Overview

This chapter includes the following sections:

Overview of the Cisco UCS S-Series Rack-Mount Server S3260

The Cisco UCS S3260 is a modular, dense storage server with dual M3, M4 or M5 server nodes, optimized for large datasets used in environments such as big data, cloud, object storage, and content delivery.

The UCS S3260 chassis is a modular architecture consisting of the following modules:

- Base chassis: contains four redundant, hot-pluggable power supplies, eight redundant, hot-pluggable fans, and a rail kit.
- Server Node: one or two M3, M4 or M5 server nodes, each with two CPUs, 64, 128, 256, or 512 GB of DIMM memory, and a pass-through controller or a RAID card with a 1 GB or 4 GB cache.
- System I/O Controller (SIOC): one or two System I/O Controllers, each of which includes an integrated 1300-series or 1400-series virtual interface capability.
- Optional Drive Expansion Node: Large Form Factor (LFF) 3.5-inch drives in a choice of capacities.
- Solid State Drives: Up to 14 solid-state disks (SSDs) of 400GB, 800 GB, 1.6TB, and 3.2 TB capacities. These replace the previously supported top-loading LFF HDDs.
- Solid-State Boot Drives: up to two SSDs per M3, M4, or M5 server node. On the M4 server node, boot drives support hardware RAID connected to the RAID controller on the server node.
- I/O Expander: provides one storage mezz slot with two PCIe expansion slots and up to two NVMe SSDs.

The enterprise-class UCS S3260 storage server extends the capabilities of Cisco's Unified Computing System portfolio in a 4U form factor that delivers the best combination of performance, flexibility, and efficiency gains.

Note An M3 Server Node has Intel E5-2600 V2 CPUs and DDR-3 DIMMs. An M4 Server Node has Intel E5-2600 v4 CPUs and DDR-4 DIMMs

Overview of the Server Software

The Cisco UCS C-Series Rack-Mount Server ships with the firmware.

Firmware

is a separate management module built into the motherboard. A dedicated ARM-based processor, separate from the main server CPU, runs the firmware. The system ships with a running version of the firmware. You can update the firmware, but no initial installation is needed.

Server OS

The Cisco UCS C-Series rack servers support operating systems such as Windows, Linux, Oracle and so on. For more information on supported operating systems, see the *Hardware and Software Interoperability for Standalone C-series servers* at http://www.cisco.com/en/US/products/ps10477/prod_technical_reference_list.html. You can use to install an OS on the server using the KVM console and vMedia.

Server Ports

Following is a list of server ports and their default port numbers:

Table 1: Server Ports

Port Name	Port Number
LDAP Port 1	389
LDAP Port 2	389
LDAP Port 3	389
LDAP Port 4	3268
LDAP Port 5	3268
LDAP Port 6	3268
SSH Port	22
HTTP Port	80
HTTPS Port	443
SMTP Port	25

Port Name	Port Number
KVM Port	2068
Intersight Management Port	8889
Intersight Cloud Port	8888
SOL SSH Port	2400
SNMP Port	161
SNMP Traps	162
External Syslog	514

Cisco Integrated Management Controller

The is the management service for the C-Series servers. runs within the server.

Note The management service is used only when the server is operating in Standalone Mode. If your C-Series server is integrated into a UCS system, you must manage it using UCS Manager. For information about using UCS Manager, see the configuration guides listed in the *Cisco UCS B-Series Servers Documentation Roadmap* at http://www.cisco.com/go/unifiedcomputing/b-series-doc.

Management Interfaces

You can use a web-based GUI or SSH-based CLI or an XML-based API to access, configure, administer, and monitor the server. Almost all tasks can be performed in either interface, and the results of tasks performed in one interface are displayed in another. However, you cannot do the following:

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

Tasks You Can Perform in

You can use to perform the following server management tasks:

- Power on, power off, power cycle, reset and shut down the server
- Toggle the locator LED
- Configuring BIOS settings
- Configure the server boot order
- View server properties and sensors
- Manage remote presence

- Create and manage local user accounts, and enable remote user authentication through Active Directory
- Configure network-related settings, including NIC properties, IPv4, VLANs, and network security
- Configure communication services, including HTTP, SSH, IPMI Over LAN, and SNMP.
- Manage certificates
- Configure platform event filters
- Update firmware
- Monitor faults, alarms, and server status
- Set time zone and view local time
- Install and activate firmware
- Install and activate BIOS firmware
- Install and activate CMC firmware

No Operating System or Application Provisioning or Management

provisions servers, and as a result, exists below the operating system on a server. Therefore, you cannot use it to provision or manage operating systems or applications on servers. For example, you cannot do the following:

- Deploy an OS, such as Windows or Linux
- Deploy patches for software, such as an OS or an application
- Install base software components, such as anti-virus software, monitoring agents, or backup clients
- Install software applications, such as databases, application server software, or web servers
- Perform operator actions, including restarting an Oracle database, restarting printer queues, or handling non- user accounts
- Configure or manage external storage on the SAN or NAS storage

CLI

The CLI is a command-line management interface for Cisco UCS C-Series servers. You can launch the CLI and manage the server over the network by SSH or Telnet.

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

Note To recover from a lost admin password, see the Cisco UCS C-Series server installation and service guide for your platform.

