Cisco Catalyst 9800-CL Cloud Wireless Controller Installation Guide

First Published: 2018-11-20

Last Modified: 2020-11-30

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

CONTENTS

Preface

This preface describes this guide and provides information about the conventions used in this guide, along with details about related documentation. It includes the following sections:

Document Revision History

The following table shows the changes made to this document:

Date	Change Summary
November 2018	First version of the document.
July 2019	Added information on support for Google Cloud Platform (GCP).
February 2020	Added information on support for Microsoft Hyper-V.

Document Objectives

This publication describes the installation of the .

Audience

This publication is primarily designed for persons responsible for installing, maintaining, and troubleshooting the . The users of this guide should:

- Be familiar with electronic circuitry and wiring practices.

- Have experience working as electronic or electromechanical technicians.
- Have experience in installing high-end networking equipment.

Note Some procedures described in this guide require a certified electrician.

Conventions

Text Type	Indication
User input	Text the user should enter exactly as shown or keys a user should press appear in this font.
Document titles	Document titles appear in *this font*.
System output	Terminal sessions and information that the system displays appear in **this font**.
CLI commands	CLI command keywords appear in **this font** . Variables in a CLI command appear in *this font* .
[ ]	Elements in square brackets are optional.
{x | y | z}	Required alternative keywords are grouped in braces and separated by vertical bars.
[x | y | z]	Optional alternative keywords are grouped in brackets and separated by vertical bars.
String	A nonquoted set of characters. Do not use quotation marks around the string or the string will include the quotation marks.
< >	Nonprinting characters such as passwords are in angle brackets.
[ ]	Default responses to system prompts are in square brackets.
! #	An exclamation point (!) or a pound sign (#) at the beginning of a line of code indicates a comment line.

Note Means *reader take note*. Notes contain helpful suggestions or references to material not covered in the document.

Tip Means *the following information will help you solve a problem*. The tips information might not be troubleshooting or even an action, but could be useful information, similar to a Timesaver.

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

Timesaver: Means *the described action saves time*. You can save time by performing the action described in the paragraph.

Warning **IMPORTANT SAFETY INSTRUCTIONS**

This warning symbol means danger. You are in a situation that could cause bodily injury. Before you work on any equipment, be aware of the hazards involved with electrical circuitry and be familiar with standard practices for preventing accidents. Use the statement number provided at the end of each warning to locate its translation in the translated safety warnings that accompanied this device.

SAVE THESE INSTRUCTIONS. Statement 1071

Related Documentation

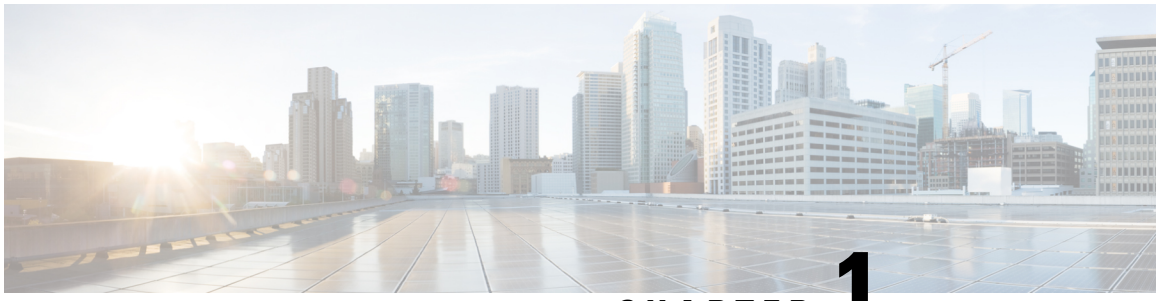
See the following documentation for more information about the Cisco Catalyst 9800 Wireless Controller:

- *Release Notes for Cisco Catalyst 9800 Wireless Controller*
- *Cisco Catalyst 9800 Series Wireless Controller Software Configuration Guide*
- *Cisco Catalyst 9800 Series Wireless Controller Command Reference*
- *Cisco Wireless Solutions Software Compatibility Matrix*

Obtaining Documentation and Submitting a Service Request

For information on obtaining documentation, using the Cisco Bug Search Tool (BST), submitting a service request, and gathering additional information, see What's New in Cisco Product Documentation.

To receive new and revised Cisco technical content directly to your desktop, you can subscribe to the What's New in Cisco Product Documentation RSS feed. RSS feeds are a free service.



CHAPTER 1

Overview of Cisco Catalyst 9800 Wireless Controller for Cloud

Introduction

The Cisco Catalyst 9800-CL Cloud Wireless Controller (referred to as "controller" in this document) is a virtual wireless controller that is deployed on a Cisco Unified Computing System (UCS) server as a virtual machine (VM) instance on a Linux-based 64-bit guest operating system. This controller supports a subset of Cisco IOS XE software features and technologies, providing Cisco IOS XE features on a virtualization platform.

When the controller is deployed as a VM, the Cisco IOS XE software functions as if it were deployed on a traditional Cisco hardware platform.

Benefits of Virtualization

The controller uses the benefits of virtualization to provide the following:

- Hardware independence—Because the controller runs on a VM, it can be supported on the x86 hardware that the virtualization platform supports.
- Sharing of resources—The resources used by the controller are managed by the hypervisor; these resources can be shared among VMs. The amount of hardware resources that the VM server allocates to a specific VM can be reallocated to another VM on the server.
- Flexibility in deployment—You can easily move a VM from one server to another. Thus, you can move the controller from a server in one physical location to a server in another physical location without moving any hardware resources.

Software Configuration and Management

You can perform software configuration and management of the controller using the following methods:

- Use the virtual video graphics array (VGA) console or the console on the virtual serial port to access the Cisco IOS XE CLI commands.
- Use remote SSH or Telnet to access the Cisco IOS XE CLI commands.

Note The controller may reload when you run the **show redundancy trace main** command from the serial console.

Serial console is not recommended for large scale deployments. We recommend that you use Telnet or SSH for this purpose. For more information on how to add a virtual serial port, see *Adding Virtual Serial Port* in Cisco Catalyst C9800-CL Wireless Controller Virtual Deployment Guide.

Virtual Machines

The controller can run as a VM. A VM is a software implementation of a computing environment in which an operating system or program can be installed. The VM typically emulates a physical computing environment, but requests for CPU, memory, hard disk, network, and other hardware resources are managed by a virtualization layer that translates these requests to the underlying physical hardware.

You can deploy an Open Virtualization Archive (OVA) file for ESXi. The OVA file package simplifies the process of deploying a VM by providing a complete definition of the parameters and resource allocation requirements for the new VM.

An OVA file consists of a descriptor (.ovf) file, a storage (.vmdk) file, and a manifest (.mf) file.

- Descriptor or .ovf file—An XML file with .ovf as the extension, and consisting of all the metadata about the package. It encodes all the product details, virtual hardware requirements, and licensing.
- Storage or .vmdk file—A file format that encodes a single virtual disk from a VM.
- Manifest or .mf file—An optional file that stores the Secure Hash Algorithm (SHA) key generated during packaging.

Hypervisor Support

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

Caution The controller might crash while taking a snapshot. We recommend that you use RAID0 configuration on the UCS to avoid a crash.

- Ensure that you use VMware ESXi Version 5.5 or later.

Supported Hypervisor Types

Installation of the controller is supported on selected Type 1 (native, bare metal) hypervisors. Installation is not supported on Type 2 (hosted) hypervisors, such as VMware Fusion, VMware Player, and Virtual Box.

Hypervisor vNIC Requirements

Depending on the controller's version number, each of the hypervisors support different virtual Network Interface Card (vNIC) types.

Table 1: vNIC Requirements for VMware ESXi

vNIC Requirements for VMware ESXi	Value
NIC Types Supported	VMXNET3
vNIC Hot Add Support	Yes
vNIC Hot Remove Support	Yes

Table 2: vNIC Requirements for Kernel-Based Virtual Machine (KVM)

vNIC Requirements for KVM	Value
NIC Types Supported	Virtio, ixgbevf, ixgbbe
vNIC Hot Add Support	Yes
vNIC Hot Remove Support	No

Table 3: vNIC Requirements for Amazon Web Services (AWS)

vNIC Requirements for AWS	Value
NIC Types Supported	VMXNET3
vNIC Hot Add Support	No
vNIC Hot Remove Support	No

Server Requirements

The server and processor requirements are different, depending on the software release. The following table captures the server requirements:

Table 4: Server Requirements

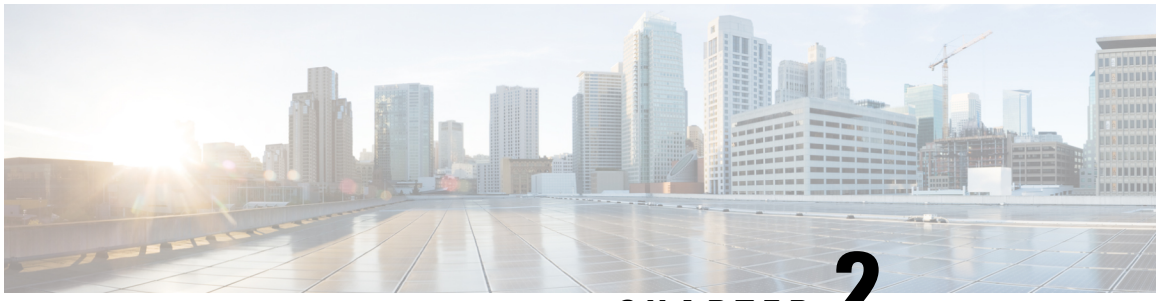
Software Release	Intel	AMD
Cisco IOS XE Gibraltar 16.10.1 and later	64-bit Intel Core2 and later-generation processors with virtualization technology extensions.	Equivalent of 64-bit Intel Core2 and later-generation processors with virtualization technology extensions.

Supported Templates and Hardware Requirements

From 17.3 release onwards, high throughput templates can be configured on the Cisco Catalyst 9800-CL Cloud Wireless Controller private cloud instances. With this enhancement, the throughput can be raised from 2 Gbps to 5 Gbps.

Table 5: Supported Templates and Hardware Requirements

Model Configuration	Small (Low Throughput)	Medium (Low Throughput)	Large (Low Throughput)	Small (High Throughput)	Medium (High Throughput)	Large (High Throughput)
Minimum number of vCPUs	4	6	10	7	9	13
Minimum CPU Allocation (MHz)	4,000	6,000	10, 000	4000	6000	10,000
Minimum Memory (GB)	8	16	32	8	16	32
Required Storage (GB)	16	16	16	16	16	16
Virtual NICs (vNIC) (*) 3rd NIC for High Availability	2/(3)*	2/(3)*	2/(3)*	2/(3)*	2/(3)*	2/(3)*



CHAPTER 2

Installing Controller in VMware Environment

Overview of VMware Environment

The controller runs on the Cisco IOS-XE operating system. The virtual installation images contain the underlying Cisco IOS-XE operating system and the Wireless Controller code. You must download the Cisco IOS XE software from Cisco.com and install it directly in the virtual machine (VM) environment. However, as part of the initial installation process, you must first provision the attributes of the VM so that the controller software can install and boot.

The high-level tasks required to install the controller are listed here.

Note The different installation options are dependent on the hypervisor being used.

Install the Controller Using an OVA File

1. Download the controller software (.ova file) from Cisco.com.
2. Create a network interface on the VM.
3. Deploy the OVA template using the VMware vSphere client to create a controller VM.
4. Power on the VM to boot the controller software.

Obtaining the Controller VM Image (OVA File)

1. Open the Cisco Catalyst 9800 Wireless Controller for Cloud product page.
2. Click the **Download Software** link to open the **Download Software** page.
3. In the **Download Software** page, select the model.
4. Click the corresponding Cisco IOS XE software. Note that the recommended Cisco IOS XE release is selected by default.
5. From the list of available images, click **Download Now** or **Add to Cart**.
6. Follow the instructions for downloading the software.

Installation Options

The controller currently supports only the following installation options:

- Deploying the OVA template in a VM environment.
- Deploying the controller using ISO installation.

Note The .ova file can be used only for first-time installation. It cannot be used for upgrading the Cisco IOS XE software version.

ROMMON and the Controller

The controller does not include a ROMMON image similar to what is included in many Cisco hardware-based devices. During the initial bootloader process, the installation script creates a clean version of the controller software image known as the Golden Image, and places it in a nonaccessible partition. This clean version can be used if the software image is not working properly or cannot be booted.

Installing in a VMware ESXi Environment

This section includes information about VMware tools and VM requirements for the controller running the latest Cisco IOS XE software, as well as a list of the supported VM features.

The controller can run on the VMware ESXi hypervisor. You can use the same hypervisor to run several VMs.

The VMware vSphere web client is a web application that runs on the PC and accesses the vCenter Server. You can use the VMware vSphere Web Client software to create, configure, and manage VMs on the VMware vCenter Server and to start or stop the controller.

For more details about installing vSphere products, see the corresponding VMware product documentation.

Note Hot delete of the interface from the vSphere client is not supported until Cisco IOS XE Amsterdam 17.1.1s.

VMware Requirements

The VMware tools required to deploy the controller are as follows:

- VMware vSphere Web Client. The following version is supported:
  - VMware vSphere Web Client 6.0
- VMware vCenter Server.

  For the list of supported versions, see the Release Notes.
- VMware vSwitch. Standard or distributed vSwitches are supported.
- Hard drive. Only a single hard disk drive is supported. Multiple hard disk drives on a VM are not supported.
- vCPUs. The following vCPU configurations are supported:
  - Small Template—4 vCPUs (requires minimum 4-GB RAM allocation)
  - Medium Template—6 vCPUs (requires minimum 16-GB RAM allocation)
  - Large Template—10 vCPUs (requires minimum 32-GB RAM allocation)
- Virtual CPU core
- Virtual hard disk space—Minimum 8 GB is required.
- Virtual Network Interface Cards (vNICs).

Supported VMware Features and Operations

VMware supports various features and operations that allow you to manage your virtual applications and perform operations such as cloning, migration, shutdown, and resume.

Some of these operations cause the runtime state of the VM to be saved and then restored upon restarting. If the runtime state includes traffic-related state, on resumption or replay of the runtime state, additional errors, statistics, or messages are displayed on the user console. If the saved state is just configuration driven, you can use these features and operations without any issues.

Caution VMware functionalities, such as, vMotion, Snapshot, Distributed Resource Scheduler (DRS), vNIC Teaming and SR-IOV modes are supported. However, cloning from snapshots is not supported.

Also, vMotion, DRS, Snapshots, and vNIC Teaming are not supported when SR-IOV mode is enabled.

For more information, see the Cisco Catalyst 9800-CL Wireless Controller for Cloud Data Sheet.

For more information about VMware features and operations, see the corresponding VMware Documentation.

Creating a Network Interface on a VM

Perform the following tasks in the VMware vSphere Client to create a network interface.

Before you begin

This procedure is required only for the first installation of the controller.

Step 1 Log in to the VMware vSphere Client.

Step 2 In the vSphere GUI, click the **Configuration** tab.

Step 3 In the **Networking** area, click **Add Networking…**

Step 4 Under **Connection Type**, retain the default settings, and click **Next**.

Step 5 Under **Network Access**, select one of the VM names.

Step 6 Click **Next**.

Step 7 Under **Connection Settings**, enter a name in the **Network Label** field.

Step 8 From the **VLAN ID (Optional)** drop-down list, choose **All (4095)**.

Step 9 Click **Next**.

Step 10 Under **Summary**, confirm the updates and click **Finish**.

The newly added network interface is now available in the **Networking** area.

Configuring NIC Teaming on a Virtual Switch

You can include two or more physical NICs in a team to increase the network capacity of a virtual switch. This is termed as NIC teaming. To distribute how the virtual switch distributes the network traffic between the physical NICs in a team, you select load balancing depending on the needs and capabilities of your environment.

Perform the following tasks in the VMware vSphere Client to configure NIC teaming on a virtual switch.

Before you begin

This procedure is required only for configuring NIC teaming.

Note VMXNET3 is the virtual adapter type supported on the controller.

Step 1 Log in to the VMware vSphere Client.

Step 2 Navigate to the virtual switch.

Step 3 Click **Edit** to view the properties of the virtual switch.

Step 4 Navigate to **NIC Teaming** tab on the virtual switch properties page.

Step 5 From the **Load Balancing** drop-down menu, specify how the virtual switch load balances the outgoing traffic between the physical NICs in a team.

