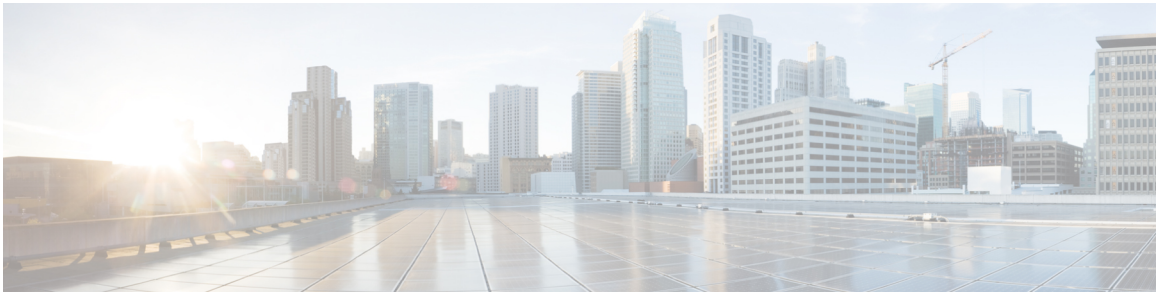
Deploying Cisco Catalyst 8000V Edge Software on Amazon Web Services

First Published: 2021-01-30

Last Modified: 2024-04-30

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

Full Cisco Trademarks with Software License ?

CHAPTER 1 **Preface** 1

Audience and Scope 1

Feature Compatibility 1

Document Conventions 2

Communications, Services, and Additional Information 3

Documentation Feedback 4

Troubleshooting 4

CHAPTER 2 **Overview of Cisco Catalyst 8000V Edge Software on Amazon Web Services** 5

Deployment Options for Cisco Catalyst 8000V Running on Amazon Web Services 5

Licensing 6

Pay As You Go Licensing 6

Cisco IOS XE Technologies Not Supported 7

CHAPTER 3 **Deploying Cisco Catalyst 8000V on AWS** 9

Supported Instance Types for AWS 9

Notes and Guidelines 11

Prerequisites for Deploying Cisco Catalyst 8000V on AWS 11

Restrictions for Deploying Cisco Catalyst 8000V on AWS 12

Deploy the Cisco Catalyst 8000V Instance 12

Select the Cisco Catalyst 8000V Marketplace Offer 12

Launch the Instance Through the Website 12

Launch the Instance Through the EC2 Console 13

Associate the Public IP Address with the Cisco Catalyst 8000V Instance 14
Connect to the Instance using SSH 15
Create SSH Key Pairs 15
Create an AMI with Encrypted Elastic Block Storage 15

CHAPTER 4 **Enable the Guest Shell 17**
Enabling the Guest Shell 17
Create an IAM Instance Role 17
Assign an IAM Instance Role to a Cisco Catalyst 8000V Instance 19
Assign an IAM Instance Role to a New Instance 20
Examples of Guest Shell 20

CHAPTER 5 **Configure L2 Extension for Public Cloud 25**
Configure LISP Layer 2 Extension 26
Prerequisites for configuring LISP Layer 2 Extension 27
Restrictions for configuring LISP Layer 2 Extension 27
Configure LISP Layer 2 Extension 27
Creating a Cisco Catalyst 8000V instance on AWS 27
Configure subnets 28
Configure a tunnel between Cisco Catalyst 8000V on AWS and Cisco Catalyst 8000V on the Enterprise System 28
Configure LISP xTR on the Instance Running on AWS 29
Verify the LISP Layer 2 Traffic between Cisco Catalyst 8000V on AWS and Cisco Catalyst 8000V on the Enterprise System 31
Support for PMD Multi-Queue 32

CHAPTER 6 **Configure Ipv6 Functionalities 35**

CHAPTER 7 **Migrating Cisco CSR1000V Instances to Cisco Catalyst 8000V Using the AWS Migration Tool 37**
Prerequisites for the Migration 38
Limitations and Notes 38
Migrating a CSR1000V Instance to Cisco Catalyst 8000V 39
Verifying Successful Migration 40
Performing a Rollback 41

CHAPTER 8 **Deploying Transit VPC With Transit Gateway** 43

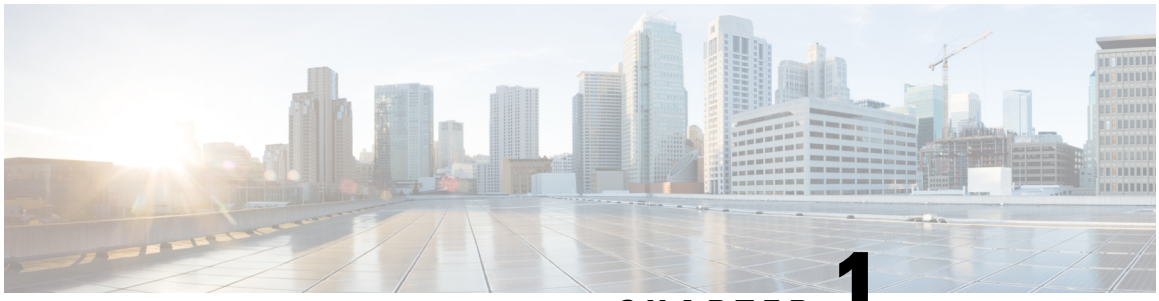
Benefits of the AWS Transit Gateway Solution 45

Prerequisites to the AWS Transit Gateway Solution 45

Limitations of the AWS Transit Gateway Solution 45

Configuring the AWS Transit Gateway Solution 45

Configuration Example 47



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, on page 1
- Feature Compatibility, on page 1
- Document Conventions, on page 2
- Communications, Services, and Additional Information, on page 3
- Documentation Feedback, on page 4
- Troubleshooting, on page 4

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 Marketplace.
- 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.

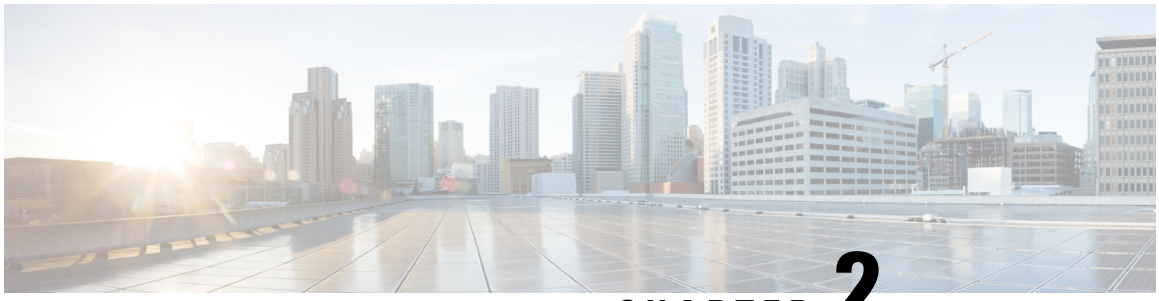
Documentation Feedback

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 Edge Software on Amazon Web Services

Cisco Catalyst 8000V Edge Software is a virtual router that offers routing, security, and network management functionalities as a cloud service, with multitenancy.

This router is supported on the Amazon Virtual Private Cloud (Amazon VPC), which enables you to provision a logically isolated section of the AWS Cloud. By doing so, you can launch the AWS resources in a virtual network that you've defined.

Cisco Catalyst 8000V has the ability to boot in either the autonomous or the controller mode. By default, Cisco Catalyst 8000V boots in the autonomous mode. If you wish to deploy and use Cisco Catalyst 8000V in the autonomous mode, continue with this guide.

This guide specifies the deployment options, procedures, and the configurations for Cisco Catalyst 8000V running on Amazon Web Services (AWS) for public and private cloud solutions.

For Cisco SD-WAN deployments or Cisco Catalyst 8000V in the controller mode, see Getting Started With the Cisco SD-WAN.

- Deployment Options for Cisco Catalyst 8000V Running on Amazon Web Services, on page 5
- Licensing, on page 6
- Pay As You Go Licensing, on page 6
- Cisco IOS XE Technologies Not Supported, on page 7

Deployment Options for Cisco Catalyst 8000V Running on Amazon Web Services

To use Cisco Catalyst 8000V on Amazon Web Services (AWS), purchase and launch the Cisco Catalyst 8000V instance as an Amazon Machine Image (AMI) on AWS Marketplace.

An Amazon Machine Image (AMI) provides the information required to launch your instance. You need to specify an AMI when you launch an instance. Note that you can launch as many instances from the AMI as you need.

Choose one of the following deployment options from the AWS Marketplace:

- Cisco Catalyst 8000V SD-WAN & Router - PAYG - DNA Advantage
- Cisco Catalyst 8000V SD-WAN & Router - PAYG - DNA Essentials

- Cisco Catalyst 8000V for SD-WAN & Routing

If you choose one of the first three options, proceed to licensing after choosing your deployment option. If you choose the Cisco SD-WAN option, see the *Getting Started with Cisco SD-WAN* guide.

Note If you are upgrading from an earlier version, use the Cisco Catalyst 8000V .bin file to upgrade the version of the Cisco Catalyst 8000V instance without having to recreate an AWS EC2 instance from a new AMI.

Licensing

After you visit the AWS Marketplace, purchase and launch the Cisco Catalyst 8000V device as an Amazon Machine Image (AMI) on the AWS Marketplace.

To use the Cisco Catalyst 8000V device, first choose the image or solution listing, purchase the image, and deploy the AMI. The next step is to either purchase the Cisco Catalyst 8000V software license(s) directly from Cisco, or use a Pay As You Go (PAYG) license that is already embedded with the image.

If you are using the Bring Your Own license (BYOL) licensing model,continue reading this section. Else, see the *Pay As You Go* section in this guide.

Bring Your Own Licensing Model

Bring Your Own License is the model where you buy a license from Cisco or a partner and install the license on the Cisco Catalyst 8000V device. If you choose the BYOL licensing model, after you deploy the Cisco Catalyst 8000V AMI from the AWS Marketplace and launch the instance, you must install the licenses using Cisco Smart Licensing Usage Policy.

Cisco Smart Licensing Usage Policy is an evolved version of the existing Smart Licensing model with the overarching objective of providing a licensing solution that does not interrupt the operations of your network. Rather, this model enables a compliance relationship to account for the hardware and software licenses you purchase and use.

After you purchase a license, no preliminary steps, such as registration or generation of keys etc., are required unless you use an export-controlled or enfored license. These licenses require authorization before use. For all the other licenses, you can configure the product features on the device right-away.

For more information about the Cisco Catalyst 8000V software licenses and the process for rehosting a license, see the *Cisco Catalyst 8000V Edge Software Configuration Guide*. For a list of license SKUs, see the latest Cisco Catalyst 8000V Release Notes.

Pay As You Go Licensing

To use Cisco Catalyst 8000V on AWS, you must purchase and launch Cisco Catalyst 8000V as an Amazon Machine Image (AMI) on the AWS Marketplace. Further, you must choose either the BYOL or the Pay As You Go (PAYG) licensing model.

If you have chosen the BYOL model, go to the *Licensing* section in this guide. If you have chosen the PAYG licensing model, continue to read this section.

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

The following Cisco IOS XE Technology Packages are available in this licensing model: **Cisco Catalyst 8000V - Essentials PAYG** and **Cisco Catalyst 8000V - Advantage PAYG**.

PAYG is subject to the following conditions:

- You are billed hourly by Amazon Web Services (AWS) for using the Cisco Catalyst 8000V AMI. This hourly usage fee is in addition to the VPC usage fees charged by AWS.
- You do not purchase the licenses directly from Cisco for Cisco Catalyst 8000V.
- You do not install the Cisco licenses on the router.
- You cannot rehost hourly-billed AMIs.

For more information about the features contained in the Cisco Catalyst 8000V technology packages, see the *Cisco Catalyst 8000V Edge Software Configuration Guide*.

Cisco IOS XE Technologies Not Supported

When you deploy your Cisco Catalyst 8000V instance on an AWS instance, Cisco Catalyst 8000V supports fewer Cisco IOS XE technologies than are supported by other hypervisors. Some technologies might not be available because they are not supported in an Amazon cloud.

The following restrictions apply when you deploy Cisco Catalyst 8000V on an AWS instance:

- Although CLI commands for unsupported features may be visible on Cisco Catalyst 8000V, testing by Cisco has determined that the unsupported features (mentioned in the table in this section) do not work in AWS deployments.
- Routing protocols are supported over a tunnel only.
- The Cisco Catalyst 8000V AMI does not support remote management of the router using Cisco Prime Network Services Controller.

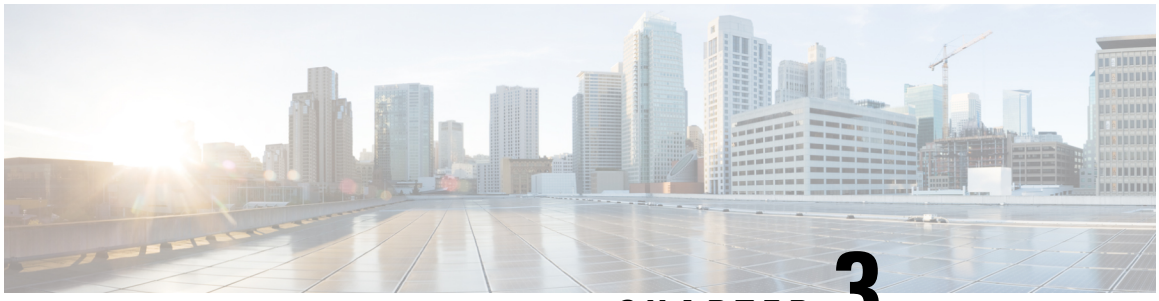
The following table lists the Cisco IOS XE technologies that are not supported when deploying Cisco Catalyst 8000V on an AWS instance.

