# Cisco Catalyst 8000V Edge Software Configuration Guide

**First Published:** 2020-11-23

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

CHAPTER 1

# Overview of Cisco Catalyst 8000V Edge Software

### About Cisco Catalyst 8000V

The Cisco Catalyst 8000V Edge Software or Cisco Catalyst 8000V router provides a cloud-based virtual router deployed on a virtual machine (VM) instance on a x86 server hardware. This router supports a subset of Cisco IOS XE software features and technologies, providing Cisco IOS XE security and switching features on a virtualization platform.

When you deploy Cisco Catalyst 8000V on a VM, the Cisco IOS XE software functions just as if it were deployed on a traditional Cisco hardware platform. This router 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.

The Cisco Catalyst 8000V router also provides a virtual IOS XE operating system for routing and forwarding on the Enterprise Network Compute System (ENCS) platform.

You can deploy a Cisco Catalyst 8000V router as a virtual machine on a hypervisor. Optionally, you can use a virtual switch (vSwitch), depending on your deployment. Use selected Cisco equipment for supporting components, which depends on your software release.

# Benefits of Virtualization Using the Cisco Catalyst 8000V Router

- **Hardware independence**: The Cisco Catalyst 8000V router uses the benefits of virtualization in the cloud to provide hardware independence. Since the 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 among the VMs. You can regulate the amount of hardware resources that the VM server allocates to a specific VM. You can reallocate resources to another VM on the server.

- **Flexibility in deployment**: You can easily move a VM from one server to another. Thus, you can move a Cisco Catalyst 8000V instance from a server in one physical location to a server in another physical location without moving any hardware resources.

# Software Configuration and Management Using the Cisco IOS XE CLI

You can perform software configuration and management of a Cisco Catalyst 8000V router using the following methods:

- Provision a serial port in the VM and connect to access the Cisco IOS XE CLI commands.
- Use the virtual VGA console or the console on the virtual serial port to access the Cisco IOS XE CLI commands.

> **Note** You can 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 example, the Citrix XenServer environment does not support serial port association. See your hypervisor documentation for more details.

- Use remote SSH/Telnet to access the Cisco IOS XE CLI commands.

# Router Interfaces

The Cisco Catalyst 8000V router interfaces perform the same functionality as those on hardware-based Cisco routers. The Cisco Catalyst 8000V interfaces function as follows:

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

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 the *Mapping the Cisco Catalyst 8000V Network Interfaces to the VM Network Interfaces* section.

### Interface Numbering

- The interface port numbering is from 1 and up to the number of interfaces supported.
- GigabitEthernet interface 0 is not supported.
- You can designate any interface as the management interface. You can also change the management interface when you deploy the OVA template when installing the router for the first time.

# Virtual Machine Requirements

The Cisco Catalyst 8000V router runs only on a virtual machine. This section describes the virtual machine requirements for the router.

## Virtual Machines

A virtual machine (VM) is a software implementation of a computing environment in which an operating system (OS) or program can be installed and run. The VM typically emulates a physical computing environment, but requests for CPU, memory, hard disk, network and other hardware resources are managed by a virtualization layer which translates these requests to the underlying physical hardware.

You can deploy an Open Virtualization Archive (OVA) file. The OVA file package simplifies the process of deploying a VM by providing a complete definition of the parameters and resource allocation requirements for the new VM.

An OVA file consists of a descriptor (.ovf) file, a storage (.vmdk) file and a manifest (.mf) file.

- ovf file - Descriptor file which is an xml file with extension .ovf which consists of all the metadata about the package. It encodes all the product details, virtual hardware requirements and licensing.
- vmdk file - File format that encodes a single virtual disk from a VM.
- mf file - Optional file that stores the SHA key generated during packaging.

You can also install the Cisco Catalyst 8000V using an .iso file and manually create the VM in the hypervisor.

For more information, see the *Installation Overview* section in this guide.

## Hypervisor Support

A hypervisor enables multiple operating systems to share a single hardware host machine. While each operating system appears to have the dedicated use of the host's processor, memory, and other resources; the hypervisor controls and allocates only needed resources to each operating system and ensures that the operating systems (VMs) do not disrupt each other.

### Supported Hypervisor Types

Cisco Catalyst 8000V installation is supported on selected **Type 1** (native, bare metal) hypervisors. Installation is not supported on **Type 2** (hosted) hypervisors, such as VMware Fusion, VMware Player, or Virtual Box.

### Amazon Cloud Marketplace

Cisco Catalyst 8000V is available in the Amazon Cloud Marketplace. For more information, see the *Cisco Catalyst 8000V Edge Software Deployment Guide for Amazon Web Services*.

### Microsoft Azure Marketplace

Cisco Catalyst 8000V is available in the Microsoft Azure Marketplace . For more information, see the *Cisco Catalyst 8000V Edge Software Deployment Guide for Microsoft Guide*.

## Server Requirements

*Table 1: Server Requirements*

| Cisco Catalyst 8000V Release | Intel | AMD |
|---|---|---|
| Cisco IOS XE 17.4 and later | 64-bit Intel Core2 and later generation processors with VT extensions and support for Streaming SIMD instructions: SSE, SSE2, SSE3 and SSSE3. | The equivalent of 64-bit Intel Core2 and later generation processors with VT extensions and support for Streaming SIMD instructions: SSE, SSE2, SSE3 and SSSE3. |

The Cisco Catalyst 8000V router uses instructions supported on Intel Core 2 and later generation processors including Streaming SIMD SSE, SSE2, SSE3 and SSSE3. The existence of the required streaming SIMD instruction sets is determined at boot time. If the required instructions are not present, the system displays the following message or a similar one:

```
%CPPDRV-3-FATAL_CPU_FEATURE: F0: cpp_driver: CPP0: CPU lacks feature

(Supplemental Streaming SIMD Extensions 3 (SSSE3)). Packet forwarding disabled.
```

For more information, see the latest Cisco Catalyst 8000V release notes.

# Supported Cisco IOS XE Technologies

The Cisco Catalyst 8000V Router supports selected Cisco IOS XE technologies. The Cisco Catalyst 8000V supports a more limited set of functionality compared to the other router platforms.

The following table lists the major Cisco IOS XE technologies that Cisco Catalyst 8000V supports. Technologies not listed here are not currently supported on this router.

Not all the features in a given technology may be supported. To verify support for specific features, use the Cisco Feature Navigator. For more information, see the *Using the Cisco Feature Navigator* section.

The information listed in this table is applicable only when you use the Cisco IOS XE CLI. Support for Cisco IOS XE technologies is more limited in the following scenarios:

- When you deploy a Cisco Catalyst 8000V instance on Amazon Web Services (AWS).
- When you deploy a Cisco Catalyst 8000V instance on Microsoft Azure.

*Table 2: Cisco IOS XE Technologies Supported on Cisco Catalyst 8000V*

| Technologies Supported | Technology Package Licenses Supported in Cisco IOS XE 17.4 and Later |
|---|---|
| IP: | |
| • IPv4 Routing<br>• IPv4 Fragmentation and Reassembly<br>• IPv6 Forwarding | • IPBase<br>• Security<br>• AX<br>• APPX |

| Technologies Supported | Technology Package Licenses Supported in Cisco IOS XE 17.4 and Later |
|---|---|
| • IPv6 Routing | • IPBase<br>• Security<br>• AX<br>• APPX |
| • Generic Routing Encapsulation (GRE) | • IPBase<br>• Security<br>• AX<br>• APPX |
| • LISP | • AX<br>• APPX |
| • Connectionless mode network service (CLNS) | • IPBase<br>• Security<br>• AX<br>• APPX |
| Basic Routing: | |
| • BGP | • IPBase<br>• Security<br>• AX<br>• APPX |
| • EIGRP | • IPBase<br>• Security<br>• AX<br>• APPX |
| • ISIS | • IPBase<br>• Security<br>• AX<br>• APPX |

| Technologies Supported | Technology Package Licenses Supported in Cisco IOS XE 17.4 and Later |
|---|---|
| • OSPF | • IPBase<br>• Security<br>• AX<br>• APPX |
| • Performance Routing | • IPBase<br>• Security<br>• AX<br>• APPX |
| IP Multicast: | |
| • IGMP | • Security<br>• AX |
| • PIM | • Security<br>• AX |
| IP Switching: | |
| • Cisco Express Forwarding | • IPBase<br>• Security<br>• AX<br>• APPX |
| Wide Area Networking: | |
| • OTV<br>(Supported beginning in Cisco IOS XE 3.10S.) | • AX<br>• APPX |
| • VxLAN<br>(Supported beginning in Cisco IOS XE 3.11S.) | • AX<br>• APPX |
| • WCCPv2 | • AX<br>• APPX |
| VPN: | |

| Technologies Supported | Technology Package Licenses Supported in Cisco IOS XE 17.4 and Later |
|---|---|
| • IPsec VPN | • Security<br>• AX |
| • DMVPN | • Security<br>• AX |
| • Easy VPN | • Security<br>• AX |
| • FlexVPN | • Security<br>• AX |
| • GETVPN<br>(Supported beginning in Cisco IOS XE Everest 16.6.1) | • Security<br>• AX |
| • SSL VPN | • Security<br>• AX |
| MPLS: | |
| • MPLS | • APPX<br>• AX |
| • EoMPLS | • APPX<br>• AX |
| • VRF | • IPBase |
| • VPLS<br>(Supported beginning in Cisco IOS XE 3.10S.) | • APPX<br>• AX |
| Network Management: | |
| • SNMP | • IPBase<br>• Security<br>• AX<br>• APPX |

| Technologies Supported | Technology Package Licenses Supported in Cisco IOS XE 17.4 and Later |
|---|---|
| • Flexible NetFlow | • IPBase<br>• Security<br>• AX<br>• APPX |
| • Secure Shell (SSH) | • IPBase<br>• Security<br>• AX<br>• APPX |
| QoS: | |
| • QoS | • IPBase<br>• Security<br>• AX<br>• APPX |
| Services: | |
| • NAT | • IPBase<br>• Security<br>• AX<br>• APPX |
| Access Control: | |
| • AAA | • IPBase<br>• Security<br>• AX<br>• APPX |
| • Access Control Lists | • IPBase<br>• Security<br>• AX<br>• APPX |

| Technologies Supported | Technology Package Licenses Supported in Cisco IOS XE 17.4 and Later |
|---|---|
| • IP SLA | • AX<br>• APPX |
| • RADIUS | • IPBase<br>• Security<br>• AX<br>• APPX |
| • TACACS+ | • IPBase<br>• Security<br>• AX<br>• APPX |
| • Layer3 Firewall | • Security<br>• AX |
| • Zone-Based Firewall | • Security<br>• AX |
| • Zone-Based Firewall Multi-tenancy for Cloud Integrated Security Solution<br>(Supported starting with Cisco IOS XE Denali 16.4.1.) | • Advanced<br>• Premium |
| Application Services: | |
| • Application Visibility and Control (AVC) | • AX<br>• APPX |
| • NBAR2 | • AX<br>• APPX |
| Broadband: | |
| • Broadband Network Gateway | • APPX<br>(Requires broadband add-on feature license (L-CSR-BB-1K=). |

| Technologies Supported | Technology Package Licenses Supported in Cisco IOS XE 17.4 and Later |
|---|---|
| • Intelligent Services Gateway | • APPX<br>(Requires broadband add-on feature license (L-CSR-BB-1K=). |
| Redundancy: | |
| • HSRP | • IPBase<br>• Security<br>• AX<br>• APPX |
| WAAS: | |
| • Integrated AppNav-XE | • AX<br>• APPX |

# Management Support

## Managing the Router Using Cisco Configuration Professional

You can manage Cisco Catalyst 8000V routers using the Cisco Configuration Professional. The minimum version required is Cisco Configuration Professional 2.8.

For more information, see the Cisco Configuration Professional documentation.

# Cisco Unified Computing System (UCS) Products

*Table 3: Cisco Catalyst 8000V Compatibility with Cisco UCS Servers*

| | |
|---|---|
| Cisco Unified Computing System (UCS) Products | The Cisco UCS server requirements are:<br>• VMware-certified<br>• 4 or more cores configured<br>• 6 GB or more memory<br>• VMware vCenter or standalone VMware vSphere client installed to manage the ESXi server<br>See http://www.cisco.com/en/US/products/ps10477/prod_technical_reference_list.html to determine the UCS hardware and software that is compatible with the supported hypervisors. |

# Finding Support Information for Platforms and Cisco Software Images

Cisco software is packaged in feature sets consisting of software images that support specific platforms. The feature sets available for a specific platform depend on which Cisco software images are included in a release. To identify the set of software images available in a specific release or to find out if a feature is available in a given Cisco IOS XE software image, you can use Cisco Feature Navigator, the Software Advisor, or the Cisco Catalyst 8000V Release Notes.

## Using Cisco Feature Navigator

Use the Cisco Feature Navigator to find information about platform support and software image support. The Cisco Feature Navigator enables you to determine the Cisco IOS XE software image support for a specific software release, the feature set, or the platform. To access the Cisco Feature Navigator, go to http://www.cisco.com/go/cfn . An account on Cisco.com is not required to access this site.

## Using the Software Advisor

The Software Advisor tool enables you to:

- See if a feature is supported in a Cisco IOS XE release
- Locate the software document for a feature
- Check the minimum Cisco IOS XE software requirements for your router

You can access this tool by visiting http://tools.cisco.com/Support/Fusion/FusionHome.do. You must be a registered user on Cisco.com to access this tool.

# Using the Software Release Notes

The Cisco IOS XE software release notes provide the following information:

- Platform support
- Memory recommendations
- Information about new features and enhancements
- Open and resolved severity 1 and 2 bugs

Release notes are intended to be release-specific for the most current release, and the information provided in these documents may not be cumulative. That is, the latest release notes document will not provide information about features that first appeared in previous releases. For more information, see the Cisco Catalyst 8000V Release Notes.

For cumulative feature information, see the Cisco Feature Navigator.

CHAPTER 2

# Using Cisco IOS XE Software

This chapter provides information about the Cisco IOS XE software used to configure Cisco Catalyst 8000V. The software for Cisco Catalyst 8000V uses standard Cisco IOS XE CLI commands and conventions.

Commands are not case sensitive. You can abbreviate the commands and parameters if the abbreviations contain enough letters to be different from any other currently available commands or parameters.

The following table lists the keyboard shortcuts for entering and editing commands.

*Table 4: Keyboard Shortcuts*

| Keystrokes | Purpose |
|---|---|
| **Ctrl-B** or the **Left Arrow** key | Move the cursor back by one character. |
| **Ctrl-F** orthe **Right Arrow** key | Move the cursor forward by one character. |
| **Ctrl-A** | Move the cursor to the beginning of the command line. |
| **Ctrl-E** | Move the cursor to the end of the command line. |
| **Esc B** | Move the cursor back by one word. |
| **Esc F** | Move the cursor forward by one word. |

The history buffer stores the last 10 commands you entered. History substitution allows you to access these commands without retyping them, by using special abbreviated commands.

*Table 5: History Substitution Commands*

| Command | Purpose |
|---|---|
| **Ctrl-P** or the **Up Arrow** key | Recall commands in the history buffer beginning with the most recent command. Repeat the key sequence to recall successively older commands. |
| **Ctrl-N** or the **Down Arrow** key | Return to more recent commands in the history buffer after recalling commands with **Ctrl-P** or the **Up Arrow** key. |

| Command | Purpose |
|---|---|
| Router# **show history** | While in the EXEC mode, list the last several commands you have just entered. |

The command modes available in the traditional Cisco IOS CLI are exactly the same as the command modes available in Cisco IOS XE.

Use the CLI to access Cisco IOS XE software. Because the CLI is divided into many different modes, the commands available to you at any given time depend on the mode that you are currently in. Entering a question mark (**?**) at the CLI prompt allows you to obtain a list of commands available for each command mode.

When you log in to the CLI, you are in the user EXEC mode. The user EXEC mode contains only a limited subset of commands. To have the access to all the commands, enter the privileged EXEC mode normally by using a password. From the privileged EXEC mode, you can issue any EXEC command - user or privileged mode - or you can enter the global configuration mode.

Most EXEC commands are one-time commands. For example, **show** commands show important status information and **clear** commands clear counters or interfaces. The EXEC commands are not saved when the software reboots.

Configuration modes allow you to make changes to the running configuration. If you later save the running configuration to the startup configuration, these changed commands are stored when the software is rebooted. To enter specific configuration modes, you must start at the global configuration mode. From the global configuration mode, you can enter the interface configuration mode and a variety of other modes such as protocol-specific modes.

The following table describes how to access and exit various common command modes of the Cisco IOS XE software. It also shows examples of the prompts displayed for each mode.

*Table 6: Accessing and Exiting Command Modes*

| Command Mode | Access Method | Prompt | Exit Method |
|---|---|---|---|
| User EXEC | Log in. | `Router>` | Use the **logout** command. |
| Privileged EXEC | From the user EXEC mode, use the **enable** EXEC command. | `Router#` | To return to the user EXEC mode, use the **disable** command. |
| Global configuration | From the privileged EXEC mode, use the **configure terminal** privileged EXEC command. | `Router(config)#` | To return to the privileged EXEC mode from global configuration mode, use the **exit** or **end** command. |
| Interface configuration | From the global configuration mode, specify an interface using an **interface** command. | `Router(config-if)#` | To return to the global configuration mode, use the **exit** command.<br>To return to the privileged EXEC mode, use the **end** command. |

Entering a question mark (**?**) at the CLI prompt displays a list of commands available for each command mode. You can also get a list of keywords and arguments associated with any command by using the context-sensitive help feature.

To view the help specific to a command mode, a command, a keyword, or an argument, use one of the commands listed in the following table.

*Table 7: Help Commands and Purpose*

| Command | Purpose |
|---|---|
| help | Provides a brief description of the help system in any command mode. |
| **abbreviated-command-entry?** | Provides a list of commands that begin with a particular character string. (No space between command and question mark.) |
| **abbreviated-command-entry<Tab>** | Completes a partial command name. |
| **?** | Lists all the commands available for a particular command mode. |
| **command ?** | Lists the keywords or arguments that you must enter next on the command line. Enter space between command and question mark. |

- NVRAM File Security, on page 15

# NVRAM File Security

Cisco Catalyst 8000V allows you to encrypt some of the disk partitions internal to the VM to provide extra security around sensitive data that may be stored on the routers. For example, information in the NVRAM is encrypted so that it is not visible to administrative entities with access to the physical hard disk upon which Cisco Catalyst 8000V is stored.

You can search and filter the output of **show** and **more** commands. This functionality is useful if you need to sort through large amounts of output or if you want to exclude output that you need not see.

To use this functionality, enter a **show** or **more** command followed by the “pipe” character ( | ); one of the keywords **begin**, **include**, or **exclude**; and a regular expression on which you want to search or filter (the expression is case sensitive):

**show** *command* { **append** | | **begin** | | **exclude** | | **exclude** | | **include** | | **redirect** | | **section** | | **tee** } *regular-expression*

**show** *command* | {**append** | **begin** | **exclude** | **include** | **redirect** | **section** | **tee**} *regular-expression*

The output matches certain lines of information in the configuration file.

To power off a Cisco Catalyst 8000V instance, you must power off the VM upon which the router is installed. For information about powering off the VM, see your VM vendor documentation.

CHAPTER 3

# Installation Overview

Cisco hardware routers are normally shipped with the Cisco IOS XE software pre-installed. Because Cisco Catalyst 8000V Edge Software is not hardware-based, you must download the Cisco IOS XE software from Cisco.com and install the virtual router directly onto the virtual machine. However, as part of the initial installation process, you must first provision the attributes of the VM so that the Cisco Catalyst 8000V software can install and boot.

The following image shows the high-level tasks required to install Cisco Catalyst 8000V on the VM. The different installation options are dependent on the hypervisor being used. See the following sections for more information.

## Obtaining the Cisco Catalyst 8000V VM Image

**Step 1** Go to the Cisco Catalyst 8000V product page.

**Step 2** Click **Download Software**.

**Step 3** Select the router model.

**Step 4** Click **IOS XE Software**. By default, the recommended Cisco IOS XE release is selected.

**Step 5** In the list of available images, click **Download Now** or **Add to Cart**. Follow the instructions for downloading the software.

## Installation Files

The following software images are available for installing Cisco Catalyst 8000V on the supported hypervisors.

- .ova

Used for deploying the OVA template on the VM (in TAR format)

- .iso

Used for installing the software image on the VM (requires manually creating the VM)

- .qcow2

Used for installing the software image in KVM OpenStack environments.

- .run

Self-installing image used for installation in a KVM environment.

- .bin

These images are used for upgrading and downgrading the software only. For more information, see the *Prerequisites for the Software Upgrade Process* section and subsequent sections.

**Note** For Cisco Catalyst 8000V running on AWS, you can use the .bin file to upgrade the instance without having to recreate AWS EC2 instance from a new AMI. This inline upgrade process is not yet available on Microsoft Azure.

# Installation Options

Cisco Catalyst 8000V supports the following installation options:

- Deploy the OVA template on the VM: Uses the .ova file. This template creates a VM using recommended preset values. See *Deploying the Cisco Catalyst 8000V Using vSphere* and *Deploying the Cisco Catalyst 8000V OVA to the VM Using COT*.

  You can use the .ova file only for first-time installation. You cannot use this file for upgrading the Cisco IOS XE software version.

- Deploy the .ova file on the VM using the Common OVF Tool (COT): The COT application is included in the file package. However, to ensure that you are using the latest version of COT, download COT directly from the GitHub site https://github.com/glennmatthews/cot/blob/master/README.md.

  Using the COT application, you can customize the VM values and easily deploy the custom VM as part of the Cisco Catalyst 8000V installation process. For more information, see *Editing the Basic Properties of Cisco Catalyst 8000V Using vSphere.*

- Manually configure the VM using the .iso file: Uses the .iso file. You can install the .iso file on your host and manually create the VM using your hypervisor software. For example, if you are installing Cisco Catalyst 8000V on VMware, you would install the .iso file on the VMware ESXi host and manually create the VM using the vSphere GUI.

- Create the Cisco Catalyst 8000V instance in KVM using OpenStack: Uses the .qcow2 file. The qcow2 (QEMU Copy on Write) image format is used to create the Cisco Catalyst 8000V tenant in the KVM OpenStack cloud environment.

### Upgrading Cisco IOS XE Software

For information about upgrading the Cisco IOS XE software, see *Prerequisite for the Software Upgrade Process* and the subsequent sections.

### Installation Options and Requirements

The following table lists the installation options for the supported hypervisors and the minimum Cisco IOS XE software release required.

*Table 8: Supported Installation Options for Cisco Catalyst 8000V*

| Installation Option | VMware ESXi | Citrix XenServer | KVM | Microsoft Hyper-V |
|---|---|---|---|---|
| Deploy OVA Template Using OVA Wizard | Supported | Not supported | Not supported | Not supported |
| Deploy OVA Using COT | Supported | Not supported | Not supported | Not supported |
| Manually Configure VM Using .iso File | Supported | Supported | Supported | Supported |
| Create the KVM instance on OpenStack Using .qcow2 File | NA | NA | Supported | NA |

**Note** When a device is in the installation mode, formatting of the boot drive, bootflash/flash is not recommended. Formatting is blocked to ensure stability of the running image and to avoid any impact to upgrade of the software.

# Guidelines and Limitations

The following list specifies the general guidelines and restrictions before installing a Cisco Catalyst 8000V router in your network:

- Cisco Catalyst 8000V may properly function within a nested VM, but this is not tested nor supported.
- If the hypervisor does not support vNIC Hot Add/Remove, do not make any changes to the VM hardware (memory, CPUs, hard drive size, and so on) while the VM is powered on.
- The GigabitEthernet0 interface is no longer available. You can designate any interface as the management interface.
- You can access the Cisco IOS XE CLI either through the virtual VGA console or the console on the virtual serial port. You can select the console from the GRUB mode during the first-time installation, or you can change the console using the Cisco IOS XE **platform console** command after the router boots. For more information, see *Booting the Cisco Catalyst 8000V as the VM* section.

**Note** Some hypervisors may not support serial console access. Verify support using your hypervisor documentation.

# ROMMON and Cisco Catalyst 8000V

Cisco Catalyst 8000V, which is software-based, does not include a ROMMON image. This differs from many Cisco hardware-based routers. During the initial bootloader process, the installation script creates a clean version of the Cisco Catalyst 8000V software image known as the Golden Image and places it in a non-accessible partition. This clean version can be used if the software image is not working properly or is not bootable.

Although Cisco Catalyst 8000V does not include ROMMON, the platform does include a GNU GRand Unified Bootloader (GRUB)-based bootloader. The GRUB function on Cisco Catalyst 8000V provides limited functionalities compared to the ROMMON available on other Cisco platforms.

Some Cisco IOS XE commands such as **show version** may show references to ROMMON in the command output.

**Note** After Cisco Catalyst 8000V completes the first-time installation, you can configure the router to automatically enter the GRUB mode when the router is booted.

# VNF Secure Boot

The secure boot feature prevents malicious software applications and unauthorized operating systems from loading into the system during the system startup process. If the secure boot feature is enabled, only the authorized software applications boots up from the device. This feature ensures that 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. The UEFI (Unified Extensible Firmware Interface) specification defines a secure boot methodology that prevents loading software which is not signed with an acceptable digital signature.

To display the system boot mode and the bootloader version use **show platform software system boot** command.

```
Router#show platform software system boot
Boot mode: EFI
Bootloader version: 2.0
```

### Restrictions

- The following secure boot environments are supported:
  - ESXi version 6.5 or higher
  - KVM RHEL 7.5 using open stack license
  - NFVIS release 3.11 or later

- Only EFI firmware modes support the secure boot.
- This feature is supported on VMs created in Cisco IOS XE Gibraltar 16.12 or later releases.

  GRUB2 and new disk partition layout is available.

**Note** Each hypervisor has a unique process to enable secure boot for the guest VMs. Refer to hypervisor specific documentation to enable secure boot. A set of high-level hypervisor specific steps to enable secure boot are mentioned below.

### ESXi Secure Boot Setup

- Create VM using ESXi 6.5 or later version using VM version 13 or greater. To choose the EFI firmware mode, navigate through **VM Options** > **Boot Options** > **Firmware** > **EFI**.
- Power down the VM after the initial boot and IOS prompt is complete.
- Enable the EFI secure boot in **Edit Settings** > **VM Options** > **Boot Options** > **Secure Boot**.
- Power up VM and the VNF boots up securely.

### KVM Secure Boot Setup

- Create the VM.
- Power down the VM after the VM is created and VNF IOS prompt is complete.
- Install PK, KEK, and db certificates from the **EFI Firmware** menu and reset.

  To create the custom keys, see Custom Keys for Secure boot. For db certificates, see MicCorUEFCA2011_2011-06-27.crt and MicWinProPCA2011_2011-10-19.crt.
- Secure boot the VM.

### NFVIS Secure Boot Setup

- Upgrade to NFVIS 3.11 release or later.
- Register an Cisco Catalyst 8000V EFI tarball with the NFVIS repository.
- Create a VM using the registered EFI image.
- Secure boot the VM.

# Where to Go Next

See the following information about installing the Cisco Catalyst 8000V in different hypervisor environments:

- VMware ESXi Support Information
- Microsoft Hyper-V Support Information

- Citrix XenServer Support Information
- Kernel Virtual Machine Support Information

**Note** For information about deploying Cisco Catalyst 8000V in an Amazon Web Services environment, see the *Cisco Catalyst 8000V Edge Router Deployment Guide for Amazon Web Services*.