You can configure the following options on a virtual switch:

- Route based on the originating virtual port ID—Selects an uplink based on the virtual port IDs on the switch.
- Route based on IP hash—Selects an uplink based on a hash of the source and destination IP address of each packet.

- Route based on source MAC hash—Selects an uplink based on a hash of the source Ethernet.
- Use explicit failover order—Uses the highest order uplink from the list of active adapters that passes failover detection criteria. No actual load balancing is performed with this option.

Step 6 From the **Network Failover Detection** drop-down menu, specify a method for failover detection.

You can configure the following options on a virtual switch:

- Link Status Only—Relies on the link status provided by the network adapter. This option detects failures, such as, physical switch power failures and removed cables.
- Beacon Probing—Sends out and listens for beacon probes on all NICs in a team, and uses this details along with the link status to determine link failure.

Step 7 From the **Notify Switches** drop-down menu, select **Yes** or **No** to notify the switch for any failover.

Step 8 From the **Failback** drop-down menu, select whether a physical adapter is returned to active status after recovering from a failure.

If failback is set to **Yes**, the adapter is returned to active immediately after recovery. By default, a failback policy is enabled on a NIC team.

If failback is set to **No**, a failed adapter is left inactive after recovery until another active adapter fails and needs to be replaced.

Note If a physical NIC that stands first in the failover order experiences intermittent failures, the failback policy might lead to frequent updates in the NIC. The physical switch undergoes frequent changes in MAC addresses, and the physical port might not accept traffic immediately after an adapter becomes online. To minimize such delays, you can change the following settings on the physical switch:

- Disable Spanning Tree Protocol (STP) on physical NICs connected to the ESXi hosts.
- Enable PortFast mode or PortFast trunk mode for access and trunk interfaces respectively. This saves around 30 seconds during the initialization of the physical switch port.

Step 9 Review your settings and apply the configuration.

Information About Deploying Controller OVA on a VM using vSphere

You can use the controller OVA file package that is provided to deploy the controller on the VM.

The OVA can be deployed using the VMware vSphere Client, VMware OVF Tool, or the Common OVF Tool (COT).

Restrictions and Requirements

The following restrictions apply when deploying the OVA package on the VM:

- If the virtual CPU configuration is changed, the controller must be rebooted. Changing the RAM allocation does not require rebooting the controller.

- When deploying the OVA, the VM requires two virtual CD/DVD drives, one for the OVF environment file and another for the .iso file.

Deploying the Controller OVA File on a VM Using vSphere

Perform the following steps in the VMware vSphere Client:

You can use the controller OVA file package that is provided, to deploy the controller on the VM.

The OVA can be deployed using the VMware vSphere Client, VMware OVF Tool, or the Common OVF Tool.

Before you begin

- If the virtual CPU configuration is changed, the controller must be rebooted. However, changing the RAM allocation does not require rebooting the controller.
- When deploying the OVA, the VM requires two virtual CD/DVD drives, one for the OVF environment file and another for the .iso file.
- Ensure that the Network Interface is set up properly.

Step 1 Log in to the VMware vSphere Client.

Step 2 From the vSphere Client menu, choose **File > Deploy OVF Template**.

Step 3 In the **OVA** wizard, select the source of the controller OVA that is to be deployed.

The **OVF Template Details** window displays information about the OVA.

Step 4 Click **Next**.

Step 5 In the **Name and Location** field, specify the name for the VM and click **Next**.

Step 6 Click **Next**.

Step 7 Under **Deployment Configuration**, select the required profile from the drop-down list.

Step 8 Under **Disk Format**, retain the default settings (**Thick Provision Lazy Zeroed**) and click **Next**.

Step 9 From the **Network Mapping** drop-down list, allocate one or more virtual Network Interface Cards (vNICs) to the destination network. Connect each network to a unique interface. We recommend the following mapping:

- GigabitEthernet 1 to device management interface and map it to the out-of-band management network.
- GigabitEthernet 2 to wireless management interface and map it to the network to reach APs and services. Usually this interface is a trunk to carry multiple VLANs.
- GigabitEthernet 3 to high-availability interface and map it to a separate network for peer-to-peer communication for SSO.

Step 10 Under **Ready to Complete**, verify all the deployment settings.

Step 11 Click **Finish** to deploy the OVA.

The controller VM now appears on the left panel.

Step 12 Click **Power On** to automatically power on the VM.

Edit the Basic Properties of VM

Perform the following tasks in the VMware vSphere Client:

Step 1 Log in to the VMware vSphere Client.

Step 2 In the vSphere GUI, click the **Configuration** tab.

Step 3 In **Networking** area, click **Properties** of the newly added network interface.

Step 4 Click **Edit** to view the properties of the network interface..

Step 5 Click the **Security** tab.

Step 6 Uncheck the checked VM name.

Step 7 In the **Promiscuous Mode**, perform the following tasks:

The **Promiscuous Mode** is set to **Reject** by default.

Note Promiscuous mode is a security policy which can be defined at the virtual switch or port-group level in vSphere ESXi. Tagged traffic will not flow properly without this mode.

- Check the check box.
- From the drop-down list, select **Accept** to view the traffic sent and received through this switch.

Note Ensure that **Forged Transmits** is also set to **Accept**.

Step 8 Click **OK**, and then click **Close**.

Configuring SR-IOV for VMware ESXi

Recommended Software Versions for SR-IOV

Table 6: List of Supported NIC Types

NIC	Firmware	Driver Version	Host OS
Intel x710	7.10	I40en 1.10.6 INETCLI Plugin version 1.4.1	VMware Version 6.5 and above

Configuring SR-IOV Mode on the Interface

Step 1 Create a port group without any ports.

Step 2 Create a dummy virtual switch and attach the port group created in **Step 1** to this switch.

Step 3 Enable SR-IOV for x710 PCI device ports from **Host > Manage > Hardware**.

Note One VF is created on each port to maximize performance.

Step 4 Create an eWLC instance. While adding the network adapter, perform the following:

a. Choose **Network Adapter** as the created port group.

b. Choose **Adapter Type** as the SR-IOV passthrough.

c. Choose **Physical Function** as the one mapped to the port on which the SR-IOV is enabled.

d. Set the **Guest OS MTU Change** to **Allow**.

e. Click **Save**.

Enabling Trusted Mode and Disabling Spoof Check

To enable SSH to ESXi from the GUI, perform the following:

Step 1 Navigate to **Host > Actions > Services > Enable SSH**.

Step 2 Set **SSH** to **ESXi**.

To disable spoof check, perform the following:

While the controller is booting up, set the trusted mode and spoof check using the following command:

esxcli intnet sriovnic vf set -t on -s off -v <*vf-id*> *-n* <*physical_port_name*>

Here,

<*physical_port_name*> is the SR-IOV port to which the VM is associated.

<*vf-id*> is the VF ID assigned to the VM instance.

Sample output:

```
[root@localhost:~] esxcli intnet sriovnic vf set -t on -s off -v 0 -n vmnic6
```

Note To verify if the VF ID has been assigned to the controller, check the **vmkernel.log** file in **/var/log** location.

Configuring SR-IOV Setting Persistence

SR-IOV configurations configured in the above way are not persistent across reboots. To resolve this issue, you can execute the above configuration as a service that is auto-enabled on host reboots.

Step 1 For firmware and driver versions prior to and including firmware version 7.0, and driver version 1.8.6, you need to stop the VM load at boot up and perform *Enabling Trusted Mode and Disabling Spoof Check*.

Step 2 For firmware and driver versions above and including firmware version 7.10, and driver version 1.10.6, enter the following commands once after setting the trusted mode and spoof check to make the setting permanent:

esxcli system module parameters set -a -p max_vfs=1,1,1,1 -m i40en

esxcli system module parameters set -m i40en -p trust_all_vfs=1,1,1,1

Verifying SR-IOV Driver and Firmware Version

You can verify the NICs using the following command:

esxcli network nic list

```
[root@localhost:~] esxcli network nic list

Name     PCI Device    Driver  Admin Status  Link Status  Speed  Duplex  MAC Address
   MTU  Description

-------  ------------  ------  ------------  -----------  -----  ------  -----------------
  ----  ------------------------------------------------------------


vmnic6   0000:87:00.0  i40en   Up            Up           10000  Full    3c:fd:fe:ee:ce:d8
  1500  Intel Corporation Ethernet Controller X710 for 10GbE SFP+

vmnic7   0000:87:00.1  i40en   Up            Down             0  Half    3c:fd:fe:ee:ce:d9
  1500  Intel Corporation Ethernet Controller X710
```

You can view the parameters for a particular interface using the following command:

esxcli network nic get -n vmnic6

```
[root@localhost:~] esxcli network nic get -n vmnic6
Advertised Auto Negotiation: true

   Advertised Link Modes: Auto, 1000BaseSR/Full, 10000BaseSR/Full

   Auto Negotiation: true

   Cable Type: FIBRE

   Current Message Level: 0

   Driver Info:

         Bus Info: 0000:87:00:0

         Driver: i40en

         Firmware Version: 7.10 0x80006471 1.2527.0

         Version: 1.10.6
[root@localhost:~]  esxcli intnet sriovnic vf get -n vmnic6
VF ID           Trusted         Spoof Check
-----           -------         -----------
0               true            false
```

You can verify the processor, memory, vNIC, hypervisor, and throughput profile details using the following command:

```
Device # show platform software system all
```

```
Device # show platform software system all

Controller Details:
==================
VM Template: medium
Throughput Profile: high
AP Scale: 3000
Client Scale: 32000
WNCD instances: 3
Processor Details
==================
Number of Processors : 9
Processor : 1 - 9
vendor_id : GenuineIntel
cpu MHz  : 2593.748
cache size : 4096 KB
Crypto Supported : Yes
model name : Intel Core Processor (Haswell, IBRS)
Memory Details
==============
Physical Memory : 16363364KB

VNIC Details
============
Name              Mac Address     Driver Name       Status  Platform MTU
GigabitEthernet1  3cfd.fede.ccbc  net_i40e_vf       DOWN    1522
GigabitEthernet2  3cfd.fede.ccbd  net_i40e_vf       DOWN    1522

Hypervisor Details
===================
Hypervisor: VMWARE
Manufacturer: VMware, Inc
Product Name: VMware Virtual Platform
Serial Number: VMware-42 06 f0 d7 62 6a fd 6d-75 0e cc 81 5d ce ac 71
UUID: 0E3546DD-DE6E-400D-9B3D-025215519CB8
image_variant :

Boot Details
====================
Boot mode: BIOS
Bootloader version: 1.1
```

For information on the firmware for Intel NIC, see:

https://downloadcenter.intel.com/product/82947/Intel-Ethernet-Controller-X710-Series

For information on the driver for Intel and Cisco NIC, see:

https://www.vmware.com/resources/compatibility/detail.php%3FdeviceCategory%3Dio%26productid%3D37996

For information on the firmware for Cisco NIC, see:

https://www.cisco.com/c/en/us/support/servers-unified-computing/ucs-c-series-rack-servers/tsd-products-support-series-home.html

Creating a VM for Controller Using an ISO Image

The following procedure provides general guidelines about how to deploy the controller using VMware vSphere. However, the exact steps that you should perform may vary, depending on the characteristics of your VMware environment and setup.

Before you begin

Ensure that the vSphere Client is installed on your machine.

Step 1 Log in to the VMware vSphere Client.

Step 2 From the vSphere Client menu, choose **File > New > Virtual Machine**.

Step 3 From the **Create New Virtual Machine** window, select **Custom** and click **Next**.

Step 4 Enter a **Name** for the VM and click **Next**.

Step 5 Select **Datastore** for the VM files and click **Next**.

Step 6 Select the **Virtual Machine Version** and click **Next**.

Step 7 In the **Guest Operating System** window, choose **Other** and from the **Version** drop-down list, choose the version as **Other (64 -bit)**, and click **Next**.

Step 8 Under **CPUs**, select the following settings:

- **Number of virtual sockets (virtual CPUs)**
- **Number of cores per socket**

The number of cores per socket should always be set to **1**, regardless of the number of virtual sockets selected. For example, a controller with a 4-vCPU configuration should be configured as 4 sockets and 1 core per socket.

The supported number of virtual CPUs and the corresponding RAM allocation required depends on the profile you want to deploy.

Step 9 Under **Memory**, configure the supported memory size for your profile, and click **Next**.

Step 10 Under **Network**, allocate two (three if HA is required) vNICs based on the profile you want to deploy.

a) From the **How many NICs do yo want to connect?** drop-down list, select the number of vNICs that you want to connect.

b) From the Network drop-down list, select the vNICs.

(Select a different network for each vNIC.)

Note We recommend that you add two or three interfaces; one for device management, one for wireless management, and one for HA, if you want to configure HA.

c) From the **Adapter** drop-down list, select the **VMXNET3** as the adapter type.

d) Select all the vNICs to connect at power-on.

e) Click **Next**.

Step 11 In the **SCSI Controller** window, select **SCSI Controller** as **VMware Paravirtual** and click **Next**.

Step 12 In the **Create a Disk** window, select the following:

- **Capacity**: Disk Size. We recommend an 8-GB disk.
- **Disk Provisioning**: Choose one of the following: **Thick Provision Lazy Zeroed** or **Thick Provision Eager Zeroed**.
- **Location**: Store with the Virtual Machine.

Step 13 Click **Next**.

Step 14 In the **Advanced Options** window, select the **Virtual Device Node** and click **Next**.

Step 15 Click **Finish**.

Step 16 Go to the newly created instance, right-click, and select **Edit Settings**.

Step 17 Under the **Hardware** tab, click **CD/DVD Drive**.

a) Select the **Device Type** that the VM will boot from as **Datastore ISO File** option. Browse to the location of the .iso file on the datastore. Ensure that the controller ISO file is selected.

b) In the **Device Status** section, check the **Connect at power on** check box.

Step 18 Click **OK**.

The VM is now configured and is ready to boot. The controller is booted when the VM is powered on.

Powering On the Controller

To launch the controller, perform the following steps:

Step 1 Select the virtual switch from the vSphere client.

Step 2 Select the VM and click **Power On**.

The VM starts the launch process. After the VM is launched, the controller starts the boot process.

Creating the Controller Using the Self-installing .Run Package for ESXI

The Cisco Catalyst 9800 Wireless Controller ESXI Installer package is a self-installing package for ESXI.

Installation uses the bundled controller image file and one of the VM configuration options such as small (1kAPs-10kClients), medium (3kAPs-32kClients), and large (6kAPs-64kClients) described in the procedure below.

Installing the Controller Instance Using .Run Package for ESXI

Before you begin

Download the .run executable from the Cisco Catalyst 9800 Wireless Controller software installation image package and copy it onto a local device.

The following tools are required to run the package:

- OVF tool
- SSHPass

The package supports the following operating systems:

- Linux
- Mac OS

Step 1 Run the executable to launch the controller VM.

```
Device:code$/bin/bash <setup_C9800-CL_esxi>.run
```

Step 2 Select a deployment profile from the following options:

- **1kAPs-10kClients:** Deploys C9800-CL with (4 vCPUs / 8 GB RAM / 3 vNICs / 8GB disk)
- **3kAPs-32kClients:** Deploys C9800-CL with (6 vCPUs / 16 GB RAM / 3 vNICs / 8GB disk)
- **6kAPs-64kClients:** Deploys C9800-CL with (10 vCPUs / 32 GB RAM / 3 vNICs / 8GB disk)

Step 3 Select a controller instance profile or press enter to use the default profile.

```
Select the C9800-CL Instance Profile [2]: 1
```

Step 4 Enter the controller instance name or press enter to use the default name.

```
Enter the C9800-CL Instance Name [C9800-CL_574]: C9800-CL_574
```

Step 5 Confirm whether you want to install the controller instance using vCenter Server.

```
Do you want to deploy via vCenter Server [y/N]: y
```

Step 6 Enter the IPv4 address of the vCenter server.

```
Enter the IPv4 Address of the vCenter server: 10.105.203.182
```

Step 7 Enter the username of the vCenter server.

```
Enter the username of the vCenter server: administrator@vsphere.local
```

Step 8 Enter the password for the vCenter server.

```
Enter the password of the vCenter server: ******
```

Step 9 Enter the IPv4 address of the Vhost server.

```
Enter the IPv4 Address of the Vhost server: 10.104.170.96
```

Step 10 Enter the username of the Vhost server.

```
Enter the username of the Vhost server: root
```

Step 11 Enter a password for the Vhost server.

```
Enter the password of the Vhost server: *****
```

Step 12 The system displays the names of the available networks, as shown in the example below:

```
Available Network Options:
1. 9.x Network
2. Dummy
3. Mgmt Network
4. VM Network
5. TLS-Private-NW
```

a) Enter the number of the device management or service network from the list displayed above or press enter to use the default value.

```
Enter the Device Management/Service Network [1]: 2
```

b) Enter the number of the wireless management network from the list displayed above or press enter to use the default value.

```
Enter the Wireless Management Network [2]: 2
```

c) Enter the number of the High Availability (HA) network from the list displayed above or press enter to use the default value.

```
Enter the High Availability Network [3]: 2
```

Step 13 Choose the datastore option from the list of options.

```
Enter the Datastore option: 1
```

Step 14 Create an HA instance.

```
Do you want to create High Availability Instance [Y/n]: y
```

Step 15 Enter the IPv4 address of the vCenter server for HA instance.

```
Enter the vCenter Server IPv4 Address for the High Availability instance[10.105.203.182]:
10.104.169.46
```

Note If you use the same IP address again, system will not prompt for the username and password.