Table 1: Cisco IOS XE Technologies Not Supported on AWS Deployments

Technology	Non-Supported Features
Basic Routing	OSPF
Data Center Interconnect	OTV and WCCPv2
MPLS	MPLS, EoMPLS, VRF and VPLS
Redundancy	HSRP
WAAS	Integrated AppNav-XE

The following caveats apply to the Cisco IOS XE technology support on AWS deployments:

- You cannot apply NAT PAT on the same interface that is configured with a crypto map. The workaround is to use a different IP Security feature such as SVTI or DMVPN, or you can configure a two-router solution with one router for NAT and another router for the IP Security crypto map.
- You cannot configure HSRP between the Cisco Catalyst 8000V nodes in an Amazon cloud. Amazon does not allow running HSRP on the hosts in the VPC. Amazon AWS blocks all broadcast and multicast traffic in a VPC.
- It is recommended that you disable the Source/Destination check on the Cisco Catalyst 8000V interfaces.
- EtherChannel is not supported.
- IP Multicast only works with the Amazon Transit Gateway solution.



CHAPTER 3

Deploying Cisco Catalyst 8000V on AWS

This chapter describes the steps involved in deploying a Cisco Catalyst 8000V instance on AWS. To deploy a Cisco Catalyst 8000V instance, you need an Amazon Machine Image (AMI), which is an AWS supported and maintained image. The AMI provides the information required to launch your instance.

After you log in to the AWS Marketplace, select the template or the marketplace offer of your choice. Further, follow the procedures mentioned in this chapter to create an AMI with an encrypted Elastic Block Storage (EBS).

Note If you are using a BYOL AMI, see Licensing, on page 6.

- Supported Instance Types for AWS, on page 9
- Prerequisites for Deploying Cisco Catalyst 8000V on AWS, on page 11
- Restrictions for Deploying Cisco Catalyst 8000V on AWS, on page 12
- Deploy the Cisco Catalyst 8000V Instance, on page 12

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

Release Number	Supported Instance Types
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
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.

Notes and Guidelines

- To optimize the performance while using instance types that support PMD multi-queue, see Support for PMD Multi-Queue, on page 32.
- To determine the maximum number of network interfaces supported per instance, see Private IP Addresses Per Network Interface Per Instance Type.
- The c5n.large, c5n.xlarge, c5n.2xlarge, and c5n.9xlarge instance types are replaced with c6in.large, c6in.xlarge, c6in.2xlarge, and c6in.8xlarge respectively from the 17.13.1a release. However, when you upgrade from an earlier release to Cisco IOS XE 17.13.1a, you will continue to see the c5n instance types. You can manually upgrade the instance types to the corresponding c6in replacement.

Prerequisites for Deploying Cisco Catalyst 8000V on AWS

Before you launch Cisco Catalyst 8000V on AWS, you must:

- Have an AWS account.
- Have an SSH client such as Putty on Windows or Terminal on Macintosh, or have access to an EC2 instance console, to access the Cisco Catalyst 8000V console.
- Determine the instance type for your Cisco Catalyst 8000V AMI.
- Create an Amazon VPC if you're planning to launch the AMI using 1-Click Launch.

Restrictions for Deploying Cisco Catalyst 8000V on AWS

The jumbo frames in a VPC have limitations. For more information on jumbo frames, see Network Maximum Transmission Unit (MTU) for Your EC2 Instance.

Deploy the Cisco Catalyst 8000V Instance

To deploy the Cisco Catalyst 8000V AMI, perform the steps as described in the following sections:

Select the Cisco Catalyst 8000V Marketplace Offer

Step 1 Log in to Amazon Web Services Marketplace.

Step 2 Click **Discover Products**.

Step 3 In the search bar, search for Cisco Catalyst 8000V. The system displays the following offers:

- Cisco Catalyst 8000V - Advantage PAYG
- Cisco Catalyst 8000V - Essentials PAYG
- Cisco Catalyst 8000V - BYOL

Step 4 Select the Cisco Catalyst 8000V AMI that you are planning to deploy.

The marketplace displays product information such as supported instance types, pricing, and support details.

Launch the Instance Through the Website

Before you begin

If you're launching the AMI using 1-Click Launch, you must first create a Virtual Private Cloud (VPC). To know how to do this, see the AWS documentation on VPC.

Step 1 After you select the Cisco Catalyst 8000V offer from the AWS Marketplace, click **Continue to Subscribe** after you accept the Terms and Conditions.

Step 2 From the **Configure Software** window, choose the **Software Version** and your **Region** for your Cisco Catalyst 8000V instance.

For information on zones and regions in Amazon EC2, see: Regions and Availability Zones.

Step 3 From the **Fulfillment Option** field, choose **Amazon Machine Image**.

Step 4 Click **Continue to Launch**.

Step 5 In the **Launch This Software** window, choose **Launch from Website**.

Step 6 Choose the **EC2 Instance Type** from the drop-down list.

Step 7 Configure the **VPC**, **Subnet**, **Security Group** and **Key Pair** settings for your instance.

For more information about these settings in AWS, see Parameters for Instance Configuration.

Step 8 Click **Launch**.

To view the newly launched instance, click **Launch** and navigate to https://console.aws.amazon.com/ec2/. Ensure that **Status Check** displays the `2/2 checks passed` message before you try to connect to the instance using SSH.

Launch the Instance Through the EC2 Console

Perform this procedure after you login to the AWS Marketplace and Subscribe to Cisco Catalyst 8000V. This procedure describes how to launch an instance through the EC2 Console.

Step 1 After you select the Cisco Catalyst 8000V offer from the AWS Marketplace, click **Continue to Subscribe** after you accept the Terms and Conditions.

Step 2 From the **Configure Software** window, choose the **Software Version** and your **Region** for your Cisco Catalyst 8000V instance.

For information on zones and regions in Amazon EC2, see: Regions and Availability Zones.

Step 3 From the **Fulfillment Option** field, choose **Amazon Machine Image**.

Step 4 Click **Continue to Launch**, and choose **Launch with E2 Console**.

Step 5 In the **Launch an Instance** window, in the **Name** field, enter the name for your instance.

Step 6 In the **Application and OS Images** area, the AMI is autofilled based on your subscription and your choice of the software version.

Step 7 From the **Instance Type** field, choose a supported instance type from the drop-down list. For more information on which instance type is supported for each IOS XE release, see Supported Instance Types for AWS, on page 9.

Step 8 Configure the **Key Pair** settings for your instance. Choose an existing key pair or create a new key by uploading your own public key. To create a new key pair, click **Create Key Pair**, enter the key pair name, and click **Create**. After the key pair is created, ensure that you have downloaded the private key from Amazon before continuing.

Note that you can access a newly created private key only once. After you download the key pair, click **Close**.

Note AWS security policies require that the private key permission level be set to 400. To set this value for the .pem file, open a UNIX shell terminal screen and run the **chmod 400***pem-file-name* command.

From Cisco IOS XE 17.10.1a, Cisco Catalyst 8000V supports the ED25519 SSH key. This is in addition to the existing SSH-RSA keys. We recommend that you use the ED25519 SSH key for faster generation and verification of the keys, and for better collision resilience and security.

Step 9 Configure the **Network** settings for your instance. Choose the VPC subnet in which you want to deploy the Cisco Catalyst 8000V instance from the drop-down list. This setting determines the availability zone of your instance.

Note By default, one insterface is configured. You can create an additional interfaces from the **Instance Details** area under **Network Interfaces**. The maximum number of interfaces that are supported depends on your instance type.

Step 10 Configure the **Security Group** settings. You can create a new security group or choose an existing security group. Cisco Catalyst 8000V requires SSH for console access. Cisco Catalyst 8000V also requires that the Security Group, at a minimum, does not block TCP/22. These settings are used to manage the Cisco Catalyst 8000V instance.

Step 11 Choose the metadata version from the **Metadata Version** drop-down list. Choose either **V1 and V2 (token optional)** or **V2 (token required)**. In both these scenarios, the instance uses session-oriented requests by creating tokens. The tokens are used to fetch all the required metadata for your instances.

For Cisco IOS XE 17.4.x and 17.5.x releases, only version 1 or V1 is applicable. From Cisco IOS XE 17.6.1, metadata versions V1 and V2 are supported.

Step 12 Configure the **Storage** settings for your instance. Retain the default hard drive setting. Note that you cannot change the size of virtual hard drives.

Step 13 Provide the Day Zero configuration data or the bootstrap properties in the custom data format using the **User Data** field. For the supported custom data format, see Day 0 Configuration.

Step 14 Click **Review and Launch**.

Step 15 Review the Cisco Catalyst 8000V instance information, and click **Launch Instance**.

After the instance is launched, a success message appears on top of the Instances page. You can also view the newly launched instance in the Instances list. Click the newly launched instance to access the instance.

Associate the Public IP Address with the Cisco Catalyst 8000V Instance

To access the management console using an SSH connection, you must first associate an interface in the Cisco Catalyst 8000V instance with the public IP address created with the VPC. Perform the following steps to associate the public IP address with your Cisco Catalyst 8000V instance.

Step 1 Choose **Services** > **EC2** > **Instances**, and select the Cisco Catalyst 8000V instance.

Step 2 In the **Network interfaces** window that is displayed, click **eth0**.

A dialog box displays detailed information about the eth0 interface. Note the interface's private IP address.

Step 3 Click **Interface ID Value**.

Step 4 Click **Actions**, and choose **Associate Address** from the drop-down list.

Step 5 Choose an available public IP address from the **Elastic IP address** drop-down list.

Step 6 If you're reassigning a public IP address that is currently in use and is mapped to another elastic network interface (ENI), click **Allow Reassociation**.

Step 7 Validate that the selected private IP address matches the one that you noted in step 3.

Step 8 Click **Associate Address**.

This action associates the public IP address (Amazon elastic IP) with the private IP address of the network interface. You can now use this interface to access the management console.

Connect to the Instance using SSH

The Cisco Catalyst 8000V instance on AWS requires SSH for console access. To access the Cisco Catalyst 8000V AMI, perform the following steps:

Step 1 After you launch the Cisco Catalyst 8000V instance and the status is displayed as `Running`, select the instance from the **Instances** window.

Step 2 Run the **ssh -i** *pem-file-name* **ec2-user**@[*public-ipaddress* | *DNS-name*] UNIX shell command to connect to the Cisco Catalyst 8000V console using SSH:

- Use the default user name for your AMI, **`ec2-user`**, to access the instance for the first time.
- Use the private key stored in the .pem file to authenticate the access to the instance.

Step 3 Start the Cisco Catalyst 8000V instance.

For information on downloading and activating the license for the BYOL AMI, see Licensing, on page 6.

Create SSH Key Pairs

When you deploy a Cisco Catalyst 8000V instance in AWS, you can provide an SSH key as the method of authentication to access your instance. In this case, you must create key pairs.

To create a key pair, you can use the Amazon EC2 to create an RSA or an ED25519 key pair. Alternatively, you can use a third-party tool to create a key pair and then import the public key to the Amazon EC2 instance.

After you create and configure the key pair, the new VM starts and the system displays a `status passes 2/2 check` message. You can access the new VM's console using the .pem key and use the private key to authenticate the access to the new VM console.

Create an AMI with Encrypted Elastic Block Storage

Amazon Elastic Block Storage (EBS) encryption is an encryption solution for the EBS resources associated with your EC2 instances. Amazon EBS encryption uses AWS KMS keys to ensure data security. To create a Cisco Catalyst 8000V AMI with encrypted Amazon EBS, perform the following steps.

Step 1 Choose **Services** > **EC2** > **Instances** to view the list of instances.

Step 2 Select the instance that you want to use as the base for creating the new AMI with encrypted Amazon EBS. Ensure that the status of the base instance is **Stopped**.

Step 3 Take a snapshot of this instance by following steps a to f :

a) Click on the root device, for example, /dev/xvda/.
The system displays the **Block Device** dialog box.

b) Click **EBS ID**.
The volume for this snapshot is displayed under **ELASTIC BLOCK STORE** > **Volumes**.

c) Choose **Actions** > **Create Snapshot**.
The system displays the **Create Snapshot** dialog box.

d) Click **Create**.

e) In the **Create Image** field on the **EBS** window, enter a name for the snapshot.
f) From the **Virtualization type** drop-down list, choose the **Hardware-assisted virtualization** option.

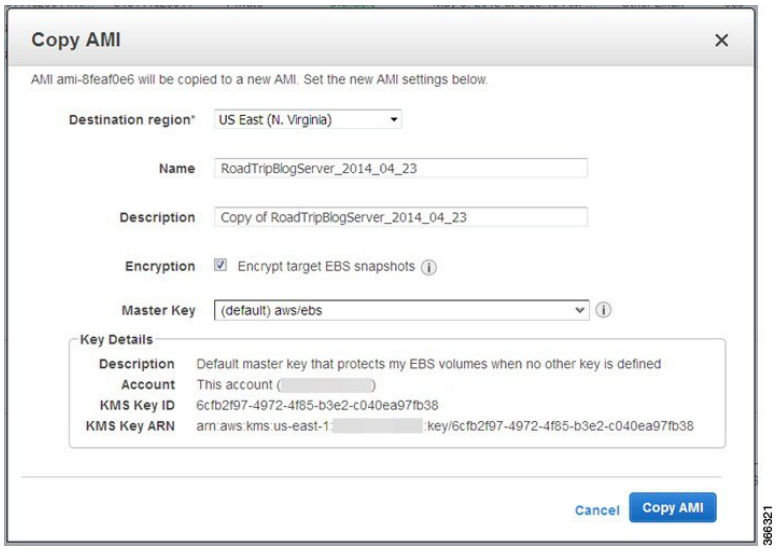
The system displays the `Snapshot Creation Started` message in the **Create Snapshot** dialog box. After the snapshot creation is complete, under **ELASTIC BLOCK STORE** > **Snapshots**, the new snapshot is listed with the status **Completed**.

