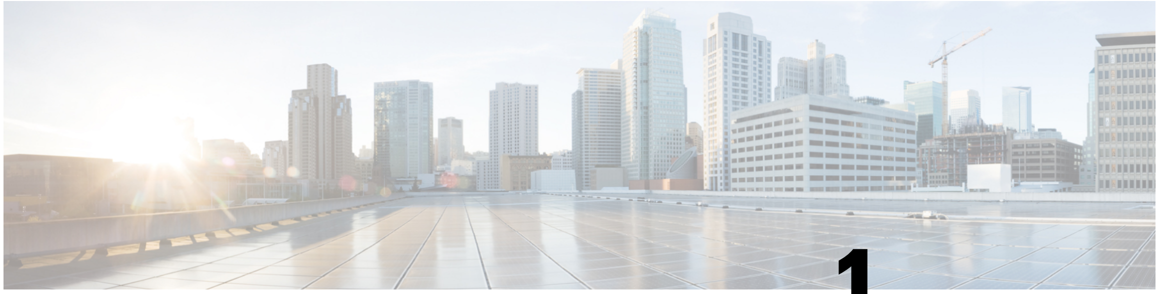
Cisco CSR1000V Series Cloud Services Router Deployment Guide for Amazon Web Services

First Published: 2018-07-18

Last Modified: 2020-05-15

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

CONTENTS



CHAPTER 1

Overview of Cisco CSR 1000v Deployment on Amazon Web Services

This section contains the following topics:

Introduction

The Cisco CSR 1000v can be deployed on Amazon Web Services (AWS) for public and private cloud solutions. The implementation and installation of the CSR 1000v on AWS is different than for the other supported hypervisors. The Cisco CSR 1000v is supported on the Amazon Virtual Private Cloud (Amazon VPC). For more information, see the AWS VPC documentation at: http://aws.amazon.com/documentation/vpc/.

Cisco CSR 1000v AMI Options for Amazon Web Services

The Cisco CSR 1000v for AWS is purchased and launched as an Amazon Machine Image (AMI) on AWS Marketplace . See Bring Your Own License, on page 2 and Cisco CSR 1000v Hourly-Billed AMIs, on page 2 .

Notes

(Cisco IOS XE Everest 16.5 and later) You can use the Cisco CSR 1000v .bin file to upgrade the version of the Cisco CSR 1000v, without having to recreate an AWS EC2 instance from a new AMI.

(Cisco IOS XE Everest 16.4 and earlier) You cannot use the Cisco CSR 1000v .bin file to upgrade the release version of an AMI. You must create a new AMI instance and migrate your configuration and license(s).

(Cisco IOS XE 3.11 or 3.12) If you are using a BYOL AMI, the Cisco IOS XE technology packages that are available are: Advanced and Premium.

(Cisco IOS XE 3.11 or 3.12) If you are using an hourly billed AMI, the Cisco IOS XE technology package that is available is: Advanced.

Bring Your Own License

The Cisco CSR 1000v for AWS is purchased and launched as an Amazon Machine Image (AMI) on AWS Marketplace.

To use the BYOL AMI, you purchase the Cisco CSR 1000v software license(s) directly from Cisco and launch the Bring Your Own license (BYOL) AMI from the AWS Marketplace. After you deploy the Cisco CSR 1000v AMI from AWS Marketplace and launch the instance, you install the Cisco licenses using the standard Cisco Software Activation process.

Licensing for the Cisco CSR 1000v BYOL AMI has the following characteristics:

- You purchase the Cisco CSR 1000v software licenses directly from Cisco, and you pay only the hourly usage fees for the AWS VPC.
- Each software license can be used for only on AWS instance.
- You can install more than one license on an AWS instance, but the multiple licenses can apply only to that instance.
- You can rehost the license if required using the Cisco Software Licensing tool. The process for rehosting a license used on a BYOL AMI is the same as for other Cisco CSR 1000v licenses.
- Cisco CSR 1000v License Activation Required using the Cisco IOS XE software activation commands after first booting the Cisco CSR 1000v.
- If you are using smart licensing on your Cisco CSR 1000v, ensure that the outbound rules of the security group allow port 443 (for HTTPS) or 80 (for HTTP) for smart licensing. Set the destination address to the address of the Cisco smart licensing server; for example:

```
https://72.163.4.38/its/service/oddce/services/DDCEService
```

Note Cisco may change this IP address for licensing in future.

The following Cisco IOS XE Technology Packages are available: (Cisco IOS XE 3.13S and later) IPBase, Security, AX and APPX.

Note The 1-Click Launch option is not currently supported for BYOL AMIs.

For more information about the Cisco CSR 1000v software licenses and the process for rehosting a license, see the Cisco CSR 1000V Series Cloud Services Router Software Configuration Guide. For a list of license SKUs, see the Cisco CSR 1000v Series Release Notes.

Cisco CSR 1000v Hourly-Billed AMIs

The Cisco CSR 1000v for AWS is purchased and launched as an Amazon Machine Image (AMI) on AWS Marketplace. This section describes the Hourly-Billed AMI.

A Cisco CSR 1000v hourly-billed AMI, launched directly from AWS Marketplace, is subject to the following conditions:

- You are billed hourly by Amazon Web Services (AWS) for using the Cisco CSR 1000v AMI. This hourly usage fee is in addition to the VPC usage fees charged by AWS.
- You do not purchase licenses directly from Cisco for the Cisco CSR 1000v.
- You do not install Cisco licenses on the router using the Cisco Software Activation process.
- The feature content of the hourly-billed AMIs corresponds to the Advanced or Premium technology package license available for the Cisco CSR 1000v. Note that some features and technologies are not supported on AWS deployments. See Cisco IOS XE Technologies Not Supported, on page 3.

The following Cisco IOS XE Technology Packages are available (Cisco IOS XE 3.13S and later): Security and AX.

Note Cisco CSR 1000v hourly-billed AMIs that correspond to the Standard technology package are not available.

- You cannot change the feature package of hourly-billed AMIs using the **license boot level** command.
- Hourly-billed AMIs cannot be rehosted.
- For more information about the features contained in the Cisco CSR 1000v technology packages, see the Cisco CSR 1000V Series Cloud Services Router Software Configuration Guide.

Cisco IOS XE Technologies Not Supported

When deployed on an AWS instance, the Cisco CSR 1000v supports fewer Cisco IOS XE technologies than are supported by other hypervisors. Some technologies may not be available because they are not supported in an Amazon cloud.

The following restrictions apply to deploying the Cisco CSR 1000v on an AWS instance:

- Although CLI commands for unsupported features may be visible on the Cisco CSR 1000v, testing by Cisco has determined that these unsupported features do not work in AWS deployments.
- Routing protocols are supported over a tunnel only.
- (Cisco IOS XE 3.11S and 3.12S) The following restrictions apply for supporting management of the router using the REST API or remote management using Cisco Prime Network Services Controller: The Cisco CSR 1000v AMI does not support management of the router using the REST API.
- (Cisco IOS XE 3.13S) The following restrictions apply for supporting management of the router using the REST API or remote management using Cisco Prime Network Services Controller: The Cisco CSR 1000v AMI supports management of the router using the REST API, but only if the shared management interface is used. For more information, see the “Configuring Support for Management Using the REST API” chapter in the Cisco CSR 1000v Series Cloud Services Router Software Configuration Guide
- (All Releases of Cisco IOS XE) The following restrictions apply for supporting management of the router using the REST API or remote management using Cisco Prime Network Services Controller: The Cisco

CSR 1000V 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 the Cisco CSR 1000v on an AWS instance.

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

Technology	Non-Supported Features
IP	IPv6 Forwarding and IPv6 Routing
Basic Routing	OSPF
IP Multicast	IGMP and PIM
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:

- Only one interface can be configured with the **ip address dhcp** command.
- 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 CSR 1000V 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.
- We recommend that you disable the Source/Destination check on the Cisco CSR 1000V interfaces.
- EtherChannel is not supported.

CHAPTER 2

Deploying the Cisco CSR 1000v on Amazon Web Services

This section contains the following topics:

Information About Launching Cisco CSR 1000v on AWS

Launching a Cisco CSR 1000v AMI occurs directly from the AWS Marketplace. Determine whether the Cisco CSR 1000v will be deployed on an Amazon EC2 instance or on an Amazon VPC instance. To proceed with Launching the Cisco CSR 1000v on AWS, perform the steps in the Launching the Cisco CSR 1000v AMI, on page 7 section.

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

Encrypted Elastic Block Storage (EBS)

When you launch a Cisco CSR 1000v from AWS marketplace, you cannot select encrypted Elastic Block Storage (EBS). (This is because encryption is not enabled on the Cisco CSR 1000v in the AMI that is available in the AWS marketplace.) However, you can follow the procedure Creating an AMI with Encrypted Elastic Block Storage, on page 13. This process is summarized below:

1. Create a CSR 1000v instance from the AWS marketplace
2. Take a snapshot of this CSR 1000v instance
3. Create a private AMI based on the snapshot
4. Copy the private AMI to a new AMI and select "Encrypt target EBS snapshots"

For further details, see Creating an AMI with Encrypted Elastic Block Storage, on page 13.

Jumbo frames in a VPC have limitations; see this document: Network Maximum Transmission Unit (MTU) for Your EC2 Instance.

Supported Instance Types

The Amazon Machine Image supports different instance types, which determine the size of the instance and the required amount of memory.

For information about supported instance types, see Cisco Cloud Services Router (CSR) 1000V for AWS.

Note To determine the maximum number of network interfaces supported per instance, see the Amazon Web Services documentation: Private IP Addresses Per Network Interface Per Instance Type

Prerequisites

Before attempting to launch the Cisco CSR 1000V on AWS, the following prerequisites apply:

- You must have an Amazon Web Services account.
- An SSH client (for example, Putty on Windows or Terminal on Macintosh) is required to access the Cisco CSR 1000v console.
- Determine the instance type that you want to deploy for the Cisco CSR 1000v. See the next section for more information.
- If you are planning to launch the AMI using the 1-Click Launch, you must first create a Virtual Private Cloud (VPC). For more information, see Amazon Virtual Private Cloud (VPC).

Note If you have deployed a CSR 1000v 16.9.X version running on AWS c5 instance, you cannot downgrade the CSR 1000v to 16.6.x versions. If you want to downgrade, you must deploy another instance type. For example, a c4.xlarge instance type.

Restrictions

The following are the restrictions when you launch the Cisco CSR 1000V on AWS:

- If you have deployed a CSR 1000v 16.9.X version running on AWS c5 instance, you cannot downgrade the CSR 1000v to 16.6.x versions. If you want to downgrade, you must deploy another instance type. For example, a c4.xlarge instance type.
- When you deploy a CSR 1000v with lower instance sizes, for example t2.medium and c4.large, the system might display the following error due to unavailability of 64k memory buffers: *%IOSXE-3-PLATFORM: R0/0: cpp_cp: QFP:0.0 Thread:000 TS:00000000023867716444 %POSIX_PMD-3-MBUF_REDUCE: Failed to allocate 65536 packet buffers. Reduced to 39480.*

Launching the Cisco CSR 1000v AMI

To launch the Cisco CSR 1000v AMI, perform the steps in the following sections:

First, see: Selecting the Cisco CSR 1000v AMI , on page 7.

If you are using an Amazon VPC instance, see: Launching the Cisco CSR 1000v AMI Using the 1-Click Launch, on page 7.

Or, if you are using an Amazon EC2 instance, see: Launching the Cisco CSR 1000v AMI Using the Manual Launch, on page 9.

Then, see: Associating the Public IP Address with Cisco CSR 1000v Instance, on page 12 and Connecting to the CSR 1000v Instance using SSH, on page 12.

If you are using a BYOL AMI, see Bring Your Own License, on page 2 and Downloading and Installing the License (BYOL AMI Only), on page 14.

Selecting the Cisco CSR 1000v AMI

To select the Cisco CSR 1000v AMI, perform the following steps:

Procedure

Step 1 Log in to Amazon Web Services Marketplace.

Step 2 Search AWS Marketplace for: “Cisco CSR 1000v”. A list of AMIs such as the following, appears:

- Cisco Cloud Services Router (CSR) 1000V - AX Pkg. Max Performance (hourly billing)
- Cisco Cloud Services Router (CSR) 1000V - Security Pkg. Max Performance (hourly billing)
- Cisco Cloud Services Router (CSR) 1000V - BYOL for Maximum Performance (BYOL billing)

Step 3 Select the Cisco CSR 1000v AMI that you are planning to deploy.

The AMI information page displays, showing the supported instance types and the hourly fees charged by AWS. Select the pricing details for your region.

Click **Continue**.

Step 4 Enter your AWS email address and password, or create a new account.

The “Launch on EC2 page” displays.

Launching the Cisco CSR 1000v AMI Using the 1-Click Launch

(Perform the following steps if you are using an Amazon VPC instance. If you are using an Amazon EC2 instance, see the Launching the Cisco CSR 1000v AMI Using the Manual Launch, on page 9).

Note Depending on the release version, the 1-Click Launch option may not be available.

Prerequisite

If you launch the AMI using the 1-Click Launch, you must first create a Virtual Private Cloud (VPC). For more information, see the AWS documentation.

Procedure

Step 1 On the Launch with EC2 page, choose the Cisco CSR 1000v release version from the Select a Version drop-down list.

Step 2 Select the Region from the drop-down list.

The hourly usage charges for your region are shown under Pricing Details.

Step 3 Select the EC2 instance type from the drop-down menu.

Step 4 Under VPC Settings, click the **Set up** button.

The VPC Settings screen displays.

Step 5 For VPC, select the VPC that you created.

Step 6 For Network interface (Public Subnet), select the interface created in the VPC.

Step 7 The security group for the public subnet is automatically created for the VPC.

This security group is predefined. You can change the security group settings after the AMI has launched within AWS. For more information, see the AWS documentation; for example, see: Amazon EC2 Security Groups for Linux Instances.

Step 8 Select the Network Interface (private subnet) in your VPC.

Step 9 Click **Done**.

