Cisco Catalyst 8000V Edge Software Installation And Configuration Guide

First Published: 2020-12-21

Last Modified: 2024-08-28

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

CONTENTS

CHAPTER 15

CHAPTER 16

CONTENTS

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

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

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

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

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

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

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

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

CHAPTER 1

Preface

This preface describes the audience, organization, and conventions of this document. It also provides information on how to obtain other documentation.

This preface includes the following sections:

Audience and Scope

This document is designed for the person who is responsible for configuring your Cisco Enterprise router. This document is intended primarily for the following audiences:

- Customers with technical networking background and experience.
- System administrators familiar with the fundamentals of router-based internetworking but who might not be familiar with Cisco IOS software.
- System administrators who are responsible for installing and configuring internetworking equipment, and who are familiar with Cisco IOS software.

Feature Compatibility

For more information about the Cisco IOS XE software, including features available on your device as described in the configuration guides, see the respective router documentation set.

To verify support for specific features, use the Cisco Feature Navigator tool. This tool enables you to determine the Cisco IOS XE software images that support a specific software release, feature set, or a platform.

Document Conventions

This documentation uses the following conventions:

Convention	Description
^ or **Ctrl**	The ^ and **Ctrl** symbols represent the Control key. For example, the key combination **^D** or **Ctrl-D** means hold down the **Control** key while you press the **D** key. Keys are indicated in capital letters but are not case sensitive.
string	A string is a nonquoted set of characters shown in italics. For example, when setting an SNMP community string to public, do not use quotation marks around the string or the string will include the quotation marks.

The command syntax descriptions use the following conventions:

Convention	Description
bold	Bold text indicates commands and keywords that you enter exactly as shown.
italics	Italic text indicates arguments for which you supply values.
[x]	Square brackets enclose an optional element (keyword or argument).
|	A vertical line indicates a choice within an optional or required set of keywords or arguments.
[x | y]	Square brackets enclosing keywords or arguments separated by a vertical line indicate an optional choice.
{x | y}	Braces enclosing keywords or arguments separated by a vertical line indicate a required choice.

Nested sets of square brackets or braces indicate optional or required choices within optional or required elements. For example, see the following table.

Convention	Description
[x {y | z}]	Braces and a vertical line within square brackets indicate a required choice within an optional element.

Examples use the following conventions:

Convention	Description
`screen`	Examples of information displayed on the screen are set in Courier font.
`bold screen`	Examples of text that you must enter are set in Courier bold font.
< >	Angle brackets enclose text that is not printed to the screen, such as passwords.
!	An exclamation point at the beginning of a line indicates a comment line. Exclamation points are also displayed by the Cisco IOS XE software for certain processes.
[ ]	Square brackets enclose default responses to system prompts.

Caution Means *reader be careful*. In this situation, you might do something that could result in equipment damage or loss of data.

Note Means *reader take note*. Notes contain helpful suggestions or references to materials that may not be contained in this manual.

Communications, Services, and Additional Information

- To receive timely, relevant information from Cisco, sign up at Cisco Profile Manager.
- To get the business impact you're looking for with the technologies that matter, visit Cisco Services.
- To submit a service request, visit Cisco Support.
- To discover and browse secure, validated enterprise-class apps, products, solutions and services, visit Cisco DevNet.
- To obtain general networking, training, and certification titles, visit Cisco Press.
- To find warranty information for a specific product or product family, access Cisco Warranty Finder.

Cisco Bug Search Tool

Cisco Bug Search Tool (BST) is a web-based tool that acts as a gateway to the Cisco bug tracking system that maintains a comprehensive list of defects and vulnerabilities in Cisco products and software. BST provides you with detailed defect information about your products and software.

Documentation Feedback

To provide feedback about Cisco technical documentation, use the feedback form available in the right pane of every online document.

Troubleshooting

For the most up-to-date, detailed troubleshooting information, see the Cisco TAC website at https://www.cisco.com/en/US/support/index.html.

Go to **Products by Category** and choose your product from the list, or enter the name of your product. Look under **Troubleshoot and Alerts** to find information for the issue that you are experiencing.

CHAPTER 2

Overview of Cisco Catalyst 8000V

The Cisco Catalyst 8000V Edge Software is a virtual, form-factor router deployed on a virtual machine (VM) running on an x86 server hardware. This guide covers the overview, installation, upgrade, and configuration of Cisco Catalyst 8000V.

Cisco Catalyst 8000V supports both Cisco IOS XE and the Cisco IOS XE SD-WAN functionalities through the autonomous mode and the controller mode, respectively. Cisco Catalyst 8000V in the autonomous mode supports a subset of the Cisco IOS XE software features and technologies, and provides Cisco IOS XE security and switching features on a virtualization platform. The controller mode delivers comprehensive SD-WAN, WAN gateway, and network services functions in the virtual and cloud environments.

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.

Cisco Catalyst 8000V supports SSL VPN. From Cisco IOS XE Release 17.x, when you are running a Cisco IOS-XE router as an SSL VPN gateway, an extra SSL VPN overhead is added due to the TLS encapsulation. To prevent IP fragmentation and reassembly of packets between SSL VPN client and server, you must adjust the TCP-MSS value optimally. Otherwise, packet drop due to the IPFragErr error could occur in the SSL VPN gateway.

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 and on the Cisco Cloud Services Platform 5000 Series.

To use the functionalities of this virtual router, read on to know how to deploy a Cisco Catalyst 8000V router as a virtual machine on a hypervisor.

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.
- **Enhanced software security - Secure Object Store**: In Cisco Catalyst 8000V, storage partitions for NVRAM, licensing, and other data are created as Object stores. The individual Object stores are encrypted to ensure data security, and this product is Cisco Secure Development lifecycle (CSDL) compliant. Further, Cisco Catalyst 8000V supports a 16G disk profile.

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

Interface Numbering

- The interface port numbering is from 1 and up to the number of interfaces supported. See VMware Requirements, on page 46 to know the supported vNICs and the minimum and maximum number of vNICs supported for each VM instance.
- Gigabit Ethernet interface 0 is not supported.
- You can designate any interface as the management interface. You can designate a management interface by performing the appropriate Day0 bootstrapping mechanisms available for your target environment. For more details, see Day 0 Configuration, on page 93.

Cisco IOS XE and Cisco Catalyst 8000V

Cisco Catalyst 8000V is a virtual router that runs on Cisco IOS XE and Cisco IOS XE SD-WAN. This guide provides the overview, installation, and configuration information for Cisco Catalyst 8000V on Cisco IOS XE.

You can configure and manage Cisco Catalyst 8000V by:

- Provisioning a serial port in the VM to connect and 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. See your hypervisor documentation for more details.

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

Note By default, Telnet is disabled for security reasons. SSH is disabled in an on-prem deployment. Although SSH is preferred for remote user management, you must manually enable SSH in an on-prem deployment.

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

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

Cisco Unified Computing System (UCS) Products

Table 1: Cisco Catalyst 8000V Compatibility with Cisco UCS Servers

Cisco Unified Computing System (UCS) Products	The Cisco UCS server requirements are: • VMware-certified. • 4 or more cores configured. • A minimum UCS memory of 16 GB. If you use the SDWAN/Controller mode, at least 128 GB memory is required to accomodate SDWAN vManage, vBond, and vSmart. • A minimum UCS storage of 1 TB. • A UCS C220 M5 minimum is recommended. 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.

CHAPTER 3

Installation Overview

This chapter provides the high-level information on how to install Cisco Catalyst 8000V. Usually, Cisco hardware routers are shipped with the Cisco IOS XE software pre-installed. However, since Cisco Catalyst 8000V is not a hardware-based router, you must download the Cisco IOS XE software from Cisco.com and install the virtual router directly onto the virtual machine. Before you proceed to the installation, first provision the attributes of the VM so that the Cisco Catalyst 8000V software can install and boot.

See the following sections to know about the various installation files and the installation options that are dependent on the hypervisor you have chosen.

Installation Files

The following table specifies the software images that are available for installing Cisco Catalyst 8000V on the supported hypervisors:

Image Type	Hypervisor	Mode	Secure Boot	Sample Filename
bin	ESXi, KVM, AWS, Microsoft Azure, GCP	Upgrade (bundle mode) Upgrade (install mode)	No	c8000v-universalk9.17.04.01a.SPA.bin
iso - Used for installing the software image on the VM	ESXi, KVM	New installation	No	c8000v-universalk9.17.04.01a.iso

Image Type	Hypervisor	Mode	Secure Boot	Sample Filename
ova - used for deploying the OVA template on the VM (in TAR format)	ESXi	New installation	Yes	c8000v-universalk9.17.04.01a.ova
qcow2 - Used for installing the software image in KVM environments.	KVM	New installation	No	c8000v-universalk9.17.04.01a.qcow2
serial.qcow2	KVM	New installation	No	c8000v-universalk9.17.04.01a.efi.qcow2
efi.qcow2	KVM	New installation	Yes	c8000v-universalk9.17.04.01a.efi.qcow2
serial.efi.qcow2	KVM	New installation	Yes	c8000v-universalk9.17.04.01a-serial.efi.qcow2
tar.gz	NFVIS	New installation	Yes	c8000v-universalk9.17.04.01a-tar.gz

Note Although secure boot is supported for certain image types, this functionality is not enabled by default. See VNF Secure Boot to know how to enable secure boot for your hypervisor.

Supported Hypervisors

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

The following are the supported hypervisors for Cisco Catalyst 8000V:

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

Virtual Machine Processing Resources

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

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

For detailed information on virtual machine processing resources, see the respective hypervisor tuning guides provided by the hypervisor. Additionally, you can refer to the appropriate hypervisor sections in this guide that describe the possible settings to increase the performance and improve the overall system determinism.

Download the Installation Files

Step 1 Go to the Cisco Software Download page.

Step 2 From the **Select a Product** field at the bottom of the page, search for Cisco Catalyst 8000V.

Step 3 Click the Cisco Catalyst 8000V link and go to the Download page.

Step 4 From the left pane, select the appropriate release. For example, *Bengaluru 17.4.1*.

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

Note To know which installation file you want to download, see Installation Files, on page 9.

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 within a nested VM has not been tested and is not recommended for this reason.
- 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.
- Gigabit Ethernet0 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. Select the console from the GRUB mode during the first-time installation or change the console using the Cisco IOS XE **platform console** command after the router boots. For more information, see Booting the Cisco Catalyst 8000V as the VM, on page 127.
- If you are running a virtual function on an I350 device, redundancy protocols like HSRP/VRRP are not supported.
- For .qcow2 files, the image that you choose during installation plays a role in the type of console you can select.
- vNICs do not support duplex settings in an interface.
- vNICs do not support auto-negotiations.

- From Cisco IOS XE 17.9.1, the **show license udi** command is no longer supported in Cisco Catalyst 8000V.
- Cisco Catalyst 8000V does not support L2TP functionality including L2TP client and L2TP Network Server (LNS).
- Although Cisco Catalyst 8000V instances in controller mode supports the 4GB memory configuration, Cisco recommends you use 8GB memory for feature-rich and high-scale environments. This is to prevent high memory utilization issues.

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

Where to Go Next

Now that you have downloaded the installation file, you can proceed to the deployment. Based on the hypervisor that you have chosen, the deployment procedures vary.

See the following chapters in this guide to know how to deploy Cisco Catalyst 8000V in the appropriate hypervisor environment:

- Installing in VMware ESXi Environment
- Installing in Kernel Virtual Machine Support (KVM) Environments

Deployment in Public Clouds

- For information about deploying Cisco Catalyst 8000V in an Amazon Web Services environment, see Deploying Cisco Catalyst 8000V Edge Software on Amazon Web Services.
- For information about deploying Cisco Catalyst 8000V in the Microsoft Azure environment, see Deploying Cisco Catalyst 8000V on Microsoft Azure.
- For information about deploying Cisco Catalyst 8000V in Google Cloud Platform, see Deploying Cisco Catalyst 8000V on Google Cloud Platform.
- For more information about deploying Cisco Catalyst 8000V in Alibaba Cloud, see Deploying Cisco Catalyst 8000V on Alibaba Cloud.

Note Refer the following chapters before you proceed with the installation:

- Day 0 Configuration
- VNF Secure Boot
- Configuring Console Access

CHAPTER 4

Installing the Software using install Commands

Installing the Software Using install Commands

From Cisco IOS XE 17.15.1a, all Cisco IOS XE platforms are shipped in install mode by default. Users can boot the platform, and upgrade or downgrade to Cisco IOS XE software versions using a set of **install** commands.

Restrictions for Installing the Software Using install Commands

- ISSU is not covered in this feature.
- Install mode requires a reboot of the system.

Information About Installing the Software Using install Commands

For routers shipped in install mode, a set of **install** commands can be used for starting, upgrading and downgrading of platforms in install mode. This update is applicable to the Cisco Catalyst 8000 Edge platforms.

From Cisco IOS XE 17.15.1a release, this update is applicable to all Cisco IOS XE platforms.

The following table describes the differences between Bundle mode and Install mode:

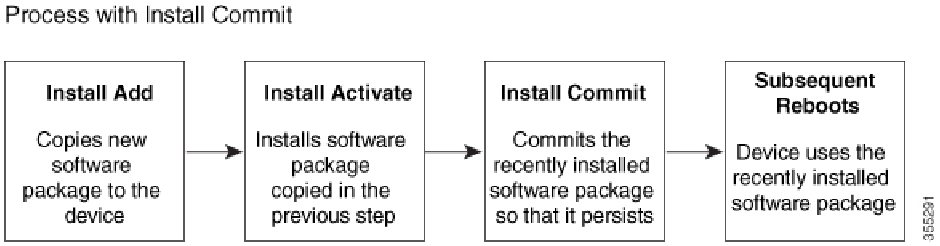
Table 2: Bundle Mode vs Install Mode

Bundle Mode	Install Mode
This mode provides a consolidated boot process, using local (hard disk, flash) or remote (TFTP) .bin image. **Note** Bundle boot from USB and TFTPBoot is not supported.	This mode uses the local (bootflash) packages.conf file for the boot process.
This mode uses a single .bin file.	.bin file is replaced with expanded .pkg files in this mode.
CLI: `#boot system file <filename>`	CLI: `#install add file bootflash: [activate commit]`
To upgrade in this mode, point the boot system to the new image.	To upgrade in this mode, use the **install** commands.
Image Auto-Upgrade: When a new Field-Replaceable Unit (FRU) is inserted in a modular chassis, manual intervention is required to get the new FRU running with the same version as the active FRUs.	Image Auto-Upgrade: When a new FRU is inserted in a modular chassis, the joining FRU is auto-upgraded to the image version in sync with the active FRUs.
Rollback: Rollback to the previous image with multiple Software Maintenance Updates (SMUs) may require multiple reloads.	Rollback: Enables rollback to an earlier version of Cisco IOS XE software, including multiple patches in single reload.

Install Mode Process Flow

The install mode process flow comprises three commands to perform installation and upgrade of software on platforms–**install add**, **install activate**, and **install commit**.

The following flow chart explains the install process with **install** commands:


Process with Install Commit
Install Add
Copies new software package to the device
Install Activate
Installs software package copied in the previous step
Install Commit
Commits the recently installed software package so that it persists
Subsequent Reboots
Device uses the recently installed software package
355291

The **install add** command copies the software package from a local or remote location to the platform. The location can be FTP, HTTP, HTTPs, or TFTP. The command extracts individual components of the .package file into subpackages and packages.conf files. It also validates the file to ensure that the image file is specific to the platform on which it is being installed.

The **install activate** command performs the required validations and provisions the packages previously added using the **install add** command. It also triggers a system reload.

The **install commit** command confirms the packages previously activated using the **install activate** command, and makes the updates persistent over reloads.

Note Installing an update replaces any previously installed software image. At any time, only one image can be installed in a device.

The following set of install commands is available:

Table 3: List of install Commands

Command	Syntax	Purpose
install add	**install add file** *location:filename.bin*	Copies the contents of the image, package, and SMUs to the software repository. File location may be local or remote. This command does the following: • Validates the file–checksum, platform compatibility checks, and so on. • Extracts individual components of the package into subpackages and packages.conf • Copies the image into the local inventory and makes it available for the next steps.
install activate	**install activate**	Activates the package added using the **install add** command. • Use the **show install summary** command to see which image is inactive. This image will get activated. • System reloads on executing this command. Confirm if you want to proceed with the activation. Use this command with the **prompt-level none** keyword to automatically ignore any confirmation prompts.

Command	Syntax	Purpose
(install activate) auto abort-timer	**install activate auto-abort timer** *<30-1200>*	The **auto-abort timer** starts automatically, with a default value of 120 minutes. If the **install commit** command is not executed within the time provided, the activation process is terminated, and the system returns to the last-committed state. • You can change the time value while executing the **install activate** command. • The **install commit** command stops the timer, and continues the installation process. • The **install activate auto-abort timer stop** command stops the timer without committing the package. • Use this command with the **prompt-level none** keyword to automatically ignore any confirmation prompts. • This command is valid only in the three-step install variant.
install commit	**install commit**	Commits the package activated using the **install activate** command, and makes it persistent over reloads. • Use the **show install summary** command to see which image is uncommitted. This image will get committed.

Command	Syntax	Purpose
install abort	**install abort**	Terminates the installation and returns the system to the last-committed state. • This command is applicable only when the package is in activated status (uncommitted state). • If you have already committed the image using the **install commit** command, use the **install rollback to** command to return to the preferred version.
install remove	**install remove {file** *<filename>* **| inactive}**	Deletes inactive packages from the platform repository. Use this command to free up space. • **file**: Removes specified files. • **inactive**: Removes all the inactive files.
install rollback to	**install rollback to {base | label | committed | id}**	Rolls back the software set to a saved installation point or to the last-committed installation point. The following are the characteristics of this command: • Requires reload. • Is applicable only when the package is in committed state. • Use this command with the **prompt-level none** keyword to automatically ignore any confirmation prompts. **Note** If you are performing install rollback to a previous image, the previous image must be installed in install mode. Only SMU rollback is possible in bundle mode.

Command	Syntax	Purpose
install deactivate	**install deactivate file** <*filename*>	Removes a package from the platform repository. This command is supported only for SMUs. • Use this command with the **prompt-level none** keyword to automatically ignore any confirmation prompts.

The following show commands are also available:

Table 4: List of show Commands

Command	Syntax	Purpose
show install log	**show install log**	Provides the history and details of all install operations that have been performed since the platform was booted.
show install package	**show install package** <*filename*>	Provides details about the .pkg/.bin file that is specified.
show install summary	**show install summary**	Provides an overview of the image versions and their corresponding install states for all the FRUs. • The table that is displayed will state for which FRUs this information is applicable. • If all the FRUs are in sync in terms of the images present and their state, only one table is displayed. • If, however, there is a difference in the image or state information among the FRUs, each FRU that differs from the rest of the stack is listed in a separate table.
show install active	**show install active**	Provides information about the active packages for all the FRUs. If there is a difference in the information among the FRUs, each FRU that differs from the rest of the stack is listed in a separate table.

Command	Syntax	Purpose
show install inactive	**show install inactive**	Provides information about the inactive packages, if any, for all the FRUs. If there is a difference in the information among the FRUs, each FRU that differs from the rest of the stack is listed in a separate table.
show install committed	**show install committed**	Provides information about the committed packages for all the FRUs. If there is a difference in the information among the FRUs, each FRU that differs from the rest of the stack is listed in a separate table.
show install uncommitted	**show install uncommitted**	Provides information about uncommitted packages, if any, for all the FRUs. If there is a difference in the information among the FRUs, each FRU that differs from the rest of the stack is listed in a separate table.
show install rollback	**show install rollback {point-id | label}**	Displays the package associated with a saved installation point.
show version	**show version [rp-slot] [installed [user-interface] | provisioned | running]**	Displays information about the current package, along with hardware and platform information.

Booting the Platform in Install Mode

You can install, activate, and commit a software package using a single command (one-step install) or multiple separate commands (three-step install).

If the platform is working in bundle mode, the one-step install procedure must be used to initially convert the platform from bundle mode to install mode. Subsequent installs and upgrades on the platform can be done with either one-step or three-step variants.

One-Step Installation or Converting from Bundle Mode to Install Mode

Note

- All the CLI actions (for example, add, activate, and so on) are executed on all the available FRUs.
- The configuration save prompt will appear if an unsaved configuration is detected.
- The reload prompt will appear after the second step in this workflow. Use the **prompt-level none** keyword to automatically ignore the confirmation prompts.
- If the prompt-level is set to None, and there is an unsaved configuration, the install fails. You must save the configuration before reissuing the command.

Use the one-step install procedure described below to convert a platform running in bundle boot mode to install mode. After the command is executed, the platform reboots in install boot mode.

Later, the one-step install procedure can also be used to upgrade the platform.

This procedure uses the **install add file activate commit** command in privileged EXEC mode to install a software package, and to upgrade the platform to a new version.

SUMMARY STEPS

1. **enable**
2. **install add file location:** *filename* [**activate commit**]
3. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device>enable`	Enables privileged EXEC mode. Enter your password, if prompted.

Step 2	**install add file location:** *filename* [**activate commit**]
	Example:
	`Device#install add file`
	`bootflash:c8000be-universalk9.BLD_V177_THROTTLE_LATEST_20211021_031123_V17_7_0_117.SSA.bin`
	`activate commit`
		Copies the software install package from a local or remote location (through FTP, HTTP, HTTPs, or TFTP) to the platform and extracts the individual components of the .package file into subpackages and packages.conf files. It also performs a validation and compatibility check for the platform and image versions, activates the package, and commits the package to make it persistent across reloads.
		The platform reloads after this command is run.

Step 3	**exit** **Example:** `Device#exit`	Exits privileged EXEC mode and returns to user EXEC mode.

Three-Step Installation

> **Note**
> - All the CLI actions (for example, add, activate, and so on) are executed on all the available FRUs.
> - The configuration save prompt will appear if an unsaved configuration is detected.
> - The reload prompt will appear after the install activate step in this workflow. Use the **prompt-level none** keyword to automatically ignore the confirmation prompts.

The three-step installation procedure can be used only after the platform is in install mode. This option provides more flexibility and control to the customer during installation.

This procedure uses individual **install add**, **install activate**, and **install commit** commands for installing a software package, and to upgrade the platform to a new version.

SUMMARY STEPS

1. **enable**
2. **install add file location:** *filename*
3. **show install summary**
4. **install activate** [**auto-abort-timer** <*time*>]
5. **install abort**
6. **install commit**
7. **install rollback to committed**
8. **install remove** {**file** *filesystem: filename* | **inactive**}
9. **show install summary**
10. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device>enable`	Enables privileged EXEC mode. Enter your password, if prompted.

Step 2	**install add file location:** *filename*
	Example:
	`Device#install add file`
	`bootflash:c8000be-universalk9.BLD_V177_THROTTLE_LATEST_20211027_030841_V17_7_0_120.SSA.bin`
		Copies the software install package from a remote location (through FTP, HTTP, HTTPs, or TFTP) to the platform, and extracts the individual components of the .package file into subpackages and packages.conf files.

Step 3	**show install summary** **Example:** `Device#show install summary`	(Optional) Provides an overview of the image versions and their corresponding install state for all the FRUs.

	Command or Action	Purpose
Step 4	**install activate** [**auto-abort-timer** <*time*>] **Example:** `Device# install activate auto-abort-timer 120`	Activates the previously added package and reloads the platform. • When doing a full software install, do not provide a package filename. • In the three-step variant, **auto-abort-timer** starts automatically with the **install activate** command; the default for the timer is 120 minutes. If the **install commit** command is not run before the timer expires, the install process is automatically terminated. The platform reloads and boots up with the last committed version.
Step 5	**install abort** **Example:** `Device#install abort`	(Optional) Terminates the software install activation and returns the platform to the last committed version. • Use this command only when the image is in activated state, and not when the image is in committed state.
Step 6	**install commit** **Example:** `Device#install commit`	Commits the new package installation and makes the changes persistent over reloads.
Step 7	**install rollback to committed** **Example:** `Device#install rollback to committed`	(Optional) Rolls back the platform to the last committed state.

Step 8	**install remove** {**file** *filesystem: filename* | **inactive**}
	Example:
	`Device#install remove inactive`
		(Optional) Deletes software installation files.
		• **file**: Deletes a specific file
		• **inactive**: Deletes all the unused and inactive installation files.

Step 9	**show install summary** **Example:** `Device#show install summary`	(Optional) Displays information about the current state of the system. The output of this command varies according to the **install** commands run prior to this command.
Step 10	**exit** **Example:** `Device#exit`	Exits privileged EXEC mode and returns to user EXEC mode.

Upgrading in Install Mode

Use either the one-step installation or the three-step installation to upgrade the platform in install mode.

Downgrading in Install Mode

Use the **install rollback** command to downgrade the platform to a previous version by pointing it to the appropriate image, provided the image you are downgrading to was installed in install mode.

The **install rollback** command reloads the platform and boots it with the previous image.

Note The **install rollback** command succeeds only if you have not removed the previous file using the **install remove inactive** command.

Alternatively, you can downgrade by installing the older image using the **install** commands.

Terminating a Software Installation

You can terminate the activation of a software package in the following ways:

- When the platform reloads after activating a new image, the auto-abort-timer is triggered (in the three-step install variant). If the timer expires before issuing the **install commit** command, the installation process is terminated, and the platform reloads and boots with the last committed version of the software image.

  Alternatively, use the **install auto-abort-timer stop** command to stop this timer, without using the **install commit** command. The new image remains uncommitted in this process.

- Using the **install abort** command returns the platform to the version that was running before installing the new software. Use this command before issuing the **install commit** command.

Configuration Examples for Installing the Software Using install Commands

The following is an example of the one-step installation or converting from bundle mode to install mode:

```
Router# install add file
bootflash:c8000be-universalk9.BLD_V177_THROTTLE_LATEST_20211021_031123_V17_7_0_117.SSA.bin
 activate commit
install_add_activate_commit: START Thu Oct 28 21:57:21 UTC 2021

System configuration has been modified.
Press Yes(y) to save the configuration and proceed.
Press No(n) for proceeding without saving the configuration.
Press Quit(q) to exit, you may save configuration and re-enter the command. [y/n/q]y
Building configuration...

[OK]Modified configuration has been saved

*Oct 28 21:57:39.818: %SYS-6-PRIVCFG_ENCRYPT_SUCCESS: Successfully encrypted private config
 file
*Oct 28 21:57:39.925: %INSTALL-5-INSTALL_START_INFO: R0/0: install_engine: Started install
 one-shot
bootflash:c8000be-universalk9.BLD_V177_THROTTLE_LATEST_20211021_031123_V17_7_0_117.SSA.bininstall_add_activate_commit:
 Adding PACKAGE
```

```
install_add_activate_commit: Checking whether new add is allowed ....

--- Starting Add ---
Performing Add on Active/Standby
  [1] Add package(s) on R0
  [1] Finished Add on R0
Checking status of Add on [R0]
Add: Passed on [R0]
Finished Add

Image added. Version: 17.07.01.0.1515
install_add_activate_commit: Activating PACKAGE
Following packages shall be activated:
/bootflash/c8000be-rpboot.BLD_V177_THROTTLE_LATEST_20211021_031123_V17_7_0_117.SSA.pkg
/bootflash/c8000be-mono-universalk9.BLD_V177_THROTTLE_LATEST_20211021_031123_V17_7_0_117.SSA.pkg
/bootflash/c8000be-firmware_sm_nim_adpt.BLD_V177_THROTTLE_LATEST_20211021_031123_V17_7_0_117.SSA.pkg
/bootflash/c8000be-firmware_sm_dsp_sp2700.BLD_V177_THROTTLE_LATEST_20211021_031123_V17_7_0_117.SSA.pkg
/bootflash/c8000be-firmware_sm_async.BLD_V177_THROTTLE_LATEST_20211021_031123_V17_7_0_117.SSA.pkg
/bootflash/c8000be-firmware_sm_1t3e3.BLD_V177_THROTTLE_LATEST_20211021_031123_V17_7_0_117.SSA.pkg
/bootflash/c8000be-firmware_sm_10g.BLD_V177_THROTTLE_LATEST_20211021_031123_V17_7_0_117.SSA.pkg
/bootflash/c8000be-firmware_prince.BLD_V177_THROTTLE_LATEST_20211021_031123_V17_7_0_117.SSA.pkg
/bootflash/c8000be-firmware_nim_xdsl.BLD_V177_THROTTLE_LATEST_20211021_031123_V17_7_0_117.SSA.pkg
/bootflash/c8000be-firmware_nim_ssd.BLD_V177_THROTTLE_LATEST_20211021_031123_V17_7_0_117.SSA.pkg
/bootflash/c8000be-firmware_nim_shdsl.BLD_V177_THROTTLE_LATEST_20211021_031123_V17_7_0_117.SSA.pkg
/bootflash/c8000be-firmware_nim_ge.BLD_V177_THROTTLE_LATEST_20211021_031123_V17_7_0_117.SSA.pkg
/bootflash/c8000be-firmware_nim_cwan.BLD_V177_THROTTLE_LATEST_20211021_031123_V17_7_0_117.SSA.pkg
/bootflash/c8000be-firmware_nim_bri_st_fw.BLD_V177_THROTTLE_LATEST_20211021_031123_V17_7_0_117.SSA.pkg
/bootflash/c8000be-firmware_nim_async.BLD_V177_THROTTLE_LATEST_20211021_031123_V17_7_0_117.SSA.pkg
/bootflash/c8000be-firmware_ngwic_t1e1.BLD_V177_THROTTLE_LATEST_20211021_031123_V17_7_0_117.SSA.pkg
/bootflash/c8000be-firmware_dsp_tilegx.BLD_V177_THROTTLE_LATEST_20211021_031123_V17_7_0_117.SSA.pkg
/bootflash/c8000be-firmware_dsp_sp2700.BLD_V177_THROTTLE_LATEST_20211021_031123_V17_7_0_117.SSA.pkg
/bootflash/c8000be-firmware_dsp_analogbri.BLD_V177_THROTTLE_LATEST_20211021_031123_V17_7_0_117.SSA.pkg
/bootflash/c8000be-firmware_dreamliner.BLD_V177_THROTTLE_LATEST_20211021_031123_V17_7_0_117.SSA.pkg

This operation may require a reload of the system. Do you want to proceed? [y/n]y
--- Starting Activate ---
Performing Activate on Active/Standby

*Oct 28 22:05:49.484: %INSTALL-5-INSTALL_AUTO_ABORT_TIMER_PROGRESS: R0/0: rollback_timer:
Install auto abort timer will expire in 7200 seconds  [1] Activate package(s) on R0
  [1] Finished Activate on R0
Checking status of Activate on [R0]
Activate: Passed on [R0]
Finished Activate

--- Starting Commit ---
Performing Commit on Active/Standby
  [1] Commit package(s) on R0

Building configuration...
  [1] Finished Commit on R0
Checking status of Commit on [R0]
Commit: Passed on [R0]
Finished Commit

[OK]
*Oct 28 22:06:55.375: %SYS-6-PRIVCFG_ENCRYPT_SUCCESS: Successfully encrypted private config
 fileSend model notification for install_add_activate_commit before reload
Install will reload the system now!
SUCCESS: install_add_activate_commit  Thu Oct 28 22:07:22 UTC 2021

Router#
*Oct 28 22:07:22.661: %INSTALL-5-INSTALL_COMPLETED_INFO: R0/0: install_engine: Completed
install one-shot PACKAGE
```

```
bootflash:c8000be-universalk9.BLD_V177_THROTTLE_LATEST_20211021_031123_V17_7_0_117.SSA.binOct
 28 22:07:26.864: %PMAN-5-EXITACTION: R0/0: pvp: Process manager is exiting: reload action
 requested

□

Press RETURN to get started!
```