**Note** For information about deploying the Cisco Catalyst 8000V in a Microsoft Azure environment, see the *Cisco Catalyst 8000V Deployment Guide for Microsoft Azure*.

CHAPTER 4

# Installing in VMware ESXi Environment

This chapter contains information about VMware tools/software and the VM requirements for Cisco Catalyst 8000V software.

Cisco Catalyst 8000V can run on the VMware ESXi hypervisor. VMware ESXi runs on x86 hardware containing virtualization extension. You can use the same hypervisor to run several VMs.

VMware vSphere Web Client is a web application that runs on a x86 hardware containing virtualization extension and accesses the VMware vCenter Server. You can use VMware vSphere Web Client software to create, configure, and manage VMs on the vCenter Server and to start or stop the Cisco Catalyst 8000V instance. Cisco Catalyst 8000V boots from a virtual disk located on the data store.

**Note** If you upgrade VMware ESXi, and ESXi contains an existing Cisco Catalyst 8000V, the interfaces of the Cisco Catalyst 8000V may be renamed. For example, GigabitEthernet1 may appear as GigabitEthernet4. To recover the original interface names, perform the following two Cisco IOS XE configuration commands from the console or terminal of the Cisco Catalyst 8000V immediately after upgrading the VMware ESXi hypervisor:

**clear platform software vnic nvtable**

**reload**

To find out more about installing VMware vSphere products, see VMware product documentation .

# VMware Requirements

The following table specifies the supported VMware tools by Cisco Catalyst 8000V using Cisco IOS XE 17.4 and later releases:

| Cisco IOS XE Release | vSphere Web Client | vCenter Server |
|---|---|---|
| Cisco IOS XE 17.4.x releases | The 6.7 and 6.5 versions of the VMware vSphere Web Client are supported. | VMware ESXi 6.7 and ESXi 6.5 |

These versions have been fully tested and meet performance benchmarks.

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 16GB and 8 GB virtual disks are supported.

vCPUs - the following vCPU configurations are supported:

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

- 1 vCPU: requires minimum 4 GB RAM allocation
- 2 vCPUs: requires minimum 4 GB RAM allocation
- 4 vCPUs: requires minimum 4 GB RAM allocation
- 8 vCPUs: requires minimum 4 GB RAM allocation

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 - minimum size of 8 GB.

Virtual Network Interface Cards (vNICs) - Three or more vNICs (max. 10) - VMXNET3, iXGBeVF, and i40eVF.

A default video, SCSI controller set is required, and an installed virtual CD/DVD drive.

# Supported VMware Features and Operations

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 cause the runtime state of the VM to be saved and then restored upon restarting. If the runtime state includes traffic-related state, then on resumption or replaying the runtime state, additional errors, statistics, or messages are displayed on the user console. If the saved state is just configuration driven, you can use these features and operations without a problem.

The *Supported VMware Features and Operations: Storage Options (for Both vCenter Server and vSphere Client)* table lists the VMware features and operations that are supported on Cisco Catalyst 8000V. For more information about VMware features and operations, see the VMware Documentation .

The following VMware features and operations are not supported in all versions of Cisco Catalyst 8000V, but can still be used or performed on non-supported versions at the risk of encountering dropped packets, dropped connections, and other error statistics:

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

# General Features (vCenter Server)

*Table 9: Supported VMware Features and Operations: General Features (for vCenter Server Only)*

| Supported Entities | Description |
|---|---|
| Cloning | Enables cloning a virtual machine or template, or cloning a virtual machine to a template. |
| Migrating | 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. |

# Operations (for vCenter Server and vSphere Web Client)

*Table 10: Supported VMware Features and Operations: 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. |
| Shut Down | Shut Down, or “soft” power off, leverages 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 hung, shut down might not succeed and using the Power off option is necessary. |
| Suspend | Suspends the virtual machine. |
| Reset/Restart | Stops the virtual machine and restarts (reboots) 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. |

| Supported Entities | Description |
|---|---|
| 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 11: Supported VMware Features and Operations: Networking Features*

| Supported Entities | Description |
|---|---|
| Custom MAC address | From both vCenter Server and vSphere Client. Allows you to set up the MAC address manually for a virtual network adapter. |
| Distributed VSwitch | From vCenter Server only. A vSphere distributed switch on a vCenter Server data center can handle networking traffic for all associated hosts on the data center. |
| Distributed Resources Scheduler | Provides automatic load balancing across hosts. |
| NIC Load Balancing | From both vCenter Server and vSphere Client. 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. |
| NIC Teaming | From both vCenter Server and vSphere Client. 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>**Note** 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 | From both vCenter Server and vSphere Client. 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. |

# High Availability

**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 12: Supported VMware Features and Operations: High Availability*

| 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. |

## Storage Options (for vCenter Server and vSphere Web Client)

*Table 13: Supported VMware Features and Operations: Storage Options (for Both vCenter Server and vSphere Client)*

| Supported Entities | Description |
|---|---|
| Storage Options (for both vCenter Server and vSphere Client) | 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. |
| Local Storage | |
| External Storage Target | You can deploy the Cisco Catalyst 8000V instance on external storage. That is, a Storage Area Network (SAN). |
| Mount or Pass Through of 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.<br>• You can use only two USB disk hot-plug devices at a time.<br>• USB hub is not supported. |

# Deploying the Cisco Catalyst 8000V OVA to the VM

## Deploying the OVA to the VM

Perform the following steps in VMware vSphere Client:

**Step 1** Log in to the VMware vSphere Client.

**Step 2** From the vSphere Client Menu Bar, choose **File** > **Deploy OVF Template**.

**Step 3** In the OVA Wizard, point the source to the Cisco Catalyst 8000V OVA to be deployed. Click **Next**.

The system displays the OVF Template Details with the information about the OVA. Click **Next**.

**Step 4** Under **Name and Inventory Location**, specify the name for the VM and click **Next**.

**Step 5** Under **Deployment Configuration**, select the desired hardware configuration profile from the drop-down menu and click **Next**.

**Step 6** Under **Storage**, select the Datastore to use for the VM. Click **Next**.

**Step 7** Under **Disk Format**, select the disk format option:

- 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.

Click **Next**.

**Step 8** Under **Network Mapping**, 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.

**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 the pre-existing networking configuration is removed..

**Step 9** Select the vNIC to connect at **Power On**. 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.

**Step 10** Configure the properties for the VM.

**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.

**Note** The bootstrap properties are optional when creating the VM. You can set these properties to easily provision the VM before starting it up.

*Table 14: OVA Bootstrap Properties*

| Property | Description |
|---|---|
| Bootstrap Properties | |
| Console | Configures the console mode.<br>Possible values: virtual, serial |
| Login Username | Sets the login username for the router. |
| Login Password | Sets the login password for the router. |
| Management Interface | Designates the management interface for the Cisco Catalyst 8000V instance. The format must be GigabitEthernetx or GigabitEthernetx.xxx.<br>**Note** The GigabitEthernet0 interface is no longer supported. |
| Management vLAN | Configures the dot1Q VLAN interface. Requires the management interface to 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, this may be left blank. |
| PNSC IPv4 Address | 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. |
| Router name | Configures the hostname of the router. |
| Resource Template | Configures the Resource Template.<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. |

After you configure the router properties, click **Next**. The system displays the Ready to Complete screen with the settings to be used when the OVA is deployed.

You can also configure advanced properties after the router boots.

**Step 11** Select **Power On After Depolyment** to automatically power on the VM.

**Step 12** Click **Finish** to deploy the OVA.

The OVA deploys the .iso file, and if you select the **Power on after deployment** setting, the VM is automatically powered on. Once the VM is powered on, the Cisco Catalyst 8000V device begins the installation and boot process. If a bootstrap configuration file was included in the OVA, the router configuration is automatically enabled.

For more information, see *Booting the Cisco Catalyst 8000V and Accessing the Console*.

# Deploying the Cisco Catalyst 8000V OVA to the VM using vSphere

## Deploying the Cisco Catalyst 8000V OVA to the VM using vSphere

The Cisco Catalyst 8000V OVA file package allows you to deploy the Cisco Catalyst 8000V to the VM. The OVA package includes an OVF file that contains a default VM configuration based on the Cisco IOS XE release and the supported hypervisor.

**Note** The Citrix XenServer, KVM and Microsoft Hyper-V implementations do not support deploying the VM using the .ova file. You must manually install the VM using the .iso file.

## Restrictions and Requirements

The following restrictions apply when deploying the OVA package to the VM:

If the virtual CPU configuration is changed, you must reboot the Cisco Catalyst 8000V instance. Changing the RAM allocation does not require you to reboot the Cisco Catalyst 8000V instance.

The OVA package provides an option to select the virtual CPU configuration.

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.

## Deploying the OVA to the VM

Perform the following steps in VMware vSphere Client:

**Step 1** Log in to the VMware vSphere Client.

**Step 2** From the vSphere Client Menu Bar, choose **File** > **Deploy OVF Template**.

**Step 3** In the OVA Wizard, point the source to the Cisco Catalyst 8000V OVA to be deployed. Click **Next**.

The system displays the OVF Template Details with the information about the OVA. Click **Next**.

**Step 4** Under **Name and Inventory Location**, specify the name for the VM and click **Next**.

**Step 5** Under **Deployment Configuration**, select the desired hardware configuration profile from the drop-down menu and click **Next**.

**Step 6** Under **Storage**, select the Datastore to use for the VM. Click **Next**.

**Step 7** Under **Disk Format**, select the disk format option:

- 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.

Click **Next**.

**Step 8** Under **Network Mapping**, 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.

**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 the pre-existing networking configuration is removed..

**Step 9** Select the vNIC to connect at **Power On**. 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.

**Step 10** Configure the properties for the VM.

**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.

**Note** The bootstrap properties are optional when creating the VM. You can set these properties to easily provision the VM before starting it up.

***Table 15: OVA Bootstrap Properties***

| Property | Description |
|---|---|
| Bootstrap Properties | |
| Console | Configures the console mode.<br>Possible values: virtual, 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>**Note** The GigabitEthernet0 interface is no longer supported. |
| Management vLAN | Configures the dot1Q VLAN interface. Requires the management interface to 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, this may be left blank. |
| PNSC IPv4 Address | 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. |
| Router name | Configures the hostname of the router. |
| Resource Template | Configures the Resource Template.<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. |
| License Boot Level | Configures the license technology level that is available when the Cisco Catalyst 8000V instance boots. |

After you configure the router properties, click **Next**. The system displays the Ready to Complete screen with the settings to be used when the OVA is deployed.

You can also configure advanced properties after the router boots.

**Step 11** Select **Power On After Depolyment** to automatically power on the VM.

**Step 12** Click **Finish** to deploy the OVA.

The OVA deploys the .iso file, and if you select the **Power on after deployment** setting, the VM is automatically powered on. Once the VM is powered on, the Cisco Catalyst 8000V device begins the installation and boot process. If a bootstrap configuration file was included in the OVA, the router configuration is automatically enabled.

For more information, see *Booting the Cisco Catalyst 8000V and Accessing the Console*.

## Editing the Basic Properties of Cisco Catalyst 8000V using COT

Before you deploy Cisco Catalyst 8000V using COT, you can edit the basic or custom properties of the Cisco Catalyst 8000V VM in the OVA package using COT.

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

**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.

**-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.

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

### Editing the Basic Properties of Cisco Catalyst 8000V using COT [Sample]

```
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                    ""
```

The following table specifies the **cot edit-properties** command and arguments used in the above example.

| Script Step | Description |
|---|---|
| `cot edit properties c8000v-universalk9.ova` | Edits the basic environment properties of the OVA file. |
| `-p "login-username=cisco"` | Sets the bootstrap login username. |

| Script Step | Description |
|---|---|
| `-p "login-password=cisco"` | Sets the bootstrap login password. |
| `-o "c8000v-universalk9-customized.ova"` | Saves a modified OVA, which contains configuration commands from the text file. |

## Editing the Custom Properties of Cisco Catalyst 8000V using COT

Before you deploy the Cisco Catalyst 8000V AM using COT, you can edit custom properties. For example, to include Cisco IOS XE CLI commands. To edit the custom properties of the OVA, use one of the following two commands:

- **cot edit-properties**
- **cot inject-config**

# Deploying the Cisco Catalyst 8000V to the VM using COT

## Deploying the Cisco Catalyst 8000V OVA to the VM using COT

The Cisco Catalyst 8000V OVA file package allows you to deploy the Cisco Catalyst 8000V to the VM. The OVA package includes an OVF file that contains a default VM configuration based on the Cisco IOS XE release and the supported hypervisor. You can deploy the OVA using VMware vSphere or COT or the Common OVF Tool. This section describes how to deploy using the COT.

The Common OVF Tool (COT) included in the Cisco Catalyst 8000V software package is a Linux-based application that enables you to create attributes for one or more VMs and quickly deploy VMs with the Cisco Catalyst 8000V software pre-installed. This tool can speed the process of deploying Cisco Catalyst 8000V on multiple VMs.

COT provides a simple command-line interface to enter the VM attributes into the .ova file. You can run COT either in a LINUX shell or on Mac OS X. However, ensure that VMware ovftools are installed.

**Danger** The Common OVF Tool (COT) is provided without official Cisco support. Use it at your own risk.

### COT Restrictions

- COT supports deployment of the OVA package directly onto an ESXi host. The tool does not support Citrix XenServer, KVM or Microsoft Hyper-V environments.

## Downloading COT

Download and install the COT libraries and script according to the instructions provided in the http://cot.readthedocs.io/en/latest/installation.html GitHub site.

# Editing the Basic Properties of Cisco Catalyst 8000V using COT

Before you deploy Cisco Catalyst 8000V using COT, you can edit the basic or custom properties of the Cisco Catalyst 8000V VM in the OVA package using COT.

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

**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.

**-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.

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

### Editing the Basic Properties of Cisco Catalyst 8000V using COT [Sample]

```
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                  ""
```

The following table specifies the **cot edit-properties** command and arguments used in the above example.

| Script Step | Description |
|---|---|
| `cot edit properties 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, which contains configuration commands from the text file. |

## Editing the Custom Properties of Cisco Catalyst 8000V using vSphere

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, do the following:

**Step 1** In the vSphere GUI, select the **Options** tab.

**Step 2** Select **vApp Options** > **Advanced**.

**Step 3** In the Advanced Property Configuration screen, click the **Properties** button.

**Step 4** Click **New** to add a property.

**Step 5** In the Edit Property Settings screen, enter the information to create the new custom property based on a 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) (Optional) Enter the label. This is a descriptive string for the property.
b) Enter the class ID as “com.cisco.csr1000v”.
c) Assign the property an ID of “ios-config-xxxx” where xxxx is a sequence number from 0001 to 9999 that determines the order in which the custom properties are applied.
d) (Optional) 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.

**Step 6** Click **OK**.

**Step 7** In the Advanced Property Configuration screen, click **OK**.

**Step 8** Reboot the Cisco Catalyst 8000V instance.

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

## cot edit-properties

Use the **cot edit-properties** command to pre-apply a small number of configuration commands to the OVA.

To use more commands, use the **cot inject-config** command.

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

### Synopsis and Description

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

**-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.

### Example

In this example, a previously created text file, iosxe_config.txt, containing IOS XE config commands is added to the OVA using the **cot edit-properties** command. Finally, the **cot info** command is used to show the modified 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 edit-properties c8000v-universalk9.ova \
     -o c8000v-universalk9-customized.ova \
     -c iosxe_config.txt
$ cot info c8000v-universalk9-customized.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"
```

The following table specifies the **cot edit properties** 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"` | 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 OVF. |

## cot inject-config

Use the **cot inject-config** command if you have a large set of configuration commands to pre-apply to the OVA. For example, if you want to add a complete running configuration. This is efficient in terms of file size and loading time as this command uses plain text for the configuration commands (instead of XML). For further details about the **cot inject-config** command, seehttp://cot.readthedocs.io/en/latest/usage_inject_config.html

### Synopsis and Description

cot inject-config ova-filename

**-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.

### Example

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
```

The following 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" | The name of the new or the modified OVA, containing the config commands from the text file. |
| -c iosxe_config.txt | The name of the text file that contains the IOS XE configuration commands. |

## Deploying the Cisco Catalyst 8000V VM using COT

To deploy the Cisco Catalyst 8000V VM, use the **cot deploy ... esxi** command as shown in the following step. Note that the following description provides general guidance. The exact steps that you need to perform may vary depending on the characteristics of your VMware environment and setup.

Run the **cot deploy ... 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.

## Example

The table below shows an example **cot deploy** command, and its arguments, that is used to deploy a Cisco Catalyst 8000V VM in a vCenter environment.

| Script Step | Description |
|---|---|
| `cot deploy` | |
| `-s '10.122.197.5/UCS/host/10.122.197.38'` | vCenter server 10.122.197.5, target host UCS/host/10.122.197.38 |
| `-u administrator -p password` | Credentials for the ESXi server. If unspecified, COT will use your userid and prompt for a password. |
| `-n XE3.13` | Name of the newly created Cisco Catalyst 8000V VM. |
| `-c 1CPU-4GB` | OVF hardware config profile. If this is not specified, COT displays a list of available profiles and prompts you to select one. |
| `-N "GigabitEthernet1=VM Network"`<br>`-N "GigabitEthernet2=VM Network"`<br>`-N "GigabitEthernet3=VM Network"` | Mapping each NIC in the Cisco Catalyst 8000V OVA to a vSwitch on the server. |
| `esxi` | Target hypervisor (currently always ESXi) |
| `~/Downloads/csr1000v-universalk9.ova` | OVA to deploy |
| `-ds=datastore38a` | Any ESXi-specific parameters - here, the datastore to use for disk storage. |

# Manually Creating the VM and Installing the Cisco Catalyst 8000V Software Using the .iso File (VMware ESXi)

## Overview of Tasks for Manually Creating the VM

The image in this section shows the typical high-level tasks required to manually create a Cisco Catalyst 8000V VM. The specific procedures, terminology and the order the steps are performed may differ depending on the hypervisor being used. See the sections following for detailed steps for creating the VM.

## Manually Creating the VM Using the .iso File (VMware ESXi)

The following steps are performed using VMware VSphere.

- Location: Store with the virtual machine

While the following procedure provides general guidance for how to deploy Cisco Catalyst 8000V, the exact steps that you need to perform may vary depending on the characteristics of your VMware environment and setup. The steps and screen displays in this procedure are based on VMware ESXi 5.0.

**Step 1** Download the C8000V_esxi.iso file from the Cisco Catalyst 8000V software installation image package and copy it onto the VM Datastore.

**Step 2** In the VSphere client, select **Create a New Virtual Machine** option.

**Step 3** Under **Configuration**, select the option to create a Custom configuration, and click **Next**.

**Step 4** Under **Name and Location**, specify the name for the VM and click **Next**.

**Step 5** Under **Storage**, select the datastore to use for the VM. Click **Next**.

**Step 6** Under **Virtual Machine Version**, select **Virtual Machine Version 13**. Click **Next**.

**Note** Cisco Catalyst 8000V is not compatible with ESXi Server versions prior to 5.0.

**Step 7** Under **Guest Operating System**, select **Linux** and the **Other 3.x Linux (64-bit)** setting from the drop-down menu. Click **Next**.

**Step 8** Under **CPUs**, select the following settings:

- Number of virtual sockets (virtual CPUs)
- 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.

Click **Next**.

**Step 9** Under **Memory**, configure the supported memory size for your **Cisco Catalyst 8000V** release. Click **Next**.

**Step 10** Under **Network**, 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 5.0 interface only allows the creation of 4 vNICS during the initial VM creation. You can add more vNICs after the VM is created and you boot the Cisco Catalyst 8000V 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.
d) Click **Next**.

**Note** 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.

**Step 11** Under **SCSI Controller**, select **VMware Paravirtual**. Click **Next**.

**Step 12** Under **Select a Disk**, click **Create a New Virtual Disk**.

**Step 13** Under **Create a Disk**, select the following:

- Capacity: Disk Size

  See the requirements sections in this guide for the virtual hard disk size required in your release.

- Disk Provisioning: select one of the following: Thick Provision Lazy Zeroed or 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.

- Location: Store with the Virtual Machine

Click **Next**.

**Step 14** Under **Advanced Options**, select **SCSI (0:0)** for the virtual device node.

**Step 15** On the Ready to Complete screen, click the **Edit the Virtual Machine** settings before completion. Click the **Continue** checkbox.

**Step 16** In the Hardware tab, click **New CD/DVD Drive**.

a) Select the **Device Type** that the VM will boot from:

Select 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.

**Step 17** In the **Resources** tab, click the **CPU** setting:

Set the **Resource Allocation** setting to **Unlimited**.

**Step 18** Click **OK**.

**Step 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. See the *Booting the Cisco Catalyst 8000V and Accessing the Console* section.

**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 booting the router.

# Increasing Performance on VMware ESXi Configurations

You can improve the performance of Cisco Catalyst 8000V running on ESXi environment by modifying the settings on the host and the virtual machine.

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

For more information about the VMware best practices for versions 6.7 and 6.5, see https://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/techpaper/performance/Perf_Best_Practices_vSphere65.pdf and https://www.vmware.com/techpapers/2019/vsphere-esxi-vcenter-server-67U2-performance-best-practices.html.

### Modifications to the Host Configuration

To improve the performance of the VMware ESXi configuration, perform the following modifications in the host configuration:

- Select the **High Performance** option under **Power Management**.
- Disable **Hyperthreading**.
- Enable SR-IOV for the supported physical adapters.

### Modifications to the Virtual Machine Configuration

To improve the performance of the VMware ESXi configuration, perform the following modifications in the host configuration:

- 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.
- Select **VMware Paravirtual** from **Virtual Hardware: SCSI Controller**.
- From the **Virtual Hardware: Network Adapter: Adapter Type** option, select SR-IOV for the supported NICs
- Set the **General Guest OS Version** > **VM Options** option to **Other 3.x or later Linux (64-bit)**.
- Set the **VM Options** option under **Advanced Latency Sensitivity** to High.
- Under **VM Options** > **Advanced Edit Configuration**, add “numa.nodeAffinity” to the same NUMA node as the SRIOV NIC.

CHAPTER 5

# Installing in KVM Environments

Cisco Catalyst 8000V supports the following Linux/KVM environments:

- Red Hat Enterprise Linux (RHEL)
- Red Hat Enterprise Virtualization (RHEV)
- Ubuntu (beginning with Cisco IOS XE Release 3.11S)

Red Hat Enterprise Linux (RHEL), an enterprise virtualization product produced by Red Hat, based on the Kernel-based Virtual Machine (KVM), is an open source, full virtualization solution for Linux on x86 hardware, containing virtualization extensions.

The Red Hat Enterprise Virtualization (RHEV) platform is a commercially packaged virtualization platform from Red Hat.

For more information on the KVM products and versions supported, see the *Installation Requirements for KVM* section in this chapter.

Cisco Catalyst 8000V installation on KVM requires the manual creation of a VM and installation using the .iso file or the qcow2 file. Deploying the OVA template into a KVM environment is not supported.

Cisco Catalyst 8000V supports the Virtio vNIC type on the KVM implementation. KVM supports a maximum of 26 vNICs.

## KVM Support on OpenStack

Cisco Catalyst 8000V supports the OpenStack environment. OpenStack support requires the .qcow2 installation file to be available on the Cisco.com download page.

For more information, see *Creating the Instance Using the OpenStack Command Line Tool* and *Creating the Instance Using the OpenStack Dashboard* sections in this chapter.

# Installation Requirements for KVM

The KVM requirements for Cisco Catalyst 8000V using Cisco IOS XE 17.4.x releases and later are as follows:

**KVM Versions**

| Cisco IOS XE Release | KVM Version |
|---|---|
| Cisco IOS XE 17.4.1 release | Linux KVM based on Red Hat Enterprise Linux 7.5 and 7.7 are recommended for release 17.4.1 - tested and meets performance benchmarks. |

- vCPUs. The following vCPU configurations are supported:
  - 1 vCPU: requires minimum 4 GB RAM allocation
  - 2 vCPUs: requires minimum 4 GB RAM allocation
  - 4 vCPUs: requires minimum 4 GB RAM allocation
- Virtual CPU cores - 1 vCPU is required
- Virtual hard disk size - 8 GB minimum
- Supported vNICs - Virtio, ixgbevf, or i40evf

**Note** If a vNIC with an i40evf driver is used, the maximum number of physical VLANs is limited to 512, shared across all VFs, and the number of VLANs for a VF can be further limited by the host (PF) driver for untrusted VFs. The latest Intel i40e PF driver limits untrusted VFs to a maximum of 8 VLANs/sub-interfaces.

- Maximum number of vNICs supported per VM instance - 26
- Virtual CD/DVD drive installed (applicable only when installing using an .iso file) - required

# Creating a Cisco Catalyst 8000V KVM Instance

## Creating the Cisco Catalyst 8000V VM Using the virt-manager GUI Tool

### Creating the Cisco Catalyst 8000V VM Using virt-manager with qcow2 or ISO Image

**Before you begin**

Download and install the virt-manager RPM package on the KVM server.

Download the .qcow2 or .iso image from the Cisco Catalyst 8000V software installation image package and copy it onto a local or network device.

**Step 1** Launch the virt-manager GUI. Click **Create a New Virtual Machine**.

**Step 2** Do one of the following: ( (

a) For .qcow2, select **Import Existing Disk Image**.
b) For .iso, select **Local Install Media (ISO Image or CDROM)**.

**Step 3** Select the Cisco Catalyst 8000V qcow2 or iso file location.

**Step 4** Configure the memory and CPU parameters.

**Step 5** Configure virtual machine storage.

**Step 6** Click **Finish**.

**Note** To add additional hardware before creating the VM, select **Customize configuration before install** before clicking **Finish**. If you select this option, the next screen displays an **Add Hardware** button that can be used one or more times to add various hardware options, such as additional disks or a serial port interface (see the following sections).

**Step 7** Access the Cisco Catalyst 8000V console by using one of the following:

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

## Creating the Cisco Catalyst 8000V VM Using virt-manager

Enables access to the Cisco Catalyst 8000V by adding a serial console.

### SUMMARY STEPS

1. Click **Add Hardware**.
2. Select the **Serial** option from the menu.
3. From the **Device Type** drop-down menu, 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**.

### DETAILED STEPS

**Step 1** Click **Add Hardware**.

**Step 2** Select the **Serial** option from the menu.

**Step 3** From the **Device Type** drop-down menu, select **TCP net console (tcp)**.

**Step 4** Specify the port number, and select the **Use Telnet** checkbox.

**Step 5** Click **Finish**.

**Step 6** After adding all necessary hardware, click **Begin Installation**.

## Creating the Cisco Catalyst 8000V VM Using Virtual Manager (Add Hard Disk)

Describes the optional steps after selecting the **Customize Configuration Before Install** option.

### Before you begin

- Perform the task by using a .qcow2 or an .iso image. Before you click **Finish**, select the **Customize configuration before install** option . The **Add Hardware** button appears.

**Step 1** Click **Add Hardware**.

**Step 2** Select the **Storage** option.

**Step 3** Select the **Select Managed Or Other Existing Storage** checkbox

**Step 4** (Applicable only when adding a Bootstrap Day0 configuration) Click the **Browse** button and navigate to the **csr_config.iso** location.

**Step 5** From the **Device-type** drop-down menu, select the **IDE CDROM** option.

**Step 6** Click **Finish**.

**Step 7** After adding all the necessary hardware, click **Begin Installation**.

# Creating a Bootstrap Day0 Configuration for virt-manager

This procedure provides additional steps to be executed when you create a virtual machine using the virtual manager.