Step 16 Enter the username of the vCenter server.

```
Enter the vCenter High Availability server username: administrator@vsphere.local
```

Step 17 Enter the password for the vCenter server.

```
Enter the vCenter High Availability server password: ******
```

Step 18 Enter the IPv4 address of the vhost server.

```
Enter the IPv4 address of the vhost server for the High Availability instance [10.104.170.96]:
10.104.169.45
```

Note If you use the same IP address again, system will not prompt for the username and password.

Step 19 Enter the username of the HA Vhost server.

```
Enter the username of the High Availability vhost server: root
```

Step 20 Enter the password of the HA Vhost server.

```
Enter the password of the High Availability vhost server: root
```

Step 21 Enter a host name or press enter to use the default name.

```
Enter Hostname [C9800-CL_574]: CL_574
```

Step 22 Enter a password to enable the host.

```
Enter Enable Password: *****
```

Step 23 Enter the IPv4 address of the device management or service interface.

```
Enter the Device Management/Service IPv4 Interface IPv4 address: 10.104.178.21
```

Step 24 Enter the netmask of the device management or service interface.

```
Enter the Device Management/Service Interface IPv4 netmask: 255.255.255.0
```

Step 25 Enter the gateway address of the device management or service interface.

```
Enter the Device Management/Service Interface IPv4 gateway: 10.104.178.1
```

Step 26 Enter the IPv4 address of the remote network to reach the device management or service interface.

```
Enter the remote network to reach the Device Management/Service Interface: 8.0.0.0
```

Step 27 Enter the netmask of the remote network to reach the device management or service interface.

```
Enter the remote netmask to reach the Device Management/Service Interface: 255.0.0.0
```

Step 28 Enter the username to access the C9800-CL.

```
Enter the Login Username: cisco
```

Step 29 Enter the password for the login username.

```
Enter the Login Password: *****
```

Step 30 Enter the local IPv4 address of the HA interface.

```
Enter the High Availability Interface Local IPv4 address: 192.168.10.2
```

Note This option is available only if you enable HA.

Step 31 Enter the local network mask of the HA interface.

```
Enter the High Availability Interface Local IPv4 netmask: 255.255.255.0
```

Note This option is available only if you enable HA.

Step 32 Enter the peer IPv4 address of the HA interface.

```
Enter the High Availability Interface Peer IPv4 address: 192.168.10.3
```

Note This option is available only if you enable HA.

Step 33 A summary configuration similar to the one given in the following example is displayed at the end.

```
Summary Configuration:
  Deployment Profile:           : 1kAPs-10kClients [1]
      vCPUs                     : 4
      Memory[GB]                : 8
      Disk[GB]                  : 8
      vNICs                     : 3
  Instance Name                 : C9800-CL_24495
  Vcenter IPv4 Address          : 10.105.203.182
  Vhost IPv4 Address            : 10.104.170.96
  Datastore                     : datastore
  Hostname                      : C9800-CL_24495
  Login Username                : cisco
  Network Configuration
    Device Management/Service Interface
      Interface                 : GigabitEthernet1
      Network                   : Dummy
      IPv4 Address              : 10.104.23.45
      IPv4 Netmask              : 255.255.255.0
      IPv4 Gateway              : 10.104.23.1
      Remote Network Route      : 8.0.0.0
      Remote Network Netmask    : 255.0.0.0
    Wireless Management Interface
      Interface                 : GigabitEthernet2
      Network                   : Dummy
```

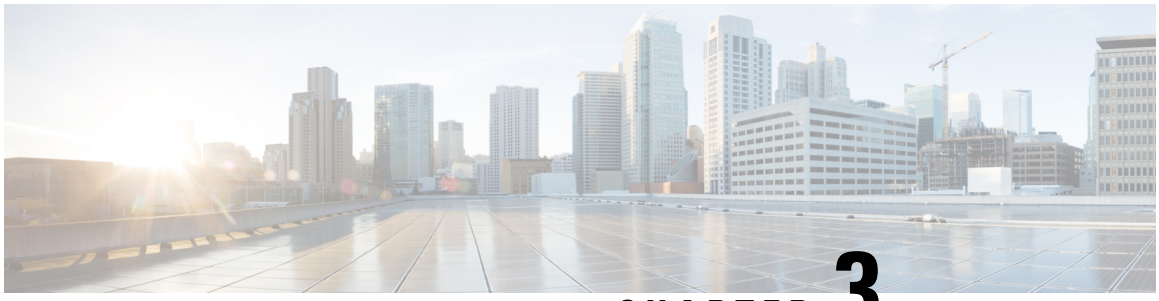
```
    High Availability Interface
      Interface                : GigabitEthernet3
      Network                  : Dummy
      Local IPv4 Address       : 192.168.3.4
      Local IPv4 Netmask       : 255.255.255.0
      Peer  IPv4 Address       : 192.168.3.5

  High Availability Instance Information
  vCenter IPv4 Address         : 10.105.203.182
  Vhost IPv4 Address           : 10.104.170.96
  Datastore                    : datastore
  Network Configuration
    Device Management/Service Interface
      Interface                : GigabitEthernet1
      Network                  : Dummy
    Wireless Management Interface
      Interface                : GigabitEthernet2
      Network                  : Dummy
    High Availability Interface:
      Interface                : GigabitEthernet3
      Network                  : Dummy
      Local IPv4 Address       : 192.168.3.5
      Local IPv4 Netmask       : 255.255.255.0
      Peer IPv4 Address        : 192.168.3.4

Do you want to create an C9800-CL instance [Y/n]: y
```

Step 34 Choose whether you want to continue with the controller instance creation or abort it.

```
Do you want to create an C9800-CL instance [Y/n]: y
```



CHAPTER 3

Installing the Controller in a KVM Environment

Overview of Kernel-Based Virtual Machine Environment

Cisco Catalyst 9800 Wireless Controller for Cloud is supported on top of Ubuntu, Red Hat Enterprise Linux (RHEL) 7.2, and Red Hat Enterprise Virtualization (RHEV) using the Kernel-Based Virtual Machine (KVM). Installation on a KVM requires the creation of a virtual machine (VM) and installation using a .iso file or a .qcow2 file. The VM can be launched using the KVM command line or Virsh.

- .qcow2—Used for booting a software image in KVM environments.
- .iso—Used to manually install the Cisco Catalyst 9800 Wireless Controller for Cloud using the Virsh tool. You must also have a virsh.xml file with a sample XML configuration to launch the controller in KVM environments using virsh commands.

Supported Profile Configurations

The supported profile configurations are:

Table 7: Supported Profile Configurations

Templates	CPUs	RAM	APs	Clients
Small	4 vCPUs	8 GB	1000	10000

Templates	CPUs	RAM	APs	Clients
Medium	6 vCPUs	16 GB	3000	320000
Large	10 vCPUs	32 GB	6000	640000

Supported Networking Options

The following are the networking options supported:

- Linux bridge
- Open vSwitch (OVS)

Required Packages for a KVM Installation

The required packages for a KVM installation are:

- Qemu-kvm
- Qemu-utils
- Uml-utilities
- Socat
- KVM
- Libvirt-bin
- Virtinst

Installation Procedure in a KVM Environment

You can install Cisco Catalyst 9800 Wireless Controller for Cloud in a KVM environment either by using the self-installing package that guides you through the installation steps or by using one of the management software supported by KVM, such as virt-manager, virt install, or virsh.

The KVM Installer package is a self-installing package for KVM. When you run this package, it provides the following modes:

- Default—Installs the controller using the bundled image file and one of the default VM configuration options (small, medium, or large).
- Interactive—Allows customization of the VM configuration and provides the option to install the bundled image file or a separate .qcow2 image.

Note For a list of unsupported VM operations, refer to *Supported VMware Features and Operations* section in Installing in a VMware ESXi Environment, on page 6 chapter.

Before you begin

Download the .run executable from the Cisco Catalyst 9800 Wireless Controller for Cloud software installation image package and copy it to a local drive of the host machine.

Installing the Controller with Linux Bridge Networking Using the .qcow2 Image

This procedure provides general guidelines for manually creating the VM for the controller; the exact steps that you should perform may vary depending on the characteristics of your KVM environment and setup. For more information, see the Red Hat Linux, Ubuntu, and Virsh documentation.

Using the **virt-install** command, create an instance and boot, using the following syntax:

```
--connect=qemu:///system \
--os-type=linux \
--os-variant=rhel4 \
--arch=x86_64 \
--cpu host \
--console pty,target_type=virtio \
--hvm \
--import \
--name=my_c9k_vm \
--disk path=<path_to_c9800-c_qcow2>,bus=ide,format=qcow2 \
--vcpus=1,sockets=1,cores=1,threads=1 \
--ram=4096 \
--network=network:<network name>,model=virtio \
--network=network:<network name>,model=virtio \
--network=network:<network name>,model=virtio  \
--noreboot \
```

Note After the installation is complete, the controller VM is shutdown. Start the controller VM using the **virsh start** command.

Installing the Controller with Vrish Using the ISO Image

This procedure provides a general guideline for manually creating the VM for the controller; the exact steps that you need to perform may vary depending on the characteristics of your KVM environment and setup. For more information, see the Red Hat Linux, Ubuntu and Virsh documentation.

Step 1 Create an 8 GB disk image in **.qcow2** format using the **qemu-img** command:

```
qemu-img create -f qcow2 c9000-c_disk.qcow2 8G
```

Step 2 Use the **virt-install** command to install the controller. This requires the correct permissions to create a new VM. The following example shows how to create a 1-vCPU VM with 4-GB of RAM, and three network interfaces.

```
virt-install \
--connect=qemu:///system \
```

```
--os-type=linux \
--os-variant=rhel4 \
--arch=x86_64 \
--cpu host \
--hvm \
--import \
--name=my_c9k_vm \
--cdrom=<path_to_c9800-c_iso> \
--disk path=c9000-c_disk.qcow2,bus=virtio,size=8,sparse=false,cache=none,format=qcow2 \
--ram=4096 \
--vcpus=1,sockets=1,cores=1,threads=1 \
--network=network:<network name>,model=virtio \
--network=network:<network name>,model=virtio \
--network=network:<network name>,model=virtio \
--noreboot \
```

Note The **virt-install** command creates a new VM instance and the controller installs the image on the specified disk file. After the installation is complete, the controller VM is shutdown. Start the controller VM using the **virsh start** command.

Installing the Controller with OVS Networking Using the .qcow2 Image

This procedure provides a general guideline for manually creating the VM for the controller; the exact steps that you need to perform may vary depending on the characteristics of your KVM environment and setup. For more information, see the Red Hat Linux, Ubuntu and Virsh documentation.

Using the **virt-install** command, create an instance and boot, using the following syntax:

```
--connect=qemu:///system \
--os-type=linux \
--os-variant=rhel4 \
--arch=x86_64 \
--cpu host \
--console pty,target_type=virtio \
--hvm \
--import \
--name=my_c9k_vm \
--cdrom=<path_to_c9800-c_iso> \
--disk path=c9000-c_disk.qcow2,bus=virtio,size=8,sparse=false,cache=none,format=qcow2 \
--ram=4096 \
--vcpus=1,sockets=1,cores=1,threads=1 \
--network=network:<network name>,model=virtio \
--network=network:<network name>,model=virtio \
--network=network:<network name>,model=virtio \
--noreboot \
```

Note After the installation is complete, the controller VM is shutdown. Start the controller VM using the **virsh start** command.

Installing the Controller with Vrish Using Bootstrap Configuration

This procedure provides a general guideline for manually creating the VM for the controller; the exact steps that you need to perform may vary depending on the characteristics of your KVM environment and setup. For more information, see the Red Hat Linux, Ubuntu and Virsh documentation.

Before you begin

Create a text file named *iosxe_config.txt* with the required configuration and create a .iso image using the following command by providing the iosxe_config.txt file as input: **mkisofs -l -o** *iso-file-name.iso iosxe_config.txt*

```
mkisofs -l -o test.iso iosxe_config.txt
```

A sample configuration file is given below:

```
hostname C9800-CL
license smart enable
username lab privilege 15 password lab
ip domain-name cisco.com
interface GigabitEthernet1
 ip address 10.0.0.5 255.255.255.0
 no shut
exit
ip route 0.0.0.0 0.0.0.0 10.0.0.1
line vty 0 4
  login local
exit
```

Use the **virt-install** command to install the controller. Use of this command requires proper privileges to create a new VM. The following example shows how to create a 1-vCPU VM with 4-GB of RAM, and three network interfaces.

```
virt-install \
--connect=qemu:///system \
--os-type=linux \
--os-variant=rhel4 \
--arch=x86_64 \
--cpu host \
--console pty,target_type=virtio \
--hvm \
--import \
--name=my_c9k_vm \
--disk path=<path_to_c9800-c_qcow2>,bus=ide,format=qcow2 \
--vcpus=1,sockets=1,cores=1,threads=1 \
--ram=4096 \
--network=network:<network name>,model=virtio \
--network=network:<network name>,model=virtio \
--network=network:<network name>,model=virtio \
--noreboot \
```

Creating Controller Instance Through VMM Using ISO Image

Step 1 Start the virt-manager using **Applications > System Tools > Virtual Machine Manager**.

You may be asked to select the hypervisor and enter your root password.

Step 2 Choose **File** option on top and select **New Virtual Machine** option.

Step 3 Enter the virtual machine details:

a) Enter a **Name** for the VM.
b) In the operating system option, select **Local install media**.
c) Click **Forward**.

Step 4 Select the **ISO image** from the disk.

Step 5 Select **Automatically Detect operating system based on install media**.

Step 6 Configure the memory and CPU options:

a) Set **Memory (RAM)** .
b) Set **CPUs**.
c) Click **Forward** to continue.

Step 7 Set disk image size as 8GB and click **Forward** .

Step 8 Enter the instance name.

Step 9 Check the **Customize configuration before install** box firs,t before you click **Finish**.

This allows you to add additional NICs.

Step 10 Select the **Network** tab to add additional NICs.

Step 11 Select the**Network** from the **Network source** drop-down.

Note Only virtio network driver is supported.

Step 12 Select the **Portgroup** using the drop-down.

Step 13 Click **Finish**.

Bootstrap Configuration with KVM VMM (virt-manager)

The virt-manager, also known as Virtual Machine Manager, is a desktop application for managing virtual machines through libvirt. It presents a summary view of running domains, their live performance and resource utilization statistics. Wizards enable the creation of new domains, and configuration and adjustment of a domain’s resource allocation and virtual hardware. An embedded VNC and SPICE client viewer presents a full graphical console to the guest domain.

Step 1 Start virt-manager **Applications > System Tools > Virtual Machine Manager**.

You may be asked to select the hypervisor and/or enter your root password.

Step 2 Select **File** option on top and click **New Virtual Machine** option.

Step 3 Enter the virtual machine details:

a) Specify a **Name**.

b) For the operating system, select **Import existing disk image**.

This method allows you to import a disk image (containing a pre-installed, bootable operating system, if you select the qcow2 image) to it.

c) Click **Forward** to continue.

Step 4 Select the controller qcow2 image path.

Step 5 Configure the memory and CPU options:

a) Set **Memory (RAM)** to *8192*.

b) Set **CPUs** to *4*.

c) Click **Forward** to continue.

Step 6 Enter the instance name.

Step 7 Check the **Customize configuration before install** box first before you click **Finish**.

This allows you to add more NICs.

Step 8 Select the **Network**.