The following is an example of the three-step installation:

```
Router# install add file
bootflash:c8000be-universalk9.BLD_V177_THROTTLE_LATEST_20211027_030841_V17_7_0_120.SSA.bin

install_add: START Thu Oct 28 22:36:43 UTC 2021

*Oct 28 22:36:44.526: %INSTALL-5-INSTALL_START_INFO: R0/0: install_engine: Started install
 add
bootflash:c8000be-universalk9.BLD_V177_THROTTLE_LATEST_20211027_030841_V17_7_0_120.SSA.bininstall_add:
 Adding PACKAGE
install_add: Checking whether new add is allowed ....

--- Starting Add ---
Performing Add on Active/Standby
  [1] Add package(s) on R0
  [1] Finished Add on R0
Checking status of Add on [R0]
Add: Passed on [R0]
Finished Add

Image added. Version: 17.07.01.0.1601
SUCCESS: install_add  Thu Oct 28 22:40:25 UTC 2021

Router#
*Oct 28 22:40:25.971: %INSTALL-5-INSTALL_COMPLETED_INFO: R0/0: install_engine: Completed
install add PACKAGE
bootflash:c8000be-universalk9.BLD_V177_THROTTLE_LATEST_20211027_030841_V17_7_0_120.SSA.bin

Router# show install log
[0|install_op_boot]: START Thu Oct 28 22:09:29 Universal 2021
[0|install_op_boot(INFO, )]: Mount IMG INI state base image
[0|install_op_boot]: END SUCCESS  Thu Oct 28 22:09:30 Universal 2021
[0|install_op_boot(INFO, )]: cleanup_trap  remote_invocation 0 operation install_op_boot
.. 0 .. 0
[1|display_install_log]: START Thu Oct 28 22:12:11 UTC 2021
[2|install_add]: START Thu Oct 28 22:36:43 UTC 2021
[2|install_add(INFO, )]: Set INSTALL_TYPE to PACKAGE
[2|install_add(CONSOLE, )]: Adding PACKAGE
[2|install_add(CONSOLE, )]: Checking whether new add is allowed ....
[2|install_add(INFO, )]: check_add_op_allowed: Install type PACKAGE
[remote|install_add]: START Thu Oct 28 22:37:12 UTC 2021
[remote|install_add]: END SUCCESS  Thu Oct 28 22:40:10 UTC 2021
[remote|install_add(INFO, )]: cleanup_trap  remote_invocation 1 operation install_add .. 0
 .. 0
[2|install_add(INFO, )]: Remote output from R0
[2|install_add(INFO, )]: install_add: START Thu Oct 28 22:37:12 UTC 2021
Expanding image file:
bootflash:c8000be-universalk9.BLD_V177_THROTTLE_LATEST_20211027_030841_V17_7_0_120.SSA.bin
Verifying parameters
Expanding superpackage
bootflash:c8000be-universalk9.BLD_V177_THROTTLE_LATEST_20211027_030841_V17_7_0_120.SSA.bin
... parameters verified
Validating package type
... package type validated
```

```
Copying package files
   c8000be-firmware_dreamliner.BLD_V177_THROTTLE_LATEST_20211027_030841_V17_7_0_120.SSA.pkg


c8000be-firmware_dsp_analogbri.BLD_V177_THROTTLE_LATEST_20211027_030841_V17_7_0_120.SSA.pkg

   c8000be-firmware_dsp_sp2700.BLD_V177_THROTTLE_LATEST_20211027_030841_V17_7_0_120.SSA.pkg

   c8000be-firmware_dsp_tilegx.BLD_V177_THROTTLE_LATEST_20211027_030841_V17_7_0_120.SSA.pkg

   c8000be-firmware_ngwic_t1e1.BLD_V177_THROTTLE_LATEST_20211027_030841_V17_7_0_120.SSA.pkg

   c8000be-firmware_nim_async.BLD_V177_THROTTLE_LATEST_20211027_030841_V17_7_0_120.SSA.pkg


c8000be-firmware_nim_bri_st_fw.BLD_V177_THROTTLE_LATEST_20211027_030841_V17_7_0_120.SSA.pkg

    c8000be-firmware_nim_cwan.BLD_V177_THROTTLE_LATEST_20211027_030841_V17_7_0_120.SSA.pkg
    c8000be-firmware_nim_ge.BLD_V177_THROTTLE_LATEST_20211027_030841_V17_7_0_120.SSA.pkg
   c8000be-firmware_nim_shdsl.BLD_V177_THROTTLE_LATEST_20211027_030841_V17_7_0_120.SSA.pkg

    c8000be-firmware_nim_ssd.BLD_V177_THROTTLE_LATEST_20211027_030841_V17_7_0_120.SSA.pkg
    c8000be-firmware_nim_xdsl.BLD_V177_THROTTLE_LATEST_20211027_030841_V17_7_0_120.SSA.pkg
    c8000be-firmware_prince.BLD_V177_THROTTLE_LATEST_20211027_030841_V17_7_0_120.SSA.pkg
    c8000be-firmware_sm_10g.BLD_V177_THROTTLE_LATEST_20211027_030841_V17_7_0_120.SSA.pkg
    c8000be-firmware_sm_1t3e3.BLD_V177_THROTTLE_LATEST_20211027_030841_V17_7_0_120.SSA.pkg
    c8000be-firmware_sm_async.BLD_V177_THROTTLE_LATEST_20211027_030841_V17_7_0_120.SSA.pkg

c8000be-firmware_sm_dsp_sp2700.BLD_V177_THROTTLE_LATEST_20211027_030841_V17_7_0_120.SSA.pkg

   c8000be-firmware_sm_nim_adpt.BLD_V177_THROTTLE_LATEST_20211027_030841_V17_7_0_120.SSA.pkg

    c8000be-mono-universalk9.BLD_V177_THROTTLE_LATEST_20211027_030841_V17_7_0_120.SSA.pkg
    c8000be-rpboot.BLD_V177_THROTTLE_LATEST_20211027_030841_V17_7_0_120.SSA.pkg
  WARNING: A different version of provisioning file packages.conf already exists in bootflash:

  WARNING: The provisioning file from the expanded bundle will be saved as
  WARNING: bootflash:c8000be-universalk9.BLD_V177_THROTTLE_LATEST_20211027_0.conf
... package files copied
SUCCESS: Finished expanding all-in-one software package.
Image file expanded
SUCCESS: install_add  Thu Oct 28 22:40:10 UTC 2021
[2|install_add]: END SUCCESS  Thu Oct 28 22:40:25 UTC 2021
[2|install_add(INFO, )]: cleanup_trap  remote_invocation 0 operation install_add .. 0 .. 0
[3|COMP_CHECK]: START Thu Oct 28 22:40:26 UTC 2021
[3|COMP_CHECK]: END FAILED exit(1)  Thu Oct 28 22:40:27 UTC 2021
[3|COMP_CHECK(INFO, )]: cleanup_trap  remote_invocation 0 operation COMP_CHECK .. 1 .. 1
[4|install_activate]: START Thu Oct 28 22:42:53 UTC 2021
[4|install_activate(INFO, require user prompt)]: install_cli
[4|install_activate(CONSOLE, )]: Activating PACKAGE
[4|install_activate(INFO, )]: Acquiring transaction lock...
[4|install_activate(INFO, )]: global_trans_lock:
/bootflash/.installer/install_global_trans_lock
[4|install_activate(INFO, )]: tmp_global_trans_lock: /tmp/tmp_install_global_trans_lock
[4|install_activate(INFO, )]: tmp lock does not exist: /tmp/tmp_install_global_trans_lock
[4|install_activate(INFO, )]: global_trans_lock:
/bootflash/.installer/install_global_trans_lock
[4|install_activate(INFO, )]: tmp_global_trans_lock: /tmp/tmp_install_global_trans_lock
[4|install_activate(INFO, )]: local_trans_lock: /bootflash/.installer/install_local_trans_lock
[4|install_activate(INFO, )]: global_trans_lock:
/bootflash/.installer/install_global_trans_lock
[4|install_activate(INFO, )]: validate_lock: lock_duration is 7200
[4|install_activate(INFO, )]: install type stored in lock PACKAGE, install type PACKAGE,
install operation install_activate
```

```
[4|install_activate(INFO, )]: lock duration: 7200
[4|install_activate(INFO, )]: extend trans lock done.
/bootflash/.installer/install_global_trans_lock
[4|install_activate(INFO, require user prompt)]: install_cli
[4|install_activate( FATAL)]: Cannot proceed activate because of user input
[4|install_activate(INFO, )]: cleanup_trap  remote_invocation 0 operation install_activate
 .. 6 .. 0
[5|install_add]: START Thu Oct 28 22:45:48 UTC 2021
[5|install_add(INFO, )]: Set INSTALL_TYPE to PACKAGE
[5|install_add(CONSOLE, )]: Adding PACKAGE
[5|install_add(CONSOLE, )]: Checking whether new add is allowed ....
[5|install_add(INFO, )]: check_add_op_allowed: Install type PACKAGE
[5|install_add( FATAL)]: Super package already added. Add operation not allowed. install
remove inactive can be used to discard added packages

Router# install activate
install_activate: START Thu Oct 28 23:57:57 UTC 2021
install_activate: Activating PACKAGE

*Oct 28 23:57:57.823: %INSTALL-5-INSTALL_START_INFO: R0/0: install_engine: Started install
 activateFollowing packages shall be activated:
/bootflash/c8000be-rpboot.BLD_V177_THROTTLE_LATEST_20211027_030841_V17_7_0_120.SSA.pkg
/bootflash/c8000be-mono-universalk9.BLD_V177_THROTTLE_LATEST_20211027_030841_V17_7_0_120.SSA.pkg
/bootflash/c8000be-firmware_sm_nim_adpt.BLD_V177_THROTTLE_LATEST_20211027_030841_V17_7_0_120.SSA.pkg
/bootflash/c8000be-firmware_sm_dsp_sp2700.BLD_V177_THROTTLE_LATEST_20211027_030841_V17_7_0_120.SSA.pkg
/bootflash/c8000be-firmware_sm_async.BLD_V177_THROTTLE_LATEST_20211027_030841_V17_7_0_120.SSA.pkg
/bootflash/c8000be-firmware_sm_1t3e3.BLD_V177_THROTTLE_LATEST_20211027_030841_V17_7_0_120.SSA.pkg
/bootflash/c8000be-firmware_sm_10g.BLD_V177_THROTTLE_LATEST_20211027_030841_V17_7_0_120.SSA.pkg
/bootflash/c8000be-firmware_prince.BLD_V177_THROTTLE_LATEST_20211027_030841_V17_7_0_120.SSA.pkg
/bootflash/c8000be-firmware_nim_xdsl.BLD_V177_THROTTLE_LATEST_20211027_030841_V17_7_0_120.SSA.pkg
/bootflash/c8000be-firmware_nim_ssd.BLD_V177_THROTTLE_LATEST_20211027_030841_V17_7_0_120.SSA.pkg
/bootflash/c8000be-firmware_nim_shdsl.BLD_V177_THROTTLE_LATEST_20211027_030841_V17_7_0_120.SSA.pkg
/bootflash/c8000be-firmware_nim_ge.BLD_V177_THROTTLE_LATEST_20211027_030841_V17_7_0_120.SSA.pkg
/bootflash/c8000be-firmware_nim_cwan.BLD_V177_THROTTLE_LATEST_20211027_030841_V17_7_0_120.SSA.pkg
/bootflash/c8000be-firmware_nim_bri_st_fw.BLD_V177_THROTTLE_LATEST_20211027_030841_V17_7_0_120.SSA.pkg
/bootflash/c8000be-firmware_nim_async.BLD_V177_THROTTLE_LATEST_20211027_030841_V17_7_0_120.SSA.pkg
/bootflash/c8000be-firmware_ngwic_t1e1.BLD_V177_THROTTLE_LATEST_20211027_030841_V17_7_0_120.SSA.pkg
/bootflash/c8000be-firmware_dsp_tilegx.BLD_V177_THROTTLE_LATEST_20211027_030841_V17_7_0_120.SSA.pkg
/bootflash/c8000be-firmware_dsp_sp2700.BLD_V177_THROTTLE_LATEST_20211027_030841_V17_7_0_120.SSA.pkg
/bootflash/c8000be-firmware_dsp_analogbri.BLD_V177_THROTTLE_LATEST_20211027_030841_V17_7_0_120.SSA.pkg
/bootflash/c8000be-firmware_dreamliner.BLD_V177_THROTTLE_LATEST_20211027_030841_V17_7_0_120.SSA.pkg

This operation may require a reload of the system. Do you want to proceed? [y/n]y
--- Starting Activate ---
Performing Activate on Active/Standby

*Oct 29 00:04:19.400: %INSTALL-5-INSTALL_AUTO_ABORT_TIMER_PROGRESS: R0/0: rollback_timer:
Install auto abort timer will expire in 7200 seconds  [1] Activate package(s) on R0
    --- Starting list of software package changes ---
    Old files list:
      Modified
c8000be-firmware_dreamliner.BLD_V177_THROTTLE_LATEST_20211021_031123_V17_7_0_117.SSA.pkg
      Modified
c8000be-firmware_dsp_analogbri.BLD_V177_THROTTLE_LATEST_20211021_031123_V17_7_0_117.SSA.pkg

      Modified
c8000be-firmware_dsp_sp2700.BLD_V177_THROTTLE_LATEST_20211021_031123_V17_7_0_117.SSA.pkg
      Modified
c8000be-firmware_dsp_tilegx.BLD_V177_THROTTLE_LATEST_20211021_031123_V17_7_0_117.SSA.pkg
      Modified
c8000be-firmware_ngwic_t1e1.BLD_V177_THROTTLE_LATEST_20211021_031123_V17_7_0_117.SSA.pkg
      Modified
c8000be-firmware_nim_async.BLD_V177_THROTTLE_LATEST_20211021_031123_V17_7_0_117.SSA.pkg
      Modified
```

```
c8000be-firmware_nim_bri_st_fw.BLD_V177_THROTTLE_LATEST_20211021_031123_V17_7_0_117.SSA.pkg

      Modified
c8000be-firmware_nim_cwan.BLD_V177_THROTTLE_LATEST_20211021_031123_V17_7_0_117.SSA.pkg
      Modified
c8000be-firmware_nim_ge.BLD_V177_THROTTLE_LATEST_20211021_031123_V17_7_0_117.SSA.pkg
      Modified
c8000be-firmware_nim_shdsl.BLD_V177_THROTTLE_LATEST_20211021_031123_V17_7_0_117.SSA.pkg
      Modified
c8000be-firmware_nim_ssd.BLD_V177_THROTTLE_LATEST_20211021_031123_V17_7_0_117.SSA.pkg
      Modified
c8000be-firmware_nim_xdsl.BLD_V177_THROTTLE_LATEST_20211021_031123_V17_7_0_117.SSA.pkg
      Modified
c8000be-firmware_prince.BLD_V177_THROTTLE_LATEST_20211021_031123_V17_7_0_117.SSA.pkg
      Modified
c8000be-firmware_sm_10g.BLD_V177_THROTTLE_LATEST_20211021_031123_V17_7_0_117.SSA.pkg
      Modified
c8000be-firmware_sm_1t3e3.BLD_V177_THROTTLE_LATEST_20211021_031123_V17_7_0_117.SSA.pkg
      Modified
c8000be-firmware_sm_async.BLD_V177_THROTTLE_LATEST_20211021_031123_V17_7_0_117.SSA.pkg
      Modified
c8000be-firmware_sm_dsp_sp2700.BLD_V177_THROTTLE_LATEST_20211021_031123_V17_7_0_117.SSA.pkg

      Modified
c8000be-firmware_sm_nim_adpt.BLD_V177_THROTTLE_LATEST_20211021_031123_V17_7_0_117.SSA.pkg
      Modified
c8000be-mono-universalk9.BLD_V177_THROTTLE_LATEST_20211021_031123_V17_7_0_117.SSA.pkg
      Modified c8000be-rpboot.BLD_V177_THROTTLE_LATEST_20211021_031123_V17_7_0_117.SSA.pkg
    New files list:
      Added
c8000be-firmware_dreamliner.BLD_V177_THROTTLE_LATEST_20211027_030841_V17_7_0_120.SSA.pkg
      Added
c8000be-firmware_dsp_analogbri.BLD_V177_THROTTLE_LATEST_20211027_030841_V17_7_0_120.SSA.pkg

      Added
c8000be-firmware_dsp_sp2700.BLD_V177_THROTTLE_LATEST_20211027_030841_V17_7_0_120.SSA.pkg
      Added
c8000be-firmware_dsp_tilegx.BLD_V177_THROTTLE_LATEST_20211027_030841_V17_7_0_120.SSA.pkg
      Added
c8000be-firmware_ngwic_t1e1.BLD_V177_THROTTLE_LATEST_20211027_030841_V17_7_0_120.SSA.pkg
      Added
c8000be-firmware_nim_async.BLD_V177_THROTTLE_LATEST_20211027_030841_V17_7_0_120.SSA.pkg
      Added
c8000be-firmware_nim_bri_st_fw.BLD_V177_THROTTLE_LATEST_20211027_030841_V17_7_0_120.SSA.pkg

      Added
c8000be-firmware_nim_cwan.BLD_V177_THROTTLE_LATEST_20211027_030841_V17_7_0_120.SSA.pkg
      Added
c8000be-firmware_nim_ge.BLD_V177_THROTTLE_LATEST_20211027_030841_V17_7_0_120.SSA.pkg
      Added
c8000be-firmware_nim_shdsl.BLD_V177_THROTTLE_LATEST_20211027_030841_V17_7_0_120.SSA.pkg
      Added
c8000be-firmware_nim_ssd.BLD_V177_THROTTLE_LATEST_20211027_030841_V17_7_0_120.SSA.pkg
      Added
c8000be-firmware_nim_xdsl.BLD_V177_THROTTLE_LATEST_20211027_030841_V17_7_0_120.SSA.pkg
      Added
c8000be-firmware_prince.BLD_V177_THROTTLE_LATEST_20211027_030841_V17_7_0_120.SSA.pkg
      Added
c8000be-firmware_sm_10g.BLD_V177_THROTTLE_LATEST_20211027_030841_V17_7_0_120.SSA.pkg
      Added
c8000be-firmware_sm_1t3e3.BLD_V177_THROTTLE_LATEST_20211027_030841_V17_7_0_120.SSA.pkg
      Added
c8000be-firmware_sm_async.BLD_V177_THROTTLE_LATEST_20211027_030841_V17_7_0_120.SSA.pkg
      Added
```

```
c8000be-firmware_sm_dsp_sp2700.BLD_V177_THROTTLE_LATEST_20211027_030841_V17_7_0_120.SSA.pkg

      Added
c8000be-firmware_sm_nim_adpt.BLD_V177_THROTTLE_LATEST_20211027_030841_V17_7_0_120.SSA.pkg
      Added
c8000be-mono-universalk9.BLD_V177_THROTTLE_LATEST_20211027_030841_V17_7_0_120.SSA.pkg
      Added c8000be-rpboot.BLD_V177_THROTTLE_LATEST_20211027_030841_V17_7_0_120.SSA.pkg
    Finished list of software package changes
  [1] Finished Activate on R0
Checking status of Activate on [R0]
Activate: Passed on [R0]
Finished Activate

Send model notification for install_activate before reload
Install will reload the system now!
SUCCESS: install_activate  Fri Oct 29 00:05:09 UTC 2021

Router#
*Oct 29 00:05:09.504: %INSTALL-5-INSTALL_COMPLETED_INFO: R0/0: install_engine: Completed
install activate PACKAGEOct 29 00:05:14.494: %PMAN-5-EXITACTION: R0/0: pvp: Process manager
 is exiting: reload action requested

Initializing Hardware ...

Checking for PCIe device presence...done
System integrity status: 0x610

System Bootstrap, Version 17.3(4.1r), RELEASE SOFTWARE
Copyright (c) 1994-2021 by cisco Systems, Inc.

Current image running   : Boot ROM1
Last reset cause        : LocalSoft
C8300-2N2S-6T platform with 8388608 Kbytes of main memory

□

Press RETURN to get started!

□

Router# install commit
install_commit: START Fri Oct 29 00:13:58 UTC 2021
install_commit: Committing PACKAGE

--- Starting Commit ---
Performing Commit on Active/Standby

*Oct 29 00:13:59.552: %INSTALL-5-INSTALL_START_INFO: R0/0: install_engine: Started install
 commit  [1] Commit package(s) on R0
  [1] Finished Commit on R0
Checking status of Commit on [R0]
Commit: Passed on [R0]
Finished Commit

SUCCESS: install_commit  Fri Oct 29 00:14:03 UTC 2021

Router#
*Oct 29 00:14:03.712: %INSTALL-5-INSTALL_COMPLETED_INFO: R0/0: install_engine: Completed
install commit PACKAGE
```

The following is an example of downgrading in install mode:

```
ROUTER# install activate file bootflash:c8000be-universalk9.17.06.01a.SPA.bin activate
commit

install_add_activate_commit: START Fri Dec 10 18:07:17 GMT 2021

*Dec 10 18:07:18.405 GMT: %INSTALL-5-INSTALL_START_INFO: R0/0: install_engine: Started
install one-shot bootflash:c8000be-universalk9.17.06.01a.SPA.bininstall_add_activate_commit:
 Adding PACKAGE
install_add_activate_commit: Checking whether new add is allowed ....

--- Starting Add ---
Performing Add on Active/Standby
  [1] Add package(s) on R0
  [1] Finished Add on R0
Checking status of Add on [R0]
Add: Passed on [R0]
Finished Add

Image added. Version: 17.06.01a.0.298
install_add_activate_commit: Activating PACKAGE
Following packages shall be activated:
/bootflash/c8000be-rpboot.17.06.01a.SPA.pkg
/bootflash/c8000be-mono-universalk9.17.06.01a.SPA.pkg
/bootflash/c8000be-firmware_sm_nim_adpt.17.06.01a.SPA.pkg
/bootflash/c8000be-firmware_sm_dsp_sp2700.17.06.01a.SPA.pkg
/bootflash/c8000be-firmware_sm_async.17.06.01a.SPA.pkg
/bootflash/c8000be-firmware_sm_1t3e3.17.06.01a.SPA.pkg
/bootflash/c8000be-firmware_sm_10g.17.06.01a.SPA.pkg
/bootflash/c8000be-firmware_prince.17.06.01a.SPA.pkg
/bootflash/c8000be-firmware_nim_xdsl.17.06.01a.SPA.pkg
/bootflash/c8000be-firmware_nim_ssd.17.06.01a.SPA.pkg
/bootflash/c8000be-firmware_nim_shdsl.17.06.01a.SPA.pkg
/bootflash/c8000be-firmware_nim_ge.17.06.01a.SPA.pkg
/bootflash/c8000be-firmware_nim_cwan.17.06.01a.SPA.pkg
/bootflash/c8000be-firmware_nim_bri_st_fw.17.06.01a.SPA.pkg
/bootflash/c8000be-firmware_nim_async.17.06.01a.SPA.pkg
/bootflash/c8000be-firmware_ngwic_t1e1.17.06.01a.SPA.pkg
/bootflash/c8000be-firmware_dsp_tilegx.17.06.01a.SPA.pkg
/bootflash/c8000be-firmware_dsp_sp2700.17.06.01a.SPA.pkg
/bootflash/c8000be-firmware_dsp_analogbri.17.06.01a.SPA.pkg
/bootflash/c8000be-firmware_dreamliner.17.06.01a.SPA.pkg

This operation may require a reload of the system. Do you want to proceed? [y/n]y
--- Starting Activate ---
Performing Activate on Active/Standby
  [1] Activate package(s) on R0
  [1] Finished Activate on R0
Checking status of Activate on [R0]
Activate: Passed on [R0]
Finished Activate

--- Starting Commit ---
Performing Commit on Active/Standby
  [1] Commit package(s) on R0
Building configuration...

  [1] Finished Commit on R0
Checking status of Commit on [R0]
Commit: Passed on [R0]
Finished Commit

[OK]
*Dec 10 18:14:57.782 GMT: %SYS-6-PRIVCFG_ENCRYPT_SUCCESS: Successfully encrypted private
config fileSend model notification for install_add_activate_commit before reload
```

```
/usr/binos/conf/install_util.sh: line 164: /bootflash/.prst_sync/reload_info: No such file
 or directory
/usr/binos/conf/install_util.sh: line 168: /bootflash/.prst_sync/reload_info: No such file
 or directory
cat: /bootflash/.prst_sync/reload_info: No such file or directory
Install will reload the system now!
SUCCESS: install_add_activate_commit  Fri Dec 10 18:15:23 GMT 2021

ROUTER#
*Dec 10 18:15:23.955 GMT: %INSTALL-5-INSTALL_COMPLETED_INFO: R0/0: install_engine: Completed
 install one-shot PACKAGE bootflash:c8000be-universalk9.17.06.01a.SPA.binDec 10 18:15:27.708:
 %PMAN-5-EXITACTION: R0/0: pvp: Process manager is exiting: reload action requested

Initializing Hardware ...

Checking for PCIe device presence...done
System integrity status: 0x610
Rom image verified correctly

System Bootstrap, Version 17.3(5r), RELEASE SOFTWARE
Copyright (c) 1994-2021 by cisco Systems, Inc.

Current image running: Boot ROM0

Last reset cause: LocalSoft
ROUTER platform with 8388608 Kbytes of main memory

□

Press RETURN to get started!

□

ROUTER#
ROUTER# show version
Cisco IOS XE Software, Version 17.06.01a
Cisco IOS Software [Bengaluru], c8000be Software (X86_64_LINUX_IOSD-UNIVERSALK9-M), Version
 17.6.1a, RELEASE SOFTWARE (fc2)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2021 by Cisco Systems, Inc.
Compiled Sat 21-Aug-21 03:27 by mcpre

Cisco IOS-XE software, Copyright (c) 2005-2021 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: 17.3(5r)

ROUTER uptime is 0 minutes
Uptime for this control processor is 2 minutes
System returned to ROM by LocalSoft
System image file is "bootflash:packages.conf"
Last reload reason: LocalSoft

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.

Technology Package License Information:

-----------------------------------------------------------------
Technology     Type         Technology-package Technology-package
                            Current            Next Reboot
-----------------------------------------------------------------
Smart License  Perpetual    None               None
Smart License  Subscription None               None

The current crypto throughput level is 250000 kbps

Smart Licensing Status: Registration Not Applicable/Not Applicable

cisco ROUTER (1RU) processor with 3747220K/6147K bytes of memory.
Processor board ID FDO2521M27S
Router operating mode: Autonomous
5 Gigabit Ethernet interfaces
2 2.5 Gigabit Ethernet interfaces
2 Cellular interfaces
32768K bytes of non-volatile configuration memory.
8388608K bytes of physical memory.
7573503K bytes of flash memory at bootflash:.
1875361792K bytes of NVMe SSD at harddisk:.
16789568K bytes of USB flash at usb0:.

Configuration register is 0x2102
```

The following is an example of terminating a software installation:

```
Router# install abort
install_abort: START Fri Oct 29 02:42:51 UTC 2021

This install abort would require a reload. Do you want to proceed? [y/n]
                                                                     *Oct 29 02:42:52.789:
 %INSTALL-5-INSTALL_START_INFO: R0/0: install_engine: Started install aborty
--- Starting Abort ---
Performing Abort on Active/Standby

  [1] Abort package(s) on R0
  [1] Finished Abort on R0
Checking status of Abort on [R0]
Abort: Passed on [R0]
Finished Abort

Send model notification for install_abort before reload
Install will reload the system now!
SUCCESS: install_abort  Fri Oct 29 02:44:47 UTC 2021

Router#
*Oct 29 02:44:47.866: %INSTALL-5-INSTALL_COMPLETED_INFO: R0/0: install_engine: Completed
install abort PACKAGEOct 29 02:44:51.577: %PMAN-5-EXITACTION: R0/0: pvp: Process manager
```

```
is exiting: reload action requested

Initializing Hardware ...

Checking for PCIe device presence...done
System integrity status: 0x610

System Bootstrap, Version 17.3(4.1r), RELEASE SOFTWARE
Copyright (c) 1994-2021 by cisco Systems, Inc.

Current image running   : Boot ROM1
Last reset cause        : LocalSoft
C8300-2N2S-6T platform with 8388608 Kbytes of main memory

□

Press RETURN to get started!

□
```

The following are sample outputs for show commands:

show install log

```
Device# show install log
[0|install_op_boot]: START Thu Oct 28 22:09:29 Universal 2021
[0|install_op_boot(INFO, )]: Mount IMG INI state base image
[0|install_op_boot]: END SUCCESS  Thu Oct 28 22:09:30 Universal 2021
```

show install summary

```
Device# show install summary
[ R0 ] Installed Package(s) Information:
State (St): I - Inactive, U - Activated & Uncommitted,
            C - Activated & Committed, D - Deactivated & Uncommitted
--------------------------------------------------------------------
Type  St   Filename/Version
--------------------------------------------------------------------
IMG   C    17.07.01.0.1515

--------------------------------------------------------------------
Auto abort timer: inactive
--------------------------------------------------------------------
```

show install package *filesystem: filename*

```
Device# show install package
bootflash:c8000be-universalk9.BLD_V177_THROTTLE_LATEST_20211021_031123_V17_7_0_117.SSA.bin
Package: c8000be-universalk9.BLD_V177_THROTTLE_LATEST_20211021_031123_V17_7_0_117.SSA.bin
  Size: 831447859
  Timestamp: 2021-10-23 17:08:14 UTC
  Canonical path:
/bootflash/c8000be-universalk9.BLD_V177_THROTTLE_LATEST_20211021_031123_V17_7_0_117.SSA.bin


  Raw disk-file SHA1sum:
    5c4e7617a6c71ffbcc73dcd034ab58bf76605e3f
  Header size:     1192 bytes
  Package type:    30000
  Package flags:   0
  Header version:  3

  Internal package information:
    Name: rp_super
    BuildTime: 2021-10-21_13.00
    ReleaseDate: 2021-10-21_03.11
    BootArchitecture: i686
    RouteProcessor: radium
    Platform: C8000BE
    User: mcpre
    PackageName: universalk9
    Build: BLD_V177_THROTTLE_LATEST_20211021_031123_V17_7_0_117
    CardTypes:

  Package is bootable from media and tftp.
  Package contents:

  Package:
c8000be-firmware_nim_ge.BLD_V177_THROTTLE_LATEST_20211021_031123_V17_7_0_117.SSA.pkg
    Size: 2966620
    Timestamp: 2021-10-21 20:10:44 UTC

    Raw disk-file SHA1sum:
      501d59d5f152ca00084a0da8217bf6f6b95dddb1
    Header size:     1116 bytes
    Package type:    40000
    Package flags:   0
    Header version:  3

    Internal package information:
      Name: firmware_nim_ge
      BuildTime: 2021-10-21_13.00
      ReleaseDate: 2021-10-21_03.11
      BootArchitecture: none
      RouteProcessor: radium
      Platform: C8000BE
      User: mcpre
      PackageName: firmware_nim_ge
      Build: BLD_V177_THROTTLE_LATEST_20211021_031123_V17_7_0_117
      CardTypes:

    Package is not bootable.
  Package:
c8000be-firmware_prince.BLD_V177_THROTTLE_LATEST_20211021_031123_V17_7_0_117.SSA.pkg
    Size: 10204252
    Timestamp: 2021-10-21 20:10:43 UTC

    Raw disk-file SHA1sum:
      a57bed4ddecfd08af3b456f69d11aaeb962865ea
    Header size:     1116 bytes
    Package type:    40000
    Package flags:   0
    Header version:  3

    Internal package information:
      Name: firmware_prince
      BuildTime: 2021-10-21_13.00
      ReleaseDate: 2021-10-21_03.11
      BootArchitecture: none
      RouteProcessor: radium
      Platform: C8000BE
      User: mcpre
      PackageName: firmware_prince
      Build: BLD_V177_THROTTLE_LATEST_20211021_031123_V17_7_0_117
      CardTypes:

    Package is not bootable.
```

show install active

```
Device# show install active
[ R0 ] Active Package(s) Information:
State (St): I - Inactive, U - Activated & Uncommitted,
            C - Activated & Committed, D - Deactivated & Uncommitted
--------------------------------------------------------------------
Type  St   Filename/Version
--------------------------------------------------------------------
IMG   C    17.07.01.0.1515

--------------------------------------------------------------------
Auto abort timer: inactive
--------------------------------------------------------------------
```

show install inactive

```
Device# show install inactive
[ R0 ] Inactive Package(s) Information:
State (St): I - Inactive, U - Activated & Uncommitted,
            C - Activated & Committed, D - Deactivated & Uncommitted
--------------------------------------------------------------------
Type  St   Filename/Version
--------------------------------------------------------------------
No Inactive Packages
```

show install committed

```
Device# show install committed
[ R0 ] Committed Package(s) Information:
State (St): I - Inactive, U - Activated & Uncommitted,
            C - Activated & Committed, D - Deactivated & Uncommitted
--------------------------------------------------------------------
Type  St   Filename/Version
--------------------------------------------------------------------
IMG   C    17.07.01.0.1515

--------------------------------------------------------------------
Auto abort timer: inactive
--------------------------------------------------------------------
```

show install uncommitted

```
Device# show install uncommitted
[ R0 ] Uncommitted Package(s) Information:
State (St): I - Inactive, U - Activated & Uncommitted,
            C - Activated & Committed, D - Deactivated & Uncommitted
--------------------------------------------------------------------
Type  St   Filename/Version
--------------------------------------------------------------------
No Uncommitted Packages
```

Troubleshooting Software Installation Using install Commands

Problem Troubleshooting the software installation

Solution Use the following show commands to view installation summary, logs, and software versions.

- **show install summary**
- **show install log**

- **show version**
- **show version running**

Problem Other installation issues

Solution Use the following commands to resolve installation issue:

- **dir** <*install directory*>
- **more location:***packages.conf*
- **show tech-support install**: this command automatically runs the **show** commands that display information specific to installation.
- **request platform software trace archive target bootflash** <*location*>: this command archives all the trace logs relevant to all the processes running on the system since the last reload, and saves this information in the specified location.

CHAPTER 5

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

This chapter describes the various public cloud compute instance sizes that are supported for Cisco Catalyst 8000V routers. The following topics specify the compute instance details for each of the supported public clouds and sovereign IaaS clouds.

Supported Instance Types for AWS

The AMI supports different instance types that determine the size of the instance and the required amount of memory. The following are the supported instance types for Cisco Catalyst 8000V:

Release Number	Supported Instance Types
Cisco IOS XE 17.13.1a	• t3.medium • c5.9xlarge, c5.2xlarge, c5.xlarge, c5.large • c5n.18xlarge, c5n.4xlarge • c6in.8xlarge, c6in.2xlarge, c6in.xlarge, c6in.large
Cisco IOS XE 17.12.2, Cisco IOS XE 17.12.1	• t3.medium • c5.9xlarge, c5.2xlarge, c5.xlarge, c5.large • c5n.18xlarge, c5n.9xlarge, c5n.4xlarge, c5n.2xlarge, c5n.xlarge, c5n.large
Cisco IOS XE 17.11.1a	• t3.medium • c5.9xlarge, c5.2xlarge, c5.xlarge, c5.large • c5n.18xlarge, c5n.9xlarge, c5n.4xlarge, c5n.2xlarge, c5n.xlarge, c5n.large

Release Number	Supported Instance Types
Cisco IOS XE 17.10.1a	• t3.medium • c5.9xlarge, c5.2xlarge, c5.xlarge, c5.large • c5n.18xlarge, c5n.9xlarge, c5n.4xlarge, c5n.2xlarge, c5n.xlarge, c5n.large
Cisco IOS XE 17.9.4a Cisco IOS XE 17.9.4 Cisco IOS XE 17.9.3a Cisco IOS XE 17.9.2a Cisco IOS XE 17.9.1a	• t3.medium • c5.9xlarge, c5.4xlarge, c5.2xlarge, c5.xlarge, c5.large • c5n.18xlarge, c5n.9xlarge, c5n.4xlarge, c5n.2xlarge, c5n.xlarge, c5n.large
Cisco IOS XE 17.8.1a	• t3.medium • c5.9xlarge, c5.4xlarge, c5.2xlarge, c5.xlarge, c5.large • c5n.9xlarge, c5n.4xlarge, c5n.2xlarge, c5n.xlarge, c5n.large
Cisco IOS XE 17.7.2 Cisco IOS XE 17.7.1a	• t3.medium • c5.9xlarge, c5.4xlarge, c5.2xlarge, c5.xlarge, c5.large • c5n.9xlarge, c5n.4xlarge, c5n.2xlarge, c5n.xlarge, c5n.large
Cisco IOS XE 17.6.6a Cisco IOS XE 17.6.6 Cisco IOS XE 17.6.5a Cisco IOS XE 17.6.5 Cisco IOS XE 17.6.4 Cisco IOS XE 17.6.3a Cisco IOS XE 17.6.2 Cisco IOS XE 17.6.1a	• t3.medium • c5.9xlarge, c5.4xlarge, c5.2xlarge, c5.xlarge, c5.large • c5n.9xlarge, c5n.4xlarge, c5n.2xlarge, c5n.xlarge, c5n.large

Release Number	Supported Instance Types
Cisco IOS XE 17.5.1a	• t3.medium, t2.medium • c4.8xlarge, c4.4xlarge, c4.2xlarge, c4.xlarge, c4.large • c5.9xlarge, c5.4xlarge, c5.2xlarge, c5.xlarge, c5.large • c5n.9xlarge, c5n.4xlarge, c5n.2xlarge, c5n.xlarge, c5n.large
Cisco IOS XE 17.4.2 Cisco IOS XE 17.4.1b Cisco IOS XE 17.4.1a	• t3.medium, t2.medium • c4.8xlarge, c4.4xlarge, c4.2xlarge, c4.xlarge, c4.large • c5.9xlarge, c5.4xlarge, c5.2xlarge, c5.xlarge, c5.large • c5n.9xlarge, c5n.4xlarge, c5n.2xlarge, c5n.xlarge, c5n.large

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

Supported Instance Types for Microsoft Azure

The following 2, 4 and 8 NIC solution templates are currently offered in the Microsoft Azure marketplace in the public cloud:

Cisco IOS XE Release	Supported Instance Types/Max NICs supported
Cisco IOS XE 17.13.1a	• D4_v2 / DS4_v2 • D3_v2 / DS3_v2 • D2_v2 / DS2_v2 • F16s_v2 • F32s_v2 • D16_v5

Cisco IOS XE Release	Supported Instance Types/Max NICs supported
Cisco IOS XE 17.12.2, Cisco IOS XE 17.12.1	• D4_v2 / DS4_v2 • D3_v2 / DS3_v2 • D2_v2 / DS2_v2 • F16s_v2 • F32s_v2 • D16_v5
Cisco IOS XE 17.11.1a	• D4_v2 / DS4_v2 • D3_v2 / DS3_v2 • D2_v2 / DS2_v2 • F16s_v2 • F32s_v2
Cisco IOS XE 17.10.1a	• D4_v2 / DS4_v2 • D3_v2 / DS3_v2 • D2_v2 / DS2_v2 • F16s_v2 • F32s_v2
Cisco IOS XE 17.9.4a Cisco IOS XE 17.9.3a Cisco IOS XE 17.9.2a Cisco IOS XE 17.9.1a	• D4_v2 / DS4_v2 • D3_v2 / DS3_v2 • D2_v2 / DS2_v2 • F16s_v2 • F32s_v2
Cisco IOS XE 17.8.1a	• D4_v2 / DS4_v2 • D3_v2 / DS3_v2 • D2_v2 / DS2_v2 • F16s_v2 • F32s_v2

Cisco IOS XE Release	Supported Instance Types/Max NICs supported
Cisco IOS XE 17.7.2 Cisco IOS XE 17.7.1a	• D4_v2 / DS4_v2 • D3_v2 / DS3_v2 • D2_v2 / DS2_v2 • F16s_v2 • F32s_v2
Cisco IOS XE 17.6.6a Cisco IOS XE 17.6.5a, Cisco IOS XE 17.6.4a Cisco IOS XE 17.6.3a Cisco IOS XE 17.6.2a Cisco IOS XE 17.6.1a	• D4_v2 / DS4_v2 • D3_v2 / DS3_v2 • D2_v2 / DS2_v2 • F16s_v2 • F32s_v2
Cisco IOS XE 17.5.1a	• D4_v2 / DS4_v2 • D3_v2 / DS3_v2 • D2_v2 / DS2_v2 • F16s_v2 • F32s_v2
Cisco IOS XE 17.4.2 Cisco IOS XE 17.4.1b Cisco IOS XE 17.4.1a	• D4_v2 / DS4_v2 • D3_v2 / DS3_v2 • D2_v2 / DS2_v2 • F16s_v2 • F32s_v2

Supported Instance Types for Google Cloud Platform

Cisco IOS XE Release	Supported Instance Types	Notes
Cisco IOS XE 17.13.1a	N1: n1-standard-4, n1-standard-8	BYOL only
Cisco IOS XE 17.12.2 Cisco IOS XE 17.12.1a	N1: n1-standard-4, n1-standard-8	BYOL only

Cisco IOS XE Release	Supported Instance Types	Notes
Cisco IOS XE 17.11.1a	N1: n1-standard-2, n1-standard-4, n1-standard-8	BYOL only
Cisco IOS XE 17.10.1a	N1: n1-standard-2, n1-standard-4, n1-standard-8	BYOL only
Cisco IOS XE 17.9.4a Cisco IOS XE 17.9.3a Cisco IOS XE 17.9.2a Cisco IOS XE 17.9.1a	N1: n1-standard-2, n1-standard-4, n1-standard-8	BYOL only
Cisco IOS XE 17.8.1a	N1: n1-standard-2, n1-standard-4, n1-standard-8	BYOL only
Cisco IOS XE 17.7.2 Cisco IOS XE 17.7.1a	N1: n1-standard-2, n1-standard-4, n1-standard-8	BYOL only
Cisco IOS XE 17.6.6a Cisco IOS XE 17.6.5a Cisco IOS XE 17.6.5 Cisco IOS XE 17.6.4 Cisco IOS XE 17.6.3a Cisco IOS XE 17.6.2 Cisco IOS XE 17.6.1a	N1: n1-standard-2, n1-standard-4, n1-standard-8	BYOL only
Cisco IOS XE 17.5.1a	N1: n1-standard-2, n1-standard-4, n1-standard-8	BYOL only
Cisco IOS XE 17.4.2 Cisco IOS XE 17.4.1b Cisco IOS XE 17.4.1a	N1: n1-standard-1, n1-standard-2, n1-standard-4, n1-standard-8	BYOL only Support for both autonomous and controller modes

Supported Instance Types for Sovereign Clouds

Table 5: Supported Instances for Alibaba Cloud (China)

Cisco IOS XE Release	Instance Types (BYOL Only)
Cisco IOS XE 17.9.1a Cisco IOS XE 17.8.1a Cisco IOS XE 17.7.2 Cisco IOS XE 17.7.1a Cisco IOS XE 17.6.6a Cisco IOS XE 17.6.6 Cisco IOS XE 17.6.5a Cisco IOS XE 17.6.5 Cisco IOS XE 17.6.4 Cisco IOS XE 17.6.3a Cisco IOS XE 17.6.2 Cisco IOS XE 17.6.1a	• ecs.g5.large • ecs.g5.4xlarge

Table 6: Supported Instances for AWS (China)

Cisco IOS XE Release	Instance Types
Cisco IOS XE 17.9.1a	• t3.medium • c4.large, • c5.large, c5.xlarge, c5.2xlarge, c5.4xlarge, c5.9xlarge
Cisco IOS XE 17.6.6a Cisco IOS XE 17.6.6 Cisco IOS XE 17.6.5a Cisco IOS XE 17.6.5 Cisco IOS XE 17.6.4 Cisco IOS XE 17.6.3a Cisco IOS XE 17.6.2 Cisco IOS XE 17.6.1a	• t3.medium • c4.large, • c5.large, c5.xlarge, c5.2xlarge, c5.4xlarge, c5.9xlarge

CHAPTER 6

Installing in VMware ESXi Environment

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

From Cisco IOS XE 17.12.1 release, Cisco Catalyst 8000V is supported on Intel Atom® C3000 processor (Denverton) CPU-based servers with Intel x550 NIC on VMware ESXi 7.0.x. You can run Cisco Catalyst 8000V on other x86 CPUs with different versions of hypervisor operating systems, but support is only available on VMware ESXi 7.0.x.

This chapter contains information about how to deploy Cisco Catalyst 8000V in ESXi, and the requirements for a successful deployment. Before you read the requirements and the deployment procedures, see the following information that tells you the various deplopyment methods for the ESXi hypervisor:

Caution Oversubscription of host resources can lead to a reduction of performance and your instance could become instable. We recommend that you follow the guidelines and the best practices for your host hypervisor

Deploying the OVA template on the VM

Deploying using the OVA file: In this method, you must download the .ova file from Software Download page, and use this file for the deployment. Further, you can use the following two methods to deploy the OVA file:

- **Deploying using the vSphere client**: In this procedure, you need a VMware vSphere Client or a vSphere Web Client to deploy the *.ova installation file. The 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 a Cisco Catalyst 8000V instance.

Note This is the recommended method of deployment for Cisco Catalyst 8000V.

- **Deploying using the Common Ovf Tool (COT)**: COT is a tool that allows you to edit virtual appliances such as Cisco Catalyst 8000V. You can also use this tool to deploy the .ova file to the ESXi server and provision the VM.

To learn more about VMware vSphere products, see VMware product documentation.

Manually deploying the .iso file

The third deployment option for the ESXi hypervisor is the manual creation of the VM and installation of Cisco Catalyst 8000V by using the .iso file. Download the .iso file from the Cisco Software Download page and use this file for the installation. In this method, you install the .iso file on the VMware ESXi host and manually create the VM using the vSphere GUI. This option is advisable only if you want to modify the OVA. However, note that this option is the least recommended since manual deployments invite opportunities to stray from supported configurations.

Important Create the VM using ESXi 6.5 or later. Ensure that you use VM version 13 or greater. To choose the EFI firmware mode, navigate through **VM Options** > **Boot Options** > **Firmware** > **EFI**. The firmware mode is required to enable the secure boot functionality. For more information, see Enabling VNF Secure Boot, on page 125.

Important You cannot modify the firmware mode (from BIOS to EFI or vice versa) after you create the VM.

VMware Requirements

The following section specifies the requirements for you to install Cisco Catalyst 8000V 17.4.1 and later releases in ESXi environment. These requirements have been fully tested and meet performance benchmarks.

ESXi Hypervisor Requirements

Cisco IOS XE Release	vSphere Web Client	vCenter Server
Cisco IOS XE 17.15.x releases Cisco IOS XE 17.14.x releases Cisco IOS XE 17.13.x releases Cisco IOS XE 17.12.x releases	The 8.0 version of the VMware vSphere Web Client is supported.	VMware ESXi 8.0

Cisco IOS XE Release	vSphere Web Client	vCenter Server
Cisco IOS XE 17.11.x releases Cisco IOS XE 17.10.x releases Cisco IOS XE 17.9.x releases Cisco IOS XE 17.8.x releases Cisco IOS XE 17.7.x releases Cisco IOS XE 17.6.x releases	The 7.0 and 6.7 versions of the VMware vSphere Web Client are supported.	VMware ESXi 7.0 and ESXi 6.7
Cisco IOS XE 17.5.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
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

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

vCPU Requirements

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

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

vNIC Requirements

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

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

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

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

Other Requirements

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

Next Step

Familiarize yourself about the secure boot configuration before you proceed with the installation. To see information about secure boot, see Enabling VNF Secure Boot, on page 125.

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 7: 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 8: 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.
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 9: 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. **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 10: 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.

Supported Entities	Description
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 11: 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. • Cisco Catalyst 8000V supports USB disk hot-plug. • You can use only two USB disk hot-plug devices at a time. • USB hub is not supported.

Deploying the 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.

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 12: OVA Bootstrap Properties

Property	Description
Bootstrap Properties	
Console	Configures the console mode. 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. **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. 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. 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	

Property	Description
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. The available license levels are: • network-essentials • network-advantage • network-premier **Note** For details on Cisco DNA licenses, see Cisco DNA Software for SD-WAN and Routing.

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 Deployment** 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 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.

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 propertie`
`s 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

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.c8000v”.
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.

Script Step	Description
`-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, see http://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"`
`-N "GigabitEthernet2=VM Network"`
`-N "GigabitEthernet3=VM Network"`
	Mapping each NIC in the Cisco Catalyst 8000V OVA to a vSwitch on the server.

Script Step	Description
`esxi`	Target hypervisor (currently always ESXi)
`~/Downloads/c8000v-universalk9.ova`	OVA to deploy
`-ds=datastore38a`	Any ESXi-specific parameters - here, the datastore to use for disk storage.

Manually Creating the VM Using the .iso File

Perform the following steps to install the .iso file on the VMware ESXi host and manually and create the VM using the vSphere GUI. While this 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 instructions 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 From the **Virtual Machine Version** field, select **Virtual Machine Version 15** or a higher version that is availble. Click **Next**.

