*

**Example**

```
Router(config-route-map)# match security-group destination tag 150
```

Configures the value for security-group destination security tag.

7. **mac address** *{mac-address}*

**Example**

```
Router(config-if)# mac-address 1.1.3
```

Specifies the MAC address for the bridge domain interface.

8. **no shut**

**Example**

```
Router(config-if)# no shut
```

Enables the bridge domain interface on the Cisco 8500 Series Catalyst Edge Platform.

9. **shut**

**Example**

```
Router(config-if)# shut
```

Disables the bridge domain interface on the Cisco 8500 Series Catalyst Edge Platform.

---

Bridge domain interface is configured.

## Display and verify bridge domain interface configuration

To display and verify bridge domain interface configuration, follow these steps.

**Procedure**

1. **enable**

   **Example**

   ```
   Router> enable
   ```

   Enables privileged EXEC mode. Enter your password if prompted.

2. **show interfaces bdi**

   **Example**

   ```
   Router#showinterfacesBDI3
   ```

   Displays the configuration summary of the corresponding BDI.

3. **show platform software interface fp active name**

   **Example**

   ```
   Router# showplatformsoftwareinterfacefpactivenameBDI4
   ```

   Displays the bridge domain interface configuration in a Forwarding Processor.

4. **show platform hardware qfp active interface if-name**

   **Example**

   ```
   Router# showplatformhardwareqfpactiveinterfaceif-nameBDI4
   ```

   Displays the bridge domain interface configuration in a data path.

5. **debug platform hardware qfp feature**

   **Example**

   ```
   Router# debugplatformhardwareqfpactivefeaturel2bdclientall
   ```

   The selected CPP L2BD Client debugging is on.

6. **platform trace runtime process forwarding-manager module**

**Example**

```
Router(config)#platformtraceruntimeslotF0bay0processforwarding-managermoduleinterfaceslevelinfo
```

Enables the Forwarding Manager Route Processor and Embedded Service Processor trace messages for the Forwarding Manager process.

7. **platform trace boottimeprocess forwarding-manager module interfaces**

**Example**

```
Router(config)#platformtraceboottimeslotR0bay1processforwarding-managerforwarding-managerlevelmax
```

Enables the Forwarding Manager Route Processor and Embedded Service Processor trace messages for the Route Processor Forwarding Manager process during bootup.

---

Bridge domain interface configuration is verified.

**What to do next**

For additional information on the commands and the options available with each command, see the Cisco IOS Configuration Fundamentals Command Reference Guide located at:

{start hypertext}http://www.cisco.com/en/US/docs/ios/fundamentals/command/reference/cf_book.html{end hypertext}

## Configure bridge domain virtual IP interface

Explains virtual IP interface configuration, including associating VIF interfaces with a bridge domain, verifying interface settings, and integrating example configurations for deploying virtual IP interfaces within bridge domain environments.

To configure bridge-domain virtual IP interface.

```
enable
configure terminal
[no] interface BD-VIF interface-number
  [[no] vrf forwarding vrf-name]
  [[no] mac address mac-address]
  [[no] ip address ip-address mask]
  [[no] ipv6 address {X:X:X:X::X link-local| X:X:X:X::X/prefix [anycast |
eui-64] | autoconfig [default]}]

exit
```

> **Note**
>
> To delete BD-VIF interface, use the 'no' form of the command.

## Associate VIF interface with a bridge domain

To associate VIF interface with a bridge domain.

```
enable
configure terminal
bridge-domain bridge-domain number
[no] member BD-VIF interface-number
exit
```

## Verify bridge domain virtual IP interface

To verify bridge-domain virtual IP interface.

All existing show commands for interface and IP interface can be used for the BD-VIF interface.

- show interface bd-vif bd-vif-id
- show ip interface bd-vif bd-vif-id
- show bd-vif interfaces in fman-fp
- show pla sof inter fp ac brief | i BD_VIF

## Example configuration bridge domain virtual IP interface

Example configuration for Bridge-Domain virtual IP interface.

```
Detail sample:
interface Port-channel1
mtu 9000
no ip address
 !Ethernet service endpoint one per neutron network
service instance 1756 ethernet
  description 4e8e5957-649f-477b-9e5b-f1f75b21c03c
  encapsulation dot1q 1756
  rewrite ingress tag pop 1 symmetric
  bridge-domain 1756
!
interface BD-VIF5001
no shutdown
vrf forwarding vrf5001
ip address 10.0.0.1 255.255.255.0
interface BD-VIF5002
no shutdown
vrf forwarding vrf5002
ip address 10.0.0.2 255.255.255.0
bridge-domain 1756
member Port-channel1 service-instance 1756
member bd-vif5001
member bd-vif5002
```

# 22 Support for software media termination point

**Topics:**


- Support for Software Media Termination Point


Describes how to configure VRF Route Sharing on the router to enable traffic forwarding and shared services across hybrid cloud environments, specifically focusing on VxLAN peers and BGP EVPN integration.

# Support for Software Media Termination Point

Overview of support for Software Media Termination Point (MTP)

The Software Media Termination Point (MTP) is a feature that enables Cisco Unified Communications Manager (CUCM) to facilitate media stream bridging and call relaying between diverse connection types.

