# Overview

## Cisco UCS E-Series M6 Servers Overview

The Cisco UCS E-Series M6 Servers are size-, weight-, and power-efficient blade servers that are housed within the Cisco Catalyst 8300 Series Edge platforms. These servers provide a general-purpose compute platform for branch-office applications deployed either as bare-metal on operating systems, such as Linux, or as virtual machines on hypervisors, such as VMware vSphere Hypervisor.

The UCS E-Series M6 Server is purpose-built with powerful Intel IceLake-D processors for general purpose compute. It comes in the double-wide form factor, that fits into two SM slots.

**Note** Forinformation about the E-Series M6 Servers,and the maximum number of servers that can be installed per router, see section Hardware Requirements in the *Hardware Installation Guide for Cisco UCS E-Series M6 Servers.*.

## Server Software

The UCS E-Series M6 Servers require three major software systems:

- CIMC firmware
- BIOS firmware
- Operating system or hypervisor

### CIMC Firmware

Cisco Integrated Management Controller (CIMC) is a separate management module built into the motherboard ofthe E-Series M6 Servers. A dedicated processor, separate from the main server CPU, runs the CIMC firmware. The system ships with a running version of the CIMC firmware. You can update the CIMC firmware, but no initial installation is needed.

CIMC is the management service for the E-Series M6 Servers. You can use a web-based GUI or SSH-based CLI to access, configure, administer, and monitor the server.

### BIOS Firmware

BIOS initializes the hardware in the system, discovers bootable devices, and boots them in the provided sequence. It boots the operating system and configures the hardware for the operating system to use. BIOS manageabilityfeatures allow you to interact with the hardware and use it. In addition, BIOS provides options to configure the system, and manage firmware.

The system ships with a running version of the BIOS firmware. You can update the BIOS firmware, but no initial installation is required.

### Operating System or Hypervisor

The main server CPU runs on an operating system, such as Linux; or on a hypervisor. You can purchase an E-Series M6 Servers with a preinstalled operating system or hypervisor, or you can install your own platform

**Note** For information about the platforms that are available on the E-Series M6 Servers, see section Software Requirements in the *Release Notes for Cisco UCS E-Series M6 Servers*.

## CIMC Overview

The Cisco Integrated Management Controller (CIMC) is the management service for the E-Series M6 Servers. CIMC runs within the server. You can use a web-based GUI or the SSH-based CLI to access, configure, administer, and monitor the server.

You can use CIMC to perform the following server management tasks:

- Power on, power off, power cycle, reset, and shut down the server.
- Configure the server boot order.
- View server properties, router information, and chassis status.
- Manage remote presence.
- Create and manage local user accounts, and enable remote user authentication through the Active Directory.
- Configure network-related settings, including NIC properties, IPv4, VLANs, and network security.
- Configure communication services, including HTTP, SSH, IPMI over LAN, SNMP, and Redfish.
- Manage certificates.
- Configure platform event filters.

- Monitor power supply, fan, temperature, voltage, current, LED and storage sensors.
- Update CIMC firmware.
- Update BIOS firmware.
- Install the host image from an internal repository.
- Monitor faults, alarms, and server status.
- Set time zone and view local time.
- Collect technical support data in the event of server failure.

Most tasks can be performed in either the GUI interface or CLI interface, and the results of tasks performed in one interface are displayed in another. However, you *cannot*:

- Use the CIMC GUI to invoke the CIMC CLI.
- View a command that has been invoked through the CIMC CLI in the CIMC GUI.
- Generate CIMC CLI output from the CIMC GUI.

# CIMC CLI

The CIMC CLI is a command-line management interface for E-Series M6 Servers. You can launch the CIMC CLI in the following ways:

- By the serial port.
- Over the network by SSH.
- From the router. Use the following command:
  - **hw-module subslot** *slot/subslot* **session imc**—Use for E-Series M6 Servers installed in a Cisco Catalyst 8300 Edge Series platform.

A CLI user can have one of the three roles: admin, user (can control but cannot configure), and read-only.

## Command Modes

The CLI is organized into a hierarchy of command modes, with the EXEC mode being the highest-level mode of the hierarchy. Higher-level modes branch into lower-level modes. You use the **scope** command to move from higher-level modes to modes in the next lower level , and the **exit** command to move up one level in the mode hierarchy. The **top** command returns to the EXEC mode.

**Note** Most command modes are associated with managed objects. The **scope** command does not create managed objects and can only access modes for which managed objects already exist.

Each mode contains a set of commands that can be entered in that mode. Most of the commands available in each mode pertain to the associated managed object. Depending on your assigned role, you may have access to only a subset of the commands available in a mode; commands to which you do not have access are hidden.

The CLI prompt for each mode shows the full path down the mode hierarchy to the current mode. This helps you to determine where you are in the command mode hierarchy and can be an invaluable tool when you need to navigate through the hierarchy.