The following steps are performed on the KVM server.

**Step 1** Create the **iosxe_config.txt** or the **ovf-env.xml** file. To know how to create these files, see the *Bootstrap Properties* section in this guide.

**Step 2** Create a disk image from this file using the command mentioned below:

**Example:**

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

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

# Creating the Cisco Catalyst 8000V VM Using virt-install with qcow2 Image

- 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 and copy it onto a local or network device.

Using the virt-install command, create the instance and boot, using the following syntax:

**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
```

(Optional) To configure a Bootstrap Day0 configuration, perform the steps described in the *Creating a Bootstrap Day0 Configuration for virt-install* section.

After the installation is complete, the Cisco Catalyst 8000V VM will be shutdown. You can start the Cisco Catalyst 8000V VM using the **virsh start** command.

### What to do next

**Red Hat Enterprise Linux - Setting Host Mode**

Due to an issue specific to Red Hat Enterprise Linux, when you launch Cisco Catalyst 8000V in a Red Hat Enterprise Linux environment using **virt-install**, set the host mode as follows:

- In Red Hat Enterprise Linux 6, use:

  ```
  --cpu host
  ```

- In Red Hat Enterprise Linux 7, use:

  ```
  --cpu host-model
  ```

# Creating the Cisco Catalyst 8000V VM Using virt-install with ISO Image

### Before you begin

- Download and install the virt-install RPM package on the KVM server.
- Download the **.iso** image from the Cisco Catalyst 8000V software installation image package and copy it onto a local or network device.

**Step 1** Create an 8G disk image in the **.qcow2** format using the **qemu-img** command.

**Example:**

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

**Step 2** Use the **virt-install** command to install the Cisco Catalyst 8000V instance. This requires the correct permissions to create a new VM. The following example creates a 1 vCPU Cisco Catalyst 8000V with 4G of RAM, one network interface, and one serial port.

**Example:**

```
virt-install               \
 --connect=qemu:///system  \
 --name=my_csr_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_csr1000v_iso>           \
 --disk path=csr_disk.qcow2,bus=virtio,size=8,sparse=false,cache=none,format=qcow2   \
 --network bridge=virbr0,model=virtio     \
 --noreboot
```

(Optional) To configure a Bootstrap Day0 configuration, perform the steps described in *Creating a Bootstrap Day0 Configuration for virt-install*.

The **virt-install** command creates a new VM instance and the Cisco Catalyst 8000V installs the image onto the specified disk file. After the installation is complete, the Cisco Catalyst 8000V VM is shutdown. You can start the Cisco Catalyst 8000V VM using the **virsh start** command.

**Step 3**

### What to do next

**Red Hat Enterprise Linux—Setting Host Mode**

Due to an issue specific to Red Hat Enterprise Linux, when launching the Cisco Catalyst 8000V in a Red Hat Enterprise Linux environment using **virt-install**, set the host mode as follows:

- In Red Hat Enterprise Linux 6, use:

  ```
  --cpu host
  ```

- In Red Hat Enterprise Linux 7, use:

  ```
  --cpu host-model
  ```

# Creating a Bootstrap Day0 Configuration for virt-install

This procedure provides additional steps to execute within one of the following procedures, as noted within the procedures:

- Creating the Cisco Catalyst 8000V Using virt-install with qcow2 Image
- Creating the Cisco Catalyst 8000V VM Using virt-install With ISO Image

**Note** Cisco Catalyst 8000V reads the day 0 configuration during the initial bootup and saves the configuration after bootup.

In order to bootstrap, perform the following steps on the KVM server.

**Step 1** Create an **iosxe_config.txt** or **ovf-env.xml** file. For more information, see the *Bootstrap Properties* section.

**Step 2** Create a disk image from the file using following command:

**Example:**

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

**Step 3** Perfrom this step within the VM creation procedure (see the options indicated above).

Add an additional disk parameter to the **virt-install** command to include the csr_config.iso disk image, as follows:

**Example:**

```
--disk path=/my/path/csr_config.iso,device=cdrom,bus=ide
```

# Creating the Cisco Catalyst 8000V on OpenStack

## Selecting a Cisco Catalyst 8000V Installation Image

There are two different installation image packages available for downloading. Each package contains a different qcow2 file.

- `csr1000v-universalk9.16.03.01a.qcow2`

  Choose this type of image to use a virtual console. This is recommended for later use with the OpenStack dashboard.

- `csr1000v-universalk9.16.03.01a-serial.qcow2`

  Choose this type of image to use a serial console to access the VM. This is useful in the lab or if you are using the Cisco Modeling Tool.

## Creating an Instance Using the OpenStack Command Line Tool

Although the following procedure provides a general guideline on how to create a Cisco Catalyst 8000V tenant instance, the exact steps that you need to perform might vary depending on the characteristics of your KVM environment and setup. For more information, see the OpenStack documentation. See also the *Installation Requirements* sections.

Perform the following steps using the Nova (OpenStack Compute) console on your server.

**Step 1** Download the .qcow2 file from the Cisco Catalyst 8000V software installation image package and copy it onto a local or network device.

**Step 2** Create the Nova flavor using the **nova flavor-create** command:

**nova flavor-create** <*flavor_name*> <*flavor_id*> <*ram size MB*> <*disk size GB*> <*num_ vCPUs*>

The disk size should be set to 0 for the Cisco Catalyst 8000V to boot. The following sample command tells you how to create a KVM instance with 4096 MB RAM, a disk size of 0, and 2 vCPUs:

**nova flavor-create c8000v_flavor 6 4096 0 2**

**Step 3** Enter the **nova flavor-list** command to verify that the nova flavor created the previous step is available.

**Step 4** Use the **glance** command to create the OpenStack image:

**glance image-create --name** <*image_name*> **--disk-format qcow2 --container-format bare --file** <*Location-of-img-file*>

The following example creates an OpenStack image using the Cisco Catalyst 8000V installation file:

**Example:**

```
glance image-create --namec8000v_image --disk-format qcow2 --container-format bare
  --file /opt/stack/c8000v/files/images/c8000v-universalk9.03.12.00.S.154-2.S-std.qcow2
```

**Step 5** Use the **nova boot** command to create the instance and boot:

**nova boot** <*instance_name*> **--image** <*image_id*> **--flavor** <*flavor_id*> **--nic net-id=**<*uuid*> **--config-drive=**<**true**/**false**> **--file**<*configuration_file_name*>

Use the **--config-drive** option to specify that the configuration is loaded on the Cisco Catalyst 8000V when it comes up. 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. There are two possible formats for the configuration file:

- “ovf-env.xml” (OVF format)
- “iosxe_config.txt”

**Note** For details , see *Bootstrap Properties* and subsequent sections.

**Note** These file names are hard-coded and required for the config-drive settings to boot.

You can specify both configuration files in the **nova boot** command line. For example:

**Example:**

```
nova boot c8000v-vm-316 --image c8000v-316 --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
   --file iosxe_config.txt=/home/stack/conf_files/ut/iosxe_config.txt
```

**Example:**

This example shows the booting of the Cisco Catalyst 8000V image on OpenStack with the “ovf-env.xml” file containing the router configuration:

```
nova boot c8000v_instance --image c8000v_image --flavor 6 --nic
net-id=546af738-bc0f-43cf-89f2-1e2c747d1764
 --config-drive=true --file ovf-env.xml=/opt/stack/c8000v/files/ovf-env.xml
```

**Example:**

The following example boots the Cisco Catalyst 8000V image on OpenStack with the "iosxe_config.txt" file containing the router configuration:

```
nova boot c8000v_instance --image c8000v_image --flavor 6 --nic
net-id=546af738-bc0f-43cf-89f2-1e2c747d1764
 --config-drive=true --file iosxe_config.txt=/opt/stack/iosxe_config.txt
```

Cisco Catalyst 8000V begins the boot process.

After the OpenStack image is created, you can access the instance on your OpenStack dashboard.

## Creating the Instance Using the OpenStack Dashboard

Perform the following steps to create the instance using the OpenStack dashboard.

**Note** To configure the KVM to run with config-drive, you must select the serial image and use the procedure described in the *Create the Instance Using the OpenStack Command Line Tool* section.

**Step 1** Download the .qcow2 file from the Cisco Catalyst 8000V software installation image package, and copy it onto a local or network device.

**Step 2** From the OpenStack dashboard, access the OpenStack console.

**Step 3** Log in as the admin into the OpenStack console.

**Step 4** Create a new flavor using the **Flavor Create** tab on the screen, and specify the <*flavor_name*> <*flavor_id*> <*ram size MB*> <*disk size GB*> <*num_ vCPUs*>.

See the *Installation Requirements* section to know the requirements for your version of IOS XE.

The disk size should be set to 0 for the Cisco Catalyst 8000V to boot. as in the tables 6-1 and 6-2.

Select the required flavor from the **System Panel** > **Flavors** tab.

**Step 5** Create a new image using the **Image Create** tab on the screen.

Specify the location of the image, the disk format (qcow2), and the container-format (raw).

Select the **System Panel** > **Images** tab. The image should show up on the list of images shown on the screen.

**Step 6** Create a new instance using the **Instance Create** tab on the screen.

Specify the image, the flavor, and the appropriate network interfaces to be attached to the instance.

Select the **System Panel** > **Instances** tab. The instance should show up on the list of instances shown on the screen, and you should be able to access the console by clicking on the instance name.

**Step 7** To launch the instance, select the instance, and select **Launch Instance**.

Click the **Details** tab. Review the instance information to ensure it is correct. When you are ready to launch the instance, click **Launch**.

The instance is launched and the Cisco Catalyst 8000V begins the boot process.

## Troubleshooting Issues While Creating Instances using OpenStack

The following issues might occur when you create an instance using the OpenStack Dashboard. To know how to create an instance using OpenStack, .

- If you do not see any output on the OpenStack dashboard's console it may be due to you having previously selected an incorrect type of image. Select the virtual qcow2 image (not the serial qcow2 image) before following the steps in .

# Bootstrapping the Cisco Catalyst 8000V Configuration

## Bootstrap Properties

The Cisco Catalyst 8000V bootstrap properties are specified in the **ovf-env.xml** file. For an example **ovf-env.xml** file, see the *Example ovf-env.xml File* section.

*Table 16: 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. |
| 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. |

| Property | Description |
|---|---|
| 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. |

# Example ovf-env.xml File

```
<?xml version="1.0" encoding="UTF-8"?>
<Environment
   xmlns:oe="http://schemas.dmtf.org/ovf/environment/1">
    <PropertySection>
        <Property oe:key="com.cisco.csr1000v.license.1" oe:value="security"/>
        <Property oe:key="com.cisco.csr1000v.console.1" oe:value="serial"/>
```

```
<Property oe:key="com.cisco.csr1000v.config-version.1" oe:value="1.0"/>
        <Property oe:key="com.cisco.csr1000v.domain-name.1" oe:value=""/>
        <Property oe:key="com.cisco.csr1000v.enable-scp-server.1" oe:value="False"/>
        <Property oe:key="com.cisco.csr1000v.enable-ssh-server.1" oe:value="False"/>
        <Property oe:key="com.cisco.csr1000v.hostname.1" oe:value="lab"/>
        <Property oe:key="com.cisco.csr1000v.license.1" oe:value="ax"/>
        <Property oe:key="com.cisco.csr1000v.login-password.1" oe:value=""/>
        <Property oe:key="com.cisco.csr1000v.login-username.1" oe:value="lab"/>
       <Property oe:key="com.cisco.csr1000v.mgmt-interface.1" oe:value="GigabitEthernet1"/>

       <Property oe:key="com.cisco.csr1000v.mgmt-ipv4-addr.1" oe:value="172.25.223.251/25"/>

       <Property oe:key="com.cisco.csr1000v.mgmt-ipv4-gateway.1" oe:value="172.25.223.129"/>

        <Property oe:key="com.cisco.csr1000v.mgmt-ipv4-network.1" oe:value=""/>
        <Property oe:key="com.cisco.csr1000v.mgmt-vlan.1" oe:value=""/>
        <Property oe:key="com.cisco.csr1000v.pnsc-agent-local-port.1" oe:value=""/>
        <Property oe:key="com.cisco.csr1000v.pnsc-ipv4-addr.1" oe:value=""/>
        <Property oe:key="com.cisco.csr1000v.pnsc-shared-secret-key.1" oe:value=""/>
        <Property oe:key="com.cisco.csr1000v.privilege-password.1" oe:value=""/>
        <Property oe:key="com.cisco.csr1000v.remote-mgmt-ipv4-addr.1" oe:value=""/>
        <Property oe:key="com.cisco.csr1000v.resource-template.1"
oe:value="service_plane_medium"/>
        <Property oe:key="com.cisco.csr1000v.ios-config-0001" oe:value="logging buffered
10000"/>
       <Property oe:key="com.cisco.csr1000v.ios-config-0002" oe:value="hostname uut-ovf"/>

        <Property oe:key="com.cisco.csr1000v.ios-config-0003" oe:value="ip domain-name
cisco.com"/>
       <Property oe:key="com.cisco.csr1000v.ios-config-0004" oe:value="crypto key generate
 rsa modulus 1024"/>
        <Property oe:key="com.cisco.csr1000v.ios-config-0005" oe:value="interface
GigabitEthernet2"/>
       <Property oe:key="com.cisco.csr1000v.ios-config-0006" oe:value="ip address 10.0.0.5
 255.255.255.0"/>
        <Property oe:key="com.cisco.csr1000v.ios-config-0007" oe:value="no shut"/>
        <Property oe:key="com.cisco.csr1000v.ios-config-0008" oe:value="exit"/>
        <Property oe:key="com.cisco.csr1000v.ios-config-0009" oe:value="ip route 0.0.0.0
0.0.0.0 10.0.0.1"/>
    </PropertySection>
</Environment>
```

# Sample iosxe_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 1024
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
```

# Increasing the KVM Configuration Performance

You can increase the performance for a Cisco Catalyst 8000V running in a KVM environment by modifying some settings on the KVM host. These settings are independent of the IOS XE configuration settings on the Cisco Catalyst 8000V instance.

To improve the KVM configuration performance, Cisco recommends that you:

- Enable vCPU pinning
- Enable emulator pinning
- Enable numa tuning. Ensure that all the vCPUs are pinned to the physical cores on the same socket.
- Set hugepage memory backing
- Use virtio instead of IDE
- Use graphics VNC instead of SPICE
- Remove unused devices USB, tablet etc.
- Disable memballoon

**Note** These settings might impact the number of VMs that you can be instantiate on a server.

Tuning steps are most impactful for a small number of VMs that you instantiate on a host.

In addition to the above mentioned, do the following:

### Enable CPU Pinning

Increase the performance for the 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 the following commands:

- **virsh nodeinfo**: To verify the host topology to find out how many vCPUs are available for pinning by using the following command.
- **virsh capabilities**: To verify the available vCPU numbers.
- **virsh vcpupin** <*vmname*> <*vcpu#*> <*host core#*>: To pin the virtual CPUs to sets of processor cores.

  This KVM command must be executed for each vCPU on your Cisco Catalyst 8000V instance. The following example pins virtual CPU 1 to host core 3:

  **virsh vcpupin c8000v 1 3**

  The following example shows the KVM commands needed if you have aCisco 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 number from 0 to 7. For more information, see the 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 as mentioned in the following table:

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

For information about Red Hat Enterprise Linux requirements, see the subsequent sections.

### Host OS Settings

In the host side, Cisco recommends that you use hugepages and enable emulator pinning. The following are some of the recommended settings in the host side:

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

In addition to enabling hugepages and emulator pinning, the following settings are also recommended: 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 for better performance. However, note that this might bring in some limitations such as 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 like fd.io VPP or OVS-DPDK, reserve at least 2 cores for the VPP worker threads or the OVS-DPDK PMD threads.

Configure the following parameters to run the VPP through command line:

- -cpu host: This parameter causes the VM to inherit the host OS flags. You require libvirt 0.9.11 or greater for this to be included in the xml configuration.
- -m 8192: You require 8GB RAM for optimal zero packet drop rates.
- rombar=0: To disable PXE boot delays, set rombar=0 to the end of each device option list or add "<rom bar=off />" to the device xml configuration.

### Sample XMLs for KVM Performance Improvement

### Sample XML for numa tuning

```
<numatune>
  <memory mode='strict' nodeset='0'/>
</numatune>
```

### Sample XML for vCPU and emulator pinning

```
 <cputune>
    <vcpupin vcpu='0' cpuset='3'/>
    <emulatorpin cpuset='3'/>
  </cputune>
```

### Sample XML for hugepages

```
 <currentMemory unit='KiB'>4194304</currentMemory>
  <memoryBacking>
    <hugepages>
      <page size='1048576' unit='KiB' nodeset='0'/>
    </hugepages>
    <nosharepages/>
  </memoryBacking>
```

### Sample XML for virtio instead of IDE

```
<devices>
    <emulator>/usr/libexec/qemu-kvm</emulator>
    <disk type='file' device='disk'>
      <driver name='qemu' type='qcow2'/>
      <source file='/var/lib/libvirt/images/rhel7.0.qcow2'/>
      <backingStore/>
      <target dev='vda' bus='virtio'/>
      <boot order='1'/>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x05' function='0x0'/>
    </disk>
```

### Sample XML for VNC graphics

```
<graphics type='vnc' port='5900' autoport='yes' listen='127.0.0.1' keymap='en-us'>
      <listen type='address' address='127.0.0.1'/>
    </graphics>
```

### XML for disabling memballon

```
<memballon model='none'>
```

# Cloning the VM

### 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 is caused by an increase in the size of the cloned image size created by **virt-manager** compared to the original Cisco Catalyst 8000V VM image. The extra bytes (in the KB range) cause the boot failure.

### Workaround

There are three workarounds:

- 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 the new Cisco Catalyst 8000V VM is created. This ensures that the cloned Cisco Catalyst 8000V VM image is able to boot up. However, this workaround does not support nested cloning. Use this method only on the first cloned Cisco Catalyst 8000V VM image.

# Configure the halt_poll_ns Parameter

`halt_poll_ns` is a KVM 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, the QEMU traditionally halts the idle CPU. This setting specifies a period of 400 nanoseconds by default, where 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 been idle when new work arrives, the vcpu must be brought out of the idle state before the new work can be started. The time taken from idle to running state induces additional latency which negatively impacts latency sensitive workloads.

With the default kernel parameters, the guest Cisco Catalyst 8000V router CPU consumes 100% of the host CPU.

You can configure halt_poll_ns in two ways:

- **Large halt_poll_ns**: In this case, more CPU is spent busy-spinning for events that wake the virtual CPU, and less acpi deep sleeps occur. This means more power is consumed. However, there are less wakeups from deep states states, which depending on the state that’s configured, can cause issues like cache misses etc.
- **Small halt_poll_ns**: In this case, less CPU time is spent busy-spinning for events that wake the CPU, more acpi deep sleeps occur. Here, less power consumed, but more wakeups from deep sleep states are required. More wakeups can cause large amounts of deep sleep instances, which depending on the configuration, can cause large amounts of cache misses and long wakeup time.

### Configuring the halt_poll_ns parameter

You can configure the halt_poll_ns parameter in the following ways:

1. At run time, run the following: `echo 0 > /sys/module/kvm/parameters/halt_poll_ns`.
2. When you load the module, perform the following 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.

CHAPTER 6

# Installing in Microsoft Hyper-V Environments

## Microsoft Hyper-V Support Information

Cisco Catalyst 8000V supports installation on Microsoft Hyper-V using Windows Server 2012 R2.

Cisco Catalyst 8000V installation on Microsoft Hyper-V requires the manual creation of a VM and installation using the .iso file. Deploying the OVA template into a Microsoft Hyper-V environment is not supported.

The following Microsoft Hyper-V features are supported:

- Live Migration
- Snapshot
- Move
- Export
- Hyper-V Replica

For more information about Microsoft Hyper-V, see the Microsoft Windows Server 2012 R2 documentation.

## Microsoft Hyper-V Limitations

This section describes the limitations when specifying VLANs on a VM interface, using the Hyper-V Manager.

You can only add one VLAN for a VM interface using the Virtual Switch Manager of the Hyper-V Manager.

**Set-VMNetworkAdapterVlan -VMName** dr-vm-6-1 -**Trunk** -**AllowedVlanIdList** 1-300 -**NativeVlanId** 0

**Note** The **Set-VMNetworkAdapterVlan** command must be re-entered every time you load a Cisco Catalyst 8000V instance. Cisco recommends that you limit the number of VLANs to 300 or below by using the **AllowedVlanIdList** parameter.

For more information on the **Set-VMNetworkAdapterVlan** powershell command, see https://technet.microsoft.com/itpro/powershell/windows/hyper-v/set-vmnetworkadaptervlan.

See also Configure virtual local area networks for Hyper-V.

# Installation Requirements for Microsoft Hyper-V

The Microsoft Hyper-V requirements for Cisco Catalyst 8000V using Cisco IOS XE 17.4 and later releases are as follows:

- The following Microsoft Hyper-V versions are supported:
  - Windows Server 2016 is recommended - tested and meets the performance benchmarks.
- vCPUs. The following vCPU configurations are supported:
  - 1 vCPU: requires minimum 4 GB RAM allocation
  - 2 vCPUs: requires minimum 4 GB RAM allocation
  - 4 vCPUs: requires minimum 4 GB RAM allocation
- Virtual CPU cores—1 vCPU is required
- Virtual hard disk size—8 GB minimum
- Supported vNICs—NetVSC (pmap)
- Maximum number of vNICs supported per VM instance—8
- Virtual CD/DVD drive installed—required

# Manually Creating the Cisco Catalyst 8000V VM using the .iso File (Microsoft Hyper-V)

## Prerequisites

### Prerequisite for Manually Creating theCisco Catalyst 8000V VM using the .iso File

While the following procedure provides a general guideline for how to manually create the VM for Cisco Catalyst 8000V, the exact steps that you need to perform may vary depending on the characteristics of your Microsoft Hyper-V environment and setup. For more information, see Microsoft Windows Server 2012 R2 documentation.

**Note** Cisco Catalyst 8000V does not support deploying the OVA file in Microsoft Hyper-V environments.

Before installing Cisco Catalyst 8000V on a Microsoft Hyper-V VM, install the following on the host:

- Hyper-V Manager
- Failover Cluster Manager
- Virtual Switch

Although not required, it is recommended that you create the Virtual Switch prior to creating the VM for Cisco Catalyst 8000V.

## Configuring the Server Manager Settings

Perform the following steps on the Server Manager on the host before creating the Cisco Catalyst 8000V VM.

**Step 1** On the Server Manager, select **Dashboard** to configure the local server.

**Step 2** From the top-right, select **Manager**, and then select **Add Roles and Features** from the drop-down menu.

The system displays the Add Roles and Features Wizard.

**Step 3** Click **Next**.

**Step 4** Select **Server** > **Roles**. From the Roles list, select the following options by selecting the checkbox:

- File and Storage Services
- Hyper-V

**Step 5** Select **Features**. In the Features list, select the **Failover Clustering** option by selecting the checkbox:

Failover clustering is a mandatory option. This option is not automatically installed, and you must ensure you select this option.

**Step 6** Click **Next**.

# Creating the VM

To create the VM, perform the following steps:

**Step 1** In the Hyper-V Manager, click on the host.

**Step 2** Select **New** > **Virtual Machine**.

**Step 3** Click **Specify Name and Location**.

- Enter the name of the VM.
- (Optional) Click the checkbox to store the VM in a different location.

Click **Next**.

**Step 4** On the Assign Memory screen, enter the **Startup Memory** value.

Cisco Catalyst 8000V requires 4096 MB for the startup memory.

Click **Next**.

**Step 5** On the Configure Networking screen, select a network connection to the virtual switch that was previously created.

The network adapter selected in this step becomes the first interface for Cisco Catalyst 8000V once the VM is launched and the router boots. The other vNICs for the VM are created in the next procedure.

**Note** Changing the MAC address of the first interface and rebooting a licensed Cisco Catalyst 8000V will de-activate the license.

Click **Next**.

**Step 6** On the Connect Virtual Hard Disk screen, select the **Attach a Virtual Hard Disk Later** option.

**Note** The New Virtual Machine Wizard only supports creating a virtual hard disk using the .vhdx format. Cisco Catalyst 8000V requires that the hard disk use the .vhd format. Note that you can create the virtual hard disk after the VM has been created.

Click **Next**.

**Step 7** Review the VM settings, and if you'd like to proceed, click **Finish**.

The new VM is created.

# Configuring the VM Settings

To configure the VM settings before launching the VM, perform the following steps:

**Step 1** In the Hyper-V Manager, select the host, and then right-click the VM that you created in the previous steps.

**Step 2** Select **Settings**.

**Step 3** Specify the number of virtual processors, also known as virtual CPU’s (vCPU’s) for the VM.

See the *Installation Requirements for Microsoft Hyper-V* table, for the supported configurations.

**Step 4** From **IDE Controller 0**, select **Hard Drive**. Select the **Virtual Hard Disk** checkbox, and click **New** to create a new, virtual hard disk.

**Step 5** The system displays the New Virtual Hard Disk Wizard. Click **Next**.

a) On the Choose Disk Format screen, select the **VHD** checkbox to create the virtual hard disk using the .vhd format. Click **Next**.

**Note** Cisco Catalyst 8000V does not support the VHDX format.

b) On the Choose Disk Type screen, click the **Fixed Size** option, and click **Next**.

Cisco Catalyst 8000V does not support the other disk type options.

c) Specify the **Name** and the **Location** for the virtual hard disk, and click **Next**.

d) On the Configure Disk screen, click the option to create a new blank virtual hard disk. For the size, specify `8 GB`.
e) Click **Next** to view the Summary of the virtual hard disk settings.
f) Click **Finish** to create the new virtual hard disk.

After the new hard disk is created, continue configuring the VM settings.

**Step 6** From the **IDE Controller1** field, select **DVD Drive**.

**Step 7** For the **Media** setting, click the **Image File** checkbox, and browse to the Cisco C8000V .iso file that you downloaded from Cisco.com. Click **OK**.

**Step 8** Select **Network Adapter** to verify that the network connection to the virtual switch is configured.

**Step 9** Select **Com 1** to configure the serial port.

This port provides access to the Cisco Catalyst 8000V console.

**Note** Telnet access to the Cisco Catalyst 8000V console is not supported for Microsoft Hyper-V. You must use a Putty session to access the console.

**Step 10** Select **Hardware**> **Add Hardware** to add the network interfaces (vNICs) to the VM.

a) Select **Network Adapter** and click **Add**.

Microsoft Hyper-V adds the network adapter and highlights that hardware with the status Virtual Switch **Not Connected**.

b) Select a virtual switch from the drop-down menu to place the network adapter onto it.

Repeat these steps for each vNIC that you want to add. Cisco Catalyst 8000V supports only the HV NETVSC vNIC type. The maximum number of vNICs supported is 8.

**Note** The hot-add of vNICs is not supported with Microsoft Hyper-V. You must add the network interfaces before launching the VM.

After the Cisco Catalyst 8000V instance boots, you can verify the vNICs and how they are mapped to the interfaces using the **show platform software vnic-if interface-mapping** command. See Mapping the *Cisco Catalyst 8000V Network Interfaces to the VM Network Interfaces* section.

**Step 11** Click **BIOS** to verify the boot sequence for the VM. The VM should be set to boot from the CD.

# Launching the VM to Boot the Cisco Catalyst 8000V

To launch the VM, perform the following steps:

**Step 1** Select the virtual switch.

**Step 2** Select the VM and click **Start**.

The Hyper-V Manager connects to the VM and starts the launch process. Once the VM is launched, Cisco Catalyst 8000V starts the boot process. For more information on the booting process, see *Installing the Cisco Catalyst 8000V in Microsoft Hyper-V Environments*.