Step 10 Enter the key pair information. The key pair consists of a public key stored in AWS and your private key used to authenticate access to the instance. Do one of the following:

a) Choose an existing key pair, or
b) Create a new key by performing the following steps:

- Upload your own public key.
- Click on **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. A newly created private key can only be accessed once. After the key pair is downloaded, click **Close**.

Click **Done**. The Launch on EC2 display reappears.

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 enter the following command: **chmod 400** *pem-file-name*

Step 11 Click on the Launch with 1-Click button to launch the AMI instance.

Step 12 The CSR 1000v AMI instance begins the launch process by initializing.

Step 13 To verify that the new instance is initializing, click on **Services** > **EC2** > **Instances**.

The new instance is visible in the display, and the Status Check should show the status “Initializing”. Proceed to the sections: Associating the Public IP Address with Cisco CSR 1000v Instance, on page 12 and Connecting to the CSR 1000v Instance using SSH, on page 12.

Launching the Cisco CSR 1000v AMI Using the Manual Launch

(Perform the following steps if you are using an Amazon EC2 instance. If you are using a VPC instance, see the Launching the Cisco CSR 1000v AMI Using the 1-Click Launch, on page 7).

Procedure

Step 1 On the Launch with EC2 page, choose the Cisco CSR 1000v release version from the “Select a Version” drop-down list.

Step 2 Select the Region from the drop-down list.

The hourly usage charges for your region are shown under Pricing Details.

Step 3 Click the **Launch with EC2 Console** button for your region.

The window to select the instance type displays.

Select the General purpose tab for the supported instance types. Select the instance type.

Click the **Next: Configure Instance Details** button.

Step 4 Configure the instance details.

Select one of the following two options:

- Launch into EC2-Classic. If you select EC2-Classic, you cannot configure additional network interfaces

  OR

- Select the network from the network drop-down list. Select a VPC subnet, into which you want to deploy the CSR 1000v, from the drop-down menu. Keep in mind that this determines the availability zone of your instance.

  You can initially create two interfaces on the Instance Details screen. Afterwards, to add more interfaces, click on **Network Interfaces**. The maximum number of interfaces that are supported depends on the instance type. For more information, see the table in Bootstrap Properties, on page 11.

- Select the availability zone from the drop-down menu.

- Select additional options available from AWS.

- (Optional) Configure the bootstrap properties by specifying the bootstrap options in the “User Data” box. The bootstrap options are described in the bootstrap properties table. Each option uses the syntax **<keyword>= “<string>**”. See Bootstrap Properties, on page 11.

Step 5 Click the **Next: Add Storage** button.

Step 6 Keep the default hard drive setting.

Note When operating the Cisco CSR 1000V in AWS, the (8 GB) size of virtual hard drives cannot be changed.

Click the **Next: Tag Instance** button.

Step 7 (Optional) Enter the tag information as needed.

Click the **Next: Configure Security Groups** button.

Step 8 (Optional) Choose one of the following:

- Create a new Security Group
- Select an existing Security Group

The Cisco CSR 1000v requires SSH for console access. The Cisco CSR 1000v also requires that the Security Group, at a minimum, does not block TCP/22. These settings are used to manage the Cisco CSR 1000V.

Click the **Review and Launch** button.

Step 9 Review the Cisco CSR 1000v instance information.

Click **Launch**.

Step 10 When prompted, enter the key pair information. The key pair consists of a public key stored in AWS and your private key used to authenticate access to the instance. Do one of the following:

a) Choose an existing key pair, or

b) Create a new key by performing the following steps:

  - Upload your own public key
  - Create a new key pair on AWS:

    Click on **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. A newly created private key can only be accessed once. After the key pair is downloaded, 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 enter the following command: **chmod 400** *pem-file-name*

Step 11 Click **Launch Instance**.

It takes approximately ten minutes to deploy the AMI instance. You can view the status by clicking on the Instances link on the menu.

Wait for the State to show **Running** and the Status Checks to show **passed**.

At this point, the Cisco CSR 1000v AWS instance is booted and ready for software configuration. Proceed to the sections: Associating the Public IP Address with Cisco CSR 1000v Instance, on page 12 and Connecting to the CSR 1000v Instance using SSH, on page 12.

Bootstrap Properties

Property	Description
hostname	Configures the hostname of the router. **Example** `hostname="csr-aws-instance"`
domain-name	Configures the network domain name. **Example** `domain-name="cisco.com"`
mgmt-vlan	Configures the dot1Q VLAN interface. Requires the management interface to be configured using the GigabitEthernetx.xxx format.
mgmt-ipv4-gateway	Configures the IPv4 management default gateway address. **Example** mgmt-ipv4-gateway="**`dhcp`**"

ios-config	Enables execution of a Cisco IOS command. To execute multiple commands, use multiple instances of ios-config, with a number appended to each instance—for example, ios-config-1, ios-config-2.
	When you specify a Cisco IOS command, use escape characters to pass special characters that are within the command: ampersand(&), double quotes(“), single quotes('), less than(<) or greater than(>). See "ios-config-5" in the example below.
	Examples
	`ios-config-1="username cisco priv 15 pass ciscoxyz"`
	`ios-config-2="ip scp server enable"`
	`ios-config-3="ip domain lookup"`
	`ios-config-4="ip domain name cisco.com”`
	`ios-config-5="event syslog pattern &quot;\(Tunnel1\) is down:`
	` BFD peer down notified&quot;"`
	In the above example, the entry for "ios-config-5" shows how to pass the IOS command:**event syslog pattern** “(Tunnel1) is down: BFD peer down notified”

license	(Cisco IOS XE 3.14.01S and later) Configures the license technology level as one of the following: • ax • ipbase • security • appx **Example** `license="security"`

Property	Description
Resource template	(Cisco IOS XE 3.16.3S and later) Configures the Resource Template. Possible values: default, service_plane_medium, service_plane_heavy **Example** `resource-template=”service_plane_medium”`

Associating the Public IP Address with Cisco CSR 1000v Instance

Before you can access the management console using an SSH connection, you must associate an interface on the Cisco CSR 1000v with the Public IP address created with the VPC. Perform the following steps:

Procedure

Step 1 On the Services > EC2 > Instances page, select the Cisco CSR 1000v instance.

Step 2 In the displayed Network interfaces, click on “eth0”.

Step 3 A popup window displays showing detailed information about the “eth0” interface.

Note the interface’s private IP address.

Step 4 Click **Interface ID value**.

Step 5 From the address drop-down menu, select the public IP address that you want the VM to use,

Step 6 Click **Allow reassociation** if you are reassigning a public IP address that is currently in use and mapped to another elastic network interface (ENI).

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. See the Connecting to the CSR 1000v Instance using SSH, on page 12.

Connecting to the CSR 1000v Instance using SSH

The Cisco CSR 1000v instance on AWS requires SSH for console access. To access the Cisco CSR 1000v AMI, perform the following steps:

Procedure

Step 1 Once the Cisco CSR 1000v status shows that is it is running, select the instance.

Step 2 Enter the following UNIX shell command to connect to the Cisco CSR 1000v console using SSH:

ssh -i *pem-file-name* **ec2-user**@[*public-ipaddress* | *DNS-name*]

Note You must log in as `ec2-user` the first time you access the instance.

The private key stored in the .pem file is used to authenticate access to the Cisco CSR 1000v instance.

Step 3 Start configuring the Cisco CSR 1000v. For information on downloading and activating the license for the BYOL AMI, see Downloading and Installing the License (BYOL AMI Only), on page 14.

Creating an AMI with Encrypted Elastic Block Storage

To create a Cisco CSR 1000v AMI with encrypted Elastic Block Storage(EBS), perform the following steps.

Before you begin

Create a Cisco CSR 1000v instance in AWS. For example, see Launching the Cisco CSR 1000v AMI Using the 1-Click Launch, on page 7.

Note When you create a Cisco CSR 1000v instance, use one of the sizes shown in the following list:

- t2.medium
- c4.large
- c4.xlarge
- c4.2xlarge
- c4.4xlarge
- c4.8xlarge

Procedure

Step 1 View the list of instances in **Services** > **EC2** > **Instances**.

Step 2 Select the name of an instance that you will use as the basis of a new AMI using encrypted EBS. For example, "CSR-1". Ensure that the instance state is "stopped".

Step 3 Take a snapshot of this instance by following steps **a** to **f** below.

a) Click on the Root device (for example, "**/dev/xvda/**").

The "Block Device" dialog box appears.

b) Click the EBS ID (for example **vol-08350aa2**).

The volume for this snapshot is displayed under **ELASTIC BLOCK STORE** > **Volumes**

c) Click **Actions** > **Create Snapshot**.

The Create Snapshot dialog box appears.

d) Click **Create**.

The "Create Image from EBS" pane appears.

e) Enter a name for the snapshot (for example, "unencrypted-CSR-1").

f) Select **Virtualization type** of "Hardware-assisted virtualization".

The message "Snapshot Creation Started" is displayed in the **Create Snapshot** dialog box. The snapshot is created after several minutes.

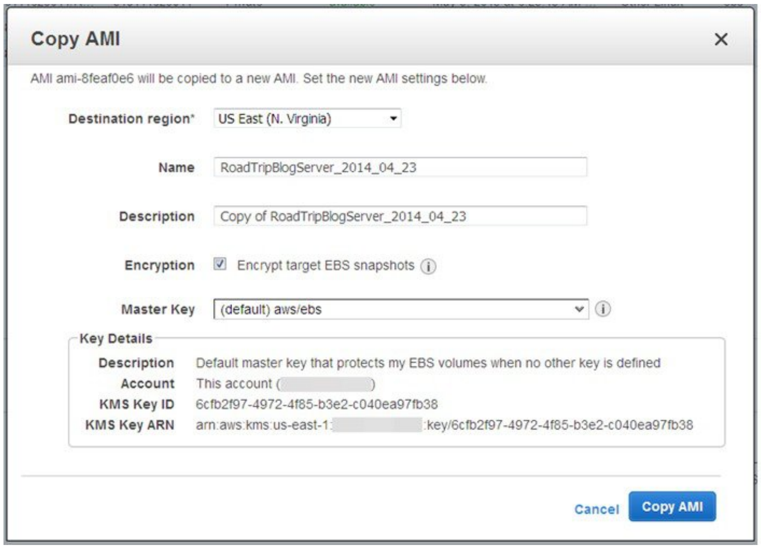
Under **ELASTIC BLOCK STORE** > **Snapshots**, the new snapshot is listed, with a status of "completed".

Step 4 Start creating a private AMI by going to **EC2** > **IMAGES** > **AMIs**.

The name of the snapshot instance that you created earlier (for example, "unencrypted-CSR-1") appears in the list of AMIs.

Step 5 Select the snapshot instance (for example, "unencrypted-CSR-1") and click **Actions** > **Copy AMI**.

The **Copy AMI** dialog box appears with input fields for Destination region, Name, Description, Encryption, Master Key and key details.



AMI ami-8feaf0e6 will be copied to a new AMI. Set the new AMI settings below.

Destination region* US East (N. Virginia)

Name RoadTripBlogServer_2014_04_23

Description Copy of RoadTripBlogServer_2014_04_23

Encryption ☑ Encrypt target EBS snapshots

Master Key (default) aws/ebs

Key Details

Description	Default master key that protects my EBS volumes when no other key is defined
Account	This account ([redacted])
KMS Key ID	6cfb2f97-4972-4f85-b3e2-c040ea97fb38
KMS Key ARN	arn:aws:kms:us-east-1:[redacted]:key/6cfb2f97-4972-4f85-b3e2-c040ea97fb38

366321

Step 6 Select a **Destination region** (for example, "US East") and enter a **Name** (for example, "encrypted-CSR-1").

Step 7 Enter a **Description**.

Step 8 For **Encryption**, check the **Encrypt target EBS snapshots** checkbox.

Step 9 For **Master Key**, you can select the default value; for example, "default(aws/ebs)".

Step 10 Click **Copy AMI**.

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

Step 11 Go to **EC2** > **IMAGES** > **AMIs** where the new AMI is listed; for example, "encrypted-CSR-1".

Downloading and Installing the License (BYOL AMI Only)

The Cisco CSR 1000v first boots with limited feature support and throughput. To achieve full feature support for your license, you must install and activate the licenses. You must obtain the PAK from the Cisco Software Licensing portal and then convert it into a license. The Cisco Software Licensing portal is available at: http: www.cisco.com go license

See the “Cisco Software Licensing (CSL)” chapter of the Cisco CSR 1000v Series Cloud Services Router Software Configuration Guide for information on installing licenses.

CHAPTER 3

Enabling the Guest Shell

Enabling the Guest Shell

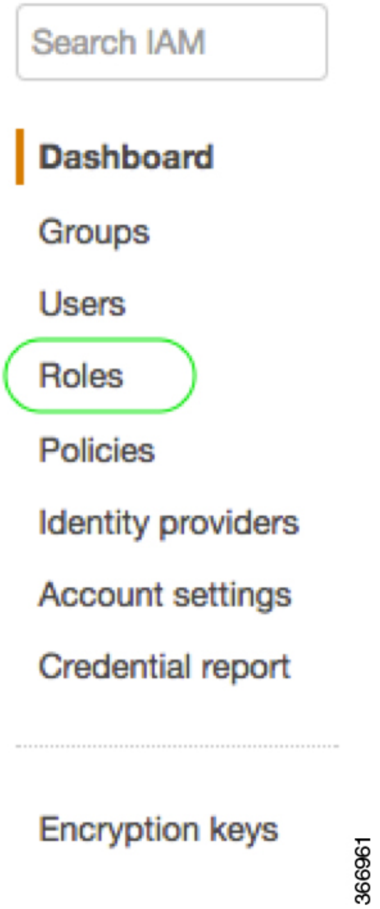
To enable the guest shell on the Cisco CSR 1000v, running on AWS, first create an IAM instance role and establish trust with an EC2 service. Then you have a choice of either assigning the IAM instance role to a preexisting Cisco CSR 1000v instance see Assign an IAM Instance Role to a Cisco CSR 1000v Instance, on page 19 below or assigning the IAM instance role to a new Cisco CSR 1000v instance, see Assign an IAM Instance Role to a New Cisco CSR 1000v Instance, on page 20.