Step 4 Choose **EC2** > **IMAGES** > **AMIs** to create a private AMI.

The name of the snapshot instance that you created earlier appears in the list of AMIs.

Step 5 Choose the snapshot instance you created earlier, and choose **Actions** > **Copy AMI**.

The **Copy AMI** dialog box is displayed with the **Destination Region**, **Name**, **Description**, **Encryption**, **Master Key**, and **Key Details** fields.



Step 6 From the **Destination region** drop-down list, choose the destination, for example, **US East**.

Step 7 Enter a **Name**, for example, **`encrypted-C8000V-1`**.

Step 8 Specify a **Description**.

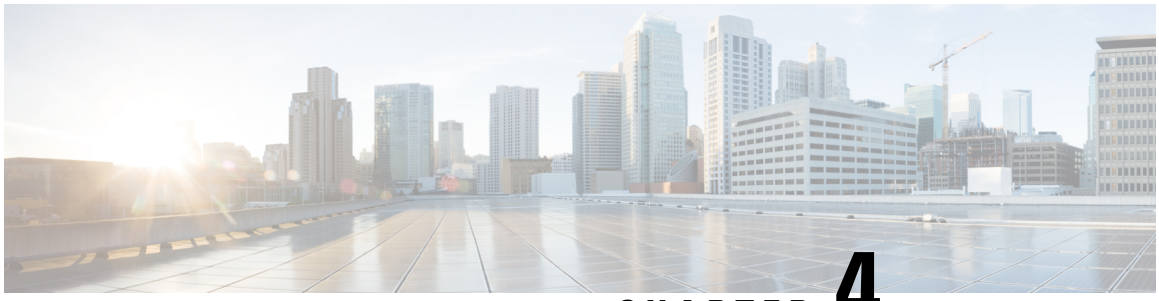
Step 9 Check the **Encrypt target EBS snapshots** check box.

Step 10 From the **Master Key** drop-down list, choose the default value.

Step 11 Click **Copy AMI**.

The new AMI, with the encrypted Amazon EBS, is created after several minutes.

Step 12 To verify the status of the new AMI, navigate to **EC2** > **IMAGES** > **AMIs**. You will see that the new AMI is listed.



CHAPTER 4

Enable the Guest Shell

- Enabling the Guest Shell, on page 17
- Create an IAM Instance Role, on page 17
- Assign an IAM Instance Role to a Cisco Catalyst 8000V Instance, on page 19
- Assign an IAM Instance Role to a New Instance, on page 20
- Examples of Guest Shell, on page 20

Enabling the Guest Shell

To enable the guest shell on Cisco Catalyst 8000V running on AWS, create an IAM instance role and establish trust with an EC2 service. You have a choice of either assigning the IAM instance role to a preexisting Cisco Catalyst 8000V instance or assigning the IAM instance role to a new Cisco Catalyst 8000V instance.

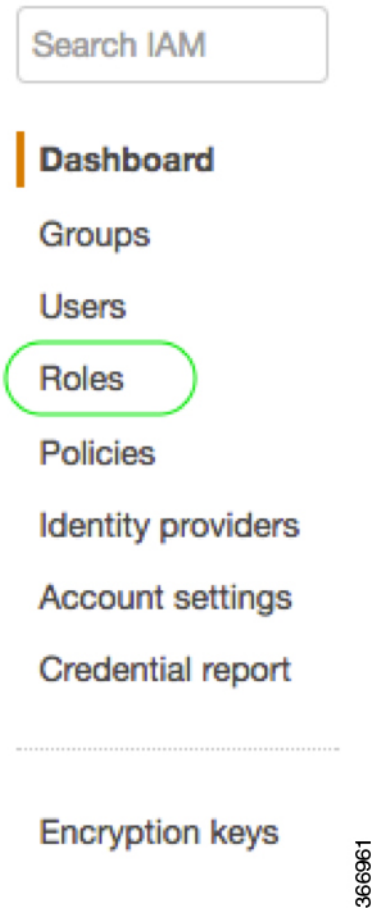
For more information on how to perform these tasks, see *Assign an IAM Instance Role to a Cisco Catalyst 8000V* and *Assign an IAM Instance Role to a New Cisco Catalyst 8000V*.

Then, perform further configuration steps on Cisco Catalyst 8000V and enter the guest shell.

Create an IAM Instance Role

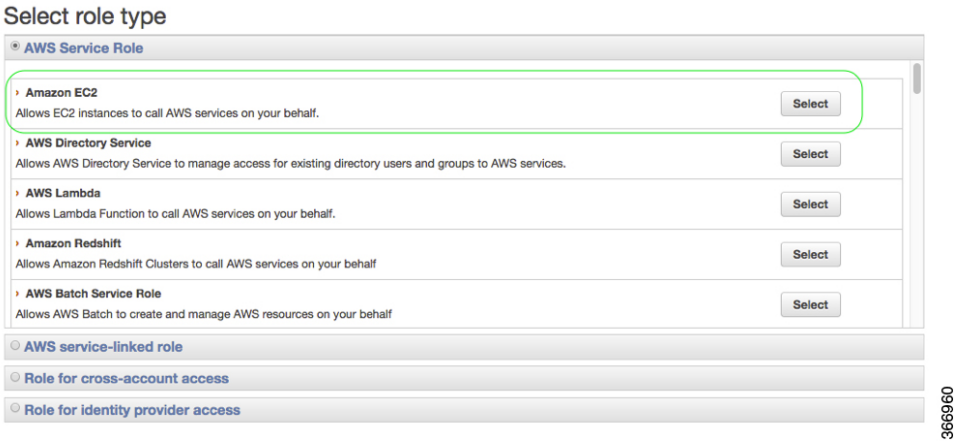
1. Sign into AWS, as an administrator with permissions to create an IAM Role
2. Click **EC2** to enter the EC2 console.
3. Click **IAM** to enter the IAM console.
4. Click **Roles**.

Figure 1: IAM Instance Roles



5. Click **Create new Role**.
6. Enter a name for your app's role.
7. Click **Continue**.
8. Select a Role Type.

Figure 2: IAM Instance Role Types



9. For the Amazon EC2 role type, click **Select**.

   This establishes trust with an EC2 service.

10. From **Set Permissions**, click **Select Policy Template**.

11. Select a template (for example, "Amazon S3 Full Access") by clicking **Select**. You can select multiple services. Use this option to specify the access in further detail. For example, you can allow an IAM instance role to read from an S3 bucket but not write to an S3 bucket.

12. Enter the role name.

13. Click **Create Role**.

Assign an IAM Instance Role to a Cisco Catalyst 8000V Instance

Specifying an IAM instance role is not a mandatory for accessing the guest shell. However, it will later allow you to access specific entities in the AWS account using a key or a password that eliminates the need to save account information on the Cisco Catalyst 8000V instance.

Step 1 Click **EC2** to enter the EC2 dashboard.

Step 2 Select one of your listed Cisco Catalyst 8000V instances, right-click and select **Instance Setup**. Then, select **Attach/Replace IAM Role**.

Step 3 From the drop-down list, select an IAM instance role that you created previously.

Step 4 Enter the following CLI configuration commands on Cisco Catalyst 8000V and relaunch Cisco Catalyst 8000V.

```
Router(config)# interface GigabitEthernet1
Router(config-if)# ip address dhcp
Router(config-if)# ip nat outside
Router(config-if)# exit
Router(config)# interface VirtualPortGroup0
Router(config-if)# ip address 192.168.35.1 255.255.255.0
Router(config-if)# ip nat inside
Router(config-if)# exit
Router(config)# ip nat inside source list GS_NAT_ACL interface GigabitEthernet1 overload
```

```
Router(config)# ip access-list standard GS_NAT_ACL
Router(config)# permit 192.168.0.0 0.0.255.255
Router(config)# app-hosting appid guestshell
Router(config-app-hosting)# vnic gateway1 virtualportgroup 0 guest-interface 0 guest-ipaddress
192.168.35.2 netmask 255.255.255.0 gateway 192.168.35.1 name-server 8.8.8.8 default
Router(config-app-hosting)# resource profile custom cpu 1500 memory 512
Router(config-app-hosting)# exit
Router(config)# exit
Router# guestshell enable
Router# guestshell run python
```

Assign an IAM Instance Role to a New Instance

The following procedure shows how to assign an IAM Instance Role to Cisco Catalyst 8000V during the creation of a new Cisco Catalyst 8000V instance.

Step 1 Launch a new Cisco Catalyst 8000V as an EC2 instance, and choose an instance type.

Step 2 Click **Next: Configure Instance Details**.

Step 3 Perform one of the following two steps:

a) Click the IAM role text box to select an existing IAM instance role from the dropdown list.

b) Click **Create new IAM role** to create a new IAM instance role.

Step 4 Enter the following CLI configuration commands on the Cisco Catalyst 8000V instance and relaunch the instance.

```
Router(config)# interface GigabitEthernet1
Router(config-if)# ip address dhcp
Router(config-if)# ip nat outside
Router(config-if)# exit
Router(config)# interface VirtualPortGroup0
Router(config-if)# ip address 192.168.35.1 255.255.255.0
Router(config-if)# ip nat inside
Router(config-if)# exit
Router(config)# ip nat inside source list GS_NAT_ACL interface GigabitEthernet1 overload
Router(config)# ip access-list standard GS_NAT_ACL
Router(config)# permit 192.168.0.0 0.0.255.255
Router(config)# app-hosting appid guestshell
Router(config-app-hosting)# vnic gateway1 virtualportgroup 0 guest-interface 0 guest-ipaddress
192.168.35.2 netmask 255.255.255.0 gateway 192.168.35.1 name-server 8.8.8.8 default
Router(config-app-hosting)# resource profile custom cpu 1500 memory 512
Router(config-app-hosting)# exit
Router(config)# exit
Router# guestshell enable
Router# guestshell run python
```

Examples of Guest Shell

The following examples show how to download packages in the Guest Shell on a Cisco Catalyst 8000V instance, and a few other useful guest shell commands.

1. Install packages using the `yum` or `pip3` commands. For example, enter the `[guestshell@guestshell ~] sudo pip3 install awscli` command to install the AWS CLI and Amazon SDK.

```
[guestshell@guestshell ~]$ sudo pip3 install awscli
WARNING: Running pip install with root privileges is generally not a good idea. Try `pip3
 install --user` instead.
Collecting awscli
  Downloading
https://files.pythonhosted.org/packages/ce/38/6f2b06f0f00e6c381ad4741d0bf97eb2e3faa23238a2dfe31577154a207c/awscli-1.18.157-py2.py3-none-any.whl
 (3.4MB)
    100% |################################| 3.4MB 369kB/s
Collecting colorama<0.4.4,>=0.2.5; python_version != "3.4" (from awscli)
  Downloading
https://files.pythonhosted.org/packages/c9/dc/45cdef1b4d119eb96316b3117e6d5708a08029992b2fee2c143c7a0a5cc5/colorama-0.4.3-py2.py3-none-any.whl
Collecting s3transfer<0.4.0,>=0.3.0 (from awscli)
  Downloading
https://files.pythonhosted.org/packages/69/79/e6afb3d8b0b4e96cefbdb690f741d7db24547ff1f94240c997a26fa908d3/s3transfer-0.3.3-py2.py3-none-any.whl
 (69kB)
    100% |################################| 71kB 7.3MB/s
Collecting docutils<0.16,>=0.10 (from awscli)
  Downloading
https://files.pythonhosted.org/packages/22/cd/a6aa959dca619918ccb55023b4cb151949c64d4d5d55b3f4ffd7eee0c6e8/docutils-0.15.2-py3-none-any.whl
 (547kB)
    100% |################################| 552kB 2.1MB/s
Collecting PyYAML<5.4,>=3.10; python_version != "3.4" (from awscli)
  Downloading
https://files.pythonhosted.org/packages/64/c2/b80047c7ac2478f9501676c988a5411ed5572f35d1beff9cae07d321512c/PyYAML-5.3.1.tar.gz
 (269kB)
    100% |################################| 276kB 3.6MB/s
Collecting rsa<=4.5.0,>=3.1.2; python_version != "3.4" (from awscli)
  Downloading
https://files.pythonhosted.org/packages/26/f8/8127fcda0294f044121d20aac7785feb810e159098447967a6103dedf6c96/rsa-4.5-py2.py3-none-any.whl
Collecting botocore==1.18.16 (from awscli)
  Downloading
https://files.pythonhosted.org/packages/2d/9e/afa41db0cd911869305b783b9b021be67ea23c8b7b317caa4632dbf3cf/botocore-1.18.16-py2.py3-none-any.whl
 (6.7MB)
    100% |################################| 6.7MB 173kB/s
Collecting pyasn1>=0.1.3 (from rsa<=4.5.0,>=3.1.2; python_version != "3.4"->awscli)
  Downloading
https://files.pythonhosted.org/packages/62/1e/a94a8d635fa3ce4cfc7f506003548d0a2447ae76fd5ca53932970fe3053f/pyasn1-0.4.8-py2.py3-none-any.whl
 (77kB)
    100% |################################| 81kB 7.5MB/s
Collecting urllib3<1.26,>=1.20; python_version != "3.4" (from botocore==1.18.16->awscli)

  Downloading
https://files.pythonhosted.org/packages/9f/f0/a391d1463ebb1b233795cabfc0ef38d3db4442339de68f847026199e69d7/urllib3-1.25.10-py2.py3-none-any.whl
 (127kB)
    100% |################################| 133kB 6.1MB/s
Collecting python-dateutil<3.0.0,>=2.1 (from botocore==1.18.16->awscli)
  Downloading
https://files.pythonhosted.org/packages/d4/70/d60450c3dd48ef87586924207ae8907090de0b306af2bce5d134d78615cb/python_dateutil-2.8.1-py2.py3-none-any.whl
 (227kB)
    100% |################################| 235kB 4.0MB/s
Collecting jmespath<1.0.0,>=0.7.1 (from botocore==1.18.16->awscli)
  Downloading
https://files.pythonhosted.org/packages/07/cb/5f001272b6faeb23c1c9e0acc04d48eaaf5c862c17709d20e3469c6e0139/jmespath-0.10.0-py2.py3-none-any.whl
Collecting six>=1.5 (from python-dateutil<3.0.0,>=2.1->botocore==1.18.16->awscli)
  Downloading
https://files.pythonhosted.org/packages/ee/ff/48bde5c0f013094d729fe4b0316ba2a24774b3ff1c52d924a8a4cb04078a/six-1.15.0-py2.py3-none-any.whl
Installing collected packages: colorama, urllib3, six, python-dateutil, jmespath,
botocore, s3transfer, docutils, PyYAML, pyasn1, rsa, awscli
  Running setup.py install for PyYAML ... done
Successfully installed PyYAML-5.3.1 awscli-1.18.157 botocore-1.18.16 colorama-0.4.3
docutils-0.15.2 jmespath-0.10.0 pyasn1-0.4.8 python-dateutil-2.8.1 rsa-4.5
```

```
s3transfer-0.3.3 six-1.15.0 urllib3-1.25.10
[guestshell@guestshell ~]$ aws s3 ls c8kv
Unable to locate credentials. You can configure credentials by running "aws configure"
```