CHAPTER 7

# Installing in Citrix XenServer Environments

Cisco Catalyst 8000V installation on Citrix XenServer requires the manual creation of a VM and installation using the .iso file. Deploying the OVA template into a Citrix XenServer environment is not supported in this release.

Cisco Catalyst 8000V supports the VIF vNIC type on the Citrix XenServer implementation.

The following Citrix XenServer features are supported:

- Virtual machine power-cycle
- Interface add and delete

  **Note** This operation requires you to shut down the Cisco Catalyst 8000V instance before performing interface add and delete.

- NIC bonding
- Virtual machine cloning

  Only cold cloning is supported. That is, you must power down the VM when the cloning takes place.

- Taking, restoring and deleting snapshots

  Using the Citrix XenServer, you can take a snapshot of the current state of the VM. Snapshots are supported when the Cisco Catalyst 8000V VM is either powered up or powered down.

- Remote storage
- Performance monitoring (CPU, network and disk)

  **Note** Cisco Catalyst 8000V does not support XenTools. XenMotion operation is not supported on Cisco Catalyst 8000V because it requires XenTools.

For more information, see the Citrix XenServer documentation.

# Installation Requirements for Citrix XenServer

The following are the installation requirements for Citrix XenServer for Cisco Catalyst 8000V running on Cisco IOS XE 17.4 and later.

- Citrix XenServer version supported: Citrix XenServer 6.5 is recommended—tested and meets performance benchmarks. Citrix XenServer 6.2 is supported
- Supported vCPU configurations. (Also depends on the throughput license and technology package installed–see Datasheet). : 1 vCPU: requires minimum 4 GB RAM allocation; 2 vCPUs: requires minimum 4 GB RAM allocation; 4 vCPUs: requires minimum 4 GB RAM allocation
- Virtual CPU cores required: 1
- Supported vNICs : VIF-netfront (pmap)
- Maximum number of vNICs supported per VM instance: 7
- Virtual CD/DVD drive Installed: Required
- Virtual Disk—a 8 GB virtual disk is supported.
- Virtual CPU cores—1 vCPU is required

# Manually Creating the Cisco Catalyst 8000V VM Using the .iso File (Citrix XenServer)

While the following procedure provides a general guideline for how to manually create the VM for Cisco Catalyst 8000V, the exact steps that you need to perform may vary depending on the characteristics of your Citrix XenServer environment and setup. For more information, see the Citrix XenServer documentation.

To determine, for example, the number of vNICs when you install Cisco Catalyst 8000V on a Citrix XenServer VM, see the relevant installation requirements section for your Cisco IOS XE release.

**Note** Cisco Catalyst 8000V does not support deploying the OVA file in KVM environments.

Perform the following steps using the Citrix XenCenter console.

**Step 1** Download the .iso file from the Cisco Catalyst 8000V software installation image package and copy it onto a local or network device.

**Step 2** In the Citrix XenCenter console, to create a new VM, select the server, and click New VM.

The system displays the Select a VM template screen.

**Step 3** Click **Template**. Scroll through the templates and select **Other Install Media**. Click **Next**.

**Step 4** In the **Name** field, enter the name of the VM.

**Step 5** When prompted for the installation media, choose from one of the following:

- Install from the ISO library or DVD drive
- Boot from network

Click **Next**.

**Step 6** Select the server where the VM should be placed.

**Step 7** Select the **Place the VM** checkbox on the server. Click **Next**.

**Step 8** Enter the number of vCPUs and memory settings, and click **Next**.

**Step 9** Add the virtual disks by configuring the following:

- Enter the description (optional).
- Select the virtual disk size from the pull-down menu.

  See the requirements sections in this guide for the supported number of vCPUs and memory requirements for your release.
- Enter the location of the virtual disk.

**Step 10** Click **Add**, and click **Next**.

**Step 11** On the Networking screen, select the networks that will connect to the Cisco Catalyst 8000V through the vNICs.

See the requirements sections in this guide for the supported number of vCPUs and memory requirements for your release.

a) Select a network and click **Add Network**.
b) Select **External**, and click **Next**.
c) Type in the network name. Click **Next**.
d) Select the NIC to use, the VLAN, and set the MTU value.

**Step 12** Click **Finish**.

The new network is added. Repeat the procedure in the previous step for each vNIC.

For more information about booting the VM, see the documentation at: http://www.citrix.com/. When the VM is booted, the Cisco Catalyst 8000V begins the first-time boot process. To continue the boot process, see the *Booting the Cisco Catalyst 8000V and Accessing the Console* section in this guide.

CHAPTER 8

# Configuring Day 0 Settings

### Information About Day0 Configuration

The Cisco Catalyst 8000V instance requires manual configuration before the device is fully functional. To automate the configuration steps or to connect to on-premise sites, you can upload the Cisco Catalyst 8000V custom data or user data in all the supported public and private clouds.

By uploading the custom data for your cloud service provider or your private cloud, you can automate the day 0 and/or the bootstrap configuration. Upload or attach a bootstrap configuration file, (iosxe_config.txt file, ciscosdwan_cloud_init.cfg file or a ciscosdwan.cfg file) or provide the user data to automate these processes to bring up the device into a functional state with minimal to no touch.

Use the universalk9 image to deploy Cisco IOS XE (autonomous mode) and Cisco IOS XE SD-WAN (controller mode) features on Cisco IOS XE devices. After you deploy the Cisco Catalyst 8000V image, the Day0 and/or the bootstrap configuration is used to determine if the router has to boot up in the controller mode or the autonomous mode.

### Autonomous and Controller Mode

You can access the Cisco IOS XE and the Cisco IOS XE SD-WAN functionalities by choosing either the autonomous mode or the controller mode, respectively. The autonomous mode is the default mode for Cisco Catalyst 8000V and includes the Cisco IOS XE functionalities. To access the Cisco IOS XE SD-WAN functionalities, switch to the controller mode.

The following are the main differences between the autonomous mode and the controller mode:

*Table 17:*

| Feature | Autonomous Mode | Controller Mode |
|---|---|---|
| Configuration method | • CLI<br>• NETCONF | • YANG-based configuration<br>• Cisco vManage<br>• NETCONF |

| Feature | Autonomous Mode | Controller Mode |
|---|---|---|
| Onboarding modes | • Config-Wizard<br>• WebUI<br>• USB<br>• Auto-install (Python script, TCL script)<br>• ZTP (using DCHP option 150 and option 67) | Plug and Play<br>USB |
| Interconnectivity | Network interface | VPN |
| Licensing | • Cisco Smart Licensing<br>• PayG | Cisco High Performance Security (HSEC) software licensing. No device licensing. |
| Dual-IOSd redundancy model | Supported | Not supported |
| High availability | Supported | Not supported |
| Global configuration mode | **configure terminal** command | **configure transition** command |

**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.

To switch between the controller and the autonomous modes, see the *Switching Between Autonomous and Controller Modes* section in this feature document.

If you are a user who wants to proceed with the autonomous mode configuration, continue reading this feature document. If you wish to deploy theCisco Catalyst 8000V instance in the controller mode, see Install and Upgrade for Cisco IOS 17.2 and Later.

# Prerequisites for Deploying the Unified Image

- If you want to deploy the Cisco Catalyst 8000V instance in the controller mode, generate the bootstrap config file from vManage.

# Restrictions for Deploying the Unified Image

- 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.
- After onboarding and determining the mode of operation, if you switch from the controller mode to the autonomous mode or vice versa, it results in the loss of configuration.
- When you switch from the autonomous mode to the controller mode or vice versa, Smart Licensing registration does not work. You must reregister for Smart Licenses to work.

# How to Perform Day0 Configuration

## Bootstrap Configuration Files

On a device that already runs a Cisco IOS XE non-SDWAN image, after you install the Cisco IOS XE 17.4 image, when you launch the Cisco Catalyst 8000V instance for the first time, in the absence of bootstrap configuration, the instance always comes up in the autonomous mode. If you provide any user-data or custom-data or bootstrap configuration to the instance, depending on the cloud environment, the data is used for the bootstrap configuration. To know more about the Day0 or bootstrap configuration for each service provider, see the *Day0 and Custom Data Configuration* section in this feature document.

On a new, out-of-box device, if you want to boot up the device in the autonomous mode, you need not provide the bootstrap configuration. In this scenario, by default, the instance always boots up in the autonomous mode. If you want to provide bootstrap related configurations, upload the iosxe_config.txt file or the ovf-env.xml file.

**Note** In the case of public clouds, the filename does not matter as the instance fetches the latest user data or the custom data from the metadata. However, in the case of private clouds, if you upload the .iso file, the filename is important.

On a new, out-of-box device, if you want to boot up the device in the controller mode, ensure that all the four parameters (OTP, UUID, VBOND, ORG) is present in the `ciscosdwan.cfg/ciscosdwan_cloud_init.cfg` file for a fresh deployment on the Cisco Catalyst 8000V devices. After the device boots up in the controller mode, the configuration present in the configuration file is applied.

## Day 0 and Custom Data Configuration for the Cloud Service Providers

Based on the cloud in which you are deploying your Cisco Catalyst 8000V instance, see the following links to perform the bootstrap and/or the day 0 configuration:

- Deploying the OVA to the VM

- Creating a Cisco Catalyst 8000V VM using the .iso file (Citrix XenServer)
- Creating a Cisco Catalyst 8000V VM using the self-installing .run package
- Creating the VM using the .iso file (Microsoft Hyper-V)
- Booting the Cisco Catalyst 8000V Instance
- Deploying a Cisco Catalyst 8000V VM Using Custom Data
- Deploying a Cisco Catalyst 8000V VM on Microsoft Azure

**Note** For a Cisco Catalyst 8000V instance running on Cisco CSP-5000 hypervisor, when you enter the settings in the Day Zero Config screen, ensure that you maintain the format mentioned here:

- **Source File Name**: Enter the value for this field in the format: *day0_filename cisco_sdwan.cfg*.
- **Destination File Name**: Enter the value for this field in the format: *day0-dest-filename /openstack/content/cisco_sdwan.cfg*.

# Verifying the Router Operation Mode and Day 0 Configuration

To verify whether you’ve deployed or upgraded to the IOS XE 17.4 or later releases successfully, run the **show version** command. The **operating device-mode** parameter displays whether the Cisco Catalyst 8000V instance is running in the autonomous or the controller mode.

### Sample configuration output for a Cisco Catalyst 8000V instance in autonomous mode

```
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
```

### Sample configuration output for a Cisco Catalyst 8000V instance in controller mode

```
Device# show version | inc operating
Router operating mode: Controller-Managed
Device# show platform software device-mode
Operating device-mode: Controller
Device-mode bootup status:
-------------------------------------
Success
Device# show platform software chasfs r0 brief | inc device_managed_mode
/tmp/chassis/local/rp/chasfs/etc/device_managed_mode : [controller]
/tmp/fp/chasfs/etc/device_managed_mode : [controller]
Device# show version | inc Last reload
Last reload reason: Enabling controller-mode
```

# Upgrading from existing IOS XE and SD-WAN Images

### Non-SD-WAN Image to Autonomous Mode Upgrade

If you are an existing non-SD WAN user and you are upgrading to the 17.4 release (autonomous mode), you can directly perform the upgrade. That is, you can directly upgrade an existing instance from either UCMk9 or universalk9 to 17.4 universalk9 images.

For example, if you use the image csr1000v-universalk9.16.12.02s.SPA.bin, and upgrade to csr1000v-universalk9.17.xx.xx.SPA.bin, the instance boots up in the autonomous mode and the configuration from UniversalK9 16.xx.xx instance gets ported over to the 17.4 autonomous mode instance.

To know more about the day0/bootstrap configuration for each cloud, see *Day 0 and Custom Data Configuration for the Cloud Service Providers* in this feature document.

### SD-WAN Image to Controller Mode Upgrade

If you are an existing SD-WAN user, and you upgrade to the IOS XE 17.4 release, for example, you upgrade from Csr1000v-ucmk9.16.xx.xx.SPA.bin to CSR1000v-universalk9.16.xx.xx.bin, the instance boots up in the controller mode automatically and the configuration from 16.xx.xx cedge instance gets ported over to the 17.2.1 controller mode instance.

### Non-SD-WAN Image to Controller Mode Upgrade

If you are an existing non-SD-WAN user (universalk9 user) who wants to upgrade to the IOS XE 17.4 release (Controller mode), perform a mode switch. In this case, the existing configuration data is deleted. To proceed with the router configuration the controller mode, see Upgrading from Existing IOS XE and SD-WAN image to Cisco IOS XE 17.2 and Later.

**Note** After you install the Cisco IOS XE 17.4 image, if you want to switch to the autonomous mode, for Cisco Catalyst 8000V instance running on public clouds, provide the appropriate bootstrap configuration. For Cisco Catalyst 8000V instances running on private clouds, the instance comes up with no bootstrap configuration unless you mount an ISO file with the iosxe_config.txt file or the ovf-env.xml file.

# Switching Between Autonomous and Controller Modes

To determine the current mode of your device, run the **show version | inc operating** command. The following table lists the commands and the configuration files needed for switching between the two modes:

| Current Mode | Command to Switch Mode | Mode Changes To | Configuration File and Location | Configuration Example |
|---|---|---|---|---|
| Autonomous mode | controller mode enable | Controller mode | ciscosdwan.cfg on bootflash, CDROM, or CDROM1<br>ciscosdwan_cloud_init.cfg on bootflash, CDROM, or CDROM1 | `cisco#controller-mode enable`<br>`Enabling controller mode will erase the nvram filesystem, remove all configuration files, and reload the box!`<br>`Ensure the BOOT variable points to a valid image`<br>`Continue? [confirm]` |
| Controller mode | controller mode disable | Autonomous mode | ciscortr.cfg in any file system available to the device | `cisco#controller-mode disable`<br>`Disabling controller mode will erase the nvram filesystem, remove all configuration files, and reload the box!`<br>`Ensure the BOOT variable points to a valid image`<br>`Continue? [confirm]` |

**Note** After you execute these commands, the device reloads before the mode change takes effect.

If you want to perform a mode switch and you have a bootstrap configuration file for your configuration, ensure that you copy the file to the bootflash.

In case of public clouds, to provide access to the Cisco Catalyst 8000V instance in either modes, the instance comes up with bare minimum bootstrap configuration. You can use an SSH to log in to the instance during the initial bootup stage or after performing the mode switch operation.

# Frequently Asked Questions

-

**Q.** I have been using Cisco IOS XE image until now. Which mode should I now choose?

**A.** 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. For more information, see *Bootstrap Configuration* section in this chapter.

**Q.** If I am upgrading to the Cisco Catalyst 8000V 17.4 release, do I need to provide the bootstrap configuration?

**A.** If you are an existing non-SD WAN user and are upgrading to the IOS XE 17.4 release (autonomous mode), you can directly perform the upgrade. You need not perform the Day 0 or custom data configuration again.

For a Cisco Catalyst 8000V instance running on Azure, the device uses the custom data that you provided the first time you configured your Cisco Catalyst 8000V instance.

For Cisco Catalyst 8000V instances running on AWS and GCP, the device fetches the custom data from the cloud service provider.

**Q.** What happens to my custom data configuration after switching modes?

**A.** The existing configuration data is deleted. Perform the bootstrap or custom data configuration just as you do for a fresh installation.

**Q.** What happens to my custom data after a factory reset?

**A.** When you perform a factory reset, the configuration and the files present on the disk are erased. The router boots up like a fresh install and looks for configuration files at the appropriate location. This action determines the mode and the associated configuration.

**Q.** Can I deploy my Cisco Catalyst 8000V instance in the controller mode with PayG license?

**A.** 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.

CHAPTER 9

# Cisco Catalyst 8000V Licensing

The Cisco Catalyst 8000V is licensed based on throughput, feature-set, and the licensing term. This product supports Cisco Smart Licensing Usage Policy as well as Cisco DNA Licensing. Based on whether you want to go for purchased licenses that go with the Cisco Catalyst 8000V instance, or a subscription-based license, choose one of the following options:

### Subscription-Based Licensing via Cisco DNA

You can purchase a subscription license for Cisco Catalyst 8000V through the following three licenses that are available via Cisco DNA:

- Cisco Catalyst 8000V - Netowork-Premier
- Cisco Catalyst 8000V - Network-Advantage
- Cisco Catalyst 8000V - Network-Essential

For more information on Cisco Catalyst 8000V DNA licensing,see Cisco DNA Software Routing Subscription Guide.

### Bring-Your-Own-Licensing

You also have an option to purchase and use licenses with Cisco Catalyst 8000V as a Bring-Your-Own-License (BYOL) instance or as a Pay-As-You-Go (PAYG) instance.

To use a Cisco Catalyst 8000V - BYOL license, see Cisco Smart Licensing Usage Policy to know to how install and configure your license.

If you have upgraded to Cisco Catalyst 8000V from a Cisco CSR 1000V or a Cisco ISRV, your device will continue to support Cisco Smart Licensing. To use Cisco Smart Licensing, you must first configure the Call Home feature and obtain the Cisco Smart Call Home Services. For more information, see Cisco Smart Licensing.

### Pay-As-You-Go Licensing

Cisco Catalyst 8000V supports the PAYG Licensing model with Amazon Web Services (AWS) and Microsoft Azure Marketplace. Cisco Catalyst 8000V hourly-billed AMI or Pay As You Go licensing model allows you to consume an instance for a defined period of time. In this licensing model, you can directly launch the instance from the AWS or Azure Marketplace and start using the instances. The licenses are embedded in the image.

- Evaluation Licenses for Cisco Catalyst 8000V, on page 80

# Evaluation Licenses for Cisco Catalyst 8000V

Evaluation licenses are available to try out Cisco Catalyst 8000V features. Evaluation licenses are obtained differently depending on the Cisco IOS XE release version.

**Default**

The Cisco Catalyst 8000V router, by default, boots with the following features:

- AX technology package features
- 100 Kbps maximum throughput

### Evaluation License Options

Evaluation licenses are valid for 60 days and are available at the Cisco licensing portal.

Evaluation licenses enable you to test drive additional technology packages and higher throughputs. (The throughputs available through evaluation licenses are the highest supported throughput levels for the package type.)

- IPBase Technology package, 10 Gbps
- SEC Technology package, 5 Gbps
- APP Technology package, 5 Gbps
- AX Technology package, 2.5 Gbps
- 1000 broadband sessions
- 12 GB memory upgrade

### Testing a Lower Maximum Throughput

To test a lower throughput license type that is not listed here, use the **platform hardware throughput level MB <throughput>** command. This command sets the throughput to a supported level below what is provided by the installed license. This has the same effect as installing a license for that throughput level. For example, on a Cisco Catalyst 8000V with a 5 Gbps license installed, the following command sets the throughput level to 250 Mbps:

```
platform hardware throughput level MB 250
```

The supported throughput levels are: 10 Mbps, 50 Mbps, 100 Mbps, 250 Mbps, 500 Mbps, 1 Gbps, 2.5 Gbps, 5 Gbps, 10 Gbps.

For any additional questions, contact your Cisco sales representative.

### Obtaining an Evaluation License from the Cisco Licensing Portal

To obtain a 60-day evaluation license for Cisco Catalyst 8000V, follow the instructions below.

When the 60-day evaluation license expires, the maximum throughput becomes limited to 100 Kbps upon reload.

**Note** These instructions are subject to change.

**Step 1** Navigate to https://www.cisco.com/go/license and log in to the site.

**Step 2** Navigate to the Product License Registration Portal.

**Step 3** On the Product License Registration page, select **Continue to Product License Registration**.

**Step 4** Click **Get Other Licenses**, and select **Demo and Evaluation** from the dropdown menu.

**Step 5** In the Product Family section, select **Routers & Switches**.

**Step 6** In the Product section, select **Cisco Catalyst 8000V**, and Click **Next**.

**Step 7** Select the desired license type. Enter the UDI Serial number, then click **Next** to generate the license. You can view the UDI Serial number of your router by entering the **show license udi** command.

CHAPTER 10

# Verifying the Cisco Catalyst 8000V Hardware and VM Requirements

To help troubleshoot issues with Cisco Catalyst 8000V, ensure that the router is installed on the supported hardware and that the VM requirements are being met:

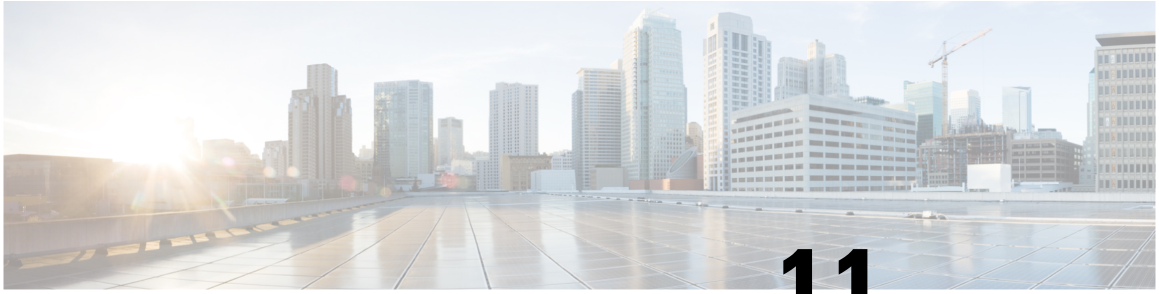
- Verify that the server hardware is supported by the hypervisor vendor.

  If you're using VMware, verify that the server is listed on the VMware Hardware Compatibility List. See the VMware documentation for more information.

- Verify that the I/O devices (for example, FC, iSCSI, SAS) being used are supported by the VM vendor.
- Verify that 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.

- Verify the hypervisor version is supported by Cisco Catalyst 8000V.
- Verify that the correct VM settings for the amount of memory, number of CPUs, and disk size are configured.
- Verify that the vNICs are configured using a supported network driver.



CHAPTER 11

# Configuring Console Access

## Booting the Cisco Catalyst 8000V as the VM

Cisco Catalyst 8000V boots when the VM is powered on. Depending on your configuration, you can monitor the installation process on the virtual VGA console or the console on the virtual serial port.

**Note** If you want 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.

**Step 1** Power-up the VM. Within 5 seconds of powering on the VM, choose a console described from one of the following three steps (2, 3,or 4) to select a console to view the router bootup and to access the Cisco Catalyst 8000V CLI.

**Step 2** (Optional) Select **Auto Console**:

Choose this option to use automatic console detection. When two virtual serial ports are detected, the IOS XE CLI ia available on the first virtual serial port and the IOS XE diagnostic CLI is available on the second virtual serial port. If the two virtual serial ports are not detected, the IOS XE CLI is available on the virtual VGA console. This is the default setting and the Cisco Catalyst 8000V instance boots using the automatic console detection if another option is not selected within the 5 second timeframe.

**Note** (for **VMware ESXi**): If you are installing on VMware ESXi without a virtual serial port concentrator (vSPC), this option may not be able to properly detect virtual serial ports when there is an active connection to the virtual serial ports. If you are not using a vSPC and wish to use virtual serial ports, choose the Serial Console option.

**Note** (for **Microsoft Hyper-V**): If you are installing on Microsoft Hyper-V, this option may be unable to properly detect virtual serial ports when there is an active connection to the virtual serial ports. If you wish to use virtual serial ports, you should choose the Serial Console option.

(Optional)**Automatic selection of virtual serial** portsFor this option, the virtual serial ports must already be present on the VM. The virtual serial port must already be present on the VM for this option to work.

If you are installing on VMware ESXi, see *Creating Serial Console Access in VMware ESXi*.

If you are installing in KVM environments, see *Creating the Serial Console Access in KVM*.

If you are installing in Microsoft Hyper-V environments, see *Creating the Serial Console Access in Microsoft Hyper-V*.

The Cisco Catalyst 8000V instance starts the boot process.

**Step 3** (Optional) Select **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 another option is not selected within the 5 second timeframe. The Cisco Catalyst 8000V instance starts the boot process.

**Step 4** (Optional) Select **Serial Console**

Choose this option to use the virtual serial port console on the VM (not supported on Citrix XenServer VMs).

The virtual serial port must already be present on the VM for this option to work.

**Note** The option to select the console port during the boot process is available only the first time Cisco Catalyst 8000V boots. To change the console port access after Cisco Catalyst 8000V has first booted, see *Changing the Console Port Access After Installation*.

The Cisco Catalyst 8000V starts the boot process.

**Step 5** Telnet to the VM using one of the following two commands: **telnet://***host-ipaddress***:***portnumber* or, from a UNIX xTerm terminal: **telnet** *host-ipaddress portnumber*. The following example shows the Cisco Catalyst 8000V initial boot output on the VM.

**Example:**

```
%IOSXEBOOT-4-BOOT_SRC: (rp/0): CD-ROM Boot%IOSXEBOOT-4-BOOT_CDROM: (rp/0):

Installing GRUB%IOSXEBOOT-4-BOOT_CDROM: (rp/0): Copying super packagecsr1000v-universalk9

2011-10-20_13.09.SSA.bin%IOSXEBOOT-4-BOOT_CDROM: (rp/0):

Creating /boot/grub/menu.lst%IOSXEBOOT-4-BOOT_CDROM: (rp/0):

CD-ROM Installation finished%IOSXEBOOT-4-BOOT_CDROM: (rp/0): Ejecting CD-ROM tray
```

The system first calculates the SHA-1, which may take a few minutes. Once the SHA-1 is calculated, the kernel is brought up. Once the initial installation process is complete, the .iso package file is removed from the virtual CD-ROM, and the VM is rebooted. This enables Cisco Catalyst 8000V to boot normally off the virtual Hard Drive.

**Note** The system reboots during first-time installation only.

The time required for the Cisco Catalyst 8000V to boot may vary depending on the release and the hypervisor you use.

**Step 6** After booting, the system presents a screen showing the main software image and the Golden Image, with an instruction that the highlighted entry is booted automatically in three seconds. Do not select the option for the Golden Image and allow 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 *Accessing and Using the GRUB Mode*.

You can now enter the router configuration environment by entering the standard commands **enable** and then **configure terminal.**. Note the following points for the initial installation:

When you boot a Cisco Catalyst 8000V instance for the first time, the mode the router boots in depends on the release version.

Cisco Catalyst 8000V boots with the AX package set of features and throughput is limited to 100 Kbps.

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 the Cisco Catalyst 8000V.

The installed license technology package must match the package level configured with the **license boot level** 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. You can either use the Cisco IOS XE CLI commands or you can manually configure the properties in the vSphere GUI. For more information, see Editing the *Basic Properties of Cisco Catalyst 8000V Using vSphere*.

# Accessing the Cisco Catalyst 8000V Console

## Accessing the Cisco Catalyst 8000V Through the Virtual VGA Console

When installing the Cisco Catalyst 8000V software image, the setting to use is the Virtual VGA console. You do not require any other configuration changes to access the Cisco Catalyst 8000V CLI through the virtual VGA console if:

- 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're using automatic console detection.

## Accessing the Cisco Catalyst 8000V Through the Virtual Serial Port

### Introduction to Accessing the Cisco Catalyst 8000V through the Virtual Serial Port

By default, you can access a Cisco Catalyst 8000V instance using the virtual VGA console. If you use the automatic console detection and two virtual serial ports are detected, the Cisco Catalyst 8000V CLI will be 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. See the following sections to configure the virtual serial port on your hypervisor.