Then perform further configuration steps on the Cisco CSR 1000v and enter the guest shell.

Create an IAM Instance Role

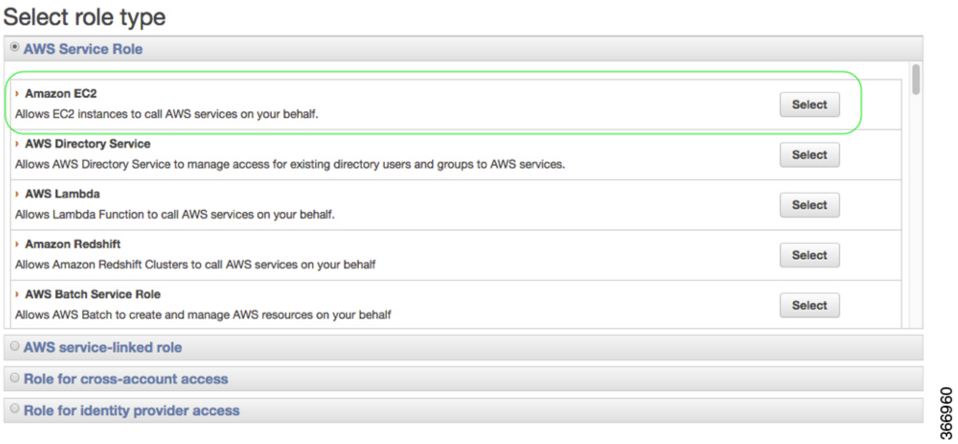
1. Sign into AWS, as an administrator with permissions to create an IAM Role
2. Click **EC2** to enter the EC2 console.
3. Click **IAM** to enter the IAM console.
4. Click **Roles**.

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



Select role type

◉ AWS Service Role

- **Amazon EC2** — Allows EC2 instances to call AWS services on your behalf.
- **AWS Directory Service** — Allows AWS Directory Service to manage access for existing directory users and groups to AWS services.
- **AWS Lambda** — Allows Lambda Function to call AWS services on your behalf.
- **Amazon Redshift** — Allows Amazon Redshift Clusters to call AWS services on your behalf
- **AWS Batch Service Role** — Allows AWS Batch to create and manage AWS resources on your behalf

○ AWS service-linked role

○ Role for cross-account access

○ Role for identity provider access

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

   This establishes trust with an EC2 service.

10. Under "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 these 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 CSR 1000v 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/password that eliminates the need to save account information on the Cisco CSR 1000v.

Procedure

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

Step 2 Select one of your listed CSR 1000v 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 the Cisco CSR 1000v and relaunch the Cisco CSR 1000v.

```
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 Cisco CSR 1000v Instance

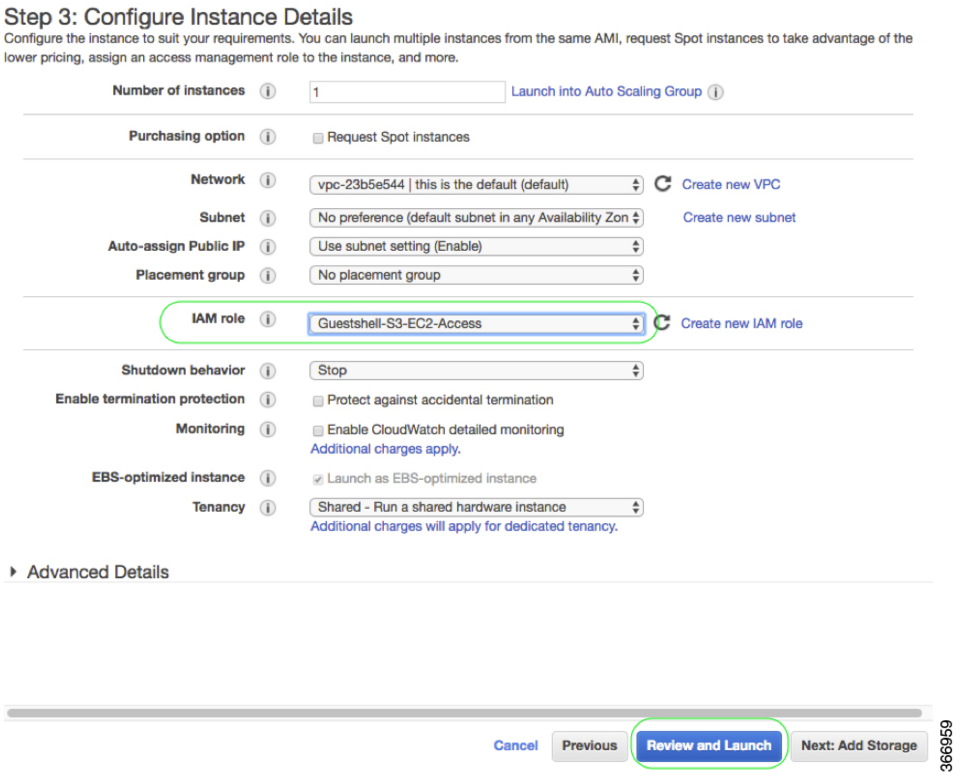
The following procedure shows how to assign an IAM Instance Role to a Cisco CSR 1000v, during the creation of a new Cisco CSR 1000v instance.

Procedure

Step 1 Launch a new CSR 1000v as an EC2 instance, and choose an instance type.

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

Figure 3: Configure Instance Details



Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the lower pricing, assign an access management role to the instance, and more.

Number of instances	1	Launch into Auto Scaling Group
Purchasing option	□ Request Spot instances	
Network	vpc-23b5e544 | this is the default (default)	Create new VPC
Subnet	No preference (default subnet in any Availability Zon	Create new subnet
Auto-assign Public IP	Use subnet setting (Enable)	
Placement group	No placement group	
IAM role	Guestshell-S3-EC2-Access	Create new IAM role
Shutdown behavior	Stop	
Enable termination protection	□ Protect against accidental termination	
Monitoring	□ Enable CloudWatch detailed monitoring Additional charges apply.	
EBS-optimized instance	☑ Launch as EBS-optimized instance	
Tenancy	Shared - Run a shared hardware instance Additional charges will apply for dedicated tenancy.	

▸ Advanced 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 CSR 1000v and relaunch the Cisco CSR 1000v.

```
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
```

Guest Shell Examples

The following examples show how to download packages in the Guest Shell on a Cisco CSR 1000v instance, and a few other useful guest shell commands.

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

```
Collecting csr_aws_guestshell
/usr/lib/python2.7/site-packages/pip-8.1.2-py2.7.egg/pip/_vendor/requests/packages/urllib3/util/ssl_.py:318:
 SNIMissingWarning: An HTTPS request has been made, but the SNI (Subject Name Indication)
 extension to TLS is not available on this platform. This may cause the server to present
 an incorrect TLS certificate, which can cause validation failures. You can upgrade to
a newer version of Python to solve this. For more information, see
https://urllib3.readthedocs.org/en/latest/security.html#snimissingwarning.
/usr/lib/python2.7/site-packages/pip-8.1.2-py2.7.egg/pip/_vendor/requests/packages/urllib3/util/ssl_.py:122:
 InsecurePlatformWarning: A true SSLContext object is not available. This prevents urllib3
 from configuring SSL appropriately and may cause certain SSL connections to fail. You
can upgrade to a newer version of Python to solve this. For more information, see
https://urllib3.readthedocs.org/en/latest/security.html#insecureplatformwarning.
  Downloading csr_aws_guestshell-0.0.7.dev.tar.gz
Collecting awscli (from csr_aws_guestshell)
  Downloading awscli-1.11.145-py2.py3-none-any.whl (1.2MB)
    100% |################################| 1.2MB 1.1MB/s
Collecting boto (from csr_aws_guestshell)
  Downloading boto-2.48.0-py2.py3-none-any.whl (1.4MB)
    100% |################################| 1.4MB 914kB/s
Collecting boto3 (from csr_aws_guestshell)
  Downloading boto3-1.4.7-py2.py3-none-any.whl (128kB)
    100% |################################| 133kB 8.5MB/s
Collecting botocore==1.7.3 (from awscli->csr_aws_guestshell)
  Downloading botocore-1.7.3-py2.py3-none-any.whl (3.6MB)
```

```
    100% |████████████████████████████████| 3.6MB 337kB/s
Collecting rsa<=3.5.0,>=3.1.2 (from awscli->csr_aws_guestshell)
  Downloading rsa-3.4.2-py2.py3-none-any.whl (46kB)
    100% |################################| 51kB 11.2MB/s
Collecting s3transfer<0.2.0,>=0.1.9 (from awscli->csr_aws_guestshell)
  Downloading s3transfer-0.1.11-py2.py3-none-any.whl (54kB)
    100% |################################| 61kB 11.5MB/s
Collecting docutils>=0.10 (from awscli->csr_aws_guestshell)
  Downloading docutils-0.14-py2-none-any.whl (543kB)
    100% |################################| 552kB 2.3MB/s
Collecting colorama<=0.3.7,>=0.2.5 (from awscli->csr_aws_guestshell)
  Downloading colorama-0.3.7-py2.py3-none-any.whl
Collecting PyYAML<=3.12,>=3.10 (from awscli->csr_aws_guestshell)
  Downloading PyYAML-3.12.tar.gz (253kB)
    100% |################################| 256kB 4.7MB/s
Collecting jmespath<1.0.0,>=0.7.1 (from boto3->csr_aws_guestshell)
  Downloading jmespath-0.9.3-py2.py3-none-any.whl
Collecting python-dateutil<3.0.0,>=2.1 (from botocore==1.7.3->awscli->csr_aws_guestshell)

  Downloading python_dateutil-2.6.1-py2.py3-none-any.whl (194kB)
    100% |################################| 194kB 5.7MB/s
Collecting pyasn1>=0.1.3 (from rsa<=3.5.0,>=3.1.2->awscli->csr_aws_guestshell)
  Downloading pyasn1-0.3.3-py2.py3-none-any.whl (63kB)
    100% |################################| 71kB 10.7MB/s
Collecting futures<4.0.0,>=2.2.0; python_version == "2.6" or python_version == "2.7"
(from s3transfer<0.2.0,>=0.1.9->awscli->csr_aws_guestshell)
  Downloading futures-3.1.1-py2-none-any.whl
Collecting six>=1.5 (from
python-dateutil<3.0.0,>=2.1->botocore==1.7.3->awscli->csr_aws_guestshell)
  Downloading six-1.10.0-py2.py3-none-any.whl
Installing collected packages: six, python-dateutil, jmespath, docutils, botocore, pyasn1,
 rsa, futures, s3transfer, colorama, PyYAML, 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-3.12 awscli-1.11.145 boto-2.48.0 boto3-1.4.7 botocore-1.7.3
 colorama-0.3.7 csr-aws-guestshell-0.0.7.dev0 docutils-0.14 futures-3.1.1 jmespath-0.9.3
 pyasn1-0.3.3 python-dateutil-2.6.1 rsa-3.4.2 s3transfer-0.1.11 six-1.10.0
/usr/lib/python2.7/site-packages/pip-8.1.2-py2.7.egg/pip/_vendor/requests/packages/urllib3/util/ssl_.py:122:
 InsecurePlatformWarning: A true SSLContext object is not available. This prevents urllib3
 from configuring SSL appropriately and may cause certain SSL connections to fail. You
can upgrade to a newer version of Python to solve this. For more information, see
https://urllib3.readthedocs.org/en/latest/security.html#insecureplatformwarning.
You are using pip version 8.1.2, however version 9.0.1 is available.
You should consider upgrading via the 'pip install --upgrade pip' command.
[guestshell@guestshell ~]$
```

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

```
[guestshell@guestshell ~]$ aws s3 ls csr1kv
2017-08-09 02:55:27  446866343 ultra_166.bin
[guestshell@guestshell ~]$
```

3. You can download a Cisco CSR 1000v AWS package containing sample scripts, using the `sudo pip install csr_aws_guestshell` command. For further information on this package, see https://github.com/CiscoDevNet/csr_aws_guestshell. Example:

```
[guestshell@guestshell ~]$ sudo pip install csr_aws_guestshell
Collecting csr_aws_guestshell
/usr/lib/python2.7/site-packages/pip-8.1.2-py2.7.egg/pip/_vendor/requests/packages/urllib3/util/ssl_.py:318:
 SNIMissingWarning: An HTTPS request has been made, but the SNI (Subject Name Indication)
 extension to TLS is not available on this platform. This may cause the server to present
 an incorrect TLS certificate, which can cause validation failures. You can upgrade to
a newer version of Python to solve this. For more information, see
https://urllib3.readthedocs.org/en/latest/security.html#snimissingwarning.
```

```
/usr/lib/python2.7/site-packages/pip-8.1.2-py2.7.egg/pip/_vendor/requests/packages/urllib3/util/ssl_.py:122:
 InsecurePlatformWarning: A true SSLContext object is not available. This prevents urllib3
 from configuring SSL appropriately and may cause certain SSL connections to fail. You
can upgrade to a newer version of Python to solve this. For more information, see
https://urllib3.readthedocs.org/en/latest/security.html#insecureplatformwarning.
  Downloading csr_aws_guestshell-0.0.7.dev.tar.gz
Collecting awscli (from csr_aws_guestshell)
  Downloading awscli-1.11.145-py2.py3-none-any.whl (1.2MB)
    100% |################################| 1.2MB 1.1MB/s
Collecting boto (from csr_aws_guestshell)
  Downloading boto-2.48.0-py2.py3-none-any.whl (1.4MB)
    100% |################################| 1.4MB 914kB/s
Collecting boto3 (from csr_aws_guestshell)
  Downloading boto3-1.4.7-py2.py3-none-any.whl (128kB)
    100% |################################| 133kB 8.5MB/s
Collecting botocore==1.7.3 (from awscli->csr_aws_guestshell)
  Downloading botocore-1.7.3-py2.py3-none-any.whl (3.6MB)
    100% |################################| 3.6MB 337kB/s
Collecting rsa<=3.5.0,>=3.1.2 (from awscli->csr_aws_guestshell)
  Downloading rsa-3.4.2-py2.py3-none-any.whl (46kB)
    100% |################################| 51kB 11.2MB/s
Collecting s3transfer<0.2.0,>=0.1.9 (from awscli->csr_aws_guestshell)
  Downloading s3transfer-0.1.11-py2.py3-none-any.whl (54kB)
    100% |################################| 61kB 11.5MB/s
Collecting docutils>=0.10 (from awscli->csr_aws_guestshell)
  Downloading docutils-0.14-py2-none-any.whl (543kB)
    100% |################################| 552kB 2.3MB/s
Collecting colorama<=0.3.7,>=0.2.5 (from awscli->csr_aws_guestshell)
  Downloading colorama-0.3.7-py2.py3-none-any.whl
Collecting PyYAML<=3.12,>=3.10 (from awscli->csr_aws_guestshell)
  Downloading PyYAML-3.12.tar.gz (253kB)
    100% |################################| 256kB 4.7MB/s
Collecting jmespath<1.0.0,>=0.7.1 (from boto3->csr_aws_guestshell)
  Downloading jmespath-0.9.3-py2.py3-none-any.whl
Collecting python-dateutil<3.0.0,>=2.1 (from botocore==1.7.3->awscli->csr_aws_guestshell)