- Bridges media streams between two connections.
- Allows CUCM to relay calls routed through SIP or H.323 endpoints, and
- Utilizes Skinny Client Control Protocol (SCCP) commands to establish an MTP for call signaling.

This feature is crucial for ensuring interoperability and proper media handling when different signaling protocols (like SIP, H.323, and SCCP) are involved in a call path within a Cisco Unified Communications environment.

## Information about support for software media termination point

Information about support for Software Media Termination Point (MTP)

The extension of software MTP support to Cisco Unified Border Element (Enterprise) is a feature that

- Extends software MTP support to the Cisco Unified Border Element (Enterprise).
- Is an essential component for large-scale deployments of Cisco Unified Communications Manager (UCM), and
- Enables new capabilities for Cisco UBE to function as an Enterprise Edge Cisco Session Border Controller for large-scale SIP trunking deployments.

This enhancement is particularly important for organizations migrating to SIP trunking, as it ensures robust media handling and signaling capabilities at the enterprise edge.

## How to configure support for software media termination point

### Prerequisites

Writer tip: Use this title only when you need to logically group multiple principles that belong to the same category.

For the software MTP to function properly, codec and packetization must be configured the same way on both incoming call legs and outgoing call legs

### Restrictions

- RSVP Agent is not supported in software MTP.
- Hardware MTP for repacketization is not supported.
- Call Threshold is not supported for standalone software MTP.
- Per-call debugging is not supported.

### Configure support for software media termination point

Perform the following tasks to enable and configure the support for Software Media Termination Point feature.

To enable and configure the Support for Software Media Termination Point feature, perform these task.

**Procedure**

1. **enable**

   **Example**

   ```
   Router > enable
   ```

   Enables privileged EXEC mode.

   - Enter your password if prompted.

2. **configure terminal**

   **Example**

   ```
   Router# configure terminal
   ```

   Enters global configuration mode.

3. **sccp local** *interface-type interface-number* [ **port** *port-number* ]

   **Example**

   ```
   Router(config)# sccp local gigabitethernet0/0/0
   ```

   Selects the local interface that SCCP applications (transcoding and conferencing) use to register with Cisco UCM.

   - *interface type* --Can be an interface address or a virtual-interface address such as Ethernet.
   - *interface number* --Interface number that the SCCP application uses to register with Cisco UCM.
   - (Optional) **port** *port-number* --Port number used by the selected interface. Range is 1025 to 65535. Default is 2000.

4. **sccp ccm** { *ipv4-address* | *ipv6-address* | *dns* } **identifier** *identifier-number* [ **port** *port-number* ] **version** *version-number*

   **Example**

   ```
   Router(config)# sccp ccm 10.1.1.1 identifier 1 version 7.0+
   ```

   Adds a Cisco UCM server to the list of available servers and sets these parameters:

   - *ipv4-address* --IP version 4 address of the Cisco UCM server.
   - *ipv6-address* --IP version 6 address of the Cisco UCM server.
   - *dns* --DNS name.
   - **identifier** --Specifies the number that identifies the Cisco UCM server. Range is 1 to 65535.
   - **port** *port-number* (Optional)--Specifies the TCP port number. Range is 1025 to 65535. Default is 2000.

- **version** *version-number* --Cisco UCM version. Valid versions are 3.0, 3.1, 3.2, 3.3, 4.0, 4.1, 5.0.1, 6.0, and 7.0+. There is no default value.

5. **sccp**

   **Example**

   ```
   Router(config)# sccp
   ```

   Enables the Skinny Client Control Protocol (SCCP) and its related applications (transcoding and conferencing).

6. **sccp ccm group** *group-number*

   **Example**

   ```
   Router(config)# sccp ccm group 10
   ```

   Creates a Cisco UCM group and enters SCCP Cisco UCM configuration mode.

   - *group-number* --Identifies the Cisco UCM group. Range is 1 to 50.

7. **associate ccm** *identifier-number* **priority** *number*

   **Example**

   ```
   Router(config-sccp-ccm)# associate ccm 10 priority 3
   ```

   Associates a Cisco UCM with a Cisco UCM group and establishes its priority within the group:

   - *identifier-number* --Identifies the Cisco UCM. Range is 1 to 65535. There is no default value.
   - **priority** *number* --Priority of the Cisco UCM within the Cisco UCM group. Range is 1 to 4. There is no default value. The highest priority is 1.

8. **associate profile** *profile-identifier* **register** *device-name*

   **Example**

   ```
   Router(config-sccp-ccm)# associate profile 1 register MTP0011
   ```

   Associates a DSP farm profile with a Cisco UCM group:

   - *profile-identifier* --Identifies the DSP farm profile. Range is 1 to 65535. There is no default value.
   - **register** *device-name* --Device name in Cisco UCM. A maximum of 15 characters can be entered for the device name.

9. **dspfarm profile** *profile-identifier* { **conference** | **mtp** | **transcode** } [ **security** ]

   **Example**

   ```
   Router(config-sccp-ccm)# dspfarm profile 1 mtp
   ```

Enters DSP farm profile configuration mode and defines a profile for DSP farm services:

- *profile-identifier* --Number that uniquely identifies a profile. Range is 1 to 65535. There is no default.
- **conference** --Enables a profile for conferencing.
- **mtp** --Enables a profile for MTP.
- **transcode** --Enables a profile for transcoding.
- **security** (Optional)-- Enables a profile for secure DSP farm services.

