



Chassis Management

- [The Cisco UCS S3260 Chassis, on page 1](#)
- [Acknowledging a Chassis, on page 2](#)
- [Decommissioning a Chassis, on page 2](#)
- [Removing a Chassis, on page 3](#)
- [Turning On the Locator LED for a Chassis, on page 4](#)
- [Turning Off the Locator LED for a Chassis, on page 4](#)

The Cisco UCS S3260 Chassis

Cisco UCS Manager Release 4.1(1) introduces support for the Cisco UCS S3260 chassis on Cisco UCS 64108 Fabric Interconnect.

Cisco UCS Manager Release 4.0(1) introduces support for the Cisco UCS S3260 chassis on Cisco UCS 6454 Fabric Interconnect.

Cisco UCS Manager Release 3.1(2) introduces support for the Cisco UCS S3260 chassis on Cisco UCS 6300 Series, and 6200 Series fabric interconnect setups.

The Cisco UCS S3260 chassis is a 4U chassis that is designed to operate in a standalone environment and also as part of the Cisco Unified Computing System. It has the following main components:

- Four 1050 Watt AC modular power supplies (2 + 2 shared and redundant mode of operation)
- Two System IO Controller (SIOC) slots
- Two storage server slots out of which one can be used for storage expansion



Note The second server slot in the chassis can be utilized by an HDD expansion tray module for an additional four 3.5” drives.

- 56 3.5” drive bays with an optional 4 x 3.5” HDD expansion tray module instead of the second server
- Up to 360 TB storage capacity by using 6 TB HDDs
- Serial Attached SCSI (SAS) expanders that can be configured to assign the 3.5” drives to individual server modules

- The two servers in the chassis can be replaced by a single, dual-height server with an IO expander

Acknowledging a Chassis

Perform the following procedure if you change the port that connects the chassis to the fabric interconnect. Acknowledging the chassis ensures that Cisco UCS Manager is aware of the change in the port.

After you change the port that connects the chassis to the fabric interconnect, wait for at least 1 minute before you reacknowledge the chassis.

SUMMARY STEPS

1. UCS-A# **acknowledge chassis** *chassis-num*
2. UCS-A* # **commit-buffer**

DETAILED STEPS

	Command or Action	Purpose
Step 1	UCS-A# acknowledge chassis <i>chassis-num</i>	Acknowledges the specified chassis.
Step 2	UCS-A* # commit-buffer	Commits the transaction to the system configuration.

Example

The following example acknowledges chassis 3 and commits the transaction:

```
UCS-A# acknowledge chassis 3
UCS-A* # commit-buffer
UCS-A #
```

Decommissioning a Chassis

SUMMARY STEPS

1. UCS-A# **decommission chassis** *chassis-num*
2. UCS-A* # **commit-buffer**

DETAILED STEPS

	Command or Action	Purpose
Step 1	UCS-A# decommission chassis <i>chassis-num</i>	Decommissions the specified chassis.
Step 2	UCS-A* # commit-buffer	Commits the transaction to the system configuration.

Decommissioning may take several minutes to complete.

Example

The following example decommissions chassis 2 and commits the transaction:

```
UCS-A# decommission chassis 2
UCS-A* # commit-buffer
UCS-A # show chassis
```

```
Chassis:
  Chassis   Overall Status   Admin State
  -----
           1 Operable       Acknowledged
           2 Accessibility Problem   Decommission
UCS-A #
```

Removing a Chassis

Before you begin

Physically remove the chassis before performing the following procedure.



Note You cannot remove a chassis if it is physically present in the system.

SUMMARY STEPS

1. UCS-A# **remove chassis** *chassis-num*
2. UCS-A* # **commit-buffer**

DETAILED STEPS

	Command or Action	Purpose
Step 1	UCS-A# remove chassis <i>chassis-num</i>	Removes the specified chassis.
Step 2	UCS-A* # commit-buffer	Commits the transaction to the system configuration.

The removal may take several minutes to complete.

Example

The following example removes chassis 3 and commits the transaction:

```
UCS-A# remove chassis 3
UCS-A* # commit-buffer
UCS-A #
```

Turning On the Locator LED for a Chassis

SUMMARY STEPS

1. UCS-A# **scope chassis** *chassis-num*
2. UCS-A /chassis # **enable locator-led** [multi-master | multi-slave]
3. UCS-A /chassis* # **commit-buffer**

DETAILED STEPS

	Command or Action	Purpose
Step 1	UCS-A# scope chassis <i>chassis-num</i>	Enters chassis mode for the specified chassis.
Step 2	UCS-A /chassis # enable locator-led [multi-master multi-slave]	Turns on the chassis locator LED. <ul style="list-style-type: none"> • multi-master—Turns on the LED for the master node only. • multi-slave—Turns on the LED for the slave node only.
Step 3	UCS-A /chassis* # commit-buffer	Commits the transaction to the system configuration.

Example

The following example turns on the locator LED for chassis 3 and commits the transaction:

```
UCS-A# scope chassis 3
UCS-A /chassis # enable locator-led
UCS-A /chassis* # commit-buffer
UCS-A /chassis #
```

The following example turns on the locator LED for the master node only and commits the transaction:

```
UCS-A# scope chassis 3
UCS-A /chassis # enable locator-led multi-master
UCS-A /chassis* # commit-buffer
UCS-A /chassis #
```

Turning Off the Locator LED for a Chassis

SUMMARY STEPS

1. UCS-A# **scope chassis** *chassis-num*
2. UCS-A /chassis # **disable locator-led** [multi-master | multi-slave]
3. UCS-A /chassis* # **commit-buffer**

DETAILED STEPS

	Command or Action	Purpose
Step 1	UCS-A# scope chassis <i>chassis-num</i>	Enters chassis mode for the specified chassis.
Step 2	UCS-A /chassis # disable locator-led [multi-master multi-slave]	Turns off the chassis locator LED. <ul style="list-style-type: none"> • multi-master—Turns off the LED for the master node only. • multi-slave—Turns off the LED for the slave node only.
Step 3	UCS-A /chassis* # commit-buffer	Commits the transaction to the system configuration.

Example

The following example turns off the locator LED for chassis 3 and commits the transaction:

```
UCS-A# scope chassis 3
UCS-A /chassis # disable locator-led
UCS-A /chassis* # commit-buffer
UCS-A /chassis #
```

The following example turns off the locator LED for the master node and commits the transaction:

```
UCS-A# scope chassis 3
UCS-A /chassis # disable locator-led multi-master
UCS-A /chassis* # commit-buffer
UCS-A /chassis #
```