  Downloading python_dateutil-2.6.1-py2.py3-none-any.whl (194kB)
    100% |################################| 194kB 5.7MB/s
Collecting pyasn1>=0.1.3 (from rsa<=3.5.0,>=3.1.2->awscli->csr_aws_guestshell)
  Downloading pyasn1-0.3.3-py2.py3-none-any.whl (63kB)
    100% |################################| 71kB 10.7MB/s
Collecting futures<4.0.0,>=2.2.0; python_version == "2.6" or python_version == "2.7"
(from s3transfer<0.2.0,>=0.1.9->awscli->csr_aws_guestshell)
  Downloading futures-3.1.1-py2-none-any.whl
Collecting six>=1.5 (from
python-dateutil<3.0.0,>=2.1->botocore==1.7.3->awscli->csr_aws_guestshell)
  Downloading six-1.10.0-py2.py3-none-any.whl
Installing collected packages: six, python-dateutil, jmespath, docutils, botocore, pyasn1,
 rsa, futures, s3transfer, colorama, PyYAML, 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-3.12 awscli-1.11.145 boto-2.48.0 boto3-1.4.7 botocore-1.7.3
 colorama-0.3.7 csr-aws-guestshell-0.0.7.dev0 docutils-0.14 futures-3.1.1 jmespath-0.9.3
 pyasn1-0.3.3 python-dateutil-2.6.1 rsa-3.4.2 s3transfer-0.1.11 six-1.10.0
/usr/lib/python2.7/site-packages/pip-8.1.2-py2.7.egg/pip/_vendor/requests/packages/urllib3/util/ssl_.py:122:
 InsecurePlatformWarning: A true SSLContext object is not available. This prevents urllib3
 from configuring SSL appropriately and may cause certain SSL connections to fail. You
can upgrade to a newer version of Python to solve this. For more information, see
https://urllib3.readthedocs.org/en/latest/security.html#insecureplatformwarning.
You are using pip version 8.1.2, however version 9.0.1 is available.
You should consider upgrading via the 'pip install --upgrade pip' command.
[guestshell@guestshell ~]$
```

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 Cisco IOS XE CLI commands to an S3 bucket

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

load-bin-from-s3.py —downloads a .bin file for the Cisco CSR 1000v and reloads

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

capture-interface.py —sets Cisco IOS XE CLI commands to monitor and capture packets for a period of time, then upload 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 CSR 1000v image:

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

CHAPTER 4

Upgrading a Cisco IOS XE Image on Amazon Web Services

The procedure in this chapter shows how to upgrade a Cisco IOS XE 16.7.1 image or later running CSR 1000V.

Consider the following before you upgrade the IOS XE image:

- To upgrade an IOS XE image for a Cisco CSR 1000V running in AWS, the current version of IOS XE running on the Cisco CSR 1000V instance must be Cisco IOS XE Fuji 16.7.1 or later.
- For Cisco IOS XE Everest 16.6 and earlier on AWS, you cannot use the Cisco CSR 1000v .bin file to upgrade a Cisco CSR1000V instance. You must re-deploy a new instance from theAWS Marketplace and migrate your configuration and licenses.
- You cannot downgrade a Cisco CSR 1000V image on AWS to Cisco IOS XE Everest 16.6.2 or earlier. For example, if you are running Cisco IOS XE Fuji 16.7.1 or later, you must not downgrade to Cisco IOS XE Everest 16.6.2 or earlier.
- To upgrade or downgrade a Cisco CSR 1000V image running on AWS, you must expand the `.bin` file and use the `packages.conf` to upgrade to the new version.
- The only currently available downgrade for a Cisco IOS XE Gibralter 16.10.1 image is to Cisco IOS XE Fuji 16.9.2. You cannot downgrade a Cisco CSR 1000V image running on AWS from Cisco IOS XE Gibralter 16.10 to Cisco IOS XE Fuji 16.9.1 or earlier.

Note To check the version of your Cisco IOS XE image for your Cisco CSR 1000V instance, run the `show version` command.

```
Router# show version
Cisco IOS XE Software, Version 2017-11-08_14.44_user4
Cisco IOS Software [Fuji], Virtual XE Software (X86_64_LINUX_IOSD-UNIVERSALK9-M), Experimental
 Version 16.8.2017110
```

How to Upgrade a Cisco IOS XE Image on Amazon Web Services

Procedure

Step 1 **scp** *-i <key.pem file> upgrade-imageecs-user@csr-public-ip-address*: *copied-upgrade-image*

Copy the new image to the CSR 1000V boot flash memory. You can choose any name for the copy of the image in bootflash; for example, `upgrade.bin`.

Step 2 **request platform software package expand file bootflash:upgrade.bin to bootflash:upgrade/**

Expand the image that is in boot flash memory.

Note If you have already performed an upgrade on this CSR 1000V instance before, use a different directory name when you perform the upgrade the second time. For example, if you used the `request platform software package expand file bootflash:upgrade.bin to bootflash:upgrade/` command when you performed the upgrade the first time, this command expands the bin file in the 'upgrade' directory of the blootflash and places the packages.conf file in the upgrade directory.

When you perform an upgrade anytime after, ensure that you use a different directory name. For example, your upgrade command could read `request platform software package expand file bootflash:upgrade2.bin to bootflash:upgrade2/`. Note that the directory name is `upgrade2` in this instance.

Example:

```
Router# request platform software package expand file bootflash:upgrade.bin to
bootflash:upgrade/
Nov  8 03:25:34.412 %INSTALL-5-OPERATION_START_INFO: R0/0: packtool: Started expand package
 bootflash:upgrade.bin
Verifying parameters
Expanding superpackage bootflash:upgrade.bin
Validating package type
Copying package files
SUCCESS: Finished expanding all-in-one software package.
```

Step 3 **configure terminal**

Enters the global configuration mode.

Step 4 **boot system bootflash:upgrade/packages.conf**

Adds a boot system entry to the `packages.conf` file that was generated in step 2. For example, the boot system entry to the `/bootflash/upgrade/packages.conf` file as shown in the example.

Example:

The following example shows how to correctly enter a boot system entry:

```
Router(config)# boot system bootflash:upgrade/packages.conf
```

Note Do **not** add the boot system entry like this: `boot system bootflash:upgrade.bin`. This command tells the Cisco CSR 1000V to boot from `upgrade.bin`. However, the CSR 1000V boot fails if the file size of the `upgrade.bin` is greater than the low memory size that is allowed by GRUB in AWS.

Note Ensure that you point the boot system command entry to the `packages.conf` file expanded in the upgrade directory as mentioned in step 2. You must use the same directory name that you have specified in step 2.

Step 5 **end**

Exits the global configuration mode and returns to the privileged EXEC mode.

Example:

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

Step 6 **show run | sec boot**

Verifies the boot system entry.

Example:

```
Router# show run | sec boot
boot-start-marker
boot system bootflash:upgrade/packages.conf
boot-end-marker
diagnostic bootup level minimal
```

Step 7 **copy running-configuration startup-configuration**

Saves the configuration.

Example:

```
Router# copy running-configuration startup-configuration
Building configuration
...