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

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 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 From the **Create a Disk** field, configure the following:

a) **Capacity: Disk Size**: See the requirements sections in this guide for the virtual hard disk size required in your release.
b) **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.

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

Click **Next**.

Step 14 From the **Advanced Options** field, 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. Select 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 Booting the Cisco Catalyst 8000V VM and Accessing the Console sections.

Note To configure the day0 settings of a manually installed Cisco Catalyst 8000V, attach a second CD/DVD drive pointing to an ISO that contains the said bootstrap configuration. For further details on the supported bootstrap ISO contents, see Day 0 Configuration, on page 93.

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 the 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 7

Installing in KVM Environments

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

From Cisco IOS XE 17.12.1 release, Cisco Catalyst 8000V is also supported on Intel Atom® C3000 processor (Denverton) CPU-based servers with Intel x550 NIC on RHEL 8.4 KVM hypervisor. You can run Cisco Catalyst 8000V on other x86 CPUs with different versions of hypervisor operating systems, but support is only available on these listed versions.

You can install the Cisco Catalyst 8000V virtual router as a virtual machine on Red Hat KVM virtualization. The installation procedure first involves the manual creation of a VM. This is followed by the installation using a .iso file or a qcow2 file. You can install Cisco Catalyst 8000V in a KVM environment by using the:

- **GUI Tool**: Download and install the virt-manager RPM package on the KVM server. Virt-manager is a desktop user interface for managing virtual machines. Installation by using the GUI is the recommended installation method.
- **Command Line Interface**: In this method of installation, use the command line interface to install the Cisco Catalyst 8000V VM.

Note Deploying the OVA template in 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.

Installation Requirements for KVM

This section specifies for installing Cisco Catalyst 8000V Cisco IOS XE 17.4.x release and later in KVM environment:

KVM Versions (Linux KVM based on Red Hat Enterprise Linux)

Cisco IOS XE Release	KVM Version
Cisco IOS XE 17.15.x releases Cisco IOS XE 17.14.x releases Cisco IOS XE 17.13.x releases Cisco IOS XE 17.12.x releases	Linux KVM based on Red Hat Enterprise Linux 9.2 and 8.4 are recommended.
Cisco IOS XE 17.11.x releases Cisco IOS XE 17.10.x releases Cisco IOS XE 17.9.x releases Cisco IOS XE 17.8.x releases Cisco IOS XE 17.7.x releases	Linux KVM based on Red Hat Enterprise Linux 7.7 and 8.4 are recommended.
Cisco IOS XE 17.4.x releases Cisco IOS XE 17.5.x releases Cisco IOS XE 17.6.x releases	Linux KVM based on Red Hat Enterprise Linux 7.5 and 7.7 are recommended.

KVM Versions (SUSE Linux® Enterprise Server)

Cisco IOS XE Release	KVM Version
Cisco IOS XE 17.15.x releases	Supports SUSE Linux Enterprise Server version 15 SP5
Cisco IOS XE 17.14.x releases Cisco IOS XE 17.13.x release Cisco IOS XE 17.12.x release Cisco IOS XE 17.11.x releases Cisco IOS XE 17.10.x releases Cisco IOS XE 17.9.x releases Cisco IOS XE 17.6.3 release	Supports SUSE Linux Enterprise Server version 15 SP3

Supported VNICs

VNIC	Supported Releases
Ixgbe	Cisco IOS XE Release 17.10.1 and later
iavf	Cisco IOS XE Release 17.9.1 and later
ConnectX-5VF	Cisco IOS XE Release 17.9.1 and later

VNIC	Supported Releases
i40evf	Cisco IOS XE Release 17.4.1 to Cisco IOS XE 17.8.x releases
Virtio	Cisco IOS XE Release 17.4.1 and later
ixgbevf	Cisco IOS XE Release 17.4.1 and later

Note The maximum number of vNICs supported per VM instance is 26.

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

Supported vCPU Configurations

The vCPU configurations listed in this section 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
- 8 vCPUs: requires minimum 8 GB RAM allocation
- 16 vCPUs: requires minimum 8 GB RAM allocation (supported from Cisco IOS XE 17.11.1a)

Creating a KVM Instance

Creating the VM Using the GUI Tool

Before you begin

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

Download either the .qcow2 image or the .iso image from the Cisco Software Download page, and copy the file onto a local device or a network device.

Step 1 Launch the virt-manager GUI.

Step 2 Click **Create a New Virtual Machine**.

Step 3 Do one of the following:

a) If you have downloaded the .qcow2 file, select **Import Existing Disk Image**.

b) If you have downloaded the .iso file, select **Local Install Media (ISO Image or CDROM)**.

Step 4 Select the Cisco Catalyst 8000V qcow2 or iso file location.

Step 5 Configure the memory and the CPU parameters.

Step 6 Configure the virtual machine storage.

Step 7 (Optional) To add additional hardware before creating the VM, select **Customize configuration before install**. The system displays the **Add Hardware** button. Click this button to add various hardware options, such as additional disks or a serial port interface.

Step 8 (Optional) To add a serial console, follow the procedure as mentioned in Adding a Serial Console, on page 68.

Step 9 (Optional) If you want to customize your configuration before you create the VM, see Customizing Configuration Before Creating the VM, on page 68.

Step 10 Click **Finish**.

Step 11 Access the Cisco Catalyst 8000V console by performing one of the following actions:

a) If you are using a virtual console, double-click the VM instance to access the VM console.

b) If you are using a serial console, see Booting the Cisco Catalyst 8000V and Accessing the Console.

Adding a Serial Console

Perfrom this task to enables acces to the Cisco Catalyst 8000V instance by adding a serial console.

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**.

Customizing Configuration Before Creating the VM

Before you begin

Perform the Creating the VM Using the GUI Tool, on page 67 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. Then, perform this procedure, which tells you how to customize your configuration before you move on to the Day Zero configuration.

Step 1 Click **Add Hardware**.

Step 2 Click the **Storage** option.

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

Step 4 Click **Browse** and navigate to the location where you have the **c8000v_config.iso** file.

This step is applicable only when you want to add a Day Zero or bootstrap configuration.

Step 5 From the **Device-type** drop-down list, choose **IDE CDROM**.

Step 6 Click **Finish**.

Step 7 To adjust the VM boot options including the boot device order in the virtual manager, select the **IDE CDROM** or the **c8000v_config.iso**, and use the Up Arrow to move this device as the first option in the boot order.

Step 8 Click **Apply** to apply the changes you made to the boot order.
The image disk device option is pushed down as the second option in the boot device order.

Step 9 After adding all the necessary hardware, click **Begin Installation** to start the new VM deployment.

What to do next

Perform the Day Zero configuration. To learn how to do this, see Day 0 Configuration, on page 93.

Creating the VM Using CLI

- 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.

Step 1 To create the VM for a .qcow2 image, use the virt-install command to create the instance and boot. Use 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
```

Step 2 To create the VM, for a .iso image, perform the following steps:

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

Example:

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

b) 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_c8000v_vm          \
--description "Test VM"   \
--os-type=linux           \
--os-variant=rhel4        \
--arch=x86_64             \
--cpu host                \
--vcpus=1,sockets=1,cores=1,threads=1    \
--hvm                     \
--ram=4096                \
--cdrom=<path_to_c8000v_iso>           \
--disk path=c8000v_disk.qcow2,bus=virtio,size=8,sparse=false,cache=none,format=qcow2   \
--network bridge=virbr0,model=virtio     \
--noreboot
```

The **virt-install** command creates a new VM instance and 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 VM by executing the **virsh start** command.

Note If you want to provide the day0 configuration through the c8000v_config.iso disk image, add an additional parameter to the **virt-install** command. For example, `--disk path=/my/path/c8000v_config.iso,device=cdrom,bus=ide`. For more information, see Day 0 Configuration, on page 93.

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
```

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.

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

Configuration	Recommended Setting
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'>
```

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 8

Installing in an NFVIS Environment

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

The Cisco Enterprise NFVIS solution helps you convert your critical network functions into software, making it possible to deploy network services in minutes across dispersed locations. This solution provides a fully integrated platform that can run on top of a diverse network of both virtual and physical devices.

This chapter specifies how you can upgrade from Cisco Integrated Services Virtual Router (ISRv) to Cisco Catalyst 8000V. If your hardware is running on Cisco NFVIS, and you want to deploy this setup on a Cisco Catalyst 8000V, perform the procedures as mentioned in the *Installing the VM on NFVIS* section.

Note From the Cisco IOS XE 17.4.x release onwards, Cisco Catalyst 8000V replaces ISRv.

Cisco Catalyst 8000V requires NFVIS version 4.4 or later for deployments.

Supported Hardware Platforms running NFVIS

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

Supported NIMS

- NIM-4G-LTE-VZ
- NIM-4G-LTE-ST
- NIM-4G-LTE-NA
- NIM-4G-LTE-GA
- NIM-4G-LTE-LA
- NIM-LTEA-EA
- NIM-LTEA-LA

- NIM-1MFT-T1/E1
- NIM-2MFT-T1/E1
- NIM-4MFT-T1/E1
- NIM-8MFT-T1/E1
- NIM-1CE1T1-PRI
- NIM-2CE1T1-PRI
- NIM-8CE1T1-PRI
- NIM-16A
- NIM-24A
- NIM-VA-B
- NIM-VAB-A
- NIM-VAB-M
- NIM-4SHDSL-EA
- NIM-1GE-CU-SFP
- NIM-2GE-CU-SFP
- NIM-ES2-8-P
- NIM-ES2-8 NIM-ES2-4

Supported NICs

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

Supported Profiles

- Mini – 1vCPU
- Small – 2vCPU

- Medium – 4vCPU
- Large - 4vCPU

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

Install the VM in NFVIS

From the Cisco IOS XE 17.4.1 release, you can either freshly install a Cisco Catalyst 8000V VM in NFVIS, or you can upgrade from an Cisco ISRv to Cisco Catalyst 8000V. The following are the key tasks that you must perform for the installation or the upgrade:

- **Register a VM image**: To register a VM image, you must first copy or download the VM image to the NFVIS server or host the image on a HTTP or HTTPs server. After you download the file, register the image using the registration API. This API allows you to specify the file path to the location (on an HTTP or HTTPs server) where the tar.gz file is hosted. Registering the image is a one-time activity. After you register an image on the HTTP or HTTPs server, and the registration is in the active state, you can perform multiple VM deployments using the registered image.

- **Create a custom profile**: After registering a VM image, you can optionally create a custom profile for the VM image. This is especially beneficial if the profiles defined in the image file do not match your requirements. Custom profiles allow you to provide specific profiling details for a VM image such as the virtual CPU on which the VM will run, the amount of virtual memory the VM will consume. Depending on the topology that you require, you can create additional networks and bridges to attach the VM during deployment.

- **Deploy the VM**: Deploy the VM by using the deployment API. This API allows you to provide values to the parameters that are passed to the system during deployment. Depending on the VM that you are deploying, some parameters are mandatory and others are optional. For more details on the APIs, see the VM Lifecycle Management APIs .

- **Manage and monitor the VM**: You can monitor a VM using APIs and commands that enable you to get the VM status and debug logs. Using the VM management APIs, you can start, stop, or reboot a VM, and view the statistics for a VM, such as CPU usage. You can also change or update a VM profile. You can change a VM profile to one of the existing profiles in the image file. Alternatively, you can create a new custom profile for the VM. The vNICs on a VM can also be added or updated.

Install the VM in NFVIS (Release 4.5.1 and Later)

Install Cisco Catalyst 8000V in NFVIS Environment

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

To install Cisco Catalyst 8000V on earlier releases of NFVIS, see Installing in an NFVIS Environment, on page 77.

Upload the Image on NFVIS

Step 1 Log in to the NFVIS Portal.

Step 2 Choose **Configuration** > **Virtual Machine** > **Images** > **Image Repository**.

Step 3 Upload the installation file by doing one of the following:

- Choose **Local** > **Select File**, and from your device, locate and select the installation file.
- Choose **Remote**.

Step 4 If you chose Remote, provide the following details:

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

Create a Network

Step 1 In the NFVIS Portal, choose **Configuration** > **Virtual Machine** > **Networking** > **Networks**.

Step 2 To create a new network, click the + icon.

Step 3 In the **Add Network** area, enter the following details:

a) **Network**: Choose the network from this drop-down list.
b) **Mode**: Enter the mode in which the VNF will boot.
c) **VLAN**: Choose the VLAN for the VM.
d) **VLAN-Range**: Enter the range of VLAN for your VM.
e) **Native VLAN**: Choose the native VLAN for your VM from this field.
f) **Bridge**: The Layer 2 domain between virtual network interface controllers (vNICs) of VMs. Choose either the **Existing** or the **Create New** radio button.
g) **Interface**: Choose the interface from this field for your VM.

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

Step 4 Click **Submit**.

Create a VM Package

Step 1 In the NFVIS portal, choose **Configuration** > **Virtual Machine** > **Images** > **Image Packaging**.

Step 2 To create a VM package, click the + icon.

Step 3 In the **Image Packaging** area, enter the following details:

a) **Name**: The name associated with the VM packaging.
b) **Version**: The version of the package.
c) **VM Type**: The type of the VM for which you're creating the package.
d) **Dedicate Cores (Optimize)**: The dedicated core a container requires. By default, the value is **False**.
e) **Serial Console**: The field to either enable or disable access through the serial console. By default, the value is **Disable**.
f) **SRIOV Driver**: The SRIOV supported by the VM interfaces.
g) **Local**: The option you must use if the image you want to bundle is available in the intdatastore.
h) **Upload Raw Images**: The option to upload an image to be packaged from your local machine.
i) **Raw Disk File Bus**: Choose the root disk image bus from this drop-down list.
j) **Thick Disk Provisioning**: Choose true from the drop-down list to enable thick provisioning. By default, the value is false.

Step 4 To upload the bootstrap file, do one of the following:

- Choose **Local** and choose the **Add Local File** option to add a locally available bootstrap file.
- Choose the **Upload Bootstrap Files** option to browse to the bootstrap configuration file from your computer.
- Choose the track state of the VM from the **Monitored** drop-down field. By default, the value is **False**.

Step 5 Click **Submit** to generate the VM Package.

Deploy the VM

Step 1 In the NFVIS Portal, choose **Configuration** > **Deploy**.

Step 2 In the **VM Deployment** window, choose the **Router** icon.

Step 3 Click on the VM. Four drag handlers appears around the VM. Drag from one of those handlers to any of the networks and provide the details.

Step 4 In the **VM Details** area, enter the following details:

a) **VM Name**: Specify the name for your VM.
b) **Image**: Choose the appropriate value from the drop-down list.
c) **Profile**: Choose a profile from the drop-down list. The default profile is used when no profile is specified during deployment.
d) **Group Name**: Choose a group if you want this VM to be associated with a specific group.
e) **VNC Password**: Enter the VNC password in this field.

f) **Controller**: Choose **non-vManage** to deploy in autonomous mode and choose **vManage** to deploy the VM in controller mode.
g) **Tech Package**: Choose the desired tech package from the drop-down field. The available options are network essentials, network advantage, and network premier.
h) **NGIO**: The Next Generation Input/Output (NGIO) option that decides the NIM enablement capability available for the VM. Choose **ENABLE** from the NGIO drop-down list to enable NGIO.
i) **SSH Username**: The username to remotely log in to the Cisco Catalyst 8000V VM.
j) **SSH Password**: The SSH password to access the VM.
k) **Port Number**: The port number that is required for the SSH connectivity to the VM.
l) **External Port Number**: The external port number that is required for the SSH connectivity to the VM.

Step 5 Click **Deploy**.

Install the VM in NFVIS (Release 4.5.0 and Earlier)

Deploy the Virtual Machine on NFVIS

Step 1 From the NFVIS Portal select **VM Lifecycle** > **Deploy**.

Step 2 From the VM Deployment window, drag and drop the Router icon to the pane below and map to the desired networks as required.

Step 3 In the VM Details section, enter the **VM Name**.

Step 4 From the **Image** drop-down field, select the appropriate value.

Step 5 From the **Mode** drop-down field, select the **non-vManage** option.

Step 6 From the **Profile** drop-down field, select the profile name.

Step 7 From the **Tech Package** drop-down field, select the desired tech package.

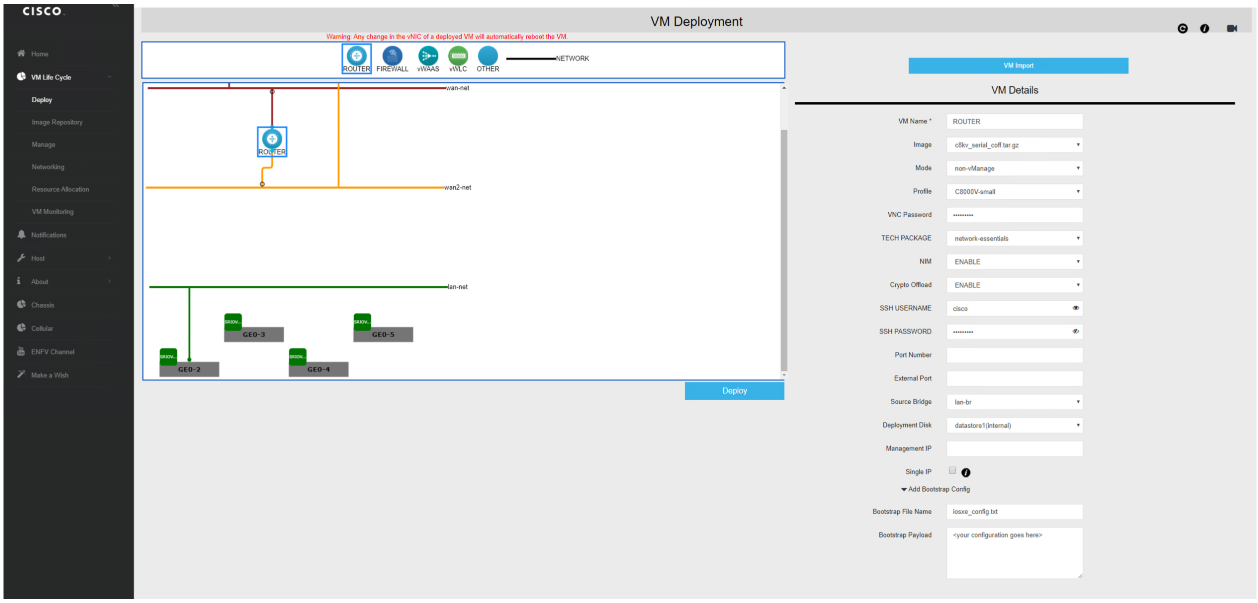
Step 8 If a specific network function physical hardware is installed, you can pass it through into the VM by selecting **ENABLE** from the **NIM** drop-down field.

Step 9 Select the **ENABLE** option from the **Crypto Offload** drop-down field to offload the crypto processing to a hardware chip.

Step 10 Enter the username and password for the ssh login for Cisco Catalyst 8000V.

Step 11 Optionally, add other VM details like **VNC Password**, **Port Number**, **External Port**, **Source Bridge**, **Deployment Disk**, and **Management IP**.

Step 12 Select the **Add Bootstrap Config** option to provide the bootstrap configuration file before deploying the VM. Ensure that you use the filename `iosxe_config.txt` for the bootstrap configuration file.



VM Deployment

Warning: Any change in the vNIC of a deployed VM will automatically reboot the VM.

VM Details

VM Name *	ROUTER
Image	c8kv_serial_coff.tar.gz
Mode	non-vManage
Profile	C8000V-small
VNC Password	•••••••••
TECH PACKAGE	network-essentials
NIM	ENABLE
Crypto Offload	ENABLE
SSH USERNAME	cisco
SSH PASSWORD	•••••••••
Port Number	
External Port	
Source Bridge	lan-br
Deployment Disk	datastore1(internal)
Management IP	
Single IP	□

Add Bootstrap Config

Bootstrap File Name	iosxe_config.txt
Bootstrap Payload	<your configuration goes here>

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

Step 13 Click **Deploy**.

What to do next

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

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

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

Download the Cisco Catalyst 8000V Image for NFVIS

Step 1 Go to https://software.cisco.com/download/home

Step 2 In the Search bar at the bottom of the page, search Cisco Catalyst 8000V.

Step 3 Select the **Software Type** from the list. For example, IOS XE Software.

Step 4 From the list of files, download the latest Cisco Catalyst 8000V image file with the tar.gz extension.

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

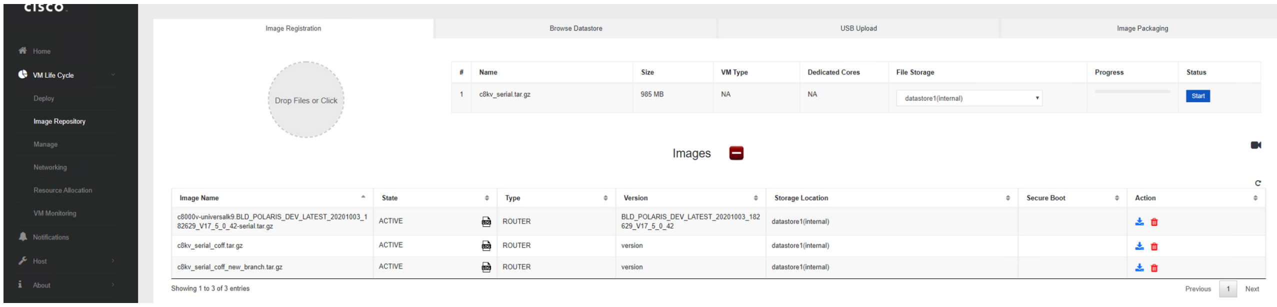
Upload the Image on NFVIS

Step 1 Log in to the NFVIS Portal.

Step 2 Select **VM Lifecycle** > **Image Repository**.

Step 3 Click the **Image Registration** tab, and click the upload arrow next to the **Images** option.

Step 4 From the **Drop Files or Click** option on the top of the page, select the appropriate file.



#	Name	Size	VM Type	Dedicated Cores	File Storage	Progress	Status
1	c8kv_serial.tar.gz	985 MB	NA	NA	datastore1(internal)		Start

Image Name	State	Type	Version	Storage Location	Secure Boot	Action
c8000v-universalk9.BLD_POLARIS_DEV_LATEST_20201003_182629_V17_5_0_42-serial.tar.gz	ACTIVE	ROUTER	BLD_POLARIS_DEV_LATEST_20201003_182629_V17_5_0_42	datastore1(internal)		
c8kv_serial_coff.tar.gz	ACTIVE	ROUTER	version	datastore1(internal)		
c8kv_serial_coff_new_branch.tar.gz	ACTIVE	ROUTER	version	datastore1(internal)		

Showing 1 to 3 of 3 entries

Previous 1 Next

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

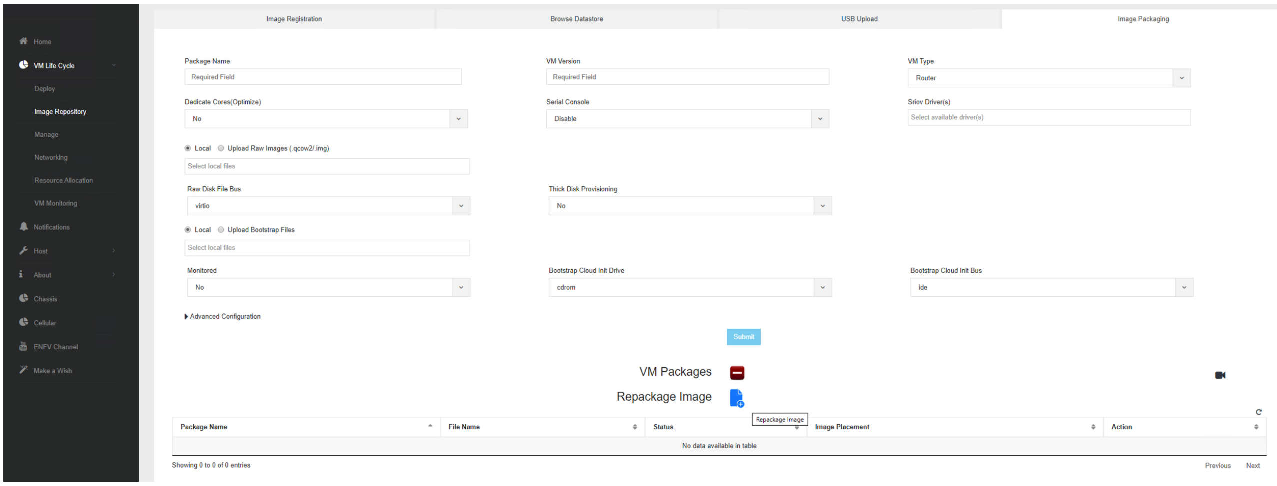
After the image is uploaded, NFVIS creates the respective profiles and registers the image. You can find your file listed under the **Images** section on the same page.

Create a VM Package Using the Web Interface

Step 1 From the NFVIS Web Portal, select **Image Repository** > **Image Packaging**. Click the **Create** icon.

Step 2 Click **VM Packages**.

Step 3 Enter the details in **Image Packaging** tab. Select **Yes** from the **Dedicated Code** drop-down list.



Package Name: Required Field
VM Version: Required Field
VM Type: Router
Dedicate Cores(Optimize): No
Serial Console: Disable
Sriov Driver(s): Select available driver(s)
● Local ○ Upload Raw Images (.qcow2/.img)
Select local files
Raw Disk File Bus: virtio
Thick Disk Provisioning: No
● Local ○ Upload Bootstrap Files
Select local files
Monitored: No
Bootstrap Cloud Init Drive: cdrom
Bootstrap Cloud Init Bus: ide

Package Name	File Name	Status	Image Placement	Action
No data available in table				

Showing 0 to 0 of 0 entries

Previous Next

Step 4 Click **Submit**. The bootstrap files are uploaded.

After the image is created, you have to register the image so that the profiles are populated in NFVIS.

Step 5 Select the image that you created and click **Register**.

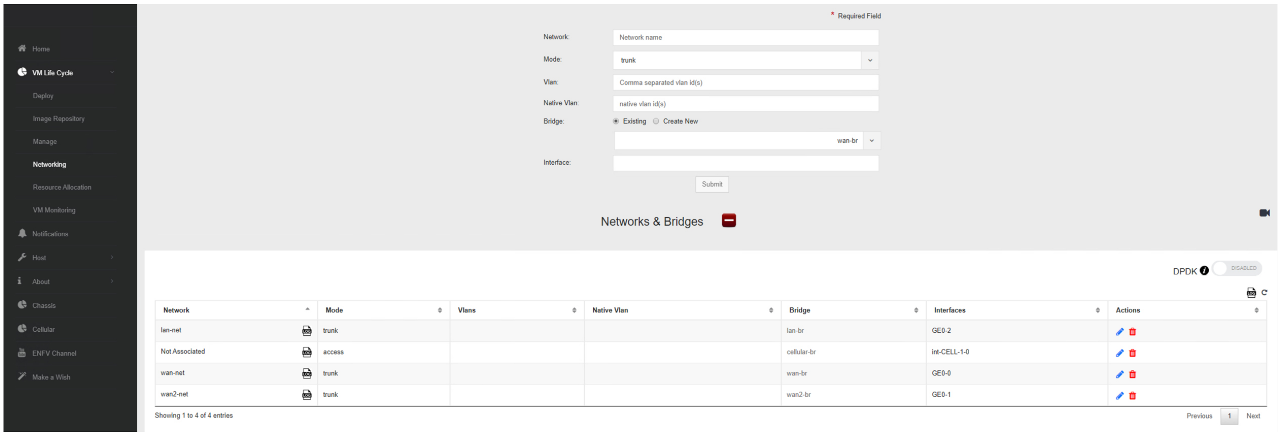
Create a Network

Step 1 From the NFVIS Portal, select **VM Lifecycle** > **Networking**. The system displays the **Networks & Bridges** page.

Step 2 Click the **Create** icon next to Networks & Bridges.

Step 3 Enter the appropriate values for the **Network**, **Mode**, **VLAN**, **Bridge**, and **Interface** fields.

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



* Required Field

Network: Network name

Mode: trunk

Vlan: Comma separated vlan id(s)

Native Vlan: native vlan id(s)

Bridge: ● Existing ○ Create New

wan-br

Interface:

Networks & Bridges

Network	Mode	Vlans	Native Vlan	Bridge	Interfaces	Actions
lan-net	trunk			lan-br	GE0-2	
Not Associated	access			cellular-br	int-CELL-1-0	
wan-net	trunk			wan-br	GE0-0	
wan2-net	trunk			wan2-br	GE0-1	

Showing 1 to 4 of 4 entries

Step 4 Click **Submit**. The network is now created.

Monitor the Virtual Machine

This procedure specifies the steps to monitor the VM and provides operational information such as resource allocation, VM statistics, and so on.

Step 1 To view the VM Resource Allocation follow these steps:

a) From the NFVIS Portal select **VM Life Cycle** > **Resource Allocation**. The system displays the VM CPU Allocation tab which displays the overall CPU allocation.

b) Click **VM Memory Allocation** to view the overall memory allocations.

c) Click **VM Disk Allocation** to view the overall disk allocations.

Step 2 To view the VM Stastics, perform the following steps:

a) From the NFVIS Portal select **VM Life Cycle** > **Resource Allocation**.

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

b) Click **Memory Allocation** to view the memory utilization per VM.

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

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

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

Upgrade and Downgrade Between Cisco ISRV and Cisco Catalyst 8000V

From the Cisco IOS XE 17.4.x release onwards, Cisco Catalyst 8000V replaces Cisco Integrated Services Virtual Router (ISRV). As a user, you have the option of upgrading your existing ISRv routers into Cisco Catalyst 8000V. To know how to upgrade to the latest release of Cisco Catalyst 8000V, see Upgrading the Cisco IOS XE Software, on page 173.

Note

- You cannot downgrade from Cisco Catalyst 8000V to Cisco ISRV.
- To upgrade from Cisco ISRV to Cisco Catalyst 8000V, the minimum version of Cisco ISRV supported are 16.12.4, 17.2.3, 17.3.2 . You cannot upgrade to Cisco Catalyst 8000V if you are using a Cisco ISRV device running on any other version than the ones mentioned above.
- When you upgrade from Cisco IOS XE 17.1.x or an earlier release to Cisco IOS XE 17.4.x, the **install add file bootflash:c8000v-universalk9.XXX.bin activate commit** command is not supported. To upgrade the Cisco ISRV to Cisco Catalyst 8000V, copy the `c8000v-universalk9.XXX.bin` file to `bootflash:`, under the Configuration folder. Then, use the **write memory** command to copy the configuration and start the upgrade process.
- If you are an existing Cisco CSR1000V user running Cisco IOS XE 16.12.3 release or earlier, and want to upgrade to Cisco Catalyst 8000V, you cannot upgrade by using the Web UI. You must first upgrade to Cisco CSR1000V releases 16.12.4, 17.2.3, or 17.3.2 before you upgrade to Cisco Catalyst 8000V.
- All licensing information is retained after you upgrade to Cisco Catalyst 8000V.

Supported Upgrade Paths

Autonomous Mode

- 16.12.x > 17.4 C8000V
- 17.2.x > 17.4 C8000V
- 17.3.x > 17.4 C8000V

Controller Mode

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

Note When you upgrade Cisco ISRV to Cisco Catalyst 8000V in the controller mode, first upgrade Cisco IOS XE to 17.3.1 and later releases or 16.12.4 and later releases.

CHAPTER 9

Installing in OpenStack Environment

From Cisco IOS XE Release 17.7.1, you can install and boot Cisco Catalyst 8000V on OpenStack Train which acts as a hypervisor manager. The OpenStack Train release is the 20th version of the open-source cloud infrastructure software on which you can launch virtual machines (VMs) or instances.

Both 8-GB and 16-GB disks are supported for this installation. You can install Cisco Catalyst 8000V VMs in OpenStack by using one of the following methods:

- Creating a VM manually through the OpenStack dashboard and then using the qcow2 image for the installation.
- Performing the installation by using a Heat template. In OpenStack, Heat is a service that orchestrates composite cloud applications using a template format through an OpenStack-core REST API. A Heat template describes the infrastructure for a cloud application in text files. These templates specify the relationships between resources, which enable Heat to call out to the OpenStack APIs. This action creates all your infrastructure in the correct order to launch your application.

After you install and launch a Cisco Catalyst 8000V instance, based on the bootstrap or the day zero configuration data that you provide, the router either starts in the autonomous mode or the controller mode.

Features Supported

The following are the features that are supported in the Cisco Catalyst 8000V installation in OpenStack:

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

Installation Requirements for OpenStack

The requirements for installing Cisco Catalyst 8000V in OpenStack are:

- OpenStack Release: Train Release
- Red Hat Enterprise Linux (RHEL) Release 8.2 (Ootpa)

- RHEL OSP version 16.1 (Train)
- CVIM version 4.2
- Virtual disk: Both 8-GB and 16-GB virtual disks are supported
- Minimum supported profile: 1 vCPU with 4-GB memory and 8-GB or 16-GB virtual disk

Restrictions for Installing in OpenStack

Console URLs generated by OpenStack heat-deployments are subject to a token time-to-live (TTL) with a default setting of 10 minutes. Any NoVNC URLs that you use expires after this default time, especially under certain conditions, for example, with a lower profile instance bootup or while using different setups.

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

Install Cisco Catalyst 8000V in OpenStack

You can install Cisco Catalyst 8000V in one of the following ways:

- By using the OpenStack GUI. To learn how to do this, see Launching an Instance, on page 90.
- By using the heat template. To learn how to perform this installation, see Installing the VM Using a Heat Temaplate, on page 91.
- By using the CLI. You can create a VM by running the **openstack server create** command in the OpenStack CLI. For more information, see https://docs.openstack.org/python-openstackclient/train/cli/command-objects/server.html#server-create.

Launching an Instance

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

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

Step 2 In the left pane, click **Details** and specify the following details:

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

Step 3 Click **Next**.

Step 4 In the left pane, click **Source**.

Step 5 From the **Select Boot Source** drop-down field, choose either **Image**, **Instance Snapshot**, **Volume**, or **Volume Snapshot**.

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

Step 6 To delete the Volume when you delete the instance, from the **Delete Volume on Instance Delete** field, choose **Yes**.

Step 7 In the left pane, click **Flavor**.

Step 8 Choose an option based on your memory and storage requirements.

Step 9 Click **Next**.

Step 10 From the **Networks** option, choose a network to connect the Cisco Catalyst 8000V VM with the servers in that network. This option is also required if you want to set up a topology.

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

Step 11 Click **Next**.

Step 12 From the **Security Groups** drop-down list, choose the security groups to launch your instance in. A default security group is also available.

Step 13 Click **Next**.

Step 14 In the **Configuration** section, copy and paste the user data in the **Customization Script** field. The following is a sample user data configuration script:

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

Step 15 You can also upload an XML file or the iosxe_config.txt file to provide user data or configuration data. Click **Choose File** and browse to your XML or .txt file.

Note For detailed information on configuring day zero settings, see Day 0 Configuration, on page 93.

Step 16 Check the **Configuration Drive** check box and click **Next**.

Step 17 Click **Launch Instance** to launch your instance.

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

Installing the VM Using a Heat Temaplate

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

To install the OpenStack VM using a Heat template, perform the following steps:

Step 1 Log in to the OpenStack portal.

Step 2 From the menu options on top, click **Project**.

Step 3 Click **Orchestration** and select **Stack**.

Step 4 In the **Stacks** window, click **Launch Stack**.

Step 5 From the **Template Source** drop-down list, choose **File**, **URL**, or **Direct Input**, based on how you want to provide the template.

Step 6 If you chose the **File** option, click the **Choose File** option, browse to the location where you have saved your template file, upload this file, and click **Next**.

Step 7 Enter a name for your stack in the **Stack Name** field.

Step 8 To enable rollback, check the **Rollback on Failure** check box.

Step 9 Enter a password for the admin in the **Password for user "admin"** field.

Step 10 Click **Launch**.

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

CHAPTER 10

Day 0 Configuration

Cisco Catalyst 8000V supports both Cisco IOS XE and the Cisco IOS XE SD-WAN functionalities. You can access the Cisco IOS XE functionalities by booting the instance in the autonomous mode. Similarly, to access and use the Cisco SD-WAN functionalities, boot your instance in the controller mode.

The autonomous mode is the default mode in which a Cisco Catalyst 8000V instance boots up. If you are a user who wants to proceed with the day 0 configuration in the autonomous mode, refer this chapter.

Note If you wish to deploy the Cisco Catalyst 8000V instance in the controller mode, see Install and Upgrade for Cisco Catalyst 8000V Controller Mode.

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

Bootstrap Support Across Hypervisors and Clouds

The following tables provide an overview of the bootstrap support across the hypervisors and the clouds for Cisco Catalyst 8000V in the autonomous mode:

Hypervisor	iosxe_config.txt on CD-ROM	ovf-env.xml on CD-ROM	OVA Installation	Config-drive Format	Custom Data	User Data
VMware	Yes	Yes	Yes	Yes	No	No
KVM	Yes	Yes	No	Yes	No	No
AWS	No	No	No	No	Yes	Yes
Azure	No	No	No	No	Yes	Yes
GCP	No	No	No	Yes	Yes	Yes

Feature Support for Day 0 Configuration

Hypervisor	iosxe_config.txt on CD-ROM	ovf-env.xml on CD-ROM	OVA Installation	Config-drive Format	Custom Data	User Data
Raw configuration copy and paste	Yes	Yes	No	Yes	Yes	Yes
Availability of specific configuration fields	No	Yes	Yes	Yes	Yes	Yes
GUI Availability	No	No	Yes	No	No	No
Guestshell Bootstrapping	Yes; via manual IOS configuration	Yes; via manual IOS configuration	No	Yes; via manual IOS configuration	Yes	Yes; via manual IOS configuration

- Public clouds have one input mechanism through which you can provide the bootstrap information to a VM. However, on the device side, three bootstrap input formats are supported for each cloud – custom-data, user-data, and SDWAN (via the ciscosdwan_cloud_init.cfg file downloaded from vManage). For example, in AWS, you can provide the bootstrap information in any of the above-mentioned formats to the instance at launch via the EC2 user data text box or the File Upload option. Cisco Catalyst 8000V then determines and processes the configuration information that you provided.

- The custom-data and the user-data columns in the table mentioned above refer to the bootstrapping input formats and not the cloud native bootstrap input mechanisms for which they were originally named. All the public clouds support both the formats, but the custom-data format is more mature and is the recommended option for most applications.

- For private clouds, you can perform the bootstrap configuration by providing a configuration file in the `iosxe_config.txt` format or the `ovf-env.xml` format. You must upload the configuration file to the VM during Cisco Catalyst 8000V installation through an attached CD-ROM.

Prerequisites for the Day0 Configuration

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

  Do not manually edit the automatically generated config file from vManage. This might cause the controller to go out of sync and the device's first power-on and bootup might not be successful.

Restrictions for the Day Zero Configuration

- If you use the PayG licensing model, you cannot perform a mode switch as controller mode does not support the PayG licensing model.
- Only the autonomous mode supports Dual-IOSd.
- Images without payload encryption and NO-LI images are not supported in the controller mode.
- 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, Cisco Federal Licensing and Smart Licensing registration does not work. You must reregister for the licenses to work.
- When you deploy a Cisco Catalyst 8000V VM by using GUI, the order of network interfaces added to the VM may not match the order in which the interfaces are created. This is because the interface numbering order is based on the name of the driver and the PCI address. Due to this behavior, the Day Zero Configuration might be applied incorrectly for some network interfaces. If you encounter this scenario, you must manually configure the affected network interfaces after you deploy the VM.

Selecting the Bootstrapping Mechanism

Now that you know the supported bootstrap methods across the hypervisors and clouds, the next step is to decide the mechanism that you should choose to perfrom the day 0 configuration. You can configure the day 0 settings for your device by using:

- **The GUI tool**: If you have installed Cisco Catalyst 8000V on VMware, and you chose an OVA deployment, you can perform the configuration by using the OVA deployment wizard. This wizard supports the bootstrap-specific fields, and you don't have to manually create a bootstrap configuration file.
- **.txt file/.xml file**: If you are in a private cloud and you want to configure the day 0 settings through IOS configuration commands, we recommend choose the iosxe_config.txt file. This method allows you to take the CLIs that you wish to apply, paste them into a file, and provide it to the VM as a CD-ROM.
- **Custom data**: When you deploy Cisco Catalyst 8000V on AWS, Microsoft Azure, or GCP, the custom-data formatted bootstrap configuration is the recommended method. This configuration method is more functional and flexible compared to configuration by using user-data. Configuring the day 0 settings using user-data is primarily meant for users with an already established user-data deployment.