**Note** Citrix XenServer does not support access through a serial console.

## Creating Serial Console Access in VMware ESXi

Perform the following steps using VMware VSphere. For more information, refer to the VMware VSphere documentation.

**Step 1** Power-down the VM.

**Step 2** Select the VM and configure the virtual serial port settings.

a) Choose **Edit Settings** > **Add**.
b) Choose **Device Type** > **Serial port**. Click **Next**.
c) Choose **Select Port Type**.

Select **Connect via Network**, and click **Next**.

**Step 3** Select **Select Network Backing** > **Server (VM listens for connection)**.

Enter the **Port URI** using the following syntax:

**telnet**://:*portnumber*

where *portnumber* is the port number for the virtual serial port.

Under the I/O mode, select the **Yield CPU on poll** option, and click **Next**.

**Step 4** Power on the VM.

**Step 5** When the VM is powered on, access the virtual serial port console.

**Step 6** Configure the security settings for the virtual serial port.

a) Select the ESXi host for the virtual serial port.
b) Click the **Configuration** tab and click **Security Profile**.
c) In the Firewall section, 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 **Auto Console** option or the **Serial Console** 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 **platform console auto** command or the Cisco IOS XE **platform console serial command** and reload the VM for the console access through the virtual serial port to work.

## Creating the Serial Console Access in KVM

Perform the following steps using the KVM console on your server. For more information, refer to the KVM documentation.

**Step 1** Power off the VM.

**Step 2** Click on the default **Serial 1** device (if it exists) and then click **Remove**. This removes the default pty-based virtual serial port which would otherwise count as the first virtual serial port.

**Step 3** Click **Add Hardware**.

**Step 4** Select **Serial** to add a serial device.

**Step 5** Under **Character Device**, choose the **TCP Net Console (tcp)** device type from the drop-down menu.

**Step 6** Under **Device Parameters**, choose the mode from the drop-down menu.

**Step 7** Under **Host**, enter 0.0.0.0. The server will accept a telnet connection on any interface.

**Step 8** Choose the port from the drop-down menu.

**Step 9** Choose the **Use Telnet** option.

**Step 10** Click **Finish**.

You can now access the Cisco IOS XE console using the Telnet port URI. For more information, see Opening a Telnet Session to the Cisco Catalyst 8000V on the Virtual Serial Port.

**Note** To use these settings, either the **Auto Console** option or the **Serial Console** option in the GRUB menu should be selected while the Cisco Catalyst 8000V booted. If you have already installed the Cisco Catalyst 8000V software using the virtual VGA console, you must configure either the Cisco IOS XE **platform console auto** command or the **platform console serial** command and reload the VM in order for the console access through the virtual serial port to work.

## Creating the Serial Console Access in Microsoft Hyper-V

The console port access for Microsoft Hyper-V is created when configuring the VM settings. For more information, see the *Configuring the VM Settings* section.

**Note** Telnet access to the Cisco Catalyst 8000V console is not supported for Microsoft Hyper-V. You must use a Putty session to access the console.

## Opening a Telnet Session to the Cisco Catalyst 8000V Console on the Virtual Serial Port

Perform the following steps using the Cisco IOS XE CLI commands:

**Step 1** Telnet to the VM.

- Use the following command **telnet://***host-ipaddress***:***portnumber*
- Or, from a UNIX terminal use the command

  **telnet** *host-ipaddress portnumber*

**Step 2** At the Cisco Catalyst 8000V IOS XE password prompt, enter your credentials. The following example shows an entry of the password *mypass*:

**Example:**

```
User Access Verification
Password: mypass
```

**Note** If no password has been configured, press **Return**.

**Step 3** From the user EXEC mode, enter the **enable** command as shown in the following example:

**Example:**

```
Router> enable
```

**Step 4** At the password prompt, enter your system password. The following example shows entry of the password *enablepass*:

**Example:**

```
Password: enablepass
```

**Step 5** When the enable password is accepted, the system displays the privileged EXEC mode prompt:

**Example:**

```
Router#
```

**Step 6** You now have access to the CLI in the privileged EXEC mode and you can enter the necessary commands to complete your desired tasks.

**Step 7** To exit the Telnet session, use the **exit** or **logout** command as shown in the following example:

**Example:**

```
Router# logout
```

# Changing the Console Port Access After Installation

After the Cisco Catalyst 8000V instance has booted successfully, you can change the console port access to the router using Cisco IOS XE commands. After you change the console port access, you must reload or power-cycle the router.

**Step 1** **enable**

**Example:**

```
Router> enable
```

Enables the privileged EXEC mode. Enter your password, if prompted.

**Step 2** **configure terminal**

**Example:**

```
Router# configure terminal
```

Enters the global configuration mode.

**Step 3** Do one of the following:

- **platform console auto**
- **platform console virtual**
- **platform console serial**

**Example:**

```
Router(config)# platform console auto
```

**Example:**

```
Router(config)# platform console virtual
```

**Example:**

```
Router(config)# platform console serial
```

Options for **platform console** *x*:

- **auto** - Specifies that the Cisco Catalyst 8000V console is detected automatically. This is the default setting during the initial installation boot process. For additional information, see Booting the Cisco Catalyst 8000V as the VM.
- **virtual** - Specifies that the Cisco Catalyst 8000V is accessed through the hypervisor virtual VGA console.
- **serial** - Specifies that the Cisco Catalyst 8000V is accessed through the serial port on the VM.

**Note**: Use this option only if your hypervisor supports serial port console access.

**Step 4** **end**

**Example:**

```
Router(config)# end
```

Exits the configuration mode.

**Step 5** **copy system:running-config nvram:startup-config**

**Example:**

```
Router# copy system:running-config nvram:startup-config
```

Copies the running configuration to the NVRAM startup configuration.

**Step 6** **reload**

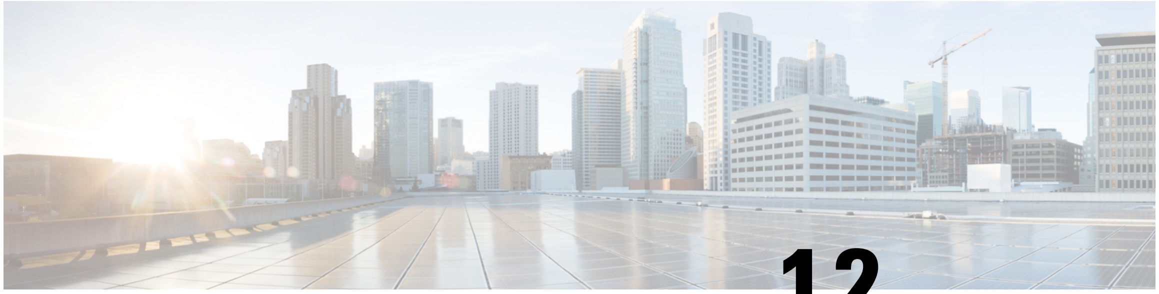
**Example:**

```
Router# reload
```

Reloads the operating system.

# License Installation

One of the first steps you need to perform after obtaining access to the console is to install the Cisco Catalyst 8000V software licenses. For more information, see the *Installing Cisco Catalyst 8000V Licenses* section in this guide..



CHAPTER 12

# Configuring the vCPU Distribution across Data, Service, and Control Planes

You can allocate and distribute the vCPUs of the following planes: Control Plane (CP), Data Plane (DP), and Service Plane (SP) by using templates. Note that the Service Plane includes containers running SNORT.

Use one of the following templates for vCPU distribution:

## vCPU Distribution: Control Plane Extra heavy

The following table shows the vCPU distribution for the Control Plane Extra heavy template.

***Table 18: Control Plane Extra heavy - vCPU Distribution***

| Number of vCPUs | 1 | 2 | 4 | 8 |
|---|---|---|---|---|
| Control Plane | 1/3 | 1/2 | 1 1/2 | 1 1/2 |
| Service Plane | 1/3 | 1/2 | 1 1/2 | 1 1/2 |
| Data Plane | 1/3 | 1 | 1 | 5 |

**Note** Using a Control Plane Extra heavy template, a service plane app can obtain 1.5 full cores for its operation. For example, in the case of WAAS.

# vCPU Distribution: Control Plane heavy

The following table shows the vCPU distribution for the Control Plane heavy template.

*Table 19: Control Plane heavy - vCPU Distribution*

| Number of vCPUs | 1 | 2 | 4 | 8 |
|---|---|---|---|---|
| Control Plane | 1/3 | 1/2 | 1 | 1 |
| Service Plane | 1/3 | 1/2 | 1 | 1 |
| Data Plane | 1/3 | 1 | 2 | 6 |

**Note** The Control Plane heavy template allocates an extra core to the Control Plane/Service Plane services compared to the Data Plane heavy template (there is one core for the Control Plane and another core for the Service Plane). If there is no Service Plane application, the Control Plane utilizes allthe resources (both the cores).

# vCPU Distribution: Data Plane heavy

**Note** The Data Plane heavy template is the default vCPU Distribution template. Even if the configuration output for the Template option reads 'None', the Data Plane heavy template is applied by default.

The following table shows the vCPU distribution for the Data Plane heavy template.

*Table 20: Data Plane heavy - vCPU Distribution*

| Number of vCPUs | 1 | 2 | 4 | 8 |
|---|---|---|---|---|
| Control Plane | 1/3 | 1/2 | 1/2 | 1/2 |
| Service Plane | 1/3 | 1/2 | 1/2 | 1/2 |
| Data Plane | 1/3 | 1 | 3 | 7 |

**Note** 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.

# vCPU Distribution: Data Plane normal

You can use the vCPU distribution for the Data Plane normal template to force the Cisco Catalyst 8000V to behave in the same way as before using a template for vCPU distribution.

That is, assume you create a Cisco Catalyst 8000V VM using the Data Plane heavy template for vCPU distribution, as specified in the ovf-env.xml file. You can later use the CLI commands in the Data Plane normal template to override the XML file settings that were previously applied by the Data Plane heavy template.

# vCPU Distribution: Service Plane heavy

The following table shows the vCPU distribution for the Service Plane heavy template.

*Table 21: Service Plane heavy - vCPU Distribution*

| Number of vCPUs | 1 | 2 | 4 | 8 |
|---|---|---|---|---|
| Control Plane | 1/3 | 1/2 | 1 | 2 |
| Service Plane | 1/3 | 1/2 | 1 | 2 |
| Data Plane | 1/3 | 1 | 2 | 4 |

**Note** Using a Service Plane heavy template, a Service Plane application (such as Snort IPS) can use up to 2 full cores for its operation.

# vCPU Distribution: Service Plane medium

The following table shows the vCPU distribution for the Service Plane medium template.

*Table 22: Service Plane medium - vCPU Distribution*

| Number of vCPUs | 1 | 2 | 4 | 8 |
|---|---|---|---|---|
| Control Plane | 1/3 | 1/2 | 1 | 1 |
| Service Plane | 1/3 | 1/2 | 1 | 1 |
| Data Plane | 1/3 | 1 | 2 | 6 |

# How to Boot the Cisco Catalyst 8000V with an OVA image

To boot the Cisco Catalyst 8000V with an OVA image, boot from a CDROM containing the ovf-env.xml file. This XML file contains the vCPU distribution templates. A template is simply an additional bootstrap property.

For more information about bootstrap properties, see the Bootstrapping the Cisco Catalyst 8000V *Configuration* section.

The following is an example of the part of the XML file that specifies a Service Plane medium template:

```
<Property oe:key="com.cisco.csr1000v.resource-template.1" oe:value="service_plane_medium"/>
```

# How to Configure vCPU Distribution across the Data, Control and Service Planes

Enter the `platform resource` command on theCisco Catalyst 8000V to select a template for vCPU distribution.

**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
```

**Note** After entering the `platform resource` command, you must reboot the Cisco Catalyst 8000V instance to activate the template.

# Determine the Active vCPU Distribution Template

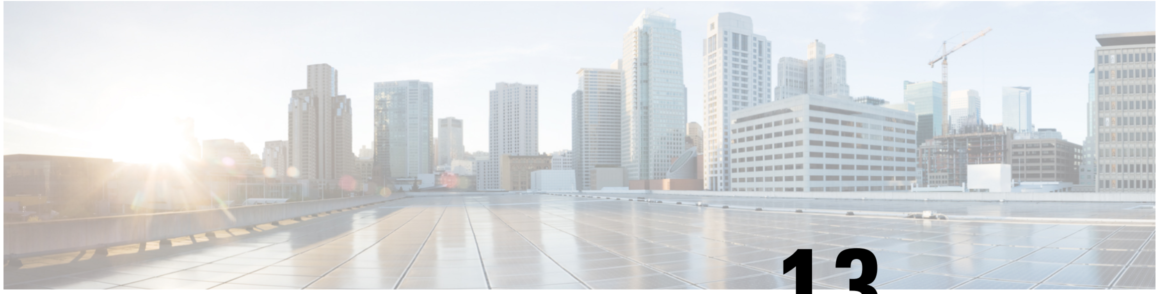
To determine which template is being used for vCPU distribution, use the following command:

**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
```

**Note** The Control plane and the Service plane share cores 0 and 1.



CHAPTER 13

# Web User Interface Management

You can access your router using a web user interface which allows you to monitor the performance of the router using an easy-to-read graphical interface.

**Note** To manage and configure crypto map tunnels, use the CLI. You can also configure the tunnels with Virtual Tunnel Interface (VTI) and then create the tunnels either by using the CLI or the GUI.

You can configure a router by performing the steps in one of the following tasks:

- Setting Up Factory Default Device Using WebUI , on page 97
- Using Basic or Advanced Mode Setup Wizard, on page 98

## Setting Up Factory Default Device Using WebUI

Quick Setup Wizard allows you to perform the basic router configuration. To configure the router:

**Note** Before you access the WebUI, you need to have the basic configuration on the device.

**Step 1** Connect the RJ-45 end of a serial cable to the RJ-45 console port on the router.

**Step 2** After the device initial configuration wizard appears, enter **No** to get into the device prompt when the following system message appears on the router.

Would you like to enter the initial configuration dialog? [yes/no]: no

**Step 3** From the configuration mode, enter the following configuration parameters.

```
!
ip dhcp pool WEBUIPool
network 192.168.1.0 255.255.255.0
default-router 192.168.1.1
username webui privilege 15 password cisco
!
interface gig 0/0/1
ip address 192.168.1.1 255.255.255.0
!
```

**Step 4** Connect your device to the router using an Ethernet cable to the gig 0/0/1 interface.

**Step 5** Set up your system as a DHCP client to obtain the IP address of the router automatically.

**Step 6** Launch the browser and enter the device IP address in your browser's address line. For a secure connection, type https://192.168.1.1/#/dayZeroRouting. For a less secure connection, enter http://192.168.1.1/#/dayZeroRouting.

**Step 7** Enter the default username (webui) and default password (cisco).

# Using Basic or Advanced Mode Setup Wizard

To configure the router using the basic or advanced mode setup:

**Step 1** Choose the **Basic Mode** or **Advanced Mode** and click **Go To Account Creation Page**.

**Step 2** Enter the username and password. Reenter the password to confirm.

**Step 3** Click **Create and Launch Wizard**.

**Step 4** Enter the device name and domain name.

**Step 5** Select the appropriate time zone from the **Time Zone** drop-down list.

**Step 6** Select the appropriate date and time mode from the **Date and Time** drop-down list.

**Step 7** Click **LAN Settings**.

# Configure LAN Settings

**Step 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.

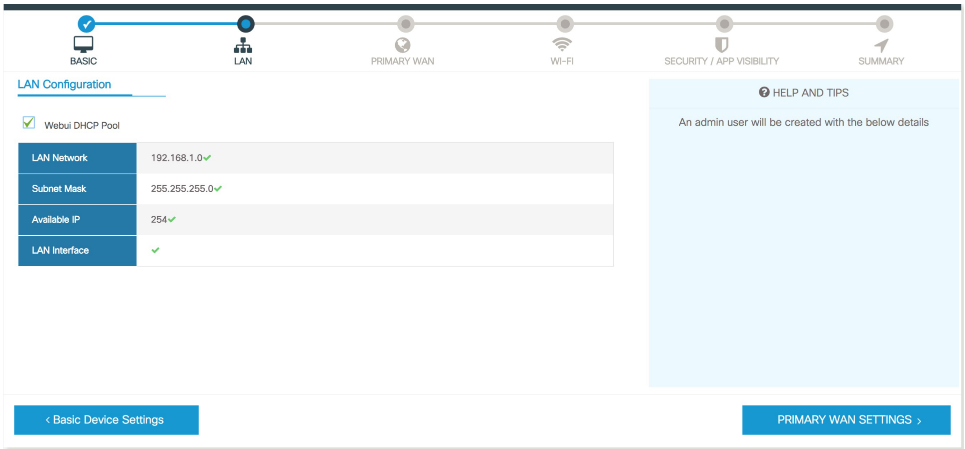
b) If you choose the Create and Associate Access VLAN option, specify the following:

**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.

**Step 2** Click **Primary WAN Settings**.



☑ Webui DHCP Pool

| | |
|---|---|
| LAN Network | 192.168.1.0✓ |
| Subnet Mask | 255.255.255.0✓ |
| Available IP | 254✓ |
| LAN Interface | ✓ |

An admin user will be created with the below details

# Configure Primary WAN Settings

**Step 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.

**Step 2** Select the interface from the drop-down list.

**Step 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.

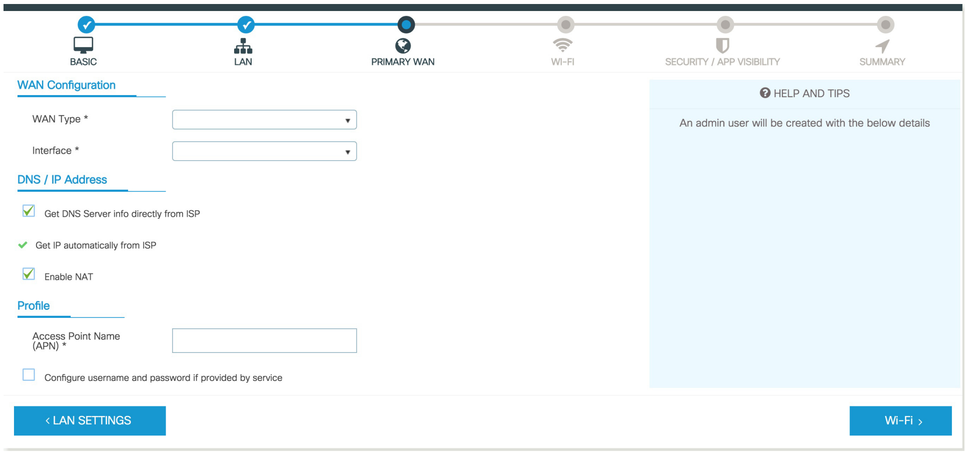
**Step 4** Check the **Get IP automatically from ISP** check box to get the IP address information directly from the service provider. You can also manually enter the IP address and subnet mask.

**Step 5** Check the **Enable NAT** check box to enable NAT. It is recommended to enable NAT.

**Step 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**.

**Step 7** Enter the user name and password provided by the service provider.

**Step 8** Click **Security / APP Visibility WAN Settings**.



WAN Configuration

WAN Type *

Interface *

DNS / IP Address

☑ Get DNS Server info directly from ISP

✓ Get IP automatically from ISP

☑ Enable NAT

Profile

Access Point Name (APN) *

□ Configure username and password if provided by service

HELP AND TIPS

An admin user will be created with the below details

## Configure Secondary WAN Settings

For advanced configuration, you should configure the secondary WAN connection.

**Step 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.

**Step 2** Select the interface from the drop-down list.

**Step 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.

**Step 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.

**Step 5** Check the **Enable NAT** check box to enable NAT. It is recommended to enable NAT.

**Step 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** .

**Step 7** Enter the user name and password provided by the service provider.

**Step 8** Click **Security / APP Visibility WAN Settings**.

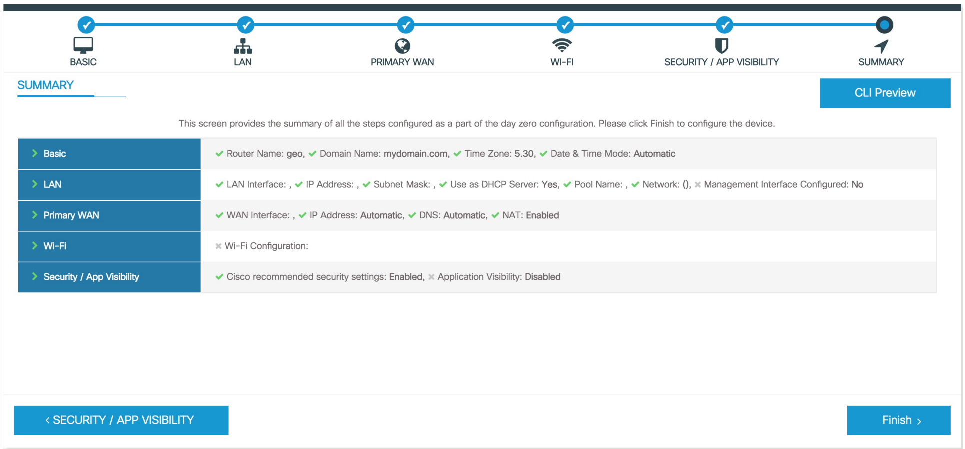
## Configure Security Settings

**Step 1** Check the **Enable Cisco Recommended Security Settings** check box to ensure that all passwords are not shown in plain text. The passwords are encrypted.

**Step 2** Click **Day 0 Config Summary**.

**Step 3** To preview the configuration, click **CLI Preview** to preview the configuration.

**Step 4** Click **Finish** to complete the Day Zero setup.



This screen provides the summary of all the steps configured as a part of the day zero configuration. Please click Finish to configure the device.

| | |
|---|---|
| › Basic | ✓ Router Name: geo, ✓ Domain Name: mydomain.com, ✓ Time Zone: 5.30, ✓ Date & Time Mode: Automatic |
| › LAN | ✓ LAN Interface: , ✓ IP Address: , ✓ Subnet Mask: , ✓ Use as DHCP Server: Yes, ✓ Pool Name: , ✓ Network: (), ✗ Management Interface Configured: No |
| › Primary WAN | ✓ WAN Interface: , ✓ IP Address: Automatic, ✓ DNS: Automatic, ✓ NAT: Enabled |
| › Wi-Fi | ✗ Wi-Fi Configuration: |
| › Security / App Visibility | ✓ Cisco recommended security settings: Enabled, ✗ Application Visibility: Disabled |

CHAPTER 14

# Finding Command Options

This section provides an example of how to display syntax for a command. The syntax can consist of optional or required keywords and arguments. To display keywords and arguments for a command, enter a question mark (**?**) at the configuration prompt or after entering a part of a command followed by a space.

The Cisco IOS XE software displays a list and brief description of the available keywords and arguments. For example, if you were in the global configuration mode and you want to see all the keywords or arguments for the **arap** command, you would type **arap ?**.

The <cr> symbol in the command help output stands for "carriage return." On older keyboards, the carriage return key is the Return key. On most modern keyboards, the carriage return key is the Enter key. The <cr> symbol at the end of the command help output indicates that you have the option to press **Enter** to complete the command and that the arguments and keywords in the list preceding the <cr> symbol are optional. The <cr> symbol by itself indicates that no more arguments or keywords are available and that you must press **Enter** to complete the command.

The following examples show how you can use the  question mark (**?**) to assist you in entering commands.

***Table 23: Finding Command Options***

| Command | Comment |
|---|---|
| `Router>` **`enable`**<br>`Password: <password>`<br>`Router#` | Enter the **enable** command and password to access the privileged EXEC commands. You are in the privileged EXEC mode when the prompt changes to a "# " from the "> ". For example, Router> to Router# . |
| `Router#`<br>**`configure terminal`**<br>`Enter configuration commands, one per line. End with CNTL/Z.`<br>`Router(config)#` | Enter the **configure terminal** privileged EXEC command to enter the global configuration mode. You are in the global configuration mode when the prompt changes to Router(config)# . |

<table>
<tr><th>Command</th><th>Comment</th></tr>
<tr><td>

```
Router(config)# interface
GigabitEthernet ?
  <0-6>     GigabitEthernet interface
number
Router(config)# interface
GigabitEthernet 1
Router(config-if)#
```

</td><td>

Enter the interface configuration mode by specifying the serial Gigabit Ethernet interface that you want to configure using the **interface GigabitEthernet** *number* global configuration command.

Enter **?** to display what you must enter next on the command line.

When the system displays the <cr> symbol, you can press Enter to complete the command.

You are in the interface configuration mode when the prompt changes to Router(config-if)# .

**Note** Cisco Catalyst 8000V supports only Gigabit Ethernet interfaces.

</td></tr>
</table>

<table>
<tr><th>Command</th><th>Comment</th></tr>
<tr><td>

```
Router(config-if)# ?
Interface configuration commands:
  .
  .
  .
  ip                   Interface Internet Protocol config commands
  keepalive            Enable keepalive
  lan-name             LAN Name command
  llc2                 LLC2 Interface Subcommands
  load-interval        Specify interval for load calculation for an
                       interface
  locaddr-priority     Assign a priority group
  logging              Configure logging for interface
  loopback             Configure internal loopback on an interface
  mac-address          Manually set interface MAC address
  mls                  mls router sub/interface commands
  mpoa                 MPOA interface configuration commands
  mtu                 Set the interface Maximum Transmission Unit (MTU)

  netbios              Use a defined NETBIOS access list or enable
                       name-caching
  no                   Negate a command or set its defaults
  nrzi-encoding        Enable use of NRZI encoding
  ntp                  Configure NTP
  .
  .
  .
Router(config-if)#
```

</td><td>

Enter **?** to display a list of all the interface configuration commands available for the Gigabit Ethernet interface. This example shows only some of the available interface configuration commands.

</td></tr>
</table>

| Command | Comment |
|---|---|
| Router(config-if)# ip **?**<br>Interface IP configuration subcommands:<br>  access-group          Specify access control for<br>packets<br>  accounting            Enable IP accounting on this<br> interface<br>  address               Set the IP address of an<br>interface<br>  authentication        authentication subcommands<br>  bandwidth-percent     Set EIGRP bandwidth limit<br>  bgp                   BGP interface<br>commands..<snipped for brevity><br>  .<br>  .<br>  .<br>Router(config-if)# **ip** | Enter the command that you want to configure for the interface. This example uses the **ip** command.<br>Enter **?** to display what you must enter next on the command line. This example shows only some of the available interface IP configuration commands. |
| Router(config-if)# ip address ?<br>  A.B.C.D            IP address<br>  dhcp               IP Address negotiated via DHCP<br> pool<br>   IP Address autoconfigured from a local DHCP pool<br>Router(config-if)# **ip address** | Enter the command that you want to configure for the interface. This example uses the **ip address** command.<br>Enter **?** to display what you must enter next on the command line. In this example, you must enter an IP address or the **negotiated** keyword.<br>A carriage return (<cr>) is not displayed; therefore, you must enter additional keywords or arguments to complete the command. |
| Router(config-if)# **ip address 172.16.0.1 ?**<br>  A.B.C.D             IP subnet mask<br>Router(config-if)# **ip address 172.16.0.1** | Enter the keyword or the argument that you want to use. This example uses the 172.16.0.1 IP address.<br>Enter **?** to display what you must enter next on the command line. In this example, you must enter an IP subnet mask.<br>A <cr> is not displayed; therefore, you must enter additional keywords or arguments to complete the command. |
| Router(config-if)# **ip address 172.16.0.1**<br>**255.255.255.0 ?**<br>  secondary           Make this IP address a<br>secondary address<br>  <cr><br>Router(config-if)# **ip address 172.16.0.1**<br>**255.255.255.0** | Enter the IP subnet mask. This example uses the 255.255.255.0 IP subnet mask.<br>Enter **?** to display what you must enter next on the command line. In this example, you can enter the **secondary** keyword, or you can press **Enter**.<br>A <cr> is displayed; you can press **Enter** to complete the command, or you can enter another keyword. |
| Router(config-if)# **ip address 172.16.0.1**<br>**255.255.255.0**<br>Router(config-if)# | In this example, **Enter** is pressed to complete the command. |

Almost every configuration command has a **no** form. In general, use the **no** form to disable a function. Use the command without the **no** keyword to re-enable a disabled function or to enable a function that is disabled by default. For example, IP routing is enabled by default. To disable IP routing, use the **noip routing** command. To re-enable IP routing, use the **ip routing** command once again.