[OK]
```

Step 8 Reload the router.

CHAPTER 5

Configuring High Availability

This section contains the following topics:

Information about High Availability

A method for deploying two Cisco CSR 1000v in a redundant pair with failover between them, is summarized below. Also see How to Configure High Availability, on page 36 for further details.

Configuring High Availability: Summary

1. Create an AWS Identity and Access Management (IAM) role to be able to access the AWS APIs.
2. Create an AWS VPC and launch each Cisco CSR1000V into the VPC with an Amazon EC2 IAM role.
3. (Cisco IOS XE Everest 16.6.1 or later) Enable the AX or SEC license, using BFD.

   (Cisco IOS XE Everest 16.5.1 or earlier) Enable the AX license, using BFD.
4. Configure the CSRs to reach the internet and access EC2 AWS API servers.
5. Configure a GRE tunnel between the CSR1000V instances and enable Bi-directional Forwarding Detection (BFD) and a routing protocol (EIGRP or BGP) on the GRE tunnel between the routers, for peer failure detection.
6. Note the route table ID and network interface ID.
7. (Cisco IOS XE 3.16 or earlier) Monitor AWS HA errors such as BFD peer down events and specify the routing changes parameters: Route-table-id, Network-interface-id and CIDR range, by configuring each CSR1000V with an Embedded Event Manager (EEM) applet. When a BFD peer down event is detected, the applet uses the AWS EC2 VPC API to modify the VPC route table to redirect traffic around the failure.

Note For private subnets, do not use the IP address 10.0.3.0/24—this is used internally on the Cisco CSR1000V for High Availability. The Cisco CSR1000V needs to have public internet accessibility to make REST API calls that change the AWS route table.

8. (Cisco IOS XE Denali16.3.1a or later) Monitor AWS HA errors such as BFD peer down events and specify the routing changes to Route-table-id, Network-interface-id and CIDR by configuring each CSR1000V using the **cloud provider aws** command. When a BFD peer down event is detected, routing changes are made to the Route-table-id, Network-interface-id and CIDR range and the VPC route table is modified to redirect traffic around the failure.

Further Information

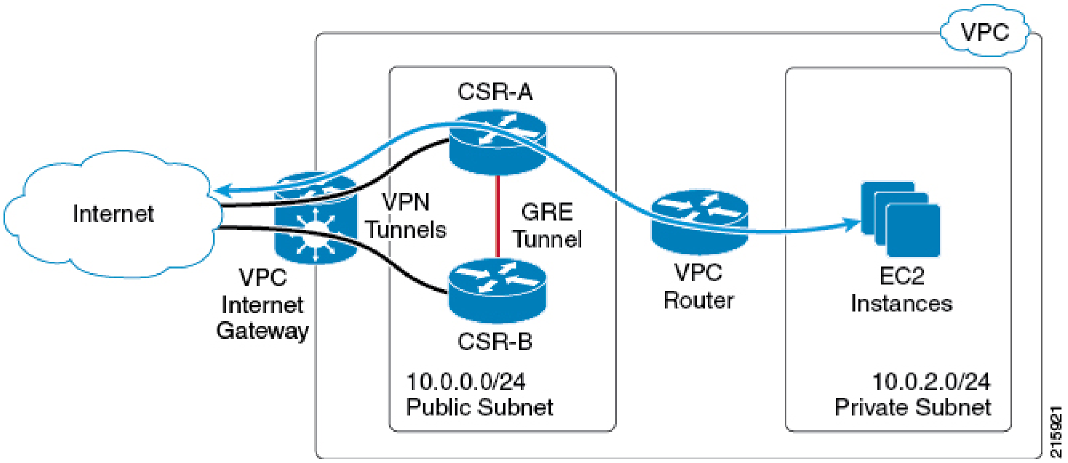
For futher information about configuring HA for CSR1000V on AWS, see CSR1000V HA Redundancy Deployment Guide on Amazon AWS.

Initial Topology

The initial topology and traffic flow are shown in the following figure, before performing the procedure: How to Configure High Availability, on page 36.

Note The procedure How to Configure High Availability, on page 36 shows how to configure high availability (VPC Gateway Redundancy) for a VPN gateway configuration in a single availability zone. For further examples, see the Deploying the Cisco Cloud Services Router 1000V Series in Amazon Web Services, Design and Implementation Guide .

Ingress and egress traffic is initially forwarded through CSR-A.



Each CSR has a primary Ethernet interface (GigabitEthernet1) that is assigned to the public subnet. The public subnet has a VPC route table with a default route target of the Internet gateway. Both CSR-A and CSR-B are launched in the same public subnet.

Each CSR also has a VPN tunnel to the Internet. These tunnels typically terminate at another VPN device located on the enterprise network or another VPC.

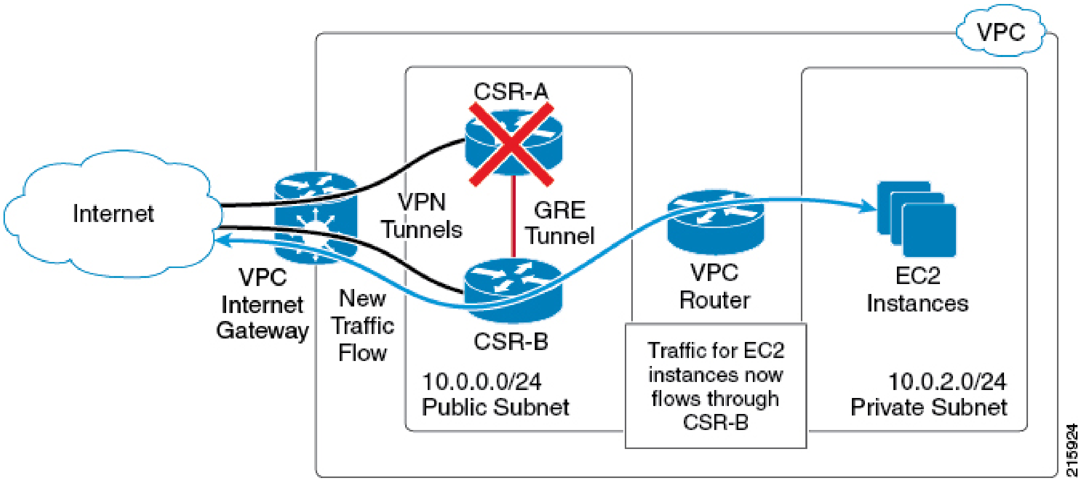
To support high availability , a GRE tunnel is configured between the local CSRs, which allows the CSRs to exchange BFD control packets that are used for peer failure detection.

The EC2 instances reside in a private subnet, Private Subnet (10.0.2.0/24), in the topology diagram. If the CSR is not directly connected to this private subnet, it is recommended to add a static route for the private subnet to each CSR. This static route points to the address of the VPC router on the public subnet. This address will always be the first usable address of a subnet. For example, the VPC router address for the subnet 10.0.0.0/24 will be 10.0.0.1.

EIGRP is used as the routing protocol, though other routing protocols could be used. The primary purpose of the routing protocol is to register as a BFD client. BFD requires at least one client protocol before it will initiate neighbor discovery. An additional benefit of the GRE tunnel and the routing protocol is that they can be used to establish a back-up path in case of VPN tunnel failures.

The EC2 private subnet has its own VPC route table. The default route for this subnet will have a target of the public subnet network interface (GigabitEthernet1) of one of the CSRs. Because the VPC route table only allows for one active target per route, only one CSR is in the egress traffic path for the subnet. Ingress traffic flow over the VPN tunnels is determined by the remote VPN devices. This means that CSR-B may be the active ingress path or that load sharing is performed between CSR-A and CSR-B.

The next figure shows the new traffic flow that occurs after you have configured the steps shown in the procedure How to Configure High Availability, on page 36 and after a BFD peer down event. The modified VPC route table causes traffic to egress through CSR-B.



Error Messages for Amazon Web Services High Availability

Errors that may occur during route replacement (for Cisco IOS XE Denali 16.3.1a or later) are shown in the following table.

Error Name	Message	Description
BFDEVENT	VXE BFD peer %i interface %s transitioned to down	The BFD interface transitioned to down triggering a VXE Cloud HA event.

Error Name	Message	Description
BFDCONFIG	VXE BFD peer %i configuration %s from %s	The BFD configuration was removed while cloud HA is still configured.
NOTCFGD	VXE Cloud HA BFD is enabled, but %s node %u not fully configured	The BFD state transitioned, but not all Cloud parameters were configured
FAILED	VXE Cloud HA BFD state transitioned, %s node %u event %s failed	The BFD state transitioned, but failed to perform route replacement
SUCCESS	VXE Cloud HA BFD state transitioned, %s node %u event %s successful %VXE_CLOUD_HA-6-SUCCESS: VXE Cloud HA BFD state transitioned, AWS node 1 event replace route successful	The BFD state transitioned, but failed to perform route replacement
INIT	VXE Cloud HA %s failed	VXE Cloud HA initialization failure

How to Configure High Availability

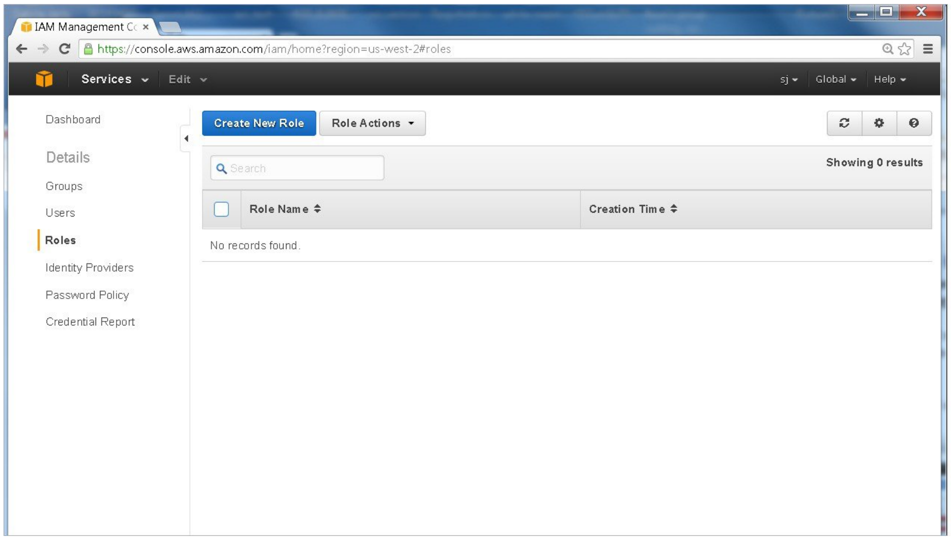
Creating an IAM Role

You can use the Cisco CSR 1000v as a proxy to run AWS API commands that modify the route table. The following procedure creates an IAM role and this role is used during the launch of a CSR 1000v instance. This provides the correct access credentials for the Cisco CSR 1000v to use and modify AWS APIs.

Procedure

Step 1 Browse to the IAM dashboard, and navigate to **Roles** > **Create Role**

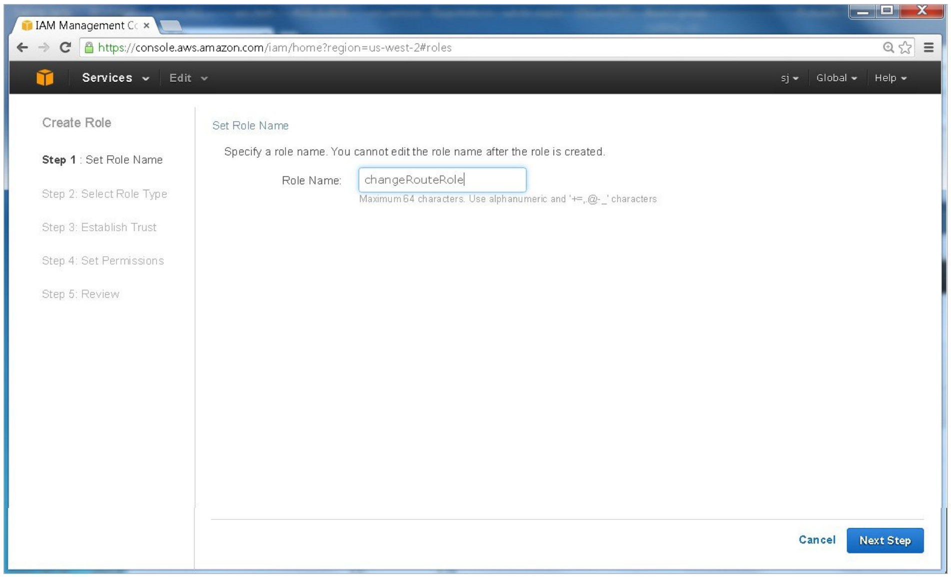
In this task you create an IAM role to access AWS APIs. Then you can lauch a Cisco CSR 1000v with the privileges of the IAM role.



No records found.

Step 2 Create a role with appropriate role name.

The example shows a role with the name ChangeRouteRole.

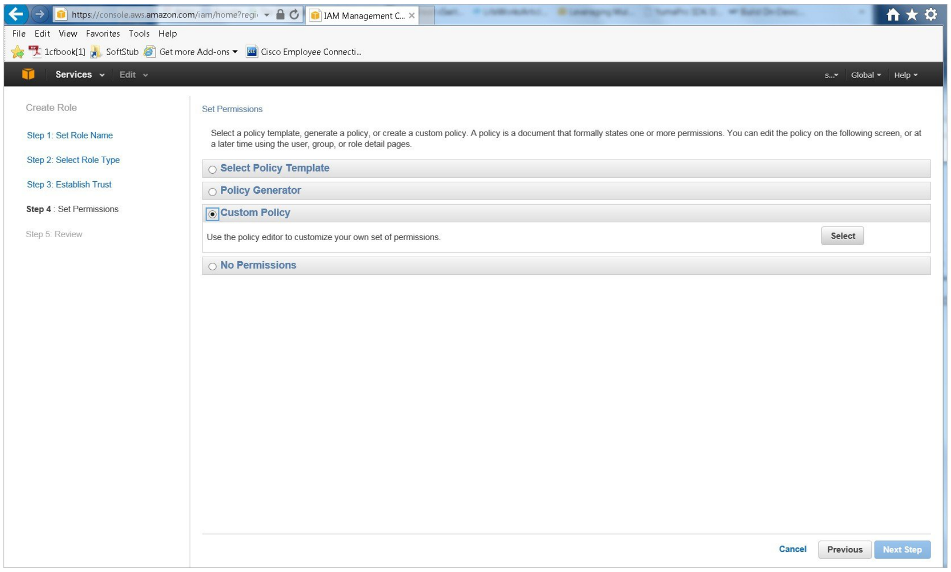


Specify a role name. You cannot edit the role name after the role is created.

Role Name: changeRouteRole

Maximum 64 characters. Use alphanumeric and '+=,.@-_' characters

Step 3 Select Amazon EC2 in the AWS Service Roles section.



Set Permissions

Select a policy template, generate a policy, or create a custom policy. A policy is a document that formally states one or more permissions. You can edit the policy on the following screen, or at a later time using the user, group, or role detail pages.

○ Select Policy Template

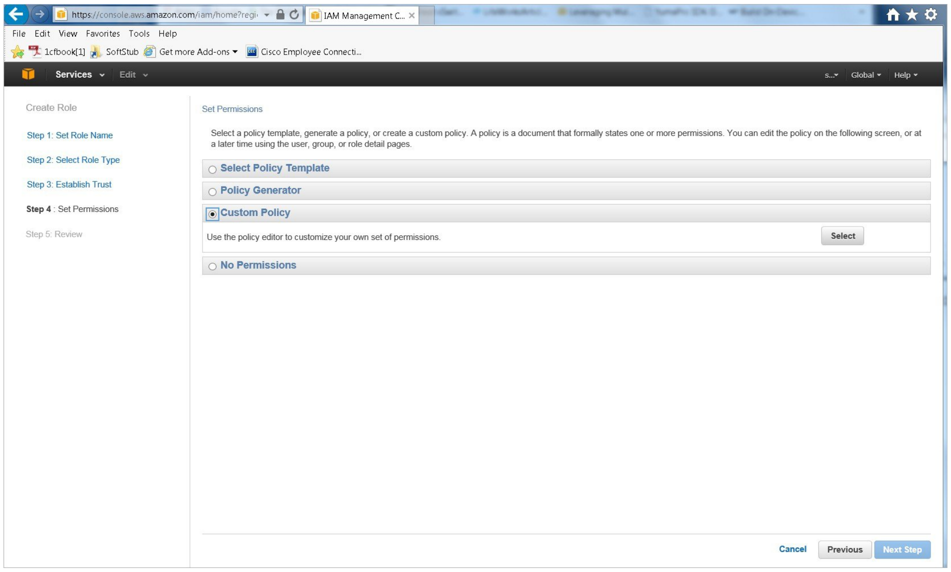
○ Policy Generator

● Custom Policy

Use the policy editor to customize your own set of permissions.

○ No Permissions

Step 4 Select Custom Policy.



Set Permissions

Select a policy template, generate a policy, or create a custom policy. A policy is a document that formally states one or more permissions. You can edit the policy on the following screen, or at a later time using the user, group, or role detail pages.

○ Select Policy Template

○ Policy Generator

● Custom Policy

Use the policy editor to customize your own set of permissions.

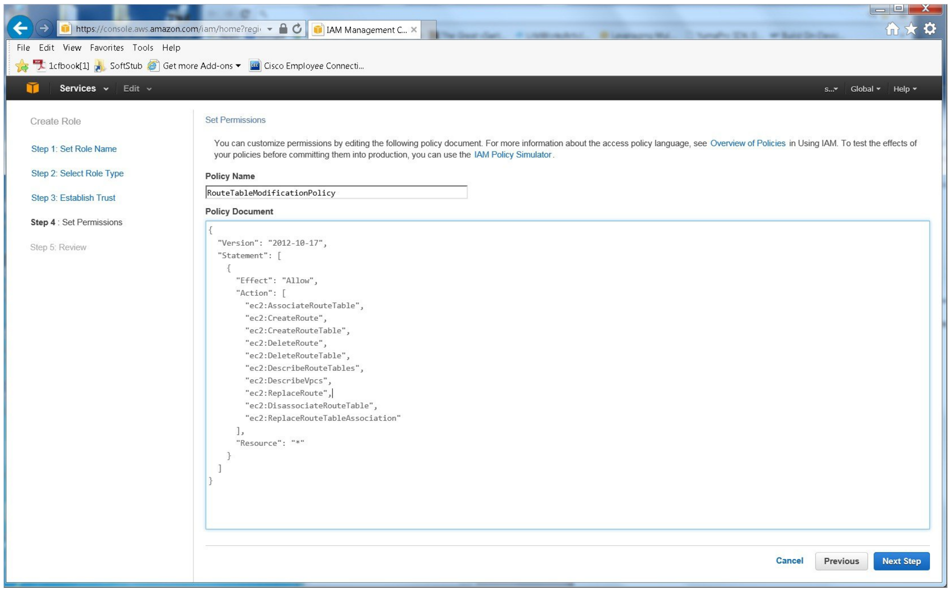
○ No Permissions

In this example, the following policy is used for this role.

Example:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:AssociateRouteTable",
        "ec2:CreateRoute",
        "ec2:CreateRouteTable",
        "ec2:DeleteRoute",
        "ec2:DeleteRouteTable",
        "ec2:DescribeRouteTables",
        "ec2:DescribeVpcs",
        "ec2:ReplaceRoute",
        "ec2:DisassociateRouteTable",
        "ec2:ReplaceRouteTableAssociation"
      ],
      "Resource": "*"
    }
  ]
}
```

Step 5 Click next and then create the role.



Set Permissions

You can customize permissions by editing the following policy document. For more information about the access policy language, see Overview of Policies in Using IAM. To test the effects of your policies before committing them into production, you can use the IAM Policy Simulator.