Choose either bridge or network.

Step 9 Click **Finish**.

Step 10 Double click on the **Instance name** to edit it.

Step 11 Select ***i***to get the Instance information

Step 12 Select **Begin Installation** to start the Instance.

Step 13 Click the **Monitor** symbol to go to the Virtual Console.

Configuring SR-IOV for KVM

Recommended Software Versions for SR-IOV

Table 8: List of Supported NIC Types

NIC	Firmware	Driver Version	Host OS
Intel x710	7.10	I40e 2.10.19.82	KVM RedHat Version 7.5 and above
Ciscoized x710	7.0	I40e 2.10.19.82	KVM RedHat Version 7.5 and above

Enabling Intel VT-D

Note You need to have root permissions to perform subsequent tasks.

To enable Intel VT-D, perform the following steps:

Step 1 In the **/etc/sysconfig/grub** file and **GRUB_CMDLINX_LINUX** line, add the **intel_iommu=on** and **iommu=pt** parameters at the end.

Step 2 Regenerate the **/etc/grub2.cfg** file by executing the following command:

grub2-mkconfig -o /etc/grub2.cfg

Note In case of EFI, execute the following command:

grub2-mkconfig -o /etc/grub2-efi.cfg

Step 3 Reboot the system for the changes to take effect.

Your system is now capable of PCI device assignment.

Configuring SR-IOV Mode Virtual Functions (VFs) on the Interface

If VF is not available, configure SR-IOV VF using the following commands:

Step 1 Configure VF on the interface:

echo “no_of_vfs” > /sys/class/net/<interface_name>/device/sriov_numvfs

Sample output:

```
echo 1 > /sys/class/net/enp129s0f0/device/sriov_numvfs
```

Here, one VF is created for each port for maximum performance.

Step 2 Configure spoofcheck, trust mode, and MAC on the VF using the following commands:

```
ip link set dev enp129s0f0 vf 0 trust on
ip link set enp129s0f0 vf 0 spoofchk off
ip link set enp129s0f0 vf 0 mac 3c:fd:fe:de:cc:bc
```

Note The MAC addresses must be unique.

Step 3 Verify the settings using the following command:

ip link show *interface_name*

Sample output:

```
ip link show enp129s0f0
6: enp129s0f0: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN mode DEFAULT group default qlen
```

```
 1000
link/ether 3c:fd:fe:de:01:bc brd ff:ff:ff:ff:ff:ff
vf 0 MAC 3c:fd:fe:de:cc:bc, spoof checking off, link-state auto, trust on
```

Configuring SR-IOV Setting Persistence

SR-IOV configurations configured in the above way are not persistent across reboots. To resolve this issue, you can execute the above configuration as a service that is auto-enabled on host reboots.

Step 1 Create a bash script with the commands to be persisted. You need to write the script in **/usr/bin/sriov-config** file as follows:

```
#!/bin/sh
echo "no_of_vfs" > /sys/class/net/<interface_name>/device/sriov_numvfs
ip link set dev <interface_name> vf 0 trust on
ip link set <interface_name> vf 0 spoofchk off
ip link set <interface_name> vf 0 mac 3c:fd:fe:de:cc:bc
```

Sample output:

```
#!/bin/sh
echo 1 > /sys/class/net/enp129s0f0/device/sriov_numvfs
ip link set dev enp129s0f0 vf 0 trust on
ip link set enp129s0f0 vf 0 spoofchk off
ip link set enp129s0f0 vf 0 mac 3c:fd:fe:de:cc:bc
```

Note You need to repeat the same steps for all VFs.

Step 2 Provide execute permission for the script:

chmod 777 /usr/bin/sriov-config

Step 3 Create the system service: Define a new system service to be executed at the end of the boot. This service executes the bash script which has the required sriov commands as mentioned in **Step 1**.

Note Create a new file named **sriov.service** in **/usr/lib/systemd/system** and add the following content:

```
[Unit]
Description=SR-IOV configuration
After=rc-local.service
Before=getty.target
[Service]
Type=oneshot
ExecStart=/usr/bin/sriov-config
[Install]
WantedBy=multi-user.target
```

Note The **ExecStart=/usr/bin/sriov-config** command line executes the script.

Step 4 Enable and start the **sriov.service** using the following command:

systemctl --now enable sriov.service

Note This command starts the service immediately and ensures that the service is run every time the host reboots.

For more information on the SR-IOV configuration for KVM, see:

https://www.intel.com/content/www/us/en/embedded/products/networking/xl710-sr-iov-config-guide-gbe-linux-brief.html

Attaching the SR-IOV to the Controller

Attaching to a New Virtual Machine Using Command Line

Use the **host device** option of **virt-install** to add the PCI VF devices. Use the information from **Step 1** (Configuring SR-IOV Mode Virtual Functions (VFs) on the Interface, on page 28) and PCI BDF number to attach the devices.

Virtual Functions on Intel Corporation Ethernet Controller X710 for 10GbE SFP+. (enp129s0f0):

```
PCI BDF                                          Interface
=========                                        ==========
0000:18:06.0                           enp129s0f0
0000:18:06.1                                     enp129s0f1
```

Creating and Launching a VM

To create and launch a VM, use the following command:

```
sudo virt-install --virt-type=kvm --name ewlc_sriov_3-18 --ram 16384 --vcpus=9 --hvm
--cdrom=/home/C9800-CL-universalk9.BLD_POLARIS_DEV_LATEST_20200318_062819-serial.iso --network
 none --host-device=pci_0000_18_06_0 --host-device=pci_0000_18_06_1 --graphics vnc --disk
path=/var/lib/libvirt/images/ewlc_sriov_3-18.qcow2,size=8,bus=virtio,format=qcow2
```

You get to view the VM console using the following command:

```
virsh console ewlc_sriov_3-18
Connected to domain ewlc_sriov_3-18
Escape character is ^]
```

You can enter the following command to verify the SR-IOV drivers for the interface:

Device > enable

Device #show platform software vnic-if interface-mapping

```
Device # show platform software vnic-if interface-mapping
------------------------------------------------------------
Interface Name      Driver Name     Mac Addr
------------------------------------------------------------
GigabitEthernet2    net_i40e_vf     3cfd.fede.ccbd
GigabitEthernet1    net_i40e_vf     3cfd.fede.ccbc
------------------------------------------------------------
```

Note The MAC address mentioned above is the same as the one that is set for the VF.

You can verify the processor, memory, vNIC, hypervisor, and throughput profile details using the following command:

Device # show platform software system all

```
Device# show platform software system all
Controller Details:
==================
VM Template: medium
Throughput Profile: high
AP Scale: 3000
Client Scale: 32000
WNCD instances: 3

Processor Details
==================
Number of Processors : 9
Processor : 1 - 9
vendor_id : GenuineIntel
cpu MHz  : 2593.748
cache size : 4096 KB
Crypto Supported : Yes
model name : Intel Core Processor (Haswell, IBRS)
Memory Details
==============
Physical Memory : 16363364KB

VNIC Details
============
Name                      Mac Address     Driver Name       Status  Platform MTU
GigabitEthernet1   3cfd.fede.ccbc   net_i40e_vf        DOWN    1522
GigabitEthernet2   3cfd.fede.ccbd   net_i40e_vf        DOWN    1522

Hypervisor Details
==================
Hypervisor: KVM
Manufacturer: Red Hat
Product Name: KVM
Serial Number: Not Specified
UUID: 0E3546DD-DE6E-400D-9B3D-025215519CB8
image_variant :

Boot Details
==================
Boot mode: BIOS
Bootloader version: 1.1
```

Attaching an Interface to the Controller Using KVM VMM (virt-manager)

In the virt-manager, select **Hardware > Add Hardware** to add the PCI host device to the VM. Navigate to the NIC card and choose the VF that needs to be attached to the VM.

Once the PCI is added to the VM, you can start the VM.

Verifying SR-IOV Driver and Firmware Version

You can verify the ethernet and driver versions using the following command:

ethtool -i <interface_name>

Note You need to execute this command on the host machine.

```
[root@cpp-rhel-perf ~]# ethtool -i enp129s0f0
driver: i40e
version: 2.10.19.82
firmware-version: 7.10 0x8000646c 1.2527.0
expansion-rom-version:
bus-info: 0000:81:00.0
```

You can print the ethernet information, driver versions, and SR-IOV VF names using the following command:

```
lspci | grep -i eth
```

```
[root@cpp-rhel-perf ~]# lspci | grep -i eth
81:00.0 Ethernet controller: Intel Corporation Ethernet Controller X710 for 10GbE SFP+ (rev
 02)
81:00.1 Ethernet controller: Intel Corporation Ethernet Controller X710 for 10GbE SFP+ (rev
 02)
81:02.0 Ethernet controller: Intel Corporation Ethernet Virtual Function 700 Series (rev
02)
81:0a.0 Ethernet controller: Intel Corporation Ethernet Virtual Function 700 Series (rev
02)
```

For information on the firmware for Intel NIC, see:

https://downloadcenter.intel.com/product/82947/Intel-Ethernet-Controller-X710-Series

For information on the driver for Intel and Cisco NIC, see:

https://downloadcenter.intel.com/download/24411/Intel-Network-Adapter-Driver-for-PCIe-40-Gigabit-Ethernet-Network-Connections-Under-Linux-?product=82947

For information on the firmware for Cisco NIC, see:

https://www.cisco.com/c/en/us/support/servers-unified-computing/ucs-c-series-rack-servers/tsd-products-support-series-home.html

Creating the Controller Using the Self-installing .Run Package for KVM

The Cisco Catalyst 9800 Wireless Controller KVM Installer package is a self-installing package for KVM.

Installation uses the bundled controller image file and one of the VM configuration options such as small (1kAPs-10kClients), medium (3kAPs-32kClients), and large (6kAPs-64kClients) described in the procedure below.

Installing the Controller Instance Using .Run Package for KVM

The following steps are performed on the KVM server.

Before you begin

Download the .run executable from the Cisco Catalyst 9800 Wireless Controller software installation image package and copy it onto a local device.

The following tools are required to run the package:

- Vrish
- Qemu
- mkisofs
- SSHPass

The package supports the following operating systems:

- Linux
- Mac OS

Step 1 Run the executable to launch the controller VM.

```
Device:code$/bin/bash setup_C9800-CL.sh
```

Step 2 Select a Deployment Profile from the following options:

- **1kAPs-10kClients:** Deploys C9800-CL with (4 vCPUs / 8 GB RAM / 3 vNICs / 8GB disk)
- **3kAPs-32kClients:** Deploys C9800-CL with (6 vCPUs / 16 GB RAM / 3 vNICs / 8GB disk)
- **6kAPs-64kClients:** Deploys C9800-CL with (10 vCPUs / 32 GB RAM / 3 vNICs / 8GB disk)

Step 3 Select a controller instance profile or press enter to use the default profile.

```
Select the C9800-CL Instance Profile [2]: 1
```

Step 4 Enter the controller instance name or press enter to use the default name.

```
Enter the C9800-CL Instance Name [C9800-CL_15392]: C9800-CL_15392
```

Step 5 Enter the IPv4 address of the KVM server.

```
Enter the IPv4 Address of the KVM server[Localhost]: 10.104.170.94
```

Step 6 Enter the username of the KVM server.

```
Enter the username of the KVM server: root
```

Step 7 Enter the password of the KVM server.

```
Enter the password of the KVM server: ******
```

Step 8 Enter the image location from where the image has to be copied.

```
Enter the Image Location path [/var/lib/libvirt/images]: <location>
```

Step 9 The system displays the names of the available networks, as shown in the example below:

```
Available Networks:
1.  virbr0                          [bridge]
2.  Dummy                           [network]
```

```
3.  vm-mgmt-network                  [network]
4.  vm-service-network               [network]
```

a) Enter the number of the device management or service network from the list displayed above or press enter to use the default value.

```
Enter the Device Management/Service Network [1]: 2
```

b) Enter the number of the wireless management network from the list displayed above or press enter to use the default value.

```
Enter the Wireless Management Network [2]: 2
```

c) Enter the number of the HA network from the list displayed above or press enter to use the default value.

```
Enter the  High Availability Network [3]: 2
```

Step 10 Create a High Availability (HA) instance.

```
Do you want to create  High Availability Instance [Y/n]: y
```

Step 11 Enter the IPv4 address of the server for HA instance.

```
Enter the IPv4 address of the KVM Server for the High Availability Instance [10.104.170.94]:
10.104.177.37
```

Note If you use the same IP address again, system will not prompt for the username and password.

Step 12 Enter the username of the HA server.

```
Enter the username of the High Availability server: ubuntu-wnbu
```

Step 13 Enter the password of the HA server.

```
Enter the password of the High Availability server: ******
```

Step 14 The system displays the names of the available networks for HA instance, as shown in the example below:

```
Available Networks for High Availability Instance:
1.  br-ha                            [bridge]
2.  br10.x                           [bridge]
3.  br9.x                            [bridge]
4.  virbr0                           [bridge]
5.  default                          [network]
```

a) Enter the number of the device management or service network from the list displayed above or press enter to use the default value.

```
Enter the Device Management/Service Network for High Availability instance [1]: 2
```

b) Enter the number of the wireless management network from the list displayed above or press enter to use the default value.

```
Enter the Wireless Management Network for High Availability instance [2]: 2
```

c) Enter the number of the HA network from the list displayed above or press enter to use the default value.

```
Enter the High Availability Network for High Availability instance [3]: 2
```

Step 15 Enter a host name or press enter to use the default name.

```
Enter Hostname [C9800-CL_32501]: CL_32501
```

Step 16 Enter a password to enable the host.

```
Enter Enable Password: *****
```

Step 17 Enter the IPv4 address of the device management or service interface.

```
Enter the Device Management/Service Interface IPv4 address: 10.104.176.34
```

Step 18 Enter the network mask of the device management or service interface.

```
Enter the Device Management/Service Interface IPv4 netmask: 255.255.255.0
```

Step 19 Enter the gateway of the device management or service interface.

```
Enter the Device Management/Service Interface IPv4 gateway: 10.104.179.1
```

Step 20 Enter the IPv4 address of the remote network to reach the device management or service interface.

```
Enter the remote network to reach the Device Management/Service Interface: 8.0.0.0
```

Step 21 Enter the network mask of the remote network to reach the device management or service interface.

```
Enter the remote netmask to reach the Device Management/Service Interface: 255.0.0.0
```

Step 22 Enter the username to access the C9800-CL.

```
Enter the Login Username: cisco
```

Step 23 Enter the password for the login username.

```
Enter the Login Password: *****
```

Step 24 Enter the local IPv4 address of the HA interface.

```
Enter the High Availability Interface Local IPv4 address: 192.168.10.2
```

Step 25 Enter the local network mask of the HA interface.

```
Enter the High Availability Interface Local IPv4 netmask: 255.255.255.0
```

Step 26 Enter the peer IPv4 address of the HA interface.

```
Enter the High Availability Interface Peer IPv4 address: 192.168.10.3
```

Step 27 A summary configuration similar to the one given in the following example is displayed at the end.