10. **trustpoint** *trustpoint-label*

**Example**

```
Router(config-dspfarm-profile)# trustpoint dspfarm
```

(Optional) Associates a trustpoint with a DSP farm profile.

11. **codec** *codec*

**Example**

```
Router(config-dspfarm-profile)# codec g711ulaw
```

Specifies the codecs supported by a DSP farm profile.

- *codec-type* : Specifies the preferred codec. Enter ? for a list of supported codecs

Repeat this step for each supported codec.

12. **maximum sessions** { **hardware** | **software** } *number*

**Example**

```
Router(config-dspfarm-profile)# maximum sessions software 10
```

Specifies the maximum number of sessions that are supported by the profile.

- **hardware** --Number of sessions that MTP hardware resources can support.
- **software** --Number of sessions that MTP software resources can support.
- *number* --Number of sessions that are supported by the profile. Range is 0 to x. Default is 0. The x value is determined at run time depending on the number of resources available with the resource provider.

13. **associate application sccp**

**Example**

```
Router(config-dspfarm-profile)# associate application sccp
```

Associates SCCP to the DSP farm profile.

14. **no shutdown**

**Example**

```
Router(config-dspfarm-profile)# no shutdown
```

Changes the status of the interface to the UP state.

---

Support for software media termination point is configured.

### Configuration examples for software media termination point

Output examples for software media termination point.

This example shows a sample configuration for the Support for Software Media Termination Point feature.

```
sccp local GigabitEthernet0/0/1
sccp ccm 10.13.40.148 identifier 1 version 6.0
sccp
!
sccp ccm group 1
 bind interface GigabitEthernet0/0/1
 associate ccm 1 priority 1
 associate profile 6 register RR_RLS6
!
 dspfarm profile 6 mtp
 codec g711ulaw
 maximum sessions software 100
 associate application SCCP
!
!
gateway
media-inactivity-criteria all
timer receive-rtp 400
```

### Troubleshoot software termination point

To troubleshoot software termination point

To verify and troubleshoot this feature, use these **show** commands.

- To verify information about SCCP, use the **show sccp** command:

```
Router#
                show sccp
SCCP Admin State: UP
Gateway IP Address: 10.13.40.157, Port Number: 2000
IP Precedence: 5
User Masked Codec list: None
Call Manager: 10.13.40.148, Port Number: 2000
                Priority: N/A, Version: 6.0, Identifier: 1
                Trustpoint: N/A
```

- To verify information about the DSPfarm profile, use the **show dspfarm profile** command:

```
Router#
                show dspfarm profile 6
Dspfarm Profile Configuration
 Profile ID = 6, Service = MTP, Resource ID = 1
 Profile Description :
 Profile Service Mode : Non Secure
 Profile Admin State : UP
 Profile Operation State : ACTIVE
 Application : SCCP   Status : ASSOCIATED
 Resource Provider : NONE   Status : NONE
 Number of Resource Configured : 100
 Number of Resource Available : 100
 Hardware Configured Resources : 0
 Hardware Available Resources : 0
 Software Resources : 100
 Codec Configuration
 Codec : g711ulaw, Maximum Packetization Period : 30
```

- To display statistics for the SCCP connections, use the **show sccp connections** command:

```
Router#
                show sccp connections
sess_id    conn_id    stype mode     codec   ripaddr         rport sport
16808048   16789079     mtp   sendrecv g711u 10.13.40.20     17510 7242
16808048   16789078     mtp   sendrecv g711u 10.13.40.157     6900 18050
```

- To display information about RTP connections, use the **show rtpspi call** command:

```
Router#
                show rtpspi call
RTP Service Provider info:
No. CallId dstCallId Mode      LocalRTP RmtRTP LocalIP    RemoteIP   SRTP
    22     19        Snd-Rcv   7242     17510  0x90D080F  0x90D0814    0
    19     22        Snd-Rcv   18050    6900   0x90D080F  0x90D080F    0
```

- To display information about VoIP RTP connections, use the **show voip rtp connections** command:

```
Router#
                show voip rtp connections
VoIP RTP Port Usage Information
Max Ports Available: 30000, Ports Reserved: 100, Ports in Use: 102
Port range not configured, Min: 5500, Max: 65499
VoIP RTP active connections :
No. CallId     dstCallId  LocalRTP  RmtRTP   LocalIP         RemoteIP
1     114        117        19822    24556  10.13.40.157     10.13.40.157
2     115        116        24556    19822  10.13.40.157     10.13.40.157
3     116        115        19176    52625  10.13.40.157     10.13.40.20
4     117        114        16526    52624  10.13.40.157     10.13.40.20
```

- Additional, more specific, **show** commands that can be used are
  - **show sccp connection callid**
  - **show sccp connection connid**
  - **show sccp connection sessionid**
  - **show rtpspi call callid**
  - **show rtpspi stat callid**
  - **show voip rtp connection callid**
  - **show voip rtp connection type**
- To isolate specific problems, use the **debug sccp** command:
  - **debug sccp [ all | config | errors | events | keepalive | messages | packets | parser | tls ]**

# 23 Radio Aware Routing

**Topics:**


- Radio Aware Routing


Explains how to configure the Resource Allocation and Reporting (RAR) functionality on the router, detailing how the router monitors and reports on the underlying virtual machine's resource allocation to ensure performance and license compliance.