Policy Name

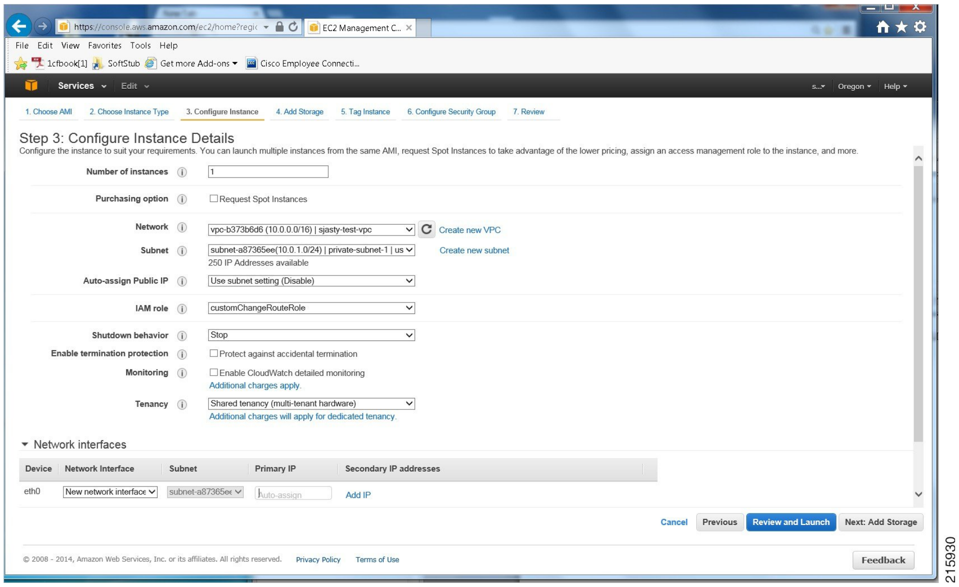
RouteTableModificationPolicy

Policy Document

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:AssociateRouteTable",
        "ec2:CreateRoute",
        "ec2:CreateRouteTable",
        "ec2:DeleteRoute",
        "ec2:DeleteRouteTable",
        "ec2:DescribeRouteTables",
        "ec2:DescribeVpcs",
        "ec2:ReplaceRoute",
        "ec2:DisassociateRouteTable",
        "ec2:ReplaceRouteTableAssociation"
      ],
      "Resource": "*"
    }
  ]
}
```

Step 6 Launch the Cisc CSR 1000v using the IAM role.

Specify the IAM role create in steps 1 to 5.



Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot Instances to take advantage of the lower pricing, assign an access management role to the instance, and more.

Number of instances: 1

Purchasing option: □ Request Spot Instances

Network: vpc-b373b6d6 (10.0.0.0/16) | sjasty-test-vpc — Create new VPC

Subnet: subnet-a87365ee(10.0.1.0/24) | private-subnet-1 | us — Create new subnet
250 IP Addresses available

Auto-assign Public IP: Use subnet setting (Disable)

IAM role: customChangeRouteRole

Shutdown behavior: Stop

Enable termination protection: □ Protect against accidental termination

Monitoring: □ Enable CloudWatch detailed monitoring — Additional charges apply.

Tenancy: Shared tenancy (multi-tenant hardware) — Additional charges will apply for dedicated tenancy.

Network interfaces

Device	Network Interface	Subnet	Primary IP	Secondary IP addresses
eth0	New network interface	subnet-a87365ee	Auto-assign	Add IP

© 2008 - 2014, Amazon Web Services, Inc. or its affiliates. All rights reserved.

How to Configure High Availability

To configure High Availability (VPC Gateway Redundancy) using two Cisco CSR 1000v's, perform the following steps:

Before you begin

Create an IAM Role, see Creating an IAM Role, on page 32.

Procedure

Step 1 Create and configure a VPC based on the topology requirements and launch two Cisco CSR 1000v's (each router has an IAM role) into the VPC. Then configure the routers, including the VPN tunnels.

In the previous section Creating an IAM Role, on page 32 section, you created an IAM role that you can use to access the AWS APIs using temporary security credentials. The Cisco CSR 1000v's can then be launched with the privileges of the IAM role. For more information on deployment steps and the Cisco CSR 1000v configuration, see the other sections in this document and the following white paper: Setting up DMVPN on the CSR in AWS Cloud.

Step 2 (Cisco IOS XE Everest 16.6.x or later) Enable the Security package or the AX package. For the security package, enter the **license boot level security** command. For the AX package, enter the **license boot level security ax** command. Save the configuration and reload. Use the `show license` command to inspect the license status.

Example:

In this example, the Security package is enabled.

```
CSR-A(config)# license boot level security
```

```
% use 'write' command to make license boot config take effect on next boot
CSR-A(config)# end
CSR-A# write memory
Building configuration...
[OK]
CSR-A# reload
```

Step 3 (Cisco IOS XE Everest 16.5.x or earlier) Enable the AX license, using BFD in this case, by entering the **license boot level security ax** command. Save the configuration and reload. Use the `show license` command to inspect the license status.

Example:

In this example, the AX package is enabled.

```
CSR-A(config)# license boot level security ax
% use 'write' command to make license boot config take effect on next boot
CSR-A(config)# end
CSR-A# write memory
Building configuration...
[OK]
CSR-A# reload
```

Step 4 Configure each Cisco CSR 1000v to reach the internet and access EC2 AWS API servers. The default route table on the public subnet of the CSR 1000v needs a route for the internet gateway. The default route table also needs to be able to reach the EC2 AWS API servers to modify the routes. The CSR 1000v interface on the public subnet (Gigabit Ethernet 1) must not be configured to block http traffic or contain access list rules that may block the access of EC2 AWS API servers.

Step 5 Configure each Cisco CSR 1000v to access a DNS server. See the IP Addressing: DNS Configuration Guide.

Step 6 Configure the GRE tunnel, using EIGRP. Configure the GRE tunnel using the Elastic IPs of the CSR 1000v's (recommended to avoid DHCP lease renewal issues detecting false failures.) The BFD values can be configured to be more aggressive than those shown in the following example, if faster convergence is required. However, this can lead to BFD peer down events during intermittent connectivity. The values in the following example will detect peer failure within 1.5 seconds, and this setup has been shown to be stable in an AWS VPC environment. There is a variable delay of about a few seconds between the time when the AWS API command is executed and when the VPC routing table changes go into effect. Example:

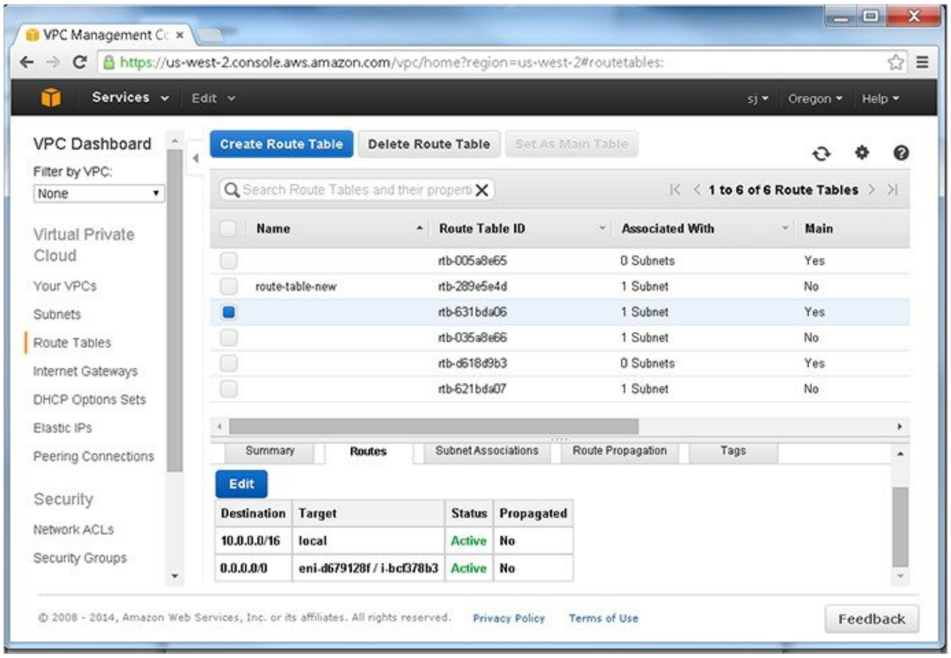
Example:

```
interface Tunnel1
ip address 172.17.1.1 255.255.255.252
bfd interval 500 min_rx 500 multiplier 3
tunnel source GigabitEthernet1
tunnel destination 54.215.144.53 /* Elastic IP of the peer CSR */
!


router eigrp 1
bfd interface Tunnel1
network 172.17.0.0
passive-interface GigabitEthernet1
!
```

Step 7 In the AWS console, look in the left nav bar, under VPC Dashboard > "Virtual Private Cloud" > Route Tables, and make a note of the Route Table ID for each Cisco CSR 1000v.

Figure 4: Route Table ID

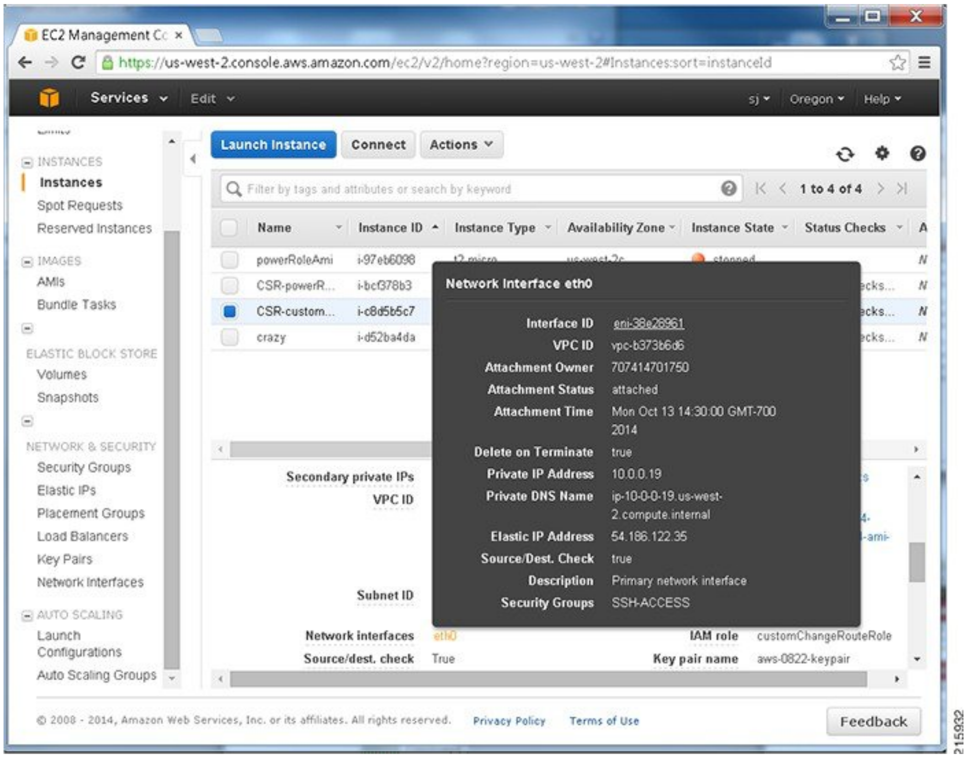


	Name	Route Table ID	Associated With	Main
□		rtb-005a8e65	0 Subnets	Yes
□	route-table-new	rtb-289e5e4d	1 Subnet	No
■		rtb-631bda06	1 Subnet	Yes
□		rtb-035a8e66	1 Subnet	No
□		rtb-d618d9b3	0 Subnets	Yes
□		rtb-621bda07	1 Subnet	No

Destination	Target	Status	Propagated
10.0.0.0/16	local	Active	No
0.0.0.0/0	eni-d679128f / i-bcf378b3	Active	No

Step 8 Look in the EC2 Dashboard, in the left nav bar, under "Instances"and select the Name of an instance in the window on the right. A dialog box shows the details of the Network Interface; for example, titled "Network Interface eth0". Make a note of the Interface ID.

Figure 5: Interface ID



Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks
powerRoleAmi	i-97eb6098	t2.micro	us-west-2c	stopped	
CSR-powerR...	i-bcf378b3				ecks...
CSR-custom...	i-c8d5b5c7				ecks...
crazy	i-d52ba4da				ecks...

Network Interface eth0

Interface ID	eni-38e28961
VPC ID	vpc-b373b6d6
Attachment Owner	707414701750
Attachment Status	attached
Attachment Time	Mon Oct 13 14:30:00 GMT-700 2014
Delete on Terminate	true
Private IP Address	10.0.0.19
Private DNS Name	ip-10-0-0-19.us-west-2.compute.internal
Elastic IP Address	54.186.122.35
Source/Dest. Check	true
Description	Primary network interface
Security Groups	SSH-ACCESS

Secondary private IPs
VPC ID
Subnet ID
Network interfaces eth0
Source/dest. check True
IAM role customChangeRouteRole
Key pair name aws-0822-keypair

(The Route Table ID and Interface ID will be needed in the following steps.)

Step 9 (Cisco IOS XE 3.16 or earlier. For Cisco IOS XE Denali 16.3.1a or later, see step 11.)

Configure the container

virtual-service csr_mgmt

ip shared host-interface GigabitEthernet1

activate

Step 10 (Cisco IOS XE 3.16 or earlier.)

Monitor BFD peer down or similar AWS HA events using an EEM applet.

Define the following EEM environment variables:

RTB—the route table ID for the private subnet VPC route table

CIDR—destination address for the route to be updated in the route table.

Note The `CIDR` for the default route is not the private subnet in the VPC. It is the destination (remote) address that you want to reach from the VPC, which is added as a route in the AWS route table. In many cases this is a default route—0.0.0.0/0.

For example, in the AWS route table, after you've added a default route with CSR-A (ENI) as gateway, if CSR-A fails then CSR-B takes over and updates the default route in the AWS route table to point to its own ENI.

ENI—the network interface ID of the CSR 1000v gigabit interface to which traffic is routed

REGION—the AWS region of CSR 1000v and DNS IP address

Configure the EEM applet in a similar way to that shown in the following example.

Example:

In this example, the four EEM environment variables (**RTB**, **ENI**, **CIDR** and **REGION**) are set for the applet replace-route2(These variables are later used by the `action 1.0 publish-event` command.)

Note For the REGION variable, the DNS IP address is commonly the second usable IP address in the VPC network range. For example, if the VPC network is 10.0.0.0/16, then the DNS IP would be 10.0.0.2.

