Installing the Controller in AWS Environment

- Using Private IP, on page 1
- Using Public IP, on page 7

Using Private IP

Overview on Amazon Web Services

The controller can be deployed in Amazon Web Services (AWS) for public cloud solutions.

Prerequisites

Before attempting to launch the controller in AWS, the following prerequisites should be met:

- Create an AWS account.
- Install an SSH client (for example, Putty on Windows or Terminal on Macintosh) to access the controller console.
- Determine the instance type that you want to deploy.
- Create an IAM user.
- Create a key pair.
- Create a VPC.
- Create a security group.
- Create a VPN gateway.
- Create subnets.
- For each remote site, create:
  - Create a customer gateway
  - Create a VPN connection.

Note The AP in Sniffer mode is not supported in AWS.

General Information

- All interfaces in the public cloud are Layer 3 and there are no trunk interfaces.
- All the public cloud IP allocations are done using DHCP in public cloud. You can decide on the IP to be assigned to the controller.
- Supports only one interface, which is shared by device management and wireless management.
- Cisco Catalyst 9800 Wireless Controller for Cloud - Ultra-Low Profile is not supported on public cloud.

Creating a Virtual Private Cloud

Perform the following procedure to configure a VPC in AWS:

Before you begin

- A VPC is a virtual network dedicated to your AWS account and logically isolated from the other virtual networks in the AWS Cloud.
- 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 AWS-managed VPN connection, making the AWS Cloud an extension of your data center.

Note A VPN connection consists of a virtual private gateway attached to your VPC and a customer gateway located in your data center. A virtual private gateway is the VPN concentrator in the Amazon side of the VPN connection. A customer gateway is a physical device or software appliance on your side of the VPN connection.

Procedure

Step 1 Choose **AWS Console > VPC Dashboard > Launch VPC Wizard > VPC with a Private Subnet Only and Hardware VPN Access** to select a VPC configuration.

Step 2 On the **VPC with a Private Subnet Only and Hardware VPN Access** page, enter the details.

Step 3 Choose **VPC Console > Subnets > Create Subnet** to create a subnet.

Step 4 Choose **VPC Console > Security Groups > Create Security Group** to create a security group.

A security group is a virtual firewall that controls traffic to and from one or more instances. When an instance is brought up, you can associate one or more security groups with it. You can use the default security group for the instances, but it is recommended that you create a security group that reflects the role for your instances.

Step 5 Click **Create**.

This creates a VPC.

Creating a Virtual Private Gateway

Perform the following to create an AWS Virtual Private Gateway:

Before you begin

Procedure

Step 1 Choose **VPN Connections > Virtual Private Gateway**.

The **Create Virtual Private Gateway** window is displayed. Enter the following details:

a) Enter a Name Tag.

Note
Use the AWS VPN router name.

b) Choose an ASN.

You can either use a custom ASN or use the default one selected by the Amazon gateway.

Note
After creating the AWS VPN gateway, it will be shown as detached and you need to attach it to a VPC.

Step 2 Click **Actions** and choose **Attach to VPC**.

Step 3 From the pop-up window, select the VPC created earlier.

This way the AWS VPN is attached to the VPC.

Creating a Customer Gateway

Perform the following procedure to create a customer gateway:

Procedure

Step 1 From the AWS console and **VPC** dashboard, click **VPN Connections > Customer Gateways**.

Step 2 Click **Create Customer Gateway**.

The Create Customer Gateway window is displayed. Enter the following details:

a) Name of your VPN router.
b) Select routing as *dynamic* or *static*.
c) Enter the external, internet routable address of your router or firewall.

Step 3 Click **Create Customer Gateway**.

Creating a VPN Connection

Perform the following procedure to create a VPN connection:

Procedure

Step 1 From the AWS console and **VPC** dashboard, choose **VPN Connections > VPN Connections**.

Step 2 Click **Create VPN Connection**.