The Cisco IOS XE software command reference publications provide the complete syntax for the configuration commands and describe what the **no** form of a command does.

Many CLI commands also have a **default** form. By issuing the command **default** *command-name*, you can configure the command to its default setting. The Cisco IOS XE software command reference publications describe the function of the **default** form of the command when the **default** form performs a different function than the plain and **no** forms of the command.

To see what default commands are available on your system, enter **default ?** in the appropriate command mode.

Use the **copy running-config startup-config** command to save your configuration changes to the startup configuration so that the changes are not lost if the software reloads or a power outage occurs. For example:

```
Router# copy running-config startup-config
Building configuration...
```

It might take a minute or two to save the configuration. After the configuration has been saved, the system displays the following output:

```
[OK]
Router#
```

This task saves the configuration to NVRAM.

On the Cisco Catalyst 8000V instance, the startup configuration file is stored in the NVRAM partition. As a matter of routine maintenance on any Cisco router, users should backup the startup configuration file by copying the startup configuration file from the NVRAM to one of the router’s other file systems and in a network server. Backing up the startup configuration file provides an easy method of recovering the startup configuration file in the event the startup configuration file in the NVRAM becomes unusable for any reason.

Use the **copy**  command to backup startup configuration files. The following examples show the startup configuration file in NVRAM being backed up:

### Example 1: Copying a Startup Configuration File to Bootflash

```
Router# dir bootflash:
Directory of bootflash:/
   11      drwx    16384     Jan 24 2012 04:53:55 -05:00     lost+found
   12      -rw-    289243620 Jan 24 2012 04:54:55 -05:00
   308257  drwx    4096      Jan 24 2012 04:57:06 -05:00    core
   876097  drwx    4096      Jan 24 2012 04:57:07 -05:00   .prst_sync
   63277  drwx   4096     Jan 24 2012 04:57:10 -05:00 .rollback_timer   13
      -rw-       0      Jan 24 2012 04:57:19 -05:00    tracelogs.
csr1000v-adventerprisek9.2012-01-23_12.39.SSA.bin
Router# copy nvram:startup-config bootflash:
Destination filename [startup-config]?
3517 bytes copied in 0.647 secs (5436 bytes/sec)
Directory of bootflash:/
   11      drwx    16384     Jan 24 2012 04:53:55 -05:00    lost+found
   12      -rw-    289243620 Jan 24 2012 04:54:55 -05:00
   308257  drwx    4096      Jan 24 2012 04:57:06 -05:00    core
   876097  drwx    4096      Jan 24 2012 04:57:07 -05:00   .prst_sync
   632737 drwx   4096     Jan 24 2012 04:57:10 -05:00 .rollback_timer   13
      -rw-       0      Jan 24 2012 04:57:19 -05:00    tracelogs.
```

```
csr1000v-adventerprisek9.2012-01-23_12.39.SSA.bin
   14 -rw-   7516       Jul 2 2012 15:01:39 -07:00     startup-config
```

### Example 2: Copying a Startup Configuration File to a TFTP Server

```
Router# copy bootflash:startup-config tftp:
Address or name of remote host []? 172.17.16.81
Destination filename [pe24_asr-1002-confg]? /auto/tftp-users/user/startup-config
!!
3517 bytes copied in 0.122 secs (28828 bytes/sec)
```

For detailed information on managing configuration files, see the *Managing Configuration Files* section in the *Configuration Fundamentals Configuration Guide*.

- NVRAM File Security, on page 107

# NVRAM File Security

Cisco Catalyst 8000V allows you to encrypt some of the disk partitions internal to the VM to provide extra security around sensitive data that may be stored on the routers. For example, information in the NVRAM is encrypted so that it is not visible to administrative entities with access to the physical hard disk upon which Cisco Catalyst 8000V is stored.

You can search and filter the output of **show** and **more** commands. This functionality is useful if you need to sort through large amounts of output or if you want to exclude output that you need not see.

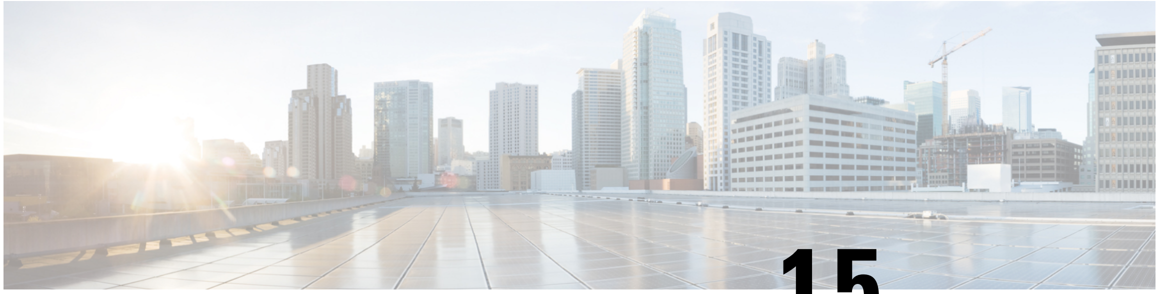
To use this functionality, enter a **show** or **more** command followed by the “pipe” character ( | ); one of the keywords **begin**, **include**, or **exclude**; and a regular expression on which you want to search or filter (the expression is case sensitive):

**show** *command* { **append** | | **begin** | | **exclude** | | **exclude** | | **include** | | **redirect** | | **section** | | **tee** } *regular-expression*

**show** *command* | {**append** | **begin** | **exclude** | **include** | **redirect** | **section** | **tee**} *regular-expression*

The output matches certain lines of information in the configuration file.

To power off a Cisco Catalyst 8000V instance, you must power off the VM upon which the router is installed. For information about powering off the VM, see your VM vendor documentation.



CHAPTER 15

# Upgrading the Cisco IOS XE Software

The Cisco Catalyst 8000V router runs on the Cisco IOS XE platform, the same platform that has powered Cisco Cloud Services Router 1000V or the Cisco ISRv. Starting from the 17.4.1 release, the Cisco Catalyst 8000V software is available in the form of .iso, .bin, .ova, and qcow2 formats. If you are an existing Cisco CSR1000V or a Cisco ISRv user, you can upgrade to Cisco Catalyst 8000V by following the procedures in this feature document.

This chapter describes how to upgrade the Cisco IOS XE software for an existing Cisco Catalyst 8000V installation on a VM. For information on installing a new Cisco Catalyst 8000V instance, see the *Cisco Catalyst 8000V Overview* section in this guide.

**Note** Upgrading a Cisco ISRv or a Cisco CSR1000V to a Cisco Catalyst 8000V does not alter the filesystem layout nor provide any of the new features such as the Secure Object Store which rely on the filesystem. You must perform a fresh installation to activate these features.

**Important** If you are an existing Cisco CSR1000V or Cisco ISRV user, and you are upgrading to Cisco Catalyst 8000V, your licenses will continue to function as is. However, if you to switch to the CDNA licensing model, you must perform a fresh installation.

# Prerequisites Before you Upgrade

- Obtain the Cisco Catalyst 8000V software image from Cisco.com. For Cisco Catalyst 8000V, see *Obtaining the Cisco Catalyst 8000V VM Image*

- Check the version of your hypervisor before you perform the upgrade. The upgrade is not successful if your hypervisor version is not supported by your current version of Cisco IOS XE on Cisco Catalyst 8000V.
- Ensure you meet the memory requirements of the VM for Cisco Catalyst 8000V software image. If the upgraded version requires more memory than your previous version, increase the memory allocation on the VM before beginning the upgrade process.

# Restrictions for the Upgrade Process

- This procedure is for upgrading to a new software version on the same VM only. It does not describe how to install or rehost an existing router running the same or upgraded software version on a different VM.
- The .bin file is applicable for upgrading or downgrading your software. The .iso and .ova files are used for first-time installation only.
- The Cisco Catalyst 8000V router does not support In-Service Software Upgrade (ISSU).
- 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 you previously had a Cisco CSR1000V and upgraded to Cisco Catalyst 8000V, you can downgrade in the case of CiscoCSR1000V but not Cisco ISRV.

# Saving Backup Copies of Your Old System Image and Configuration

To avoid unexpected downtime in the event you encounter serious problems using a new system image or startup configuration, Cisco recommends that you save backup copies of your current startup configuration file and Cisco IOS XE software system image file on a server.

To save backup copies of the startup configuration file and the system image file, perform the following steps.

**Step 1** **enable**

**Example:**

```
Router> enable
```

Enables the privileged EXEC mode. Enter your password, if prompted.

**Step 2** **copy nvram:startup-config** {**ftp:** | **rcp:** | **tftp:**}

**Example:**

```
Router# copy nvram:startup-config ftp:
```

Copies the startup configuration file to a server.

- The configuration file copy can serve as a backup copy.
- Enter the destination URL, when prompted.

**Step 3** **dir bootflash:**

**Example:**

```
Router# dir bootflash:
```

Displays the layout and contents of a bootflash memory file system. **bootflash:** is aliased onto **flash:**. Learn the name of the system image file.

**Step 4** **copy bootflash:** {**ftp:** | **rcp:** | **tftp:**}

**Example:**

```
Router# copy bootflash: ftp:
```

Copies a file from the bootflash memory to a server.

- Copy the system image file to a server. This file serves as a backup copy.
- Enter the bootflash memory partition number, if prompted.
- Enter the filename and destination URL, when prompted.

### What to do next

The following example shows the startup configuration being copied to a TFTP server:

```
Router# copy nvram:startup-config tftp:

Remote host[]? 192.0.2.1

Name of configuration file to write [rtr2-confg]? rtr2-config-b4upgrade

Write file rtr2-confg-b4upgrade on host 192.0.0.1?[confirm] <cr>

![OK]
```

The following example uses the **dir bootflash:** command in the EXEC mode to learn the name of the system image file and the **copy bootflash: tftp:** command in the privileged EXEC mode to copy the system image to a TFTP server. The router uses the default user name and password.

```
Router#
Router# dir bootflash:
Directory of bootflash:/
   1  -rw-    48311224   Mar 2 1901 11:32:50 +00:00 csr1000v-universalk9-mz.SSA.XFR_20090407

    2 -rw-         983  Feb 14 2021 12:41:52 +00:00  running-config
260173824 bytes total (211668992 bytes free)
Router# copy bootflash: tftp:
Source filename [running-config]?
Address or name of remote host []? 192.0.2.1
Destination filename [router-confg]? running-config
```

```
983 bytes copied in 0.048 secs (20479 bytes/sec)
Router#
```

# Installing Subpackages from a Consolidated Package

### Before you begin

Copy the consolidated package to the TFTP server. erform the following command steps to install the subpackages:

**Step 1** Run **show version** to check the current version of your router.

**Step 2** Run the **dir bootflash:** command to display the list of files in the bootflash.

**Step 3** **show platform**

**Step 4** Run the **mkdir bootflash: URL-to-directory-name** command to create a directory.

**Step 5** Run the **request platform software package expand file URL-to-consolidated-package to URL-to-directory-name** to extract the individual modules from a Cisco IOS-XE image.

**Step 6** Run the **reload** command to perform a reload.

**Step 7** Run the **boot URL-to-directory-name/packages.conf** command to boot the router.

**Step 8** Run **show version installed** to verify whether the upgraded version is installed.

```
Cisco IOS XE Software, Version 2020-09-17_09.24_kam
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 kami


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"
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:
License Type: Perpetual
Next reload license Level:

Addon License Level:
Addon License Type: Subscription
Next reload addon license Level:

The current throughput level is 10000 kbps


Smart Licensing Status: Registration Not Applicable/Not Applicable

cisco C8000V (VXE) processor (revision VXE) with 2036249K/3075K bytes of memory.
Processor board ID 94RQ5TL0K67
Router operating mode: Autonomous
3 Gigabit Ethernet interfaces
32768K bytes of non-volatile configuration memory.
3965952K bytes of physical memory.
5234688K bytes of virtual hard disk at bootflash:.

Configuration register is 0x2102
```

# Configuring the Router to Boot the Consolidated Package via TFTP/RCP

The following details the logistics of upgrading the system image. Install a TFTP server or an RCP server application on a TCP/IP-ready workstation or a hardware containing the virtualization extension. Many third-party vendors provide free TFTP server software. You can find these software by searching for “TFTP server” in a web search engine.

If you use TFTP:

- Configure the TFTP application to operate as a TFTP server and not a TFTP client.
- Specify the outbound file directory to which you download and store the system image.
- Download the new Cisco IOS XE software image into the workstation or the x86 hardware containing virtualization extension.
- Verify that the TFTP or RCP server has IP connectivity to the router. If you cannot successfully ping between the TFTP or RCP server and the router, either configure a default gateway on the router, or make sure that the router and server each have an IP address in the same network or subnet.

**Step 1** **enable**

Use this command to enter the privileged EXEC mode. Enter your password, if prompted:

**Example:**

```
Router> enable

Password: <password>

Router#
```

**Step 2** Use one of the following commands to copy a file from a server to the bootflash memory:

- **copy tftp bootflash:**
- **copy rcp bootflash**

**Example:**

```
Router# copy tftp bootflash:
```

**Step 3** When prompted, enter the IP address of the TFTP or RCP server.

**Example:**

```
Address or name of remote host []? 10.10.10.2
```

**Step 4** When prompted, enter the filename of the Cisco IOS software image that you are installing.

**Example:**

```
Source filename ? csr1000v-universalk9.03.10.00.S.153-3.S-ext.SPA.bin
```

**Note** The filename is case sensitive.

**Step 5** When prompted, enter the filename as you want it to appear on the router. Typically, the same filename that was used in the previous step is entered.

**Example:**

```
Destination filename ? csr1000v-universalk9.03.10.00.S.153-3.S-ext.SPA.bin
```

**Step 6** If the system displays the following error message - *Not enough space on device* - do the following:

- If you are certain that all the files in the bootflash memory should be erased, enter **y** when prompted twice. This action confirms that the bootflash memory will be erased before copying.

**Example:**

```
Accessing tftp://10.10.10.2/csr1000v-universalk9.03.10.00.S.153-3.S-ext.SPA.bin...
     Erase bootflash: before copying? [confirm] y
     Erasing the flash filesystem will remove all files! Continue? [confirm] y

     Erasing device...
```

- If you are not certain that all the files in the bootflash memory should be erased, press **Ctrl-Z**.

**Step 7** If the error message does not appear, enter **no** when prompted to erase the bootflash memory before copying.

**Example:**

```
Accessing tftp://10.10.10.2/csr1000v-universalk9.03.10.00.S.153-3.S-ext.SPA.bin...
     Erase bootflash: before copying? [confirm] no
```

# Loading the New System Image from the Cisco IOS XE Software

**Step 1** **dir bootflash:**

Use this command to display a list of all files and directories in the bootflash memory:

**Example:**

```
Router# dir bootflash:

Directory of bootflash:/
    3  -rw-     6458388   Mar 01 1993 00:00:58 csr1000v.tmp
 1580  -rw-     6462268   Mar 06 1993 06:14:02 csr1000v-ata
63930368 bytes total (51007488 bytes free)
```

**Step 2** **configure terminal**

Use this command to enter the global configuration mode.

**Example:**

```
Router# configure terminal
Router(config)#
```

**Step 3** **no boot system**

Use this command to delete all the entries in the bootable image list which specifies the order in which the router attempts to load the system images at the next system reload or power cycle:

**Example:**

```
Router(config)# no boot system
```

**Step 4** **boot system bootflash:***system-image-filename*.bin

**Note** If the new system image is the first file or the only file displayed in the **dir bootflash:** command output in Step 1, you need not perform this step.

Use this command to load the new system image after the next system reload or power cycle. For example:

**Example:**

```
Router(config)# boot system bootflash:
c8000v-universalk9.17.04.01.SPA.bin
```

**Step 5** (Optional) Repeat the previous step to specify the order in which the router should attempt to load any backup system images.

**Step 6** **exit**

Use this command to exit the global configuration mode.

**Example:**

```
Router(config)# exit
 Router#
```

**Step 7** **write**

or

**write memory**

**Example:**

```
Router# write memory
```

**Note** Entering the **write** or **write memory** command updates the GRUB menu list of images available on the bootflash disk.

**Step 8** **show version**

Use this command to display the configuration register setting.

**Example:**

```
Router# show version

Cisco Internetwork Operating System Software
.
.
.
Configuration register is 0x0

Router#
```

**Step 9** If the last digit in the configuration register is 0 or 1, proceed to the next step. However, if the last digit in the configuration register is between 2 and F, proceed to the "copy running-config startup-config" step in this procedure.

**Step 10** **configure terminal**

Use this command to enter the global configuration mode.

**Example:**

```
Router# configure terminal
Router(config)#
```

**Step 11** **config-register 0x2102**

Use this command to set the configuration register so that after the next system reload or power cycle, the router loads a system image from the **boot system** command in the startup configuration file.

**Example:**

```
Router(config)# config-register 0x2102
```

**Note** The 0x2102 value is the default configuration register setting. If you didn’t change this setting from the default, this step is not required.

**Step 12** **exit**

Use this command to exit the global configuration mode.

**Example:**

```
Router(config)# exit
 Router#
```

**Step 13** **copy running-config startup-config**

Use this command to copy the running configuration to the startup configuration.

**Example:**

```
Router# copy running-config startup-config
```

**Step 14** **write memory**

**Note** Entering the **write memory** command updates the GRUB menu list of the images available on the bootflash disk.

**Step 15** **reload**

Use this command to reload the operating system.

**Example:**

```
Router# reload
```

**Step 16** When prompted to save the system configuration, enter **no**:

**Example:**

```
System configuration has been modified. Save? [yes/no]: no
```

**Step 17** When prompted to confirm the reload, enter **y**:

**Example:**

```
Proceed with reload? [confirm] y
```

**Step 18** **show version**

Use this command to verify that the router loaded the proper system image.

**Example:**

```
Router# show version
00:22:25: %SYS-5-CONFIG_I: Configured from console by console
Cisco Internetwork Operating System Software
.
.
.
System returned to ROM by reload
System image file is "bootflash:
c8000v-universalk9.17.04.01.SPA.bin"

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"
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:
License Type: Perpetual
Next reload license Level:

Addon License Level:
Addon License Type: Subscription
Next reload addon license Level:

The current throughput level is 10000 kbps


Smart Licensing Status: Registration Not Applicable/Not Applicable

cisco C8000V (VXE) processor (revision VXE) with 2036249K/3075K bytes of memory.
Processor board ID 94RQ5TL0K67
Router operating mode: Autonomous
3 Gigabit Ethernet interfaces
32768K bytes of non-volatile configuration memory.
3965952K bytes of physical memory.
5234688K bytes of virtual hard disk at bootflash:.

Configuration register is 0x2102
```

# Entering the GRUB Mode and Selecting the Image

To load the new system image from the GR and Unified Bootloader (GRUB) mode, follow these steps, beginning in EXEC mode.

**Step 1** **dir bootflash:**

Use this command to display a list of all files and directories in bootflash memory:

**Example:**

```
Router# dir bootflash:

Directory of bootflash:/
    3  -rw-     6458388   Mar 01 1993 00:00:58  csr1000v.tmp
 1580  -rw-     6462268   Mar 06 1993 06:14:02 csr1000v-ata
63930368 bytes total (51007488 bytes free)
```

**Step 2** **configure terminal**

Use this command to enter the global configuration mode:

**Example:**

```
Router# configure terminal
Router(config)#
```

**Step 3** **boot system bootflash:***system-image-filename*.bin

Use this command to load the new system image after the next system reload or power cycle. For example:

**Example:**

```
Router(config)# boot system bootflash:
c8000v-universalk9.03.10.00.S.153-3.S-ext.SPA.bin
```

**Note** If the new system image is the first file or the only file displayed in the **dir bootflash:** command output , you do not need to perform this step.

**Step 4** **do write**

or

**do write memory**

**Example:**

```
Router(config)# do write memory
```

**Note** Entering the **do write** or **do write memory** command updates the GRUB menu list of images available on the bootflash disk.

**Step 5** **config-register 0x0000**

Use this command to enter the GRUB mode.

The following shows a sample configuration output of entering 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.

**Step 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:**

```
GNU GRUB  version 0.97  (638K lower / 3143616K upper memory)
 +-------------------------------------------------------------------------+
 | C8000v - c8000v-universalk9.03.10.00.S.153-3.S-ext.SPA.bin          |
 | C8000v - packages.conf                                                |
 | 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.
```

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**.

**Step 7** Select the .bin file to upgrade the software image on the router to the new version.

**Step 8** Press **Enter** to boot the selected image which begins the upgrade process.

# Saving Backup Copies of Your New System Image and Configuration

To aid file recovery and to minimize the downtime in the event of file corruption, Cisco recommends that you save backup copies of the startup configuration file and the Cisco IOS software system image file on a server.

**Tip** Do not erase any existing backup copies of your configuration and system image that you saved before upgrading your system image. If you encounter serious problems using your new system image or startup configuration, you can quickly revert to the previous working configuration and system image.

To save backup copies of the startup configuration file and the system image file, complete the following steps.

**Step 1** **enable**

**Example:**

```
Router> enable
```

Enables the privileged EXEC mode. Enter your password, if prompted.

**Step 2** **copy nvram:startup-config** {**ftp:** | **rcp:** | **tftp:**}

**Example:**

```
Router# copy nvram:startup-config ftp:
```

Copies the startup configuration file to a server.

- The configuration file copy serves as a backup copy.
- Enter the destination URL, when prompted.

**Step 3** **dir bootflash:**

**Example:**

```
Router# dir bootflash:
```

Displays the layout and contents of a bootflash memory file system. Write down the name of the system image file.

**Step 4** **copy bootflash:** {**ftp:** | **rcp:** | **tftp:**}

**Example:**

```
Router# copy bootflash: ftp:
```

Copies a file from bootflash memory to a server.

- Copy the system image file to a server to serve as a backup copy.
- Enter the bootflash memory partition number, if prompted.
- Enter the filename and destination URL, when prompted.

### What to do next

The following example shows the startup configuration being copied to a TFTP server:

```
Router# copy nvram:startup-config tftp:

Remote host[]? 172.16.101.101

Name of configuration file to write [rtr2-confg]? <cr>

Write file rtr2-confg on host 172.16.101.101?[confirm] <cr>

![OK]
```

The following example uses the **dir bootflash:** privileged EXEC command to obtain the name of the system image file and the **copy bootflash: tftp:** privileged EXEC command to copy the system image to a TFTP server. The router uses the default user name and password.

```
Router# dir bootflash:

System flash directory:
File Length Name/status
1 4137888 csr1000v-universalk9.03.10.00.S.153-3.S-ext.SPA.bin
[4137952 bytes used, 12639264 available, 16777216 total]
16384K bytes of processor board System flash (Read/Write)\
Router# copy bootflash: tftp:

IP address of remote host [255.255.255.255]? 192.0.2.1
filename to write on tftp host? csr1000v-advernterprisek9-mz

writing csr1000v-universalk9.03.10.00.S.153-3.S-ext.SPA!!!!...
successful ftp write.
```

# Rebooting the Cisco Catalyst 8000V

After you copy the new system image into the bootflash memory, load the new system image, and save a backup copy of the new system image and configuration, you must reboot the VM. See your VM vendor documentation for more information about rebooting the VM. After rebooting, the router VM should include the new system image with a newly installed version of the Cisco IOS XE software.

CHAPTER 16

# Accessing and Using GRUB Mode

The Cisco Catalyst 8000V has a 16-bit configuration register in NVRAM. Each bit has value 1 (on or set) or value 0 (off or clear), and each bit setting affects the router behavior upon the next reload power cycle. The GRUB mode supports a subset of configuration register options compared to ROMMON options on other Cisco routers.

You can use the configuration register to:

- Force the router to boot into the GRUB (bootstrap program)
- Select a boot source and default boot filename
- Recover a lost password

The table below describes the configuration register bits.

***Table 24: Configuration Register Bit Descriptions***

| BitNumber | Hexadecimal | Meaning |
|---|---|---|
| 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.<br>See the table "Boot Field Configuration Register Bit Descriptions" for details. |
| 06 | 0x0040 | Causes the system software to ignore the contents of NVRAM. This can be used for password recovery. |

The next table describes the boot field, which is the lowest four bits of the configuration register (bits 3, 2, 1, and 0). The boot field setting determines whether the router loads an operating system.

*Table 25: Boot Field Configuration Register Bit Descriptions*

| Boot Field(Bits 3, 2, 1, and 0) | Meaning |
|---|---|
| 0000<br>(0x0) | At the next power cycle or reload, the router boots to the GRUB (bootstrap program).<br>In the GRUB mode, you must manually boot the system image or any other image by using the **boot** command. |
| 0001 - 1111<br>(0x01 - 0x0F) | At the next power cycle or reload, the router sequentially processes each **boot system** command in global configuration mode that is stored in the configuration file until the system boots successfully.<br>If no **boot system** commands are stored in the configuration file, or if executing those commands is unsuccessful, then the router attempts to boot the first image file in flash memory. |

**Note** Use the 0x000 setting to configure the router to automatically enter the GRUB mode when the router reboots.

- Accessing GRUB Mode, on page 124
- Using the GRUB Menu, on page 125
- Modifying the Configuration Register (confreg), on page 126
- Changing the Configuration Register Settings, on page 127
- Displaying the Configuration Register Settings, on page 128

# Accessing GRUB Mode

Perform the following step to access GRUB mode:

**Step 1** **enable**

**Example:**

```
Router> enable
```

Enables the privileged EXEC mode.

- Enter your password, if prompted.

**Step 2** **config-register 0x0000**

**Example:**

```
Router# config-register 0x0000
```

Enters the GRUB mode by entering the “0000” value (0x0).

### What to do next

The following shows an example of entering GRUB mode.

```
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>
```

If you enter a question mark at the grub> prompt, the system shows you the two options available - for either viewing the system help or for entering the **confreg** command.

# Using the GRUB Menu

The GRUB menu is used to display the software images loaded on the router, and to select which image to boot from. To access the GRUB menu, enter **ESC** at the GRUB prompt. The following shows the GRUB menu display.

```
GNU GRUB  version 0.97  (638K lower / 3143616K upper memory)
 +-------------------------------------------------------------------------+
 | CSR1000v - csr1000v-universalk9.03.10.00.S.153-3.S-ext.SPA.bin          |
 | CSR1000v - packages.conf                                                |
 | CSR1000v - 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.
```

Select the image to boot the router from using the up and down arrow key. To return to the GRUB prompt, enter the letter **c**.

# Modifying the Configuration Register (confreg)

This section describes how to modify the configuration register by using the **confreg** GRUB command. This command is similar to the **confreg** ROMMON command on other Cisco hardware routers. Because the router does not include a ROMMON mode, the similar functionality is handled in GRUB command mode.

