Installing Controller on GCP

Installing Cisco Catalyst 9800 Wireless Controller for Cloud on GCP

The Cisco Catalyst 9800 Wireless Controller for Cloud is a virtual controller running Cisco IOS XE. Most of the Cisco IOS XE features are available on the cloud controller and you can choose to deploy the controller software on Google Cloud Platform (GCP).

To deploy a Cisco Catalyst 9800 Wireless Controller for Cloud on GCP, you must create a project with the following resources: virtual machines, interfaces, virtual private cloud (VPC) networks, routes, public IP addresses, firewall rules, and storage. Resources that exist in different projects can only connect through an external network.

Google Compute Engine instances can run the public images for Linux and Windows Server that Google provides as well as private custom images that you can create or import from your existing systems. Compute instances use SSH public-key authentication. Certain Compute Engine resources live in regions or zones. Resources that live in a zone, such as instances or persistent disks, are referred to as zonal resources.

Other resources, like static external IP addresses, are regional. Regional resources can be used by any resources in that region, regardless of zone, while zonal resources can only be used by other resources in the same zone. A firewall enables you to specify the protocols, ports, and source IP ranges that can reach your instances using security groups. Static IPv4 addresses are used for dynamic cloud computing. Metadata, also known as tags, allows you to create and assign your GCP compute resources.

Note

- The AP in Sniffer mode is not supported in GCP.
- Cisco Catalyst 9800 Wireless Controller for Cloud - Ultra-Low Profile is not supported on public cloud.

GCP VPC Concepts

- A VPC network, sometimes just called a *network*, is a virtual version of a physical network, like a data center network.
- You can launch your GCP cloud resources, such as GCP compute instances, into your VPC.
- You can specify an IP address range for the VPC, add subnets, associate security groups, and configure route tables.
- You can optionally connect your VPC to your own corporate data center using an IPsec GCP managed VPN connection, making the GCP Cloud an extension of your data center.

Creating a VPC in GCP

Perform the following procedure to configure a VPC network in GCP:

Step 1 From the navigation menu in the GCP console, scroll down to **VPC network** and select **VPC networks**.

Step 2 Click **CREATE VPC NETWORK**.

Step 3 Enter a **Name** for the network.

For example, use *custom-network1*.

Step 4 Enter a **Description** for the network.

Step 5 In the **Subnets** section, click **Add Subnet**.

The New subnet dialog box opens. Enter a name for the subnet, for example *subnet-europe-west-192*.

Step 6 Select a **Region**.

For example, use *europe-west1*.

Step 7 Enter an **IP address range**.

For example, use *192.168.5.0/24*.

Step 8 Click **Done**.

This creates a subnet.

Perform Step 5 to Step 9 to create a subnet for the VPC network. You can add multiple subnets to the network.

Step 9 Click **Create**.

This creates a VPC network.

Creating a VPN Connection Using Dynamic Routing

Perform the following procedure to create a VPN connection using dynamic routing:

Step 1 From the GCP console, go to the **VPN** page.

Step 2 Click **Create VPN Connection**.

The Create VPN Connection window is displayed. Enter the following details:

a) Name of the **VPN gateway**.
b) Select the **VPC** network.

The network containing the instances the VPN gateway is going to serve.

c) Select the **Region**.

The region to locate the VPN gateway. Normally, this is the region that contains the instances you wish to reach.

d) Enter the **IP address**.

Select a pre-existing static external IP address. If you don't have a static external IP address, create one by clicking New static IP address from the drop-down menu.

e) Enter the **Peer IP address**.

Public IP address of the peer gateway.

f) Enter **IKE** version.

IKEv2 is preferred, but IKEv1 is supported if that is all the peer gateway can manage.

g) Enter the **Shared Secret**.

Character string used in establishing encryption for that tunnel. You must enter the same shared secret into both VPN gateways. If the VPN gateway device on the peer side of the tunnel doesn't generate one automatically, you can create one using the Generate option.

h) Select the **Routing Option**.
i) Create a **Cloud Router**, by entering the details. Click **Save and Continue**.