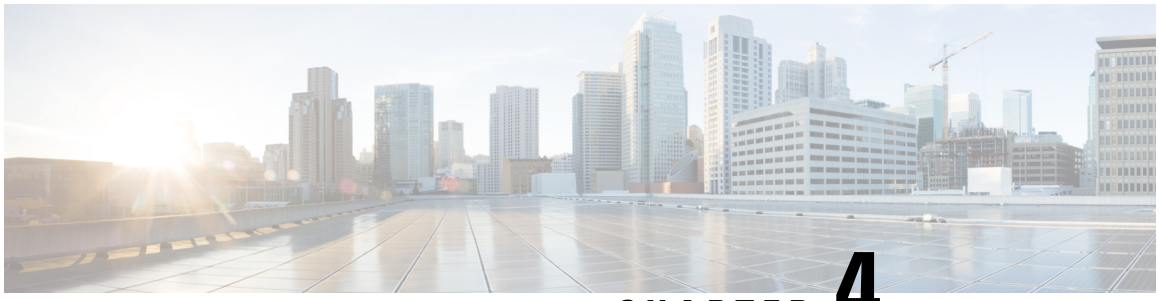
```
Summary Configuration
  Deployment Profile         : 1kAPs-10kClients [1]
      vCPU's                 : 4
      Memory[GB]             : 8
      Disk[GB]               : 8
      vNICs                  : 3
  Instance Name              : C9800-CL_7957
  Instance Image Location    : /var/lib/libvirt/images
  KVM Server IPv4 Address    : 10.104.170.94
  Hostname                   : C9800-CL_7957
  Login Username             : cisco
  Network Configuration
    Device Management/Service Interface
      Interface              : GigabitEthernet1
      Network / Bridge       : Dummy [Network]
      IPv4 Address           : 10.104.23.45
      IPv4 Netmask           : 255.255.255.0
      IPv4 Gateway           : 10.104.23.1
      Remote Network Route   : 8.0.0.0
      Remote Network Netmask : 255.0.0.0
    Wireless Management Interface
      Interface              : GigabitEthernet2
      Network / Bridge       : Dummy [Network]
    High Availability Interface
```

```
    Interface              : GigabitEthernet3
    Network / Bridge       : Dummy [Network]
    Local IPv4 Address     : 192.168.10.2
    Local IPv4 Netmask     : 255.255.255.0
    Peer  IPv4 Address     : 192.168.10.3

High Availability Instance Information
KVM Server IPv4 Address    : 10.104.177.37
Network Configuration
  Device Management/Service Interface
    Interface              : GigabitEthernet1
    Network / Bridge       : br-ha [Bridge]
  Wireless Management Interface:
    Interface              : GigabitEthernet2
    Network / Bridge       : br-ha [Bridge]
  High Availability Interface
    Interface              : GigabitEthernet3
    Network                : br-ha [Bridge]
    Local IPv4 Address     : 192.168.10.3
    Local IPv4 Netmask     : 255.255.255.0
    Peer  IPv4 Address     : 192.168.10.2
```

Step 28 Choose whether you want to continue with the controller instance creation or abort it.

```
Do you want to create an C9800-CL instance [Y/n]: y
```



CHAPTER 4

Installing the Controller in NFVIS Environment

Overview of Cisco Enterprise Network Function Virtualization Infrastructure Software

Cisco Enterprise Network Function Virtualization Infrastructure Software (Cisco Enterprise NFVIS) is a Linux-based infrastructure software designed to help service providers and enterprises dynamically deploy virtualized network functions, such as a virtual router, firewall, and WAN acceleration, on a supported Cisco device. Addition of a physical device for every network function is not required; you can use automated provisioning and centralized management.

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

The Cisco 5400 Series Enterprise Network Compute System combines routing, switching, storage, processing, and a host of other computing and networking activities into a compact 1-RU box. This high-performance unit achieves this goal by providing the infrastructure to deploy virtualized network functions and acting as a server that addresses processing, workload, and storage challenges.

Installation Procedure

VM lifecycle management refers to the entire process of registering, deploying, updating, monitoring VMs, and getting them service chained as per your requirements. You can perform these tasks using the Cisco Enterprise NFVIS portal.

Register a VM Image

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

Customizing the Setup

After registering a VM image, you can optionally create a custom profile or flavor for the VM image if the profiles defined in the image file do not match your requirement. The flavor creation option lets you provide specific profiling details for a VM image, such as the virtual CPU on which the VM will run, and the amount of virtual memory the VM will consume.

Depending on the topology requirement, you can create additional networks and bridges to attach the VM to during deployment.

Deploying a VM

A VM can be deployed using the deployment API. This API allows you to provide values to the parameters that are passed to the system during deployment. Depending on the VM you are deploying, some parameters are mandatory and others optional.

Managing and Monitoring a VM

You can monitor a VM using APIs and commands that enable you to get the VM status and debug logs. Using VM management APIs, you can start, stop, or reboot a VM, and view the statistics for a VM, such as CPU usage.

A VM can also be managed by changing or updating its profile. You can change a VM's profile to one of the existing profiles in the image file; alternatively, you can create a new custom profile for the VM. The vNICs on a VM can also be added or updated.

Uploading Image on NFVIS

Follow the procedure given below to upload an image to NFVIS:

Step 1 Select **VM Life Cycle > Image Repository**.

Step 2 Select the **Image Registration** tab and click upload arrow next to **Images**.

Step 3 Select the file from **Drop Files or Click** option.

Step 4 Click **Start** to upload the image.

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

Creating a VM Package Using Web Interface

Follow the procedure given below to create VM image using web interface:

Step 1 From ECNS, select **Image Packaging** tab and click on the create icon next to **VM Packages**.

Step 2 Enter the details in **Image Packaging** tab.

Step 3 Click **Submit**.

The bootstrap files are uploaded.

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

Step 4 Select the image you created and click on **Register**.

Creating a Network

Follow the procedure given below to create a network:

Step 1 From ECNS, select **VM Life Cycle> Networking** .

This opens up Networks & Bridges window.

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

Step 3 Enter values for **Network**, **Mode**, **Vlan**, **Bridge** and **Interface**.

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

Step 4 Click **Submit**.

This creates the network.

Deploying the Controller on NFVIS

Follow the procedure given below to deploy the controller on NFVIS:

Step 1 From ENCS, select **VM Life Cycle > Deploy**.

This opens up the VM Deployment window.

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

Note We support only 1000 APs and 10000 clients.

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

Step 4 Select the **Image** name from the drop-down.

Step 5 Select the **Profile** name from the drop-down.

Step 6 Select the **Bootstrap Config** option for providing the bootstrap configuration file before deploying the VM.

Ensure that you use the filename “iosxe_config.txt" for the bootstrap configuration file.

Step 7 Click **Deploy**.

What to do next

After deploying the VM instance, you can check the Instance details in **Manage** tab, which lists the summary of VM instances.

You can click the **Console** symbol next to the VM to get the console access.

Viewing VM Resource Allocation

Follow the procedure given below to view the VM resource allocations:

Step 1 From ECNS, select **VM Life Cycle> Resource Allocation**.

This opens up the VM CPU Allocation tab, which displays the overall CPU allocations.

Step 2 Click **VM Memory Allocation** tab.

This tab shows the overall memory allocations.

Step 3 Click **VM Disk Allocation** tab.

This tab shows the overall disk allocations.

Viewing VM Statistics

Follow the procedure given below to view the VM resource utilization:

Step 1 From ECNS, select **VM Life Cycle> VM Monitoring**.

This opens up the VM CPU Utilization tab, which displays the overall CPU utilization per VM.

Step 2 Click **Memory Allocation** tab.

This tab displays the memory utilization per VM.

Step 3 Click **VNIC Utilization** tab.

This tab displays the VNIC utilization per VM.

Step 4 Click **Disk Utilization** tab.

This tab displays the disk utilization per VM.

Creating the Controller Using the Self-installing .Run Package for Cisco Enterprise NFVIS

The Cisco Catalyst 9800 Wireless Controller Installer package (with .run extension) is a self-installing package for Cisco Enterprise NFVIS.

Installation uses the bundled controller image file and one VM configuration option: small (1kAPs-10kClients).

Installing the Controller Instance Using .Run Package on Cisco Enterprise NFVIS

Before you begin

Download the .run executable from the Cisco Catalyst 9800 Wireless Controller software installation image package and copy it onto a local device.

The following tools are required to run the package:

- curl
- SSHPass

The package supports the following operating systems:

- Linux
- Mac OS

Step 1 Run the executable to launch the controller VM.

```
Device:code$/bin/bash setup_C9800-CL_nfvis.sh
```

Step 2 Select a deployment profile:

- **1kAPs-10kClients:** Deploys C9800-CL with (4 vCPUs / 8 GB RAM / 3 vNICs / 8GB disk)

Step 3 Select a controller instance profile or press enter to use the default profile.

```
Select the C9800-CL Instance Profile [2]: 1
```

Step 4 Enter the controller instance name or press enter to use the default name.

```
Enter the C9800-CL Instance Name [C9800_CL_32001] C9800_CL_32001
```

Step 5 Enter the IPv4 address of the ENCS server.

```
Enter the IPv4 Address of the Cisco ENCS server: 10.105.203.33
```

Step 6 Enter the username of the ENCS server.

```
Enter the username of the Cisco ENCS server: admin
```

Step 7 Enter the password for the ENCS server.

```
Enter the password of the Cisco ENCS server: ******
```

Step 8 The system displays the names of the available networks, as shown in the example below:

```
Available Networks:
1  wan-net
2  lan-net
3  10-nw
4  ha-net
```

a) Enter the number of the device management or service network from the list displayed above or press enter to use the default value.

```
Enter the Device Management/Service Network [1]: 3
```

b) Enter the number of the wireless management network from the list displayed above or press enter to use the default value.

```
Enter the Wireless Management Network [2]: 2
```

c) Enter the number of the HA network from the list displayed above or press enter to use the default value.

```
Enter the High Availability interface Network [3]: 4
```

Step 9 Create an High Availability (HA) instance.

```
Do you want to create High Availability Instance [Y/n]: y
```

Step 10 Enter the IPv4 address of the Cisco ENCS server for HA instance.

```
Enter the IPv4 address of the Cisco ENCS Server for the High Availability instance [10.105.203.33]:
 10.104.176.241
```

Step 11 Enter the username of the HA ENCS server.

```
Enter the username of the High Availability Cisco ENCS server: ubuntu-wnbu
```

Step 12 Enter the password of the HA Cisco ENCS server.

```
Enter the password of the High Availability Cisco ENCS server: ******
```

Step 13 Enter a host name or press enter to use the default name.

```
Enter Hostname [C9800_CL_28062]:C9800_CL_28062
```

Step 14 Enter a password to enable the host.

```
Enter Enable Password: *****
```

Step 15 Enter the IPv4 address of the device management or service interface.

```
Enter the Device Management/Service Interface IPv4 address: 10.104.23.45
```

Step 16 Enter the netmask of the device management or service interface.

```
Enter the Device Management/Service Interface IPv4 netmask: 255.255.255.0
```

Step 17 Enter the gateway address of the device management or service interface.

```
Enter the Device Management/Service Interface IPv4 gateway: 10.104.23.1
```

Step 18 Enter the IPv4 address of the remote network to reach the device management or service interface.

```
Enter the remote network to reach the Device Management/Service Interface: 8.0.0.0
```

Step 19 Enter the netmask of the remote network to reach the device management or service interface.

```
Enter the remote netmask to reach the Device Management/Service Interface: 255.0.0.0
```

Step 20 Enter the username to access the C9800-CL.

```
Enter the Login Username: cisco
```

Step 21 Enter the password for the login username.

```
Enter the Login Password: *****
```

Step 22 Enter the local IPv4 address of the HA interface.

```
Enter the High Availability Interface Local IPv4 address: 192.168.10.2
```

Step 23 Enter the local network mask of the HA interface.

```
Enter the High Availability Interface Local IPv4 netmask: 255.255.255.0
```

Step 24 Enter the peer IPv4 address of the HA interface.

```
Enter the High Availability Interface Peer IPv4 address: 192.168.10.3
```

Step 25 A summary configuration similar to the one given in the following example is displayed at the end.

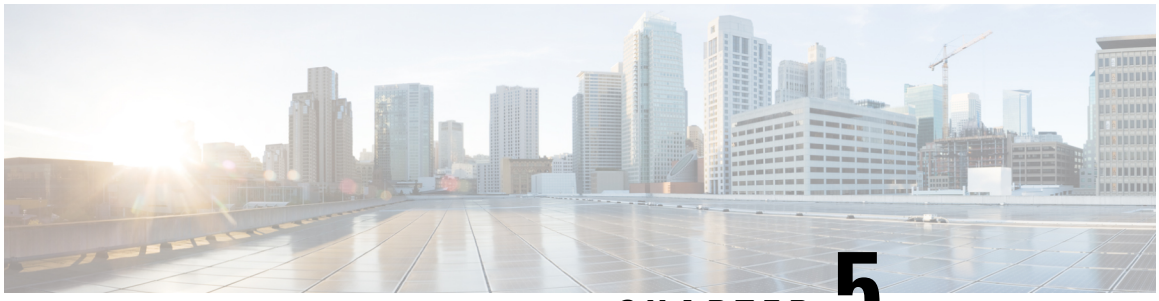
```
Summary Configuration
  Deployment Profile         :  1kAPs-10kClients [1]
      vCPU's                 :  4
      Memory[GB]             :  8
      Disk[GB]               :  8
      vNICs                  :  3
  Instance Name              :  C9800_CL_32001
  ENCS Server IPv4 Address   :  10.105.203.33
  Hostname                   :  C9800_CL_32001
  Login Username             :  cisco
  Network Configuration
    Device Management/Service Interface
      Interface              :  GigabitEthernet1
      Network                :  9nx
      IPv4 Address           :  10.104.23.45
      IPv4 Netmask           :  255.255.255.0
      IPv4 Gateway           :  10.104.23.1
      Remote Network Route   :  8.0.0.0
      Remote Network Netmask :  255.0.0.0
    Wireless Management Interface
      Interface              :  GigabitEthernet2
      Network                :  lan-net
    High Availability Interface
      Interface              :  GigabitEthernet3
      Network                :  ha-net
      Local IPv4 Address     :  192.168.10.3
      Local IPv4 Netmask     :  255.255.255.0
      Peer  IPv4 Address     :  192.168.10.2

  High Availability Instance Information
  ENCS Server IPv4 Address   : 10.105.203.33
  Network Configuration
    Device Management/Service Interface
      Interface              :  GigabitEthernet1
```

```
  Network                :  9nx
Wireless Management Interface
  Interface              :  GigabitEthernet2
  Network                :  lan-net
High Availability Interface
  Interface              :  GigabitEthernet3
  Network                :  ha-net
  Local IPv4 Address     :  192.168.10.2
  Local IPv4 Netmask     :  255.255.255.0
  Peer  IPv4 Address     :  192.168.10.3
```

Step 26 Choose whether you want to continue with the controller instance creation or abort it.

```
Do you want to create an C9800-CL instance [Y/n]: y
```



CHAPTER 5

Installing the Controller in AWS Environment

Overview on Amazon Web Services

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

Prerequisites

Before attempting to launch the controller on 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.

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 on 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.

Creating a Virtual Private Cloud

Follow the procedure given below to configure a VPC in AWS:

Before you begin

- 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.
- 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 on the Amazon side of the VPN connection. A customer gateway is a physical device or software appliance on your side of the VPN connection.

Step 1 Select a VPC configuration, using the navigation path: **AWS Console>VPC Dashboard> Launch VPC Wizard> VPC with a Private Subnet Only and Hardware VPN Access**.

Step 2 Enter details at the **VPC with a Private Subnet Only and Hardware VPN Access** window.

Step 3 Create a subnet, using the navigation path: **VPC Console> Subnets> Create Subnet**

Step 4 Create a security group, using the navigation path: **VPC Console> Security Groups> Create 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 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 Virtual Private Gateway

Follow the procedure given below to create an AWS Virtual Private Gateway:

Before you begin

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

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

a) Enter a Name Tag.

Use the AWS VPN router name.

b) Choose an ASN.

You can either use a custom ASN or use the default one selected by 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 on **Actions** button, choose **Attach to VPC**.

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

Attaches the AWS VPN to the VPC.

Creating a Customer Gateway

Follow the procedure given below to create a customer gateway:

Step 1 From the AWS console, go to the **VPC** dashboard.

Step 2 Click **VPN Connections> Customer Gateways**.

Step 3 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 4 Click **Create Customer Gateway**.

Creating a VPN Connection

Follow the procedure given below to create a customer gateway:

Step 1 From the AWS console, go to the **VPC** dashboard.

Step 2 Click **VPN Connections> VPN Connections**.

Step 3 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 4 (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 5 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 6 While the VPN is being created, you can download the configuration to deploy in the customer VPN router. Click **Download Configuration**.

Step 7 From the pop-up window, select the brand and type of customer VPN router.

Step 8 Click **Download**.

Creating a Key Pair

Follow the procedure given below to create a customer gateway:

Step 1 From the AWS console, go to the **EC2** dashboard.

Step 2 Click **Network & Security > Key pairs** .

Step 3 Click **Create Key Pair**.

Installing the Controller on AWS Using 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 AWS marketplace and save it on your computer.

Step 1 From the AWS console, go to the **CloudFormation** page.

Step 2 Click **Create Stack** .

Step 3 From **Choose a template** section, select upload template to Amazon S3 option.

This loads the *json* file directly to AWS.

Step 4 Click **Next**.

This opens the **Specify Details** page.

Step 5 Enter the **Stack** and **Instance Details**.

Enter any name for the stack you want. Hostname is the controller name. Instance Key Pair is the name of the keypair. AMI id is the AMI for the EC2 instance.

Step 6 Click **Next**.

This opens the **Network Details** page.

Step 7 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 8 Click **Next**.