# Radio Aware Routing

Details the Radio Aware Routing (RAR) feature, which allows routers to interact with radios to optimize routing based on link conditions, including configuration examples for Bypass and Aggregate modes.

Radio aware routing (RAR) is a mechanism that uses radios to interact with the routing protocol OSPFv3 to signal the appearance, disappearance, and link conditions of one-hop routing neighbors.

In a large mobile networks, connections to the routing neighbors are often interrupted due to distance and radio obstructions. When these signals do not reach the routing protocols, protocol timers are used to update the status of a neighbor. Routing protocols have lengthy timer, which is not recommended in mobile networks.

## Benefits

The radio aware routing feature offers these benefits:

- Provides faster network convergence through immediate recognition of changes.
- Enables routing for failing or fading radio links.
- Facilitates routing between line-of-sight and non-line-of-sight paths with ease.
- Provides faster convergence and optimizes route selection to ensure uninterrupted delivery of delay-sensitive traffic, such as voice and video.
- Provides efficient radio resources and bandwidth usage.
- Reduces impact on the radio links by performing congestion control in the router.
- Allows route selection based on radio power conservation.
- Enables decoupling of the routing and radio functionalities.
- Provides a simple Ethernet connection to radios that are compliant with RFC 5578, R2CP, and DLEP standards.

## Restrictions

These restrictions have to be noted before you configure radio aware routing feature:

- The DLEP and R2CP protocols are not supported on Cisco Catalyst 8300 Edge Platform.
- Multicast traffic is not supported in aggregate mode.
- Cisco High Availability (HA) technology is not supported.

## License requirements

The radio aware routing feature is made available with the AppX license.

## System components

The radio aware routing feature is implemented using the MANET (Mobile ad hoc network) infrastructure comprising of different components:

- PPPoE
- Virtual multipoint interface (VMI)
- QoS
- routing protocol interface and

- RAR protocols

#### Point-to-Point Protocol over Ethernet (PPPoE)

PPPoE is a well-defined communication mechanism between the client and the server. In the RAR implementation, radio takes the role of the PPPoE client and router takes the role of the PPPoE server. This allows a loose coupling of radio and router, while providing a well-defined and predictable communication mechanism.

As PPPoE is a session or a connection oriented protocol, it extends the point-to-point radio frequency (RF) link from an external radio to an IOS router.

#### PPPoE Extensions

PPPoE extensions are used when the router communicates with the radio. In the Cisco IOS implementation of PPPoE, each individual session is represented by virtual access interface (connectivity to a radio neighbor) on which, QoS can be applied with these PPPoE extensions.

RFC5578 provides extensions to PPPoE to support credit-based flow control and session-based real time link metrics, which are very useful for connections with variable bandwidth and limited buffering capabilities (such as radio links).

#### Virtual Multipoint Interface (VMI)

VMI manages and translates events for higher layers, such as routing protocols and operates in the Bypass mode.

In Bypass mode, every Virtual Access Interface (VAI) representing a radio neighbor is exposed to routing protocols OSPFv3 and EIGRP, so that, the routing protocol directly communicates with the respective VAI for both unicast and multicast routing protocol traffic.

In Aggregate mode, VMI is exposed to the routing protocols (OSPF) so that the routing protocols can leverage VMI for their optimum efficiency. When the network neighbors are viewed as a collection of networks on a point-to-multipoint link with broadcast and multicast capability at VMI, VMI helps in aggregating the multiple virtual access interfaces created from PPPoE. VMI presents a single multi access layer 2 broadcast capable interface. The VMI layer handles re-directs unicast routing protocol traffic to the appropriate P2P link (Virtual-Access interface), and replicates any Multicast/Broadcast traffic that needs to flow. Since the routing protocol communicates to a single interface, the size of the topology database is reduced, without impacting the integrity of the network.

## QoS provisioning on PPPoE extension session

The example describes QoS provisioning on PPPoE extension session:

```
policy-map rar_policer
 class class-default
  police 10000 2000 1000 conform-action transmit  exceed-action drop
violate-action drop
policy-map rar_shaper
 class class-default
  shape average percent 1

interface Virtual-Template2
 ip address 192.0.2.7 255.255.255.0
 no peer default ip address
 no keepalive
 service-policy input rar_policer
end
```

## Configuration examples for the RAR feature in Bypass Mode

The example shows an end-to-end configuration of RAR in the bypass mode:

**Note**

Before you begin the RAR configuration, you must first configure the **subscriber authorization enable** command to bring up the RAR session. Without enabling authorization, the Point-to-Point protocol does not recognize this as a RAR session and it may not tag *manet_radio* in presentation of a PPPoE Active Discovery Initiate (PADI). By default, bypass mode does not appear in the configuration. It appears only if the mode is configured as bypass.