```
event manager environment RTB rtb-631bda06
event manager environment ENI eni-d679128f
event manager environment CIDR 0.0.0.0/0
event manager environment REGION us-west-2/10.0.0.2
event manager applet replace-route2
event syslog pattern "\(Tunnel1\) is down: BFD peer down notified"
```

Example:

The following command uses the previously defined EEM environment variables, which are to be used in the event of an AWS HA error.

```
action 1.0 publish-event sub-system 55 type 55 arg1 $RTB arg2 $CIDR arg3 $ENI arg4 $REGION
```

After an AWS HA error occurs, routing changes are made to the VPC's route-table-id, network-interface-id and CIDR according to the values specified in the environment variables.

Step 11 (Cisco IOS XE Denali 16.3.1a or later). Use the **cloud provider aws** command.

Monitor BFD peer down events by configuring each CSR 1000v using the **cloud provider aws** command specified below. Use the command to define the routing changes to (VPC) Route-table-id, Network-interface-id and CIDR after an AWS HA error such as BFD peer down, is detected.

Note For the `cidr` command below, this CIDR for the default route is not the private subnet in the VPC. It is the destination (remote) network that you want to reach inside the VPC, which is added as a route in the AWS route table. In many cases this is a default route–0.0.0.0/0.

For example, in the AWS route table, after you've added a default route with CSR-A (ENI) as gateway, if CSR-A fails then CSR-B takes over and updates the default route in the AWS route table to point to its own ENI.

CSR-RTR-A(config)# **redundancy**

CSR-RTR-A(config-red)# **cloud provider** [**aws** | **azure**] *node-id*

# **bfd peer** *ipaddr*

# **route-table** *table-name*

# **cidr ip** *ipaddr*/*prefix*

# **eni** *elastic-network-intf-name*

# **region** *region-name*

Example:

```
CSR-RTR-A(config)# redundancy
```

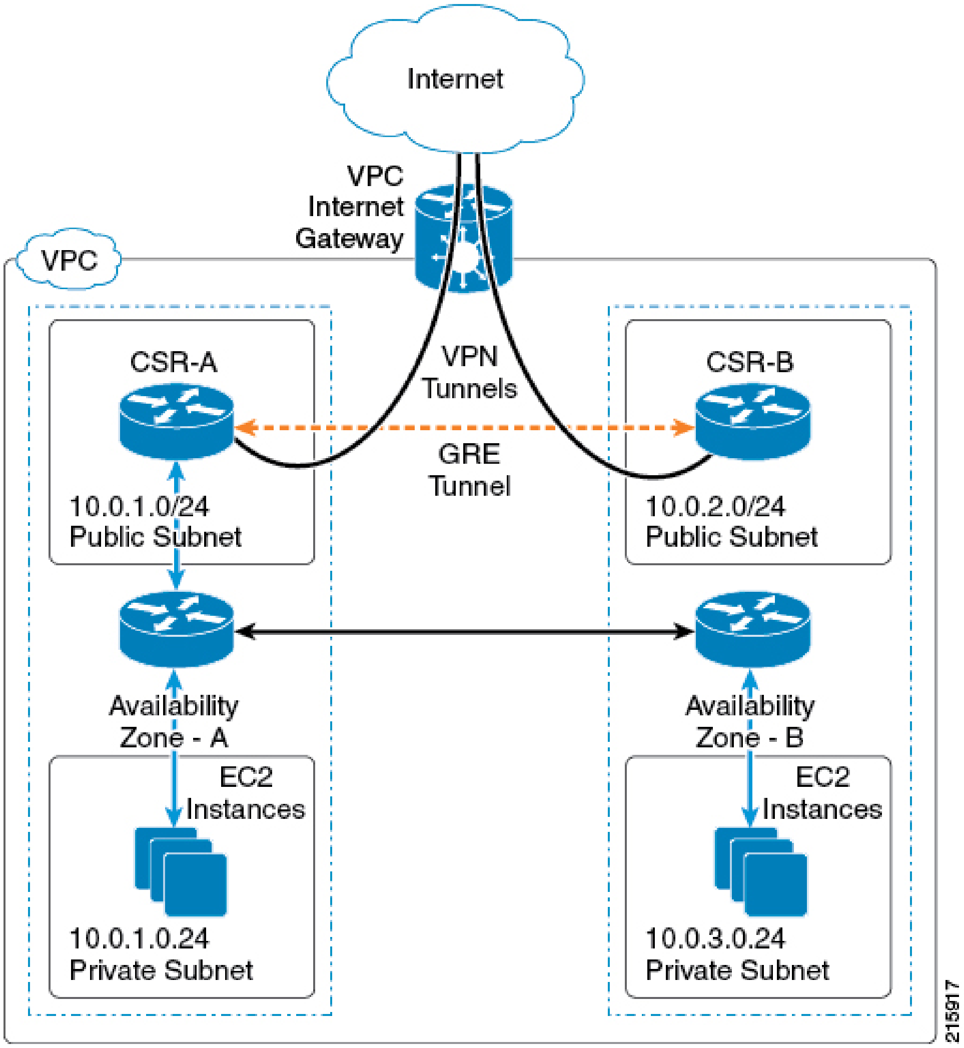
```
CSR-RTR-A(config-red)# cloud provider aws 1
# bfd peer 172.17.1.2
# route-table rtb-30535b54
# cidr ip 172.33.0.0/16
# eni eni-3029b64f
# region us-west-2
```

Next, go to Verifying High Availability, on page 45.

Deployment in Multiple Availability Zones

The following figure "Initial Topology and Traffic Flow" is an example of a VPN gateway configuration that is deployed in two availability zones in a single region, which is a widely used deployment.

Figure 6: Initial Topology and Traffic Flow



This topology uses multiple availability zones and four VPC subnets.

For this scenario, each CSR 1000v is launched in a different availability zone and has a primary Ethernet interface (GigabitEthernet1) that is assigned to the public subnet. The public subnet has a VPC route table with a default route target of the Internet gateway.

Each CSR 1000v also has a VPN tunnel to the internet. These tunnels typically terminate at another VPN device that is located on the enterprise network or another VPC.

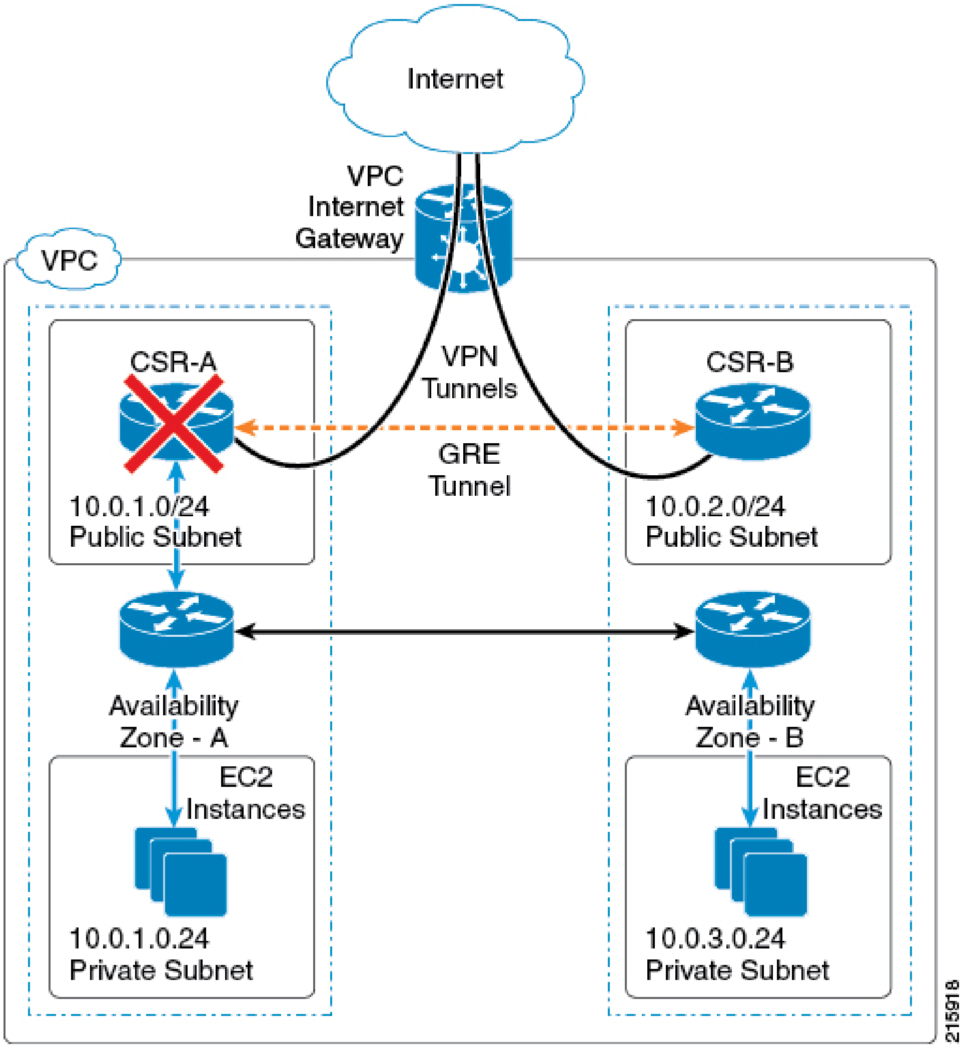
To support the high availability solution, configure a GRE tunnel between the local CSR 1000vs. A GRE tunnel allows each CSR 1000v to exchange the BFD control packets that are used for peer failure detection. The GRE tunnel is established using Elastic-IPs of CSR to avoid the DHCP lease renewal triggering false failures.

EIGRP is used as the routing protocol, though other routing protocols could be used. The primary purpose of the routing protocol is to register as a BFD client. BFD requires at least one client protocol before it initiates neighbor discovery. An extra benefit of the GRE tunnel and the routing protocol is that they can be used to establish a back-up path in case of VPN tunnel failures.

The EC2 instances in private subnet, has its own VPC route table. The default route for this subnet has a target of the public subnet network interface (GigabitEthernet1) of one of the CSRs. Since the CSR is not directly connected to this private subnet, it is recommended to add a static route for the private subnet to each CSR. This static route points to the address of the VPC router on the public subnet. This address is always the first usable address of a subnet. For example, the VPC router address for the subnet 10.0.0.0/24 is 10.0.0.1. Because the VPC route table only allows for one active target per route, only one CSR is in the egress traffic path for this subnet. The remote VPN devices determine the ingress traffic flow over the VPN tunnels, so that it is possible that CSR-B is the active ingress path or that load sharing is being done between CSR-A and CSR-B.

CSR-A then fails, as shown in the following figure "CSR-A Failure" The goal is to shift traffic so that it will egress through CSR-B and no longer ingress through CSR-A.

Figure 7: CSR-A Failure



For the ingress traffic flow, the remote VPN device detects that the VPN tunnel that is terminated at CSR-A is no longer available, by using high availability techniques such as routing protocols (with or without BFD) and IKE dead peer detection.

For the egress traffic direction, CSR-B detects the failure of CSR-A and modifies the VPC route table to redirect traffic to CSR-B.

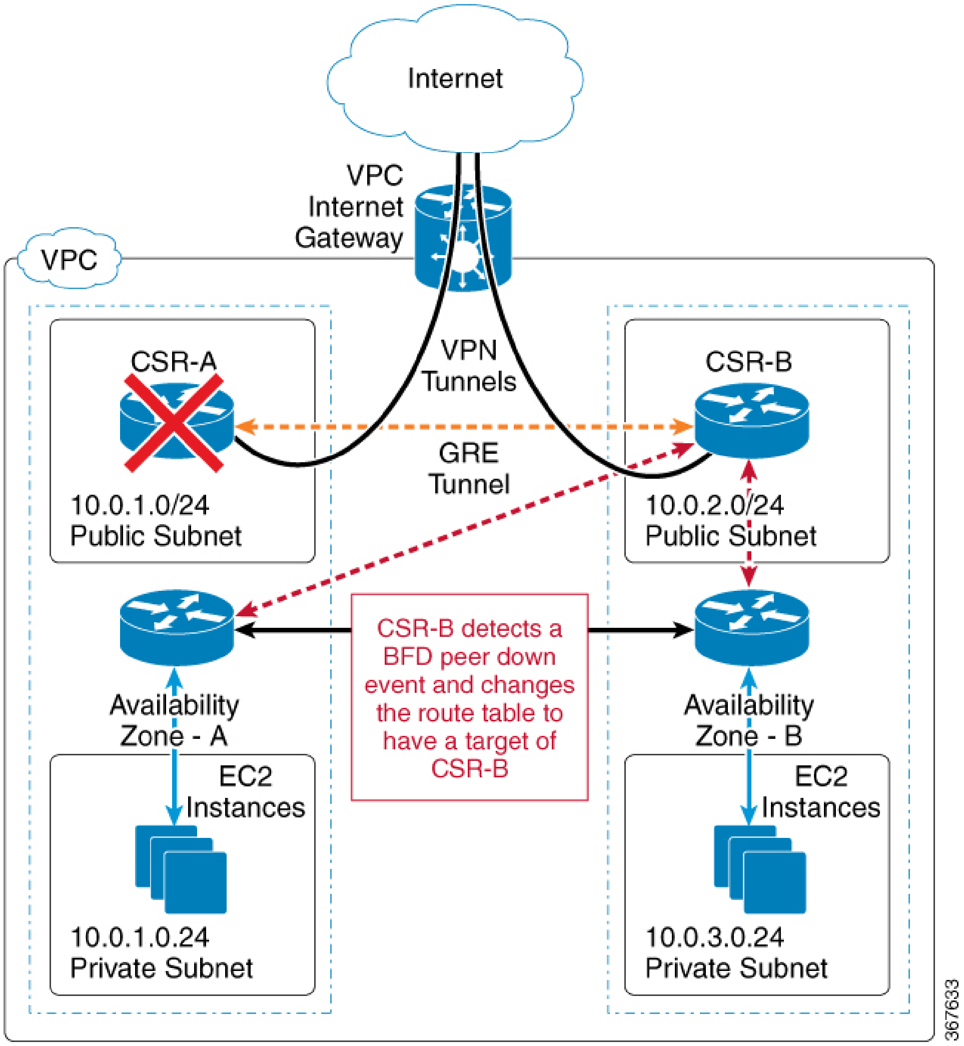
When BFD times out on CSR-B, a log message similar the following is generated.