Wait for the status to go from "CREATE_IN_PROGRESS" to "CREATE_COMPLETE".

Step 9 Select the **Instance Type**.

Step 10 Go to **EC2** dashboard, click **Running Instances**.

The new instance will be in Status Checks (System Status Checks & Instance Status Checks) initializing. 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 using SSH using the defined credentials or using the .pem file.

Installing the Controller Using AWS Console

Follow the procedure given below to install controller with AWS console:

Step 1 From the AWS console, go to the **EC2 Management** page.

Step 2 Click **Launch Instance** .

Step 3 Click **My AMIs** to select the Cisco Catalyst 9800 Wireless Controller for Cloud AMI.

Step 4 Choose an **Instance Type**.

We recommended that you choose the instances as per your requirements.

Step 5 Configure **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 public IP during bring-up.

Step 6 Go to **Add Storage** page.

You can use this optional step to specify additional volumes to be attached to the instance.

Step 7 Go to **Add Tags** page.

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

Step 8 Go to **Configure Security Group**. Choose a security group. If a relevant one does not exist, create a new one.

Step 9 Click **Review and Launch** . Review the configuration of your instance.

Step 10 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 on your terminal:

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

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

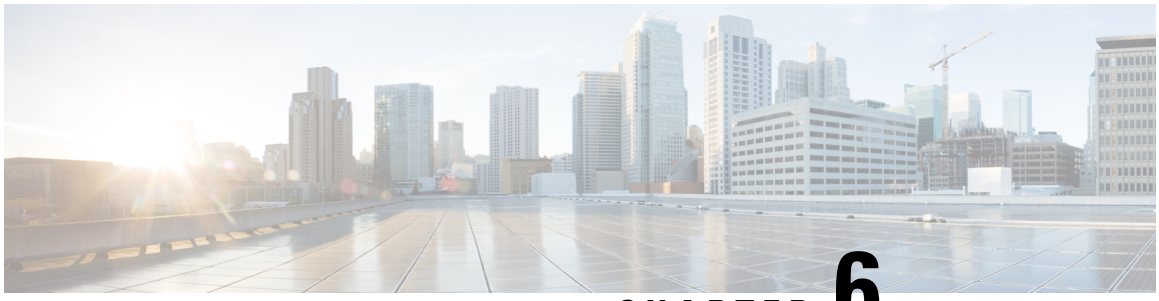
Bootstrap Properties for AWS

Table 9: 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"`

Property	Description
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 &quot;\(Tunnel1\) is down: BFD peer down notified&quot;"`



CHAPTER 6

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.

GPC 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

Follow the procedure given below 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

Follow the procedure given below to create a customer gateway:

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

Follow the procedure given below to create a customer gateway:

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.

To enable traffic to pass to a VM instance, 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

Use the following procedure to deploy a controller instance on 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.

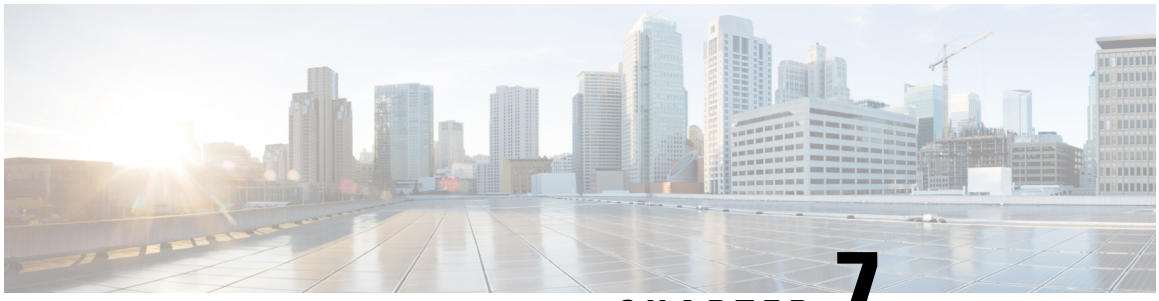
Accessing Controller Instance on GCP

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

Follow the procedure given below to access the controller on 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
```



CHAPTER 7

Installing the Controller in Microsoft Hyper-V Hypervisor

Microsoft Hyper-V Support Information

The Catalyst 9800-CL Cloud Wireless Controller installation on Microsoft Hyper-V requires the manual creation of a VM and installation, using the .iso file.

The following Microsoft Hyper-V features are supported:

- Snapshot
- Export
- Hyper-V Replica

For more information about Microsoft Hyper-V, see the Microsoft documentation.

Note While running Microsoft Hyper-V VM, you may get the following traceback log continuously in the console:

```
 "PLATFORM_INFRA-5-IOS_INTR_OVER_LIMIT_HIGH_STIME: IOS thread blocked due to SYSTEM LEVEL
ISSUE"
```

To avoid this issue, perform the following steps:

1. Configure the controller in serial mode, using the commands given below.

```
Device# configure terminal
Device(config)# platform console serial
Device(config)# end
Device# reload
```

2. Run the following command:

```
PS C:\> Set-VMComPort TestVM 1 \\.\pipe\TestPipe
```

3. Use Putty to access the console.

Installation Requirements for Microsoft Hyper-V

Before installing the controller on a Microsoft Hyper-V VM, the following must be installed on the host:

- Hyper-V Manager
- Failover Cluster Manager
- Virtual Switch

Note We recommended that you create the Virtual Switch prior to creating the VM.

The hardware profiles and the recommended resources are listed in the table given below:

Table 10: Hardware Requirements

Settings	Small	Medium	Large
Minimum Number of vCPUs	4	6	10
Minimum Memory	8 GB	16 GB	32 GB
Required Storage	16 GB	16 GB	16 GB
Minimum Number of vNICs	2	2	2
Maximum Access Points	1000	3000	6000
Maximum Clients Support	10,000	32,000	64,000

Creating the VM

To create the VM, perform the following steps:

Note You can install the controller on Microsoft Hyper-V using Microsoft Hyper-V Manager or Microsoft System Center VMM.

Step 1 In Hyper-V Manager, click on the host.

Step 2 Select **New > Virtual Machine**.

Step 3 Click **Specify Name and Location**.

- Enter the name of the VM.
- (Optional) Click the checkbox to store the VM in a different location.

Step 4 Click **Next**.

Step 5 On the **Specify Generation** screen, specify the generation of the machine to be loaded.

Note The choice of Generation 1 or Generation 2 depends on your requirements. Generation 2 supports advance features like boot from Small Computer System Interface (SCSI), secure boot, higher hardware limits, Unified Extensible Firmware Interface (UEFI) BIOS, GUID Partition Table (GPT) partitioning, and so on. If Generation 2 is selected, unselect the **Enable Secure Boot** checkbox after the deployment, as the controller does not support secure boot.

Step 6 On the **Assign Memory** screen, enter the **Startup Memory** value.

The controller requires 8196 MB for the startup memory.

Step 7 Click **Next**.

Step 8 On the **Configure Networking** screen, select a network connection to the virtual switch that was previously created.

The network adapter selected in this step will become the first interface for the controller when the VM is launched and the router boots. The other vNICs for the VM are created in the next procedure.

Step 9 Click **Next**.

Step 10 On the **Connect Virtual Hard Disk Screen**, select the following option:

- Attach a virtual hard disk later.

Note The New Virtual Machine Wizard only supports creating a virtual hard disk using the .vhdx format. The controller requires that the hard disk uses the .vhd format. Create the virtual hard disk after the VM has been created.

Step 11 Click **Next**. The **Summary** screen is displayed.

Step 12 Review the VM settings, and if it is looking good, click **Finish**.

The new VM is created.

Configuring the VM Settings

To configure the VM settings before launching the VM, perform the following steps:

Before you begin

Before launching the instance, add the network adapters (as required), disk, and load the .iso image in to the disk drive.

We recommended that you create and use separate network interfaces for Management, Wireless Management and High Availability. In case of HA deployments, create 3 network interfaces and attach the VM to the appropriate networks. For non-HA deployments, create 2 network interfaces.

The creation of management, wireless management and HA networks should be done before launching VM. The IP addressing on these interfaces could be either static or DHCP and should be configured as part of the bootstrap configuration.

The order in which the networks are attached to the interface is important as the first network attached is used for Management, second for Wireless Management (unless configured explicitly) and third for the HA.

Step 1 In Hyper-V Manager, select the host, and then right-click on the VM that was created in the previous steps.

Step 2 Select **Settings**.

Step 3 Specify the number of virtual processors, also known as virtual CPUs (vCPUs) for the VM.

Step 4 Under IDE Controller 0, select the Hard Drive.

Click the **Virtual Hard Disk** checkbox and click **New** to create a new virtual hard disk.

The New Virtual Hard Disk Wizard opens. Click **Next**.

a) On the Choose Disk Format screen, click the VHD checkbox to create the virtual hard disk using the .vhd format. Click **Next**.
b) On the **Choose Disk Type** screen, click on the **Fixed Size** option. Click **Next**.
c) Specify the Name and Location for the virtual hard disk. Click **Next**.
d) On the **Configure Disk** screen, click the option to create a new blank virtual hard disk. For the size, specify 16 GB.
e) Click **Next** to view the Summary of the virtual hard disk settings.
f) Click **Finish** to create the new virtual hard disk.

When the new hard disk has been created, continue configuring the VM settings with the next step.

Step 5 Under IDE Controller1, select the **DVD Drive**.

The DVD Drive screen is displayed.

For the **Media** setting, click the **Image File** checkbox, and browse to the .iso file that you downloaded from Cisco.com.

Step 6 Click **Ok**.

Step 7 Select **Network Adapter** to verify that the network connection to the virtual switch is configured.

Step 8 Select **Com 1** to configure the serial port.

This port provides access to the controller console.

Step 9 Select **Hardware > Add Hardware** to add the network interfaces (vNICs) to the VM.

a) Select **Network Adapter** and click **Add**.

Microsoft Hyper-V adds the network adapter and highlights that hardware with the status Virtual Switch "Not Connected".

b) Select a virtual switch on the drop-down menu to place the network adapter onto it.

Repeat these steps for each vNIC. The controller supports only the HV NETVSC vNIC type. The maximum number of vNICs supported is 8.

Note The hot-add of vNICs is not supported with Microsoft Hyper-V, so the network interfaces need to be added before launching the VM.

After the controller boots, you can verify the vNICs and how they are mapped to the interfaces using the **show platform software vnic-if interface-mapping** command.

Step 10 Click **BIOS** to verify the boot sequence for the VM.

The VM should be set to boot from the CD.

Launching the VM to Boot the Controller

To launch the VM, perform the following steps:

Step 1 Select the virtual switch.

Step 2 Select the VM and click **Start**.

The Hyper-V Manager connects to the VM, and starts the launch process. Once the VM is launched, the controller starts the boot process

Configuring Tagged Ports

The tagged port configuration is done on the host OS. By default, the VLAN tagged packets are dropped at the host OS at the vNIC. To allow these packets through to the controller, set the specific vNIC on the controller as tagged.

Note If you use GUI to create network interfaces, you cannot specify interface names and all the interfaces will be named as Network Adapter. So, using these commands, all the network adapters in the controller can be converted to tagged.

These commands are to be entered in a Power Shell.

Step 1 To see the list of adapters and assignment, use the following script:

```
Get-VMNetworkAdapter -VMName <C9800-name>
```

Note To rename the adapter name, use the following command:

```
Rename-VMNetworkAdapter -VMName <C9800-name> -Name '<C9800-adapter-name>' -NewName 'Eth1'
```

Here, **Eth1** is the adapter name.

Step 2 To configure Ethernet1 (data port/management) as Trunk, with Native VLAN id as 0, use the following script:

```
Set-VMNetworkAdapterVlan -VMName "C9800" -VMNetworkAdapterName Eth1 -Trunk -AllowedVlanIdList "1-4000"
 -NativeVlanId 0
```

Step 3 To configure Ethernet0 (serial port) as access or untagged, use the following script:

```
Set-VMNetworkAdapterVlan -VMName "C9800" -VMNetworkAdapterName Eth0 -Untagged
```

Step 4 Enable MAC address spoofing to allow the trunk port to pass the tagged traffic.

To enable MAC address spoofing, perform the following:

a. Select the virtual machine and select **Actions > Settings**.

b. Expand **Network Adapter** and select **Advanced Features**.

c. Select **Enable MAC Address spoofing**.

Creating a Bootstrap Day0 Configuration

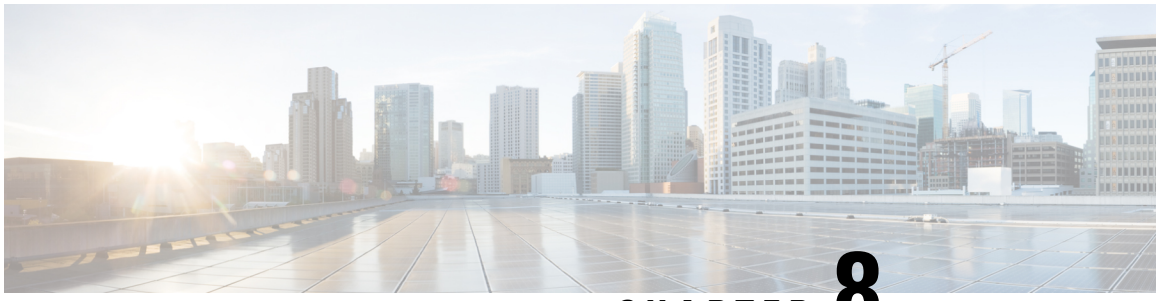
The following steps are performed on the Linux server.

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

Step 2 Create a disk image from this file using the command:

```
mkisofs -l -o ./c9800_config.iso <configuration_filename>
```

Step 3 Mount the c9800_config.iso as an additional disk during creation of the virtual machine and power on the VM.



CHAPTER 8

Booting the Controller and Accessing the Console

Day 0 WebUI Wizard for Public Cloud

Follow the procedure given below to create a Day 0 configuration and push it to the controller:

Step 1 In the address bar of a web browser, enter the **IP address** of the controller.

Step 2 Enter the **Username** and **Password**.

This displays the **Configuration Setup Wizard** window.

Enter the details in **General Settings** window.

a) Select the **Deployment Mode**.
b) Select the **Country**.
c) Select the **Date**.
d) Enter the **Time** or select the **Timezone** using the drop down list.
e) Enter the **NTP Servers** name.
f) Enter the **AAA Servers** name.

Step 3 Enter the **Wireless Management Settings**:

a) Choose **Port Number**.
b) Choose **IP Address**.

Step 4 Click **Next**.

Step 5 Enter the **Wireless Network Settings**:

a) Enter a **Network Name**.
b) Select the **Network Type**.
c) Select the **Security** option using the drop-down.
d) Enter the **Pre-Shared Key**.

e) Click **Add**.

Note Enter three wireless network settings, one for wireless management, another for device management and one more for guest management.

Step 6 Click **Next**.

This opens the Advanced Settings page.

Step 7 Enter the details in **Advanced Settings** page.

a) Select the **Client Density** using the slider.
b) Enter the **RF Group Name**.
c) Use the drop-down to select **Traffic Type**.
d) Enter the **Virtual IP Address**.
e) Use the **Generate Certificate** slider to generate certificates for APs.

This certificate is required for APs to join the controller.

f) Use the drop-down to select **RSA Key-Size**.
g) Enter the **Signature Algorithm**.
h) Enter the **Password**.
i) Review the details in **Summary** page.

Step 8 Click **Finish**.

Step 9 Click **Yes**.

This creates the configuration and pushes it to the controller.

Day 0 WebUI Wizard for Private Cloud

Follow the procedure given below to create a Day 0 configuration and push it to the controller:

Step 1 n the address bar of a web browser, enter the **IP address** of the controller.

Step 2 Enter the **Username** and **Password**.

This displays the **Configuration Setup Wizard** window. Enter the details in the **General Settings** window.

a) Select the **Deployment Mode**.
b) Select the **Country**.
c) Select the **Date**.
d) Enter the **Time** or select the **Timezone** using the drop down list.
e) Enter the **NTP Servers** name.
f) Enter the **AAA Servers** name.

Step 3 Enter the **Service Port Settings**:

a) Choose **DHCP**.
b) Enter the **Static IP** address.
c) Enter the **Subnet Mask**.

Step 4 Enter the **Static Route Settings (Optional)**:

a) Enter the **IP Address**.
b) Enter the **Subnet Mask**.
c) Enter the **Gateway** address.

Step 5 Enter the **Wireless Management Settings**:

a) Choose **Port Number**.
b) Enter the **VLAN**.
c) Choose **IPv4** or **IPv6**.
d) Enter the **Wireless Management IP** address.
e) Enter the **Subnet Mask**.
f) Enter the **Management VLAN DHCP Server**.

Step 6 Click **Next**.

This opens the Wireless Network Settings page.

Step 7 Enter the **Wireless Network Settings**:

a) Enter a **Network Name**.
b) Select the **Network Type**.
c) Select the **Security** option using the drop-down.
d) Enter the **Pre-Shared Key**.
e) Click **Add**.

Note Enter three wireless network settings, one for wireless management, another for device management and one more for guest management.

Step 8 Click **Next**.

This opens the Advanced Settings page.

Step 9 Enter the details in **Advanced Settings** page.

a) Select the **Client Density** using the slider.
b) Enter the **RF Group Name**.
c) Use the drop-down to select **Traffic Type**.
d) Enter the **Virtual IP Address**.
e) Enter the **Local IP**, **Subnet Mask**, **Remote IP** for High Availability.

Note Available only when the deployment mode is set to ACTIVE.

f) Use the **Generate Certificate** slider to generate certificates for APs.

This certificate is required for APs to join the controller.

g) Use the drop-down to select **RSA Key-Size**.
h) Enter the **Signature Algorithm**.
i) Enter the **AP password**.
j) Review the details in **Summary** page.

Step 10 Click **Finish**.

Step 11 Click **Yes**.

This creates the configuration and pushes it to the controller.

Booting the Controller

The controller boots when the VM is powered on. Depending on your configuration, you can monitor the installation process on the virtual VGA console.

Follow the procedure given below to boot up the controller:

1. Power up the VM. Within 5 seconds of powering up the VM, choose a console described from Step 2 to Step 4 to view the device's bootup and to access the controller CLI.
2. (Optional) Click **Auto Console** to use automatic console detection. This is the default setting, and the controller will boot using automatic console detection if another option is not selected within 5 seconds.
3. (Optional) Click **Virtual Console** to use the virtual VGA console. If you choose to use the virtual console, the rest of the steps in this procedure do not apply. The controller starts the boot process.
4. Use one of the following commands to Telnet to the VM:
   - **telnet://***host-ipaddress:portnumber*
   - **telnet***host-ipaddress portnumber* (from a UNIX xTerm terminal)
5. After booting, the system displays the main software image and the Golden image, with an instruction that the highlighted entry is booted automatically in 3 seconds. Do not select the option for the Golden image, and allow the main software image to boot.

Note While doing backup restore of configs, make sure you do not have **platform console serial**, as it could make the controller boot into grub mode and recovery is not possible.

Accessing the Controller Through the Virtual VGA Console

You will be prompted for wireless configuration after the Day 0 banner.

For information on modifying the configuration after you create it, see the Cisco Catalyst 9800 Series Wireless Controller Software Configuration Guide and the Cisco Catalyst 9800 Series Wireless Controller Command Reference Guide.

This section covers the following:

- Configuring the device management interface.
- Configuring the device management IP.
- [Optional] Setting a static route.
- Configuring the management credentials.

- Configuring the wireless management interface.
- Choosing the deployment mode.
- Configuring the system name or hostname.
- Configuring credentials for management access on access points.
- Configuring the country code.
- Configuring the time using an NTP server or manually.
- [Optional] Configuring a time zone.
- [Optional] Configuring the wireless client density.
- [Optional] Configuring AAA servers.
- [Optional] Configuring the wireless network settings.
- [Optional] Configuring a network name or SSID.
- [Optional] Configuring a virtual IP.
- [Optional] Configuring an RF network name.
- [Optional] Configuring a self-signed certificate.
- [Optional] Configuring high avalability.

Note Presently, there is no direct method to get back to your previous configuration. Press **Ctrl-C** to restart the configuration and return to the setup without saving the configuration.

Day 0 CLI Wizard for the Controller

Step 1 You can get into the Day 0 setup wizard using the **write erase** command or directly on the Day 0 device.

Step 2 Device management interface setup configures the device management or service port. This interface enables the basic configuration to access the device using the GUI. This is an optional configuration where you can opt to configure only the wireless management interface and not the device management.

```
Configure device management interface?[yes]:
```

Note There is no dedicated device management port for Cisco Catalyst 9800-CL Cloud Wireless Controller. So, you are prompted to select one of the options from the given range.

```
Select interface to be used for device management
1. GigabitEthernet1 [Up]
2. GigabitEthernet2 [Up]
3. GigabitEthernet3 [Up]
Choose the interface to config [1]:
```

Step 3 Device management IP helps access the device using the GUI.

```
Configure static IP address? [yes]:
   Enter the interface IP [GigabitEthernet1]: 192.168.1.10
   Enter the subnet mask [GigabitEthernet1] [255.0.0.0]: 255.255.255.0
```

Step 4 [Optional] Setting a static route to access the device using the GUI.

```
Configure static route? [yes]:
Enter the destination prefix: 192.168.1.0
Enter the destination mask: 255.255.255.0
Enter the forwarding router IP: 192.168.1.1
```

Step 5 Enter the management username and password. This is a mandatory step.

```
Enter the management username: cisco
  Enter the password: ********
  Reenter the password: ********
```

Step 6 Configure the wireless management if you haven't configured a device management interface.

```
Basic management setup is now complete. At this point, it is possible to save the above and
continue wireless setup using the webUI (for this, choose 'no' below)

Would you like to continue with the wireless setup? [yes]: yes
```

Note This prompt is not applicable for 17.4 release.

Note If you have not configured the device management, the setup moves to **Step 7** before displaying the above banner.

In 17.3 release, you will be allowed to exit the wizard after configuring at least one of the interfaces, that is, device or wireless management.

This banner is no longer available in 17.4. You cannot exit the wizard without completing the configuration.

If you select **Yes**, you need to follow the upcoming steps. Also, you can access the device using the IP configured in **Step 4**.

Step 7 Wireless management interface is a mandatory configuration:

```
Configuring wireless management interface
  Select interface to be used for wireless management
   1. GigabitEthernet2 [Up]
   2. GigabitEthernet3 [Up]
  Choose the interface to config [1]:
```

Note If GigabitEthernet1 is used for device management interface then the remaining GigabitEthernet interfaces will be displayed.

Step 8 Enter a VLAN ID:

```
Enter the vlan ID (1-4094): 112
```

Step 9 Configure an IPv4 or IPv6 address:

```
  Configure IPv4 address? [yes]:
    Enter the interface IP [GigabitEthernet1]: 9.11.112.40
```

```
    Enter the subnet mask [GigabitEthernet1] [255.0.0.0]: 255.255.255.0
  Configure IPv6 address? [yes]: no
```

Step 10 Configure a VLAN DHCP server and IP address:

```
 Do you want to configure a VLAN DHCP Server? [yes]: yes
   Enter the VLAN DHCP Server IP [GigabitEthernet1]: 9.11.112.45
```

Step 11 [Optional] Setting a static route to attach an AP client to the controller. The default options for static route prompts you to configure a default route. However, you can specify a different route as well.

```
 Configure static route? [yes/no]: yes
  Enter the destination prefix [0.0.0.0]:
  Enter the destination mask [0.0.0.0]:
  Enter the forwarding router IP: 9.11.112.1
```

Note If you configure the device as HA RMI and you haven't configured a default route (that is, source and destination as 0.0.0.0), the wizard asks for the default route information.

```
Basic management setup is now complete. At this point, it is possible to save the above and
continue wireless setup using the webUI(for this, choose 'no' below)

Would you like to continue with the wireless setup? [yes]
```

Step 12 Choose the deployment mode:

```
 Choose the deployment mode
    1. Standalone
    2. Active
    3. Standby
 Enter your selection [1]:
```

Note You can choose from one of the following deployment modes:

- **Standalone**: In this mode, you do not get to view any high availability pairing information.
- **Active**: In this mode, the controller needs to be configured with all the Day 0 information.
- **Standby**: In this mode, the configuration proceeds to the **High Availability** configuration.

Step 13 Configure the system name or hostname:

```
Enter the hostname [WLC]: ciscowlc
```

Note This is a mandatory step. The hostname needs to confirm to the RFC standards.

Step 14 [Optional] Configure the login credentials for an AP.

```
Configure credentials for management access on Access Points? [yes]:
  Enter the management username: cisco
  Enter the management password: ****
    Reenter the password: ****
  Enter the privileged mode access password: ****
    Reenter the password: ****
```

Step 15 Configure the country code. You can specify multiple country codes by separating them with a comma.

```
Configure country code for wireless operation in ISO format ? [US]:
```

Step 16 Configure the date and NTP to allow access points to join the controller. You can configure time using an NTP server or manually.

Note Enter the date in the following format:

MM/DD/YYYY

```
Configure a NTP server now ? [yes]: no

Configure the system time now? [yes]: yes

Enter the date in MM/DD/YYYY format: 10/05/2021
Enter the time in HH:MM:SS format: 10:22:13
```

Step 17 [Optional] Configure a timezone:

```
 Configure timezone? [yes]:
  Enter name of timezone: ind
  Enter hours offset from UTC (-23,23): 5
  Enter mins offset from UTC (0,59) [0]: 30
```

Step 18 [Optional] Configure the expected client density:

```
 Configure Wireless client density? [yes]:
  Choose the client density
    1. Low
    2. Typical
    3. High
  Enter your selection [2]: 3
```

Step 19 [Optional] Configure AAA servers:

Note You can configure a maximum of 6 servers during Day 0 configuration.

```
 Configure AAA servers? [yes]:
  Enter the AAA server address: 9.11.112.46
  Enter the AAA key: ***
Do you want to add more AAA servers? [yes]:
  Enter the AAA server address: 9.11.112.47
  Enter the AAA key: ***
Do you want to add more AAA servers? [yes]: no
```

Note The AAA servers are required for WPA2 Enterprise. In 17.4 release, you need to configure AAA only in one place. If you follow **Step 21**, WPA2 Enterprise will not ask for AAA servers in **Step 22**.

Step 20 [Optional] Configure wireless network settings to configure WLAN information for an AP and client join:

```
Configure Wireless network settings? [yes]:
```

Step 21 [Optional] Configure an SSID for client join:

```
Enter the network name or service set identifier (SSID):
Choose the network type
    1. Employee
    2. Guest
```

If you choose **Employee** as the network type, the following options are displayed:

```
Choose the security type
    1. WPA Personal
    2. WPA Enterprise
   Enter your selection [2]:
```

If you choose **WPA2 Personal**, you will need to enter a pre-shared key (ASCII).

```
Enter the pre-shared key (ASCII):
```

If you choose **WPA2 Enterprise**, you will be able to add multiple AAA servers.

```
Enter the AAA server address:
Enter the AAA key:
Enter more AAA server details? [yes]
```

If you choose **Guest**, you get to view the following options:

```
Please choose the security type:
1. Webauth
2. Authbypass
3. Consent
4. Webconsent
Enter the security type:
```

Step 22 [Optional] Configure a virtual IP address. The default virtual IP address is 192.0.6.1.

```
 Configure virtual IP? [yes]:
  Enter the virtual IP [192.0.6.1]:
```

Step 23 [Optional] Configure an RF network name.

```
 Configure RF-Network Name? [yes]:
  Enter the RF-Network Name: ciscorf
```

Step 24 [Optional] Configure a self-signed certificate.

```
Auto generate certificate for AP join? [yes]:
Choose key size
1.2048
2.3072
3.4096
Enter your selection [1]:
Choose the signature algorithm
1.SHA256
2.SHA384
Enter your selection [1]:
Enter secret key(minimum 8 characters): ********
Self Signed Certificate generation will be done after system boots up.
```

Step 25 [Optional] Configure high availability.

If you choose the deployment mode as Active or Standby, you will need to choose from one of the HA pairing type:

a. RMI

b. RP-RP

Note For information on HA pairing types, see **Part: High Availability (High Availability > Information About Redundancy Management Interface)** in *Cisco Catalyst 9800 Series Wireless Controller Software Configuration Guide, Cisco IOS XE Bengaluru 17.4.x*.

```
High Availability configuration
 Please choose the HA pairing type
   1. RMI
   2. RP-RP
 Enter your selection [1]:
```

If you choose RMI+RP, you need to select an interface to be used as redundancy port:

```
Select interface to be used as redundancy port
  1. GigabitEthernet3 [Up]
 Choose the interface to config [1]: 2
 Enter the RMI IP for local chassis: 9.11.112.50
 Enter the RMI IP for remote chassis: 9.11.112.51
 Enter the gateway IP of the last resort: 9.11.112.1
```

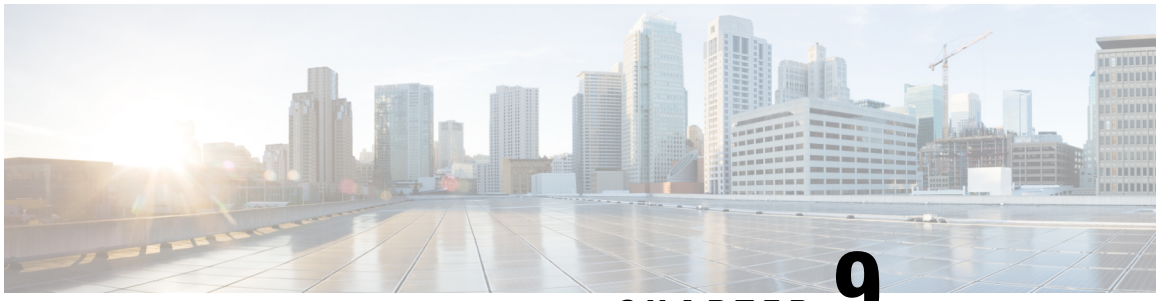
If you choose the deployment mode as Standby, you need to specify the VLAN ID for completing the pairing:

```
 Enter the RMI IP for local chassis: 9.11.112.51
 Enter the RMI IP for remote chassis: 9.11.112.50
 Enter the wireless management VLAN: 112
```

If you choose RP, you need to select an interface to be used as redundancy port:

```
Select interface to be used as redundancy port
  1. GigabitEthernet3 [Up]
 Choose the interface to config [1]: 2
 Enter the local IP:
Enter the subnet mask:
Enter the remote IP:
```

Note It is recommended to use GigabitEthernet1 for device managemt interface, GigabitEthernet2 for wireless management interface, and GigabitEthernet3 for HA.



CHAPTER 9

Upgrading the Software

Prerequisites for the Software Upgrade Process

This section describes how to upgrade the Cisco IOS XE software for an existing controller installation on a VM.

Note

- This procedure provides details about upgrading to a new software version of the controller on the same VM.
- We recommend that you use Web UI method for a faster upgrade process.

Be sure to complete the following prerequisites before upgrading the Cisco IOS XE version of the controller software image:

- Compatibility with the hypervisor vendor and version being used. If you want to upgrade to a new hypervisor version that is not supported by your current version of controller, you need to upgrade the version of controller before upgrading to the new hypervisor version.
- Memory requirements of the VM for the controller software image:
  - If the new controller version requires more memory than your previous version, you must increase the memory allocation on the VM before starting the upgrade process.
  - You must use the **.bin** file to upgrade or downgrade your software. Use the **.iso** and **.ova** files for first-time installation only.

Upgrading the Controller Software (CLI)

Follow these instructions to upgrade from one release to another, in install mode.

Before you begin

- Clean up the old installation files using the **install remove inactive** command.
- For upgrading the software using CLI, we recommend that you use install mode for the upgrade. Use the **show version** command to verify the boot mode.
- To perform a software image upgrade, you must be booted into IOS through **boot flash:packages.conf**.
- Ensure that boot parameter is set to boot only from *flash:packages.conf*.

Step 1 Go to the software download page: https://software.cisco.com/download/home/286316412/type

a) Click IOS XE Software link.

b) Select the release number you want to install.

Note Cisco recommended release is selected by default. For information on release designations, see this link: https://software.cisco.com/download/static/assets/i18n/reldesignation.html?context=sds

c) Click **download**.

Step 2 Copy the new image to flash by running the following command: **copy tftp:***image* **flash:**

Note Transferring large files over TFTP is a time-consuming process

```
Device# copy tftp://10.8.0.6//C9800-universalk9_wlc.xx.xx.xx.SPA.bin flash:

Destination filename [C9800-universalk9_wlc..xx.xx.xx..SPA.bin]?
Accessing tftp://10.8.0.6//C9800-universalk9_wlc.xx.xx.xx.SPA.bin...
Loading /C9800-universalk9_wlc.xx.xx.xx.SPA.bin from 10.8.0.6 (via GigabitEthernet0/0):
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
[OK - 601216545 bytes]

601216545 bytes copied in 50.649 secs (11870255 bytes/sec)
```

Step 3 Verify that the image has been successfully copied to flash by running the following command: **dir flash:**

```
Device# dir flash:*.bin
```

Step 4 Install the software image to flash by running the following command: **install add file bootflash:***image* **activate commit**

Note You can also use multi-step installation of the software. To perform multi-step installation, go to Step 5.