2. Having installed the AWS CLI, enter the `aws s3` command such as `aws s3 ls`.

```
[guestshell@guestshell ~]$ aws s3 ls c8kv
2020-10-14 19:44:08  433546509 upgrade.bin
[guestshell@guestshell ~]$
```

3. You can download a Cisco Catalyst 8000V AWS package containing sample scripts using the `sudo pip3 install csr_aws_guestshell` command.

   Example:

```
[guestshell@guestshell ~]$ sudo pip3 install csr_aws_guestshell
WARNING: Running pip install with root privileges is generally not a good idea. Try `pip3
 install --user` instead.
Collecting csr_aws_guestshell
Downloading
https://files.pythonhosted.org/packages/42/a7/c772726166f809d8223ef448f5d7fa2cfb8a809525a11991612d81cdd080a/csr_aws_guestshell-0.0.17.dev.tar.gz
Collecting awscli (from csr_aws_guestshell)
Downloading
https://files.pythonhosted.org/packages/23/1b/2b65dbb4e18ab8c232d413b75591ccd22021233fb3c3fd1dbb099d0170ce/awscli-1.18.162-py2.py3-none-any.whl
 (3.4MB)
100% |################################| 3.4MB 352kB/s
Collecting boto (from csr_aws_guestshell)
Downloading
https://files.pythonhosted.org/packages/23/10/c0b78c27298029e4454a472a1919bde20cb182dab1662cec7f2ca1dcc523/boto-2.49.0-py2.py3-none-any.whl
 (1.4MB)
100% |################################| 1.4MB 794kB/s
Collecting boto3 (from csr_aws_guestshell)
Downloading
https://files.pythonhosted.org/packages/30/3c/c9b65c398de6589c93b3c65aa27a695a6cdf7588c50a7934ad3ac6599a8a/boto3-1.16.2-py2.py3-none-any.whl
 (129kB)
100% |################################| 133kB 7.2MB/s
Collecting rsa<=4.5.0,>=3.1.2; python_version != "3.4" (from awscli->csr_aws_guestshell)
Downloading
https://files.pythonhosted.org/packages/26/f8/8127fdda0294f044121d20aac7785feb810e15909447967a6103dedfb96/rsa-4.5-py2.py3-none-any.whl
Collecting botocore==1.19.2 (from awscli->csr_aws_guestshell)
Downloading
https://files.pythonhosted.org/packages/1f/96/35fa364675cf17e3a190ae08716ac4b078ca86a62ef071d32cd8b86d52bc/botocore-1.19.2-py2.py3-none-any.whl
 (6.7MB)
100% |################################| 6.7MB 164kB/s
Collecting PyYAML<5.4,>=3.10; python_version != "3.4" (from awscli->csr_aws_guestshell)
Downloading
https://files.pythonhosted.org/packages/64/c2/b80047c7ac2478f9501676c988a5411ed5572f35d1beff9cae07d321512c/PyYAML-5.3.1.tar.gz
 (269kB)
100% |################################| 276kB 3.6MB/s
Collecting s3transfer<0.4.0,>=0.3.0 (from awscli->csr_aws_guestshell)
Downloading
https://files.pythonhosted.org/packages/69/79/e6afb3d8b0b4e96cefbdb690f741d7dd24547ff1f94240c997a26fa908d3/s3transfer-0.3.3-py2.py3-none-any.whl
 (69kB)
100% |################################| 71kB 7.6MB/s
Collecting docutils<0.16,>=0.10 (from awscli->csr_aws_guestshell)
Downloading
https://files.pythonhosted.org/packages/22/cd/a6aa959dca619918ccb55023b4cb151949c64d4d5d55b3f4ffd7eee0c6e8/docutils-0.15.2-py3-none-any.whl
 (547kB)
100% |################################| 552kB 1.9MB/s
Collecting colorama<0.4.4,>=0.2.5; python_version != "3.4" (from
awscli->csr_aws_guestshell)
Downloading
https://files.pythonhosted.org/packages/c9/dc/45cdef1b4d119bb96316b3117e6d5708a08029992b2fee2c143c7a0a5cc5/colorama-0.4.3-py2.py3-none-any.whl
Collecting jmespath<1.0.0,>=0.7.1 (from boto3->csr_aws_guestshell)
Downloading
```

```
https://files.pythonhosted.org/packages/07/dc/5f001272b6faab23c1c9e0acc04d48eaaf5c862c17709d20e3469c6e0139/jmespath-0.10.0-py2.py3-none-any.whl
Collecting pyasn1>=0.1.3 (from rsa<=4.5.0,>=3.1.2; python_version !=
"3.4"->awscli->csr_aws_guestshell)
Downloading
https://files.pythonhosted.org/packages/62/1e/a94a8d635fa3ce4cfc7f506003548d0a2447ae76fd5ca53932970fe3053f/pyasn1-0.4.8-py2.py3-none-any.whl
 (77kB)
100% |################################| 81kB 9.4MB/s
Collecting urllib3<1.26,>=1.25.4; python_version != "3.4" (from
botocore==1.19.2->awscli->csr_aws_guestshell)
Downloading
https://files.pythonhosted.org/packages/56/aa/4ef5aa67a9a62505db124a5cb5262332d1d4153462eb8fd89c9fa41e5d92/urllib3-1.25.11-py2.py3-none-any.whl
 (127kB)
100% |################################| 133kB 6.5MB/s
Collecting python-dateutil<3.0.0,>=2.1 (from botocore==1.19.2->awscli->csr_aws_guestshell)
Downloading
https://files.pythonhosted.org/packages/d4/70/d60450c3dd48ef87586924207ae8907090de0b306af2bce5d134d78615cb/python_dateutil-2.8.1-py2.py3-none-any.whl
 (227kB)
100% |################################| 235kB 4.6MB/s
Collecting six>=1.5 (from
python-dateutil<3.0.0,>=2.1->botocore==1.19.2->awscli->csr_aws_guestshell)
Downloading
https://files.pythonhosted.org/packages/ee/ff/48bde5c0f013094d729fe4b0316ba2a24774b3ff1c52d924a8a4cb04078a/six-1.15.0-py2.py3-none-any.whl
Installing collected packages: pyasn1, rsa, urllib3, six, python-dateutil, jmespath,
botocore, PyYAML, s3transfer, docutils, colorama, awscli, boto, boto3, csr-aws-guestshell
Running setup.py install for PyYAML ... done
Running setup.py install for csr-aws-guestshell ... done
Successfully installed PyYAML-5.3.1 awscli-1.18.162 boto-2.49.0 boto3-1.16.2
botocore-1.19.2 colorama-0.4.3 csr-aws-guestshell-0.0.17.dev0 docutils-0.15.2
jmespath-0.10.0 pyasn1-0.4.8 python-dateutil-2.8.1 rsa-4.5 s3transfer-0.3.3 six-1.15.0
urllib3-1.25.11
```

The following scripts are included in the csr_aws_guestshell package:

get-metadata.py - retrieves and prints instance metadata from AWS.

get-route-table.py - retrieves instances in VPC along with routes, route tables, and associations.

save-config-to-s3.py - saves the Cisco IOS XE CLI commands to an S3 bucket.

save-tech-support-to-s3.py - saves the tech support output to an S3 bucket.

load-bin-from-s3.py - downloads a .bin file for Cisco Catalyst 8000V and reloads.

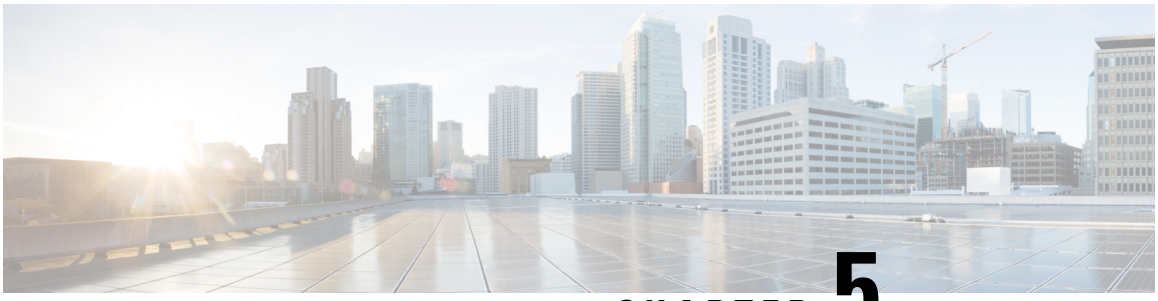
get-stat-drop.py - retrieves the CLI statistics and pushes them to cloudwatch.

capture-interface.py - sets the Cisco IOS XE CLI commands to monitor and captures packets for a period of time, then uploads the file to S3.

4. In the following example, the load-bin-from-s3.py script loads a binary from S3 and boots a Cisco Catalyst 8000V image:

```
[guestshell@guestshell ~]$ load-bin-from-s3.py csr1kv ultra_167.bin
/bootflash/ultra_167.bin  446866343 / 446866343  (100.00%)
Download Complete
```

Note The `csr_aws_guestshell` package will continue to work with Cisco Catalyst 8000V.



CHAPTER 5

Configure L2 Extension for Public Cloud

This chapter describes how to enable enterprise and cloud providers to configure an L2 extension for public clouds with Cisco Catalyst 8000V instances using LISP. Use the command-line interface to extend a layer 2 domain between your public cloud network and the enterprise network.

The following are some of the terminologies and concepts that you should be aware before you configure the LISP Layer 2 Extension:

- **Locator/ID Separation Protocol (LISP)**: LISP is a network architecture and protocol that implements the use of two namespaces instead of a single IP address:
  - Endpoint identifiers (EIDs) - assigned to end hosts.
  - Routing locators (RLOCs) - assigned to devices (primarily routers) that make up the global routing system.
- **LISP-enabled virtualized router**: A virtual machine or appliance that supports routing and LISP functions, including host mobility.
- **Endpoint ID (EID)**: An EID is an IPv4 or IPv6 address used in the source and destination address fields of the first (most inner) LISP header of a packet.
- **Routing locator (RLOC)**: The IPv4 or IPv6 addresses that are used to encapsulate and transport the flow between the LISP nodes. An RLOC is the output of an EID-to-RLOC mapping lookup.
- **Egress Tunnel Router (ETR)**: An ETR is a device that is the tunnel endpoint and connects a site to the LISP-capable part of a core network (such as the Internet), publishes EID-to-RLOC mappings for the site, responds to Map-Request messages, and decapsulates and delivers LISP-encapsulated user data to the end systems at the site. During operation, an ETR sends periodic Map-Register messages to all its configured map servers. These Map-Register messages contain all the EID-to-RLOC entries for the EID-numbered networks that are connected to the ETR's site.
- **Ingress Tunnel Router (ITR)**: An ITR is a device that is the tunnel start point. An ITR is responsible for finding EID-to-RLOC mappings for all traffic destined for LISP-capable sites. When the ITR receives a packet destined for an EID, it first looks for the EID in its mapping cache. If the ITR finds a match, it encapsulates the packet inside a LISP header with one of its RLOCs as the IP source address and one of the RLOCs from the mapping cache entry as the IP destination. The ITR then routes the packet normally.
- **xTR**: A generic name for a device performing both Ingress Tunnel Router (ITR) and Egress Tunnel Router (ETR) functions.

