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. The supported private cloud providers are VMware ESXi, KVM, Hyper-V, and Cisco NFVIS (on ENCS).

From Cisco IOS XE 17.12.3 onwards, the Cisco Catalyst 9800 Wireless Controller for Cloud - Ultra-Low Profile is introduced as the low memory variant of the Cisco Catalyst 9800-CL Cloud Wireless Controller.

The 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.

Limitations for Cisco Catalyst 9800-CL Ultra-Low Profile

- Cisco Catalyst 9800-CL Ultra-Low Profile is not supported as an Infrastructure-as-a-Service (IaaS) solution on the Amazon Web Services (AWS), Google Cloud Platform (GCP), Microsoft Azure Marketplaces.
- Local mode AP deployment is not supported on the Cisco Catalyst 9800-CL Ultra-Low Profile. Only FlexConnect Local Switching is supported.
- High Availability is not supported although configuration options will still be available on the controller.

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 For Small/Medium/Large profiles, 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.
Cisco IOS XE Dublin 17.12.3	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	Ultra-Low	Small (Low Throughput)	Medium (Low Throughput)	Large (Low Throughput)	Small (High Throughput)	Medium (High Throughput)	Large (High Throughput)
Minimum number of vCPUs (Hyperthreading is not supported)	2	4	6	10	7	9	13

Model Configuration	Ultra-Low	Small (Low Throughput)	Medium (Low Throughput)	Large (Low Throughput)	Small (High Throughput)	Medium (High Throughput)	Large (High Throughput
Minimum CPU Allocation (MHz)	2000	4,000	6,000	10, 000	4000	6000	10,000
Minimum Memory (GB)	6	8	16	32	8	16	32
Required Storage (GB)	16	16	16	16	16	16	16
Virtual NICs (vNIC) (*) 3rd NIC for High Availability (#) High Availability is not supported even though the configuration options are available.	2/(3)*#	2/(3)*	2/(3)*	2/(3)*	2/(3)*	2/(3)*	2/(3)*

Secure Boot

The secure boot feature prevents malicious software applications and unauthorized operating systems from loading into the controller during the controller startup process. If secure boot feature is enabled, only the authorized software applications boot up from the controller.

This feature ensures that the software applications that boot up on the controller are certified by Cisco. A secure compute system ensures that the intended software on the controller runs without malware or tampered software. The Unified Extensible Firmware Interface (UEFI) specification defines a secure boot methodology that prevents loading software that does not have an acceptable digital signature.

To view the secure boot mode and bootloader version, use the **show platform software system boot** command:

```
Device# show platform software system boot
Boot mode: EFI or EFI Secure
Bootloader version: 3.3
```

Guidelines

- The following secure boot environments are supported:
  - ESXi version 6.5
  - KVM RHEL 7.5 using open stack license
  - NFVIS release 3.11
- Only EFI firmware modes support the secure boot feature.
- This feature is supported on VM created in Cisco IOS XE Bengaluru 17.6 release.
- GRUB3 and new disk partition layout is available from Cisco IOS XE Bengaluru 17.6 release.

Note Any new VM instance created with version greater than or equal to 17.6.x cannot be downgraded to any lower versions. This restriction is due to the newer disk partitions required for UEFI.

Note If VM is installed using 17.6 ISO or OVA image, the downgrade to 17.3.4 fails during bootup with the following error message:

```
IOSXE image not compatible with installation.Failing boot..
```

Note Each hypervisor has a unique process to enable secure boot for the guest VMs. Refer to the relevant hypervisor documentation to enable secure boot. A set of high-level hypervisor specific steps to enable secure boot are mentioned below.

ESXi Secure Boot Setup

1. Create VM using ESXi version 6.5 or a later version using VM version 13.
2. To choose the EFI firmware mode, perform the following:
   a. Navigate to **Actions** > **Edit Settings**.

      The **Edit Settings** page is displayed.

   b. Navigate to **VM Options** > **Boot Options** > **Firmware**.

   c. From the **Choose which firmware should be used to boot the virtual machine** drop-down list, choose **EFI** option.

   d. Click **Save**.
3. Power up the VM to initialize the boot and wait for IOS prompt to complete.
4. Power down the VM.

5. To enable EFI secure boot, perform the following:
   a. Navigate to **Actions** > **Edit Settings**.

      The **Edit Settings** page is displayed.

   b. Navigate to **VM Options** > **Boot Options** > **Firmware**.

   c. Check the **Whether or not to enable UEFI secure boot for this VM** check box to enable EFI secure boot.

   d. Click **Save**.

6. Power up VM and the VNF boots up securely.

KVM Secure Boot Setup

1. Create a VM with a user-defined name.
2. Power down the VM after the VM is created and VNF IOS prompt is complete.
3. Install PK, KEK, and db certificates from the **EFI Firmware** menu and reset.

   To create the custom keys, see Custom Keys for Secure boot. For db certificates, see MicCorUEFCA2011_2011-06-27.crt and MicWinProPCA2011_2011-10-19.crt.
4. Secure boot the VM.

NFVIS Secure Boot Setup

1. Upgrade to NFVIS 3.11 release or a later one.
2. Register an ISRv EFI tarball with the NFVIS repository.
3. Create a VM using the registered EFI image.
4. Secure boot the VM.

Note Secure boot is disabled by default. To enable secure boot, you must change the firmware configurations from CIMC. Secure boot needs to boot from a separate UEFI partition.

To enable secure boot, perform the following:

a. Login to CIMC and use the **show bios detail** command to view the BIOS version.

```
ENCS# scope bios
ENCS/bios # show detail
BIOS:
    BIOS Version: " ENCS54_2.6 (Build Date: 07/12/2018)"
    Boot Order: EFI
    FW Update/Recovery Status: Done, OK
    Active BIOS on next reboot: main
    UEFI Secure Boot: disabled
ENCS/bios #
```

b. Enable secure boot.

```
ENCS/bios # set secure-boot enable
Setting Value : enable
Commit Pending.
ENCS/bios *# commit
ENCS/bios # show detail
BIOS:
    BIOS Version: "ENCS54_2.6 (Build Date: 07/12/2018)"
    Boot Order: EFI
    FW Update/Recovery Status: None, OK
    Active BIOS on next reboot: main
    UEFI Secure Boot: enabled
ENCS/bios #
```

Note Legacy boot, UEFI boot, and UEFI secure boot are the three boot modes. Secure boot can only be used on a disk that has UEFI partition.