Read on to know more about each of these mechanisms in detail.

Day 0 Configuration Using .txt or .xml Files

On a new, out-of-box device, during the installation, if you want to boot up the device in the autonomous mode, you can provide the bootstrap related configuration.

In a private cloud such as KVM environment, you can perform the bootstrap configuration by providing a iosxe_config.txt file or an ovf-env.xml file. This method allows you to gather the configurations that you wish to apply via the CLI, paste them into a file, and provide this content to the VM as a CD-ROM. Depending on the hyervisor environment, the data is then used for the bootstrap configuration.

The following sections explain this bootstrap configuration method in detail:

Creating the Bootstrap File

This procedure provides the steps that you need to perform to create a bootstrap configuration file. This file, which is either in the .txt or .xml format, allows you to provide the day0 configuration for your device in a simple and flexible manner.

You can perform this procedure when you create the virtual machine in hypervisors such as KVM.

Step 1 Create the **iosxe_config.txt** or the **ovf-env.xml** file.

a) To create the isoxe_config.txt file, create a file with this name that contains the IOS conf t commands line by line.
b) To create the ovf-env.xml file, select the properties that you wish to configure from Bootstrap Properties, and place them in a file with the specified name.

Note To know more about the individual properties in the .xml file, see Bootstrap Properties, on page 96.

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

Example:

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

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

Bootstrap Properties

See the following table to know about the individual bootstrap properties using which you can create the **ovf-env.xml** file.

Table 13: Bootstrap Properties

Property	Description
console	Configures the console mode. Possible values include auto, virtual, serial.

Property	Description
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.
	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.
	Example
	`ios-config-1="username cisco priv 15 pass ciscoxyz"`
	`ios-config-2="ip scp server enable"`
	`ios-config-3="ip domain lookup"`
	`ios-config-4="ip domain name cisco.com”`

license	Configures the license technology level that is available when the Cisco Catalyst 8000V instance boots.
login-password	The login password for the router.
login-username	The user name for the router.
mgmt-interface	Designates the management interface for the Cisco Catalyst 8000V instance. The format must be GigabitEthernetx or GigabitEthernetx.xxx.
mgmt-ipv4-addr	The management gateway address/mask in the IPv4 format for the GigabitEthernet0 management interface.
mgmt-ipv4-gateway	The IPv4 management default gateway address. If you're using DHCP, enter **dhcp** in the field.
mgmt-ipv4-network	Configures the IPv4 Network (such as “192.168.2.0/24” or “192.168.2.0 255.255.255.0”) that the management gateway should route to. If this value is not specified, the default route (0.0.0.0/0) is used.
mgmt-vlan	Configures the dot1Q VLAN interface. Requires the management interface to be configured using the GigabitEthernetx.xxx format.
pnsc-agent-local-port	(Optional) Configures the Cisco Prime Network Services Controller service agent SSL port on the local Cisco Catalyst 8000V to receive policies from the service manager. This setting is used if you plan to remotely manage the Cisco Catalyst 8000V using the Cisco Prime Network Services Controller.

Property	Description
pnsc-ipv4-addr	Configures the IP address of the Cisco Prime Network Services Controller. 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. 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.

Note For a sample **ovf-env.xml** file, see Sample ovf-env.xml File, on page 99.

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
```

Sample iosxe_config.txt File for OpenStack Environment

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

Sample ovf-env.xml File

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

<Property oe:key="com.cisco.c8000v.config-version.1" oe:value="1.0"/>
        <Property oe:key="com.cisco.c8000v.domain-name.1" oe:value=""/>
        <Property oe:key="com.cisco.c8000v.enable-scp-server.1" oe:value="False"/>
        <Property oe:key="com.cisco.c8000v.enable-ssh-server.1" oe:value="False"/>
        <Property oe:key="com.cisco.c8000v.hostname.1" oe:value="lab"/>
        <Property oe:key="com.cisco.c8000v.license.1" oe:value="ax"/>
        <Property oe:key="com.cisco.c8000v.login-password.1" oe:value=""/>
        <Property oe:key="com.cisco.c8000v.login-username.1" oe:value="lab"/>
        <Property oe:key="com.cisco.c8000v.mgmt-interface.1" oe:value="GigabitEthernet1"/>
       <Property oe:key="com.cisco.c8000v.mgmt-ipv4-addr.1" oe:value="172.25.223.251/25"/>

       <Property oe:key="com.cisco.c8000v.mgmt-ipv4-gateway.1" oe:value="172.25.223.129"/>

        <Property oe:key="com.cisco.c8000v.mgmt-ipv4-network.1" oe:value=""/>
        <Property oe:key="com.cisco.c8000v.mgmt-vlan.1" oe:value=""/>
        <Property oe:key="com.cisco.c8000v.pnsc-agent-local-port.1" oe:value=""/>
        <Property oe:key="com.cisco.c8000v.pnsc-ipv4-addr.1" oe:value=""/>
        <Property oe:key="com.cisco.c8000v.pnsc-shared-secret-key.1" oe:value=""/>
        <Property oe:key="com.cisco.c8000v.privilege-password.1" oe:value=""/>
        <Property oe:key="com.cisco.c8000v.remote-mgmt-ipv4-addr.1" oe:value=""/>
        <Property oe:key="com.cisco.c8000v.resource-template.1"
oe:value="service_plane_medium"/>
        <Property oe:key="com.cisco.c8000v.ios-config-0001" oe:value="logging buffered
10000"/>
        <Property oe:key="com.cisco.c8000v.ios-config-0002" oe:value="hostname uut-ovf"/>
        <Property oe:key="com.cisco.c8000v.ios-config-0003" oe:value="ip domain-name
cisco.com"/>
        <Property oe:key="com.cisco.c8000v.ios-config-0004" oe:value="crypto key generate
rsa modulus 1024"/>
        <Property oe:key="com.cisco.c8000v.ios-config-0005" oe:value="interface
GigabitEthernet2"/>
        <Property oe:key="com.cisco.c8000v.ios-config-0006" oe:value="ip address 10.0.0.5
255.255.255.0"/>
        <Property oe:key="com.cisco.c8000v.ios-config-0007" oe:value="no shut"/>
        <Property oe:key="com.cisco.c8000v.ios-config-0008" oe:value="exit"/>
        <Property oe:key="com.cisco.c8000v.ios-config-0009" oe:value="ip route 0.0.0.0
0.0.0.0 10.0.0.1"/>
    </PropertySection>
</Environment>
```

Sample ovf-env.xml File for OpenStack

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

<Property oe:key="com.cisco.c8000v.config-version.1" oe:value="1.0"/>
<Property oe:key="com.cisco.c8000v.domain-name.1" oe:value=""/>
<Property oe:key="com.cisco.c8000v.enable-scp-server.1" oe:value="False"/>
```

```
<Property oe:key="com.cisco.c8000v.enable-ssh-server.1" oe:value="False"/>
<Property oe:key="com.cisco.c8000v.hostname.1" oe:value="lab"/>
<Property oe:key="com.cisco.c8000v.login-password.1" oe:value="lab#123"/>
<Property oe:key="com.cisco.c8000v.login-username.1" oe:value="lab"/>
<Property oe:key="com.cisco.c8000v.mgmt-interface.1" oe:value="GigabitEthernet1"/>
<Property oe:key="com.cisco.c8000v.mgmt-ipv4-addr.1" oe:value=""/>
<Property oe:key="com.cisco.c8000v.mgmt-ipv4-gateway.1" oe:value="192.168.8.1"/>
<Property oe:key="com.cisco.c8000v.mgmt-ipv4-network.1" oe:value=""/>
<Property oe:key="com.cisco.c8000v.mgmt-vlan.1" oe:value=""/>
<Property oe:key="com.cisco.c8000v.pnsc-agent-local-port.1" oe:value=""/>
<Property oe:key="com.cisco.c8000v.pnsc-ipv4-addr.1" oe:value=""/>
<Property oe:key="com.cisco.c8000v.pnsc-shared-secret-key.1" oe:value=""/>
<Property oe:key="com.cisco.c8000v.privilege-password.1" oe:value="lab#123"/>
<Property oe:key="com.cisco.c8000v.remote-mgmt-ipv4-addr.1" oe:value=""/>
<Property oe:key="com.cisco.c8000v.resource-template.1" oe:value="service-plane-medium"/>
<Property oe:key="com.cisco.c8000v.ios-config-0001" oe:value="logging buffered 10000"/>
<Property oe:key="com.cisco.c8000v.ios-config-0002" oe:value="hostname uut-ovf"/>
<Property oe:key="com.cisco.c8000v.ios-config-0003" oe:value="ip domain name cisco.com"/>
<Property oe:key="com.cisco.c8000v.ios-config-0005" oe:value="interface GigabitEthernet2"/>
<Property oe:key="com.cisco.c8000v.ios-config-0006" oe:value="ip address dhcp"/>
<Property oe:key="com.cisco.c8000v.ios-config-0007" oe:value="no shut"/>
<Property oe:key="com.cisco.c8000v.ios-config-0008" oe:value="exit"/>
<Property oe:key="com.cisco.c8000v.ios-config-0009" oe:value="ip route 0.0.0.0 0.0.0.0
192.168.8.1"/>
<Property oe:key="com.cisco.c8000v.ios-config-0010" oe:value="interface GigabitEthernet1"/>
<Property oe:key="com.cisco.c8000v.ios-config-0011" oe:value="ip address dhcp"/>
<Property oe:key="com.cisco.c8000v.ios-config-0012" oe:value="no shut"/>
 </PropertySection>
</Environment>
```

Day 0 Configuration for OVF Templates

OVF deployments with full support for Day 0 bootstrapping are only supported in VMware via the vCenter UI or the COT tool. The Day 0 configuration for Cisco Catalyst 8000V running on the ESXi hypervisor is available in Deploying the OVA to the VM, on page 52.

To know how to perform the day 0 configuration for deployments using the COT tool, see Editing the Basic Properties of Cisco Catalyst 8000V using COT, on page 55.

Day 0 Configuration Using Config-drive

Use the **--config-drive** option to specify that the configuration is loaded when Cisco Catalyst 8000V is booting. CD-ROMs and the second hard drive can also contain configuration information in the config-drive format. In either of these cases, this information is a file with contents that match the format of either the `iosxe_config.txt` file or the `ovf-env.xml` file.

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

As an XML/TXT File

In this option, you must provide the configuration file in one of the two possible formats:

- As an xml file in the ovf-env.xml file format (for OVF deployments)

- As a text file in the iosxe_config.txt file format

We strongly recommend that you use only one configuration file type, either the .txt file or the .xml file, and not both.

See the following sample configuration. Use one of these configurations to provide your configuration file in the filesystem:

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

OR

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

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

Using User Data

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

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

Day 0 Configuration Using Custom Data

After you download the Cisco Catalyst 8000V installation files and deploy the image in your environment, 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) or provide the user data to automate these processes to bring up the device into a functional state with minimal to no touch.

The Day 0 bootstrap file allows you to run Cisco IOS XE configuration commands, install Python packages in guestshell on Day0, run scripts in guestshell on Day0, and provide licensing information to boot the Cisco Catalyst 8000V instance with a desired technology package.

To launch a Cisco Catalyst 8000V instance by using custom data, perform the following steps:

Editing the Day 0 Bootstrap File

To edit the bootstrap file, configure these properties: IOS Configuration, Scripts, Script credentials, Python package, and Licensing. The properties can be placed in the bootstrap file in any order. Dependencies between the properties are noted in each of the following property descriptions. See the example bootstrap files at: https://github.com/csr1000v/customdata-examples.

After you have defined the properties of the bootstrap file, upload the file .

Configuring the IOS Configuration Property

If you want to bootstrap certain IOS configuration on Day0, configure the IOS Configuration property. See the following example:

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

After the first line that reads `Section: IOS configuration`, enter a list of Cisco IOS XE configuration commands to be run on theCisco Catalyst 8000V router.

When you run this command, the above mentioned IOS configuration is applied to the Cisco Catalyst 8000V router on Day0.

Configuring the Scripts Property

Scripts property helps you to automate your deployment and achieve other automation goals. If you want to run a python or a bash script on Day0 under guestshell context, you can achieve the same by providing the public URL and arguments of the python or the bash script in Scripts property.

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

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

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

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

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

In the second line of the property, enter the URL of the script and the script's arguments. The script can be either a python or a bash script. The script is run in guestshell in the first boot when the bootstrap file is uploaded when you create theCisco Catalyst 8000V instance.

To view more examples of the scrips, see the *Scripts* section in https://github.com/csr1000v/customdata-examples. Also refer to the following two examples:

Example 1

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

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

Example 2

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

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

Note If a script in the Scripts property requires a Python package that is not included in the standard CentOS Linux release (the CentOS Linux release that is used by the guestshell, which is currently CentOS Linux release 7.1.1503), you must include information about the Python package in the Python package property. For more information, see Configuring the Python package Property, on page 104.

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

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

If the curl command is successful, a copy of the python script is downloaded which verifies whether the URL is correct.

Configuring the Script credentials Property

If you have specified an FTP server in the Script property, and the server requires a user name and password credentials, specify the credentials using the Script credentials property. If the FTP server can be accessed anonymously, you need not use the Script credentials property.

Configure the Scripts property with a URL and parameters that match those in the Script credentials property. To configure the Script credentials property, follow the format given below:

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

Example 1

```
Section: Script credentials

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

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

Include the name of the FTP server that is also in the Scripts property. An example line in the Scripts property is: `ftp://10.11.0.4/dir1/dir2/script1.py -a arg1 -s arg2`. See example 2 in Configuring the Scripts Property, on page 102.

Configuring the Python package Property

If a Python package is required by a script in the Scripts property and is not a part of the standard CentOS Linux release 7.1.1503, you must include information about the package in the Python package property. By including the Python package property in the bootstrap file, you ensure that the Cisco Catalyst 8000V downloads and installs the required Python package before running the script that you specified in the Scripts property.

Note Cisco Catalyst 8000V supports only Python3 in guestshell.

To configure the Python package property, follow the format as specified here:

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

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

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

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

Sample Configuration (Microsoft Azure)

Example 1

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

```
Section: Python package

ncclient 0.5.2
```

Example 2

```
Section: Python package

c8000v_azure_guestshell 1.1.2 sudo {--user}
```

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

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

Sample Configuration (Google Cloud Platform)

Example 1

```
Section: Python package
```

```
ncclient 0.5.2
```

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

Example 2

```
Section: Python package
```

```
c8000v_gcp_ha 3.0.0 sudo {--user}
```

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

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

Configuring the License property

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

Enter the first line of the property: `Section: License`. Enter the second line of the property which specifies the tech level of the license, using the following format: **TechPackage:***tech_level* .

Note There must be no spaces between TechPackage: and the *tech_level*. The possible *tech_level* values include ax, security, appx, or ipbase)

tech_level must be in lowercase.

Example 1

```
Section: License
```

```
TechPackage:security
```

Providing the Day 0 Bootstrap File

Provide the Day 0 bootstrap file which creates a Cisco Catalyst 8000V VM by executing the following Azure CLI command:

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

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

See the following example:

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

When you execute this command, a Cisco Catalyst 8000V VM is created. The router is configured using the commands in the bootstrap file: "bootstrap.txt".

Use the **Cisco C8000V Settings** option to provide the custom data bootstrap config file.

For further information on managing Linux VMs, see: Tutorial: Create and Manage Linux VMs with the Azure CLI 2.0.

Verifying the Custom Data Configuration (Microsoft Azure)

After you upload the Day 0 bootstrap file, the VM is created and configuration commands are executed. Perform the following commands to verify the configuration commands of each property.

To help determine if the license property worked, in Cisco IOS XE CLI on Cisco Catalyst 8000V, enter the **show version** command. For example, you should see a reference to the security license.

To see if errors occurred after running the commands in the scripts property, look at the customdata.log file in the `/home/guestshell/customdata` directory. The *scriptname*.log file stores any output sent to STDOUT by the script.

To check if the Python property worked, enter the **pip freeze | grep***package-name* command to view the currently installed python packages. Search for the package *package-name* in which you are interested.

To check if the Cisco IOS XE commands were successful in the IOS Configuration property, enter the **show running-configuration** command. The following is a sample output for this command:

```
Router#show version
Cisco IOS XE Software, Version
Copyright (c) 1986-2020 by Cisco Systems, Inc.

Cisco IOS-XE software, Copyright (c) 2005-2020 by cisco Systems, Inc.
All rights reserved.  Certain components of Cisco IOS-XE software are
licensed under the GNU General Public License ("GPL") Version 2.0.  The
software code licensed under GPL Version 2.0 is free software that comes
with ABSOLUTELY NO WARRANTY.  You can redistribute and/or modify such
GPL code under the terms of GPL Version 2.0.  For more details, see the
documentation or "License Notice" file accompanying the IOS-XE software,
or the applicable URL provided on the flyer accompanying the IOS-XE
software.


ROM: IOS-XE ROMMON

Router uptime is 1 minute
Uptime for this control processor is 7 minutes
System returned to ROM by reload
System image file is "bootflash:packages.conf"
Last reload reason: Unknown reason



This product contains cryptographic features and is subject to United
States and local country laws governing import, export, transfer and
use. Delivery of Cisco cryptographic products does not imply
```

```
third-party authority to import, export, distribute or use encryption.
Importers, exporters, distributors and users are responsible for
compliance with U.S. and local country laws. By using this product you
agree to comply with applicable laws and regulations. If you are unable
to comply with U.S. and local laws, return this product immediately.

A summary of U.S. laws governing Cisco cryptographic products may be found at:
http://www.cisco.com/wwl/export/crypto/tool/stqrg.html

If you require further assistance please contact us by sending email to
export@cisco.com.

License Level: ipbase
License Type: N/A(Smart License Enabled)
Next reload license Level: ipbase

The current throughput level is 250000 kbps


Smart Licensing Status: Registration Not Applicable/Not Applicable

cisco C8000V (VXE) processor (revision VXE) with 2271486K/3075K bytes of memory.
Processor board ID 9MUG8CATY8R
Router operating mode: Controller-Managed
1 Gigabit Ethernet interface
32768K bytes of non-volatile configuration memory.
8106756K bytes of physical memory.
11530240K bytes of virtual hard disk at bootflash:.

Configuration register is 0x2102

[guestshell@guestshell ~]$ pip3 freeze | grep  gpg==1.10.0
gpg==1.10.0
[guestshell@guestshell ~]$
```

```
Router#show running-config
Building configuration...

Current configuration : 6982 bytes
!
! Last configuration change at 14:34:36 UTC Fri Nov 6 2020 by NETCONF
!
version 17.3
service timestamps debug datetime msec
service timestamps log datetime msec
service password-encryption
! Call-home is enabled by Smart-Licensing.
service call-home
platform qfp utilization monitor load 80
platform punt-keepalive disable-kernel-core
platform console serial
!
hostname Router
!
boot-start-marker
boot-end-marker
!
vrf definition 65528
 !
 address-family ipv4
 exit-address-family
!
no logging buffered
no logging rate-limit
!
```

```
aaa new-model
!
aaa authentication login default local
aaa authentication enable default enable
aaa authorization console
aaa authorization exec default local
!
aaa session-id common
fhrp version vrrp v3
!
no ip dhcp use class
!
no ip igmp ssm-map query dns
login on-success log
ipv6 unicast-routing
!
subscriber templating
!
multilink bundle-name authenticated
!
crypto pki trustpoint TP-self-signed-2465303444
 enrollment selfsigned
 subject-name cn=IOS-Self-Signed-Certificate-2465303444
 revocation-check none
 rsakeypair TP-self-signed-2465303444
!
crypto pki trustpoint SLA-TrustPoint
 enrollment pkcs12
 revocation-check crl
!
!
crypto pki certificate chain TP-self-signed-2465303444
 certificate self-signed 01
  30820330 30820218 A0030201 02020101 300D0609 2A864886 F70D0101 05050030
  31312F30 2D060355 04031326 494F532D 53656C66 2D536967 6E65642D 43657274
  69666963 6174652D 32343635 33303334 3434301E 170D3230 31313036 31343333
  35345A17 0D333031 31303631 34333335 345A3031 312F302D 06035504 03132649
  4F532D53 656C662D 5369676E 65642D43 65727469 66696361 74652D32 34363533
  30333434 34308201 22300D06 092A8648 86F70D01 01010500 0382010F 00308201
  0A028201 0100B02F AD33A0FF 0C50D3F2 D06CFDC6 F3CB73BB 4070D649 E07D16CE
  E6271C90 34E86882 822C8D71 E4BAC29D 85285258 51E748E1 8C9FB2C5 12242A22
  7FB71551 02CB4DBC 64089D2F 8DBB6C4A D3E2F112 8E16E71F FE70D102 F59862A3
  E920E77E 52E62E02 1979F800 3D13601F 27C42F81 483BFB34 697F1C20 3952626A
  CA1F5805 26D50A39 33F264D6 1AD485A0 8EB45882 FC97DCA2 106C8FAD 8CDBC0E6
  FF609188 B4677AB0 FBBE77F2 359EA002 E1A5D37D EA895FF3 92732A2B 63465DFD
  4A2A277C 17E7F720 2007A6B6 A7C7296F D0CD2707 8C7C9690 F86B0642 1BA9F28C
  F729157B 8C472E40 78A4E6BE 70471018 4B62EE36 48193FCA 062DB09F 38BC420B
  687E5866 DFA10203 010001A3 53305130 0F060355 1D130101 FF040530 030101FF
  301F0603 551D2304 18301680 14ABBD00 3D02C6E1 7706FA96 29B037A8 583E7B2E
  69301D06 03551D0E 04160414 ABBD003D 02C6E177 06FA9629 B037A858 3E7B2E69
  300D0609 2A864886 F70D0101 05050003 82010100 40C60BF0 2184CF86 08CACB66
  73E74D63 E87A6661 DC839037 D0DB08D0 33C4993C EC326432 E3573D1B EC3B42AF
  F410BF72 2AAB6D8F 1406B352 FE6B5365 CCA7E094 96980FC7 A4B77A02 49CB8C01
  3EC87F01 58BFEE33 0DA222DB 0A1BA130 0AC01F1F FDBF2085 D41EFA45 7A4C7F5E
  2D004D04 D11433BF 69337D90 117A86ED 2CF57A49 AD7DA227 129E53DF 55E12E03
  4D8E0097 A29DC365 11E8B386 891C310E F19EDF6D D9B3EA1E E26ABDBD EF82D8E9
  B0484E26 C0FC1D71 91B19B70 221E1A1A 090F8EA1 3A5FC4FD A4EF36CD EFD2F1F4
  6056C87D 8A76ED1A 68FB76F5 956C6B50 7EFA9D8C 90EA910F 187EBD13 0BF76E5A
  0B9CE20E AA5927C4 7AD13C28 58C6E920 76E36475
        quit
crypto pki certificate chain SLA-TrustPoint
 certificate ca 01
  30820321 30820209 A0030201 02020101 300D0609 2A864886 F70D0101 0B050030
  32310E30 0C060355 040A1305 43697363 6F312030 1E060355 04031317 43697363
  6F204C69 63656E73 696E6720 526F6F74 20434130 1E170D31 33303533 30313934
  3834375A 170D3338 30353330 31393438 34375A30 32310E30 0C060355 040A1305
  43697363 6F312030 1E060355 04031317 43697363 6F204C69 63656E73 696E6720
  526F6F74 20434130 82012230 0D06092A 864886F7 0D010101 05000382 010F0030
  82010A02 82010100 A6BCBD96 131E05F7 145EA72C 2CD686E6 17222EA1 F1EFF64D
  CBB4C798 212AA147 C655D8D7 9471380D 8711441E 1AAF071A 9CAE6388 8A38E520
  1C394D78 462EF239 C659F715 B98C0A59 5BBB5CBD 0CFEBEA3 700A8BF7 D8F256EE
  4AA4E80D DB6FD1C9 60B1FD18 FFC69C96 6FA68957 A2617DE7 104FDC5F EA2956AC
  7390A3EB 2B5436AD C847A2C5 DAB553EB 69A9A535 58E9F3E3 C0BD23CF 58BD7188
  68E69491 20F320E7 948E71D7 AE3BCC84 F10684C7 4BC8E00F 539BA42B 42C68BB7
  C7479096 B4CB2D62 EA2F505D C7B062A4 6811D95B E8250FC4 5D5D5FB8 8F27D191
  C55F0D76 61F9A4CD 3D992327 A8BB03BD 4E6D7069 7CBADF8B DF5F4368 95135E44
  DFC7C6CF 04DD7FD1 02030100 01A34230 40300E06 03551D0F 0101FF04 04030201
  06300F06 03551D13 0101FF04 05300301 01FF301D 0603551D 0E041604 1449DC85
  4B3D31E5 1B3E6A17 606AF333 3D3B4C73 E8300D06 092A8648 86F70D01 010B0500
  03820101 00507F24 D3932A66 86025D9F E838AE5C 6D4DF6B0 49631C78 240DA905
  604EDCDE FF4FED2B 77FC460E CD636FDB DD44681E 3A5673AB 9093D3B1 6C9E3D8B
  D98987BF E40CBD9E 1AECA0C2 2189BB5C 8FA85686 CD98B646 5575B146 8DFC66A8
  467A3DF4 4D565700 6ADF0F0D CF835015 3C04FF7C 21E878AC 11BA9CD2 55A9232C
  7CA7B7E6 C1AF74F6 152E99B7 B1FCF9BB E973DE7F 5BDDEB86 C71E3B49 1765308B
  5FB0DA06 B92AFE7F 494E8A9E 07B85737 F3A58BE1 1A48A229 C37C1E69 39F08678
  80DDCD16 D6BACECA EEBC7CF9 8428787B 35202CDC 60E4616A B623CDBD 230E3AFB
  418616A9 4093E049 4D10AB75 27E86F73 932E35B5 8862FDAE 0275156F 719BB2F0
  D697DF7F 28
        quit
!
license udi pid C8000V sn 9MUG8CATY8R
diagnostic bootup level minimal
memory free low-watermark processor 69848
!
!
username admin privilege 15 secret 9
$14$vKLj$yfnFjRidlKJg9.$4obKgKyy4TsoUs0sJ2t3HXPnA3XjYWRBnnYKBwVeJrw
!
redundancy
!
interface Loopback65528
 vrf forwarding 65528
 ip address 192.168.1.1 255.255.255.255
!
```

Verifying the Custom Data Configuration (Google Cloud Platform)

After you run the custom data script, the VM is created and the configuration commands are executed. To verify the same, use the following commands and scripts:

- **show version**: To help determine if the license property worked in Cisco IOS XE CLI on the Cisco Catalyst 8000V instance, enter the **show version** command. For example, the output displays a reference to the security license.

- To see if errors occurred after running commands in the scripts property, look at the `customdata.log` file in the `/bootflash/<cloud>/` directory. The *scriptname*.log file stores any output that is sent to STDOUT by the script.

- To verify whether the Python property worked, enter the `pip freeze | grep <package-name>` command from the Guestshell to view the currently installed Python packages. Here, *package-name* refers to the package that you are specifically searching for.

- To verify the Cisco IOS XE commands in the IOS Configuration property, run the **show running-configuration** command.

Day 0 Configuration in the Controller Mode

If you want to perfrom the day 0 configuration for a Cisco Catalyst 8000V in the controller (SD-WAN) mode, you must provide the contents of the ciscosdwan_cloud_init.cfg file downloaded from vManage.

If you want to switch to the Controller mode, or if you are looking to bootstrap Cisco Catalyst 8000V with the Cisco SD-WAN functionalities, see Install and Upgrade for Cisco Catalyst 8000V Controller Mode.

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

Note With the SD-WAN format configurations, if the confd cannot apply the config successfully at the first boot, the box might not have a working config at Day0. This is particularly critical in public cloud environments where SSH is necessary to login. Review the configuration carefully if you encounter issues upon provisioning.

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. This command displays the version of your instance, and the **operating device-mode** parameter displays the mode in which your Cisco Catalyst 8000V instance is running.

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
```

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.

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 Microsoft Azure or Google Cloud Platform, 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, 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. You must 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 any 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 11

Support for Security-Enhanced Linux

This chapter describes the SELinux feature, and includes the following sections:

Overview

Security-Enhanced Linux (SELinux) is a solution composed of Linux kernel security module and system utilities to incorporate a strong, flexible Mandatory Access Control (MAC) architecture into Cisco IOS-XE platforms.

SELinux provides an enhanced mechanism to enforce the separation of information, based on confidentiality and integrity requirements, which addresses threats of tampering and bypassing of application security mechanisms and enables the confinement of damage that malicious or flawed applications can cause.

Prerequisites for SELinux

There are no specific prerequisites for this feature.

Restrictions for SELinux

There are no specific restrictions for this feature.

Information About SELinux

SELinux enforces mandatory access control policies that confine user programs and system services to the minimum privilege required to perform their assigned functionality. This reduces or eliminates the ability of these programs and daemons to cause harm when compromised (for example, through buffer overflows or misconfigurations). This is a practical implementation of principle of least privilege by enforcing MAC on Cisco IOS-XE platforms. This confinement mechanism works independently of the traditional Linux access control mechanisms. SELinux provides the capability to define policies to control the access from an application process to any resource object, thereby allowing for the clear definition and confinement of process behavior.

SELinux can operate either in **Permissive mode** or **Enforcing mode** when enabled on a system.

- In Permissive mode, SELinux does not enforce the policy, and only generates system logs for any denials caused by violation of the resource access policy. The operation is not denied, but only logged for resource access policy violation.
- In Enforcing mode, the SELinux policy is enabled and enforced. It denies resource access based on the access policy rules, and generates system logs.

From Cisco IOS XE 17.13.1a, SELinux is enabled in Enforcing mode by default on supported Cisco IOS XE platforms. In the Enforcing mode, any system resource access that does not have the necessary allow policy is treated as a violation, and the operation is denied. The violating operation fails when a denial occurs, and system logs are generated. In Enforcing mode, the solution works in access-violation prevention mode.

Supported Platforms

From Cisco IOS XE 17.13.1a, SELinux is enabled on the following platforms:

- Cisco 1000 Series Aggregation Services Routers
- Cisco 1000 Series Integrated Services Routers
- Cisco 4000 Series Integrated Services Routers
- Cisco Catalyst 8000v Edge Software
- Cisco Catalyst 8200 Series Edge Platforms
- Cisco Catalyst 8300 Series Edge Platforms
- Cisco Catalyst 8500 and 8500L/8530L Series Edge Platforms
- Cisco VG Series Gateways: VG400, VG410, VG420, and VG450
- Cisco 1100 Terminal Services Gateway

Configuring SELinux

The are no additional requirements or configuration steps needed to enable or use the SELinux feature in Enforcing mode.

The following commands are introduced as part of the SELinux feature:

```
set platform software selinux {default | enforcing | permissive}
```

```
platform security selinux {enforcing | permissive}
```

```
show platform software selinux
```

Note These new commands are implemented as **service internal** commands.

Configuring SELinux (EXEC Mode)

Use the **set platform software selinux** command to configure SELinux in EXEC mode.

The following example shows SELinux configuration in EXEC mode:

```
Device# set platform software selinux ?

default  Set SELinux mode to default
enforcing  Set SELinux mode to enforcing
permissive  Set SELinux mode to permissive
```

Configuring SELinux (CONFIG Mode)

Use the **platform security selinux** command to configure SELinux in configuration mode.

The following example shows SELinux configuration in CONFIG mode:

```
Device(config)# platform security selinux

enforcing  Set SELinux policy to Enforcing mode
permissive  Set SELinux policy to Permissive mode

Device(config)# platform security selinux permissive

Device(config)#
*Oct 20 21:52:45.155: %IOSXE-1-PLATFORM: R0/0:
SELINUX_MODE_PROG: Platform Selinux confinement mode downgraded to permissive!

Device(config)#
```

Examples for SELinux

The following example shows the output for changing the mode from Enforcing to Permissive:

```
“*Oct 20 21:44:03.609: %IOSXE-1-PLATFORM: R0/0:
SELINUX_MODE_PROG: Platform Selinux confinement mode downgraded to permissive!”
```

The following example shows the output for changing the mode from Permissive to Enforcing:

```
“*Oct 20 21:44:34.160: %IOSXE-1-PLATFORM: R0/0:
SELINUX_MODE_PROG: Platform Selinux confinement mode upgraded to enforcing!”
```

Note If the SELinux mode is changed, this change is considered a system security event, and a system log message is generated.

SysLog Message Reference

Facility-Severity-Mnemonic	**%SELINUX-1-VIOLATION**
Severity-Meaning	Alert Level Log
Message	N/A
Message Explanation	Resource access was made by the process for which a resource access policy does not exist. The operation was flagged, and resource access was denied. A system log was generated with information that process resource access has been denied.
Component	SELINUX
Recommended Action	Contact Cisco TAC with the following relevant information as attachments: • The exact message as it appears on the console or in the system • Output of the **show tech-support** command (text file) • Archive of Btrace files from the box using the following command: **request platform software trace archive target <URL>** • Output of the **show platform software selinux** command

The following examples demonstrate sample syslog messages:

Example 1:

```
*Nov 14 00:09:04.943: %SELINUX-1-VIOLATION: R0/0: audispd: type=AVC
msg=audit(1699927057.934:129): avc: denied { getattr } for pid=5899 comm="ls"
path="/root/test" dev="rootfs" ino=25839
scontext=system_u:system_r:polaris_iosd_t:s0
tcontext=system_u:object_r:admin_home_t:s0 tclass=file permissive=0
```

Example 2:

```
*Nov 14 00:09:04.947: %SELINUX-1-VIOLATION: R0/0: audispd: t type=AVC
msg=audit(1699927198.486:130): avc: denied { write } for pid=6012 comm="echo"
path="/root/test" dev="rootfs" ino=25839
scontext=system_u:system_r:polaris_iosd_t:s0
tcontext=system_u:object_r:admin_home_t:s0 tclass=file permissive= 0
```

Verifying SELinux Enablement

Use the **show platform software selinux** command to view the SELinux configuration mode:

```
Device# show platform software selinux
=========================================
IOS-XE SELINUX STATUS
=========================================
SElinux Status :    Enabled
Current Mode :      Enforcing
Config file Mode :  Enforcing
```

Troubleshooting SELinux

If there is an instance of an SELinux violation on your device or network, please reach out to Cisco TAC with the following details:

- The message exactly as it appears on the console or in the system log. For example:

  ```
  device#request platform software trace archive target
      flash:selinux_btrace_logs
  ```

- Output of the **show tech-support** command (text file)
- Archive of Btrace files from the box using the following command:

  request platform software trace archive target <URL>

- Output of the **show platform software selinux** command

CHAPTER 12

Mapping the Cisco Catalyst 8000V Network Interfaces to VM Network Interfaces

This chapter specifies how you can map the logical interface on Cisco Catalyst 8000V with the vNIC and the vNIC MAC address. The sections in this chapter also tell you how you can add and delete network interfaces and the supported interface speed for Cisco Catalyst 8000V.

Supported Network Interface Speeds

This table specifies all the network interface speeds that are supported for Cisco Catalyst 8000V

Cisco IOS XE Release	Interface Speed (Mbps)
Cisco IOS XE 17.15.1a and later	Up to 40000
Cisco IOS XE 17.9.1 to Cisco IOS XE 17.14.x releases	Up to 25000
Cisco IOS XE 17.4.1 to Cisco IOS XE 17.8.x releases	Up to 10000

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.

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
--------------------------------------------------------------------------
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 on Cisco 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 .

Note Interface hot add/delete is not supported on Cisco Catalyst 8000V that operates in the Controller mode. If you need to perform interface hot add/delete, configure the reset operation in controller mode using the CLI: `request platform software sdwan config reset`.

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.

```
Router# 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
--------------------------------------------------------------------------
```

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.