- **PxTR**: The point of interconnection between an IP network and a LISP network, playing the role of ITR and ETR at this peering point.
- **Map-Server (MS)**: An MS is a LISP Infrastructure device that LISP site ETRs register to with their EID prefixes. An MS implements part of the distributed LISP mapping database by accepting registration requests from its client egress tunnel routers (ETRs), aggregating the successfully registered EID prefixes of those ETRs, and advertising the aggregated prefixes into the alternative logical topology (ALT) with border gateway protocol (BGP).

  In a small private mapping system deployment, an MS may be configured to stand alone (or there may be several MSs) with all ETRs configured to register to each MS. If more than one, all MSs have full knowledge of the mapping system in a private deployment.

  In a larger or public mapping system deployment, an MS is configured with a partial mesh of generic routing encapsulation (GRE) tunnels and BGP sessions to other map server systems
- **Map-Resolver (MR)**: An MR is a LISP Infrastructure device to which the ITRs send LISP Map-Request queries when resolving EID-to-RLOC mappings. MRs receive the request and select the appropriate map server

For detailed overview information on LISP and the terminologies, see Locator ID Separation Protocol Overview.

- Configure LISP Layer 2 Extension, on page 26
- Prerequisites for configuring LISP Layer 2 Extension, on page 27
- Restrictions for configuring LISP Layer 2 Extension, on page 27
- Configure LISP Layer 2 Extension, on page 27
- Verify the LISP Layer 2 Traffic between Cisco Catalyst 8000V on AWS and Cisco Catalyst 8000V on the Enterprise System, on page 31
- Support for PMD Multi-Queue, on page 32

Configure LISP Layer 2 Extension

You can deploy Cisco Catalyst 8000V on public, private, and hybrid clouds. When enterprises move to a hybrid cloud, they need to migrate the servers to the cloud without making any changes to the servers. Enterprises may want to use the same server IP address, subnet mask, and default gateway configurations. They might want to use their own IP addressing scheme in the cloud, and not be limited by the addressing scheme of the cloud provider infrastructure.

To fulfill this requirement, Cisco offers the LISP Layer 2 Extension to Cisco Catalyst 8000V running on Amazon Web Services (AWS), where the Cisco Catalyst 8000V instance acts as the bridge between the enterprise data center and the public cloud. By configuring the LISP Layer 2 Extension, you can extend your Layer 2 networks in the private data center to a public cloud to achieve host reachability between your site and the public cloud. You can also enable the migration of your application workload between the data center and the public cloud.

Benefits

- Carry out data migration with ease and optimize the workload IP address or the firewall rules in your network. Thereby, you can ensure subnet continuity with no broadcast domain extension.
- Virtually add a VM that is on the provider site to facilitate leverage cloud bursting to virtually insert a VM in the Enterprise server while the VM runs on the provider site.

- Provide backup services for partial disaster recovery and disaster avoidance.

Prerequisites for configuring LISP Layer 2 Extension

You must configure each Cisco Catalyst 8000V routerwith one external IP address. In this case, an IPsec tunnel is built between the IP addresses of the two Cisco Catalyst 8000V intances, and the IPsec tunnel has a private address.

Restrictions for configuring LISP Layer 2 Extension

- Enterprise VRF number and VM address numbers are limited on an AWS ECS subnet.
- IPv6 address format is not supported in a Cisco Catalyst 8000V Amazon Machine Image (AMI).

Configure LISP Layer 2 Extension

To configure the L2 extension functionality, you must first deploy the Cisco Catalyst 8000V instance on AWS and configure the instance as an xTR. You must then configure the mapping system to complete the deployment.

The LISP site uses the Cisco Catalyst 8000V instance configured as both an ITR and an ETR (also known as an xTR) with two connections to upstream providers. The LISP site then registers to the standalone device that you have configured as the map resolver/map server (MR/MS) in the network core. The mapping system performs LISP encapsulation and de-encapsulation of the packets that are going to the migrated public IPs. Optionally, for traffic that is leaving AWS, whenever a route to the destination is not found on the routing table, the Cisco Catalyst 8000V instance routes that traffic through the PxTR at the enterprise data center.

Perform the following steps to enable and configure the LISP xTR functionality when using a LISP map server and map resolver for mapping services:

Creating a Cisco Catalyst 8000V instance on AWS

Step 1 Log in to Amazon Web Services. In the left navigation pane, click **VPC**.

Step 2 Click the Start VPC Wizard, and select **VPC with Single Public Subnet** from the left pane.

Step 3 Click **Select**.

Step 4 Create the subnet in the Virtual Private Cloud. Use the following properties:

a) Default Subnet-10.0.0.0/24 (mapped to public IP).

b) Additional subnets-0.0.1.0/24 and 1.0.0.2.0/24. These are private IP addresses and might be internal for the Cisco Catalyst 8000V instance.

Step 5 Select **Create VPC**.

Step 6 Select **Security** > **Network ACLs**.

Step 7 Click **Create Security Group** to create a security group for the Cisco Catalyst 8000V instance. Configure the following properties:

a) Name-SSH-Access

b) TCP Port 22 traffic-Permitted inbound
c) SSH access to C8000V for management-Enabled

Step 8 To create additional security groups, perform step 6.

Step 9 Go to the Cisco Catalyst 8000V product page, and click **Continue**.

Step 10 Click **Launch with E2 Console** to launch the Cisco Catalyst 8000V in accordance with your geographical region.

Step 11 Choose the appropriate instance type. Refer tables 2-1 and 2-2 for supported instance types.

The minimum memory requirement for a medium instance type (m1.medium) is 10Mbps; large instance type (m1.large) is 50Mbps.

ECU stands for Elastic Compute Unit. ECU is Amazon's proprietary way of measuring the CPU capacity.

All the EC2 instances are hyperthreaded.

Step 12 Launch the Cisco Catalyst 8000V instance in the VPC that you created. Use the following properties:

a) Set the **Shutdown** behaviour to **Stop**.
b) Set the **Tenancy** to **Shared**. Choose the Shared option to run a shared hardware instance.

Step 13 Associate the instance with a security group (SSH-ACCESS). The security rules enable you to configure firewall rules to control traffic for your Cisco Catalyst 8000V instance.

Step 14 Associate a private key with the Cisco Catalyst 8000V instance. A key pair is a private key and a public key. You must provide the private key to authenticate and connect to the Cisco Catalyst 8000V instance. The public key is stored on AWS. If required, you can create a new key pair.

Step 15 Click **Launch Instances**.

Step 16 Verify whether the Cisco Catalyst 8000V instance is deployed on AWS.

After successful deployment, the status changes to *2/2/ checks passed*.

Configure subnets

Step 1 Select the Cisco Catalyst 8000V instance.

Step 2 Select **Actions** > **Networking** > **Manage IP Addresses**.

Step 3 Specify the enterprise host address. This IP address is the secondary address of eth1.

Step 4 Click **Yes, Update**.

Configure a tunnel between Cisco Catalyst 8000V on AWS and Cisco Catalyst 8000V on the Enterprise System

The communication between the Cisco Catalyst 8000V instance deployed within the enterprise data center and the Cisco Catalyst 8000V instance deployed within the public cloud is secured by an IP Security (IPsec) tunnel established between them. The LISP-encapsulated traffic is protected with the IPsec tunnel that provides data origin authentication, integrity protection, anti-reply protection, and confidentiality between the public cloud and the enterprise.

Step 1 Configure a Cisco Catalyst 8000V instance on AWS.

```
interface Loopback1
 ip address 33.33.33.33 255.255.255.255
!
interface Tunnel2
 ip address 30.0.0.2 255.255.255.0
 tunnel source GigabitEthernet1
 tunnel mode ipsec ipv4
 tunnel destination 173.39.145.79
 tunnel protection ipsec profile p2p_pf1
!
interface GigabitEthernet2
 ip address 10.10.10.140 255.255.255.0
 negotiation auto
 lisp mobility subnet1 nbr-proxy-reply requests 3
 no mop enabled
 no mop sysid
!
```

Step 2 Configure a second Cisco Catalyst 8000V instance on the enterprise site.

```
interface Loopback1
 ip address 11.11.11.11 255.255.255.255

interface Tunnel2
 ip address 30.0.0.1 255.255.255.0
 tunnel source GigabitEthernet2
 tunnel mode ipsec ipv4
 tunnel destination 52.14.116.161
 tunnel protection ipsec profile p2p_pf1
!
!
interface GigabitEthernet3
 ip address 10.10.10.2 255.255.255.0
 negotiation auto
 lisp mobility subnet1 nbr-proxy-reply requests 3
 no mop enabled
 no mop sysid
!
```

Configure LISP xTR on the Instance Running on AWS

To configure LISP xTR on the Cisco Catalyst 8000V instance running on AWS, follow the configuration steps in the Configuring LISP (Location ID Separation Protocol) section.

Example:

```
router lisp
 locator-set aws
  33.33.33.33 priority 1 weight 100
  exit-locator-set
 !
 service ipv4
  itr map-resolver 11.11.11.11
  itr
  etr map-server 11.11.11.11 key cisco
```

```
  etr
  use-petr 11.11.11.11
  exit-service-ipv4
 !
 instance-id 0
  dynamic-eid subnet1
   database-mapping 10.10.10.0/24 locator-set aws
   map-notify-group 239.0.0.1
   exit-dynamic-eid
  !
  service ipv4
   eid-table default
   exit-service-ipv4
  !
  exit-instance-id
 !
 exit-router-lisp
!
router ospf 11
 network 30.0.0.2 0.0.0.0 area 11
 network 33.33.33.33 0.0.0.0 area 11
!

router lisp
 locator-set dmz
  11.11.11.11 priority 1 weight 100
  exit-locator-set
 !
 service ipv4
  itr map-resolver 11.11.11.11
  etr map-server 11.11.11.11 key cisco
  etr
  proxy-etr
  proxy-itr 11.11.11.11
  map-server
  map-resolver
  exit-service-ipv4
 !
 instance-id 0
  dynamic-eid subnet1
   database-mapping 10.10.10.0/24 locator-set dmz
   map-notify-group 239.0.0.1
   exit-dynamic-eid
  !
  service ipv4
   eid-table default
   exit-service-ipv4
  !
  exit-instance-id
 !
 site DATA_CENTER
  authentication-key cisco
  eid-record 10.10.10.0/24 accept-more-specifics
  exit-site
 !
 exit-router-lisp
!
router ospf 11
 network 11.11.11.11 0.0.0.0 area 11
 network 30.0.0.1 0.0.0.0 area 11
!
```

```
!
!
```

Verify the LISP Layer 2 Traffic between Cisco Catalyst 8000V on AWS and Cisco Catalyst 8000V on the Enterprise System

Perform the following steps to verify the LISP Layer 2 traffic:

Example:

```
Router#show ip lisp database
LISP ETR IPv4 Mapping Database for EID-table default (IID 0), LSBs: 0x1
Entries total 2, no-route 0, inactive 0

10.0.1.1/32, dynamic-eid subnet1, inherited from default locator-set aws
  Locator  Pri/Wgt  Source     State
33.33.33.33    1/100  cfg-addr   site-self, reachable
10.0.1.20/32, dynamic-eid subnet1, inherited from default locator-set aws
  Locator  Pri/Wgt  Source     State
33.33.33.33    1/100  cfg-addr   site-self, reachable
Router#show ip lisp map-cache
LISP IPv4 Mapping Cache for EID-table default (IID 0), 4 entries

0.0.0.0/0, uptime: 00:09:49, expires: never, via static-send-map-request
  Negative cache entry, action: send-map-request
10.0.1.0/24, uptime: 00:09:49, expires: never, via dynamic-EID, send-map-request
  Negative cache entry, action: send-map-request
10.0.1.4/30, uptime: 00:00:55, expires: 00:00:57, via map-reply, forward-native
  Encapsulating to proxy ETR
10.0.1.100/32, uptime: 00:01:34, expires: 23:58:26, via map-reply, complete
  Locator  Uptime    State      Pri/Wgt     Encap-IID
11.11.11.11  00:01:34  up           1/100       -
Router#show lisp dynamic-eid detail
% Command accepted but obsolete, unreleased or unsupported; see documentation.

LISP Dynamic EID Information for VRF "default"

Dynamic-EID name: subnet1
  Database-mapping EID-prefix: 10.0.1.0/24, locator-set aws
  Registering more-specific dynamic-EIDs
  Map-Server(s): none configured, use global Map-Server
  Site-based multicast Map-Notify group: 239.0.0.1
  Number of roaming dynamic-EIDs discovered: 2
  Last dynamic-EID discovered: 10.0.1.20, 00:01:37 ago
    10.0.1.1, GigabitEthernet2, uptime: 00:09:23
      last activity: 00:00:42, discovered by: Packet Reception
    10.0.1.20, GigabitEthernet2, uptime: 00:01:37
      last activity: 00:00:40, discovered by: Packet Reception

Router-DC#show ip lisp
Router-DC#show ip lisp data
Router-DC#show ip lisp database
LISP ETR IPv4 Mapping Database for EID-table default (IID 0), LSBs: 0x1
Entries total 1, no-route 0, inactive 0

10.0.1.100/32, dynamic-eid subnet1, inherited from default locator-set dc
```

```
  Locator  Pri/Wgt  Source     State
11.11.11.11    1/100  cfg-addr   site-self, reachable
Router-DC#show ip lisp
Router-DC#show ip lisp map
Router-DC#show ip lisp map-cache
LISP IPv4 Mapping Cache for EID-table default (IID 0), 2 entries

10.0.1.0/24, uptime: 1d08h, expires: never, via dynamic-EID, send-map-request
  Negative cache entry, action: send-map-request
10.0.1.20/32, uptime: 00:00:35, expires: 23:59:24, via map-reply, complete
  Locator  Uptime    State      Pri/Wgt     Encap-IID
33.33.33.33  00:00:35  up           1/100

Router-DC#show lisp dynamic-eid detail
% Command accepted but obsolete, unreleased or unsupported; see documentation.

LISP Dynamic EID Information for VRF "default"

Dynamic-EID name: subnet1
  Database-mapping EID-prefix: 10.0.1.0/24, locator-set dc
  Registering more-specific dynamic-EIDs
  Map-Server(s): none configured, use global Map-Server
  Site-based multicast Map-Notify group: 239.0.0.1
  Number of roaming dynamic-EIDs discovered: 1
  Last dynamic-EID discovered: 10.0.1.100, 1d08h ago
    10.0.1.100, GigabitEthernet2, uptime: 1d08h
      last activity: 00:00:47, discovered by: Packet Reception

Router-DC#show lisp site
LISP Site Registration Information
* = Some locators are down or unreachable
# = Some registrations are sourced by reliable transport

Site Name      Last      Up     Who Last             Inst     EID Prefix
               Register         Registered           ID
dc             never     no     --                            10.0.1.0/24
               00:08:41  yes#   33.33.33.33                   10.0.1.1/32
               00:01:00  yes#   33.33.33.33                   10.0.1.20/32
               1d08h     yes#   11.11.11.11                   10.0.1.100/32
Router-DC#show ip cef 10.0.1.20
10.0.1.20/32
  nexthop 33.33.33.33 LISP0
Router-DC#
```