### Configure a service for radio aware routing

```
policy-map type service rar-lab
pppoe service manet_radio //note: Enter the pppoe service policy name as
manet_radio
!
```

### Configure broadband

```
bba-group pppoe VMI2
virtual-template 2
service profile rar-lab
!
interface GigabitEthernet0/0/0
description Connected to Client1
 negotiation auto
 pppoe enable group VMI2
!
```

### Configure a service for radio aware routing

```
policy-map type service rar-lab
pppoe service manet_radio //note: Enter the pppoe service policy name as
manet_radio
!
```

Configuration in Bypass Mode

- IP Address Configured under Virtual-Template Explicitly

```
interface Virtual-Template2
ip address 192.0.2.7 255.255.255.0
no ip redirects
peer default ip address pool PPPoEpool2
ipv6 enable
ospfv3 1 network manet
ospfv3 1 ipv4 area 0
ospfv3 1 ipv6 area 0
no keepalive
service-policy input rar_policer Or/And
service-policy output rar_shaper
```

- VMI Unnumbered Configured under Virtual Template

```
interface Virtual-Template2
ip unnumbered vmi2
no ip redirects
peer default ip address pool PPPoEpool2
ipv6 enable
ospfv3 1 network manet
ospfv3 1 ipv4 area 0
ospfv3 1 ipv6 area 0
no keepalive
service-policy input rar_policer Or/And
service-policy output rar_shaper
```

**Configure the Virtual Multipoint Interface in Bypass Mode**

```
interface vmi2 //configure the virtual multi interface
ip address 192.0.2.5 255.255.255.0
physical-interface GigabitEthernet0/0/0
mode bypass
```

```
interface vmi3//configure the virtual multi interface
 ip address 192.0.2.6 255.255.255.0
 physical-interface GigabitEthernet0/0/1
mode bypass
```

**Configure OSPF Routing**

```
router ospfv3 1
router-id 192.0.2.1
!
address-family ipv4 unicast
 redistribute connected metric 1 metric-type 1
 log-adjacency-changes
exit-address-family
!
address-family ipv6 unicast
 redistribute connected metric-type 1
 log-adjacency-changes
exit-address-family
!
ip local pool PPPoEpool2 192.0.2.8 192.0.2.4
```

## Configuration examples for the RAR Feature in Aggregate Mode

An example configuration of radio aware routing in aggregate mode is provided.

**Note**

Before you begin the RAR configuration, you must first configure the **subscriber authorization enable** command to bring up the RAR session. Without enabling authorization, the Point-to-Point protocol does not recognize this as a RAR session and may not tag manet_radio in PADI.

Configure a service for radio aware routing

```
policy-map type service rar-lab
pppoe service manet_radio //note: Enter the pppoe service policy name as
manet_radio
!
```

Configure broadband

```
bba-group pppoe VMI2
virtual-template 2
service profile rar-lab

!
interface GigabitEthernet0/0/0
description Connected to Client1
 negotiation auto
 pppoe enable group VMI2

!
```

Configure a service for radio aware routing

```
policy-map type service rar-lab
pppoe service manet_radio //note: Enter the pppoe service policy name as
manet_radio
!
```

Configure the aggregate mode

```
interface Virtual-Template2
ip unnumbered vmi2
no ip redirects
no peer default ip address
ipv6 enable
no keepalive
service-policy input rar_policer Or/And
service-policy output rar_shaper
```

Configure the Virtual Multipoint Interface in aggregate mode

```
interface vmi2 //configure the virtual multi interface
ip address 192.0.2.8 255.255.255.0
```

```
physical-interface GigabitEthernet0/0/0
mode aggregate
```

```
interface vmi3//configure the virtual multi interface
 ip address 192.0.2.4 255.255.255.0
 no ip redirects
 no ip split-horizon eigrp 1
 physical-interface GigabitEthernet0/0/1
mode aggregate
```

Configure OSPF Routing

```
router ospfv3 1
router-id 192.0.2.1
!
address-family ipv4 unicast
 redistribute connected metric 1 metric-type 1
 log-adjacency-changes
exit-address-family
!
address-family ipv6 unicast
 redistribute connected metric-type 1
 log-adjacency-changes
exit-address-family
!
ip local pool PPPoEpool2 192.0.2.4 192.0.2.8
ip local pool PPPoEpool3 192.0.2.6 192.0.2.2
```