```
Router# 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
--------------------------------------------------------------------------
```

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.

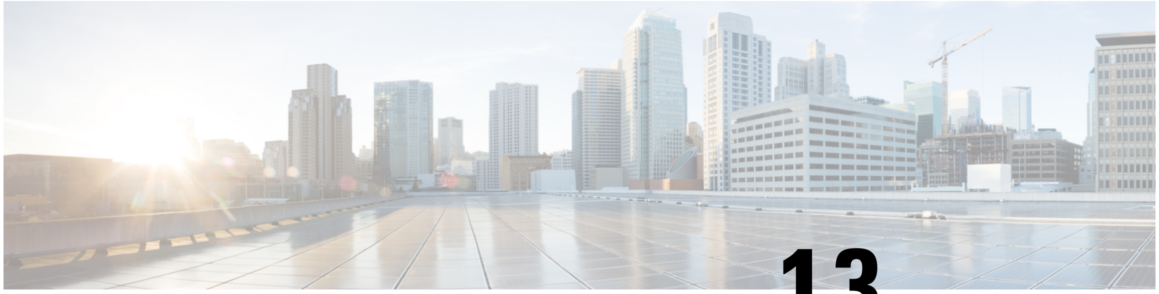
The router configuration on the cloned VM should match the configuration of the original VM.

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 13

Enabling VNF Secure Boot

Secure boot is part of the Unified Extensible Firmware Interface (**UEFI**) standard which ensures that a device boots only using a software that is trusted by the Original Equipment Manufacturer (OEM). The UEFI (Unified Extensible Firmware Interface) specification defines a secure boot methodology that prevents loading software which is not signed with an acceptable digital signature. When the device starts, the firmware checks the signature of the boot software and the operating system. If the signatures are valid, the device boots, and the firmware gives the control to the operating system.

The secure boot feature prevents malicious software applications and unauthorized operating systems from loading into the system during the system startup process. If you enable the secure boot feature, 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.

To display the system boot mode and the bootloader version, run the **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
- GRUB2 and new disk partition layout is available

Note Each hypervisor has a unique process to enable secure boot for the guest VMs. To enable secure boot, see the hypervisor specific documentation.

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 the IOS prompt is complete.
- Enable the EFI secure boot in **Edit Settings** > **VM Options** > **Boot Options** > **Secure Boot**.
- Power up the VM and the VNF boots up securely.

Important You cannot modify the firmware mode (from BIOS to EFI or vice versa) after you create the VM.

KVM Secure Boot Setup

- Create the VM.
- Power down the VM after the VM is created and the VNF IOS prompt is complete.
- Install the 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.

CHAPTER 14

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 two steps (steps 2 or 3) to select a console to view the router bootup and to access the Cisco Catalyst 8000V CLI.

Step 2 (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 you do not select any other option within the 5 second timeframe. The Cisco Catalyst 8000V instance starts the boot process.

Step 3 (Optional) Select **Serial Console**

Choose this option to use the virtual serial port console on the VM.

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

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 booted for the first time, see Changing the Console Port Access After Installation, on page 131.

The Cisco Catalyst 8000V starts the boot process.

Step 4 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.

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 5 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 the GRUB Mode, on page 202.

You can now enter the router configuration environment by entering the standard commands **enable** and then **configure terminal**.

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

You must install the software license or enable an evaluation license to obtain the supported throughput and features. Depending on the release version, you must enable the boot level or change the maximum throughput level, and reboot Cisco Catalyst 8000V.

The installed license technology package must match the package level configured with the **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.

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 Console on the Virtual Serial Port, on page 130.

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.

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 virtual**
- **platform console serial**

Example:

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

Example:

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

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

- **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.

What to do next

After you configure the console access, install the Cisco Catalyst 8000V licenses. To know how to install and use the licenses, see the *Licensing* chapter in this guide.

CHAPTER 15

Licenses and Licensing Models

This chapter provides information about the licenses that are available on Cisco Catalyst 8000 Edge Platforms Family, supported throughput options, and how to configure the available licenses and throughput. It also outlines the licensing models available on Cisco Catalyst 8000 Edge Platforms Family.

Note The information in this chapter applies predominantly to a device operating in the autonomous mode. References to the controller mode are included in certain sections for the sake of comparison and completeness. Where the information applies to controller mode, this has been called-out categorically.

For a more detailed overview on Cisco Licensing, go to https://cisco.com/go/licensingguide.

This chapter includes the following major sections:

Feature Information for Available Licenses and Licensing Models

The following table provides a summary of license related changes applicable to the Cisco Catalyst 8000 Edge Platforms Family. The table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Table 14: Feature Information for Available Licenses and Licensing Models

Feature Name	Release	Feature Information
500 Mbps Aggregate for Tier 1 and 250 Mbps Throughput Configuration in Autonomous Mode	Cisco IOS XE 17.14.1a	On virtual platforms - when you configure a throughput of 250 Mbps or T1, if an HSECK9 license is available on the device, then throughput is capped at 500 Mbps transmitted (Tx) data only. In earlier releases, throughput was capped at 200 Mbps Tx. On physical platforms - when you configure a throughput of 250 Mbps or T1, if an HSECK9 license is available on the device, then aggregate throughput throttling is effective. Throughput is capped at 500 Mbps and any distribution of traffic in the upstream and downstream direction is allowed. In earlier releases, bidirectional throughput throttling was applicable to T1 and 250 Mbps - throughput was capped at 250 Mbps in each direction. See Release-Wise Changes in Throttling Behavior, on page 144.
Aggregate Throughput Throttling - Virtual Platforms	Cisco IOS XE Cupertino 17.9.1a	On virtual platforms of the Cisco Catalyst 8000 Edge Platforms Family, *for all throughput levels*, when you configure a bidirectional throughput value on the device, aggregate throughput throttling is effective. This enhancement does not change the throttling behaviour that has always been applicable to virtual platforms: any throttling applies only to data that is transmitted (Tx). Data that is received (Rx) is unthrottled. See Throughput , on page 141 and Numeric and Tier-Based Throughput, on page 141.
Aggregate Throughput Throttling - Physical Platforms	Cisco IOS XE Cupertino 17.8.1a	On the *physical* platforms of Cisco Catalyst 8000 Edge Platforms Family, for throughput levels greater than 250 Mbps and Tier 2 and higher tiers, when you configure the bidirectional throughput value on the device, aggregate throughput throttling is effective. This means that traffic is throttled in an aggregate manner irrespective of the distribution of the traffic in the upstream and downstream direction. The bidirectional throughput is represented in the license PID (For example, DNA-C-**500M**-E-3Y and DNA-C-**T2**-E-3Y). The aggregate throughput is double the bidirectional throughput. See Release-Wise Changes in Throttling Behavior, on page 144.

Feature Name	Release	Feature Information
Tier-Based Licenses	Cisco IOS XE Cupertino 17.7.1a	Support for tier-based throughput configuration was introduced in addition to existing bandwidth-based (numeric) throughput configuration. Starting with the lowest throughput level, the available tiers are Tier 0 (T0), Tier 1 (T1), Tier 2 (T2), and Tier3 (T3). Each tier represents a throughput level. If the license PID for a product is tier-based, the license is displayed with the tier value in the CSSM Web UI. For a product with a tier-based license, you can *configure* a tier-based throughput value, and you can also *convert* to a tier-based throughput value. See Throughput , on page 141 and Numeric and Tier-Based Throughput, on page 141.
Cisco Digital Network Architecture (DNA) licenses	Cisco IOS XE Amsterdam 17.3.2	Support for Cisco DNA licenses was introduced on Cisco Catalyst 8000 Edge Platforms Family. Cisco DNA Licenses are categorised into network-stack licenses and a DNA-stack add-on licenses. See Cisco DNA License, on page 136.
High Security License (HSECK9)	Cisco IOS XE Amsterdam 17.3.2	Support for the HSECK9 license was introduced on Cisco Catalyst 8000 Edge Platforms Family. See High Security License , on page 138.
Cisco Unified Border Element license (Cisco UBE license) Cisco Unified Communications Manager Express license (Cisco Unified CME license) Cisco Unified Survivable Remote Site Telephony license (Cisco Unified SRST license)	Cisco IOS XE Amsterdam 17.3.2	Support for Cisco UBE, Cisco Unified CME, Cisco Unified SRST licenses was introduced on Cisco Catalyst 8000 Edge Platforms Family See Cisco CUBE License, on page 140, Cisco Unified CME License, on page 140, and Cisco Unified SRST License, on page 141.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Available Licenses

This section lists all the licenses that are available on Cisco Catalyst 8000 Edge Platforms Family, usage guidelines, and ordering considerations.

Cisco DNA License

A Cisco Digital Network Architecture (DNA) software license combines several feature-specific licenses.

Note A Cisco DNA license includes all feature licenses except the following: High Security (HSECK9), Cisco Unified Border Element (Cisco UBE), Cisco Unified Communications Manager Express (Cisco Unified CME), and Cisco Unified Survivable Remote Site Telephony (Cisco Unified SRST). See Ordering Considerations for a Cisco DNA License, on page 137.

Cisco DNA licenses are categorized into network-stack licenses and DNA-stack add-on licenses.

Cisco DNA Licenses Available on Catalyst 8000V Edge Software, Catalyst 8200, and 8300 Series Edge Platforms:

Network-stack licenses:

- Network Essentials
- Network Advantage: includes features available with Network Essentials, and more.
- Network Premier: includes features available Network Essentials, Network Advantage, and more.

DNA-stack add-on licenses:

- Cisco DNA Essentials: add-on license available only with Network Essentials.
- Cisco DNA Advantage: add-on license available only with Network Advantage. Includes features available with DNA Essentials and more.
- Cisco DNA Premier: add-on license available only with Network Premier. Includes features available with DNA Essentials, DNA Advantage and more.

Cisco DNA Licenses Available on Catalyst 8500 Series Edge Platforms:

Network-stack licenses:

- Network Advantage
- Network Premier: includes features available Network Advantage, and more.

DNA-stack add-on licenses:

- Cisco DNA Advantage
- Cisco DNA Premier: add-on license available only with Network Premier. Includes features available with DNA Advantage and more.

Guidelines for Using a Cisco DNA License

- Guidelines that apply to all platforms in the Cisco Catalyst 8000 Edge Platforms Family:
    - A network-stack license is a perpetual or permanent license and has no expiration date.
    - A DNA-stack add-on license is a subscription or term license and is valid only until a certain date. A 3-year and 5-year option is available for all DNA-stack add-on licenses. A 7-year subscription option is available for certain DNA-stack add-on licenses.
    - Tier 3 (T3) or higher tiers are not supported with the Network Essentials and DNA Essentials licenses.

      This also means that if you have configured T3 or higher tiers as the throughput, you cannot change the boot level license to Network Essentials and DNA Essentials.

      For information about the various tiers available with Cisco DNA Licenses, see Tier and Numeric Throughput Mapping, on page 144.

- Guidelines that apply only to Catalyst 8000V Edge Software:

  On Catalyst 8000V Edge Software, when you configure a network-stack license, you must also configure the corresponding DNA-stack add-on license.

- Guidelines that apply only to Catalyst 8200, 8300, 8500 Series Edge Platforms:
    - The DNA-stack add-on license that is available with each network-stack license is optional. You can configure a network-stack license without a DNA-stack add-on license, but you cannot configure DNA-stack add-on license without the corresponding network-stack license.
    - If you use a DNA-stack add-on license, renew the license before term expiry to continue using it, or deactivate the DNA-stack add-on license and then reload the device to continue operating with the network-stack license capabilities.

Ordering Considerations for a Cisco DNA License

A Cisco DNA license subsumes all performance, boost, and technology package licenses (securityk9, uck9, and appxk9). This means that when you order a Cisco DNA network-stack license, or a Cisco DNA-stack add-on license, if a performance, boost, and technology package license is required or applicable, it is automatically added to the order.

The license Product ID (PID) you purchase can only be a DNA-stack add-on license PID.

Even if you order a Cisco DNA license along with new hardware, the license is not preconfigured on the device. You must configure the boot level license and then the throughput, on the device.

When ordering a Cisco DNA license, you are also specifiying a throughput value. If the throughput you order is greater than 250 Mbps, an HSECK9 license is *required* on all variants of Cisco Catalyst 8000 Edge Platforms Family - except for Catalyst 8500 and 8500L/8530L Series Edge Platforms. For more information, see High Security License , on page 138.

When you order a license PID with a tier-based throughput value of *T1*, an HSECK9 license is automatically added to the order.

High Security License

The High Security license (HSECK9 license) is an export-controlled license and is restricted by U.S. export control laws. This license is required for the use of full cryptographic functionality, that is, throughput greater than 250 Mbps, and tunnel count over and above a certain number (refer to table below). This requirement applies to all devices of Cisco Catalyst 8000 Edge Platforms Family except for Catalyst 8500 and 8500L Series Edge Platforms.

Only on Catalyst 8500 and 8500L/8530L Series Edge Platforms, throughput and tunnel scale are not impacted by the non-availability of the HSECK9 license. On these platforms, the HSECK9 license is required only for compliance purposes. On all remaining models of Cisco Catalyst 8000 Edge Platforms Family, supported tunnel count and throughput are restricted in the absence of an HSECK9 license. The table below specifies supported tunnel count and supported throughput without the HSECK9 license:

PID	No. Of Tunnels *Without* HSECK9 License	Supported Throughput *Without* HSECK9 License
C8000V	150	T0, T1
C8200-1N-4T	1000	T0, T1
C8200L-1N-4T	1000	T0, T1
C8300-1N1S-4T2X	1000	T0, T1
C8300-1N1S-6T	1000	T0, T1
C8300-2N2S-4T2X	1000	T0, T1
C8300-2N2S-6T	1000	T0, T1
C8500-12X4QC	N/A	N/A
C8500-12X	N/A	N/A
C8500-20X6C	N/A	N/A
C8500L-8S4X	N/A	N/A
C8530L-8S8X4Y	N/A	N/A
C8530L-8S2X2Y	N/A	N/A

Note The term "throughput" refers to encrypted throughput on physical platforms. On virtual platforms, it refers to encrypted *and* unencrypted throughput - combined.

By using an HSECK9 license, the tunnel count restriction is lifted and you can also configure throughput greater than 250 Mbps. For detailed information about the available throughput options, see Tier and Numeric Throughput Mapping, on page 144.

To know if an HSECK9 license is being used on a device, enter the **show license summary** command in privileged EXEC mode. On all devices in the Cisco Catalyst 8000 Edge Platforms Family, the HSECK9 license as displayed as: `Router US Export Lic. for DNA (DNA_HSEC)`. For example:

```
Device# show license summary

Account Information:
  Smart Account: Eg-SA As of Dec 03 15:26:02 2021 UTC
  Virtual Account: Eg-VA

License Usage:
  License                 Entitlement Tag               Count Status
  ---------------------------------------------------------------------------
  network-advantage_T2    (NWSTACK_T2_A)                    1 IN USE
  dna-advantage_T2        (DSTACK_T2_A)                     1 IN USE
  Router US Export Lic... (DNA_HSEC)                        1 IN USE
```

Guidelines for Using an HSECK9 License

The HSECK9 license is tied to the chassis. Therefore, one HSECK9 license is required for each chassis UDI where you want to use cryptographic functionality.

An HSECK9 license requires authorization before use. This authorisation is provided by a Smart Licensing Authorization Code (SLAC). You must install a SLAC for each HSECK9 license you use. A SLAC is generated in and obtained from CSSM. How you obtain SLAC from CSSM depends on the topology you have implemented. For more information, see Installing SLAC for an HSECK9 License, on page 157.

To know if SLAC is installed, enter the **show license authorization** command in privileged exec mode, to confirm. If SLAC is installed, the status field displays: `SMART AUTHORIZATION INSTALLED on <timestamp>`. For example:

```
Device# show license authorization
Overall status:
  Active: PID:C8300-1N1S-4T2X,SN:FDO2250A0J5
      Status: SMART AUTHORIZATION INSTALLED on Dec 03 08:24:35 2021 UTC
      Last Confirmation code: 418b11b3

Authorizations:
  Router US Export Lic. for DNA (DNA_HSEC):
    Description: U.S. Export Restriction Compliance license for DNA based Routers
    Total available count: 1
    Enforcement type: EXPORT RESTRICTED
    Term information:
      Active: PID:C8300-1N1S-4T2X,SN:FDO2250A0J5
        Authorization type: SMART AUTHORIZATION INSTALLED
        License type: PERPETUAL
          Term Count: 1

Purchased Licenses:
  No Purchase Information Available
```

Ordering Considerations for an HSECK9 License

If you order your DNA licenses in the same order as Catalyst 8000 hardware platforms, the option to order an HSECK9 license is available or is selected, if applicable. For example, in case of Catalyst 8500 Series Edge Platforms, when you order hardware, an HSECK9 license is automatically added to the order, because throughput support *starts* at greater than 250 Mbps on these platforms. Further, the requisite SLAC for the HSECK9 license is also factory-installed on the device.

If you order your DNA licenses in an order that is separate from your Catalyst 8000 hardware platforms, you must separately order the HSECK9 license in the order for the Catalyst 8000 hardware platforms, if required.

If you plan to use an HSECK9 license with new hardware that you are ordering, provide your Smart Account and Virtual Account information *with* the hardware order. This enables Cisco to factory-install SLAC for the HSECK9 license on the hardware. You must still configure throughput on the device before you start using it.

Note If the HSECK9 license is ordered separately (not with the hardware order), SLAC cannot be factory-installed.

Cisco CUBE License

A Cisco Unified Border Element License (Cisco UBE license) does not require any boot level configuration before you enable it. After purchase, you can refer to the configuration guide to configure the available Cisco UBE features.

For information about the features available with a Cisco UBE license, see the *Cisco Unified Border Element Configuration Guide* for the required release at: https://www.cisco.com/c/en/us/support/unified-communications/unified-border-element/products-installation-and-configuration-guides-list.html.

For information about supported platforms and about purchasing a Cisco UBE license, see the datasheet at: https://www.cisco.com/c/en/us/products/collateral/unified-communications/unified-border-element/data-sheet-c78-729692.html. You must order a Cisco UBE license separately if required. It is not automatically included with any other license.

For information about how to report usage of a Cisco UBE license, see Smart Licensing Using Policy for Cisco Enterprise Routing Platforms. In the context of this licensing model, a Cisco UBE license is an *unenforced* license.

Cisco Unified CME License

A Cisco Unified Communications Manager Express License (Cisco Unified CME license) does not require any boot level configuration before you enable it. After purchase, you can refer to the configuration guide to configure the available features.

For information about the features available with a Cisco Unified CME license, see the Cisco Unified Communications Manager Express System Administrator Guide.

For information about supported platforms and about purchasing a Cisco Unified CME license, see the datasheet at: https://www.cisco.com/c/en/us/products/collateral/unified-communications/unified-communications-manager-express/datasheet-c78-744069.html. You must order a Cisco Unified CME license separately if required. It is not automatically included with any other license.

For information about how to report usage of a Cisco Unified CME license, see Smart Licensing Using Policy for Cisco Enterprise Routing Platforms. In the context of this licensing model, a Cisco Unified CME license is an *unenforced* license.

Cisco Unified SRST License

A Cisco Unified Survivable Remote Site Telephony License (Cisco Unified SRST license) does not require any boot level configuration before you enable it. After purchase, you can refer to the configuration guide to configure the available Unified SRST features.

For information about the features available with a Cisco Unified SRST license, see the Cisco Unified SCCP and SIP SRST System Administrator Guide (All Versions).

For information about supported platforms and about purchasing a Cisco Unified SRST license, see the datasheet at: https://www.cisco.com/c/en/us/products/collateral/unified-communications/unified-communications-manager-express/datasheet-c78-744069.html. You must order a Cisco Unified SRST license separately if required. It is not automatically included with any other license.

For information about how to report usage of a Unified SRST license, see Smart Licensing Using Policy for Cisco Enterprise Routing Platforms. In the context of this licensing model, a Unified SRST license is an *unenforced* license.

Throughput

The *throughput* tells you how much data is allowed to be transferred through the device. You configure this value in the autonomous mode. Data is then transmitted (Tx) and received (Rx) at the configured rate.

If you don't explicitly configure a throughput, default throughput is effective.

To know the configured throughput of a device, enter the applicable command:

- For physical platforms enter the **show platform hardware throughput crypto** command, in privileged EXEC mode.
- For virtual platforms enter the **show platform hardware throughput level** command, in privileged EXEC mode.

The following sections provide information about how a throughput value is represented, whether the throughput on a device refers to encrypted or unencrypted throughput and what this means, and if and how a limit may be enforced on device throughput.

Numeric and Tier-Based Throughput

The throughput you are entitled to, is specified in the device's Cisco DNA license product ID (PID). It is a value that can be represented by a number or by a tier. It is this same value that is also configured on the device.

Numeric Throughput Value

When throughput is represented by a number, it is called a numeric throughput value. For example, DNA-C-**10M**-E-3Y is a license PID with a numeric throughput value of 10M, that is, 10 Mbps.

Depending on the device, some of the other available numeric throughput values are: 15M, 25M, 50M, 100M, 250M, 500M, 1G, 2.5G, 5G, 10G, and so on. Throughput *greater* than 250 Mbps requires an HSECK9 license.

Tier-Based Throughput Value

When throughput is represented by a tier, it is called a tier-based throughput value. A tier represents a throughput level and is mapped to a numeric throughput value. For example, DNA-C-**T0**-E-3Y is a license PID with a tier-based throughput value of T0. The numeric equivalent it is mapped to is a throughput of up to 25 Mbps.

Note Tier-based throughput configuration is supported starting with Cisco IOS XE Cupertino 17.7.1a. From this release onwards, tier-based throughput configuration is also the recommended way of configuring throughput on the device.

Starting with the lowest throughput level, the available tiers are Tier 0 (T0), Tier 1 (T1), Tier 2 (T2), Tier 3 (T3), Tier 4 (T4), and Tier 5 (T5). T2 and higher tiers require an HSECK9 license.

Note the following about tiers:

- Not all tiers are available with all Cisco DNA licenses.

  For example, T3 and higher tiers are not available with the Network Essentials and DNA-Essentials licenses. This also means that if you have T3 as the configured throughput, you cannot change the boot level license to Network Essentials and DNA Essentials.

- Each tier maps to or means a different numeric value for different platforms.

  The different platforms in the Cisco Catalyst 8000 Edge Platforms Family support different maximum throughput levels. For example, T2 means 1G throughput for C8300-2N2S-4T2X, 500M for C8200-1N-4T, and 250M for C8200L-1N-4T.

To know which tiers are available with a particular DNA License and to know the numeric equivalent of each tier for a particular platform and see the Tier and Numeric Throughput Mapping, on page 144 section in this chapter.

To know when to configure a numeric throughput value and when to configure tier-based throughput on your device, see the Numeric vs. Tier-Based Throughput Configuration, on page 151 section in this chapter.

Encrypted and Unencrypted Throughput

Encrypted throughput, also known as crypto throughput, is throughput that is protected by a cryptographic algorithm.

Unencrypted throughput on the other hand, is in plain text. Unencrypted throughput is also referred to as Cisco Express Forwarding (CEF) traffic.

Important In case of physical platforms (Catalyst 8200, 8300, and 8500 Series Edge Platforms), all references to “throughput” in this document refer to cryptographic throughput.

In case of virtual platforms (Catalyst 8000V Edge Software), all references to “throughput” in this document refer to encrypted *and* unencrypted throughput, combined.

Throttled and Unthrottled Throughput

Throttled throughput, is throughput on which a limit has been enforced. (When you configure a throughput value, you are throttling device throughput to the configured extent.)

Unthrottled throughput means that no limit is enforced, and the device throughput is at the maximum capability of the device.

Note On virtual platforms, if throughput is throttled, throttling applies only to Tx data. Rx is always unthrottled. On physical platforms, if throughput is throttled, throttling applies to Tx and Rx data.

On physical platforms (Catalyst 8200, 8300, and 8500 Series Edge Platforms), unencrypted throughput (Tx and Rx), is unthrottled by default.

Types of Throttling Behavior: Aggregate and Bidirectional

The system can impose throttling in a bidirectional manner or an aggregate manner.

Bidirectional throughput throttling

Here the system throttles data in each direction. When bidirectional throttling is effective, Tx data is capped at the bidirectional throughput value and the Rx data is capped at the bidirectional throughput value - separately. (Note the exception that always applies to virtual platforms: Rx is unthrottled.)

For example, if the bidirectional throughput value is 25 Mbps or T0 and bidirectional throughput throttling is effective:

- On virtual platforms, Tx data is capped at 25 Mbps. Rx is unthrottled.
- On physical platforms, Tx data is capped at 25 Mbps and Rx data is capped at 25 Mbps.

Note The value that you see in a license PID (whether numeric or tier-based) represents a bidirectional throughput value.

Aggregate throughput throttling

Here the system doubles the configured value and throttles throughput at this aggregate limit. When aggregate throughput throttling is effective, traffic is not throttled separately in each direction.

For example, if the bidirectional throughput value that is configured is 500 Mbps and aggregate throughput throttling is effective:

- On virtual platforms, Tx data is capped at 1 Gbps. Rx is unthrottled.
- On physical platforms, traffic in the upstream and downstream direction can be any ratio within the 1 Gbps aggregate limit. For instance, 800 Mbps Tx and 200 Mbps Rx, or, 300 Mbps Tx and 700 Mbps Rx)

Release-Wise Changes in Throttling Behavior

To know if the throughput on your device will be throttled in a bidirectional manner or in an aggregate manner, check the software version running on the device, and refer to the release-wise changes in throttling behavior described below.

- **Until Cisco IOS XE Cupertino 17.7.x**: Only bidirectional throughput throttling is effective. This applies to physical and virtual platforms.
- **Starting with Cisco IOS XE Cupertino 17.8.1a**:
  - Only on physical platforms, when you configure a *throughput value greater than 250 Mbps* or T2 and higher tiers, aggregate throughput throttling is effective.

    On C8200L-1N-4T, if you configure a numeric value of 250 Mbps, bidirectional throughput throttling is effective and a maximum of 250 Mbps is available in each direction. But if you configure tier T2, aggregate throttling is effective and 500 Mbps is available for use in any Tx and Rx ratio.
  - On virtual platforms, Tx throttling continues to apply, and Rx continues to remain unthrottled.
- **Starting with Cisco IOS XE Cupertino 17.9.1a**: On virtual platforms, for all throughput levels and all tiers, aggregate throughput throttling is effective.

Note If the aggregate for the throughput level you configure on a virtual platform amounts to *greater than 250 Mbps*, aggregate throughput throttling is not effective unless an HSECK9 license is available on the device (that is, SLAC is installed).

- **Starting with Cisco IOS XE 17.14.1a**: On physical and virtual platforms, when you configure a throughput of 250 Mbps or T1, aggregate throughput throttling is effective - as long as an HSECK9 license is available on the device. On virtual platforms, this means that Tx throughput is capped at 500 Mbps. On physical platforms, this means an aggregate limit of 500 Mbps is available for use in any Tx and Rx ratio.

  If an HSECK9 license is not available on the device and you configure a throughput value of 250 Mbps, or T1, then bidirectional throughput throttling is effective. On virtual platforms this means Tx throughput is throttled at 250 Mbps. On physical platforms throughput is throttled at 250 Mbps in each direction.

Tier and Numeric Throughput Mapping

The following tables provide information about about the numeric equivalent of each tier, and the DNA licenses that each tier is available with.

Tip The mapping tables clarify only the numeric equivalent of a tier. This mapping does not reflect the final throughput that you are entitled to. The entitled throughput depends on the device's capability, the software version running on the device, and throttling behavior for that version.

Note When you purchase a license PID with a tier-based throughput value of *T1*, an HSECK9 license is automatically provided.

Y: Network Premium and DNA Premium

Y: Network Advantage and DNA Advantage

Y: Network Essentials and DNA Essentials

* = HSECK9 license required. On C8500 and C8500L/C8530L, the HSECK9 license is required for compliance purposes only.

Table 15: Tier and Numeric Throughput Mapping for Virtual Platforms (C8000v)

Tiers from 17.9.1a:	**T0**		**T1**		**T2***			**T3***			**T4***
Tiers in 17.7.x, 17.8.x:	**T0**	**T1**			**T2***			**T3***			**T4***
Numeric Mapping:	**15M**	**25M**	**50M**	**100M**	**250M**	**500M**	**1G**	**2.5G**	**5G**	**10G**	**Unthrottled**
Available DNA Licenses:	Y Y Y	Y Y Y	Y Y Y	Y Y Y	Y Y Y	Y Y Y	Y Y Y	Y Y	Y Y	Y Y	Y Y

Table 16: Tier and Numeric Throughput Mapping for Physical Platforms (C8200, C8300, C8500)

Tiers from 17.8.1a:	T0			T1		T2*			T3*			T4*	T5*
Tiers in 17.7.x:	T0		T1			T2*			T3*			n.a.	n.a.
Configured Numeric Value:	**10M**	**15M**	**25M**	**50M**	**100M**	**250M**	**500M**	**1G**	**2.5G**	**5G**	**10G**	**50G**	**Unthrottled**
C8200-1N-4T	Y Y Y	Y Y Y	Y Y Y	Y Y Y	Y Y Y	Y Y Y	Y Y Y						
C8200L-1N-4T	Y Y Y	Y Y Y	Y Y Y	Y Y Y	Y Y Y	Y Y Y							
C8300-1N1S-4T2X	Y Y Y	Y Y Y	Y Y Y	Y Y Y	Y Y Y	Y Y Y	Y Y Y	Y Y Y	Y Y				
C8300-1N1S-6T	Y Y Y	Y Y Y	Y Y Y	Y Y Y	Y Y Y	Y Y Y	Y Y Y	Y Y Y					
C8300-2N2S-4T2X	Y Y Y	Y Y Y	Y Y Y	Y Y Y	Y Y Y	Y Y Y	Y Y Y	Y Y Y	Y Y				
C8300-2N2S-6T	Y Y Y	Y Y Y	Y Y Y	Y Y Y	Y Y Y	Y Y Y	Y Y Y	Y Y Y					
C8500-12X									Y Y	Y Y	Y Y		
C8500-12X4QC									Y Y	Y Y	Y Y		
C8500-20X6C												Y Y	Y Y

C8500L-8S4X								Y Y	Y Y	Y Y	Y Y		
C8530L-8S8X4Y								Y Y	Y Y	Y Y	Y Y		
C8530L-8S2X2Y								Y Y	Y Y	Y Y	Y Y		

Entitled Throughput and Throttling Specifications in the Autonomous Mode

These tables tell you about the throughput you are entitled to. This is based on the device, the throughput value, which can be aggregate or numeric, and the release, which determines if throttling is imposed in an aggregate or bidirectional manner.

Table 17: C8000v

Throughput = Encrypted and Unencrypted Throughput Rx is Unthrottled * HSECK9 license is required.				
Supported Throughput Values **(default 10M)**	**Entitled Throughput & Throttling in >= 17.4.1a**	**Entitled Throughput & Throttling in >= 17.7.1a**	**Entitled Throughput & Throttling in >=17.9.1a**	**Entitled Throughput & Throttling in >=17.14.1a**
10M	10M Tx Only	10M Tx Only	20M Tx Only	20M Tx Only
15M	15M Tx Only	15M Tx Only	30M Tx Only	30M Tx Only
25M	25M Tx Only	25M Tx Only	50M Tx Only	50M Tx Only
50M	50M Tx Only	50M Tx Only	100M Tx Only	100M Tx Only
100M	100M Tx Only	100M Tx Only	200M Tx Only	200M Tx Only
250M	250M Tx Only	250M Tx Only	250M Tx Only	With HSECK9: 500M Tx Without HSECK9: 250M Tx
500M*	500M Tx Only	500M Tx Only	1G Tx Only	1G Tx Only
1G*	1G Tx Only	1G Tx Only	2G Tx Only	2G Tx Only
2.5G*	2.5G Tx Only	2.5G Tx Only	5G Tx Only	5G Tx Only
5G*	5G Tx Only	5G Tx Only	10G Tx Only	10G Tx Only
10G*	10G Tx Only	10G Tx Only	20G Tx Only	20G Tx Only
T0	-	15M Tx Only	50M Tx Only	50M Tx Only
T1	-	100M Tx Only	200M Tx Only	With HSECK9: 500M Tx Without HSECK9: 250M Tx
T2*	-	1G Tx Only	2G Tx Only	2G Tx Only

T3*	-	10 Tx Only	20G Tx Only	20G Tx Only
T4*	-	Unthrottled	Unthrottled	Unthrottled

Table 18: C8200-1N-4T

Throughput = Encrypted Throughput * HSECK9 license is required.				
Supported Throughput Values **(default 10M)**	**Entitled Throughput & Throttling in >= 17.4.1a**	**Entitled Throughput & Throttling in >= 17.7.1a**	**Entitled Throughput & Throttling in >= 17.8.1a**	**Entitled Throughput & Throttling in >= 17.14.1a**
10M	10M Bidirectional	10M Bidirectional	10M Bidirectional	10M Bidirectional
15M	15M Bidirectional	15M Bidirectional	15M Bidirectional	15M Bidirectional
25M	25M Bidirectional	25M Bidirectional	25M Bidirectional	25M Bidirectional
50M	50M Bidirectional	50M Bidirectional	50M Bidirectional	50M Bidirectional
100M	100M Bidirectional	100M Bidirectional	100M Bidirectional	100M Bidirectional
250M	250M Bidirectional	250M Bidirectional	250M Bidirectional	With HSECK9: 500M Aggregate Without HSECK9: 250M Bidirectional
500M*	500M Bidirectional	500M Bidirectional	1G Aggregate	1G Aggregate
T0	-	15M Bidirectional	25M Bidirectional	25M Bidirectional
T1	-	100M Bidirectional	100M Bidirectional	With HSECK9: 500M Aggregate Without HSECK9: 250M Bidirectional
T2	-	500M Bidirectional	1G Aggregate	1G Aggregate

Table 19: C8200L-1N-4T

Throughput = Encrypted Throughput * HSECK9 license is required.				
Supported Throughput Values **(default 10M)**	**Entitled Throughput & Throttling in >= >= 17.5.1a**	**Entitled Throughput & Throttling in >= 17.7.1a**	**Entitled Throughput & Throttling in >= 17.8.1a**	**Entitled Throughput & Throttling in >= 17.14.1a**
10M	10M Bidirectional	10M Bidirectional	10M Bidirectional	10M Bidirectional
15M	15M Bidirectional	15M Bidirectional	15M Bidirectional	15M Bidirectional

25M	25M Bidirectional	25M Bidirectional	25M Bidirectional	25M Bidirectional
50M	50M Bidirectional	50M Bidirectional	50M Bidirectional	50M Bidirectional
100M	100M Bidirectional	100M Bidirectional	100M Bidirectional	100M Bidirectional
250M	250M Bidirectional	250M Bidirectional	250M Bidirectional	With HSECK9: 500M Aggregate Without HSECK9: 250M Bidirectional
T0	-	15M Bidirectional	25M Bidirectional	25M Bidirectional
T1	-	100M Bidirectional	100M Bidirectional	With HSECK9: 500M Aggregate Without HSECK9: 250M Bidirectional
T2*	-	250M Bidirectional	500M Aggregate	500M Aggregate
-			**Note** From 17.8.1a, On C8200-1N-4T-L, if you configure a numeric value of 250 Mbps, a maximum of 250 Mbps is available in each direction. But if you configure tier-based value T2 (which requires an HSECK9 license), 500 Mbps is available for use in any Tx and Rx ratio.	

Table 20: C8300-1N1S-4T2X, C8300-2N2S-4T2X

Throughput = Encrypted Throughput * HSECK9 license is required.				
Supported Throughput Values **(default 10M)**	**Entitled Throughput & Throttling in >= 17.3.2**	**Entitled Throughput & Throttling in >= 17.7.1a**	**Entitled Throughput & Throttling in >= 17.8.1a**	**Entitled Throughput & Throttling in >= 17.14.1a**
10M	10M Bidirectional	10M Bidirectional	10M Bidirectional	10M Bidirectional
15M	15M Bidirectional	15M Bidirectional	15M Bidirectional	15M Bidirectional
25M	25M Bidirectional	25M Bidirectional	25M Bidirectional	25M Bidirectional
50M	50M Bidirectional	50M Bidirectional	50M Bidirectional	50M Bidirectional
100M	100M Bidirectional	100M Bidirectional	100M Bidirectional	100M Bidirectional
250M	250M Bidirectional	250M Bidirectional	250M Bidirectional	With HSECK9: 500M Aggregate Without HSECK9: 250M Bidirectional
500M*	500M Bidirectional	500M Bidirectional	1G Aggregate	1G Aggregate

1G*	1G Bidirectional	1G Bidirectional	2G Aggregate	2G Aggregate
2.5G*	2.5G Bidirectional	2.5G Bidirectional	5G Aggregate	5G Aggregate
T0	-	15M Bidirectional	25M Bidirectional	25M Bidirectional
T1	-	100M Bidirectional	100M Bidirectional	With HSECK9: 500M Aggregate Without HSECK9: 250M Bidirectional
T2*	-	1G Bidirectional	2G Aggregate	2G Aggregate
T3*	-	10G Bidirectional	20G Aggregate	20G Aggregate

Table 21: C8300-1N1S-6T, C8300-2N2S-6T

Throughput = Encrypted Throughput * HSECK9 license is required.				
Supported Throughput Values **(default 10M)**	**Entitled Throughput & Throttling in >= 17.3.2**	**Entitled Throughput & Throttling in >= 17.7.1a**	**Entitled Throughput & Throttling in >= 17.8.1a**	**Entitled Throughput & Throttling in >= 17.14.1a**
10M	10M Bidirectional	10M Bidirectional	10M Bidirectional	10M Bidirectional
15M	15M Bidirectional	15M Bidirectional	15M Bidirectional	15M Bidirectional
25M	25M Bidirectional	25M Bidirectional	25M Bidirectional	25M Bidirectional
50M	50M Bidirectional	50M Bidirectional	50M Bidirectional	50M Bidirectional
100M	100M Bidirectional	100M Bidirectional	100M Bidirectional	100M Bidirectional
250M	250M Bidirectional	250M Bidirectional	250M Bidirectional	With HSECK9: 500M Aggregate Without HSECK9: 250M Bidirectional
500M*	500M Bidirectional	500M Bidirectional	1G Aggregate	1G Aggregate
1G*	1G Bidirectional	1G Bidirectional	2G Aggregate	2G Aggregate
T0	-	15M Bidirectional	25M Bi-directional	25M Bi-directional
T1	-	100M Bidirectional	100M Bi-directional	With HSECK9: 500M Aggregate Without HSECK9: 250M Bidirectional
T2*	-	1G Bidirectional	2G Aggregate	2G Aggregate

Table 22: C8500-12X, C8500-12X4QC

Throughput = Encrypted Throughput *HSECK9 license required for compliance purposes only.			
Supported Throughput Values **(default 10M)**	**Entitled Throughput & Throttling in >= 17.3.2**	**Entitled Throughput & Throttling in >= 17.7.1a**	**Entitled Throughput & Throttling in >= 17.8.1a**
2.5G*	2.5G Bidirectional	2.5G Bidirectional	5G Aggregate
5G*	5G Bidirectional	5G Bidirectional	10G Aggregate
10G*	10G Bidirectional	10G Bidirectional	20G Aggregate
T3*	-	10G Bidirectional	20G Aggregate

Table 23: C8500L-8S4X

Throughput = Encrypted Throughput *HSECK9 license required for compliance purposes only.			
Supported Throughput Values **(default 10M)**	**Entitled Throughput & Throttling in >= 17.4.1a**	**Entitled Throughput & Throttling in >= 17.7.1a**	**Entitled Throughput & Throttling in >= 17.8.1a**
1G*	1G Bidirectional	1G Bidirectional	2G Aggregate
2.5G*	2G Bidirectional	2G Bidirectional	5G Aggregate
5G*	5G Bidirectional	5G Bidirectional	10G Aggregate
10G*	10G Bidirectional	10G Bidirectional	20G Aggregate
T2*	-	1G Bidirectional	2G Aggregate
T3*	-	10G Bidirectional	20G Aggregate

Table 24: C8500-20X6C

Throughput = Encrypted Throughput *HSECK9 license required for compliance purposes only.	
Supported Throughput Values (default T4)	**Entitled Throughput and Throttling in >= 17.10.1a**
T4*	50G Aggregate
T5*	Unthrottled

Entitled Throughput and Throttling Specifications in the SD-WAN Controller Mode

PID	Introductory Release for PID	Throughput Without HSECK9 - Bi-directional	Throughput With HSECK9 (>=17.3.2 and <17.8.1a, Bi-directional)	Throughput With HSECK9 (>17.8.1a, Aggregate)
C8300-1N1S-4T2X (default 250M)	17.3.2	250M	unthrottled	unthrottled
C8300-2N2S-6T (default 250M)	17.3.2	250M	1G	2G
C8300-1N1S-6T (default 250M)	17.3.2	250M	1G	2G
C8300-2N2S-4T2X (default 250M)	17.3.2	250M	unthrottled	unthrottled
C8200-1N-4T (default 250M)	17.4.1a	250M	500M	1G
C8200L-1N-4T (default 250M)	17.5.1a	250M	250M	500M
C8500-12X4QC (default unthrottled)	17.3.2	unthrottled	unthrottled	unthrottled
C8500-12X (default unthrottled)	17.3.2	unthrottled	unthrottled	unthrottled
C8500L-8S4X (default unthrottled)	17.4.1a	unthrott led	unthrottled	unthrottled
C8500-20X6C (default T4)	17.10.1a	unthrottled	-	unthrottled
C8000v (default 250M)	17.4.1a	250M	unthrottled	unthrottled

Numeric vs. Tier-Based Throughput Configuration

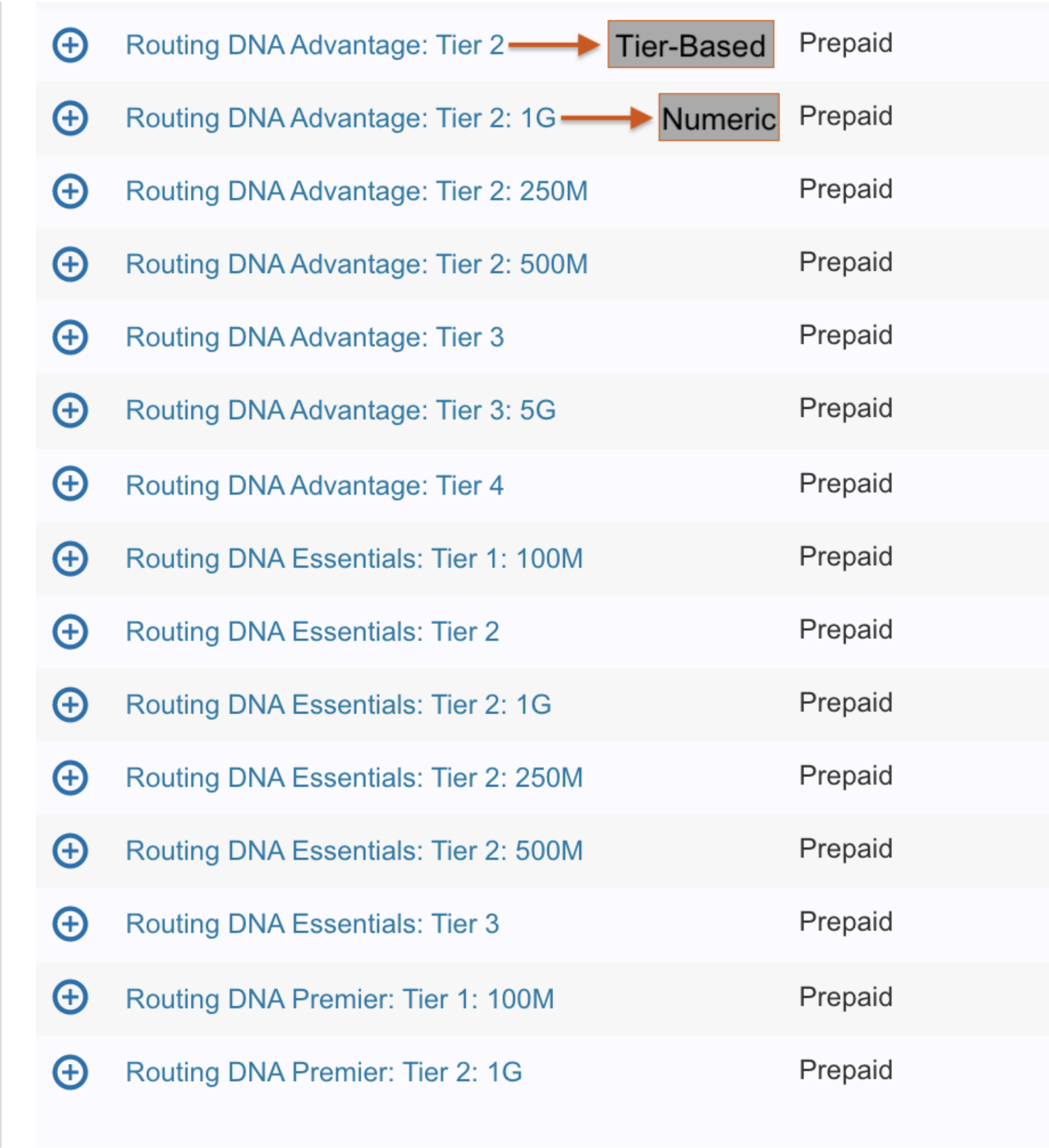
With the introduction of tier-based throughput configuration in Cisco IOS XE Cupertino 17.7.1a, when you configure throughput on the device, both numeric and tier-based options are available. This section provides information about when to configure a numeric throughput value and when to configure tier-based throughput.

Identifying whether you have tier-based or numeric licenses

Cisco Smart Software Manager (CSSM) is a portal that enables you to manage all your Cisco software licenses. All the license PIDs you purchase are listed in the CSSM Web UI at: https://software.cisco.com → **Manage licenses**. One way of identifying whether you have a tier-based or numeric licenses is to see how the license is displayed in CSSM.

To do this, log in to the portal and in the corresponding Smart Account and Virtual Account, navigate to **Inventory** > **Licences**, to display the licenses in the account. The screenshot below shows you how both are displayed:

Figure 1: Numeric and Tier Values Displayed in the CSSM Web UI



Routing DNA Advantage: Tier 2	Tier-Based	Prepaid
Routing DNA Advantage: Tier 2: 1G	Numeric	Prepaid
Routing DNA Advantage: Tier 2: 250M		Prepaid
Routing DNA Advantage: Tier 2: 500M		Prepaid
Routing DNA Advantage: Tier 3		Prepaid
Routing DNA Advantage: Tier 3: 5G		Prepaid
Routing DNA Advantage: Tier 4		Prepaid
Routing DNA Essentials: Tier 1: 100M		Prepaid
Routing DNA Essentials: Tier 2		Prepaid
Routing DNA Essentials: Tier 2: 1G		Prepaid
Routing DNA Essentials: Tier 2: 250M		Prepaid
Routing DNA Essentials: Tier 2: 500M		Prepaid
Routing DNA Essentials: Tier 3		Prepaid
Routing DNA Premier: Tier 1: 100M		Prepaid
Routing DNA Premier: Tier 2: 1G		Prepaid

Recommendations for whether to configure a numeric or tier-based throughput value

- If you purchase a numeric license PID, the license is displayed with the numeric throughput value *and* tier-based value in the CSSM Web UI. For such a license, we recommend that you configure only a numeric throughput value.

  See Configuring a Numeric Throughput, on page 158.

- If you purchase a tier-based license PID, the license is displayed with only the tier value in the CSSM Web UI. For such a license, you can either configure a tier-based throughput value to match the display in the CSSM Web UI, or you can configure a numeric throughput value.

  See Configuring a Tier-Based Throughput, on page 160 or Configuring a Numeric Throughput, on page 158.

Note There is no functional impact if you have tier-based license PID in CSSM and you configure a numeric throughput value on the device.

When to *convert* the configured value to a numeric or tier-based one

The following scenarios further clarify when you can *convert* from numeric to tier-based throughput configuration, or from tier-based throughput configuration to numeric, when conversion is required, and when it is optional:

- You have configured a numeric throughput value on the device and the license PID is a numeric license: *You must not* convert to tier-based throughput value.
- You have configured a numeric throughput value on the device and the license PID is a tier-based license: You can convert the throughput configuration to tier-based value - but this is optional. There is no functional impact if you do not convert to a tier-based throughput value.

  If you want to convert to a tier-based value, see Converting From a Numeric Throughput Value to a Tier, on page 165
- You are upgrading to a release where tier-based throughput values are supported and the license PID is tier-based: You can convert the throughput to tier-based value after upgrade - but this is optional. There is no functional impact if you do not convert to a tier-based throughput value.

  See Upgrading from a Release Supporting Numeric Throughput to a Release Supporting Tiers, on page 167.
- You are upgrading to a release where tier-based throughput values are supported, and your license PID is numeric: *You must not* convert to a tier-based throughput value.
- You are downgrading to a release where only numeric throughput values are supported and your license PID and throughput configuration are tier-based: *You must* change configuration to a numeric throughput value, *before you downgrade.*

  See Downgrading from a Release Supporting Tiers to a Release Supporting Only Numeric Throughput, on page 168.

How to Configure Available Licenses and Throughput

This section provides information about the sequence in which you must complete tasks, for the various licenses available on the Cisco Catalyst 8000 Edge Platforms Family - before you can start using them.

For a Cisco DNA license: **Configure a Boot Level License** → **Configure Numeric or Tier-Based Throughput** → **Implement a Smart Licensing Using Policy Topology** → **Report License Usage (If Applicable)**.

For an HSECK9 license: **Configure a Boot Level License** → **Implement a Smart Licensing Using Policy Topology** → **Install SLAC**[1] → **Enable HSECK9 on applicable platforms**[2]→ **Configure Numeric or Tier-Based Throughput** → **Report License Usage (If Applicable)**.

For a Cisco UBE, or Cisco Unified CME, or Cisco Unified SRST license: **Implement a Smart Licensing Using Policy Topology** → **Report License Usage (If Applicable)**.

Configuring a Boot Level License

If you have purchased a Cisco DNA license for a new device, or if you have an existing device and you want to change (upgrade or downgrade, add or remove) the currently configured license on your device, complete the following task.

This sets a boot level license and requires a reload before the configured changes are effective.

Step 1 **show version**

Displays the currently set boot level license.

In the accompanying example, Network Advantage and DNA Advantage licences are configured on the device.

Example:

```
Device# show version
<output truncated>
Technology Package License Information:

-----------------------------------------------------------------
Technology     Type         Technology-package Technology-package
                            Current            Next Reboot
-----------------------------------------------------------------
Smart License  Perpetual    network-advantage  network-advantage
Smart License  Subscription dna-advantage      dna-advantage

<output truncated>
```

Step 2 **configure terminal**

Enters global configuration mode.

Example:

```
Device# configure terminal
```

Step 3 Depending on whether the device is a physical or virtual one, configure the applicable command:

- For physical platforms: [**no**] **license boot level** {**network-advantage** [**addon dna-advantage**] | **network-essentials** [**addon dna-essentials**] | **network-premier** [**addon dna-premier**] }
- For virtual platforms: [**no**] **license boot level** {**network-advantage** {**addon dna-advantage**} | **network-essentials** {**addon dna-essentials**} | **network-premier** {**addon dna-premier**} }

Sets a boot level license.

On all platforms, first configure a network-stack license. Only after this can you configure the corresponding add-on license.

[1] If a SLAC has been factory-installed by Cisco (in case of new hardware), skip this step

[2] Enter the **license feature hseck9** command in global configuration mode for Catalyst 8200, and 8300 Series Edge Platforms only.

In the command syntax note how the configuration of a DNA-stack add-on license is optional on physical platforms, but mandatory on virtual platforms.

The accompanying example, shows configuration on a C8300-1N1S-4T2X router, which is a physical platform. The network-stack license, Network Premier and the corresponding add-on license, DNA-Premier are configured.

Example:

```
Device(config)# license boot level network-premier addon dna-premier
% use 'write' command to make license boot config take effect on next boot
```

Step 4 **exit**

Exits global configuration mode and returns to privileged EXEC mode.

Example:

```
Device# exit
```

Step 5 **copy running-config startup-config**

Saves your entries in the configuration file.

Example:

```
Device# copy running-config startup-config
Destination filename [startup-config]?
Building configuration...
[OK]
<output truncated>
```

Step 6 **reload**

Reloads the device. License levels configured in Step 3 are effective and displayed only after this reload.

Example:

```
Device# reload
Proceed with reload? [confirm]

*Dec  8 01:04:12.287: %SYS-5-RELOAD: Reload requested by console.
Reload Reason: Reload Command.
<output truncated>
```

Step 7 **show version**

Displays the currently set boot level license.

In the accompanying example, the output confirms that Network Premier and DNA-Premier licenses are configured.

Example:

```
Device# show version
<output truncated>
Technology Package License Information:

-----------------------------------------------------------------
Technology     Type         Technology-package Technology-package
                            Current            Next Reboot
-----------------------------------------------------------------
Smart License  Perpetual    network-premier    network-premier
Smart License  Subscription dna-premier        dna-premier
<output truncated>
```

Step 8 **show license summary**

Displays a summary of license usage, which includes information about licenses being used, the count, and status.

Example:

```
Device# show license summary

Account Information:
  Smart Account: Eg-SA As of Dec 08 08:10:33 2021 UTC
  Virtual Account: Eg-VA

License Usage:
  License                 Entitlement Tag               Count Status
  -----------------------------------------------------------------------------
  network-premier_T2      (NWSTACK_T2_P)                    1 IN USE
  dna-premier_T2          (DSTACK_T2_P)                     1 IN USE
```

Step 9 Complete usage reporting - if required

After you configure a license level, you may have to send a RUM report (Resource Utilization Measurement Report) to CSSM to report license usage information. To know if reporting is required, you can wait for a system message or refer to the policy using show commands.

- The system message, which indicates that reporting is required: %SMART_LIC-6-REPORTING_REQUIRED: A Usage report acknowledgement will be required in [dec] days. [dec] is the amount of time (in days) left to meet reporting requirements.
- If using **show** commands, refer to the output of the **show license status** privileged EXEC command and check the `Next ACK deadline` field. This means a RUM report must be sent and the acknolwedgement (ACK) from CSSM must be installed by this date.

How you send the RUM report, depends on the topology you have implemented in the Smart Licensing Using Policy environment. For more information, see How to Configure Smart Licensing Using Policy: Workflows by Topology.

Installing SLAC for an HSECK9 License

A Smart Licensing Authorization Code (SLAC) is generated in and obtained from Cisco Smart Software Manager (CSSM) portal.

There are multiple ways in which a product may be connected to the CSSM, in order to obtain a SLAC. Each way of connecting to CSSM is called a topology. You must implement one of the supported topologies so you can then install SLAC in the corresponding method.

For information about all the methods, see the Supported Topologies section of the Smart Licensing Using Policy for Cisco Enterprise Routing Platforms document.

Note Ensure that a boot level license is already configured on the device. See Configuring a Boot Level License, on page 155. In the output of the show version privileged EXEC command ensure that a license is mentioned in the `License Level` field.

Required Tasks After Installing SLAC

Complete the following required tasks after installing SLAC - only if applicable to the platform:

Platform	Required Tasks After Installing SLAC
For Catalyst 8200 and 8300 Series Edge Platforms	Enter the **license feature hseck9** command in global configuration mode. This *enables* the HSECK9 license on these platforms.
For the *C8500L/C8530L* models of the Catalyst 8500 Series Edge Platforms	Reload the device after installing SLAC.

Configuring a Numeric Throughput

This task shows you how to change the numeric throughput level on physical and virtual platforms. If you do not configure a throughput level, the platform's default throughput level is effective.

Configuration of a throughput level requires a reload on physical platforms (Catalyst 8200, 8300, and 8500 Series Edge Platforms). A reload is not required for virtual platforms (Catalyst 8000V Edge Software).

Before you begin

- Read sections Numeric and Tier-Based Throughput, on page 141 and Numeric vs. Tier-Based Throughput Configuration, on page 151.
- Ensure that a boot level license is already configured on the device. Otherwise you will not be able to configure a throughput value. See Configuring a Boot Level License, on page 155. In the output of the **show version** privileged EXEC command ensure that a license is mentioned in the `License Level` field.
- If you are configuring throughput greater than 250 Mbps, you must install a Smart Licensing Authorization Code (SLAC) before you start with this task. See Installing SLAC for an HSECK9 License, on page 157.
- You can configure the `250M` value with or without an HSECK9 license. The system allows both. The difference is that aggregate throttling is effective if HSECK9 is available on the device. See: Release-Wise Changes in Throttling Behavior, on page 144.
- Note the throughput you are entitled to. This is indicated in the Cisco DNA license PID you purchase.

Step 1 Depending on whether the device is a physical or virtual one, enter the applicable command:

- For physical platforms: **show platform hardware throughput crypto**
- For virtual platforms: **show platform hardware throughput level**

Displays the current throughput level on the device.

In the accompanying example:

- The **show platform hardware throughput crypto** sample output is of a physical platform (a C8300-2N2S-4T2X). Here the throughput level is throttled at 250M.
- The **show platform hardware throughput level** sample output is of a virtual platform (a C8000V).

Example:

```
Device# show platform hardware throughput crypto
Current configured crypto throughput level: 250M
     Level is saved, reboot is not required
Current enforced crypto throughput level: 250M
```

```
Crypto Throughput is throttled at 250M
Default Crypto throughput level: 10M
Current boot level is network-advantage

OR

Device# show platform hardware throughput level
The current throughput level is 1000000 kb/s
```

Step 2 **configure terminal**

Enters global configuration mode.

Example:

```
Device# configure terminal
```

Step 3 Depending on whether the device is a physical or virtual one, configure the applicable command:

- For physical platforms: **platform hardware throughput crypto** {**100M** | **10M** | **15M** | **1G** | **2.5G** | **250M** | **25M** | **500M** | **50M**}
- For virtual platforms: **platform hardware throughput level MB** {**100** | **1000** | **10000** | **15** | **25** | **250** | **2500** | **50** | **500** | **5000**}

Configures the throughput level. The displayed throughput options depend on the device.

Note On physical and virtual platforms, ensure that a boot level license is configured. Otherwise the command is not recognized as a valid one on the command line interface.

In the accompanying example:

- 1 Gbps is configured on the physical platform. The software version running on the device is Cisco IOS XE Cupertino 17.8.1a and this means aggregate throughput throttling applies. After reload, the sum of upstream and downstream throughput will not exceed the 2 Gbps limit.
- 5000 Mbps is configured on the virtual platform. The software version running on the device is Cisco IOS XE Cupertino 17.8.1a and this means Tx data is throttled at 5000 Mbps. Rx is unthrottled.

Example:

```
Device(config)# platform hardware throughput crypto ?
  100M  100 mbps bidirectional thput
  10M   10 mbps bidirectional thput
  15M   15 mbps bidirectional thput
  1G    2 gbps aggregate thput
  2.5G  5 gbps aggregate thput
  250M  250 mbps bidirectional thput
  25M   25 mbps bidirectional thput
  500M  1gbps aggregate thput
  50M   50 mbps bidirectional thput
Device(config)# platform hardware throughput crypto 1G
% These values don't take effect until the next reboot.
Please save the configuration.

OR

Device(config)# platform hardware throughput level MB 5000
%Throughput has been set to 5000 Mbps.
```

Step 4 **exit**

Exits global configuration mode and returns to privileged EXEC mode.

Example:

```
Device# exit
```

Step 5 **copy running-config startup-config**

Saves your entries in the configuration file.

Example:

```
Device# copy running-config startup-config
Destination filename [startup-config]?
Building configuration...
[OK]
```

Step 6 **reload**

Reloads the device.

Note Perform this step only if the device you are configuring throughput on a physical platform.

Skip this step if you are configuring throughput on a virtual platform.

Example:

```
Device# reload
```

Step 7 Depending on whether the device is a physical or virtual one, enter the applicable command:

- **For physical platforms: show platform hardware throughput crypto**
- **For virtual platforms: show platform hardware throughput level**

Displays the current throughput level on the device.

Tip On physical platforms, you can also enter the **show platform hardware qfp active feature ipsec state** privileged EXEC command to display the configured throughput level.

Example:

```
Device# show platform hardware throughput crypto
Current configured crypto throughput level: 1G
     Level is saved, reboot is not required
Current enforced crypto throughput level: 1G
Crypto Throughput is throttled at 2G(Aggregate)
Default Crypto throughput level: 10M

OR

Device# show platform hardware throughput level
The current throughput level is 5000000 kb/s
```

Configuring a Tier-Based Throughput

This task shows you how to configure a tier-based throughput level on physical and virtual platforms. If you do not configure a throughput level, the platform's default throughput level is effective.

Tier-based throughput levels are supported starting with Cisco IOS XE Cupertino 17.7.1a only.

Configuration of a throughput level requires a reload on physical platforms (Catalyst 8200, 8300, and 8500 Series Edge Platforms). A reload is not required for virtual platforms (Catalyst 8000V Edge Software).

Before you begin

- Read sections Numeric and Tier-Based Throughput, on page 141 and Numeric vs. Tier-Based Throughput Configuration, on page 151.
- Ensure that a boot level license is already configured on the device. Otherwise you will not be able to configure a throughput value. See Configuring a Boot Level License, on page 155. In the output of the **show version** privileged EXEC command ensure that a license is mentioned in the `License Level` field.
- If you are configuring Tier 2 (T2) or a higher tier, you must install a Smart Licensing Authorization Code (SLAC) before you start with this task. See Installing SLAC for an HSECK9 License, on page 157.
  - On physical platforms, T2 or higher tiers are not displayed if SLAC is not installed.
  - On virtual platforms, all tier options are displayed even if SLAC is not installed. But SLAC is required if you want to configure T2 or a higher tier.
- If you want to configure Tier 3 (T3) ensure that the boot level license is Network Advantage/ DNA Advantage, or Network Premier/DNA Premier. T3 and higher tiers are not supported with Network Essentials and DNA Essentials.
- You can configure the `T1` value with or without an HSECK9 license. The system allows both. The difference is that aggregate throttling is effective if HSECK9 is available on the device. See: Release-Wise Changes in Throttling Behavior, on page 144.
- Note the throughput you are entitled to. This is indicated in the Cisco DNA license PID you purchase.

Step 1 Depending on whether the device is a physical or virtual one, enter the applicable command:

- For physical platforms: **show platform hardware throughput crypto**
- For virtual platforms: **show platform hardware throughput level**

Displays the current throughput level on the device.

In the accompanying example:

- The **show platform hardware throughput crypto** sample output is of a physical platform (a C8300-2N2S-4T2X). Here throughput is currently throttled at 250 Mbps.
- The **show platform hardware throughput level** sample output is of a virtual platform (a C8000V). Here the current throughput level is 10 Mbps.

Example:

```
Device# show platform hardware throughput crypto
show platform hardware throughput crypto
Current configured crypto throughput level: 250M
     Level is saved, reboot is not required
Current enforced crypto throughput level: 250M
Crypto Throughput is throttled at 250M
Default Crypto throughput level: 10M
Current boot level is network-premier

OR

Device# show platform hardware throughput level
The current throughput level is 10000 kb/s
```

Step 2 **show license authorization**

(Optional) Displays SLAC information on the product instance.

In the accompanying example:

- SLAC is installed on the physical platform. This is so we can configure T2.
- SLAC is not available on the virtual platform. Note how this affects throughput configuration in the subsequent steps.

Example:

```
Device# show license authorization
Overall status:
  Active: PID:C8300-2N2S-4T2X,SN:FDO2250A0J5
      Status: SMART AUTHORIZATION INSTALLED on Mar 02 05:05:19 2022 UTC
      Last Confirmation code: 418b11b3

Authorizations:
  Router US Export Lic. for DNA (DNA_HSEC):
    Description: U.S. Export Restriction Compliance license for
    DNA based Routers
    Total available count: 1
    Enforcement type: EXPORT RESTRICTED
    Term information:
      Active: PID:C8300-1N1S-4T2X,SN:FDO2250A0J5
        Authorization type: SMART AUTHORIZATION INSTALLED
        License type: PERPETUAL
          Term Count: 1

Purchased Licenses:
  No Purchase Information Available

OR

Device# show license authorization
Overall status:
  Active: PID:C8000V,SN:9I8GRCH8CMN
      Status: NOT INSTALLED
```

Step 3 **configure terminal**

Enters global configuration mode.

Example:

```
Device# configure terminal
```

Step 4 Depending on whether the device is a physical or virtual one, configure the applicable command:

- For physical platforms: **platform hardware throughput crypto** {**T0** | **T1** | **T2** | **T3** | **T4** | **T5**}
- For virtual platforms: **platform hardware throughput level MB** {**T0** | **T1** | **T2** | **T3** | **T4** }

Configures a tier-based throughput. The throughput options that are displayed, depend on the device.

Note Only tiers are mentioned in command, for the sake of clarity. When you enter the command on the CLI, numeric and tier values are displayed - as shown in the accompanying example.

The following apply to both physical and virtual platforms:

- Ensure that you have configured a boot level license already. Otherwise the command for throughput configuration is not recognized as a valid one on the command line interface.

- If you are configuring T2 or a higher tier, you have installed SLAC.

  On a physical platform, you will not be able to configure T2 or a higher tier if SLAC is not installed.

  On a virtual platform, if you configure T2 or a higher tier without SLAC, the product instance automatically tries to reach CSSM to request and install SLAC. If it is successful, throughput is set to the configured tier. If it is not successful, the system sets the throughput *to* 250 Mbps. If and when SLAC is installed, the throughput is automatically set to the last configured value.

In the accompanying example:

- 1 Gbps is configured on the physical platform. The software version running on the device is Cisco IOS XE Cupertino 17.8.1a and this means aggregate throughput throttling applies. After reload, the sum of upstream and downstream throughput will not exceed the 2 Gbps limit.

- 5000 Mbps is configured on the virtual platform. The software version running on the device is Cisco IOS XE Cupertino 17.8.1a and this means Tx data is throttled at 5000 Mbps. Rx is unthrottled.

- On the physical platform (**platform hardware throughput crypto**), T2 and higher tiers are displayed, because SLAC is installed. If SLAC were not available, T1 would have been the highest tier displayed.

  The software version running on the device is Cisco IOS XE Cupertino 17.8.1a and this means aggregate throughput throttling applies. After reload, the sum of upstream and downstream throughput will not exceed the 2 Gbps limit.

- On the virtual platform (**platform hardware throughput level MB**), all tiers are displayed. After T2 is configured, the system message alerts you to the fact that the configuration is not set, because SLAC is not installed.

Example:

```
Device(config)# platform hardware throughput crypto ?
  100M  100 mbps bidirectional thput
  10M   10 mbps bidirectional thput
  15M   15 mbps bidirectional thput
  1G    2 gbps aggregate thput
  2.5G  5 gbps aggregate thput
  250M  250 mbps bidirectional thput
  25M   25 mbps bidirectional thput
  500M  1gbps aggregate thput
  50M   50 mbps bidirectional thput
  T0    T0(up to 15 mbps) bidirectional thput
  T1    T1(up to 100 mbps) bidirectional thput
  T2    T2(up to 2 gbps) aggregate thput
  T3    T3(up to 5 gbps) aggregate thput

Device(config)# platform hardware throughput crypto T2
% These values don't take effect until the next reboot.
Please save the configuration.
*Mar 02 05:06:19.042: %CRYPTO_SL_TP_LEVELS-6-SAVE_CONFIG_AND_RELOAD:
New throughput level not applied until reload; please save config

OR
Device(config)# platform hardware throughput level MB ?
  100    Mbps
  1000   Mbps
  10000  Mbps
  15     Mbps
  25     Mbps
  250    Mbps
  2500   Mbps
  50     Mbps
  500    Mbps
  5000   Mbps
  T0     Tier0(up to 15M throughput)
  T1     Tier1(up to 100M throughput)
  T2     Tier2(up to 1G throughput)
  T3     Tier3(up to 10G throughput)
  T4     Tier4(unthrottled)

Device(config)# platform hardware throughput level MB T2
%Requested throughput will be set once HSEC authorization
code is installed
```

Step 5 **exit**

Exits global configuration mode and returns to privileged EXEC mode.

Example:

```
Device# exit
```

Step 6 **copy running-config startup-config**

Saves your entries in the configuration file.

Example:

```
Device# copy running-config startup-config
Destination filename [startup-config]?
Building configuration...
[OK]
```

Step 7 **reload**

Reloads the device.

Note Perform this step only if the device you are configuring throughput on a physical platform.

Skip this step if you are configuring throughput on a virtual platform.

Example:

```
Device# reload
```

Step 8 Depending on whether the device is a physical or virtual one, enter the applicable command:

- **For physical platforms: show platform hardware throughput crypto**
- **For virtual platforms: show platform hardware throughput level**

Displays the current throughput level on the device.

In the accompanying example:

- On the physical platform, the tier value is set to T2.

  Tip On a physical platform, you can also enter the **show platform hardware qfp active feature ipsec state** privileged EXEC command to display the configured throughput level.

- On the virtual platform, throughput is set to 250 Mbps. If and when SLAC is installed, the throughput will be automatically set to the last configured value, which is T2.

Example:

```
Device# show platform hardware throughput crypto
Current configured crypto throughput level: T2
     Level is saved, reboot is not required
```

```
Current enforced crypto throughput level: 1G
Crypto Throughput is throttled at 2G(Aggregate)
Default Crypto throughput level: 10M
Current boot level is network-premier

OR

Device# show platform hardware throughput level
The current throughput level is 250000 kb/s
```

Converting From a Numeric Throughput Value to a Tier

This task shows you how to convert a numeric throughput value to a tier-based throughput value. To know how numeric throughput values are mapped to tier values refer to the table here:Tier and Numeric Throughput Mapping, on page 144 .

Converting the throughput level requires a reload on physical platforms (Catalyst 8200, 8300, and 8500 Series Edge Platforms). A reload is not required for virtual platforms (Catalyst 8000V Edge Software).

Before you begin

- Read section Numeric vs. Tier-Based Throughput Configuration, on page 151.
- If you are converting numeric throughput that is equal or greater than 250 Mbps, ensure that a SLAC is installed on the device. See Installing SLAC for an HSECK9 License, on page 157.
- The software version running on the device is Cisco IOS XE Cupertino 17.7.1a or a later release.

Step 1 Depending on whether the device is a physical or virtual one, enter the applicable command:

- For physical platforms: **show platform hardware throughput crypto**
- For virtual platforms: **show platform hardware throughput level**

Displays the currently running throughput on the device.

Example:

```
Device# show platform hardware throughput crypto
Current configured crypto throughput level: 500M
     Level is saved, reboot is not required
Current enforced crypto throughput level: 500M
Crypto Throughput is throttled at 500M
Default Crypto throughput level: 10M
Current boot level is network-premier

OR

Device# show platform hardware throughput level
The current throughput level is 100000 kb/s
```

Step 2 Depending on whether the device is a physical or virtual one, enter the applicable command:

- For physical platforms: **license throughput crypto auto-convert**
- For virtual platforms: **license throughput level auto-convert**

Converts the numeric throughput to a tier-based throughput value. The converted tier value is displayed on the CLI.

Example:

```
Device# license throughput crypto auto-convert
Crypto throughput auto-convert from level 500M to T2

% These values don't take effect until the next reboot.
Please save the configuration.
*Dec  8 03:21:01.401: %CRYPTO_SL_TP_LEVELS-6-SAVE_CONFIG_AND_RELOAD:
New throughput level
not applied until reload; please save config

OR

Device# license throughput level auto-convert
%Throughput tier set to T1 (100 Mbps)
% Tier conversion is successful.
Please write memory to save the tier config
```

Step 3 **copy running-config startup-config**

Saves your entries in the configuration file.

Note Even though the command you use to convert from numeric to tier-based throughput is a privileged EXEC command, it changes running configuration from a numeric value to a tier-based value. You must therefore save configuration for the next reload to be displayed with a tier value.

Example:

```
Device# copy running-config startup-config
Destination filename [startup-config]?
Building configuration...
[OK]
```

Step 4 **reload**

Reloads the device.

Note A reload is required only on physical platforms.

Example:

```
Device# reload
Proceed with reload? [confirm]
*Dec  8 03:24:09.534: %SYS-5-RELOAD: Reload requested by console.
Reload Reason:
Reload Command
```

Step 5 Depending on whether the device is a physical or virtual one, enter the applicable command:

- For physical platforms: **show platform hardware throughput crypto**
- For virtual platforms: **show platform hardware throughput level**

Displays the currently running throughput on the device.

Example:

```
Device# show platform hardware throughput crypto
Current configured crypto throughput level: T2
     Level is saved, reboot is not required
Current enforced crypto throughput level: 1G
Crypto Throughput is throttled at 1G
Default Crypto throughput level: 10M
Current boot level is network-premier
```

```
OR

Device# show platform hardware throughput level
The current throughput level is 100000 kb/s
```

Step 6 Verify that conversion is complete.

- For physical platforms: **license throughput crypto auto-convert**
- For virtual platforms: **license throughput level auto-convert**

Tip To cross-check that conversion is complete, you can also enter the conversion command again. If the numeric throughput value has already been converted, the system displays a message confirming this.

Example:

```
Device# license throughput crypto auto-convert
Crypto throughput is already tier based, no need to convert.

OR

Device# license throughput level auto-convert
% Tier conversion not possible since the device is already
in tier licensing
```

Upgrading from a Release Supporting Numeric Throughput to a Release Supporting Tiers

If you are upgrading to Cisco IOS XE Cupertino 17.7.1 or later release ***and*** the license PID is a tier-based one, you can convert throughput configuration to a tier-based value, or you can retain the numeric throughput configuration.

Note There is no functional impact if you have tier-based license PID in CSSM and a numeric throughput value is configured on the device.

If you want to convert to a tier-based value note the required action depending on the throughput level that is configured:

Throughput Configuration Before Upgrade	Action Before Upgrade	Action After Upgrade to 17.7.1 or Later
Lesser than 250 Mbps	No action required.	Converting From a Numeric Throughput Value to a Tier, on page 165
Equal to 250 Mbps	Obtain an HSECK9 license and install SLAC if you want to convert to T2.	Converting From a Numeric Throughput Value to a Tier, on page 165
Greater than 250 Mbps	No action required.	Converting From a Numeric Throughput Value to a Tier, on page 165

Downgrading from a Release Supporting Tiers to a Release Supporting Only Numeric Throughput

If you are downgrading to a release where only numeric throughput configuration is supported, you *must* convert tier-based throughput configuration to a numeric throughput value before downgrade. This is applicable even if the license PID is a tier-based license PID.

Caution If a tier-based throughput value was configured before downgrade and you downgrade without changing to a numeric value, tier configuration is not recognized by a pre-17.7.1 image and configuration fails. Further, throughput may not be restored to the pre-downgrade level and you have to configure a numeric throughput level after downgrade.

Throughput Configuration Before Downgrade	Action Before Downgrade	Action After Downgrade to a pre-17.7.1 Version
Numeric	No action required.	No action required.
Tier	Configuring a Numeric Throughput, on page 158	No action required.

Available Licensing Models

The licensing model defines *how* you account for or report the licenses that you use, to Cisco. The following licensing models are available on the Cisco Catalyst 8000 Edge Platforms Family:

Smart Licensing Using Policy

With this licensing model, you purchase the licenses you want to use, configure them on the device, and then report license usage – as required. You do not have to complete any licensing-specific operations, such as registering or generating keys before you start using the software and the licenses that are tied to it - unless you are using export-controlled and enforced licenses.

This licensing model is supported on all products in the Cisco Catalyst 8000 Edge Platforms Family.

For more information, see Smart Licensing Using Policy for Cisco Enterprise Routing Platforms.

Pay As You Go (PAYG) Licensing

Note This licensing model is available only on Catalyst 8000V Edge Software.

Cisco Catalyst 8000V supports the PAYG licensing model with Amazon Web Services (AWS) and Microsoft Azure Marketplace - in both the autonomous mode and the controller mode. The Cisco Catalyst 8000V hourly-billed Amazon Machine Image (AMI) or the Pay As You Go licensing model allows you to consume an instance for a defined period of time.

- In the autonomous mode, you can directly launch an instance from the AWS or Azure Marketplace and start using it. The licenses are embedded in the image and the selected license package and configured throughput level are effective when you launch the instance
- In the controller mode, which is supported from Cisco IOS-XE Bengaluru 17.5.1, you must first onboard the device into Cisco SD-WAN as per Onboard Cisco Catalyst 8000V Edge Software Hosted by a Cloud Service, Using PAYG Licensing. After this, when you launch the instance from AWS, the device comes-up with the license already installed for unlimited throughput.

Managed Service Licensing Agreement

A Managed Service License Agreement (MSLA) is a buying program agreement, designed for Service Providers.

- **MSLA in Cisco SD-WAN Controller Mode**

  In the Cisco SD-WAN controller mode, an MSLA is supported on all products in the Cisco Catalyst 8000 Edge Platforms Family. For more information, see:

  Managed Service Licensing Agreement (MSLA) for Cisco SD-WAN At-a-Glance

  Cisco SD-WAN Getting Started Guide → *Manage Licenses for Smart Licensing Using Policy*.

  Cisco vManage How-Tos for Cisco IOS XE SD-WAN Devices → *Manage Licenses for Smart Licensing Using Policy*.

- **MSLA in Autonomous Mode**

  In the autonomous mode, an MSLA is available only with Catalyst 8000V Edge Software, starting from Cisco IOS XE Cupertino 17.9.1a.

  For more information, see: MSLA.

CHAPTER 16

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.
- From Cisco IOS XE 17.6.1, you can enable the FIPS mode if the host and VM supports RDRAND or RDSEED, or both instructions. Otherwise, an error message is displayed.

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

CHAPTER 17

Upgrading the Cisco IOS XE Software

The Cisco Catalyst 8000V virtual router runs on the Cisco IOS XE platform, the same platform that has powered Cisco CSR1000V or Cisco ISRv. To use the Cisco Catalyst 8000V router, first obtain the software image from the Cisco Software Download page. Obtain the installation files and then begin the installation or upgrade. To know more about the installation files, see Installation Files, on page 9.

If you are an existing Cisco CSR1000V or a Cisco ISRv user, you must download the latest installation file from the Cisco Software Download page and begin the upgrade process by following the procedures mentioned in this chapter.

Software Packaging for Cisco Catalyst 8000V

The software image for Cisco Catalyst 8000V is available as a consolidated package and as optional subpackages. Each consolidated package contains a collection of software subpackages, and each software subpackage is an individual software file that controls a different element or elements of the virtual router. Using a consolidated package, you can upgrade all the individual subpackages with a single software image download.

You can upgrade an individual software subpackage individually, or upgrade all the software subpackages for a specific consolidated package as part of a complete consolidated package upgrade. If you want to run the router using individual subpackages that are part of a consolidated package, download the image from Cisco.com and extract the individual subpackages from the image.

Upgrading using subpackage consumes less memory than upgrading through a consolidated package. For this reason, upgrading through subpackages is the recommended method, especially for deployments with small footprints.

Note Upgrading a Cisco ISRv or a Cisco CSR1000V to Cisco Catalyst 8000V does not alter the file system layout nor provide any of the new features such as the Secure Object Store which rely on the file system. You must perform a fresh installation to activate these features.

Important If you are an existing Cisco CSR1000V or Cisco ISRV user, and you are upgrading to Cisco Catalyst 8000V, your licenses continue to function as-is. However, an HSECK9 license is mandatory to run any throughput level greater than 250 Mbps. If you were running a throughput level greater than 250 Mbps prior to the upgrade, you must purchase an HSECK9 license for service continuity after the upgrade. If an HSECK9 license is not available after upgrade, throughput is restricted to 250 Mbps. If you want to switch to Cisco DNA subscription-based licensing model, you must perform a fresh Catalyst 8000V deployment.

Prerequisites for Upgrading Cisco Catalyst 8000V

- Obtain the Cisco Catalyst 8000V software image from the Cisco Software Download page. To know how to obtain the installation files, see Download the Installation Files.
- 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 that you meet the memory requirements of the VM for the Cisco Catalyst 8000V software image. If the upgraded version requires more memory than your previous version, increase the memory allocation on the VM before you begin the upgrade process.

HSECK9 License Requirements for Cisco CSR1000V and Cisco ISRV Upgrade

If you are upgrading a Cisco CSR1000V or Cisco ISRV router where *throughput is greater than 250 Mbps*, to Cisco Catalyst 8000V (Cisco IOS XE Bengaluru 17.4.1 and later), a High Security (HSECK9) license is required.

Depending on your pre-upgrade setup, ensure that you meet the corresponding HSECK9 license requirements, before you upgrade:

- If the Cisco CSR1000V or Cisco ISRV is connected to CSSM, then you must ensure the following:
    - Throughput greater than 250 Mbps is part of start-up configuration.

      To check start-up configuration, enter the **show running-config** command in privileged EXEC mode. For example:

```
Device# show running-config | include throughput
platform hardware throughput level MB 500
```

- There is a positive balance of the required number of HSECK9 licenses (DNA_HSECK9) in the corresponding Smart Account and Virtual Account in CSSM.

No further pre-upgrade action is required. As long as the device is connected to CSSM, on upgrade, the device automatically triggers the HSECK9 request and installs the required Smart Licensing Authorization Code (SLAC).

- If the Cisco CSR1000V or Cisco ISRV is using Specific License Reservation (SLR), then you must update the SLR authorization code to include an HSECK9 license (DNA_HSECK9) and only then upgrade the device. This ensures uninterrupted throughput after upgrade.

  This example shows you how to update the SLR authorization code: Example: Smart Licensing (SLR With Throughput >250 Mbps, Without Export-Controlled License) to Smart Licensing Using Policy.

If throughput is lesser than or equal to 250 Mbps, an HSECK9 license is not required.

Restrictions for Upgrading Cisco Catalyst 8000V

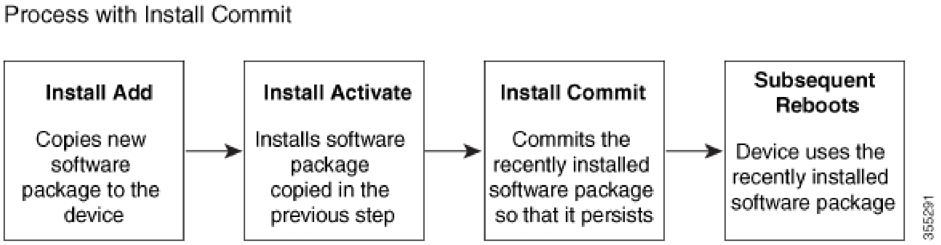
- You can upgrade to a new software version on the same VM only. The procedures do not describe how to install or rehost an existing router running the same or upgraded software version on a different VM.
- The .bin file is applicable for upgrading or downgrading your software. The .iso, .qcow2, and .ova files are used for first-time installation only.
- If you are upgrading to Cisco Catalyst 8000V, your licenses will continue to function as is. However, if you wish to switch to the CDNA licensing model, you must perform a fresh installation.
- The Cisco Catalyst 8000V router does not support In-Service Software Upgrade (ISSU).
- The system requirements for the x86 hardware might differ from those of the hardware currently running on the router.
- In the case of an upgrade from Cisco CSR1000V or Cisco ISRv, the disk partition structure remains the same as the previous version, and the secure object storage functionality is not available.
- If you want to upgrade to Cisco Catalyst 8000V from a Cisco CSR1000V or a Cisco ISRv prior to 16.12.x, first upgrade from your current version to 16.12.x. Then, upgrade to the latest version of Cisco Catalyst 8000V.
- You cannot upgrade a Cisco CSR1000V running PCI pass-through to Cisco Catalyst 8000V as Cisco Catalyst 8000V does not support PCI pass-through.
- If you 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 Cisco CSR1000V but not Cisco ISRv.
- If you want to upgrade from Cisco CSR1000V to Cisco Catalyst 8000V, or if you're upgrading from a lower to a higher version of Cisco Catalyst 8000V, only an N-2 or an N-1 to N release upgrade path is supported. Here, N-1 and N-2 refer to extended maintenance releases. For example, if you want to upgrade a CSR1000V 17.3.x instance to a Cisco Catalyst 8000V 17.11.1a release, 17.6.x is the lowest N-x version you need to update.

- Cisco Catalyst 8000V does not support L2TP functionality including L2TP client and L2TP Network Server (LNS).

Install Mode Process Flow

The install mode process flow comprises three commands to perform the installation and the upgrade of Cisco Catalyst 8000V–**install add**, **install activate**, and **install commit**.

The following flow chart explains the install process with the install commands:


Process with Install Commit
Install Add
Copies new software package to the device
Install Activate
Installs software package copied in the previous step
Install Commit
Commits the recently installed software package so that it persists
Subsequent Reboots
Device uses the recently installed software package
355291

The **install add** command copies the software package from a local or remote location to the platform. The location can be FTP, HTTP, HTTPs, or TFTP. This command extracts individual components of the .package file into subpackages and packages.conf files. It also validates the file to ensure that the image file is specific to Cisco Catalyst 8000V.

The **install activate** command performs the required validations and provisions the packages previously added using the **install add** command. It also triggers a system reload.

The **install commit** command confirms the packages previously activated using the **install activate** command, and ensures the updates are persistent over reloads.

Note Installing an update replaces any previously installed software image. At any time, you can install only one image in your instance.

The following table specifies the list of commands that are used when you install or upgrade your Cisco IOS XE platform:

Table 25: List of install Commands

Command	Syntax	Purpose
install add	**install add file** *location:filename.bin*	Copies the contents of the image and the package to the software repository. File location may be local or remote. This command does the following: • Validates the file–checksum, platform compatibility checks, and so on. • Extracts individual components of the package into subpackages and packages.conf • Copies the image into the local inventory and makes it available for the next steps.
install activate	**install activate**	Activates the package added using the **install add** command. • Use the **show install summary** command to see which image is inactive. • The system reloads on executing this command. Confirm if you want to proceed with the activation. Use this command with the **prompt-level none** keyword to automatically ignore any confirmation prompts.

Command	Syntax	Purpose
(install activate) auto abort-timer	**install activate auto-abort timer** *<30-1200>*	The **auto-abort timer** starts automatically with a default value of 120 minutes. If the **install commit** command is not executed within the time provided, the activation process is terminated, and the system returns to the last-committed state. • You can change the time value while executing the **install activate** command. • The **install commit** command stops the timer and continues the installation process. • The **install activate auto-abort timer stop** command stops the timer without committing the package. • Use this command with the **prompt-level none** keyword to automatically ignore any confirmation prompts. • This command is valid only in the three-step install variant.
install commit	**install commit**	Commits the package activated using the **install activate** command and makes it persistent over reloads. • Use the **show install summary** command to see which image is not committed.

Command	Syntax	Purpose
install abort	**install abort**	Terminates the installation and returns the system to the last-committed state. • This command is applicable only when the package is in the activated status (uncommitted state). • If you have already committed the image using the **install commit** command, use the **install rollback to** command to return to the preferred version.
install remove	**install remove {file** *<filename>* **| inactive}**	Deletes the inactive packages from the platform repository. Use this command to free up space. • **file**: Removes the specified files. • **inactive**: Removes all the inactive files.
install rollback to	**install rollback to {base | label | committed | id}**	Rolls back the software set to a saved installation point or to the last-committed installation point. The following are the characteristics of this command: • Requires reload. • Is applicable only when the package is in the committed state. • Use this command with the **prompt-level none** keyword to automatically ignore any confirmation prompts. **Note** If you are performing install rollback to a previous image, the previous image must be installed in the install mode.

Apart from the above-mentioned commands, you can also use the following show commands to verify the installation or upgrade:

Table 26: List of show Commands

Command	Syntax	Purpose
show install log	**show install log**	Provides the history and details of all the install operations that have been performed since the platform was booted.
show install package	**show install package** <*filename*>	Provides details about the .pkg/.bin file that is specified.
show install summary	**show install summary**	Provides an overview of the image versions and their corresponding install states.
show install active	**show install active**	Provides information about the active packages.
show install inactive	**show install inactive**	Provides information about the inactive packages, if any.
show install committed	**show install committed**	Provides information about the committed packages.
show install uncommitted	**show install uncommitted**	Provides information about uncommitted packages, if any.
show install rollback	**show install rollback {point-id | label}**	Displays the package associated with a saved installation point.
show version	**show version [rp-slot] [installed [user-interface] | provisioned | running]**	Displays information about the current package along with the platform information.

Booting Cisco Catalyst 8000V in the Install Mode

You can install, activate, and commit a software package using a single command (one-step install procedure) or multiple separate commands (three-step install procedure).

If your Cisco Catalyst 8000V device is working in the bundle mode, you must use the one-step install procedure to initially convert the platform from the bundle mode to the install mode. You can then perform subsequent installs and upgrades by using either the one-step or the three-step installation method.

One-Step Installation or Converting from Bundle Mode to Install Mode

This procedure uses the **install add file activate commit** command in the privileged EXEC mode to install a software package and to upgrade the platform to a newer version.

The one-step install procedure converts a platform running in the bundle boot mode to the install mode. After the command is executed, the platform reboots in the install boot mode.

Note

- All the CLI actions (for example, add, activate, and so on) are executed.
- The configuration save prompt appears if an unsaved configuration is detected.
- The reload prompt appears after the second step in this workflow. Use the **prompt-level none** keyword to automatically ignore the confirmation prompts.
- If the prompt-level is set to None, and there is an unsaved configuration, the install fails. You must save the configuration before reissuing the command.

SUMMARY STEPS

1. **enable**
2. **install add file location:** *filename* [**activate commit**]
3. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables the privileged EXEC mode. Enter your password, if prompted.
Step 2	**install add file location:** *filename* [**activate commit**] **Example:** `Device# install add file bootflash:c8000v-universalk9.BLD_POLARIS_DEV_LATEST_20220227_153436.SSA.bin activate commit`	Copies the software install package from a local or remote location (through FTP, HTTP, HTTPs, or TFTP) to the platform and extracts the individual components of the .package file into subpackages and packages.conf files. It also performs a validation and compatibility check for the platform and image versions, activates the package, and commits the package to make it persistent across reloads. The platform reloads after this command is run.
Step 3	**exit** **Example:** `Device# exit`	Exits the privileged EXEC mode and returns to the user EXEC mode.

Three-Step Installation

The three-step installation procedure uses individual **install add**, **install activate**, and **install commit** commands for installing a software package, and to upgrade the platform to a newer version.

Note

- You can perform this procedure only after the platform is in the install mode.
- All the CLI actions (for example, add, activate, and so on) are executed.
- The configuration save prompt appears if an unsaved configuration is detected.
- The reload prompt appears after the install activate step in this workflow. Use the **prompt-level none** keyword to automatically ignore the confirmation prompts.

SUMMARY STEPS

1. **enable**
2. **install add file location:** *filename*
3. **show install summary**
4. **install activate** [**auto-abort-timer** <*time*>]
5. **install abort**
6. **install commit**
7. **install rollback to committed**
8. **install remove** {**file** *filesystem: filename* | **inactive**}
9. **show install summary**
10. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables the privileged EXEC mode. Enter your password, if prompted.
Step 2	**install add file location:** *filename* **Example:** `Device# install add file bootflash:c8000v-universalk9.BLD_POLARIS_DEV_LATEST_20220227_153436.SSA.bin`	Copies the software install package from a remote location (through FTP, HTTP, HTTPs, or TFTP) to the platform, and extracts the individual components of the .package file into subpackages and packages.conf files.
Step 3	**show install summary** **Example:** `Device# show install summary`	(Optional) Provides an overview of the image versions and their corresponding install state.
Step 4	**install activate** [**auto-abort-timer** <*time*>] **Example:** `Device# install activate auto-abort-timer 120`	Activates the previously added package and reloads the platform. • When you're performing a full software install, do not provide a package filename. • The **auto-abort-timer** starts automatically with the **install activate** command; the default for the timer is 120 minutes. If the **install commit** command is not

	Command or Action	Purpose
		run before the timer expires, the install process is automatically terminated. The platform reloads and boots up with the last committed version.
Step 5	**install abort** **Example:** `Device# install abort`	(Optional) Terminates the software install activation and returns the platform to the last committed version. Use this command only when the image is in the activated state and not when the image is in the committed state.
Step 6	**install commit** **Example:** `Device# install commit`	Commits the new package installation and makes the changes persistent over reloads.
Step 7	**install rollback to committed** **Example:** `Device# install rollback to committed`	(Optional) Rolls back the platform to the last committed state.

Step 8	**install remove** {**file** *filesystem: filename* | **inactive**}
	Example:
	`Device# install remove inactive`
		(Optional) Deletes the software installation files.
		• **file**: Deletes a specific file.
		• **inactive**: Deletes all the unused and inactive installation files.

Step 9	**show install summary** **Example:** `Device# show install summary`	(Optional) Displays information about the current state of the system. The output of this command varies according to the **install** commands run prior to this command.
Step 10	**exit** **Example:** `Device# exit`	Exits the privileged EXEC mode and returns to the user EXEC mode.

Sample Upgrade Output from Release 17.06.02 To Release 17.07.01

```
======================================
Upgrade steps
install add file bootflash:/ c8000v-universalk9.17.07.01a.SPA.bin
install activate
install commit
======================================

Router#show version | inc IOS XE
Cisco IOS XE Software, Version 17.06.02
Router#show version | inc mode
Router operating mode: Autonomous

Router# dir bootflash:*bin*
Directory of bootflash:/*bin*

Directory of bootflash:/
```

```
   31  -rw-   832807301   Mar 7 2022 02:07:28 +00:00  c8000v-universalk9.17.07.01a.SPA.bin
5183766528 bytes total (2348220416 bytes free)

Router#install add file bootflash:/c8000v-universalk9.17.07.01a.SPA.bin
install_add: START Mon Mar 7 02:16:30 UTC 2022
install_add: Adding PACKAGE
install_add: Checking whether new add is allowed ....

--- Starting Add ---
Performing Add on Active/Standby
  [1] Add package(s) on R0
  [1] Finished Add on R0
Checking status of Add on [R0]
Add: Passed on [R0]
Finished Add

Image added. Version: 17.07.01a.0.1883
SUCCESS: install_add  Mon Mar  7 02:20:07 UTC 2022
VK5-C8K-8G-1762-1#

Router# show install summary
[ R0 ] Installed Package(s) Information:
State (St): I - Inactive, U - Activated & Uncommitted,
            C - Activated & Committed, D - Deactivated & Uncommitted
--------------------------------------------------------------------------------
Type  St   Filename/Version
--------------------------------------------------------------------------------
IMG   C    17.06.02.0.2786
IMG   I    17.07.01a.0.1883

--------------------------------------------------------------------------------
Auto abort timer: inactive
--------------------------------------------------------------------------------

=====================
install activate
=====================

Router# show install summary
[ R0 ] Installed Package(s) Information:
State (St): I - Inactive, U - Activated & Uncommitted,
            C - Activated & Committed, D - Deactivated & Uncommitted
--------------------------------------------------------------------------------
Type  St   Filename/Version
--------------------------------------------------------------------------------
IMG   C    17.06.02.0.2786
IMG   I    17.07.01a.0.1883

Router# install activate
install_activate: START Mon Mar  7 02:50:00 UTC 2022
install_activate: Activating PACKAGE
Following packages shall be activated:
/bootflash/c8000v-rpboot.17.07.01a.SPA.pkg
/bootflash/c8000v-mono-universalk9.17.07.01a.SPA.pkg
/bootflash/c8000v-firmware_nim_xdsl.17.07.01a.SPA.pkg
/bootflash/c8000v-firmware_nim_shdsl.17.07.01a.SPA.pkg
/bootflash/c8000v-firmware_nim_ge.17.07.01a.SPA.pkg
/bootflash/c8000v-firmware_nim_cwan.17.07.01a.SPA.pkg
/bootflash/c8000v-firmware_nim_async.17.07.01a.SPA.pkg
/bootflash/c8000v-firmware_ngwic_t1e1.17.07.01a.SPA.pkg
/bootflash/c8000v-firmware_dsp_sp2700.17.07.01a.SPA.pkg
/bootflash/c8000v-firmware_dreamliner.17.07.01a.SPA.pkg
```

```
This operation may require a reload of the system. Do you want to proceed? [y/n]y
--- Starting Activate ---
Performing Activate on Active/Standby

  [1] Activate package(s) on R0
    --- Starting list of software package changes ---
    Old files list:
      Modified c8000v-firmware_dreamliner.17.06.02.SPA.pkg
      Modified c8000v-firmware_dsp_sp2700.17.06.02.SPA.pkg
      Modified c8000v-firmware_ngwic_t1e1.17.06.02.SPA.pkg
      Modified c8000v-firmware_nim_async.17.06.02.SPA.pkg
      Modified c8000v-firmware_nim_cwan.17.06.02.SPA.pkg
      Modified c8000v-firmware_nim_ge.17.06.02.SPA.pkg
      Modified c8000v-firmware_nim_shdsl.17.06.02.SPA.pkg
      Modified c8000v-firmware_nim_xdsl.17.06.02.SPA.pkg
      Modified c8000v-mono-universalk9.17.06.02.SPA.pkg
      Modified c8000v-rpboot.17.06.02.SPA.pkg
    New files list:
      Added c8000v-firmware_dreamliner.17.07.01a.SPA.pkg
      Added c8000v-firmware_dsp_sp2700.17.07.01a.SPA.pkg
      Added c8000v-firmware_ngwic_t1e1.17.07.01a.SPA.pkg
      Added c8000v-firmware_nim_async.17.07.01a.SPA.pkg
      Added c8000v-firmware_nim_cwan.17.07.01a.SPA.pkg
      Added c8000v-firmware_nim_ge.17.07.01a.SPA.pkg
      Added c8000v-firmware_nim_shdsl.17.07.01a.SPA.pkg
      Added c8000v-firmware_nim_xdsl.17.07.01a.SPA.pkg
      Added c8000v-mono-universalk9.17.07.01a.SPA.pkg
      Added c8000v-rpboot.17.07.01a.SPA.pkg
    Finished list of software package changes
  [1] Finished Activate on R0
Checking status of Activate on [R0]
Activate: Passed on [R0]
Finished Activate

Send model notification for install_activate before reload
Install will reload the system now!
SUCCESS: install_activate  Mon Mar  7 02:57:34 UTC 2022


===================================
install commit
===================================

Router# show version | inc IOS XE
Cisco IOS XE Software, Version 17.07.01a
Router# show version | inc mode
Router operating mode: Autonomous
Router# show license udi
UDI: PID:C8000V,SN:9JM01Z7G2JH
```

Upgrading in Install Mode

Use either the one-step installation or the three-step installation procedures mentioned in this chapter to upgrade Cisco Catalyst 8000V in the install mode.

Downgrading in Install Mode

Use the **install rollback** command to downgrade the platform to a previous version by pointing it to the appropriate image, provided the image you are downgrading to was installed in the install mode.

The **install rollback** command reloads the platform and boots it with the previous image.

Note The **install rollback** command succeeds only if you have not removed the previous file using the **install remove inactive** command.

Note If you're unable to use this command, you can downgrade by installing the older image using the **install** commands.

Sample Downgrade Configuration

```
==========================
install rollback
==========================

Router# install rollback to base
install_rollback: START Tue Mar 01 03:25:46 UTC 2022
install_rollback: Rolling back to base
This operation may require a reload of the system. Do you want to proceed? [y/n]
*Mar 29 21:17:36.496: %INSTALL-5-INSTALL_START_INFO: R0/0: install_mgr: Started install
rollback
--- Starting Rollback ---
Performing Rollback on all members
 [1] Rollback package(s) on  R0
 [1] Finished Rollback package(s) on  R0
Checking status of Rollback on [R0]
Rollback: Passed on [R0]
Finished Rollback operation
SUCCESS: install_rollback Tue Mar 01 03:30:16 UTC UTC 2022
```

Terminating a Software Installation

You can terminate the activation of a software package in the following ways:

- By allowing the auto-abort-timer to expire before issuing the **install commit** command. When the platform reloads after activating a new image, the auto-abort-timer is triggered (in the three-step install method). When this timer expires, the installation process is terminated and the platform reloads and boots with the last committed version of the software image.

  By using the **install auto-abort-timer stop** command to stop this timer without using the **install commit** command. The new image remains uncommitted in this process.

- By using the **install abort** command which returns the platform to the version that was running before installing the new software. Use this command before issuing the **install commit** command.

Sample Abort Configuration

```
===========================
install abort
===========================
Router# show install summary
[ R0 ] Installed Package(s) Information:
State (St): I - Inactive, U - Activated & Uncommitted,
            C - Activated & Committed, D - Deactivated & Uncommitted
Type  St   Filename/Version
--------------------------------------------------------------------------------
IMG   U    17.09.01.0.154628

--------------------------------------------------------------------------------
Auto abort timer: active , time before rollback - 01:56:56
--------------------------------------------------------------------------------

Router# show version | inc IOS XE
Cisco IOS XE Software, Version BLD_POLARIS_DEV_LATEST_20220227_153436
Router# show version | inc mode
Router operating mode: Autonomous
Router# install abort
install_abort: START Tue Mar 01 04:03:52 UTC 2022

This operation may require a reload of the system. Do you want to proceed? [y/n]y

--- Starting Abort ---
Performing Abort on all members
 [1] Abort packages(s) on  R0
 [1] Finished Abort packages(s) on  R0
Checking status of Abort on [R0]
Abort: Passed on [R0]
Finished Abort operation

SUCCESS: install_abort Tue Mar 01 04:04:45 UTC 2022

Router# Mar  1 04:04:50.161: %PMAN-5-EXITACTION: R0/0: pvp: Process manager is exiting:
reload action requested
```

Troubleshooting Software Installation Using install Commands

Problem Troubleshooting the software installation

Solution Use the following show commands to view installation summary, logs, and software versions.

- **show install summary**
- **show install log**
- **show version**
- **show version running**

Problem Other installation issues

Solution Use the following commands to resolve installation issue:

- **dir** <*install directory*>

- **more location:***packages.conf*
- **show tech-support install**: this command automatically runs the **show** commands that display information specific to installation.
- **request platform software trace archive target bootflash** <*location*>: this command archives all the trace logs relevant to all the processes running on the system since the last reload, and saves this information in the specified location.

Frequently Asked Questions

Q. Can I downgrade from Cisco Catalyst 8000V to Cisco CSR1000V or Cisco ISRv?

A. You can downgrade from Cisco Catalyst 8000V only if you've upgraded to Cisco Catalyst 8000V from a Cisco CSR1000V or a Cisco ISRv 17.3.x or a later version.

Note You cannot downgrade to Cisco CSR1000V or Cisco ISRv if you have freshly installed Cisco Catalyst 8000V.

Q. When I upgrade from a Cisco CSR1000V 16.12.x version or below, will Secure Object Storage be supported?

A. No, Secure Object Storage is not carried over through upgrades. You must perform a fresh installation or reinstall the VM to enable Secure Object Storage support.

Q. Will my license need to change when I upgrade from a Cisco CSR1000V or a Cisco ISRv to Cisco Catalyst 8000V?

A. When you upgrade to Cisco Catalyst 8000V, the licenses remain the same. However, the licenses move from SL to SLE after the upgrade. If the throughput was <=250M before the upgrade, it is retained as is after the upgrade.

If the throughput was >250M and the device was registered to CSSM, the connection stays intact and the throughput automatically triggers the SLAC installation on the device. The corresponding throughput is set once SLAC is installed.

If the device was not connected to CSSM and throughput was >250M, you must manually install SLAC in the offline mode or configure SLE commands to establish trust with CSSM. Then, configure the throughput to trigger the SLAC installation.

Note If SLAC is not installed, the throughput remains at 250M.

Q. Is automation available for the upgrade process?

A. No, automation is currently not supported for the migration.

Q. What is the failure mode handling when I perform a downgrade?

A. When a Cisco CSR1000V image is booting up as a result of a downgrade, the system checks for the partition format. If the partition format does not match the requirements, the boot up is halted. If a Cisco

Catalyst 8000V image is booting up as a result of an upgrade or a downgrade, it continues to boot using the existing partition format.

Q. What is the memory and performance impact after the upgrade?

A. The size of the Cisco Catalyst 8000V image might be slightly larger which could affect the overall memory footprint. However, this does not alter the overall memory requirements. The minimum required RAM for this image is 4GB, and there is no impact on performance by this feature.

CHAPTER 18

Configuring the vCPU Distribution

This chapter specifies the allocation and distribution of the vCPUs in 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 27: Control Plane Extra heavy - vCPU Distribution

Number of vCPUs	1	2	4	8	16
Control Plane	1/3	1/2	1 1/2	1 1/2	6
Service Plane	1/3	1/2	1 1/2	1 1/2	6
Data Plane	1/3	1	1	5	10

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 Wide Area Application Services (WAAS).

vCPU Distribution: Control Plane heavy

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

Table 28: Control Plane heavy - vCPU Distribution

Number of vCPUs	1	2	4	8	16
Control Plane	1/3	1/2	1	1	4
Service Plane	1/3	1/2	1	1	4
Data Plane	1/3	1	2	6	12

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 above mentioned statement is not applicable for Cisco Catalyst 8000V instances running in the controller mode.

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

Table 29: Data Plane heavy - vCPU Distribution

Number of vCPUs	1	2	4	8	16
Control Plane	1/3	1/2	1/2	1/2	2
Service Plane	1/3	1/2	1/2	1/2	2
Data Plane	1/3	1	3	7	14

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 30: Service Plane heavy - vCPU Distribution

Number of vCPUs	1	2	4	8	16
Control Plane	1/3	1/2	1	2	8
Service Plane	1/3	1/2	1	2	8
Data Plane	1/3	1	2	4	8

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. This is the recommended vCPU template to use when NGFW features are enabled on the device. This applies to both SD-WAN and autonomous mode deployments.

vCPU Distribution: Service Plane medium

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

Table 31: Service Plane medium - vCPU Distribution

Number of vCPUs	1	2	4	8	16
Control Plane	1/3	1/2	1	1	4
Service Plane	1/3	1/2	1	1	4
Data Plane	1/3	1	2	6	12

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

Enter the `platform resource` command on theCisco Catalyst 8000V CLI 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.

Determining 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 19

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 195
- Using Basic or Advanced Mode Setup Wizard, on page 196

Setting Up Factory Default Device Using WebUI

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

Before you begin

- 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.

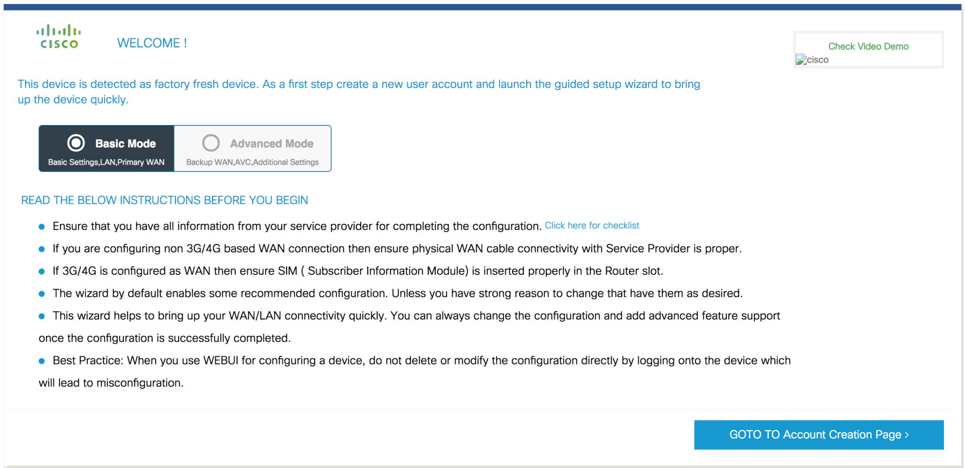
Step 3 Click **Create and Launch Wizard**.

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**.


cisco
WELCOME !
Check Video Demo
cisco

This device is detected as factory fresh device. As a first step create a new user account and launch the guided setup wizard to bring up the device quickly.

● Basic Mode
Basic Settings,LAN,Primary WAN
○ Advanced Mode
Backup WAN,AVC,Additional Settings

READ THE BELOW INSTRUCTIONS BEFORE YOU BEGIN

- Ensure that you have all information from your service provider for completing the configuration. Click here for checklist
- If you are configuring non 3G/4G based WAN connection then ensure physical WAN cable connectivity with Service Provider is proper.
- If 3G/4G is configured as WAN then ensure SIM (Subscriber Information Module) is inserted properly in the Router slot.
- The wizard by default enables some recommended configuration. Unless you have strong reason to change that have them as desired.
- This wizard helps to bring up your WAN/LAN connectivity quickly. You can always change the configuration and add advanced feature support once the configuration is successfully completed.
- Best Practice: When you use WEBUI for configuring a device, do not delete or modify the configuration directly by logging onto the device which will lead to misconfiguration.

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.

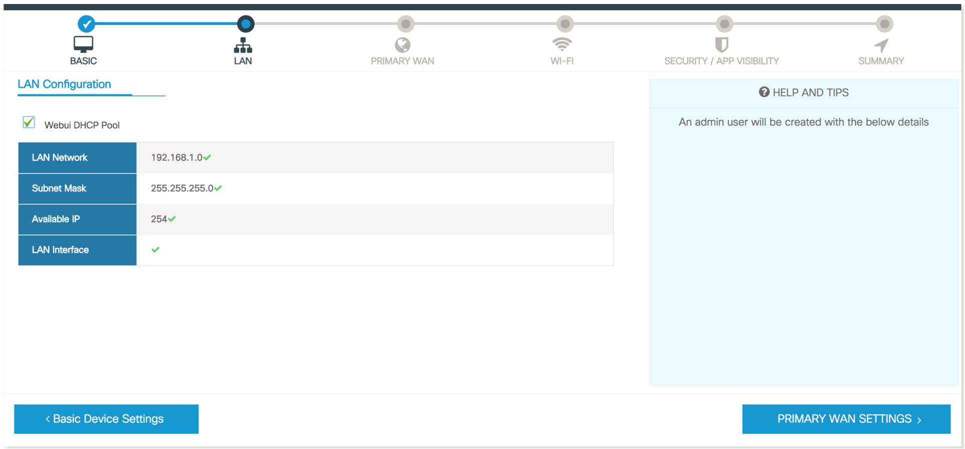
b) If you choose the Create and Associate Access VLAN option, specify the following:

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**.



LAN Configuration

☑ Webui DHCP Pool

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

HELP AND TIPS

An admin user will be created with the below details

Configure Primary WAN Settings

Step 1 Select the primary WAN type. You can 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.

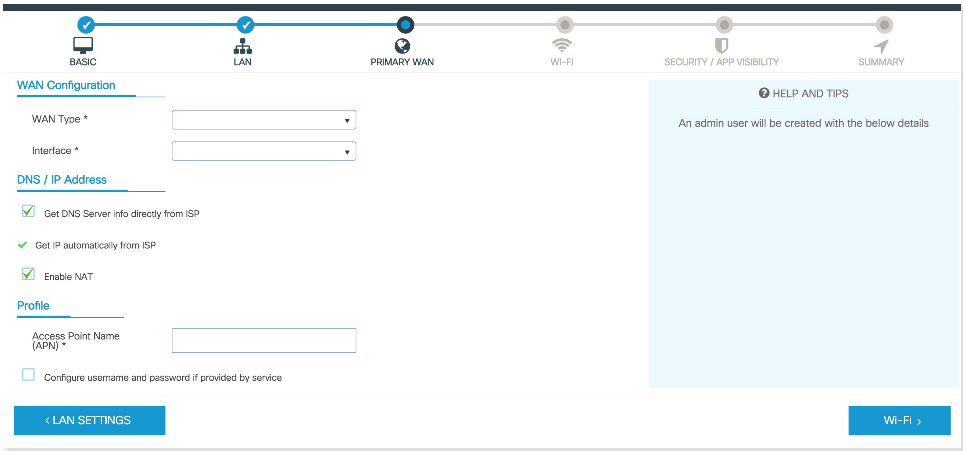
Step 4 Check the **Get IP automatically from ISP** check box to get the IP address information directly from the service provider. You can also manually enter the IP address and subnet mask.

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 username 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 can 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 username and password provided by the service provider.

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

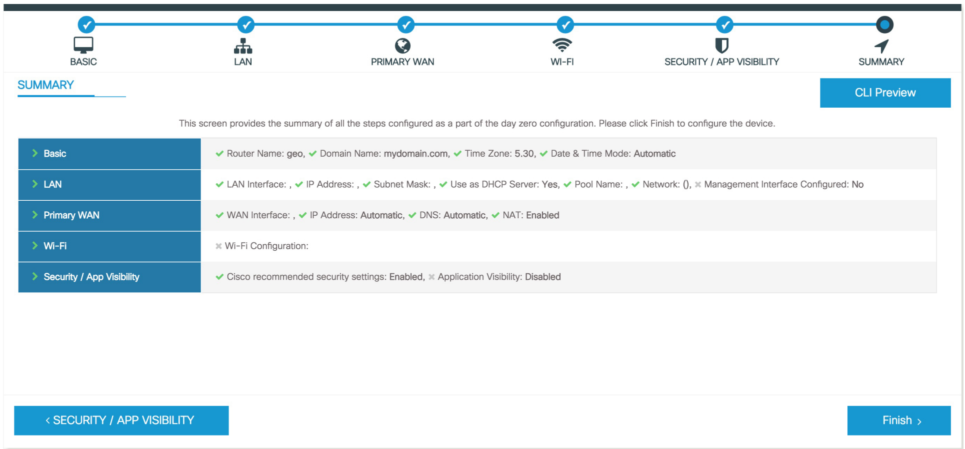
Configure Security Settings

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 20

Accessing and Using the GRUB Mode

Cisco Catalyst 8000V has a 16-bit configuration register in NVRAM. Each bit has the 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 which is comparable to the ROMMON options on other Cisco routers.

You can use the configuration register to:

- Force the router to boot into the GRUB mode (bootstrap program)
- Select a boot source and the default boot filename
- Recover a lost password

The following table describes the configuration register bits.

Table 32: 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. 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.

Note Entering the GRUB mode for Cisco Catalyst 8000V running on cloud solutions depends on the console access capabilities of the cloud provider. If the cloud provider provides limited access to console, you cannot access the GRUB mode for password recovery.

Note Use the 0x000 setting to configure the router to automatically enter the GRUB mode when the router reboots.

Accessing the GRUB Mode

Perform the following step to access the 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).

The following shows an example of entering GRUB mode.

```
Router(config)# config-register 0x0000

GNU GRUB  version 2.02

Minimal BASH-like line editing is supported. For the first word, TAB
lists possible command completions. Anywhere else TAB lists possible
device or file completions. ESC at any time exits.

grub> confreg 0x2102
```

If you enter a question mark at the grub> prompt, the system shows you the two options available - for either viewing the system help or for entering the **config register** 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.

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**.

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   Dec 18 2020 00:00:58 c8000v.tmp
 1580  -rw-     6462268   Dec 18 2020 06:14:02 c8000v-ata
63930368 bytes total (51007488 bytes free)
```

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-universalk9.17.04.01a.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:

```
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"
```

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.

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 2.02  (638K lower / 3143616K upper memory)
-------------------------------------------------------------------
 0: C8000v - packages.conf
 1: C8000v - c800v-packages-universalk9
 2: C8000v - GOLDEN IMAGE
-------------------------------------------------------------------
      Use the ^ and v keys to select which entry is highlighted.
      Press enter to boot the selected OS, or 'c' for a command-line.
    Highlighted entry is 0:
  Booting 'C8000v - packages.conf'
root (hd0,0)
 Filesystem type is ext2fs, partition type 0x83
kernel /packages.conf rw root=/dev/ram console=ttyS1,9600 max_loop=64 HARDWARE=
virtual SR_BOOT=harddisk:packages.conf
Calculating SHA-1 hash...done
SHA-1 hash:
        calculated   817e1716:e8e62778:7dd0b806:32db2bdd:13e51407
        expected     817e1716:e8e62778:7dd0b806:32db2bdd:13e51407
package header rev 1 structure detected
Calculating SHA-1 hash...done
SHA-1 hash:
        calculated   d4eaba99:34cbda63:26151233:9d0e9aa4:9c625302
        expected     d4eaba99:34cbda63:26151233:9d0e9aa4:9c625302
Package type:0x7531, flags:0x0
   [Linux-bzImage, setup=0x2e00, size=0x2c18c00]
   [isord @ 0x7e6d0000, 0x191f000 bytes]
```

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**

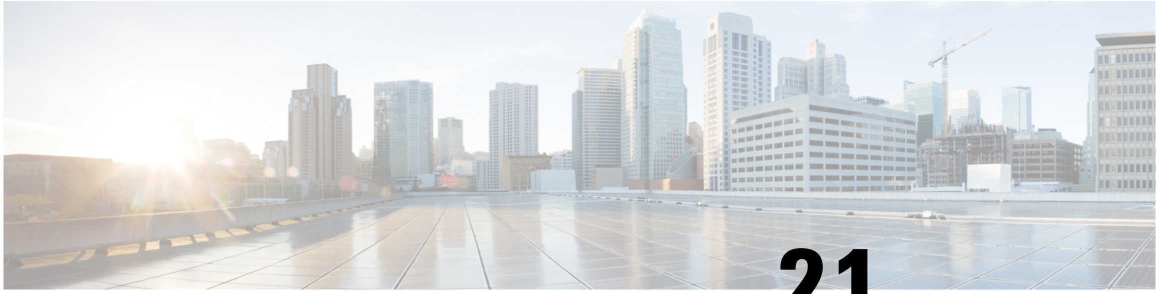
The new configuration register settings are saved to NVRAM, but they do not take effect until the next router reload or power cycle.

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 21

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 172.18.25.29 closed by remote host.
Connection to 172.18.25.29 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:C8000V 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: Dec 14 22:04:06 2020 UTC
  Registration Expires: Dec 15 21:59:05 2021 UTC

License Authorization:
  Status: AUTHORIZED on Dec 15 22:04:11 2020 UTC
  Last Communication Attempt: SUCCEEDED on Feb 15 22:04:11 2019 UTC
  Next Communication Attempt: Dec 17 22:04:11 2020 UTC
  Communication Deadline: Dec 16 21:58:10 2020 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 22

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.

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:192.168.1.1

VRF2 - RT:RT-EVPN-2, prefix:192.168.2.2

VRF3 and VRF4 are configured on the Cisco Catalyst 8000V instance. These two VRFs 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:192.168.3.3

VRF4 – RT for EVPN: RT-EVPN-4, RT for IP BGP: RT-4, prefix:192.168.4.4

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 10.4.4.1 255.255.255.224
bridge-domain 100
member vni 10100
!
interface nve1
source-interface loopback0
host-reachability protocol bgp
member vni 10100 vrf vrf3
!
router bgp 100
bgp router-id 10.11.11.11
no bgp default ipv4-unicast
neighbor 192.168.22.22 remote-as 200
neighbor 198.162.22.22 update-source loopback0
neighbor 198.162.22.22 ebgp-multihop 255
address-family ipv4 vrf vrf3
redistribute connected
neighbor 10.0.0.2 remote-as 300
neighbor 10.0.0.2 activate
neighbor 10.0.0.2 send-community both
advertise l2vpn evpn
exit-address-family
!
address-family l2vpn evpn
neighbor 198.162.22.22 activate
neighbor 198.162.22.22 send-community both
exit-address-family
end
```