Support for PMD Multi-Queue

From Cisco IOS XE 17.7.1, the PMD Multi-Queue functionality is supported on Cisco Catalyst 8000V instances running on AWS. Currently, Cisco Catalyst 8000V allocates only one PMD RX queue and PMD TX queue per interface. With this functionality, Cisco Catalyst 8000V allocates 4 PMD RX queues and 8 PMD TX queues, thereby increasing the performance by increasing the packet processing rate.

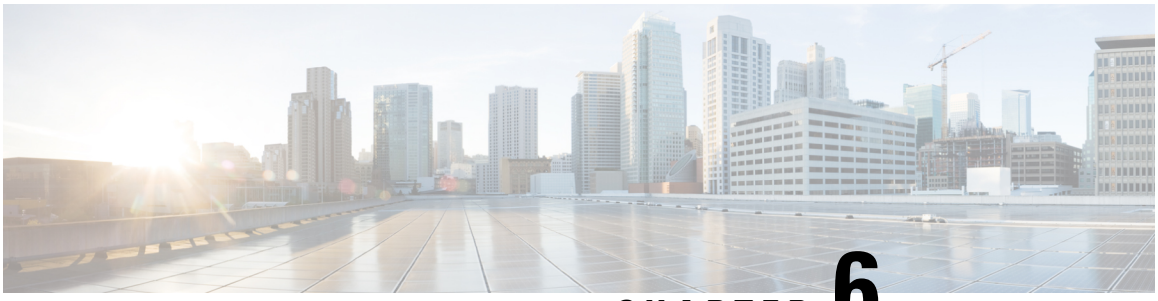
From Cisco IOS XE 17.9.1, Cisco Catalyst 8000V has the capacity to allocate 12 PMD TX queues.

Note The IP address pair in the IPsec tunnels are hashed to PMD TXQ. Therefore, the address can cause collision and reduce the performance. To avoid this issue, check whether the traffic is evenly distributed across all the 8 queues for optimal performance by using the **show platform hardware qfp active datapath infrastructure sw-nic** command.

The following is a sample command output of the **show platform hardware qfp active datapath infrastructure sw-nic** command.

```
Router# show platform hardware qfp act datapath infrastructure sw-nic
pmd b19811c0 device Gi1
  RX: pkts 418  bytes 37655 return 0 badlen 0
    pkts/burst 1  cycl/pkt 0  ext_cycl/pkt 0
    Total ring read 91995516, empty 91995113
  TX: pkts 355  bytes 57833
      pri-0: pkts 60  bytes 5590
             pkts/send 1
      pri-1: pkts 32  bytes 2616
             pkts/send 1
      pri-2: pkts 6  bytes 303
             pkts/send 1
      pri-3: pkts 38  bytes 6932
             pkts/send 1
      pri-4: pkts 176  bytes 39279
             pkts/send 1
      pri-5: pkts 25  bytes 1962
             pkts/send 1
      pri-6: pkts 8  bytes 459
             pkts/send 1
      pri-7: pkts 10  bytes 692
             pkts/send 1
  Total: pkts/send 1  cycl/pkt 3160
    send 343  sendnow 0
    forced 343  poll 0  thd_poll 0
    blocked 0  retries 0  mbuf alloc err 0
    TX Queue 0: full 0  current index 0  hiwater 0
    TX Queue 1: full 0  current index 0  hiwater 0
    TX Queue 2: full 0  current index 0  hiwater 0
    TX Queue 3: full 0  current index 0  hiwater 0
    TX Queue 4: full 0  current index 0  hiwater 0
    TX Queue 5: full 0  current index 0  hiwater 0
    TX Queue 6: full 0  current index 0  hiwater 0
    TX Queue 7: full 0  current index 0  hiwater 0
pmd b1717380 device Gi2
  RX: pkts 289216546  bytes 102405925473 return 0 badlen 0
    pkts/burst 7  cycl/pkt 326  ext_cycl/pkt 381
    Total ring read 141222555, empty 103047391
  TX: pkts 757922  bytes 260498122
      pri-0: pkts 94302  bytes 32428428
             pkts/send 1
      pri-1: pkts 95525  bytes 32791822
             pkts/send 1
      pri-2: pkts 93002  bytes 31950500
             pkts/send 1
      pri-3: pkts 96799  bytes 33381108
             pkts/send 1
      pri-4: pkts 90823  bytes 31179044
             pkts/send 1
      pri-5: pkts 97436  bytes 33455916
             pkts/send 1
      pri-6: pkts 93243  bytes 32113540
```

```
            pkts/send 1
        pri-7: pkts 96792  bytes 33197764
            pkts/send 1
    Total: pkts/send 1  cycl/pkt 760
      send 685135  sendnow 3
      forced 685117  poll 0  thd_poll 0
      blocked 0  retries 0  mbuf alloc err 0
      TX Queue 0: full 0  current index 0  hiwater 31
      TX Queue 1: full 0  current index 0  hiwater 31
      TX Queue 2: full 0  current index 0  hiwater 0
      TX Queue 3: full 0  current index 0  hiwater 0
      TX Queue 4: full 0  current index 1  hiwater 31
      TX Queue 5: full 0  current index 0  hiwater 0
      TX Queue 6: full 0  current index 0  hiwater 0
      TX Queue 7: full 0  current index 0  hiwater 0
pmd b14ad540 device Gi3
    RX: pkts 758108  bytes 302121148 return 0 badlen 0
      pkts/burst 1  cycl/pkt 572  ext_cycl/pkt 811
      Total ring read 78867251, empty 78155478
    TX: pkts 756904  bytes 301747138
        pri-0: pkts 9  bytes 540
            pkts/send 1
        pri-1: pkts 200064  bytes 80223776
            pkts/send 1
        pri-3: pkts 244086  bytes 97204792
            pkts/send 1
        pri-4: pkts 3  bytes 822
            pkts/send 1
        pri-5: pkts 250502  bytes 99404344
            pkts/send 1
        pri-7: pkts 62240  bytes 24912864
            pkts/send 1
    Total: pkts/send 1  cycl/pkt 737
      send 705364  sendnow 3
      forced 705355  poll 0  thd_poll 0
      blocked 0  retries 0  mbuf alloc err 0
      TX Queue 0: full 0  current index 0  hiwater 0
      TX Queue 1: full 0  current index 0  hiwater 31
      TX Queue 2: full 0  current index 0  hiwater 0
      TX Queue 3: full 0  current index 0  hiwater 31
      TX Queue 4: full 0  current index 0  hiwater 0
      TX Queue 5: full 0  current index 0  hiwater 0
      TX Queue 6: full 0  current index 0  hiwater 0
      TX Queue 7: full 0  current index 0  hiwater 0
```



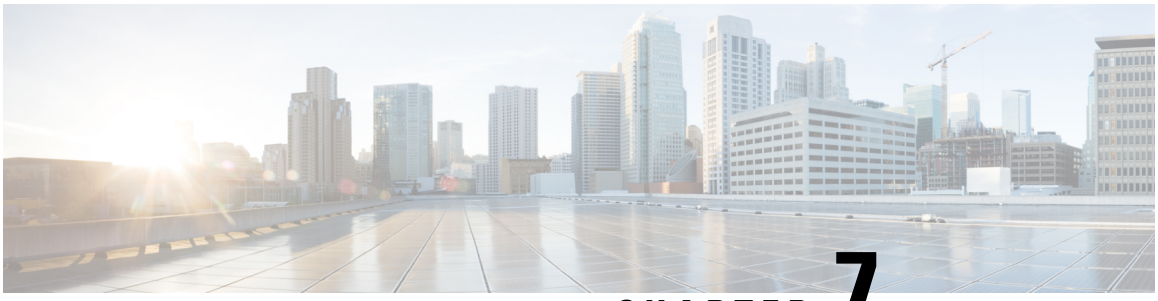
CHAPTER 6

Configure Ipv6 Functionalities

Internet Protocol version 6 (IPv6) expands the number of network address bits from 32 bits (in IPv4) to 128 bits which provides more than enough globally unique IP addresses for every networked device on the planet. The unlimited address space provided by IPv6 allows Cisco to deliver more and newer applications and services with reliability, improved user experience, and increased security.

Implementing basic IPv6 connectivity in the Cisco software consists of assigning IPv6 addresses to individual device interfaces. You can also enable IPv6 traffic forwarding globally, and Cisco Express Forwarding switching for IPv6. You can enhance basic connectivity functionality by configuring support for AAAA record types in the Domain Name System (DNS) name-to-address and address-to-name lookup processes and by managing IPv6 neighbor discovery.

IPv6 addressing is supported for Cisco Catalyst 8000V instances running on Amazon Web Services. To know how to configure the IPv6 functionalities for your instance, see the IPv6 Addressing and Basic Connectivity Configuration Guide.



CHAPTER 7

Migrating Cisco CSR1000V Instances to Cisco Catalyst 8000V Using the AWS Migration Tool

If you are a user who has deployed Cisco CSR1000V instances in AWS and you're upgrading to Cisco Catalyst 8000V, you must migrate your instances to Cisco Catalyst 8000V. Currently, you can migrate your instances from Cisco CSR1000V to Cisco Catalyst 8000V by performing a fresh Cisco Catalyst 8000V installation and copying the configuration files from Cisco CSR1000V.

From the Cisco IOS XE 17.14.1a release, to ease your effort and make the migration process seamless using automation, an AWS migration tool is available as a deployable CloudFormation template in AWS. This one-click tool enables you to automatically migrate users from existing Cisco CSR1000V deployments to fresh Cisco Catalyst 8000V deployments. This migration offers access to advanced features that are available with Cisco Catalyst 8000V such as secure object storage.

This tool is available for autonomous (non-SDWAN) PAYG and BYOL Cisco CSR1000V deployments.

To migrate your Cisco CSR1000V instances using this tool, log in to the AWS Marketplace and select the CloudFormation template. As the next step, provide additional inputs, such as the list of CSR1000V instances that need to be migrated, the Cisco IOS XE version that you want to migrate to, and so on. Submit your inputs and initiate the migration process.

The tool receives all the inputs and creates the following two AWS Lambda functions that are nothing but AWS resources that you can invoke to run your code in Lambda:

- **Trigger Lambda**: This Lambda function triggers the migration workflow. This Lambda validates whether all the prerequisites for the migration are valid. Further, this component also validates each of the CSR1000V instances that you have specified to be migrated before invoking a worker Lambda.
- **Worker Lambda**: This Lambda function processes the request from the Trigger Lambda. It launches identical Cisco Catalyst 8000V instances for every CSR1000V instance that you want to migrate, with the same cloud configuration. For every CSR1000V instance to be migrated, the tool generates one worker Lambda.

After the workflow is processed, the migration is successfully completed. The Cisco Catalyst 8000V instances will then be available in your AWS account.

- Prerequisites for the Migration, on page 38
- Limitations and Notes, on page 38
- Migrating a CSR1000V Instance to Cisco Catalyst 8000V, on page 39
- Verifying Successful Migration, on page 40
- Performing a Rollback, on page 41

Prerequisites for the Migration

- You must be subscribed to Cisco Catalyst 8000V with appropriate license, before initiating a migration. The migration tool uses the AMIs from the C8000V product listings in the marketplace, and you must be subscribed in order for the tool to get the AMIs from your AWS account.
- You must have the username and password for your Cisco CSR1000V instance handy. These credentials are used as inputs in the tool to gain access to the Cisco CSR1000V instances, since the automation does not have access to the user's PEM files. Before you begin the migration, perform the **username [user name] privilege 15 password [password]** configuration.
- Your Cisco CSR 1000V instance must have a public IP address. Public IP address is required to have connectivity over public internet so that the AWS migration tool can login to the device.
- Your Cisco CSR1000V instance must be in the autonomous mode and in a running state.