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

a) Name of the VPN connection.
b) Select the AWS VPN gateway and customer gateway.
c) Select routing as *dynamic* or *static*.
d) Enter the remote subnets reachable through VPN.

The remote subnets are the remote network where your APs will be on-prem.

Step 3 (Optional) Assign subnet and keys for tunnel interfaces for IPSEC VPN.

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

Step 4 Click **Create VPN Connection**.

This creates a VPN connection. It takes a few minutes to set up the connection and change the status from *pending* to *available*.

Step 5 While the VPN is being created, you can download the configuration to deploy in the customer VPN router. Click **Download Configuration**.

Step 6 From the pop-up window, select the brand and type of customer VPN router.

Step 7 Click **Download**.

Creating a Key Pair

Perform the following procedure to create a key pair:

Procedure

Step 1 From the AWS console and **EC2** dashboard, choose **Network & Security > Key pairs**.

Step 2 Click **Create Key Pair**.

Installing the Controller on AWS Using Cloud Formation Template

Perform the following procedure to install the controller on AWS using the Cloud Formation Template:

Before you begin

- A VPC is created with the desired subnet for the controller management interface.
- A managed VPN connection is created from the Enterprise site or sites to the VPC.
- Download the CloudFormation template from the AWS marketplace and save it to your computer.

Procedure

Step 1 From the AWS console and **CloudFormation** page, click **Create Stack**.

Step 2 From **Choose a template** section, select upload template to Amazon S3 option.

The *json* file is loaded directly to AWS.

Step 3 Click **Next**.

The **Specify Details** page is displayed.

Step 4 Enter the stack and instance details:

- Enter any name for the stack you want.
- Enter the controller name as the hostname.
- Enter a name for the Instance Key Pair.
- Enter an AMI for the EC2 instance.

Step 5 Click **Next**.

The **Network Details** page is displayed.

Step 6 Enter **Network** and **User** details.

For the Management Network and Management Security, use the drop-downs to select subnet and security group. Enter an username and password to connect to the instance remotely.

Step 7 Click **Next**.

You get to view the status change from **CREATE_IN_PROGRESS** to **CREATE_COMPLETE**.

Step 8 Choose the **Instance Type**.

Step 9 From the **EC2** dashboard, click **Running Instances**.

The new instance will be in Status Checks (System Status Checks & Instance Status Checks) initializing. You will need to wait for few minutes until it turns green.

When the status turns green, your controller in the cloud is ready to use. You can connect to the controller using SSH with the defined credentials or **.pem** file.

Installing the Controller Using AWS Console

Perform the following procedure to install the controller with the AWS console:

Procedure

Step 1 From the AWS console and **EC2 Management** page, click **Launch Instance**.

Step 2 Click **My AMIs** to select the Cisco Catalyst 9800 Wireless Controller for Cloud AMI.

Step 3 Choose an **Instance Type**.

You will need to choose the instances as per your requirements.

Step 4 Perform the following to configure the instance details:

a) Choose **Availability Zone**.
b) Choose **Network**.
c) Select **Subnet**.
d) Associate an IAM role to restrict or allow usage of instance to other users.

Note
You must disable the public IP during bring-up.

Step 5 From the **Add Storage** page, you can use this optional step to specify additional volumes to be attached to the instance.

Step 6 Perform the following in the **Add Tags** page:

a) Enter **Tag Volumes**.
b) Select **Interfaces**.
c) Select **Instance**.

Step 7 From the **Configure Security Group** page, choose a security group. If a relevant one does not exist, create a new one.

Step 8 Click **Review and Launch**. Review the configuration of your instance.

Step 9 Click **Launch Instances**.

Before launching your instance, you need a key pair to access the instance. Key pair consists of a public key that AWS stores and a private key that you store. If you do not have a key, click **Create a new keypair**, and create a new one, else choose an existing keypair.