## Command Mode Table

The following table lists the first four levels of command modes, the commands used to access each mode, and the CLI prompt associated with each mode.

| Mode Name | Command to Access | Mode Prompt |
|---|---|---|
| `EXEC` | **top** command from any mode | # |
| `bios` | **scope bios** command from EXEC mode | /bios # |
| `certificate` | **scope certificate** command from EXEC mode | /certificate # |
| `chassis` | **scope chassis** command from EXEC mode | /chassis # |
| `cimc` | **scope cimc** command from EXEC mode | /cimc # |
| `fault` | **scope fault** command from EXEC mode | /fault # |
| `host-image-mapping` | **scope host-image-mapping** command from EXEC mode | /host-image-mapping# |
| `http` | **scope http** command from EXEC mode | /http # |
| `ipmi` | **scope ipmi** command from EXEC mode | /ipmi # |
| `kvm` | **scope kvm** command from EXEC mode | /kvm # |
| `ldap` | **scope ldap** command from EXEC mode | /ldap # |
| `sel` | **scope sel** command from EXEC mode | /sel # |
| `sensor` | **scope sensor** command from EXEC mode | /sensor # |
| `snmp` | **scope snmp** command from EXEC mode | /snmp # |
| `sol` | **scope sol** command from EXEC mode | /sol # |

| Mode Name | Command to Access | Mode Prompt |
|---|---|---|
| ssh | **scope ssh** command from EXEC mode | /ssh # |
| tacacs+ | **scope tacacs+** command from EXEC mode | /tacacs+ |
| user | **scope user** *user-number* command from EXEC mode | /user # |
| user-policy | **scope user-policy** *policy-number* command from EXEC mode | /user-policy # |
| user-session | **scope user-session** *session-number* command from EXEC mode | /user-session # |
| vmedia | **scope vmedia** command from EXEC mode | /vmedia # |

# Completing or Exiting a Command

You can use the Tab key in any mode to complete a command. Partially typing a command name and pressing Tab causes the command to be displayed in full or to the point where another keyword must be chosen or an argument value must be entered.

When you are inside a scope, the **exit** command allows you to move one level up. For example, if the scope is **/chassis/dimm-summary**, and you enter **exit**, the scope will move one level up to **/chassis**.

# Command History

The CLI stores all commands used in the current session. You can step through the previously used commands by using the Up Arrow or Down Arrow keys. The Up Arrow key steps to the previous command in the history, and the Down Arrow key steps to the next command in the history. If you get to the end of the history, pressing the Down Arrow key does nothing.

All commands in the history can be entered again by simply stepping through the history to recall the desired command and pressing Enter. The command is entered as if you had manually typed it. You can also recall a command and change it before you enter it.

# Committing, Discarding, and Viewing Pending Commands

When you enter a configuration command in the CLI, the command is not applied until you enter the **commit** command. Until committed, a configuration command is pending and can be discarded by entering a **discard** command. When any command is pending, an asterisk (*) appears before the command prompt. The asterisk disappears when you enter the **commit** command, as shown in this example:

```
Server# scope kvm
Server /kvm # set enabled yes
Server /kvm *# commit
Server /kvm #
```

You can accumulate pending changes in multiple command modes and apply them together with a single **commit** command. You can view the pending commands by entering the **show configuration pending** command in any command mode.

**Note** Committing multiple commands together is not an atomic operation. If any command fails, the successful commands are applied despite the failure. Failed commands are reported in an error message.

**Caution** The **commit** command must be used to commit changes that are made within the same scope. If you try to use the **commit** command to submit changes made in a different scope, you will get an error, and you will have to redo and recommit those changes.

# Command Output Formats

Most CLI **show** commands accept an optional **detail** keyword that causes the output information to be displayed as a list rather than as a table.

Depending on how you want the output information of the **detail** command to be displayed, use one of the following commands:

- **set cli output default**—Default format for easy viewing. The command output is presented in a compact list.

  This example shows the command output in the default format:

  ```
  Server /chassis # set cli output default
  Server /chassis # show hdd detail
  Name HDD_01_STATUS:
      Status : present
  Name HDD_02_STATUS:
      Status : present
  Name HDD_03_STATUS:
      Status : present

  Server /chassis #
  ```

- **set cli output yaml**—YAML format for easy parsing by scripts. The command output is presented in the YAML Ain't Markup Language (YAML) data serialization language, delimited by defined character strings.

  This example shows the command output in the YAML format:

  ```
  Server /chassis # set cli output yaml
  Server /chassis # show hdd detail
  ---
      name: HDD_01_STATUS
      hdd-status: present

  ---
      name: HDD_02_STATUS
      hdd-status: present
  ```

```
---
    name: HDD_03_STATUS
    hdd-status: present

...

Server /chassis #
```

For detailed information about YAML, see http://www.yaml.org/about.html.

# Online Help for the CLI

At any time, you can type the **?** character to display the options available at the current state of the command syntax. If you have not typed anything at the prompt, typing ? lists all available commands for the mode you are in. If you have partially typed a command, typing ? lists all available keywords and arguments available at your current position in the command syntax.