Limitations and Notes

- You can migrate only the following cloud properties: subnets, AZ, Security Groups, secondry IPs, elastic IPs, and tags (multiple tags).
- You can perform this migration only for Cisco CSR1000V instances having the t3, c5n, and c5 instance types. The instance type and the size you selected must be supported by both Cisco CSR 1000V and Cisco Catalyst 8000V.
- Migration from one license type to another is not supported, for example, you cannot migrate an instance with PayG license type to BYOL license type.
- You can perform this migration only for Cisco CSR1000V instances that run in autonomous mode. The AWS migration tool does not work with Cisco CSR1000V instances deployed in SD-WAN mode.
- At release, none of the other logs, configuration files, or packages will be migrated. The migration will fail if you provide a Cisco CSR1000V instance in controller mode.
- The following day-0 configuration commands are not supported for this migration:
  - `'Building.*?bytes\n',`

    `'hostname.*?\n',`
  - `'\ncrypto pki.*?quit',`
  - `'no aaa new-model\n',`
  - `'login local\n',`
  - `'spanning-tree extend system-id\n',`
  - `'no mop enabled\n’,`
  - `'no mop sysid'`

Any IOS configurations from Cisco CSR1000V that are not supported on Cisco Catalyst 8000V will not not be applied to the Cisco Catalyst 8000V instances. Although this scenario does not cause migration failure, you might experience discrepancies in the Cisco Catalyst 8000V configuration.

- To minimize the risk of failure, it is recommended you migrate only up to 10 Cisco CSR1000V instances at a time with a single CloudFormation deployment.
- Only the current IOS configuration that is running will be migrated.

Migrating a CSR1000V Instance to Cisco Catalyst 8000V

Step 1 Log in to the AWS Marketplace.

Step 2 Choose the appropriate Cisco CSR1000V to Cisco Catalyst 8000V Migration Tool template by clicking one of the following:

- BYOL
- PAYG Network-Advantage
- PAYG Network-Essentials

Step 3 The **Create Stack** window appears where a template is provided already. You do not have to fill in any details on this page. Click **Next** to go to the next screen.

Step 4 On the **Specify Stack Details** window, in the **Stack Name** field, specify a name for your stack. The **Name** field can contain alphabets A to Z, a to z, numbers 0 to 9, and dashes. The Stack Name must start with an alphabet and the maximum allowed character length for this field is 128.

Step 5 In the **Parameters** area, choose the **Version** that you want to migrate to.

Step 6 From the **CSR1000V Instance IDs** drop-down list, choose the instances that you want to migrate. The field lists all your current EC2 instances. Choose up to 10 Cisco CSR1000V instances to migrate at one time. After a successful migration, you will have equal number of Cisco Catalyst 8000V instances.

Step 7 (Optional) In the **C8000V BYOL Token** field, provide the token so that it is registered on the Cisco Catalyst 8000V instances. This token is required for the Cisco Catalyst 8000V to boot up with the proper licensing level. This token is required for Cisco Catalyst 8000V to establish trust with the CSSM server and get the target license registered on Cisco Catalyst 8000V. This field is required only when you migrate a BYOL Cisco CSR1000V instance. If you do not provide this token, the migration will go through. However, the tool will not register a license in Cisco Catalyst 8000V.

Step 8 In the **SSH Key to Associate With C8000V** field, enter the **SSH key**. Use this key to log in to your Cisco Catalyst 8000V instance after it is created.

Step 9 (Optional) Specify the SNS ARN in the **SNS Topic ARN** field if you want to receive failure notifications through email and SMS.

Note You must enable the Amazon Simple Notification Service and have the ARN handy before you specify the SRS ARN.

Step 10 In the **Username to SSH Into Device** field, enter the username that you created in your CSR1000V instance for the migration.

Step 11 In the **Password to SSH Into Device field**, enter the password that you created in your CSR1000V instance for the migration.

This SSH username and password enable SSH access to the CSR1000V instances. The same SSH credentials are used to log in to Cisco Catalyst 8000V once the migration is complete.

Step 12 From the **Rollback Option** drop-down list, choose **Yes** if you want the stack to rollback the resources and re-deploy your Cisco CSR1000V instances if the migration fails.

Step 13 From the **Terminate C8KV Option** drop-down list, choose **Yes** if you want to terminate the Cisco Catalyst 8000V instances if the migration fails.

Step 14 From the **Terminate CSR Option** drop-down list, choose **Yes** if you want to automatically terminate the CSR1000V instances after successful migration. Otherwise, choose **No**.

Step 15 Click **Next**.

Step 16 (Optional) In the **Stack Failure Options** area, in the **Select Rollback All Stack Resources** field, click the **Rollback to Last Known Stable State** radio button.

Step 17 Click **Next**.

Step 18 On the Review and Create window, review all your settings. At the bottom of this page, acknowledge the resources that will be created in this migration by checking the check boxes.

Step 19 Click **Submit** to launch the migration.

Verifying Successful Migration

This task specifies the steps to verify whether the migration of instances from Cisco CSR1000V to Cisco Catalyst 8000V is successful.

Before you begin

Perform these verification steps after you initiate the migration.

Step 1 After you submit the stack, the **CloudFormation Stack** window appears. View the creation and completion of the stack events on this page. The events of the migration tool trigger the lambda functions that validate the Cisco CSR1000V instances and create identical Cisco Catalyst 8000V instances. The CREATE_COMPLETE of the stack indicates a successful creation of stack resources, or the lambda functions.

Step 2 To view each of the resources and logs for the migration, go to **CloudWatch** > **Log Groups**. Your stack name will be displayed on this window. Click your stack name to view all the details and logs that are related to this stack.

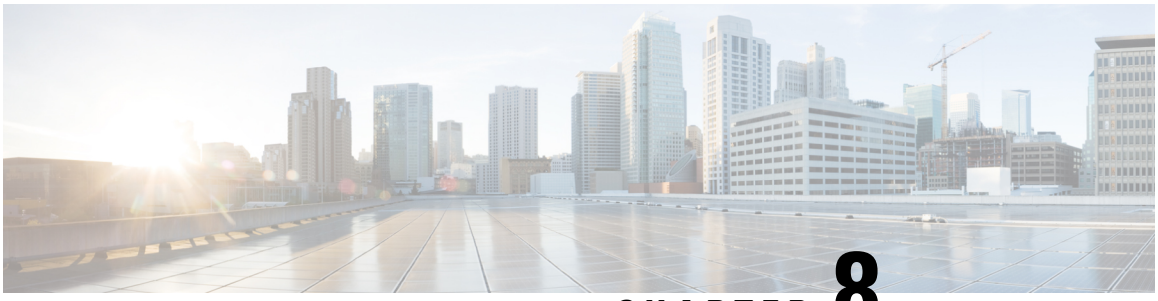
Step 3 Verify whether the Cisco Catalyst 8000V instances are created. The Cisco Catalyst 8000V instances have the same name as your CSR1000V instances. However, the instance IDs will be different for the CSR1000V instances and the corresponding Cisco Catalyst 8000V instances.

Note If a migration fails during the Cisco Catalyst 8000V deployment, the Cisco Catalyst 8000V console logs are saved to the migration S3 bucket. You can also choose not to terminate the Cisco Catalyst 8000V instances to debug a failed migration on the new Cisco Catalyst 8000V device.

Performing a Rollback

If the migration workflow is not successful, the rollback process occurs automatically. However, ensure that you have chosen **Yes** from the **Rollback Option** drop-down list before you click **Submit** and trigger the migration.

If you chose **No** from the **Rollback Option** drop-down list, the migration stops and the Cisco CSR1000V instances will not be restored to its running state.



CHAPTER 8

Deploying Transit VPC With Transit Gateway

Information About the Transit Gateway Solution

Amazon Virtual Private Cloud (Amazon VPC) provides you with the ability to create as many virtual networks as you need. AWS also provides different options for connecting these networks to each other and to non-AWS infrastructure, such as on-premises data centres, remote headquarters, or other offices.

When you deploy a Cisco Catalyst 8000V instance with the Transit VPC solution, you can build a hub-and-spoke topology on Amazon VPCs to centralize edge connectivity. Transit VPC allows you to implement shared services or packet inspection/replication in a VPC. It works across accounts and is easy to set up through an AWS CloudFormation stack. However, there is some level of complexity while adding a new spoke as this solution uses a VPN Gateway as opposed to the Transit Gateway.

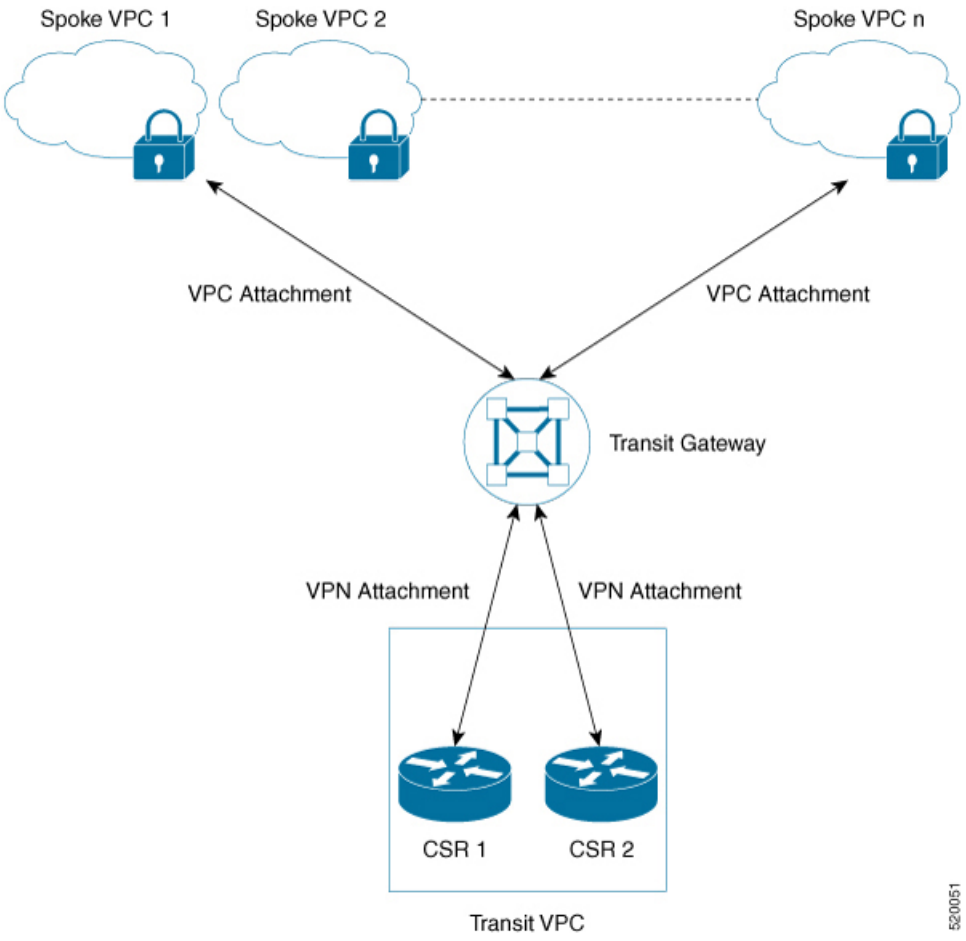
To overcome this limitation, you can now deploy a Cisco Catalyst 8000V Transit VPC with the Transit Gateway solution. A transit gateway is a regional network transit hub service provided by AWS to interconnect your VPCs in AWS cloud and on-premise network. In the Cisco Catalyst 8000V transit VPC with transit gateway solution, you use a transit gateway on the spoke side to provide connectivity between all the spoke VPCs in the same region. The transit gateway is attached to two Cisco Catalyst 8000V instances in the transit VPC using a VPN attachment. The Cisco Catalyst 8000V instance provides VPN connectivity to various on-premise branch locations.

To know how to deploy the AWS Transit VPC with Transit Gateway solution, perform the configuration steps as mentioned in this chapter.

Transit VPC-Transit Gateway Components

The Transit Gateway solution has a transit gateway that acts as a hub for providing spoke-to-spoke VPC connectivity. The transit VPC is another core component that acts as the central hub for traffic flowing from any spoke VPC to a remote network. The transit VPC hosts two Cisco Catalyst 8000V instances that allow VPN termination and routing.

Figure 3: Sample Topology of the Transit Gateway Solution



This solution uses two AWS Lambda functions, the Solution Helper, and the Cisco Configurator to automatically configure the VPN connections between these instances and the spoke VPCs.

- **Solution Helper Lambda**: This component is triggered when you deploy the cloudformation template. This component creates the transit gateway, the VPN connections with the Cisco Catalyst 8000V instances, and the VPN attachment between the instances and the transit gateway. The lambda function then saves the VPN connection information to the Amazon S3 bucket using S3 SSE-KMS.
- **Cisco Configurator Lambda**: The S3 Put event invokes the Cisco Configurator Lambda function which parses the VPN connection information and generates the necessary configuration files to create new VPN connections. The Cisco Configurator Lambda pushes the IOS configuration to the Cisco Catalyst 8000V instances using SSH. As soon as the Cisco configuration is applied on the Cisco Catalyst 8000V instances, the VPN tunnels come up and the Border Gateway Protocol (BGP) neighbour relationships are established with the transit gateway.

- Benefits of the AWS Transit Gateway Solution, on page 45
- Prerequisites to the AWS Transit Gateway Solution, on page 45
- Limitations of the AWS Transit Gateway Solution, on page 45
- Configuring the AWS Transit Gateway Solution, on page 45
- Configuration Example, on page 47

Benefits of the AWS Transit Gateway Solution

- The Transit Gateway solution is scalable and resilient.
- The Transit Gateway solution is a managed service. That is, high availability and monitoring is built-in, and you can track the solution using metrics like CloudWatch.
- By using the Transit Gateway solution, you can simplify your network architecture, thereby reducing the operational cost.
- You can centrally manage your solution, including security.