What to do next

After the instance is up, you can connect to the Cisco Catalyst 9800 Wireless Controller for Cloud instance using the following UNIX command in your terminal:

```
ssh -i path_to_pem_file ec2-user@[public-ip|DNS name]
```

Note You can obtain the IP and the DNS name from the description of the instance in the EC2 instance console.

Bootstrap Properties for AWS

Table 1: Bootstrap Properties for AWS

Property	Description
hostname	Configures the hostname of the router, as shown in the following example: `hostname="c9800-aws-instance"`
domain-name	Configures the network domain name, as shown in the following example: `domain-name="cisco.com"`
mgmt-ipv4-gateway	Configures the IPv4 management default gateway address, as shown in the following 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, and so on.
	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 following example:
	`ios-config-1="username cisco priv 15 pass ciscoxyz"`
	`ios-config-2="ip scp server enable"`
	`ios-config-3="ip domain lookup"`
	`ios-config-4="ip domain name cisco.com”`
	`ios-config-5="event syslog pattern "\(Tunnel1\) is down: BFD peer down notified""`

Using Public IP

Creating a Virtual Private Cloud

A VPC is a virtual network dedicated to your AWS account and logically isolated from other virtual networks in the AWS Cloud. You can specify an IP address range for the VPC, add subnets, associate security groups, and configure route tables.

Note The AP join to public IP does not require VPN connection.

Perform the following procedure to configure a VPC in AWS:

Procedure

Step 1 Choose **AWS Console > VPC Dashboard > Launch VPC Wizard > VPC with a Private Subnet Only and Hardware VPN Access**.

Step 2 In the **VPC with a Private Subnet Only and Hardware VPN Access** window, enter the details.

Step 3 Choose **VPC Console > Subnets > Create Subnet** to create a subnet.

Step 4 Choose **VPC Console > Security Groups > Create Security Group** to create a security group.

Note
A security group is a virtual firewall that controls traffic to and from one or more instances. When an instance is brought up, you can associate one or more security groups with it. You can use the default security group for the instances, but we recommend that you create a security group that reflects the role of your instances.

Step 5 Click **Create**.

This creates a VPC.

Creating a Key Pair

Perform the following procedure to create a key pair:

Procedure

Step 1 From the AWS console and **EC2** dashboard, choose **Network & Security > Key pairs**.

Step 2 Click **Create Key Pair**.

Installing the Controller on AWS Using Cloud Formation Template

Perform the following procedure to install the controller on AWS using the Cloud Formation Template:

Before you begin

- A VPC is created with the desired subnet for the controller management interface.
- A managed VPN connection is created from the Enterprise site or sites to the VPC.
- Download the CloudFormation template from the AWS marketplace and save it to your computer.

Procedure

Step 1 From the AWS console and **CloudFormation** page, click **Create Stack**.

Step 2 From **Choose a template** section, select upload template to Amazon S3 option.

The *json* file is loaded directly to AWS.

Step 3 Click **Next**.

The **Specify Details** page is displayed.

Step 4 Enter the stack and instance details:

- Enter any name for the stack you want.
- Enter the controller name as the hostname.
- Enter a name for the Instance Key Pair.
- Enter an AMI for the EC2 instance.

Step 5 Click **Next**.

The **Network Details** page is displayed.

Step 6 Enter **Network** and **User** details.

For the Management Network and Management Security, use the drop-downs to select subnet and security group. Enter an username and password to connect to the instance remotely.

Step 7 Click **Next**.

You get to view the status change from **CREATE_IN_PROGRESS** to **CREATE_COMPLETE**.

Step 8 Choose the **Instance Type**.

Step 9 From the **EC2** dashboard, click **Running Instances**.

The new instance will be in Status Checks (System Status Checks & Instance Status Checks) initializing. You will need to wait for few minutes until it turns green.

When the status turns green, your controller in the cloud is ready to use. You can connect to the controller using SSH with the defined credentials or **.pem** file.