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, but the new value does not take effect until you reset or power-cycle the router.

**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. The value range is from 0x0 to 0xFFFF.
- If you do not enter the value, the router prompts for each bit of the 16-bit configuration register.

### What to do next

The following code is an example of 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 0.97  (638K lower / 3143616K upper memory)
-------------------------------------------------------------------
 0: CSR1000v - packages.conf
 1: CSR1000v - csr100v-packages-universalk9
 2: CSR1000v - 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 'CSR1000v - 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]
```

# Changing the Configuration Register Settings

You can change the configuration register settings from either the GRUB or the Cisco IOS XE CLI. This section describes how to modify the configuration register settings from the Cisco IOS XE CLI.

To change the configuration register settings from the Cisco IOS XE CLI, complete the following steps:

**Step 1** Power on the router.

**Step 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
```

After a few seconds, the system displays the user EXEC prompt ( Router> ).

**Step 3** Enter the privileged EXEC mode by typing enable, and if prompted, enter your password:

**Example:**

```
Router> enable
Password: password
Router#
```

**Step 4** Enter 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
```

**Step 5** To change the configuration register settings, enter the **config-register** *value* command, where *value* is a hexadecimal number preceded by **0x**:

**Example:**

```
Router(config)# config-register 0x
value
```

**Step 6** Exit the global configuration mode:

**Example:**

```
Router(config)# end
Router#
```

**Step 7** Save the configuration changes to NVRAM:

Router# **copy running-config startup-config**

The new configuration register settings are saved to NVRAM, but they do not take effect until the next router reload or power cycle.

# Displaying the Configuration Register Settings

To display the configuration register settings that are currently in effect and the settings that will be used at the next router reload, enter the **show version** command in privileged EXEC mode.

The configuration register settings are displayed in the last line of the **show version** command output:

```
Configuration register is 0x142 (will be 0x142 at next reload)
```

CHAPTER 17

# Performing a Factory Reset

This chapter provides information on performing a factory reset for Cisco Catalyst 8000V. The factory reset feature helps remove any sensitive information from the router, or to reset the router to a fully functional state.

# Information About Factory Reset

The 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.

**Note** The time taken for factory reset on a Cisco Catalyst 8000V instance is dependent on factors such as the type of storage and the devices present on the router.

### Information deleted:

When you perform a factory reset, the following information is deleted:

- Licenses – user installed, and manufacturer provided
- Non-volatile random-access memory data
- User credentials
- Start-up configuration
- All writable file systems and personal data
- ROMMON variable
- Persistent storage devices
- Any containers running on bootflash

**Information retained:**

However, the following information will be retained even after the factory reset:

- Critical information including files that provide access to the router after the reset is complete
- The software packages that are installed before you perform factory reset
- UDI and Smart Licensing files

**Supported Scenarios:**

You can use the factory reset feature in the following 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 once again for further use.

**Supported Platforms:**

Factory reset is supported on a Cisco Catalyst 8000V instance running on all the platforms including Amazon Web Services, Microsoft Azure, GCP cloud, VMware ESXi, and Hyper-V.

# Prerequisites for Performing Factory Reset

- Ensure that you take a backup of all the software images, configurations and personal data before performing the factory reset operation.
- Ensure that there is uninterrupted power supply when the feature reset process is in progress.
- Ensure that the instance has at least 8 GB memory in the bootflash.

# Restrictions for Performing a Factory Reset

- Any software patches that are installed on the router are not restored after the factory reset operation.
- You must not restart the Cisco Catalyst 8000V instance during the factory reset process.
- 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.

# How to Perform a Factory Reset

**Step 1** Log in to a Cisco Catalyst 8000V instance.

**Step 2** At the command prompt, execute the **factory-reset all** command.

The system displays the following:

```
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 35.231.25.29 closed by remote host.
Connection to 135.231.25.29 closed.
```

**Step 3** Enter confirm to proceed with the factory reset.

**Note** The time taken for the factory reset process depends on the type of storage and on which cloud service you deploy the Cisco Catalyst 8000V instances.

**Note** If you want to quit the factory reset process, press the **Escape** key.

### What to do next

After the factory reset process is completed, you receive a log file in the bootflash that indicates whether the process was successful or not.

## Restoring Smart Licensing after a Factory Reset

After the reset, Smart Licensing configuation is also deleted. You must reconfigure Smart Licensing on the router by using the token ID. In the connected mode, when you register your instance for Smart Licensing, you must use the force option. That is, you must use the **license smart register idtoken *****token***** force** command. The registration process begins.

When you do not use the force option, and configure Smart Licensing directly, the license registration fails. The following is an example of a 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:CSR1000V has already been registered.

License Authorization:
  Status: No Licenses in Use

Export Authorization Key:
  Features Authorized:
```

After you execute the license smart register idtoken *****token***** force command, the license goes to the Registered state. The following is an example of a configuration output in the Registered state:

```
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: REGISTERED
  Smart Account: InternalTestDemoAccount8.cisco.com
  Virtual Account: RTP-CSR-DT-Prod
  Export-Controlled Functionality: ALLOWED
  Initial Registration: SUCCEEDED on Feb 15 22:04:07 2019 UTC
  Last Renewal Attempt: None
  Next Renewal Attempt: Aug 14 22:04:06 2019 UTC
  Registration Expires: Feb 15 21:59:05 2020 UTC

License Authorization:
  Status: AUTHORIZED on Feb 15 22:04:11 2019 UTC
  Last Communication Attempt: SUCCEEDED on Feb 15 22:04:11 2019 UTC
  Next Communication Attempt: Mar 17 22:04:11 2019 UTC
  Communication Deadline: May 16 21:58:10 2019 UTC

Export Authorization Key:
  Features Authorized:
    <none>
```

# What Happens after a Factory Reset

After the factory reset is successfully completed, the router boots up. However, before the factory reset process started, if the configuration register was set to manually boot from ROMMON, the router stops at ROMMON.

**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.

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, execute the **factoryreset#show license udi** command before and after the factory reset process.

After you configure Smart Licensing, execute the **#show license status** command, to check whether Smart Licensing is enabled for your instance.

**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.

CHAPTER 18

# Troubleshooting VM Issues

## Troubleshooting Network Connectivity Issues

To troubleshoot network connectivity issues for Cisco Catalyst 8000V, do the following:

- Verify that there is an active and unexpired license installed on the VM.

Enter the **show license** command. The License State should be shown as “Active, In Use”.

- Verify that the vNIC for the VMs are connected to the correct physical NIC, or to the proper vSwitch.
- If you're using virtual LANS (VLANs), ensure the vSwitch is configured with the correct VLAN.
- If you're using static MAC addresses or VMs that are cloned, make sure there are no duplicate MAC addresses.

  Duplicate MAC addresses can cause the Cisco Catalyst 8000V feature license to become invalidated, which will disable the router interfaces.

## Troubleshooting VM Performance Issues

Cisco Catalyst 8000V operates within a set of supported VM parameters and settings to provide certain levels of performance that have been tested by Cisco. Use the vSphere Client to view data to troubleshoot VM performance. If you are using vCenter, you can view historical data. If you are not using vCenter, you can view live data from the host.

This is a list of troubleshooting tips for performance issues:

### Troubleshooting - MTU

Verify that the router has the correct setting for maximum MTU.

By default, the maximum MTU on the router is 1500. To support jumbo frames, edit the default VMware vSwitch settings. For more information, see the VMware vSwitch documentation.

**Note** ESXi 5.0 supports a maximum MTU of 9000, even if jumbo frames are enabled on the router.

## Troubleshooting—Memory

Cisco Catalyst 8000V does not support memory sharing between VMs. On the ESXi host, check the memory counters to find out how much used memory and shared memory is on the VM. Verify that the balloon and swap used counters are zero.

If a specific VM does not have enough memory to support Cisco Catalyst 8000V, increase the memory size of the VM. Insufficient memory on the VM or the host can cause the Cisco Catalyst 8000V console to hang and be non-responsive.

**Note** With troubleshooting performance issues, note that other VMs on the same host as the Cisco Catalyst 8000V can impact the performance of the Cisco Catalyst 8000V VM. Verify that the other VMs on the host are not causing memory issues that are impacting the Cisco Catalyst 8000V VM.

## Troubleshooting - Network Packets

Verify that no network packets are being dropped. On the ESXi host, check the network performance and view the counters to measure the number of receive packets and transmit packets dropped.

## Troubleshooting - Throughput

Verify the current maximum throughput level with the **show platform hardware throughput level** command.

## Troubleshooting - Instruction Extensions

Some x86 processors support instruction extensions for performing certain cryptographic transforms. Using these instructions is more efficient than not using them. Cisco Catalyst 8000V detects at run-time if the instruction extensions are available and will use them if they are available. To determine if the extensions are available, enter the **show platform software system all** command.

If the output shows that "Crypto Supported" is "No", then Cisco Catalyst 8000V may not exhibit the expected throughput. This is an issue with either the underlying physical hardware or the hypervisor. Check to see if the underlying physical hardware is capable of exposing the extensions and also check to see if the hypervisor can expose the extensions.

If the output shows that "Crypto Supported" is "Yes", then Cisco Catalyst 8000V provides the expected throughput because the physical hardware and the hypervisor can expose the extensions.

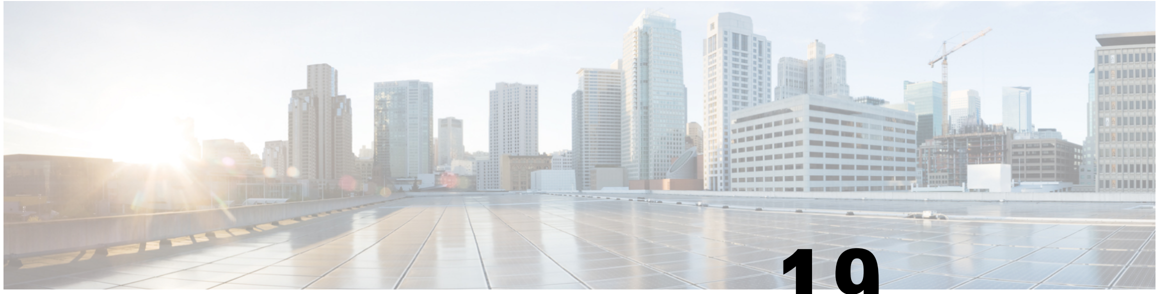
In the following example, "Crypto Supported" is "Yes". Therefore the cryptographic transforms can use instruction extensions, and perform efficiently.

```
Router# show platform software system all
Processor Details
=================
Number of Processors : 4
Processor : 1 - 4
vendor_id : GenuineIntel
cpu MHz  : 3192.307
cache size : 20480 KB
Crypto Supported : Yes
```

# IP address Inconsistency Issues on the vSphere Web Client

You might face inconsistencies in the IP addresses that is configured on the router and what is shown on the vSphere Web Client. At this moment there are no resolutions for this issue. See the following list to know why these inconsistencies might occur:

- ipv4 addresses for interfaces that are up or down are detected, while ipv6 addresses are only detected for interfaces that are up.
- After you perform an Interface Hot Delete, the vSphere Web Client continues to display the IP Address of the deleted interface.
- When you perform a reload on a Cisco Catalyst 8000V with addresses configured but not written to memory, the vSphere Web client continues to display the addresses even after the router comes up again. This occurs even though there are no addresses configured on the router. For example, configure Loopback, port-channel, port-group, and subinterfaces on a Cisco Catalyst 8000V router so that 63 addresses are displayed by the vSphere Web Client. Do not write the configuration to memory and reload the Cisco Catalyst 8000V. After the reload completes, all the 63 addresses are displayed on the Web Client. This occurs even though no addresses are configured on the Cisco Catalyst 8000V router. You can resolve this issue by configuring an address on the Cisco Catalyst 8000V router. When you do so, the web client then removes the 63 address and just displays the newly configured address.
- When you configure multiple ipv6 addresses on an interface, only the last address that you configured is detected. If you remove that address, none of the remaining configured ipv6 address on that interface are detected. This creates a state with multiple ipv6 addresses configured on an interface, but none displayed by the Web Client.
- When you delete interfaces, some of the addresses of the new interfaces are not displayed. This happens when the maximum number of IP Addresses are displayed and then you delete interfaces. For example, configure 32 Loopback interfaces with addresses and then delete each interface. Then, configure 32 GigabitEthernet sub interfaces with addresses. The addresses for the subinterfaces are not detected. This is because the router maintains entries for the deleted Loopback interfaces and is not able to add new interfaces.
- Addresses are detected for GigabitEthernet, Loopback, PortChannel, and VirtualPort-Group Interfaces as well as subinterfaces. However, Tunnel interface addresses are not detected.
- Secondary IP Addresses for IPv4 interfaces are not detected



CHAPTER 19

# Mapping the Cisco Catalyst 8000V Network Interfaces to VM Network Interfaces

## Mapping the Router Network Interfaces to vNICs

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 the 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. The following image shows the relationship between the vNICs and the Cisco Catalyst 8000V router interfaces.

After you boot the Cisco Catalyst 8000V instance, you need to display the mapping between the logical interface on the router with the vNIC and the vNIC MAC address using the **show platform software vnic-if interface-mapping** command. The output for this command depends on your Cisco IOS XE release version.

**Note** GigabitEthernet0 interface is no longer supported.

```
Router# show platform software vnic-if interface-mapping
---------------------------------------------------------------------------
Interface Name        Short Name     vNIC Name                Mac Addr
---------------------------------------------------------------------------
GigabitEthernet0       Gi0        eth0 (vmxnet3)         000c.2946.3f4d
GigabitEthernet2       Gi2        eth2 (vmxnet3)         0050.5689.0034
GigabitEthernet1       Gi1        eth1 (vmxnet3)         0050.5689.000b
---------------------------------------------------------------------------
```

The vNIC name shown in the display is a logical interface that the Cisco Catalyst 8000V instance uses to map to the interface on the hypervisor. It does not always map to the corresponding NIC name added during the

VM installation. For example, the logical “eth1” vNIC name in the display may not necessarily map to “NIC1” that was added in the VM installation process.

**Caution** It is important that you verify the interface mapping before you begin configuring the Gigabit Ethernet network interfaces onCisco Catalyst 8000V. This ensures that the network interface configuration applies to the correct physical MAC address interface on the VM host.

If you reboot the router and do not add or delete any vNICs, the interface mapping remains the same as before. If you reboot the router and delete vNICs, ensure that the configuration for the remaining interfaces remains intact. For more information, see *Adding and Deleting Network Interfaces on Cisco Catalyst 8000V*.

# Adding and Deleting Network Interfaces on Cisco Catalyst 8000V

Cisco Catalyst 8000V maps the router GigabitEthernet interfaces to the logical vNIC name assigned by the VM which in turn is mapped to a MAC address on the VM host. You can add or delete vNICs on the VM to add or delete GigabitEthernet interfaces on Cisco Catalyst 8000V. You can add vNICs while the router is active.

To delete a vNIC from the VM, you must first power down the VM. If you delete any vNICs, you must reboot the router. For more information about adding and deleting vNICs, see the VMware Documentation .

**Caution** If you remove a vNIC without first updating the Cisco Catalyst 8000V network interface configuration, you risk a configuration mismatch when the router reboots. When you reboot the router and remove a vNIC, the remaining logical vNIC names could get reassigned to different MAC addresses. As a result, the GigabitEthernet network interfaces on theCisco Catalyst 8000V instances can be reassigned to different physical interfaces on the hypervisor.

Before you add or delete network interfaces, first verify the interface-to-vNIC mapping using the **show platform software vnic-if interface-mapping** command.

```
csr1000v# show platform software vnic-if interface-mapping
--------------------------------------------------------------------------
Interface Name             Driver Name               Mac Addr
--------------------------------------------------------------------------
GigabitEthernet3            vmxnet3                 000c.2946.3f4d
GigabitEthernet2            vmxnet3                 0050.5689.0034
GigabitEthernet1            vmxnet3                 0050.5689.000b
GigabitEthernet0            vmxnet3                 000c.2946.3f4d
--------------------------------------------------------------------------
```

After adding or deleting network interfaces on the VM, verify the new interface-to-vNIC mapping before making configuration changes to the network interfaces. The following example shows the interface mapping after a new vNIC has been added. The new vNIC maps to the GigabitEthernet4 network interface on the Cisco Catalyst 8000V instance.

```
csr1000v# show platform software vnic-if interface-mapping
--------------------------------------------------------------------------
Interface Name             Driver Name               Mac Addr
--------------------------------------------------------------------------
GigabitEthernet4            vmxnet3                 0010.0d40.37ff
```

```
GigabitEthernet3          vmxnet3               000c.2946.3f4d
GigabitEthernet2          vmxnet3               0050.5689.0034
GigabitEthernet1          vmxnet3               0050.5689.000b
GigabitEthernet0          vmxnet3               000c.2946.3f4d
----------------------------------------------------------------------
```

# Removing a vNIC from a Running VM

To remove a vNIC from a running VM, use the `clear platform software` command (described below). Perform this command before removing a vNIC from the hypervisor configuration. This is part of a "two-step hot remove".

To see which hypervisors support a two-step hot remove, look for hypervisors with vNIC Two-Step Hot Remove Support = Yes

**clear platform software vnic-if interface GigabitEthernet***interface-number*

*interface-number* - value from 0–32.

Example:

```
Router# clear platform software vnic-if interface GigabitEthernet4
```

Next, remove the vNIC from the hypervisor configuration.

**Note** You no longer need to execute the `clear platform software vnic-int interface` command before you remove the vNIC configuration from the hypervisor. This command will be deprecated in a future release.

# Cisco Catalyst 8000V Network Interfaces and VM Cloning

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 the router network interfaces and the vNICs only applies to the current VM that the Cisco Catalyst 8000V is installed on. If the VM is cloned, the stored UUID will not match the current UUID and the interface mapping will not match the router configuration.

To prevent the interface mapping from becoming mis-matched, perform the following steps on the original VM before cloning:

**Note** Ensure that the original VM includes the number of configured vNICs required on the cloned VM before beginning the cloning process.

**Step 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.

**Step 2** Reboot the Cisco Catalyst 8000V.

**Step 3** On the cloned VM, verify the interface mapping using the **show platform software vnic-if interface-mapping** command.

**Step 4** Configure the router interfaces on the cloned VM accordingly.

The router configuration on the cloned VM should match the configuration of the original VM.

# Mapping the Cisco Catalyst 8000V Network Interfaces with vSwitch Interfaces

You can configure the network interfaces in ESXi in different ways to accommodate the Cisco Catalyst 8000V interfaces. You can configure the network interfaces so that each Cisco Catalyst 8000V router interface is mapped to one host Ethernet interface.

Alternatively, you can also configure the network interfaces so that multiple Cisco Catalyst 8000V interfaces share one host ESXi Ethernet interface.

The third possibility is mapping the Cisco Catalyst 8000V interfaces directly to a trunk interface on the vSwitch.

CHAPTER 20

# Packet Trace

First Published: August 03, 2016

The Packet-Trace feature provides a detailed understanding of how data packets are processed by the Cisco IOS XE platform, and thus helps customers to diagnose issues and troubleshoot them more efficiently. This module provides information about how to use the Packet-Trace feature.

# Information About Packet Trace

The Packet-Trace feature provides three levels of inspection for packets: accounting, summary, and path data. Each level provides a detailed view of packet processing at the cost of some packet processing capability. However, Packet Trace limits inspection to packets that match the debug platform condition statements, and is a viable option even under heavy-traffic situations in customer environments.

The following table explains the three levels of inspection provided by packet trace.

*Table 26: Packet-Trace Level*

| Packet-Trace Level | Description |
|---|---|
| Accounting | Packet-Trace accounting provides a count of packets that enter and leave the network processor. Packet-Trace accounting is a lightweight performance activity, and runs continuously until it is disabled. |
| Summary | At the summary level of packet trace, data is collected for a finite number of packets. Packet-Trace summary tracks the input and output interfaces, the final packet state, and punt, drop, or inject packets, if any. Collecting summary data adds to additional performance compared to normal packet processing, and can help to isolate a troublesome interface. |

| Packet-Trace Level | Description |
|---|---|
| Path data | The packet-trace path data level provides the greatest level of detail in packet trace. Data is collected for a finite number of packets. Packet-Trace path data captures data, including a conditional debugging ID that is useful to correlate with feature debugs, a timestamp, and also feature-specific path-trace data.<br>Path data also has two optional capabilities: packet copy and Feature Invocation Array (FIA) trace. The packet-copy option enables you to copy input and output packets at various layers of the packet (layer 2, layer 3 or layer 4). The FIA- trace option tracks every feature entry invoked during packet processing and helps you to know what is happening during packet processing.<br>**Note** Collecting path data consumes more packet-processing resources, and the optional capabilities incrementally affect packet performance. Therefore, path-data level should be used in limited capacity or in situations where packet performance change is acceptable. |

# Usage Guidelines for Configuring Packet Trace

Consider the following best practices while configuring the Packet-Trace feature:

- Use of ingress conditions when using the Packet-Trace feature is recommended for a more comprehensive view of packets.
- Packet-trace configuration requires data-plane memory. On systems where data-plane memory is constrained, carefully consider how you will select the packet-trace values. A close approximation of the amount of memory consumed by packet trace is provided by the following equation:

memory required = (statistics overhead) + number of packets * (summary size + data size + packet copy size).

When the Packet-Trace feature is enabled, a small, fixed amount of memory is allocated for statistics. Similarly, when per-packet data is captured, a small, fixed amount of memory is required for each packet for summary data. However, as shown by the equation, you can significantly influence the amount of memory consumed by the number of packets you select to trace, and whether you collect path data and copies of packets.

# Configuring Packet Trace

Perform the following steps to configure the Packet-Trace feature.

**Note** The amount of memory consumed by the Packet-Trace feature is affected by the packet-trace configuration. You should carefully select the size of per-packet path data and copy buffers and the number of packets to be traced in order to avoid interrupting normal services. You can check the current data-plane DRAM memory consumption by using the **show platform hardware qfp active infrastructure exmem statistics** command.

### SUMMARY STEPS

1. **enable**
2. **debug platform packet-trace packet** *pkt-num* **[fia-trace | summary-only] [circular] [data-size** *data-size***]**
3. **debug platform packet-trace {punt |inject|copy|drop|packet|statistics}**
4. **debug platform condition [ipv4 | ipv6] [interface** *interface***][access-list** *access-list -name* **|** *ipv4-address / subnet-mask* **|** *ipv6-address / subnet-mask***] [ingress | egress |both]**
5. **debug platform condition start**
6. **debug platform condition stop**
7. **show platform packet-trace {configuration | statistics | summary | packet {all |** *pkt-num***}}**
8. **clear platform condition all**
9. **exit**

### DETAILED STEPS

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable**<br>**Example:**<br>`Router> enable` | Enables the privileged EXEC mode. Enter your password if prompted. |
| **Step 2** | **debug platform packet-trace packet** *pkt-num* **[fia-trace \| summary-only] [circular] [data-size** *data-size***]**<br>**Example:**<br>`Router# debug platform packet-trace packets 2048 summary-only` | Collects summary data for a specified number of packets. Captures feature path data by default, and optionally performs FIA trace.<br>*pkt-num*—Specifies the maximum number of packets maintained at a given time.<br>**fia-trace**—Provides detailed level of data capture, including summary data, feature-specific data. Also displays each feature entry visited during packet processing.<br>**summary-only**—Enables the capture of summary data with minimal details.<br>**circular**—Saves the data of the most recently traced packets.<br>*data-size*—Specifies the size of data buffers for storing feature and FIA trace data for each packet in bytes. When very heavy packet processing is performed on packets, users can increase the size of the data buffers if necessary. The default value is 2048. |
| **Step 3** | **debug platform packet-trace {punt \|inject\|copy\|drop\|packet\|statistics}**<br>**Example:**<br>`Router# debug platform packet-trace punt` | Enables tracing of punted packets from data to control plane. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 4** | **debug platform condition [ipv4 \| ipv6] [interface** *interface***][access-list** *access-list -name* \| *ipv4-address* / *subnet-mask* \| *ipv6-address* / *subnet-mask***] [ingress \| egress \|both]**<br>**Example:**<br>`Router# debug platform condition interface g0/0/0 ingress` | Specifies the matching criteria for tracing packets. Provides the ability to filter by protocol, IP address and subnet mask, access control list (ACL), interface, and direction. |
| **Step 5** | **debug platform condition start**<br>**Example:**<br>`Router# debug platform condition start` | Enables the specified matching criteria and starts packet tracing. |
| **Step 6** | **debug platform condition stop**<br>**Example:**<br>`Router# debug platform condition start` | Deactivates the condition and stops packet tracing. |
| **Step 7** | **show platform packet-trace {configuration \| statistics \| summary \| packet {all \|** *pkt-num***}}**<br>**Example:**<br>`Router# show platform packet-trace 14` | Displays packet-trace data according to the specified option. See {start cross reference} Table 21-1 {end cross reference} for detailed information about the **show** command options. |
| **Step 8** | **clear platform condition all**<br>**Example:**<br>`Router(config)# clear platform condition all` | Removes the configurations provided by the **debug platform condition** and **debug platform packet-trace** commands. |
| **Step 9** | **exit**<br>**Example:**<br>`Router# exit` | Exits the privileged EXEC mode. |

# Displaying Packet-Trace Information

Use these **show** commands to display packet-trace information.

*Table 27: show Commands*

| Command | Description |
|---|---|
| **show platform packet-trace configuration** | Displays packet trace configuration, including any defaults. |
| **show platform packet-trace statistics** | Displays accounting data for all the traced packets. |

| Command | Description |
|---|---|
| **show platform packet-trace summary** | Displays summary data for the number of packets specified. |
| **show platform packet-trace {all \|** *pkt-num***} [decode]** | Displays the path data for all the packets or the packet specified. The **decode** option attempts to decode the binary packet into a more human- readable form. |

# Removing Packet-Trace Data

Use these commands to clear packet-trace data.