## Verify radio aware routing session details

Use these show commands to retrieve radio aware routing session details:

```
Router#show pppoe session packets all
Total PPPoE sessions 2


session id: 9
local MAC address: 006b.f10e.a5e0, remote MAC address: 0050.56bc.424a
virtual access interface: Vi2.1, outgoing interface: Gi0/0/0
   1646 packets sent, 2439363 received
   176216 bytes sent, 117250290 received

PPPoE Flow Control Stats
Local Credits: 65535   Peer Credits: 65535   Local Scaling Value 64 bytes
Credit Grant Threshold: 28000    Max Credits per grant: 65535
Credit Starved Packets: 0
PADG xmit Seq Num: 32928     PADG Timer index: 0
PADG last rcvd Seq Num: 17313
PADG last nonzero Seq Num: 17306
PADG last nonzero rcvd amount: 2
PADG Timers: (ms)   [0]-1000    [1]-2000    [2]-3000    [3]-4000    [4]-5000
PADG xmit: 33308  rcvd: 17313
PADC xmit: 17313  rcvd: 19709
In-band credit pkt xmit: 7 rcvd: 2434422
Last credit packet snapshot

 PADG xmit: seq_num = 32928, fcn = 0, bcn = 65535
```

```
 PADC rcvd: seq_num = 32928, fcn = 65535, bcn = 65535
 PADG rcvd: seq_num = 17313, fcn = 0, bcn = 65535
 PADC xmit: seq_num = 17313, fcn = 65535, bcn = 65535
 In-band credit pkt xmit: fcn = 61, bcn = 65533
 In-band credit pkt rcvd: fcn = 0, bcn = 65534
   ==== PADQ Statistics ====
    PADQ xmit: 0  rcvd: 0


session id: 10
local MAC address: 006b.f10e.a5e1, remote MAC address: 0050.56bc.7dcb
virtual access interface: Vi2.2, outgoing interface: Gi0/0/1
   1389302 packets sent, 1852 received
   77869522 bytes sent, 142156 received

PPPoE Flow Control Stats
Local Credits: 65535   Peer Credits: 65535   Local Scaling Value 64 bytes
Credit Grant Threshold: 28000    Max Credits per grant: 65535
Credit Starved Packets: 0
PADG xmit Seq Num: 18787     PADG Timer index: 0
PADG last rcvd Seq Num: 18784
PADG last nonzero Seq Num: 18768
PADG last nonzero rcvd amount: 2
PADG Timers: (ms)   [0]-1000    [1]-2000    [2]-3000    [3]-4000    [4]-5000
PADG xmit: 18787  rcvd: 18784
PADC xmit: 18784  rcvd: 18787
In-band credit pkt xmit: 1387764 rcvd: 956
Last credit packet snapshot
 PADG xmit: seq_num = 18787, fcn = 0, bcn = 65535
 PADC rcvd: seq_num = 18787, fcn = 65535, bcn = 65535
 PADG rcvd: seq_num = 18784, fcn = 0, bcn = 65535
 PADC xmit: seq_num = 18784, fcn = 65535, bcn = 65535
 In-band credit pkt xmit: fcn = 0, bcn = 64222
 In-band credit pkt rcvd: fcn = 0, bcn = 65534
   ==== PADQ Statistics ====
    PADQ xmit: 0  rcvd: 1
```

```
Router#show pppoe session packets
Total PPPoE sessions 2

SID     Pkts-In         Pkts-Out        Bytes-In        Bytes-Out
9       2439391         1651            117252098       176714
10      1858            1389306         142580          77869914
```

```
Router#show vmi counters
Interface vmi2: - Last Clear Time =

Input Counts:
 Process Enqueue      =          0 (VMI)
 Fastswitch           =          0
 VMI Punt Drop:
      Queue Full      =          0

Output Counts:
 Transmit:
      VMI Process DQ  =       4280
      Fastswitch VA   =          0
      Fastswitch VMI  =          0
```

```
 Drops:
      Total           =          0
      QOS Error       =          0
      VMI State Error =          0
      Mcast NBR Error =          0
      Ucast NBR Error =          0
Interface vmi3: - Last Clear Time =

Input Counts:
 Process Enqueue      =          0 (VMI)
 Fastswitch           =          0
 VMI Punt Drop:
      Queue Full      =          0

Output Counts:
 Transmit:
      VMI Process DQ  =       2956
      Fastswitch VA   =          0
      Fastswitch VMI  =          0
 Drops:
      Total           =          0
      QOS Error       =          0
      VMI State Error =          0
      Mcast NBR Error =          0
      Ucast NBR Error =          0
Interface vmi4: - Last Clear Time =

Input Counts:
 Process Enqueue      =          0 (VMI)
 Fastswitch           =          0
 VMI Punt Drop:
      Queue Full      =          0

Output Counts:
 Transmit:
      VMI Process DQ  =          0
      Fastswitch VA   =          0
      Fastswitch VMI  =          0
 Drops:
      Total           =          0
      QOS Error       =          0
      VMI State Error =          0
      Mcast NBR Error =          0
      Ucast NBR Error =          0
Router#
```

```
Router#show vmi neighbor details
1 vmi2 Neighbors
        1 vmi3 Neighbors
        0 vmi4 Neighbors