Prerequisites to the AWS Transit Gateway Solution

- You must have sufficient Elastic IP, VPC, TGW and VPN connection limits.
- Ensure that you have IAM permission to manage the *cloudformation* service.

Limitations of the AWS Transit Gateway Solution

- Autoscaling is not supported with this version of the solution.
- You must manually add the spoke VPCs to the Transit Gateway through VPC attachments after you deploy this solution.

Configuring the AWS Transit Gateway Solution

Step 1 Log in to the Amazon Web Services Marketplace.

Step 2 Search the **Cisco Catalyst 8000V – Transit Network VPC** template and select the template.

Step 3 Launch the template in the appropriate region where you are located. The system displays the AWS Cloudformation Service page. Click **Next**.

Step 4 Specify the following **Stack Details**:

Parameter	Description
C8000V Throughput Requirements	The required throughput for the Cisco Catalyst 8000V instance. This determines the instance type to be launched. The default value is 2 x 500 Mbps.
SSH Key to access C8000V	The public/private key pair that allows a secure connection to be made to a Cisco Catalyst 8000V instance after you launch the instance.

Parameter	Description
	You must enter a public/private key pair. The key pair is created in your preferred region at the time when you created the AWS account.
License Model	BYOL is the only license model that is currently supported.
Enable Termination Protection	Enable this field to enable termination protection for the Cisco Catalyst 8000V instances. This prevents accidental Cisco Catalyst 8000V termination. Cisco recommends that you enable this field for production deployments. By default, this field is set to **Yes**.
Prefix for S3 Objects	The text string that you need to use as a prefix when Amazon S3 objects are created. By default, the value is **vpnconfigs/**.
Additional AWS Account ID	The account ID of an AWS account associated with the transit network which allows access to the S3 bucket and the AWS KMS customer master key. **Note** You can only enter one additional AWS account ID in this field. If you want to connect more than one additional AWS account to the transit network, you must manually configure the permissions for the additional accounts.
Transit VPC CIDR Block	The CIDR block for the transit VPC. Modify the VPC and subnet CIDR address ranges to avoid collisions with your network. By default, the value is **100.64.127.224/27**.
1st Subnet Network	The CIDR block for the transit VPC subnet created in AZ1. By default, the value is **100.64.127.224/28**.
2nd Subnet Network	The CIDR block for the transit VPC subnet created in AZ2. By default, the value is **100.64.127.240/28**.
Transit VPC BGP ASN	The BGP Autonomous System Number (ASN) for the transit VPC. By default, the value is **64512**.
Spoke VPC Tag Name	The tag to use to identify the spoke VPCs to connect to the Transit VPC.
Preferred VPN Endpoint Tag Name	The tag to use to configure a preferred Cisco Catalyst 8000V VPN endpoint to control the traffic flow through the Transit VPC Cisco Catalyst 8000V instances. For example, when integrating with stateful on-prem firewalls.
Optional AZ configuration 1st Subnet	The availability Zone number for Public Subnet1.
Optional AZ configuration 2nd Subnet	The availability Zone number for Public Subnet2.

Step 5 Review and confirm the settings. Select the check box to acknowledge that resources might be created by the AWS Identity and Access Management (IAM) and CAPABILITY_AUTO_EXPAND capabilities might be required.

Step 6 Click **Create** to deploy the stack.
If the deployment is successful, the **Status** column in the AWS Cloud Formation console displays **CREATE_COMPLETE**.

Configuration Example

The following is a configuration example of deploying the AWS Transit VPC with Transit Gateway solution:

```
ip-100-64-127-234#sh run
Building configuration...

Current configuration : 7284 bytes
!
! Last configuration change at 14:10:57 UTC Thu Oct 10 2020
!
version 17.4
service timestamps debug datetime msec
service timestamps log datetime msec
service password-encryption
platform qfp utilization monitor load 80
no platform punt-keepalive disable-kernel-core
platform console virtual
!
hostname ip-100-64-127-234
!
boot-start-marker
boot-end-marker
!
!
vrf definition GS
 rd 100:100
 !
 address-family ipv4
 exit-address-family
!
logging persistent size 1000000 filesize 8192 immediate
!
no aaa new-model
!
ip vrf vpn-0f56b2afc60b1d492
 rd 64525:1
 route-target export 64525:0
 route-target import 64525:0
!
ip vrf vpn0
 rd 64525:0
!
ip admission watch-list expiry-time 0
!
subscriber templating
!
multilink bundle-name authenticated
!
crypto pki trustpoint TP-self-signed-572041569
 enrollment selfsigned
subject-name cn=IOS-Self-Signed-Certificate-572041569
```

```
 revocation-check none
 rsakeypair TP-self-signed-572041569
!
!
crypto pki certificate chain TP-self-signed-572041569
 certificate self-signed 01
  3082032E 30820216 A0030201 02020101 300D0609 2A864886 F70D0101 05050030
  30312E30 2C060355 04031325 494F532D 53656C66 2D536967 6E65642D 43657274
  69666963 6174652D 35373230 34313536 39301E17 0D313931 30313031 34303631
  355A170D 33303031 30313030 30303030 5A303031 2E302C06 03550403 1325494F
  532D5365 6C662D53 69676E65 642D4365 72746966 69636174 652D3537 32303431
  35363930 82012230 0D06092A 864886F7 0D010101 05000382 010F0030 82010A02
  82010100 A974EDB7 292BBB6A 09026F6A 381F7852 714775E3 E25F1F89 CED40FCB
  F45204F9 2F2F5FEE C46A9D16 A8D7307A C5433234 10D3F709 B4B18B3D 009B4A7A
  85980EEB 1282D1F7 C3CD4429 16042D4D 544315F4 E3ABA673 21E66C52 187AD1E6
  6B21F98A F0537D0A 8171618E 6CDF3B70 E2C8B553 8096C2D6 B4CD1AE4 B6DFD615
  844924B8 83DBE166 3CBC90F1 889CB00F 1644ECCE F2E70D81 CA35B555 D9757BE4
  34440FD9 D15580FA C50181CD D646AB6C 22F707A7 1D9F98CA 19897AF4 7488762B
  35ECA78F D2B249C7 8079255F 72BE5CF8 214B5135 E97B1104 A9CB449E A4A1D996
  9B99EC0E 18EF94FE FE73706A BF417262 12771D33 FF61A325 4479CAFB 10D0EEAA
  810E3437 02030100 01A35330 51300F06 03551D13 0101FF04 05300301 01FF301F
  0603551D 23041830 16801476 E85FEE9B EAE114A4 74C542FD E923856D 6F17F830
  1D060355 1D0E0416 041476E8 5FEE9BEA E114A474 C542FDE9 23856D6F 17F8300D
  06092A86 4886F70D 01010505 00038201 010043A6 03287F7E 1F13A7D4 26D661FE
  D11FED41 FE195D3E 6ADEA111 267C534B 266F587A 6A2F395D C50F5894 4C01F62B
  A179B852 F5F8ED62 DFF35587 3CFF352C 523F8D3D 8A786E61 A73EA8BB C8FC0A8D
  C2F0C260 0BB25D28 01B26B2B 27D71A31 2CE81DA5 6296D4AA 756A6658 0ADB89FB
  52BE1E9F A8BF17AA B2A0379A 1921AF64 834455CF B6307205 CE12C83A 2D29AEF2
  D79B79F7 9701F86E EB51B8E2 95BA7D5A C67A05F8 2AA7A8E0 3626D155 FC2D79EC
  9506D897 D79B8E65 A1D89F8A 6EC21FD1 15BFBD79 8A6FEB77 15C10DEE 0A50A7A5
  C8109573 9C58A869 D2740BC4 61D953F2 7AA92870 69BF035C 08DA0EFB B4AB9AC1
  BD4DB053 66ADD9E3 B5957D2B 8E467A91 258A
   quit
!
license udi pid CSR1000V sn 9YGGWBVUY3N
no license smart enable
diagnostic bootup level minimal
!
spanning-tree extend system-id
!
username ec2-user privilege 15 secret 5 $1$Gf9p$OfANl/ujuCIvpunuRDwKi1
username automate privilege 15 secret 8
$8$g62y2e1pz0O4/n$M8DmVAM/G9yySvjbB1I2tBJAW4IWZRIc44Icent4bps
!
redundancy
!
crypto keyring keyring-vpn-0f56b2afc60b1d492-2
  local-address GigabitEthernet1
  pre-shared-key address 52.54.79.47 key lhvPlpTYxUTno.lNTbR25F9743HEguaH
crypto keyring keyring-vpn-0f56b2afc60b1d492-1
  local-address GigabitEthernet1
  pre-shared-key address 52.44.80.94 key Qq4fLolOMf1iW3d7gJhtzF8h8Tu3IlNT
!
crypto isakmp policy 200
 encr aes
 authentication pre-share
 group 2
 lifetime 28800
crypto isakmp keepalive 10 10 periodic
crypto isakmp profile isakmp-vpn-0f56b2afc60b1d492-1
   keyring keyring-vpn-0f56b2afc60b1d492-1
   match identity address 52.44.80.94 255.255.255.255
   local-address GigabitEthernet1
   rekey
```

```
crypto isakmp profile isakmp-vpn-0f56b2afc60b1d492-2
   keyring keyring-vpn-0f56b2afc60b1d492-2
   match identity address 52.54.79.47 255.255.255.255
   local-address GigabitEthernet1
   rekey
!
crypto ipsec security-association replay window-size 1024
!
crypto ipsec transform-set ipsec-prop-vpn-aws esp-aes esp-sha-hmac
 mode tunnel
crypto ipsec df-bit clear
no crypto ipsec nat-transparency udp-encapsulation
!
crypto ipsec profile ipsec-vpn-aws
 set transform-set ipsec-prop-vpn-aws
 set pfs group2
!
interface Tunnel1
 description vpn-0f56b2afc60b1d492 from TGW to cgw-00d8fbb76cc59295e for account 902347396780

 ip vrf forwarding vpn-0f56b2afc60b1d492
 ip address 169.254.185.70 255.255.255.252
 ip tcp adjust-mss 1387
 tunnel source GigabitEthernet1
 tunnel mode ipsec ipv4
 tunnel destination 52.44.80.94
 tunnel protection ipsec profile ipsec-vpn-aws
 ip virtual-reassembly
!
interface Tunnel2
 description vpn-0f56b2afc60b1d492 from TGW to cgw-00d8fbb76cc59295e for account 902347396780

 ip vrf forwarding vpn-0f56b2afc60b1d492
 ip address 169.254.232.90 255.255.255.252
 ip tcp adjust-mss 1387
 tunnel source GigabitEthernet1
 tunnel mode ipsec ipv4
 tunnel destination 52.54.79.47
 tunnel protection ipsec profile ipsec-vpn-aws
 ip virtual-reassembly
!
interface VirtualPortGroup0
 vrf forwarding GS
 ip address 192.168.35.101 255.255.255.0
 ip nat inside
 no mop enabled
 no mop sysid
!
interface GigabitEthernet1
 ip address 100.64.127.234 255.255.255.240
 ip nat outside
 negotiation auto
 no mop enabled
 no mop sysid
!
router bgp 64525
 bgp log-neighbor-changes
 !
 address-family ipv4 vrf vpn-0f56b2afc60b1d492
  neighbor 169.254.185.69 remote-as 64526
  neighbor 169.254.185.69 timers 10 30 30
  neighbor 169.254.185.69 activate
  neighbor 169.254.185.69 next-hop-self
  neighbor 169.254.185.69 default-originate
```

```
  neighbor 169.254.185.69 as-override
  neighbor 169.254.185.69 soft-reconfiguration inbound
  neighbor 169.254.232.89 remote-as 64526
  neighbor 169.254.232.89 timers 10 30 30
  neighbor 169.254.232.89 activate
  neighbor 169.254.232.89 next-hop-self
  neighbor 169.254.232.89 default-originate
  neighbor 169.254.232.89 as-override
  neighbor 169.254.232.89 soft-reconfiguration inbound
 exit-address-family
!
iox
ip nat inside source list GS_NAT_ACL interface GigabitEthernet1 vrf GS overload
ip forward-protocol nd
ip tcp window-size 8192
ip http server
ip http authentication local
ip http secure-server
ip route 0.0.0.0 0.0.0.0 GigabitEthernet1 100.64.127.225
ip route vrf GS 0.0.0.0 0.0.0.0 GigabitEthernet1 100.64.127.225 global
!
ip ssh rsa keypair-name ssh-key
ip ssh version 2
ip ssh pubkey-chain
  username ec2-user
   key-hash ssh-rsa F1B0DF92FC2E25F7D98A01B99FCE5F13 ec2-user
  username automate
   key-hash ssh-rsa ED4B0757CE2AC22C89B28BE55EDE7691
ip ssh server algorithm authentication publickey
ip scp server enable
!
ip access-list standard GS_NAT_ACL
 permit 192.168.35.0 0.0.0.255
!
control-plane
!
line con 0
 stopbits 1
line vty 0 4
 login local
 transport input ssh
!
app-hosting appid guestshell
 app-vnic gateway1 virtualportgroup 0 guest-interface 0
  guest-ipaddress 192.168.35.102 netmask 255.255.255.0
 app-default-gateway 192.168.35.101 guest-interface 0
 name-server0 8.8.8.8
end
```