Installing the Controller Using AWS Console

Perform the following procedure to install the controller with the AWS console:

Procedure

Step 1 From the AWS console and **EC2 Management** page, click **Launch Instance**.

Step 2 Click **My AMIs** to select the Cisco Catalyst 9800 Wireless Controller for Cloud AMI.

Step 3 Choose an **Instance Type**.

You will need to choose the instances as per your requirements.

Step 4 Perform the following to configure the instance details:

a) Choose **Availability Zone**.
b) Choose **Network**.
c) Select **Subnet**.
d) Associate an IAM role to restrict or allow usage of instance to other users.

Note
You must disable the public IP during bring-up.

Step 5 From the **Add Storage** page, you can use this optional step to specify additional volumes to be attached to the instance.

Step 6 Perform the following in the **Add Tags** page:

a) Enter **Tag Volumes**.
b) Select **Interfaces**.
c) Select **Instance**.

Step 7 From the **Configure Security Group** page, choose a security group. If a relevant one does not exist, create a new one.

Step 8 Click **Review and Launch**. Review the configuration of your instance.

Step 9 Click **Launch Instances**.

Before launching your instance, you need a key pair to access the instance. Key pair consists of a public key that AWS stores and a private key that you store. If you do not have a key, click **Create a new keypair**, and create a new one, else choose an existing keypair.

What to do next

After the instance is up, you can connect to the Cisco Catalyst 9800 Wireless Controller for Cloud instance using the following UNIX command in your terminal:

```
ssh -i path_to_pem_file ec2-user@[public-ip|DNS name]
```

Note You can obtain the IP and the DNS name from the description of the instance in the EC2 instance console.

Bootstrap Properties for AWS

Table 2: Bootstrap Properties for AWS

Property	Description
hostname	Configures the hostname of the router, as shown in the following example: `hostname="c9800-aws-instance"`

Property	Description
domain-name	Configures the network domain name, as shown in the following example: `domain-name="cisco.com"`
mgmt-ipv4-gateway	Configures the IPv4 management default gateway address, as shown in the following 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, and so on.
	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 following example:
	`ios-config-1="username cisco priv 15 pass ciscoxyz"`
	`ios-config-2="ip scp server enable"`
	`ios-config-3="ip domain lookup"`
	`ios-config-4="ip domain name cisco.com”`
	`ios-config-5="event syslog pattern "\(Tunnel1\) is down: BFD peer down notified""`

Enabling Public IP on the Controller

Perform the following procedure to enable the Wireless Management Interface with Public IP:

Before you begin

- The controller has two types of IPs:
  - Private IP
  - Public IP

Note By default, the public IP is not enabled.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device> configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 2	**wireless management interface** *interface-type interface-number* **Example:** `Device(config)# wireless management interface gigabit1`	Defines the management interface. Where, • *interface-type* refers to the Gigabit interface. • *interface-number* is 1. **Note** The Public cloud VM supports only the following: • Gigabit as the interface-type. • 1 as the interface-number.
Step 3	**public-ip** *public-ip* **Example:** `Device(config-mgmt-interface)# public-ip 2.2.2.2`	Defines the external Public IP.
Step 4	**end** **Example:** `Device(config-mgmt-interface)# end`	Returns to privileged EXEC mode.

Enabling CAPWAP Discovery to Respond Only with Public or Private IP

- By default, the CAPWAP discovery response covers both private and public IP.
- To enable the controller to respond only with a private or public IP, see the *Configuring CAPWAP Discovery to Respond Only with Public or Private IP (CLI)* section in Cisco Catalyst 9800 Series Wireless Controller Software Configuration Guide.

AP Authorization Configuration

Kindly refer to the following link:

https://www.cisco.com/c/en/us/support/docs/wireless/catalyst-9800-series-wireless-controllers/213916-catalyst-9800-wireless-controllers-ap-au.html

Note When priming the AP, ensure that you use the Public IP to connect to the controller. You can restrict the APs using AP authorization.