        2 Total Neighbors

vmi2   IPV6 Address=FE80::21E:E6FF:FE43:F500
      IPV6 Global Addr=::
      IPV4 Address=192.0.2.6, Uptime=05:15:01
      Output pkts=89, Input pkts=0
      No Session Metrics have been received for this neighbor.
      Transport PPPoE, Session ID=9
      INTERFACE STATS:
         VMI Interface=vmi2,
            Input qcount=0, drops=0, Output qcount=0, drops=0
```

```
         V-Access intf=Virtual-Access2.1,
            Input qcount=0, drops=0, Output qcount=0, drops=0
         Physical intf=GigabitEthernet0/0/0,
            Input qcount=0, drops=0, Output qcount=0, drops=0

PPPoE Flow Control Stats
Local Credits: 65535   Peer Credits: 65535   Local Scaling Value 64 bytes
Credit Grant Threshold: 28000    Max Credits per grant: 65535
Credit Starved Packets: 0
PADG xmit Seq Num: 33038     PADG Timer index: 0
PADG last rcvd Seq Num: 17423
PADG last nonzero Seq Num: 17420
PADG last nonzero rcvd amount: 2
PADG Timers: (ms)   [0]-1000    [1]-2000    [2]-3000    [3]-4000    [4]-5000
PADG xmit: 33418  rcvd: 17423
PADC xmit: 17423  rcvd: 19819
In-band credit pkt xmit: 7 rcvd: 2434446
Last credit packet snapshot

 PADG xmit: seq_num = 33038, fcn = 0, bcn = 65535

 PADC rcvd: seq_num = 33038, fcn = 65535, bcn = 65535

 PADG rcvd: seq_num = 17423, fcn = 0, bcn = 65535

 PADC xmit: seq_num = 17423, fcn = 65535, bcn = 65535

 In-band credit pkt xmit: fcn = 61, bcn = 65533

 In-band credit pkt rcvd: fcn = 0, bcn = 65534
   ==== PADQ Statistics ====
    PADQ xmit: 0  rcvd: 0


vmi3   IPV6 Address=FE80::21E:7AFF:FE68:6100
      IPV6 Global Addr=::
      IPV4 Address=192.0.2.10, Uptime=05:14:55
      Output pkts=6, Input pkts=0
      METRIC DATA: Total rcvd=1, Avg arrival rate (ms)=0
         CURRENT: MDR=128000 bps, CDR=128000 bps
                  Lat=0 ms, Res=100, RLQ=100, load=0
         MDR      Max=128000 bps, Min=128000 bps, Avg=128000 bps
         CDR      Max=128000 bps, Min=128000 bps, Avg=128000 bps
         Latency  Max=0, Min=0, Avg=0 (ms)
         Resource Max=100%, Min=100%, Avg=100%
         RLQ      Max=100, Min=100, Avg=100
         Load     Max=0%, Min=0%, Avg=0%
      Transport PPPoE, Session ID=10
      INTERFACE STATS:
         VMI Interface=vmi3,
            Input qcount=0, drops=0, Output qcount=0, drops=0
         V-Access intf=Virtual-Access2.2,
            Input qcount=0, drops=0, Output qcount=0, drops=0
         Physical intf=GigabitEthernet0/0/1,
            Input qcount=0, drops=0, Output qcount=0, drops=0

PPPoE Flow Control Stats
Local Credits: 65535   Peer Credits: 65535   Local Scaling Value 64 bytes
Credit Grant Threshold: 28000    Max Credits per grant: 65535
Credit Starved Packets: 0
PADG xmit Seq Num: 18896     PADG Timer index: 0
PADG last rcvd Seq Num: 18894
```

```
PADG last nonzero Seq Num: 18884
PADG last nonzero rcvd amount: 2
PADG Timers: (ms)   [0]-1000    [1]-2000    [2]-3000    [3]-4000    [4]-5000
PADG xmit: 18896  rcvd: 18894
PADC xmit: 18894  rcvd: 18896
In-band credit pkt xmit: 1387764 rcvd: 961
Last credit packet snapshot
 PADG xmit: seq_num = 18896, fcn = 0, bcn = 65535
 PADC rcvd: seq_num = 18896, fcn = 65535, bcn = 65535
 PADG rcvd: seq_num = 18894, fcn = 0, bcn = 65535
 PADC xmit: seq_num = 18894, fcn = 65535, bcn = 65535
 In-band credit pkt xmit: fcn = 0, bcn = 64222
 In-band credit pkt rcvd: fcn = 0, bcn = 65534
   ==== PADQ Statistics ====
    PADQ xmit: 0  rcvd: 1
```

```
Router#show vmi neighbor details vmi 2
             1 vmi2 Neighbors

vmi2   IPV6 Address=FE80::21E:E6FF:FE43:F500
      IPV6 Global Addr=::
      IPV4 Address=192.0.2.4, Uptime=05:16:03
      Output pkts=89, Input pkts=0
      No Session Metrics have been received for this neighbor.
      Transport PPPoE, Session ID=9
      INTERFACE STATS:
         VMI Interface=vmi2,
            Input qcount=0, drops=0, Output qcount=0, drops=0
         V-Access intf=Virtual-Access2.1,
            Input qcount=0, drops=0, Output qcount=0, drops=0
         Physical intf=GigabitEthernet0/0/0,
            Input qcount=0, drops=0, Output qcount=0, drops=0

PPPoE Flow Control Stats
Local Credits: 65535   Peer Credits: 65535   Local Scaling Value 64 bytes
Credit Grant Threshold: 28000    Max Credits per grant: 65535
Credit Starved Packets: 0
PADG xmit Seq Num: 33100     PADG Timer index: 0
PADG last rcvd Seq Num: 17485