```
Device# install add file bootflash:C9800-universalk9_wlc.xx.xx.xx.SPA.bin activate commit

install_add_activate_commit: START Thu Dec  6 15:43:57 UTC 2018
Dec  6 15:43:58.669 %INSTALL-5-INSTALL_START_INFO: R0/0: install_engine: Started install one-shot
bootflash:C9800-xx-universalk9.xx.xx.xx.SPA.bin
install_add_activate_commit: Adding PACKAGE

--- Starting initial file syncing ---
Info: Finished copying bootflash:C9800-xx-universalk9.xx.xx.xx.SPA.bin to the selected chassis
Finished initial file syncing

--- Starting Add ---
Performing Add on all members
  [1] Add package(s) on chassis 1
  [1] Finished Add on chassis 1
```

```
Checking status of Add on [1]
Add: Passed on [1]
Finished Add

Image added. Version: xx.xx.xx.216
install_add_activate_commit: Activating PACKAGE
Following packages shall be activated:
/bootflash/C9800-xx-rpboot.xx.xx.xx.SPA.pkg
/bootflash/C9800-xx-mono-universalk9.xx.xx.xx.SPA.pkg

This operation requires a reload of the system. Do you want to proceed? [y/n]y
--- Starting Activate ---
Performing Activate on all members
  [1] Activate package(s) on chassis 1
    --- Starting list of software package changes ---
    Old files list:
      Removed C9800-xx-mono-universalk9.BLD_Vxxxx_THROTTLE_LATEST_20181022_153332.SSA.pkg
      Removed C9800-xx-rpboot.BLD_Vxxxx_THROTTLE_LATEST_20181022_153332.SSA.pkg
    New files list:
      Added C9800-xx-mono-universalk9.xx.xx.xx.SPA.pkg
      Added C9800-xx-rpboot.xx.xx.xx.SPA.pkg
    Finished list of software package changes
  [1] Finished Activate on chassis 1
Checking status of Activate on [1]
Activate: Passed on [1]
Finished Activate

--- Starting Commit ---
Performing Commit on all members
  [1] Commit package(s) on chassis 1
  [1] Finished Commit on chassis 1
Checking status of Commit on [1]
Commit: Passed on [1]
Finished Commit

Install will reload the system now!
SUCCESS: install_add_activate_commit  Thu Dec  6 15:49:21 UTC 2018
Dec  6 15:49:21.294 %INSTALL-5-INSTALL_COMPLETED_INFO: R0/0: install_engine: Completed install one-shot
 PACKAGE bootflash:C9800-xx-universalk9.xx.xx.xx.SPA.bin
```

Note The system reloads automatically after executing the **install add file activate commit** command. You do not have to manually reload the system.

Step 5 (Optional) You can also perform multi-step installation of the software:

Note Ensure that boot parameter is set to boot only from *flash:packages.conf*.

a) Add the controller software image to the flash and expanded it, using the **install add file** command.

```
Device# install add file bootflash:C9800-universalk9_wlc.xx.xx.xx.SPA.bin
```

b) Perform predownload of the AP image, using the **ap image predownload** command.

```
Device# ap image predownload
```

c) Check the predownload status of the AP, using the **show ap image** command.

```
Device# show ap image
```

d) Activate the package, using the **install activate** command.

```
Device# install activate
```

e) Commit the activation changes to be persistent across reloads using the **install commit** command.

```
Device# install commit
```

Step 6 Verify the installation by running the following command: **show version**

Note When you boot the new image, the boot loader is automatically updated, but the new bootloader version is not displayed in the output until the next reload.

Step 7 To see a summary of the active packages in a system, run the following command: **show install summary**

```
Device# show install summary

[ Chassis 1 2 ] Installed Package(s) Information:
State (St): I - Inactive, U - Activated & Uncommitted,
            C - Activated & Committed, D - Deactivated & Uncommitted
--------------------------------------------------------------------------------
Type  St   Filename/Version
--------------------------------------------------------------------------------
IMG   I    <v1>
IMG   C    <v2>
```

Upgrading the Controller Software (GUI)

Before you begin

Clean up the old installation files using the **Remove Inactive Files** link.

Note For GUI options such as *Software Maintenance Upgrade*, *AP Service Package*, and *AP Device Package*, see the respective feature sections.

Step 1 Choose **Administration > Software Management** .

Step 2 Choose an option from the **Upgrade Mode** drop-down list:

- **INSTALL**: The Install mode uses a package-provisioning file named *packages.conf* in order to boot a device.
- **BUNDLE**: The Bundle mode uses monolithic Cisco IOS images to boot a device. The Bundle mode consumes more memory than the Install mode because the packages are extracted from the bundle and copied to RAM.

Note You get to view the **Destination** field only for BUNDLE upgrade mode.

Step 3 From the **Transport Type** drop-down list, choose the transfer type to transfer the software image to your device as **TFTP**, **SFTP**, **FTP**, **Device**, or **Desktop (HTTP)**.

- If you choose **TFTP** as the **Transport Type**, enter the **Server IP Address** of the TFTP server that you want to use. Also, enter the complete **File Path**.

  In controllers, the IP TFTP source is mapped to the service port by default.

- If you choose **SFTP** as the **Transport Type**, enter the **Server IP Address** of the SFTP server that you want to use. Also, enter the **SFTP Username**, **SFTP Password**, and the complete **File Path**.
- If you choose **FTP** as the **Transport Type**, enter the **Server IP Address** of the FTP server that you want to use. Also, enter the **FTP Username**, **FTP Password**, and the complete **File Path**.
- If you choose **Device** as the **Transport Type**, choose the **File System** from the drop-down list. In the **File Path** field, browse through the available images or packages from the device and select one of the options, and click **Select**.
- If you choose **Desktop (HTTPS)** as the **Transport Type**, choose the **File System** from the drop-down list. In the **Source File Path** field, click **Select File** to select the file, and click **Open**.

Step 4 Click **Download & Install**.

Step 5 To boot your device with the new software image, click **Save Configuration & Reload**.

Rebooting the Controller

After you have copied the new system image into the bootflash memory, loaded the new system image, and saved a backup copy of the new system image and configuration, reboot the VM using the **reload** command.

Note When you reload an active device, it reloads the whole stack.

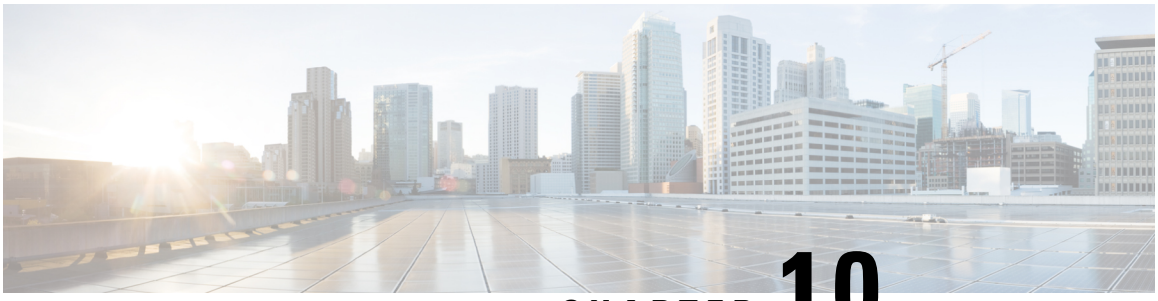
For more information about rebooting the VM, see your VMware documentation.

After rebooting, the controller VM must include the new system image with a newly installed Cisco IOS XE software version.

Note After an upgrade from 16.11 to an higher release, you should be able to view the new login page.

If not, perform either one of the following to redirect to the login page:

- Refresh GUI.
- Clear cache.



CHAPTER 10

License Information

Evaluation License

The wireless controller operates on evaluation mode when the device is not registered. The evaluation mode is for 90 days. After the expiry of the evaluation period, if the wireless controller is not registered to a smart account, the wireless controller will start displaying syslog evaluation expiration messages. These error messages are purely for informational purpose only and will not affect the functionality of the wireless controller.

The number of APs supported on the wireless controller when the wireless controller is on EVAL mode will be equal to the capacity of the wireless controller and the wireless controller will be fully operational. No other license is required to use the wireless controller in evaluation mode.

Viewing License Information

Use the **show license udi** command to determine the Universal Device Identifier (UDI) information of your chassis. This may be required at the time of purchasing a new license.

The following example displays sample output from the **show license udi** command:

```
Device# show license udi
SlotID        PID            SN               UDI
-------------------------------------------------------------------------------
*        C9800-CL             xxxxxxxxxxx    C9800-CL:xxxxxxxxxxx
```

Viewing the Cisco IOS License Level

Use the **show version** command to determine the Cisco IOS license level in the controller.

Example:

```
WLC# show version | section License
```

```
licensed under the GNU General Public License ("GPL") Version 2.0.  The
documentation or "License Notice" file accompanying the IOS-XE software,
License Type: Smart License is permanent
License Level: adventerprise
AIR License Level: AIR DNA Advantage
```

Table 11: Show version Command Output Description

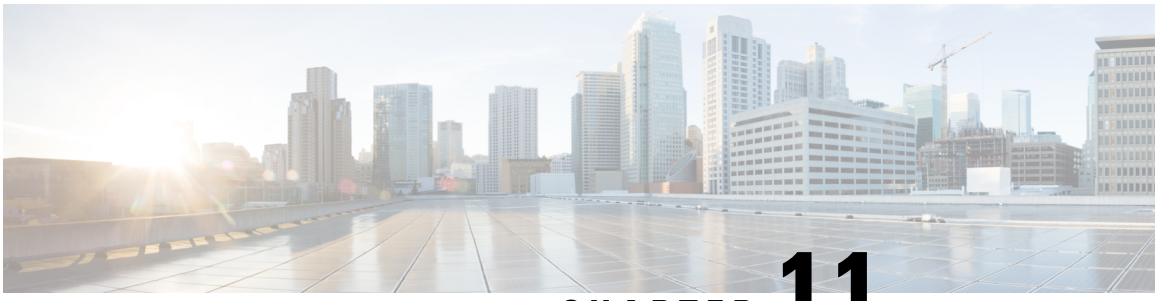
Field Name	Description
License Level: adventerprise	Indicates the current Cisco IOS license code level.
License Type: Smart License is permanent	Indicates the type of license that is used. This example shows that the Cisco Smart license is used that provides floating licenses for your user account. Other license types could be: Permanent (purchased) license or an Evaluation 60-day license.
AIR License Level: AIR DNA Advantage	Indicates the AIR network advantage license level.

Use the **show running-config** command or the **show startup-config** command to view the license-level information. The following example displays sample output from the **show running-config** command:

```
WLC# show running-config
.
.
.
license boot level adventerprise
```

Table 12: show running-config Command Output Description

Field Name	Description
license boot level adventerprise	Indicates the current requested Cisco lOS license level to boot.



CHAPTER 11

Troubleshooting

Verifying the Hardware and VM Requirements

To help troubleshoot issues with the controller, make sure that the device is installed on the supported hardware and the following VM requirements are being met:

- Verify that the server hardware is supported by the hypervisor vendor. If you are using VMware, verify that the server is listed in the VMware Hardware Compatibility List. For more information, see the VMware documentation set.
- Verify that the I/O devices, for example, Fibre Channel (FC), Internet Small Computer System Interface (iSCSI), and SAS that are being used are supported by the VM vendor.
- Verify that sufficient RAM is allocated on the server for the VMs and the hypervisor host.
- If you are using VMware, make sure the server has enough RAM to support both VMs and VMware ESXi.
- Verify if the hypervisor version is supported by the controller or not.
- Verify that the correct VM settings are configured based on the amount of memory, number of CPUs, and disk size.
- Verify that the vNICs are configured using a supported network driver.

Network Connectivity Issues

To troubleshoot network connectivity issues for the controller, ensure that the following requirements are met:

- Promiscuous mode should be set to accept to see the traffic sent and received through the vSwitch. Tagged traffic will not flow properly without this mode.
- Verify that there is an active and unexpired license installed on the VM. Enter the **show license** command. The **License State** should be shown as **Active**, **In Use**.
- Verify that the vNIC for the VMs are connected to the correct physical NIC or to the proper vSwitch.
- Ensure that the vSwitch is configured with the correct VLAN, if you are using virtual LANs (VLANs).

- Ensure that there are no duplicate MAC addresses, if you are using static MAC addresses or VMs that are cloned.

Caution Duplicate MAC addresses might cause the controller feature license to become invalidated, which will disable the device interfaces.

VM Performance Issues

The controller operates within a set of supported VM parameters and settings to provide certain levels of performance that have been tested by Cisco.

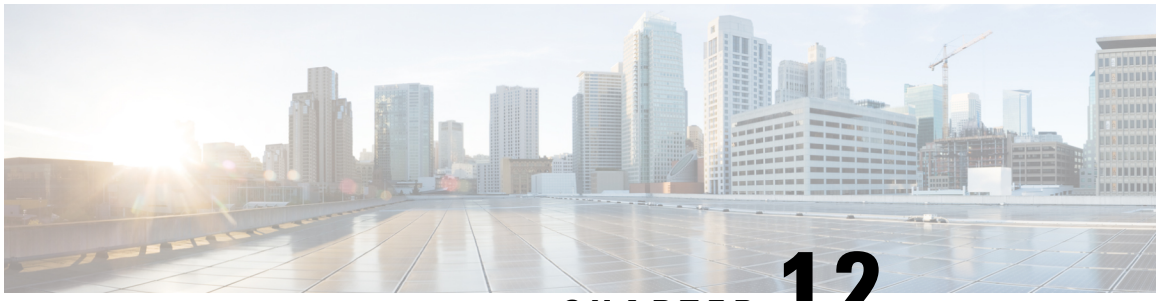
Use vSphere Client to view data and troubleshoot VM performance. If you are using vCenter, you can view historical data. If you are not using vCenter, you can view live data from the host.

Ensure that the following requirements are met to troubleshoot performance issues:

- Verify that the device is configured for the correct MTU setting.
- By default, the maximum MTU setting on the device is set to 1500. To support jumbo frames, you need to edit the default VMware vSwitch settings. For more information, see the VMware vSwitch documentation.
- The controller does not support memory sharing between VMs. On the ESXi host, check the memory counters to determine the used and shared memory on the VM. Verify that the counters used by the balloon and swap are zero.
- If a given VM does not have enough memory to support the controller, increase the size of the VM's memory. Insufficient memory on the VM or the host might cause the controller console to hang and be nonresponsive.

Caution When troubleshooting performance issues, note that other VMs on the same host as the controller can impact the performance of the controller VM. Verify that the other VMs on a host are not causing memory issues that impact the controller VM.

- Verify that no network packets are being dropped. On the ESXi host, check the network performance and view the counters to measure the number of receive and transmit packets dropped.



CHAPTER 12

Finding Support Information for Platforms and Cisco Software Images

Support Information for Platforms and Cisco Software Images

Cisco software is packaged in feature sets consisting of software images that support specific platforms. The feature sets available for a specific platform depend on which Cisco software images are included in a release. To identify the set of software images available in a specific release or find if a feature is available in a given Cisco IOS XE software image, you can use the Cisco Feature Navigator, Software Advisor, or the corresponding Release Notes document.

For all Cisco Wireless Controller software-related documentation, see:

https://www.cisco.com/c/en/us/support/wireless/catalyst-9800-series-wireless-controllers/tsd-products-support-series-home.html

Using Cisco Feature Navigator

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS XE software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn.

You need not be a registered user on Cisco.com to access this tool.

Using the Software Advisor

To determine if a feature is supported by a Cisco IOS XE release, locate the software document for that feature, or check the minimum Cisco IOS XE software requirements with your device, Cisco maintains the Software Advisor tool on Cisco.com at: http://tools.cisco.com/Support/Fusion/FusionHome.do

You must be a registered user on Cisco.com to access this tool.