```
%DUAL-5-NBRCHANGE: EIGRP-IPv4 1: Neighbor 10.0.0.3 (Tunnel33) is down: BFD peer down notified
```

For Cisco IOS XE 3.16 or earlier, EEM is an event detection and automation technology that is available on the CSR 1000v. The EEM applet is configured to run whenever the BFD peer down log message is generated. When it is triggered, the EEM applet uses the AWS API `ec2-replace-route` command to modify the VPC route table to make itself the new target for the default route. See the following figure "CSR-B Modifies the VPC Route Table".

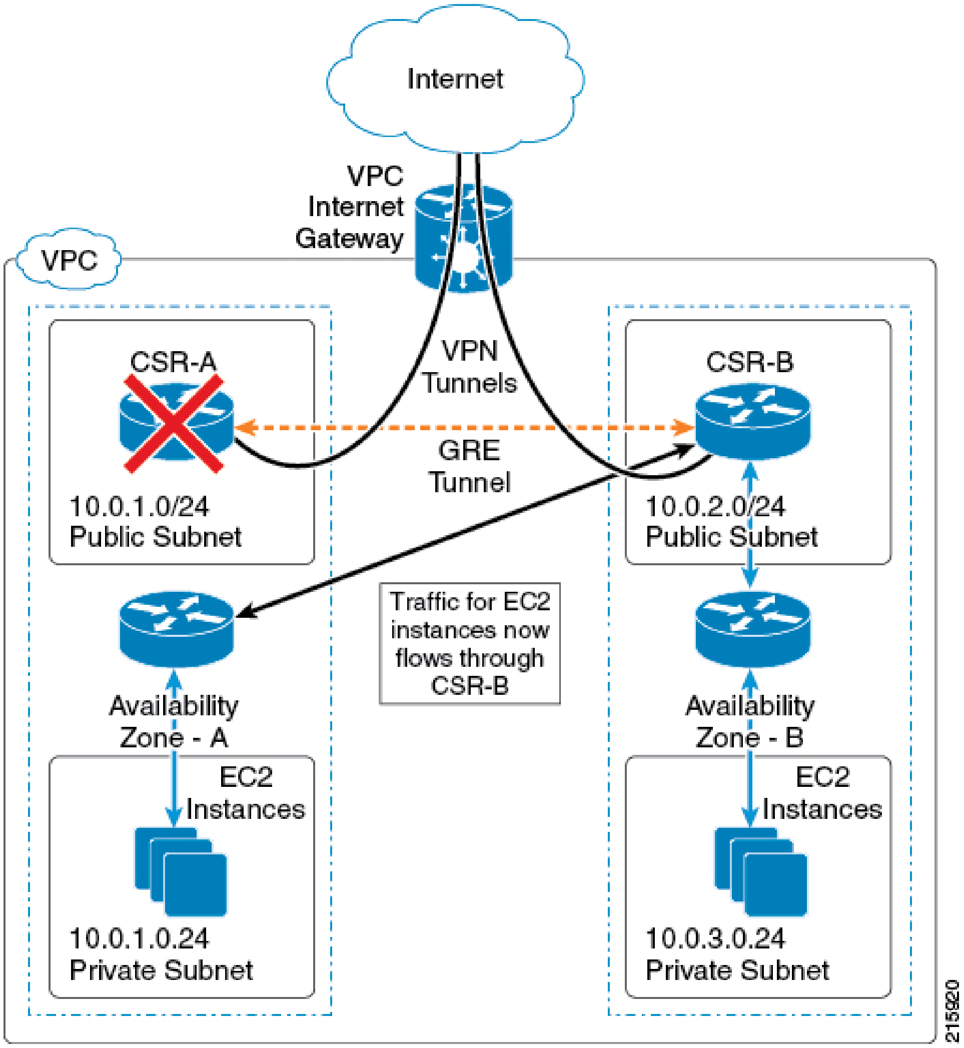
For Cisco IOS XE Denali 16.3.1a or later, use the **cloud provider aws** command to detect the BFD peer down event. Configure the CSR 1000v using the **cloud provider aws** command. This command defines the routing changes to the (VPC) Route-table-id, Network-interface-id, and CIDR when an AWS HA error, such as the BFD peer down event, is detected. See the following figure "CSR-B Modifies the VPC Route Table".

Figure 8: CSR-B Modifies the VPC Route Table



After the VPC route table is modified, the EC2-instances in Private Subnet-1 begin directing egress traffic to the CSR-B, as show in the following figure "New Traffic Flow Through CSR-B".

Figure 9: New Traffic Flow Through CSR-B


Internet
VPC Internet Gateway
VPC
VPN Tunnels
CSR-A
CSR-B
GRE Tunnel
10.0.1.0/24 Public Subnet
10.0.2.0/24 Public Subnet
Traffic for EC2 instances now flows through CSR-B
Availability Zone - A
Availability Zone - B
EC2 Instances
EC2 Instances
10.0.1.0.24 Private Subnet
10.0.3.0.24 Private Subnet
215920

For further deployment and configuration information for a CSR1000v on AWS, see How to Configure High Availability, on page 36.

Verifying High Availability

Verify that the BFD and EIGRP relationships are established and normal on both peers. This example shows the local peer on Tunnel 33, and also the remote peer on Tunnel 98.

```
Router# show bfd neighbors
IPv4 Sessions
NeighAddr                              LD/RD         RH/RS     State     Int
172.17.1.2                           4097/4097       Up        Up        Tu1
Router# show ip eigrp neighbors
EIGRP-IPv4 Neighbors for AS(1)
H   Address                 Interface              Hold Uptime    SRTT   RTO  Q  Seq
```

```
                                                    (sec)         (ms)       Cnt Num
0   172.17.1.2              Tu1                       11 02:02:31   19  1470  0  7
```

For Cisco IOS XE 16.3.1a or later, the following two additional verification commands are available:

show redundancy cloud provider [**aws** | **azure**] *node-id*

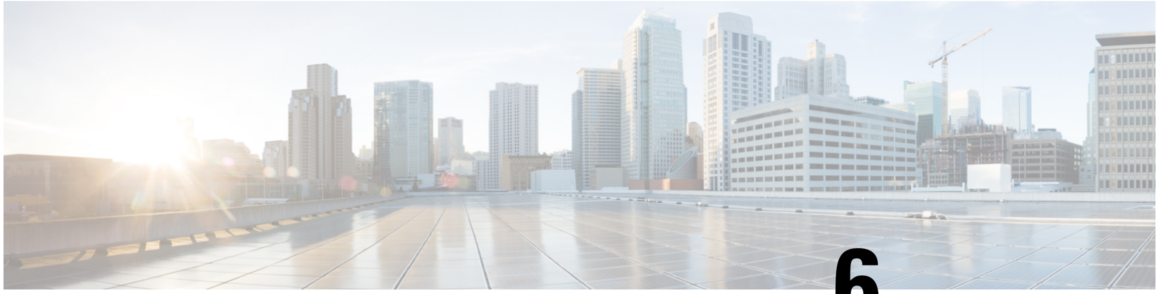
debug redundancy cloud [**all** | **trace** | **detail** | **error**]

Example:

Note In this example, the **show redundancy cloud provider aws** command produces output that includes: "Cloud HA: work_in_progress=FALSE"—you can ignore this message.

```
show redundancy cloud provider aws 1

Cloud HA: work_in_progress=FALSE
Provider : AWS node 1
        State : running
        BFD peer     = 172.18.1.2
        BFD intf     = Tunnel5
        route-table  = rtb-30535b54
        cidr         = 172.33.0.0/16
        eni          = eni-4527b83a
        region       = us-west-2
```



CHAPTER 6

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 CSR 1000V 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.

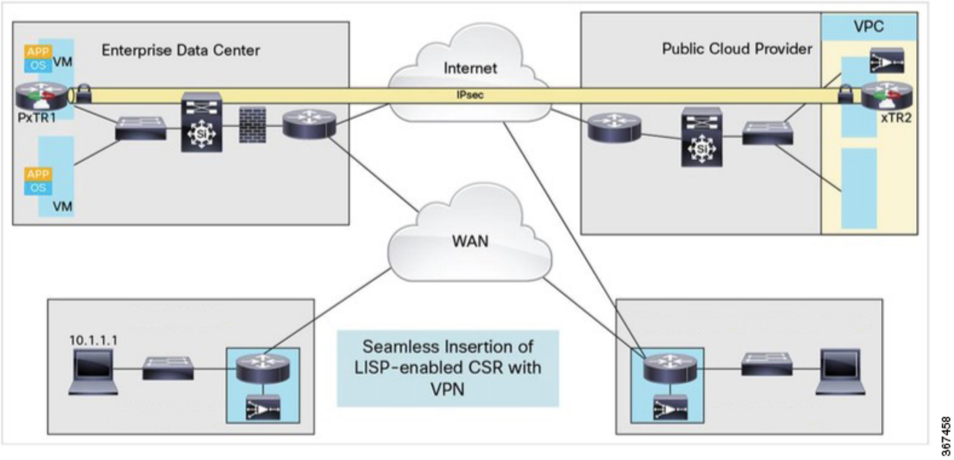
  In a larger or public mapping system deployment, an MS is configured with a partial mesh of generic routing encapsulation (GRE) tunnels and BGP sessions to other map server systems
- **Map-Resolver (MR)**: An MR is a LISP Infrastructure device to which the ITRs send LISP Map-Request queries when resolving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

Information about Configuring LISP Layer 2 Extension

The Cisco CSR 1000v can be deployed 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, 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 CSR 1000v on Amazon Web Services (AWS), where CSR 1000v 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 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

Each CSR 1000V router must be configured with one external IP address. In this case, an IPsec tunnel is built between the IP addresses of the two CSR 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 an AWS CSR1000v Amazon Machine Image (AMI).

How to configure LISP Layer 2 Extension

To configure the L2 extension functionality, you must first deploy the CSR 1000v 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 CSR 1000v 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 CSR routing table, the CSR 1000v 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 CSR 1000v instance on AWS

Procedure

Step 1 Log into Amazon Web Services. In the left navigation pane, click VPC.

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

Step 3 Click Select.

Step 4 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 CSR 1000v instance.

Step 5 Select Create VPC.

Step 6 Select Security > Network ACLs. Click Create Security Group to create a security group for the CSR instance. Configure the following properties:

a) Name-SSH-Access

b) TCP Port 22 traffic-Permitted inbound

c) SSH access to CSR for management-Enabled

Step 7 To create additional security groups, perform step 6.

Step 8 Go to the CSR product page, and click Continue. Click Launch with E2 Console to launch the CSR in accordance with your geographical region.

Step 9 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 10 Launch the CSR 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 11 Associate the instance with a security group (SSH-ACCESS). The security rules enable you to configure firewall rules to control traffic for your CSR 1000v instance.

Step 12 Associate a private key with the CSR 1000v instance. A key pair is a private key and a public key. You must provide the private key to authenticate and connect to the CSR 1000v instance. The public key is stored on AWS. If required, you can create a new key pair.

Step 13 Click Launch Instances.

Step 14 Verify whether the CSR 1000v instance is deployed on AWS.

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

Configure subnets

Procedure

Step 1 Select the CSR 1000v instance.

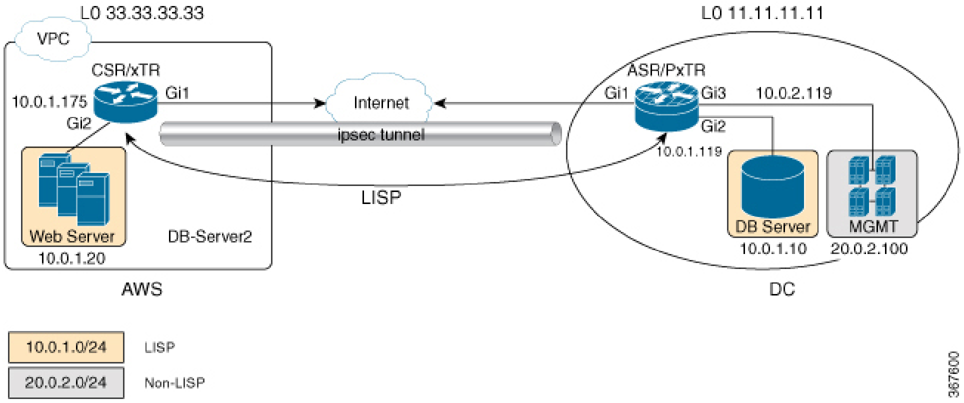
Step 2 Select Actions > Networking > Manage IP Addresses.

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 CSR 1000v on AWS and CSR 1000v on the enterprise system

The communication between the CSR 1000v instance deployed within the enterprise data center and the CSR 1000v 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.



Procedure

Step 1 Configure a CSR 1000v 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 CSR 1000v 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 CSR1000v instance running on AWS

Procedure

To configure LISP xTR on the CSR 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 CSR 1000v on AWS and CSR 1000v on the enterprise system

Procedure

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

Example:

```
csr-aws#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
csr-aws#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       -
csr-aws#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

CSR-DC#show ip lisp
CSR-DC#show ip lisp data
CSR-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
CSR-DC#show ip lisp
```

```
CSR-DC#show ip lisp map
CSR-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

CSR-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

CSR-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
CSR-DC#show ip cef 10.0.1.20
10.0.1.20/32
  nexthop 33.33.33.33 LISP0
CSR-DC#
```

CHAPTER 7

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.

From the 16.12.1 release, IPv6 addressing is supported for CSR 1000v instances running on Amazon Web Services.

To know how to configure the IPv6 functionalities for your CSR 1000v instances, see IPv6 Addressing and Basic Connectivity Configuration Guide.

-