PADG last nonzero Seq Num: 17449
PADG last nonzero rcvd amount: 2
PADG Timers: (ms)   [0]-1000    [1]-2000    [2]-3000    [3]-4000    [4]-5000
PADG xmit: 33480  rcvd: 17485
PADC xmit: 17485  rcvd: 19881
In-band credit pkt xmit: 7 rcvd: 2434460
Last credit packet snapshot
 PADG xmit: seq_num = 33100, fcn = 0, bcn = 65535
 PADC rcvd: seq_num = 33100, fcn = 65535, bcn = 65535
 PADG rcvd: seq_num = 17485, fcn = 0, bcn = 65535
 PADC xmit: seq_num = 17485, fcn = 65535, bcn = 65535
 In-band credit pkt xmit: fcn = 61, bcn = 65533
 In-band credit pkt rcvd: fcn = 0, bcn = 65534
   ==== PADQ Statistics ====
    PADQ xmit: 0  rcvd: 0
```

```
Router#show platform hardware qfp active feature ess session
Current number sessions: 2
Current number TC flow: 0
Feature Type: A=Accounting D=Policing(DRL) F=FFR M=DSCP Marking L=L4redirect
```

```
P=Portbundle T=TC

  Session    Type      Segment1      SegType1      Segment2      SegType2
Feature Other
-------------------------------------------------------------------------------

       21     PPP 0x0000001500001022    PPPOE 0x0000001500002023    LTERM
-------
       24     PPP 0x0000001800003026    PPPOE 0x0000001800004027    LTERM
-------
```

```
Router#show platform software subscriber pppoe_fctl evsi 21
PPPoE Flow Control Stats
Local Credits: 65535   Peer Credits: 65535   Local Scaling Value 64 bytes
Credit Grant Threshold: 28000    Max Credits per grant: 65535
Credit Starved Packets: 0
PADG xmit Seq Num: 33215     PADG Timer index: 0
PADG last rcvd Seq Num: 17600
PADG last nonzero Seq Num: 17554
PADG last nonzero rcvd amount: 2
PADG Timers: (ms)   [0]-1000    [1]-2000    [2]-3000    [3]-4000    [4]-5000
PADG xmit: 33595  rcvd: 17600
PADC xmit: 17600  rcvd: 19996
In-band credit pkt xmit: 7 rcvd: 2434485
Last credit packet snapshot
 PADG xmit: seq_num = 33215, fcn = 0, bcn = 65535
 PADC rcvd: seq_num = 33215, fcn = 65535, bcn = 65535
 PADG rcvd: seq_num = 17600, fcn = 0, bcn = 65535
 PADC xmit: seq_num = 17600, fcn = 65535, bcn = 65535
 In-band credit pkt xmit: fcn = 61, bcn = 65533
 In-band credit pkt rcvd: fcn = 0, bcn = 65534

BQS buffer statistics
Current packets in BQS buffer: 0
Total en-queue packets: 0 de-queue packets: 0
Total dropped packets: 0

Internal flags: 0x0
```

```
Router#show platform hardware qfp active feature ess session id 21
Session ID: 21

 EVSI type: PPP
 SIP Segment ID: 0x1500001022
 SIP Segment type: PPPOE
 FSP Segment ID: 0x1500002023
 FSP Segment type: LTERM
 QFP if handle: 16
 QFP interface name: EVSI21
 SIP TX Seq num: 0
 SIP RX Seq num: 0
 FSP TX Seq num: 0
 FSP RX Seq num: 0
 Condition Debug: 0x00000000
   session
```

```
Router#show ospfv3 neighbor

         OSPFv3 1 address-family ipv4 (router-id 192.0.2.3)

Neighbor ID     Pri   State           Dead Time   Interface ID    Interface
192.0.2.1           0   FULL/  -        00:01:32    19
Virtual-Access2.1

         OSPFv3 1 address-family ipv6 (router-id 192.0.2.3)

Neighbor ID     Pri   State           Dead Time   Interface ID    Interface
192.0.2.1           0   FULL/  -        00:01:52    19
Virtual-Access2.1
Router#
```

```
Router#sh ip route

Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2
i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
ia - IS-IS inter area, * - candidate default, U - per-user static route
o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
a - application route
+ - replicated route, % - next hop override, p - overrides from PfR

Gateway of last resort is not set

     192.0.2.8/8 is variably subnetted, 3 subnets, 2 masks
C        192.0.2.5/24 is directly connected, Virtual-Access2.1
O        192.0.2.6/32 [110/1] via 192.0.2.22, 00:00:03, Virtual-Access2.1
L        192.0.2.7/32 is directly connected, Virtual-Access2.1
     192.0.2.12/32 is subnetted, 1 subnets
C        192.0.2.20 is directly connected, Virtual-Access2.1
```

## Troubleshoot radio aware routing

You can troubleshoot the radio aware routing using these debug commands:

- **debug pppoe errors**
- **debug pppoe events**
- **debug ppp error**
- **debug vmi error**
- **debug vmi neighbor**
- **debug vmi packet**
- **debug vmi pppoe**
- **debug vmi registries**
- **debug vmi multicast**
- **debug vtemplate cloning**

- **debug vtemplate event**
- **debug vtemplate error**
- **debug plat hard qfp ac feature subscriber datapath pppoe detail**