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	#
server	**scope server** *index* command from EXEC mode	/server #
bios	**scope bios** command from server mode	/server/bios #
advanced	**scope advanced** command from bios mode	/server/bios/advanced #
main	**scope main** command from bios mode	/server/bios/main #
server-management	**scope server-management** command from bios mode	/server/bios/server-management #
bmc	**scope bmc** command from server mode	/server/bmc #
firmware	**scope firmware** command from bmc mode	/server/bios/bmc #
import-export	**scope import-export** command from bmc mode	/server/bios/import-export #

Mode Name	Command to Access	Mode Prompt
network	**scope network** command from bmc mode	/server/bios/network #
power-restore-policy	**scope power-restore-policy** command from bmc mode	/server/bios/power-restore-policy #
kvm	**scope kvm** command from server mode	/server/kvm #
ipmi	**scope ipmi** command from server mode	/server/ipmi #
dimm-blacklisting	**scope dimm-blacklisting** command from server mode	/server/dimm-blacklisting #
reset-ecc	**scope reset-ecc** command from server mode	/server/reset-ecc #
sel	**scope sel** command from server mode	/server/sel #
sol	**scope sol** command from server mode	/server/sol #
vmedia	**scope vmedia** command from server mode	/server/vmedia #
certificate	**scope certificate** command from EXEC mode	/certificate #
fault	**scope fault** command from EXEC mode	/fault #
http	**scope http** command from EXEC mode	/http #
ldap	**scope ldap** command from EXEC mode	/ldap #
binding	**scope binding** command from ldap mode	/ldap/binding #
dns-search	**scope dns-search** command from ldap mode	/ldap/dns-search #
ldap-group-rule	**scope ldap-group-rule** command from ldap mode	/ldap/ldap-group-rule #
ldap-server	**scope ldap-server** command from ldap mode	/ldap/ldap-server #
role-group	**scope role-group** command from ldap mode	/ldap/role-group #

Mode Name	Command to Access	Mode Prompt
network	**scope network** command from EXEC mode	/network #
ipblocking	**scope ipblocking** command from network mode	/network/ipblocking #
chassis	**scope chassis** command from EXEC mode	/chassis #
adapter	**scope adapter** *index* command from chassis mode	/chassis/adapter #
host-eth-if	**scope host-eth-if** command from adapter mode	/chassis/adapter/host-eth-if #
host-fc-if	**scope host-fc-if** command from adapter mode	/chassis/adapter/host-fc-if #
port-profiles	**scope port-profiles** command from adapter mode	/chassis/adapter/port-profiles #
vmfex	**scope vmfex** *index* command from adapter mode	/chassis/adapter/vmfex #
cmc	**scope cmc** *index* command from chassis mode	/chassis/cmc #
ipmi	**scope ipmi** command from cmc mode	/chassis/cmc/ipmi #
network	**scope network** command from cmc mode	/chassis/cmc/network #
firmware	**scope firmware** command from chassis mode	/chassis/firmware #
import-export	**scope import-export** command from chassis mode	/chassis/import-export #
log	**scope log** command from chassis mode	/chassis/log #
server	**scope server** command from log mode	/chassis/log/server #
sas-expander	**scope sas-expander** *index* command from chassis mode	/chassis/sas-expander #
phy-stats	**scope phy-stats** command from sas-expander mode	/chassis/sas-expander/phy-stats #
server	**scope server** *index* command from chassis mode	/chassis/server #

Mode Name	Command to Access	Mode Prompt
storageadapter	**scope storageadapter** command from server mode	/chassis/server/storageadapter #
dimm-summary	**scope dimm-summary** command from server mode	/chassis/server/dimm-summary #
tech-support	**scope tech-support** command from chassis mode	/chassis/tech-support #
sensor	**scope sensor** command from EXEC mode	/sensor #
snmp	**scope snmp** command from EXEC mode	/snmp #
trap-destinations	**scope trap-destinations** command from snmp mode	/snmp/trap-destinations #
v3users	**scope v3users** command from snmp mode	/snmp/v3users #
ssh	**scope ssh** command from EXEC mode	/ssh #
time	**scope time** command from EXEC mode	/time #
ntp	**scope ntp** command from time mode	/time/ntp #
user	**scope user** *user-number* command from EXEC mode	/user #
user-policy	**scope user-policy** command from EXEC mode	/user-policy #
user-session	**scope user-session** *session-number* command from EXEC mode	/user-session #
xmlapi	**scope xmlapi** command from EXEC mode	/xmlapi #

Complete 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.

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 press **Enter**.

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 chassis
Server /chassis # set locator-led off
Server /chassis *# commit
Server /chassis #
```

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.

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 a table. You can configure either of two presentation formats for displaying the output information when the **detail** keyword is used. The format choices are as follows:

- Default—For easy viewing, the command output is presented in a compact list.

  This example shows 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
Name HDD_04_STATUS:
    Status : present

Server /chassis #
```

- YAML—For easy parsing by scripts, the command output is presented in the YAML (YAML Ain't Markup Language) data serialization language, delimited by defined character strings.

  This example shows 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

---
    name: HDD_04_STATUS
    hdd-status: present

...

Server /chassis #
```

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

In most CLI command modes, you can enter **set cli output default** to configure the default format, or **set cli output yaml** to configure the YAML format.

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.

Logging In to

Step 1 Connect to the console port.

Step 2 When logging in to an unconfigured system for the first time, use **admin** as the username and **password** as the password.

The following situations occur when you login to the CLI for the first time:

- You cannot perform any operation until you change default admin credentials on the web UI or CLI.

Note After an upgrade from version 1.5(x) or 2.0(1) to the latest version, or when you do a factory reset, during first login prompts for a password change. You cannot choose the word 'password' as your new password. If this creates problems for any scripts you may be running, you could change it to password by logging back into the user management options, but this is ENTIRELY at your own risk. It is not recommended by Cisco.

Example

The following example shows how to login in to first time:

```
Login as # admin
admin10.101.255.255's password # password

**********WARNING**********
Default credentials were used for login.
Administration passwords needs to be changed for security purpose.
**********************************************************************

Enter current password # abcxyz
Re-enter new password # abcxyz
Updating password...
Password updated successfully.
Server #
```