Step 3 Create a **Cloud Router**.

a) Enter **Google ASN**.

The private ASN (64512 - 65534, 4200000000 - 4294967294) for the router you are configuring. It can be any private ASN you are not already using. For example, 65002.

AWS creates 2 tunnel interfaces for redundancy. If you do not specify details, AWS randomly generates tunnel options.

Step 4 Enter **BGP** session details.

a) Enter name of the BGP.
b) Enter **Peer ASN**.

The private ASN (64512 - 65534, 4200000000 - 4294967294) for the router you are configuring. It can be any private ASN you are not already using. For example, 65001.

c) Enter **Google BGP IP address**.

The BGP interface IP addresses must be link-local IP addresses belonging to the same /30 subnet in 169.254.0.0/16. For example, 169.254.1.1.

d) Enter **Peer BGP IP address**

AWS creates 2 tunnel interfaces for redundancy. If you do not specify details, AWS randomly generates tunnel options.

Step 5 Click **Create**.

This create the gateway, cloud router, and all the tunnels. Remember that the tunnels will not connect until the peer router is configured.

What to do next

Configure the firewall rules for VPN to allow inbound traffic from the peer network subnets.

Creating a VPN Connection Using Static Routing

Perform the following procedure to create a VPN connection using static routing:

Step 1 From the GCP console, go to the **VPN** page.

Step 2 Click **Create VPN Connection**.

The Create VPN Connection window is displayed. Enter the following details:

a) Name of the **VPN gateway**.

b) Select the **VPC** network.

The network containing the instances the VPN gateway is going to serve. Ensure this network does not conflict with your on-premises networks.

c) Select the **Region**.

The region to locate the VPN gateway. Normally, this is the region that contains the instances you wish to reach.

d) Enter the **IP address**.

Select a pre-existing static external IP address. If you don't have a static external IP address, create one by clicking New static IP address from the drop-down menu.

e) Enter the **Peer IP address**.

Public IP address of the peer gateway.

f) Enter **IKE** version.

IKEv2 is preferred, but IKEv1 is supported if that is all the peer gateway can manage.

g) Enter the **Shared Secret**.

Character string used in establishing encryption for that tunnel. You must enter the same shared secret into both VPN gateways. If the VPN gateway device on the peer side of the tunnel doesn't generate one automatically, you can create one using the Generate option.

h) Enter the **Remote Network IP** range.

For example, 10.0.0.0/8. The range, or ranges, of the peer network, which is the network on the other side of the tunnel from the Cloud VPN gateway you are currently configuring.

i) Specify the **Local Subnet**.

Specifies which IP ranges are routed through the tunnel. This value cannot be changed after the tunnel is created because it is used in the IKE handshake.

j) Specify the **Gateway Subnet**.

You can leave it blank as the local subnet is the default option.

k) Enter the **Local IP** ranges.

You can leave it blank except for the gateway's subnet.

Step 3 Click **Create**.

This create the gateway, and initiates all the tunnels. Remember that the tunnels will not connect until the peer router is configured.

What to do next

Configure the firewall rules for VPN to allow inbound traffic from the peer network subnets.

Create Firewall Rules

Firewall rules allow inbound traffic from the peer network subnets, and you must configure the peer network firewall to allow inbound traffic from your Compute Engine prefixes.

Perform the following procedure to enable traffic to pass to a VM instance and create a firewall rule:

Step 1 From the navigation menu in the Google Cloud Platform Console, scroll down to **VPC network** and select **Firewall Rules**.

Step 2 Click **CREATE FIREWALL RULE** and enter the details.

a) Enter **Name** of the firewall rule.

b) Enter **VPC Network**.

c) Enter **Source filter**.

Choose to filter the traffic using up to four different source filter types.