*Table 28: clear Commands*

| Command | Description |
|---|---|
| **clear platform packet-trace statistics** | Clears the collected packet-trace data and statistics. |
| **clear platform packet-trace configuration** | Clears the packet-trace configuration and the statistics. |

# Configuration Examples for Packet Trace

This section provides the following configuration examples:

## Example: Configuring Packet Trace

This example describes how to configure packet trace and display the results. In this example, incoming packets to Gigabit Ethernet interface 0/0/1 are traced, and FIA-trace data is captured for the first 128 packets. Also, the input packets are copied. The **show platform packet-trace packet 0** command displays the summary data and each feature entry visited during packet processing for packet 0.

```
Router>
enable
Router# debug platform packet-trace packet 128 fia-trace
Router# debug platform packet-trace punt
Router# debug platform condition interface g0/0/1 ingress
Router# debug platform condition start
Router#! ping to UUT
Router# debug platform condition stop
Router# show platform packet-trace packet 0
Packet: 0           CBUG ID: 9
Summary
  Input     : GigabitEthernet0/0/1
  Output    : GigabitEthernet0/0/0
  State     : FWD
  Timestamp
    Start   : 1819281992118 ns (05/17/2014 06:42:01.207240 UTC)
    Stop    : 1819282095121 ns (05/17/2014 06:42:01.207343 UTC)
Path Trace
  Feature: IPV4
    Source      : 172.16.10.2
    Destination : 172.16.20.2
    Protocol    : 1 (ICMP)
  Feature: FIA_TRACE
    Entry     : 0x8059dbe8 - DEBUG_COND_INPUT_PKT
    Timestamp : 3685243309297
  Feature: FIA_TRACE
    Entry     : 0x82011a00 - IPV4_INPUT_DST_LOOKUP_CONSUME
    Timestamp : 3685243311450
  Feature: FIA_TRACE
    Entry     : 0x82000170 - IPV4_INPUT_FOR_US_MARTIAN
    Timestamp : 3685243312427
  Feature: FIA_TRACE
    Entry     : 0x82004b68 - IPV4_OUTPUT_LOOKUP_PROCESS
    Timestamp : 3685243313230
  Feature: FIA_TRACE
    Entry     : 0x8034f210 - IPV4_INPUT_IPOPTIONS_PROCESS
    Timestamp : 3685243315033
  Feature: FIA_TRACE
    Entry     : 0x82013200 - IPV4_OUTPUT_GOTO_OUTPUT_FEATURE
    Timestamp : 3685243315787
  Feature: FIA_TRACE
    Entry     : 0x80321450 - IPV4_VFR_REFRAG
    Timestamp : 3685243316980
  Feature: FIA_TRACE
    Entry     : 0x82014700 - IPV6_INPUT_L2_REWRITE
    Timestamp : 3685243317713
  Feature: FIA_TRACE
    Entry     : 0x82000080 - IPV4_OUTPUT_FRAG
    Timestamp : 3685243319223
  Feature: FIA_TRACE
    Entry     : 0x8200e500 - IPV4_OUTPUT_DROP_POLICY
    Timestamp : 3685243319950
  Feature: FIA_TRACE
    Entry     : 0x8059aff4 - PACTRAC_OUTPUT_STATS
    Timestamp : 3685243323603
  Feature: FIA_TRACE
    Entry     : 0x82016100 - MARMOT_SPA_D_TRANSMIT_PKT
    Timestamp : 3685243326183

Router# clear platform condition all
Router# exit
```

Linux Forwarding Transport Service (LFTS) is a transport mechanism to forward packets punted from the CPP into applications other than IOSd. This example displays the LFTS-based intercepted packet destined for binos application.

```
Router# show platform packet-trace packet 10
Packet: 10     CBUG ID: 52
Summary
  Input  : GigabitEthernet0/0/0
  Output : internal0/0/rp:1
  State  : PUNT 55 (For-us control)
  Timestamp
    Start : 597718358383 ns (06/06/2016 09:00:13.643341 UTC)
    Stop : 597718409650 ns (06/06/2016 09:00:13.643392 UTC)
Path Trace
  Feature: IPV4
    Input  : GigabitEthernet0/0/0
    Output : <unknown>
    Source : 10.64.68.2
    Destination : 224.0.0.102
    Protocol : 17 (UDP)
      SrcPort : 1985
      DstPort : 1985
```

```
Feature: FIA_TRACE
  Input  : GigabitEthernet0/0/0
  Output : <unknown>
  Entry  : 0x8a0177bc - DEBUG_COND_INPUT_PKT
  Lapsed time : 426 ns
Feature: FIA_TRACE
  Input  : GigabitEthernet0/0/0
  Output : <unknown>
  Entry  : 0x8a017788 - IPV4_INPUT_DST_LOOKUP_CONSUME
  Lapsed time : 386 ns
Feature: FIA_TRACE
  Input  : GigabitEthernet0/0/0
  Output : <unknown>
  Entry  : 0x8a01778c - IPV4_INPUT_FOR_US_MARTIAN
  Lapsed time : 13653 ns
Feature: FIA_TRACE
  Input  : GigabitEthernet0/0/0
  Output : internal0/0/rp:1
  Entry  : 0x8a017730 - IPV4_INPUT_LOOKUP_PROCESS_EXT
  Lapsed time : 2360 ns
Feature: FIA_TRACE
  Input  : GigabitEthernet0/0/0
  Output : internal0/0/rp:1
  Entry  : 0x8a017be0 - IPV4_INPUT_IPOPTIONS_PROCESS_EXT
  Lapsed time : 66 ns
Feature: FIA_TRACE
  Input  : GigabitEthernet0/0/0
  Output : internal0/0/rp:1
  Entry  : 0x8a017bfc - IPV4_INPUT_GOTO_OUTPUT_FEATURE_EXT
  Lapsed time : 680 ns
Feature: FIA_TRACE
  Input  : GigabitEthernet0/0/0
  Output : internal0/0/rp:1
  Entry  : 0x8a017d60 - IPV4_INTERNAL_ARL_SANITY_EXT
  Lapsed time : 320 ns
Feature: FIA_TRACE
  Input  : GigabitEthernet0/0/0
  Output : internal0/0/rp:1
  Entry  : 0x8a017a40 - IPV4_VFR_REFRAG_EXT
  Lapsed time : 106 ns
Feature: FIA_TRACE
  Input  : GigabitEthernet0/0/0
  Output : internal0/0/rp:1
  Entry  : 0x8a017d2c - IPV4_OUTPUT_DROP_POLICY_EXT
  Lapsed time : 1173 ns
Feature: FIA_TRACE
  Input  : GigabitEthernet0/0/0
  Output : internal0/0/rp:1
  Entry  : 0x8a017940 - INTERNAL_TRANSMIT_PKT_EXT
  Lapsed time : 20173 ns
LFTS Path Flow: Packet: 10    CBUG ID: 52
  Feature: LFTS
  Pkt Direction: IN
  Punt Cause  : 55
       subCause : 0
```

# Example: Using Packet Trace

This example provides a scenario in which packet trace is used to troubleshoot packet drops for a NAT configuration on a Cisco device. This example shows how you can effectively utilize the level of detail provided by the Packet-Trace feature to gather information about an issue, isolate the issue, and then find a solution.

In this scenario, you can detect that there are issues, but are not sure where to start troubleshooting. You should, therefore, consider accessing the Packet-Trace summary for a number of incoming packets.

```
Router# debug platform condition ingress
Router# debug platform packet-trace packet 2048 summary-only
Router# debug platform condition start
Router# debug platform condition stop
Router# show platform packet-trace summary
Pkt   Input             Output            State  Reason
0     Gi0/0/0           Gi0/0/0           DROP   402 (NoStatsUpdate)
1     internal0/0/rp:0  internal0/0/rp:0  PUNT   21  (RP<->QFP keepalive)
2     internal0/0/recycle:0  Gi0/0/0      FWD
```

The output shows that packets are dropped due to NAT configuration on Gigabit Ethernet interface 0/0/0, which enables you to understand that an issue is occurring on a specific interface. Using this information, you can limit which packets to trace, reduce the number of packets for data capture, and increase the level of inspection.

```
Router# debug platform packet-trace packet 256
Router# debug platform packet-trace punt
Router# debug platform condition interface Gi0/0/0
Router# debug platform condition start
Router# debug platform condition stop
Router# show platform packet-trace summary
Router# show platform packet-trace 15
Packet: 15          CBUG ID: 238
Summary
  Input     : GigabitEthernet0/0/0
  Output    : internal0/0/rp:1
  State     : PUNT 55  (For-us control)
  Timestamp
    Start   : 1166288346725 ns (06/06/2016 09:09:42.202734 UTC)
    Stop    : 1166288383210 ns (06/06/2016 09:09:42.202770 UTC)
Path Trace
  Feature: IPV4
    Input       : GigabitEthernet0/0/0
    Output      : <unknown>
    Source      : 10.64.68.3
    Destination : 224.0.0.102
    Protocol    : 17 (UDP)
      SrcPort   : 1985
      DstPort   : 1985
IOSd Path Flow: Packet: 15    CBUG ID: 238
  Feature: INFRA
    Pkt Direction: IN
    Packet Rcvd From CPP
  Feature: IP
    Pkt Direction: IN
    Source      : 10.64.68.122
    Destination : 10.64.68.255
  Feature: IP
    Pkt Direction: IN
    Packet Enqueued in IP layer
    Source      : 10.64.68.122
    Destination : 10.64.68.255
    Interface   : GigabitEthernet0/0/0
  Feature: UDP
    Pkt Direction: IN
    src         : 10.64.68.122(1053)
    dst         : 10.64.68.255(1947)
    length      : 48
```

```
Router#show platform packet-trace packet 10
Packet: 10          CBUG ID: 10
Summary
  Input     : GigabitEthernet0/0/0
  Output    : internal0/0/rp:0
  State     : PUNT 55  (For-us control)
  Timestamp
    Start   : 274777907351 ns (01/10/2020 10:56:47.918494 UTC)
    Stop    : 274777922664 ns (01/10/2020 10:56:47.918509 UTC)
Path Trace
  Feature: IPV4(Input)
    Input       : GigabitEthernet0/0/0
    Output      : <unknown>
    Source      : 10.78.106.2
    Destination : 224.0.0.102
    Protocol    : 17 (UDP)
      SrcPort   : 1985
      DstPort   : 1985

IOSd Path Flow: Packet: 10    CBUG ID: 10
  Feature: INFRA
    Pkt Direction: IN
Packet Rcvd From DATAPLANE
 Feature: IP
    Pkt Direction: IN
    Packet Enqueued in IP layer
    Source      : 10.78.106.2
    Destination : 224.0.0.102
    Interface   : GigabitEthernet0/0/0

  Feature: UDP
    Pkt Direction: IN DROP
    Pkt : DROPPED
    UDP: Discarding silently
    src        : 881 10.78.106.2(1985)
    dst         : 224.0.0.102(1985)
    length      : 60

Router#show platform packet-trace packet  12
Packet: 12          CBUG ID: 767
Summary
  Input     : GigabitEthernet3
  Output    : internal0/0/rp:0
  State     : PUNT 11  (For-us data)
  Timestamp
    Start   : 16120990774814 ns (01/20/2020 12:38:02.816435 UTC)
    Stop    : 16120990801840 ns (01/20/2020 12:38:02.816462 UTC)
Path Trace
  Feature: IPV4(Input)
    Input       : GigabitEthernet3
    Output      : <unknown>
    Source      : 12.1.1.1
    Destination : 12.1.1.2
    Protocol    : 6 (TCP)
      SrcPort   : 46593
      DstPort   : 23
IOSd Path Flow: Packet: 12    CBUG ID: 767
  Feature: INFRA
    Pkt Direction: IN
    Packet Rcvd From DATAPLANE

  Feature: IP
    Pkt Direction: IN
    Packet Enqueued in IP layer
    Source      : 12.1.1.1
    Destination : 12.1.1.2
    Interface   : GigabitEthernet3

  Feature: IP
    Pkt Direction: IN
    FORWARDEDTo transport layer
    Source        : 12.1.1.1
    Destination   : 12.1.1.2
    Interface     : GigabitEthernet3

  Feature: TCP
    Pkt Direction: IN
    tcp0: I NoTCB 12.1.1.1:46593 12.1.1.2:23 seq 1925377975 OPTS 4 SYN  WIN 4128
```

```
Router# show platform packet-trace summary
Pkt   Input                     Output                    State  Reason
0     INJ.2                     Gi1                       FWD
1     Gi1                       internal0/0/rp:0          PUNT   11  (For-us data)
2     INJ.2                     Gi1                       FWD
3     Gi1                       internal0/0/rp:0          PUNT   11  (For-us data)
4     INJ.2                     Gi1                       FWD
5     INJ.2                     Gi1                       FWD
6     Gi1                       internal0/0/rp:0          PUNT   11  (For-us data)
7     Gi1                       internal0/0/rp:0          PUNT   11  (For-us data)
8     Gi1                       internal0/0/rp:0          PUNT   11  (For-us data)
9     Gi1                       internal0/0/rp:0          PUNT   11  (For-us data)
10    INJ.2                     Gi1                       FWD
11    INJ.2                     Gi1                       FWD
12    INJ.2                     Gi1                       FWD
13    Gi1                       internal0/0/rp:0          PUNT   11  (For-us data)
14    Gi1                       internal0/0/rp:0          PUNT   11  (For-us data)
15    Gi1                       internal0/0/rp:0          PUNT   11  (For-us data)
16    INJ.2                     Gi1                       FWD
```

The following example displays the packet trace data statistics.

```
Router#show platform packet-trace statistics
Packets Summary
  Matched  3
  Traced   3
Packets Received
  Ingress  0
  Inject   0
Packets Processed
  Forward  0
  Punt     3
    Count       Code  Cause
    3           56    RP injected for-us control
  Drop     0
  Consume  0

          PKT_DIR_IN
             Dropped       Consumed       Forwarded
INFRA            0             0             0
TCP              0             0             0
UDP              0             0             0
IP               0             0             0
IPV6             0             0             0
ARP              0             0             0

         PKT_DIR_OUT
             Dropped       Consumed       Forwarded
INFRA            0             0             0
TCP              0             0             0
```

```
UDP              0              0              0
IP               0              0              0
IPV6             0              0              0
ARP              0              0              0
```

The following example displays packets that are injected and punted to the forwarding processor from the control plane.

```
Router#debug platform condition ipv4 10.118.74.53/32 both
 Router#Router#debug platform condition start
Router#debug platform packet-trace packet 200
Packet count rounded up from 200 to 256

Router#show platform packet-tracer packet 0
show plat pack pa 0
Packet: 0           CBUG ID: 674
Summary
  Input     : GigabitEthernet1
  Output    : internal0/0/rp:0
  State     : PUNT 11  (For-us data)
  Timestamp
    Start   : 17756544435656 ns (06/29/2020 18:19:17.326313 UTC)
    Stop    : 17756544469451 ns (06/29/2020 18:19:17.326346 UTC)
Path Trace
  Feature: IPV4(Input)
    Input       : GigabitEthernet1
    Output      : <unknown>
    Source      : 10.118.74.53
    Destination : 172.18.124.38
    Protocol    : 17 (UDP)
      SrcPort   : 2640
      DstPort   : 500

IOSd Path Flow: Packet: 0    CBUG ID: 674
  Feature: INFRA
  Pkt Direction: IN
    Packet Rcvd From DATAPLANE

  Feature: IP
  Pkt Direction: IN
    Packet Enqueued in IP layer
    Source      : 10.118.74.53
    Destination : 172.18.124.38
    Interface   : GigabitEthernet1

  Feature: IP
  Pkt Direction: IN
  FORWARDED To transport layer
    Source        : 10.118.74.53
    Destination   : 172.18.124.38
    Interface     : GigabitEthernet1

  Feature: UDP
  Pkt Direction: IN
  DROPPED
 UDP: Checksum error: dropping
 Source      : 10.118.74.53(2640)
 Destination : 172.18.124.38(500)

Router#show platform packet-tracer packet 2
Packet: 2           CBUG ID: 2

IOSd Path Flow:
  Feature: TCP
  Pkt Direction: OUTtcp0: O SYNRCVD 172.18.124.38:22 172.18.124.55:52774 seq 3052140910
OPTS 4 ACK 2346709419 SYN  WIN 4128

  Feature: TCP
  Pkt Direction: OUT
  FORWARDED
 TCP: Connection is in SYNRCVD state
 ACK         : 2346709419
 SEQ         : 3052140910
 Source      : 172.18.124.38(22)
 Destination : 172.18.124.55(52774)


  Feature: IP
  Pkt Direction: OUTRoute out the generated packet.srcaddr: 172.18.124.38, dstaddr:
172.18.124.55

  Feature: IP
  Pkt Direction: OUTInject and forward successful srcaddr: 172.18.124.38, dstaddr:
172.18.124.55

  Feature: TCP
  Pkt Direction: OUTtcp0: O SYNRCVD 172.18.124.38:22 172.18.124.55:52774 seq 3052140910
OPTS 4 ACK 2346709419 SYN  WIN 4128
Summary
  Input     : INJ.2
  Output    : GigabitEthernet1
  State     : FWD
  Timestamp
    Start   : 490928006866 ns (06/29/2020 13:31:30.807879 UTC)
    Stop    : 490928038567 ns (06/29/2020 13:31:30.807911 UTC)
Path Trace
  Feature: IPV4(Input)
    Input       : internal0/0/rp:0
    Output      : <unknown>
    Source      : 172.18.124.38
    Destination : 172.18.124.55
    Protocol    : 6 (TCP)
      SrcPort   : 22
      DstPort   : 52774
  Feature: IPSec
    Result    : IPSEC_RESULT_DENY
    Action    : SEND_CLEAR
    SA Handle : 0
    Peer Addr : 55.124.18.172
    Local Addr: 38.124.18.172


Router#
```

# Additional References

### Standards

| Standard | Title |
|---|---|
| None | — |

### MIBs

| MIB | MIBs Link |
|---|---|
| None | To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at this URL:<br>{start hypertext}http://www.cisco.com/go/mibs{end hypertext} |

### RFCs

| RFC | Title |
|---|---|
| None | — |

### Technical Assistance

| Description | Link |
|---|---|
| The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password. | {start hypertext}http://www.cisco.com/cisco/web/support/index.html{end hypertext} |

# Feature Information for Packet Trace

{start cross reference}Table 21-4{end cross reference} lists the features in this module and provides links to specific configuration information.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to{start hypertext} http://www.cisco.com/go/cfn{end hypertext}. An account on Cisco.com is not required.

**Note** {start cross reference}Table 21-4{end cross reference} lists only the software releases that support a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

*Table 29: Feature Information for Packet Trace*

| Feature Name | Releases | Feature Information |
|---|---|---|
| Packet Trace | Cisco IOS XE 3.10S | The Packet Trace feature provides information about how data packets are processed by the Cisco IOS XE software.<br>In Cisco IOS XE Release 3.10S, this feature was introduced.<br>The following commands were introduced or modified:<br>• **debug platform packet-trace packet** *pkt-num* [**fia-trace** \| **summary-only**] [**data-size** *data-size*] [**circular**]<br>• **debug platform packet-trace copy packet** {**input** \| **output** \| **both**} [**size** *num-bytes*] [**L2** \| **L3** \| **L4**]<br>• **show platform packet-trace** {**configuration** \| **statistics** \| **summary** \| **packet** {**all** \| *pkt-num*}} |
| | Cisco IOS XE 3.11S | In Cisco IOS XE Release 3.11S, this feature was enhanced to include the following features:<br>• Matched versus traced statistics.<br>• Trace stop timestamp in addition to trace start timestamp.<br>The following commands were introduced or modified:<br>• **debug platform packet-trace drop** [**code** *drop-num*]<br>• **show platform packet-trace packet** {**all** \| *pkt-num*} [*decode*] |
| | Cisco IOS XE Denali 16.3.1 | In Cisco IOS XE Denali 16.3.1, this feature was enhanced to include Layer3 packet tracing along with IOSd.<br>The following commands were introduced or modified: **debug platform packet-trace punt**. |
| | Cisco IOS XE Amsterdam 17.3.1 | The output of the **show platform packet-trace** command now includes additional trace information for packets either originated from IOSd or destined to IOSd or other BinOS processes. |

CHAPTER 21

# Configuring VRF Route Sharing

The following chapter describes how you can configure VRF Route Sharing on a Cisco Catalyst 8000V instance. VRF Route Sharing is required when you need to forward traffic between an On-Premise Site and a Public Cloud Site. Configure VRF Route Sharing across VxLAN peers to deploy shared services across the cloud.

## Information About VRF Route Sharing

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 the import/export route target is associated with the BGP EVPN session with VXLAN encapsulation and L3 routing information in the On-Premise router. The other set of import/export route-target 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 forwaring chain.

Using the VRF Route Sharing functionality, you can deploy shared services across hybrid clouds. The shared services that run on the public cloud can be consumed by the endpoints on the On-Premise Site. The Cisco Catalyst 8000V instance shares the L3 prefix to multiple VRFs on the On-Premise Site, and vice versa. The APIC layer imports the addresses and the services are thus consumed in the APIC side.

### Platforms Supported

The VRF Route Sharing functionality is currently supported on Cisco Catalyst 8000V and Cisco CSR1000V Series.

# Prerequisites of 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 are 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&2 from ACI into VRF3&4.
- The IP BGP on the Cisco Catalyst 8000V side redistributes the routes to the gateway in the public cloud.
- The next-hop of routes from ACI are the spine of the border leaf of the ACI.
- There are no overlaps of prefix across the Route Sharing VRF.
- Advertise the L3 VPN routing and to forward the VRF prefixes to the EVPN neighbours. Run the advertise l2vpn evpn command and export stitching RTs to push the native routes towards the EVPN.

# Restrictions for VRF Route Sharing

- The VRF Sharing functionality supports up to 32 common VRFs, and 1000 customer VRF combination.
- This functionality does not support RT filters.
- VRF Route Sharing is supported only for IPv4 addresses and not IPv6 addresses.

# How to Configure VRF Route Sharing

## Sample Topology and Use Cases

Consider a sample topology to explain the VRF Route Sharing in a hybrid cloud. In a 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 are the Virtual Port Channel (vPC) pair for ACI. These two vPCs are configured with different Route Distinguishers (RD). Here, VRF 1 and VRF 2 are configured on the vPC pair for ACI. For example,

VRF1 - RT:RT-EVPN-1, prefix:1.1.1.1

VRF2 - RT:RT-EVPN-2, prefix:2.2.2.2

VRF3 and VRF4 are configured on the Cisco Catalyst 8000V instance. These two VRFs pair with the Voice Gateway (VGW), and these two VRFs have two different Route Targets (RT). For example,

VRF3 – RT for EVPN: RT-EVPN-3, RT for IP BGP: RT-3, prefix:3.3.3.3

VRF4 – RT for EVPN: RT-EVPN-4, RT for IP BGP: RT-4, prefix:4.4.4.4

In the topology, the BGP-EVPN fabric is present between the ACI and the Cisco Catalyst 8000V instance in the public cloud and the IP BGP protocol is used between the Cisco Catalyst 8000V instance and the Cloud Service Provider such as 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, perform the following configuration:

```
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, perform the following configuration:

```
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, perform the following configuration:

```
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, perform the following configuration:

```
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
```

**Note** For the above-mentioned use case, the Cisco Catalyst 8000V instance must configure EVPN on both VRF3 and VRF4.

Even IP BGP already imports all the routes from VRF3 and VRF4, BGP does not advertise the imported routes of the VRF to the EVPN peer.

You need to use the **Stitching** keyword in the configuration only when the sharing happens across the EVPN.

# Configuring VRF Route Sharing

Perform the following configuration to configure VRF Route Sharing in a hybrid cloud where VRF 1 and VRF 2 (On-Premise) can talk to to VRF 3 and VRF 4 (in the public cloud). In this sample solution, VRF3 and VRF4 cannot talk to each other.

**Example:**

```
vrf definition vrf3
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
!
!
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 4.4.4.1 255.255.255.224
bridge-domain 100
member vni 10100
!
interface nve1
source-interface loopback0
host-reachability protocol bgp
member vni 10100 vrf vrf3
!
router bgp 100
bgp router-id 11.11.11.11
no bgp default ipv4-unicast
neighbor 22.22.22.22 remote-as 200
neighbor 22.22.22.22 update-source loopback0
neighbor 22.22.22.22 ebgp-multihop 255
address-family ipv4 vrf vrf3
redistribute connected
neighbor 4.4.4.2 remote-as 300
neighbor 4.4.4.2 activate
neighbor 4.4.4.2 send-community both
advertise l2vpn evpn
exit-address-family
!
address-family l2vpn evpn
neighbor 22.22.22.22 activate
neighbor 22.22.22.22 send-community both
exit-address-family
end
```

# Verifying VRF Route Sharing

**Step 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 11.11.11.11, 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
22.22.22.22     4          200       6       5        4    0    0 00:01:23        4
Device#
```

**Step 2** **show ip route vrf vrf3 bgp | in binding**.

Displays the IP routing table information associated with the VRF. When you see the output with the binding label, it indicates that 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
Device#
```

CHAPTER 22

# Radio Aware Routing

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.

The RAR feature is supported on Cisco ISR G2 and G3 Series Routers, Cisco ISR 4000 Series Routers.

PPPoE Extensions is the RAR protocol supported in Cisco 4000 Series ISRs. PPPoE Extensions with Aggregate support is introduce from Cisco IOS XE Fuji 16.7. release. OSPFv3 and EIGRP are the supported routing protocols.

## Benefits of Radio Aware Routing

The Radio Aware Routing feature offers the following benefits:

- Provides faster network convergence through immediate recognition of changes.
- Enables routing for failing or fading radio links.
- Allows easy routing between line-of-sight and non-line-of-sight paths.
- Provides faster convergence and optimal route selection so that delay-sensitive traffic, such as voice and video, is not disrupted
- 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 simple Ethernet connection to RFC 5578, R2CP, and DLEP compliant radios.

# Restrictions and Limitations

The Radio Aware Routing feature has the following restrictions and limitations:

- The DLEP and R2CP protocols are not supported in Cisco 4000 Series ISRs.
- Multicast traffic is not supported in aggregate mode.
- Cisco High Availability (HA) technology is not supported.

# License Requirements

This feature is available with the AX license.

# Performance

The Radio Aware Routing feature has the ability to support a maximum of 10 neighbors per radio or VMI interface; and a total of 30 to 40 neighbors.

# System Components

The Radio Aware Routing (RAR) feature is implemented using the MANET (Mobile adhoc network) infrastructure comprising of different components such as PPPoE, Virtual multipoint interface (VMI), QoS, routing protocol interface and RAR protocols.

## Point-to-Point Protocol over Ethernet PPPoE or 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)

Though PPPoE Extensions provides the most of the setup to communicate between a router and a radio, VMI addresses the need to manage and translate events that higher layers (example, routing protocols) consume. In addition, VMI 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 Aggregae 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 replicats 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 following 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 92.92.2.1 255.255.255.0
 no peer default ip address
 no keepalive
 service-policy input rar_policer
end
```

# Example: Configuring the RAR Feature in Bypass Mode

The following example is an end-to-end configuration of RAR in the bypass mode:

**Note** Before you being the RAR configuration, you must first configure the **subscriber authorization enable** command to bring up the RAR session. Without enbaling authorization, the Point-to-Point protocol does not recognize this as a RAR session and may not tag *manet_radio* in presentation of a PPPoE Active Discovery Initiate (PADI). By default, bypass mode does not appears in the configuration. It appears only if the mode is configured as bypass.

**Configure a Service for RAR**

```
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 RAR

```
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 90.90.90.3 255.255.255.0
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 92.92.2.1 255.255.255.0
```

```
physical-interface GigabitEthernet0/0/0
mode bypass
```

```
interface vmi3//configure the virtual multi interface
 ip address 93.93.2.1 255.255.255.0
 physical-interface GigabitEthernet0/0/1
mode bypass
```

### Configure OSPF Routing

```
router ospfv3 1
 router-id 1.1.1.1
!
 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 92.92.2.3 92.92.2.254
```

# Verifying RAR Session Details

To retrieve RAR session details, use the following show commands:

```
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=92.92.2.2, 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=91.91.91.4, 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=92.92.2.2, 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 3.3.3.3)

Neighbor ID     Pri   State           Dead Time   Interface ID    Interface
1.1.1.1           0   FULL/  -        00:01:32    19              Virtual-Access2.1

          OSPFv3 1 address-family ipv6 (router-id 3.3.3.3)

Neighbor ID     Pri   State           Dead Time   Interface ID    Interface
1.1.1.1           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

      90.0.0.0/8 is variably subnetted, 3 subnets, 2 masks
C        90.90.90.0/24 is directly connected, Virtual-Access2.1
O        90.90.90.4/32 [110/1] via 90.90.90.4, 00:00:03, Virtual-Access2.1
L        90.90.90.5/32 is directly connected, Virtual-Access2.1
```

```
      92.0.0.0/32 is subnetted, 1 subnets
C        92.92.2.21 is directly connected, Virtual-Access2.1
```