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 10.11.11.11, local AS number 100
BGP table version is 8, main routing table version 8
7 network entries using 2408 bytes of memory
......
BGP activity 14/0 prefixes, 16/0 paths, scan interval 60 secs
7 networks peaked at 17:34:38 Aug 14 2019 CST (00:00:26.895 ago)
Neighbor        V           AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
198.162.22.22     4          200       6       5        4    0    0 00:01:23        4
Device#
```

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 23

Configuring Bridge Domain Interfaces

The Cisco C8000V routers support the bridge domain interface (BDI) feature for packaging Layer 2 Ethernet segments into Layer 3 IP address.

Restrictions for Bridge Domain Interfaces

The following are the restrictions pertaining to bridge domain interfaces:

- Only 4096 bridge domain interfaces are supported per system.
- For a bridge domain interface, the maximum transmission unit (MTU) size can be configured between 1500 and 9216 bytes.
- Bridge domain interfaces support only the following features:
  - IPv4 Multicast
  - QoS marking and policing. Shaping and queuing are not supported
  - IPv4 VRF
  - IPv6 unicast forwarding
  - Dynamic routing such as BGP, OSPF, EIGRP, RIP, IS-IS, and STATIC
  - Hot Standby Router Protocol (HSRP) from IOS XE 3.8.0 onwards.
  - Virtual Router Redundancy Protocol (VRRP) from IOS XE 3.8.0 onwards.
  - Flexible NetFlow

Note Flexible NetFlow is supported from Cisco IOS XE 17.7.1a and later releases.

- Bridge domain interfaces do not support the following features:
    - PPP over Ethernet (PPPoE)
    - Bidirectional Forwarding Detection (BFD) protocol
    - QoS
    - Network-Based Application Recognition (NBAR) or Advanced Video Coding (AVC)

Information About Bridge Domain Interface

Bridge domain interface is a logical interface that allows bidirectional flow of traffic between a Layer 2 bridged network and a Layer 3 routed network traffic. Bridge domain interfaces are identified by the same index as the bridge domain. Each bridge domain represents a Layer 2 broadcast domain. Only one bridge domain interface can be associated with a bridge domain.

Bridge domain interface supports the following features:

- IP termination
- Layer 3 VPN termination
- Address Resolution Protocol (ARP), G-ARP, and P-ARP handling
- MAC address assignment

Prior to configuring a bridge domain interface, you must understand the following concepts:

- Ethernet Virtual Circuit Overview
- Bridge Domain Interface Encapsulation
- Assigning a MAC Address
- Support for IP Protocols
- Support for IP Forwarding
- Packet Forwarding
- Bridge Domain Interface Statistics

Ethernet Virtual Circuit Overview

An Ethernet Virtual Circuit (EVC) is an end-to-end representation of a single instance of a Layer 2 service that is offered by a provider. It embodies the different parameters on which the service is being offered. In the Cisco EVC Framework, the bridge domains are made up of one or more Layer 2 interfaces known as service instances. A service instance is the instantiation of an EVC on a given port on a given router. Service instance is associated with a bridge domain based on the configuration.

An incoming frame can be classified as service instance based on the following criteria:

- Single 802.1Q VLAN tag, priority-tagged, or 802.1ad VLAN tag
- Both QinQ (inner and outer) VLAN tags, or both 802.1ad S-VLAN and C-VLAN tags

- Outer 802.1p CoS bits, inner 802.1p CoS bits, or both
- Payload Ethernet type (five choices are supported: IPv4, IPv6, PPPoE-all, PPoE-discovery, and PPPoE-session)

Service instance also supports alternative mapping criteria:

- Untagged—Mapping to all the frames lacking a 802.1Q or 802.1ad header
- Default—Mapping to all the frames

For more information on the EVC architecture, see the section *Configuring Ethernet Virtual Connections on the Cisco ASR 1000 Router* in the Carrier Ethernet Configuration Guide .

Bridge Domain Interface Encapsulation

Security Group classification includes both Source and Destination Group, which is specified by source SGT and DGT. SGT Based PBR feature provides the PBR route-map match clause for SGT/DGT based packet classification. SGT Based PBR feature supports configuration of unlimited number of tags, but it is recommended to configure the tags based on memory available in the platform.

An EVC provides the ability to employ different encapsulations on each Ethernet flow point (EFP) present in a bridge domain. A BDI egress point may not be aware of the encapsulation of an egress packet because the packet may have egressed from one or more EFPs with different encapsulations.

In a bridge domain, if all the EFPs have different encapsulations, the BDI must be untagged (using the no 802.1Q tag). Encapsulate all the traffic in the bridge domain (popped or pushed) at the EFPs. Configure rewrite at each EFP to enable encapsulation of the traffic on the bridge domain.

In a bridge domain, if all the EFPs have the same encapsulation, configure the encapsulations on the BDI using the encapsulation command. Enabling encapsulation at the BDI ensures effective pushing or popping of tags, thereby eliminating the need for configuring the rewrite command at the EFPs. For more information on configuring the encapsulations on the BDI, see the How to Configure a Bridge Domain Interface.

Assigning a MAC Address

All the bridge domain interfaces on the Cisco C8000V routers share a common MAC address. The first bridge domain interface on a bridge domain is allocated a MAC address. Thereafter, the same MAC address is assigned to all the bridge domain interfaces that are created in that bridge domain.

Note You can configure a static MAC address on a bridge domain interface using the **mac-address** command.

Support for IP Protocols

Bridge domain interfaces enable the Cisco C8000V routers to act as a Layer 3 endpoint on the Layer 2 bridge domain for the following IP-related protocols:

- ARP
- DHCP

- HTTP
- ICMP
- NTP
- RARP
- SNMP
- TCP
- Telnet
- TFTP
- UDP

Support for IP Forwarding

Bridge domain interface supports the following IP forwarding features:

- IPv4 input and output access control lists (ACL)
- IPv4 input and output QoS policies. The operations supported for the input and output service policies on a bridge domain interface are:
    - Classification
    - Marking
    - Policing
- IPv4 L3 VRFs

Packet Forwarding

A bridge domain interface provides bridging and forwarding services between the Layer 2 and Layer 3 network infrastructure.

Layer 2 to Layer 3

During a packet flow from a Layer 2 network to a Layer 3 network, if the destination MAC address of the incoming packet matches the bridge domain interface MAC address, or if the destination MAC address is a multicast address, the packet or a copy of the packet is forwarded to the bridge domain interface.

Note MAC address learning cannot not be performed on the bridge domain interface.

Layer 3 to Layer 2

When a packet arrives at a Layer 3 physical interface of a router, a route lookup action is performed. If route lookup points to a bridge domain interface, then the bridge domain interface adds the layer 2 encapsulation and forwards the frame to the corresponding bridge domain. The byte counters are updated.

During a Layer 2 lookup on a bridge domain to which the bridge domain interface belongs, the bridge domain forwards the packets to the correct service instance based on the destination MAC address.

Link States of a Bridge Domain and a Bridge Domain Interface

Bridge domain interface acts as a routable IOS interface on Layer 3 and as a port on a bridge domain. Both bridge domain interfaces and bridge domains operate with individual administrative states.

Shutting down a bridge domain interface stops the Layer 3 data service, but does not override or impact the state of the associated bridge domain.

Shutting down a bridge domain stops Layer 2 forwarding across all the associated members including service instances and bridge domain interfaces. The associated service instances influence the operational state of a bridge domain. Bridge domain interface cannot be operational unless one of the associated service instances is up.

Note Because a bridge domain interface is an internal interface, the operational state of bridge domain interface does not affect the bridge domain operational state.

BDI Initial State

The initial administrative state of a BDI depends on how the BDI is created. When you create a BDI at boot time in the startup configuration, the default administrative state for the BDI is up. It will remain in this state unless the startup configuration includes the shutdown command. This behavior is consistent with all the other interfaces. When you create a BDI dynamically at command prompt, the default administrative state is down.

BDI Link State

A BDI maintains a link state that comprises of three states: administratively down, operationally down, and up. The link state of a BDI is derived from two independent inputs: the BDI administrative state set by the corresponding users and the fault indication state from the lower levels of the interface states. It defines a BDI link state based on the state of the two inputs.

Fault Indication State	BDI Admin	
{start emdash}{end emdash}	**Shutdown**	**No Shutdown**
No faults asserted	Admin-down	Up
At least one fault asserted	Admin-down	Operationally-Down

Bridge Domain Interface Statistics

For virtual interfaces, such as the bridge domain interface, protocol counters are periodically queried from the QFP.

When packets flow from a Layer 2 bridge domain network to a Layer 3 routing network through the bridge domain interface, the packets are treated as bridge domain interface input packets and bytes. When packets arrive at a Layer 3 interface and are forwarded through the bridge domain interface to a Layer 2 bridge domain, the packets are treated as output packets and bytes, and the counters are updated accordingly.

A BDI maintains a standard set of Layer 3 packet counters as the case with all Cisco IOS interfaces. Use the show interface command to view the Layer 3 packet counters.

The convention of the counters is relative to the Layer 3 cloud. For example, input refers to the traffic entry to the Layer 3 cloud from the Layer 2 BD, while output refers to the traffic exit from the Layer 3 cloud to the Layer 2 BD.

Use the **show interfaces accounting** command to display the statistics for the BDI status. Use the **show interface** <*if-name*> command to display the overall count of the packets and bytes that are transmitted and received.

Creating or Deleting a Bridge Domain Interface

When you define an interface or subinterface for a Cisco IOS router, you name it and specify how it is assigned an IP address. You can create a bridge domain interface before adding a bridge domain to the system. This new bridge domain interface will be activated after the associated bridge domain is configured.

Note When a bridge domain interface is created, a bridge domain is automatically created.

When you create the bridge domain interface and the bridge domain, the system maintains the required associations for mapping the bridge domain-bridge domain interface pair.

The mapping of bridge domain and bridge domain interface is maintained in the system. The bridge domain interface uses the index of the associated bridge domain to show the association.

Bridge Domain Interface Scalability

The following table lists the bridge domain interface scalability numbers, based on the type of Cisco C8000V routers' Forwarding Processors (FPs).

Table 33: Bridge Domain Interface Scalability Numbers Based on the Type of Cisco C8000V routers' Forwarding Processor

Description			**0**
Maximum bridge domain interfaces per router			

Bridge-Domain Virtual IP Interface

The Virtual IP Interface (VIF) feature helps to associate multiple BDI interfaces with a BD instance. The BD-VIF interface inherits all the existing L3 features of IOS logical IP interface.

Note You must configure every BD-VIF interface with a unique MAC address and it should belong to a different VRF.

The Virtual IP Interface (VIF) feature has the following limitations:

- BD-VIF interface does not support IP multicast.

- Number of BD-VIF interfaces with automatically generated MAC address varies on the basis of platforms.
- BD-VIF Interface does not support MPLS.
- The maximum number of BD-VIF interfaces per bridge-domain and the total number of BD-VIF interface for per system vary based on the type of platforms.

The maximum number of BD-VIF supported on different platforms varies:

- ASR 1000 supports maximum 100 BD-VIF for a Bridge Domain
- CSR 1000v supports maximum 16 BD-VIF for a Bridge Domain
- ISR 4000 support maximum 16 BD-VIF for a Bridge Domain

From Cisco IOS XE 17.7.1a release, BD-VIF supports Flexible Netflow (FNF).

How to Configure a Bridge Domain Interface

To configure a bridge domain interface, perform the following steps:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface BDI** *{interface number}*
4. **encapsulation** *encapsulation dot1q <first-tag> [second-dot1q <second-tag>]*
5. Do one of the following:
6. **match security-group destination tag** *sgt-number*
7. **mac address** *{mac-address}*
8. **no shut**
9. **shut**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

Step 3	**interface BDI** *{interface number}*
	Example:
	`Router(config-if)# interface BDI3`
		Specifies a bridge domain interface.

	Command or Action	Purpose
Step 4	**encapsulation** *encapsulation dot1q <first-tag> [second-dot1q <second-tag>]* **Example:** `Router(config-if)# encapsulation dot1Q 1 second-dot1q 2`	Defines the encapsulation type. The example shows how to define dot1q as the encapsulation type.

Step 5	Do one of the following:
	Example:
	`ip address` *`ip-address mask`*
	Example:
	Example:
	`ipv6 address` *`{X:X:X:X::X`* **`link-local`**`|` *`X:X:X:X::X/prefix [`***`anycast`** `|` **`eui-64`***`] |`* **`autoconfig`** *`[`***`default`***`]}`*
	Example:
	`Router(config-if)# ip address 10.2.2.1 255.255.255.0`
	Example:
	Example:
	`Router(config-if)# ipv6 address AB01:CD1:123:C::/64 eui-64`
		Specifies either the IPv4 or IPv6 address for the bridge domain interface.

Step 6	**match security-group destination tag** *sgt-number* **Example:** `Router(config-route-map)# match security-group destination tag 150`	Configures the value for security-group destination security tag.

Step 7	**mac address** *{mac-address}*
	Example:
	`Router(config-if)# mac-address 1.1.3`
		Specifies the MAC address for the bridge domain interface.

Step 8	**no shut** **Example:** `Router(config-if)# no shut`	Enables the bridge domain interface.
Step 9	**shut** **Example:**	Disables the bridge domain interface.

	Command or Action	Purpose
	`Router(config-if)# shut`	

Example

The following example shows the configuration of a bridge domain interface at IP address 10.2.2.1 255.255.255.0:

```
Router# configure terminal
Router(config)# interface BDI3
Router(config-if)# encapsulation dot1Q 1 second-dot1q 2
Router(config-if)# ip address 10.2.2.1 255.255.255.0
Router(config-if)# mac-address 1.1.3
Router(config-if)# no shut
Router(config-if)# exit
```

Displaying and Verifying Bridge Domain Interface Configuration

SUMMARY STEPS

1. **enable**
2. **show interfaces bdi**
3. **show platform software interface fp active name**
4. **show platform hardware qfp active interface if-name**
5. **debug platform hardware qfp feature**
6. **platform trace runtime process forwarding-manager module**
7. **platform trace boottime process forwarding-manager module interfaces**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	**show interfaces bdi** **Example:** `Router# show interfaces BDI3`	Displays the configuration summary of the corresponding BDI.
Step 3	**show platform software interface fp active name** **Example:** `Router# show platform software interface fp active name BDI4`	Displays the bridge domain interface configuration in a Forwarding Processor.

	Command or Action	Purpose
Step 4	**show platform hardware qfp active interface if-name** **Example:** `Router# show platform hardware qfp active interface if-name BDI4`	Displays the bridge domain interface configuration in a data path.
Step 5	**debug platform hardware qfp feature** **Example:** `Router# debug platform hardware qfp active feature l2bd client all`	The selected CPP L2BD Client debugging is on.
Step 6	**platform trace runtime process forwarding-manager module** **Example:** `Router(config)# platform trace runtime slot F0 bay 0 process forwarding-manager module interfaces level info`	Enables the Forwarding Manager Route Processor and Embedded Service Processor trace messages for the Forwarding Manager process.
Step 7	**platform trace boottime process forwarding-manager module interfaces** **Example:** `Router(config)# platform trace boottime slot R0 bay 1 process forwarding-manager forwarding-manager level max`	Enables the Forwarding Manager Route Processor and Embedded Service Processor trace messages for the Route Processor Forwarding Manager process during bootup.

What to do next

For additional information on the commands and the options available with each command, see the Cisco IOS Configuration Fundamentals Command Reference Guide.

Configuring Bridge-Domain Virtual IP Interface

```
enable
configure terminal
[no] interface BD-VIF interface-number
  [ [no] vrf forwarding vrf-name]
  [ [no] mac address mac-address]
  [ [no] ip address ip-address mask]
  [ [no] ipv6 address {X:X:X:X::X link-local| X:X:X:X::X/prefix [anycast | eui-64] |
autoconfig [default]}]

exit
```

To delete BD-VIF interface, use the 'no' form of the command.

Associating VIF Interface with a Bridge Domain

```
enable
configure terminal
bridge-domain bridge-domain number
[no] member BD-VIF interface-number
exit
```

To dissociate the VIF interface, use the 'no' form of the command.

Verifying Bridge-Domain Virtual IP Interface

All existing show commands for interface and IP interface can be used for the BD-VIF interface.

show interface bd-vif *bd-vif-id*

show ip interface bd-vif *bd-vif-id*

show bd-vif interfaces in fman-fp

show pla sof inter fp ac brief | i BD_VIF

Example Configuration Bridge-Domain Virtual IP Interface

```
Detail sample:

interface Port-channel1
mtu 9000
no ip address
 !Ethernet service endpoint one per neutron network
service instance 1756 ethernet
  description 4e8e5957-649f-477b-9e5b-f1f75b21c03c
  encapsulation dot1q 1756
  rewrite ingress tag pop 1 symmetric
  bridge-domain 1756
!
interface BD-VIF5001
no shutdown
vrf forwarding vrf5001
ip address 10.0.0.1 255.255.255.0
interface BD-VIF5002
no shutdown
vrf forwarding vrf5002
ip address 10.0.0.2 255.255.255.0

bridge-domain 1756
member Port-channel1 service-instance 1756
member bd-vif5001
member bd-vif5002
```

Configuring Flexible NetFlow over a Bridge Domain Virtual IP Interface

SUMMARY STEPS

1. **enable**
2. **configure terminal**

3. **interface** *type number*
4. {**ip** | **ipv6**}**flow monitor** *monitor-name* [**sampler** *sampler-name*] {**input** | **output**}
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device (config)# interface BD-VIF 100`	Specifies an interface and enters interface configuration mode. Enter the BD-VIF number.

Step 4	{**ip** | **ipv6**}**flow monitor** *monitor-name* [**sampler** *sampler-name*] {**input** | **output**}
	Example:
	`Device(config-if)# ip flow monitor FLOW-MONITOR-1 input`
		Enables a Flexible NetFlow flow monitor for IP traffic that the router is receiving or transmitting on the interface.

Step 5	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to privileged EXEC mode.

Examples: Flexible NetFlow over a Bridge Domain Virtual IP Interface

The following is a sample output for the **show platform hardware qfp active interface if-name** command showing the QFP information and flow direction for flow monitors. The table below provides the key to the CLI output.

Configuration	Output
ip flow monitor <monitor-name> input	IPV4_INPUT_FNF_FIRST IPV4_INPUT_FNF_FINAL
ip flow monitor <monitor-name> output	IPV4_BDI_OUTPUT_FNF_FINAL
ipv6 flow monitor <monitor-name> input	IPV6_INPUT_FNF_FIRST IPV6_INPUT_FNF_FINAL
ipv6 flow monitor <monitor-name> output	IPV6_BDI_OUTPUT_FNF_FINAL

```
Device# show run interface bd-vif2
Building configuration...

Current configuration: 227 bytes
!
interface BD-VIF2
vrf forwarding vrf1
ip flow monitor test1 input
ip flow monitor test1 output
ip address 10.11.11.11 255.255.255.0
ipv6 flow monitor test2 input
ipv6 flow monitor test2 output
ipv6 address 2001:DB8::1/32
end

Device# show platform hardware qfp active interface if-name BD-VIF 2
General interface information
  Interface Name: BD-VIF2
  Interface state: VALID
  Platform interface handle: 20
  QFP interface handle: 17
  Rx uidb: 262138
  Tx uidb: 262127
  Channel: 0
Interface Relationships

BGPPA/QPPB interface configuration information
  Ingress: BGPPA/QPPB not configured. flags: 0000
  Egress: BGPPA not configured. flags: 0000

ipv4_input enabled.
ipv4_output enabled.
ipv6_input enabled.
ipv6_output enabled.
layer2_input enabled.
layer2_output enabled.
ess_ac_input enabled.

Features Bound to Interface:
2 GIC FIA state
66 PUNT INJECT DB
70 cpp_l2bd_svr
43 icmp_svr
45 ipfrag_svr
46 ipreass_svr
47 ipv6reass_svr
44 icmp6_svr
58 stile
Protocol 0 - ipv4_input
FIA handle - CP:0x55a7f59df038 DP:0x3fff1000
  IPV4_INPUT_DST_LOOKUP_ISSUE (M)
  IPV4_INPUT_ARL_SANITY (M)
  IPV4_INPUT_SRC_LOOKUP_ISSUE
  IPV4_INPUT_DST_LOOKUP_CONSUME (M)
  IPV4_INPUT_SRC_LOOKUP_CONSUME
  IPV4_INPUT_FOR_US_MARTIAN (M)
  IPV4_INPUT_STILE_LEGACY
  IPV4_INPUT_FNF_FIRST
  IPV4_INPUT_LOOKUP_PROCESS (M)
  IPV4_INPUT_FNF_FINAL
  IPV4_INPUT_IPOPTIONS_PROCESS (M)
  IPV4_INPUT_GOTO_OUTPUT_FEATURE (M)
Protocol 1 - ipv4_output
FIA handle - CP:0x55a7f59df0d8 DP:0x3ffeff00
```

```
  IPV4_VFR_REFRAG (M)
  IPV4_OUTPUT_SRC_LOOKUP_ISSUE
  IPV4_OUTPUT_L2_REWRITE (M)
  IPV4_OUTPUT_SRC_LOOKUP_CONSUME
  IPV4_OUTPUT_STILE_LEGACY
  IPV4_OUTPUT_FRAG (M)
  IPV4_BDI_OUTPUT_FNF_FINAL.
  BDI_VLAN_TAG_ATTACH_AND_LAYER2_LOOKUP_GOTO
  LAYER2_BRIDGE
  BDI_OUTPUT_GOTO_OUTPUT_FEATURE
  IPV4_OUTPUT_DROP_POLICY (M)
  DEF_IF_DROP_FIA (M)
Protocol 6 - ipv6_input
FIA handle - CP:0x55a7f59dee58 DP:0x3fff4300
  IPV6_INPUT_SANITY_CHECK (M)
  IPV6_INPUT_DST_LOOKUP_ISSUE (M)
  IPV6_INPUT_SRC_LOOKUP_ISSUE
  IPV6_INPUT_ARL (M)
  IPV6_INPUT_DST_LOOKUP_CONT (M)
  IPV6_INPUT_SRC_LOOKUP_CONT
  IPV6_INPUT_DST_LOOKUP_CONSUME (M)
  IPV6_INPUT_SRC_LOOKUP_CONSUME
  IPV6_INPUT_STILE_LEGACY
  IPV6_INPUT_FNF_FIRST
  IPV6_INPUT_FOR_US (M)
  IPV6_INPUT_LOOKUP_PROCESS (M)
  IPV6_INPUT_FNF_FINAL
  IPV6_INPUT_LINK_LOCAL_CHECK (M)
  IPV6_INPUT_GOTO_OUTPUT_FEATURE (M)
Protocol 7 - ipv6_output
FIA handle - CP:0x55a7f59dee08 DP:0x3fff4b80
  IPV6_VFR_REFRAG (M)
  IPV6_OUTPUT_SRC_LOOKUP_ISSUE
  IPV6_OUTPUT_SRC_LOOKUP_CONT
  IPV6_OUTPUT_SRC_LOOKUP_CONSUME
  IPV6_OUTPUT_L2_REWRITE (M)
  IPV6_OUTPUT_STILE_LEGACY
  IPV6_OUTPUT_FRAG (M)
  IPV6_BDI_OUTPUT_FNF_FINAL
  BDI_VLAN_TAG_ATTACH_AND_LAYER2_LOOKUP_GOTO
  LAYER2_BRIDGE
  BDI_OUTPUT_GOTO_OUTPUT_FEATURE
  IPV6_OUTPUT_DROP_POLICY (M)
  DEF_IF_DROP_FIA (M)
□
```

The following is a sample out of the **show flow monitor** [[**name**] [**cache** [**format** {**csv** | **record** | **table**}]] [**statistics**]] command showing the cache output in record format.

```
Device# show flow monitor name FLOW-MONITOR-1 cache format record

Cache type:  Normal
Cache size:  1000
Current entries:  4
High Watermark:  4
Flows added:  101
Flows aged:  97
- Active timeout (1800 secs) 3
- Inactive timeout  (15 secs) 94
- Event aged  0
- Watermark aged  0
- Emergency aged
IPV4 DESTINATION ADDRESS:
198.51.100.1 0
ipv4 source address: 10.10.11.1
```

```
trns source port: 25
trns destination port: 25
counter bytes: 72840
counter packets: 1821
IPV4 DESTINATION ADDRESS: 198.51.100.2
ipv4 source address: 10.10.10.2
trns source port: 20
trns destination port: 20
counter bytes: 3913860
counter packets: 7326
IPV4 DESTINATION ADDRESS: 198.51.100.200
ipv4 source address: 192.168.67.6
trns source port: 0
trns destination port: 3073
counter bytes: 51072
counter packets: 1824


Device# show flow monitor name FLOW-MONITOR-2 cache format record

Cache type:  Normal
Cache size: 1000
Current entries: 2
High Watermark: 3
Flows added: 95
Flows aged: 93
- Active timeout (1800 secs) 0
- Inactive timeout (15 secs) 93
- Event aged 0
- Watermark aged 0
- Emergency aged 0
IPV6 DESTINATION ADDRESS: 2001:DB8:0:ABCD::1
ipv6 source address: 2001:DB8:0:ABCD::2
trns source port: 33572
trns destination port: 23
counter bytes: 19140
counter packets: 349
IPV6 DESTINATION ADDRESS: FF02::9
ipv6 source address: 2001:DB8::A8AA:BBFF:FEBB

trns source port: 521
trns destination port: 521
counter bytes: 92
counter packets: 1
```

The following is a sample out of the **show flow interface** command showing the flow status for an interface.

```
Device# show flow interface BD-VIF2001

Interface GigabitEthernet0/0/0
FNF: monitor: FLOW-MONITOR-1
direction: Input
traffic(ip): on
FNF: monitor: FLOW-MONITOR-2
direction:    Input traffic(ipv6): on

Device# show flow interface BD-VIF2002

Interface GigabitEthernet1/0/0
FNF: monitor: FLOW-MONITOR-1
direction: Output
traffic(ip): on
FNF: monitor: FLOW-MONITOR-2
direction:    Input traffic(ipv6): on
```

The following is a sample output of the **show platform hardware qfp active interface if-name | in FNF** command showing the QFP information and flow direction for flow monitors in Flexible NetFlow configuration. The table below provides the key to the CLI output.

Configuration	Output
ip flow monitor <monitor-name> input	IPV4_INPUT_FNF_FIRST IPV4_INPUT_FNF_FINAL
ip flow monitor <monitor-name> output	IPV4_BDI_OUTPUT_FNF_FINAL
ipv6 flow monitor <monitor-name> input	IPV6_INPUT_FNF_FIRST IPV6_INPUT_FNF_FINAL
ipv6 flow monitor <monitor-name> output	IPV6_BDI_OUTPUT_FNF_FINAL

```
Device# show run interface bd-vif2
Building configuration...

Current configuration : 227 bytes
!
interface BD-VIF2
vrf forwarding vrf1
ip flow monitor test1 input
ip flow monitor test1 output
ip address 10.11.11.11 255.255.255.0
ipv6 flow monitor test2 input
ipv6 flow monitor test2 output
ipv6 address 2001::8/64
end
 Device# show platform hardware qfp active interface if-name BD-VIF 2  | in FNF
  IPV4_INPUT_FNF_FIRST
  IPV4_INPUT_FNF_FINAL
  IPV4_BDI_OUTPUT_FNF_FINAL.
  IPV6_INPUT_FNF_FIRST
  IPV6_INPUT_FNF_FINAL
  IPV6_BDI_OUTPUT_FNF_FINAL
```

The **clear flow monitor name** *monitor-name* [**cache** [**force-export**] | **force-export** | **statistics**] command clears a Flexible NetFlow flow monitor, flow monitor cache, or flow monitor statistics, and can be used to force the export of the data in the flow monitor cache.

For more details on configuring Flexible NetFlow, see the Flexible NetFlow Configuration Guide, Cisco IOS XE 17.

Additional References

Related Documents

Related Topic	Document Title
Configuring Ethernet Virtual Connections on the Cisco ASR 1000 Series Aggregation Services Routers	Carrier Ethernet Configuration Guide
EVC Quality of Service	http://www.cisco.com/en/US/docs/ios/ios_xe/qos/configuration/guide/qos_evc_xe.html

MIBs

MIB	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

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.	https://www.cisco.com/c/en_in/support/index.html

Feature Information for Configuring Bridge Domain Interfaces

The following table 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 www.cisco.com/go/cfn. An account on Cisco.com is not required.

Note The table below lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Table 34: Feature Information for Configuring Bridge Domain Interfaces

Feature Name	Releases	Feature Information
Configuring Bridge Domain Interface	Cisco IOS XE Cupertino 17.7.1a	This feature was introduced on the Cisco C8000V routers.
Bridge-Domain Virtual IP Interface	Cisco IOS XE Cupertino 17.7.1a	This feature was introduced on the Cisco C8000V routers. The Bridge-Domain Virtual IP Interface (VIF) now connects multiple Bridge Domain Interfaces (BDI) with a single BD instance so that each IP subnet within an L2 network can be associated with a single VRF.
Flexible NetFlow (FNF) on Bridge-Domain Virtual IP Interface (BD-VIF)	Cisco IOS XE Cupertino 17.7.1a	This feature was introduced on the Cisco C8000V routers. The following command was introduced: **{ip | ipv6} flow monitor** *monitor-name* **[sampler** *sampler-name***] {input | output}**

CHAPTER 24

Configuring MTP Software Support

A Media Termination Point (MTP) software device is an essential component of large-scale deployments of Cisco Unified Communications Manager (CUCM). In these deployments, the software MTP bridges the media streams between two connections by allowing the CUCM to relay the calls that are routed through Session Initiation Protocol (SIP) or H.323 endpoints through Skinny Client Control Protocol (SCCP) commands. The SCCP commands allow the CUCM to establish MTP for call signaling.

From Cisco IOS XE 17.8.1, you can configure the support for software MTP on Cisco Catalyst 8000V devices. If you use voice functionalities with your Cisco Catalyst 8000V device, you can leverage software MTP to enable and use supplementary services, such as Call Park and Call Transfer routed through an H.323 endpoint or an H.323 gateway.

Benefits

Configuring software MTP in Cisco Catalyst 8000V allows you to:

- Register a Cisco Catalyst 8000V instance with the Unified CM as a Trusted Relay Point.
- Leverage the SWMTP support when one of the end points does not support DTMF signaling.

Prerequisites for Configuring Support for Software MTP

- Configure codec and packetization in the inbound-call legs and the outbound-call legs.

SRTP-DTMF Interworking

From Cisco IOS XE 17.10.1a, Secure Real-time Transport Protocol (SRTP) Dual-Tone Multi-Frequency (DTMF) interworking is supported with Software MTP in pass through mode. SMTP supports DTMF Interworking for nonsecure calls, and this feature adds support for SRTP DTMF interworking for secure calls.

CUCM support for this feature is expected to be implemented in a later release.

Restrictions for SRTP-DTMF Interworking

- The SRTP-DTMF Interworking feature supports only the codec-passthrough format.
- The SRTP-DTMF Interworking feature does not support multiple concurrent Synchronised Sources (SSRCs) with the same destination IP and port.
- The calls that support SRTP-DTMF Interworking may have a minor performance impact as compared to calls supported on nonsecure DTMF interworking.

Supported Platforms for SRTP-DTMF Interworking

From Cisco IOS XE 17.10.1a, the following platforms support SRTP DTMF interworking with SMTP:

- Cisco 4461 Integrated Services Router (ISR)
- Cisco Catalyst 8200 Edge Series Platforms
- Cisco Catalyst 8300 Edge Series Platforms
- Cisco Catalyst 8000V Edge Software

Configuring Support for Software MTP

To enable and configure support for software MTP, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **sccp local** *interface-type interface-number* [**port** *port-number*]
4. **sccp ccm** {*ipv4-address* | *ipv6-address* | *dns*} **identifier** *identifier-number* [**port** *port-number*] **version** *version-number*
5. **sccp**
6. **sccp ccm group** *group-number*
7. **associate ccm** *identifier-number* **priority** *number*
8. **associate profile** *profile-identifier* **register** *device-name*
9. **dspfarm profile** *profile-identifier* {**conference** | **mtp** | **transcode**} [**security**]
10. **trustpoint** *trustpoint-label*
11. **codec** *codec*
12. **maximum sessions** {**hardware** | **software**} *number*
13. **associate application sccp**
14. **no shutdown**

DETAILED STEPS

	Command or Action	Purpose
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	**sccp local** *interface-type interface-number* [**port** *port-number*] **Example:** `Router(config)# sccp local gigabitethernet0/0/0`	Selects the local interface that SCCP applications (transcoding and conferencing) use to register with the Cisco UCM. • *interface type* : The interface address or a virtual-interface address such as Ethernet. • *interface number* : The interface number that the SCCP application uses to register with the Unified CM. • (Optional) **port** *port-number*: The port number used by the selected interface. The applicable range is 1025 to 65535, and the default is 2000.
Step 4	**sccp ccm** {*ipv4-address* | *ipv6-address* | *dns*} **identifier** *identifier-number* [**port** *port-number*] **version** *version-number* **Example:** `Router(config)# sccp ccm 10.1.1.1 identifier 1 version 7.0+`	Adds a Unified CM server to the list of available servers and sets the following parameters: • *ipv4-address* : The IP version 4 address of the Cisco UCM server. • *ipv6-address* : The IP version 6 address of the Cisco UCM server. • *dns* : The DNS name. • **identifier** : The number that identifies the Unified CM server. The applicable range is 1 to 65535. • **port** *port-number* (Optional): The TCP port number. The applicable range is 1025 to 65535, and the default is 2000. • **version** *version-number* : The Unified CM version. The valid versions are 3.0, 3.1, 3.2, 3.3, 4.0, 4.1, 5.0.1, 6.0, and 7.0+.
Step 5	**sccp** **Example:**	Enables the SCCP and its related applications (transcoding and conferencing).

	Command or Action	Purpose
	`Router(config)# sccp`	
Step 6	**sccp ccm group** *group-number* **Example:** `Router(config)# sccp ccm group 10`	Creates a Unified CM group and enters the SCCP Unified CM configuration mode. • *group-number* : Identifies the Cisco Unified CM group. The applicable range is 1 to 50.
Step 7	**associate ccm** *identifier-number* **priority** *number* **Example:** `Router(config-sccp-ccm)# associate ccm 10 priority 3`	Associates a Unified CM with a group and establishes its priority within the group. • *identifier-number* : The Unified CM identifier. The applicable range is 1 to 65535. • **priority** *number* : The priority of the Unified CM within the Unified CM group. The applicable range is 1 to 4. The highest priority is 1.
Step 8	**associate profile** *profile-identifier* **register** *device-name* **Example:** `Router(config-sccp-ccm)# associate profile 1 register MTP0011`	Associates a Digital Signal Processor (DSP) farm profile with a Unified CM group. • *profile-identifier* : The DSP farm profile. The applicable range is 1 to 65535. • **register** *device-name* : The device name in Unified CM. A maximum of 15 characters can be entered for the device name.
Step 9	**dspfarm profile** *profile-identifier* {**conference** | **mtp** | **transcode**} [**security**] **Example:** `Router(config-sccp-ccm)# dspfarm profile 1 mtp`	Enters the DSP farm profile configuration mode and defines a profile for the DSP farm services. • *profile-identifier* : The number that uniquely identifies a profile. The applicable range is 1 to 65535, and there is no default. • **conference** : Enables a profile for conferencing. • **mtp** : Enables a profile for MTP. • **transcode** : Enables a profile for transcoding. • **security** (Optional): Enables a profile for secure DSP farm services. For more information on configuration examples, see section Sample Software MTP Support Configuration, on page 243.
Step 10	**trustpoint** *trustpoint-label* **Example:** `Router(config-dspfarm-profile)# trustpoint dspfarm`	(Optional) Associates a trustpoint with a DSP farm profile.
Step 11	**codec** *codec*	Specifies the codecs supported by a DSP farm profile.

	Command or Action	Purpose
	Example: `Router(config-dspfarm-profile)# codec g711ulaw`	• codec-type: Specifies the preferred codec. Enter ? for a list of supported codecs. Repeat this step for each supported codec.
Step 12	**maximum sessions** {**hardware** | **software**} *number* **Example:** `Router(config-dspfarm-profile)# maximum sessions software 10`	Specifies the maximum number of sessions supported by the profile. • **hardware** : The number of sessions that the MTP hardware resources support. • **software** : The number of sessions that the MTP software resources support. • *number* : The number of sessions that are supported by the profile. The applicable range is 0 to x, and the default is 0. The value of x is determined at runtime depending on the number of resources available with the resource provider.
Step 13	**associate application sccp** **Example:** `Router(config-dspfarm-profile)# associate application sccp`	Associates SCCP to the DSP farm profile.
Step 14	**no shutdown** **Example:** `Router(config-dspfarm-profile)# no shutdown`	Changes the status of the interface to the UP state.

Sample Software MTP Support Configuration

The following output is a sample of the software MTP support configuration in a Cisco Catalyst 8000V device:

```
sccp local GigabitEthernet1
sccp ccm 9.35.46.100 identifier 1 priority 1 version 7.0
!
sccp ccm group 1
 bind interface GigabitEthernet1
 associate ccm 1 priority 1
 associate profile 10 register SWMTP1
 associate profile 1 register c8kvsmall-mtp1
 associate profile 2 register c8kv-sec-swmtp1
!
!
!
dspfarm profile 1 mtp
 codec g711ulaw
 maximum sessions software 20000
 associate application SCCP
```

The following example shows a sample configuration for the SRTP-DTMF Interworking feature-with secure dspfarm profile:

```
sccp local GigabitEthernet0/0/0
sccp ccm 172.18.151.125 identifier 1 version 7.0
sccp
!
sccp ccm group 1
 bind interface GigabitEthernet0/0/0
 associate ccm 1 priority 1
 associate profile 1 register Router
!
dspfarm profile 1 mtp security
 trustpoint IOSCA
 codec g711ulaw
 codec pass-through
 tls-version v1.2
 maximum sessions software 5000
 associate application SCCP
```

Note SR-TP traffic can pass through an SMTP resource when the dspfarm profile is provisioned with codec pass-through, and if it does not have TLS and security-related configuration. For traffic flows that require SRTP-DTMF interworking support, the SMTP dspfarm profile must include the **security** keyword and the TLS and codec pass-through configuration. This dspfarm resource profile can also pass through SRTP traffic independent of SRTP-DTMF interworking support.

Verifying Software MTP Support

To verify whether you have successfully configured the support for SWMTP in your Cisco Catalyst 8000V device, run the **show sccp** command:

```
Router# show sccp

SCCP Admin State: UP
Gateway IP Address: 10.13.40.157, Port Number: 2000
IP Precedence: 5
User Masked Codec list: None
Call Manager: 10.13.40.148, Port Number: 2000
                Priority: N/A, Version: 6.0, Identifier: 1
                Trustpoint: N/A
```

To verify the dspfarm profile, run the **show dspfarm profile** command:

```
Router# show dspfarm profile 1
Dspfarm Profile Configuration

 Profile ID = 1, Service = MTP, Resource ID = 1
 Profile Service Mode : Non Secure
 Profile Admin State : UP
 Profile Operation State : RESOURCE ALLOCATED
 Application : SCCP   Status : NOT ASSOCIATED
 Resource Provider : NONE   Status : NONE
 Total Number of Resources Configured : 20000
 Total Number of Resources Available : 20000
 Total Number of Resources Out of Service : 0
 Total Number of Resources Active : 0
 Hardware Configured Resources : 0
 Hardware Resources Out of Service: 0
 Software Configured Resources : 20000

 Number of Hardware Resources Active : 0
 Number of Software Resources Active : 0
 Codec Configuration: num_of_codecs:1
 Codec : g711ulaw, Maximum Packetization Period : 30
```

To verify information about the secure dspfarm profile status, use the **show dspfarm profile** command and check that the secure service mode is set:

```
Router# show dspfarm profile 2
Dspfarm Profile Configuration
 Profile ID = 2, Service = MTP, Resource ID = 2
 Profile Service Mode : secure
 Trustpoint : IOSCA
 TLS Version  : v1.2
 TLS Cipher   : AES128-SHA
 Profile Admin State : UP
 Profile Operation State : ACTIVE
 Application : SCCP   Status : ASSOCIATED
 Resource Provider : NONE   Status : NONE
 Total Number of Resources Configured : 8000
 Total Number of Resources Available : 8000
 Total Number of Resources Out of Service : 0
 Total Number of Resources Active : 0
 Hardware Configured Resources : 0
 Hardware Resources Out of Service: 0
 Software Configured Resources : 8000
 Number of Hardware Resources Active : 0
 Number of Software Resources Active : 0
 Codec Configuration: num_of_codecs:2
 Codec : pass-through, Maximum Packetization Period : 0
 Codec : g711ulaw, Maximum Packetization Period : 30
```

To verify the call connection between the endpoints, run the **show sccp connection details** command. This command shows that the connection is successfully established. This is indicated through the active connections and call legs at the end of the configuration output:

```
Router# show sccp connection details

bridge-info(bid, cid) - Normal bridge information(Bridge id, Calleg id)

mmbridge-info(bid, cid) - Mixed mode bridge information(Bridge id, Calleg id)

sess_id    conn_id    call-id   codec  pkt-period  dtmf_method     type    bridge-info
mmbridge-info  srtp_cryptosuite  dscp
call_ref   spid       conn_id_tx                                           (bid, cid)
(bid, cid)
16782237   16777254   110       g711u  20          rfc2833_pthru   rtpspi  (40,0)
N/A            N/A               184
29751839   16777216   -
16782237   16777253   109       g711u  20          rfc2833_report  rtpspi  (40,0)
N/A            N/A               184
29751839   16777216   -
Total number of active session(s) 1, connection(s) 2, and callegs 2
```

For SMTP secure DTMF, the **show sccp connections** command displays the codec type (pass-th), the s-type (s-mtp), and information about the DTMF method (rfc2833_pthru):

```
Router#sh sccp connections

sess_id    conn_id    stype   mode      codec    sport  rport   ripaddr conn_id_tx
dtmf_method
```

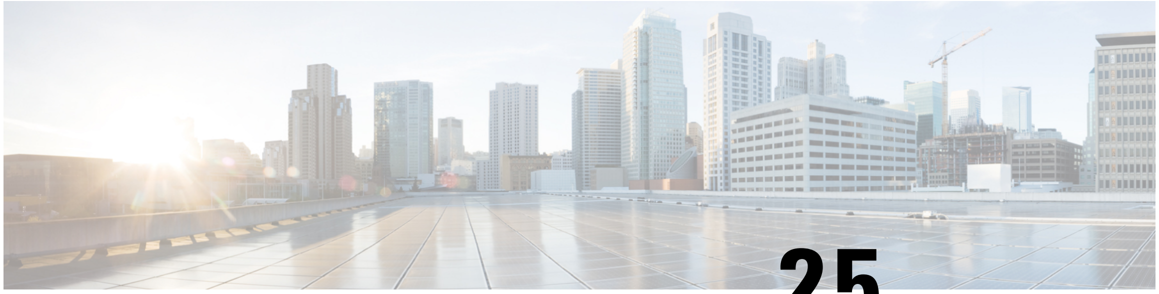
```
16791234   16777308   s-mtp   sendrecv  pass_th  8006   24610   172.18.153.37
rfc2833_pthru
16791234   16777306   s-mtp   sendrecv  pass_th  8004   17576   172.18.154.2
rfc2833_report

Total number of active session(s) 1, and connection(s) 2
```

To display information about RTP connections, use the **show rtpspi call** command:

```
Router# show rtpspi call
RTP Service Provider info:
No. CallId  dstCallId   Mode      LocalRTP  RmtRTP  LocalIP     RemoteIP    SRTP
1    22      19         Snd-Rcv   7242      17510   0x90D080F   0x90D0814    0
2    19      22         Snd-Rcv   18050     6900    0x90D080F   0x90D080F    0
```

If SRTP DTMF interworking is active, the SRTP field shows a non-zero value:

```
Router# show rtpspi call
RTP Service Provider info:
No. CallId  dstCallId   Mode      LocalRTP  RmtRTP  LocalIP     RemoteIP    SRTP
1   13      14          Snd-Rcv   8024      18270   0xA7A5355   0xAC129A02   1
2   14      13          Snd-Rcv   8026      24768   0xA7A5355   0xAC129925   1
```



CHAPTER 25

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.

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 10.92.2.1 255.255.255.0
 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 GigabitEthernet0/0/1
 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 192.168.90.3 255.255.255.0
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 192.168.2.1 255.255.0.0
physical-interface GigabitEthernet0/0/1
mode bypass
```

```
interface vmi3//configure the virtual multi interface
 ip address 192.168.3.1 255.255.0.0
 physical-interface GigabitEthernet0/0/1
mode bypass
```

Configure OSPF Routing

```
router ospfv3 1
 router-id 192.168.1.1
!
 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 192.168.12.3 192.168.12.254
```

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=192.168.2.2, 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=192.168.2.2, 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 192.168.3.3)

Neighbor ID     Pri   State           Dead Time   Interface ID    Interface
192.168.1.1           0   FULL/  -        00:01:32    19              Virtual-Access2.1

          OSPFv3 1 address-family ipv6 (router-id 192.168.3.3)

Neighbor ID     Pri   State           Dead Time   Interface ID    Interface
192.168.1.1           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

      192.168.0.3/8 is variably subnetted, 3 subnets, 2 masks
C        192.168.3.0/24 is directly connected, Virtual-Access2.1
O        192.168.4.0/32 [110/1] via 192.168.4.0, 00:00:03, Virtual-Access2.1
L        192.168.5.0/32 is directly connected, Virtual-Access2.1
      192.168.0.5/32 is subnetted, 1 subnets
C        192.168.2.21 is directly connected, Virtual-Access2.1
```