For example, if you choose to specify a source IP range, you can enter 0.0.0.0/0 to select any IP address.

d) Enter **Source IP ranges**

0.0.0.0/0 (selects all IP ranges in the network).

e) Enter allowed protocols and ports.

A protocol and port range.

String multiple protocol and port ranges together. For example: "icmp", "udp:4789-4790", "tcp:0-6553".

Step 3 Click **Create**.

Creates a firewall rule. To add another firewall rule, repeat the previous steps.

Installing Controller on GCP

Perform the following procedure to deploy a controller instance in GCP:

Before you begin

The following prerequisites apply when deploying a controller on GCP:

- An user account or subscription with GCP.
- A Cloud Identity and Access Management (IAM) user.
- A VPC.
- Subnets.
- A security group.
- A VPN connection.
- For every remote site, create:
  - A customer gateway
  - A VPN connection

Step 1 Click **Compute Engine** and **VM Instances**.

Step 2 Click **CREATE INSTANCE**.

Select a boot disk to create a new controller VM instance (from "OS Images" or custom images) and enter values for the following fields.

a) Specify **Name**.

Name for your VM, using only lowercase letters.

b) Specify **Region**.

c) Specify **Zone**.

A zone is often a data center within a region.

d) Select a **Machine type**.

Supports Small (4 CPU, 8GB RAM), Medium (8 CPU, 16 GB RAM) and Large (10 CPU, 32 GB RAM) profiles.

e) (Optional) Click **Customize** to select the number of cores(vCPUs), memory size, and GPUs.

Step 3 Leave container unselected.

Step 4 Click **Change** on the Boot disk.

Step 5 Go to **OS Images** tab and select the required image using radio buttons.

Note
- The custom image is required only during the initial instance.
- Do not change the boot disk.

Step 6 Click **Select**.

Step 7 In the **Firewall** section, select either: **Allow HTTP traffic** or **Allow HTTPS traffic** to access Web UI.

Step 8 In the **Deletion protection** section, check the **Enable deletion protection** checkbox to prevent the instance from getting deleted.

Step 9 In the **Automation** section, specify the **Startup** script.

This allows you to run scripts when your instance boots up or restarts.

Use this section to add the *username* and *password* to access the instance.

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(>). An example is provide below:

```
Section: IOS configuration
hostname ewlc
username cisco priv 15 pass 0 cisco
!if you want to add more IOS commands, you can add here
Section: Scripts
Section: Python Package
```

Step 10 Click **Networking** tab from the **Management, Security, Disks, Networking, Sole Tenancy** section.

Step 11 Add SSH-key information in the **Network tags**.

Step 12 Click **Add network interface**.

Step 13 In the **Networking Interfaces** dialog box, select the default interface.

For example, the default security group is 10.142.0.0.0/20.

Step 14 In the **Networking Interface** window, select the first *default* interface.

Step 15 Set **IP Forwarding** to **On**.

This prevents the traffic from being blocked.

Step 16 Set **Primary internal IP** as Ephemeral (automatic).

This private IP address is obtained automatically from the selected subnet.

Step 17 Specify **External IP** as Ephemeral (automatic).

You can use this public IP address when you start an SSH session from a terminal server. You may also choose to specify this External IP address as static. The external IP address of each interface is either ephemeral or static.

Step 18 Click **Done**.

Creates the first interface.

Step 19 Click **Create**.

The newly created controller VM instance boots up. It may take a few minutes to complete the boot process.

Accessing Controller Instance on GCP

After completing the configuration, you can connect to the controller using SSH. For that you need private key of the SSH.

Perform the following procedure to access the controller in GCP using SSH:

- Enter the command: **ssh -i** *private-key-file-path username-in-key@ip-address-of-eth1*
- Or, Login using Username and Password that was created using the IOS command during the boot: **ssh** *username@ ip-address-of-eth1*

```
ssh -i user1.key user1@35.100.100.50
```

```
or
```

```
ssh user1@35.100.100.50
```