# Cisco UCS Manager Storage Management Guide using the CLI, Release 6.0

**First Published:** 2025-09-02

**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



## Audience

This guide is intended primarily for data center administrators with responsibilities and expertise in one or more of the following:

- Server administration
- Storage administration
- Network administration
- Network security

## Conventions

| Text Type | Indication |
|---|---|
| GUI elements | GUI elements such as tab titles, area names, and field labels appear in **this font**.<br>Main titles such as window, dialog box, and wizard titles appear in **this font**. |
| Document titles | Document titles appear in *this font*. |
| TUI elements | In a Text-based User Interface, text the system displays appears 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**.<br>Variables in a CLI command appear in *this font*. |
| [ ] | Elements in square brackets are optional. |

| Text Type | Indication |
|---|---|
| {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.

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

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

**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

# Related Cisco UCS Documentation

### Documentation Roadmaps

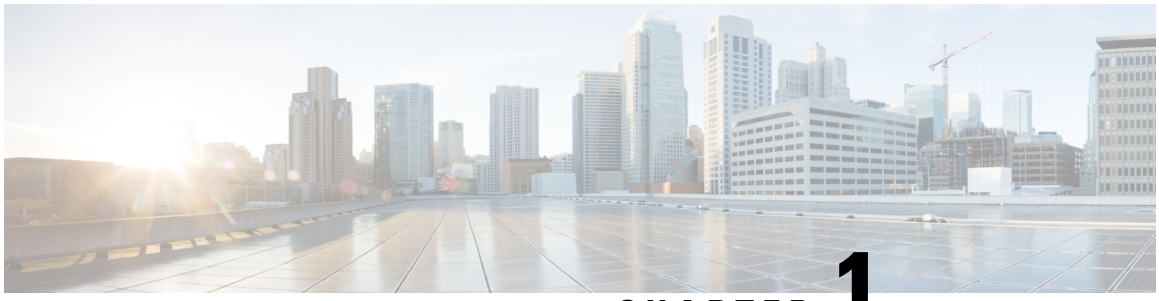
For a complete list of all B-Series documentation, see the *Cisco UCS B-Series Servers Documentation Roadmap* available at the following URL: https://www.cisco.com/c/en/us/td/docs/unified_computing/ucs/overview/guide/UCS_roadmap.html

For a complete list of all C-Series documentation, see the *Cisco UCS C-Series Servers Documentation Roadmapdoc roadmap* available at the following URL: https://www.cisco.com/c/en/us/td/docs/unified_computing/ucs/overview/guide/ucs_rack_roadmap.html.

For information on supported firmware versions and supported UCS Manager versions for the rack servers that are integrated with the UCS Manager for management, refer to Release Bundle Contents for Cisco UCS Software.

# Documentation Feedback

To provide technical feedback on this document, or to report an error or omission, please send your comments to ucs-docfeedback@external.cisco.com. We appreciate your feedback.



CHAPTER 1

# New and Changed Information



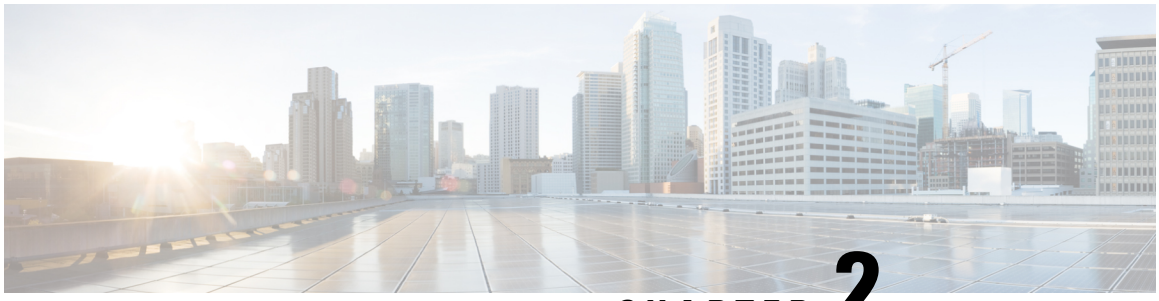
## New and Changed Information

This section provides information on new features and changed behaviors in this release.

***Table 1: New Features and Changed Behavior in Cisco UCS Manager, Release 6.0(1b)***

| Feature | Description | Where Documented |
|---|---|---|
| Support for Cisco UCS 6600 Series Fabric Interconnect. | Cisco UCS Manager now supports Cisco UCS 6664 Fabric Interconnect. | — |

New and Changed Information



CHAPTER 2

# Overview



## Overview

This guide describes how to configure the following storage management tasks:

- Ports and Port Channels
- Named VSANs
- SAN Pin Groups
- SAN Uplinks
- Pools
- FC Identity Assignment
- Storage-Related Policies
- Storage Profiles
- FlexFlash SD Card Support
- Direct Attached Storage
- Storage Inventory

## Cisco UCS Manager User CLI Documentation

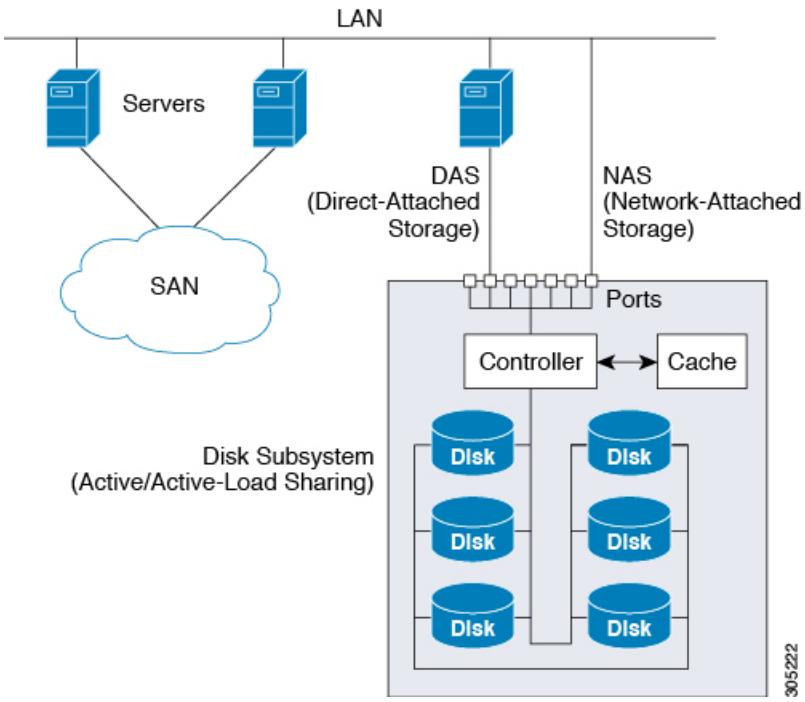
Cisco UCS Manager offers you a new set of smaller, use-case based documentation described in the following table:

| Guide | Description |
|---|---|
| Cisco UCS Manager Getting Started Guide | Discusses Cisco UCS architecture and Day 0 operations, including Cisco UCS Manager initial configuration, and configuration best practices. |
| Cisco UCS Manager Administration Guide | Discusses password management, role-based access configuration, remote authentication, communication services, CIMC session management, organizations, backup and restore, scheduling options, BIOS tokens and deferred deployments. |
| Cisco UCS Manager Infrastructure Management Guide | Discusses physical and virtual infrastructure components used and managed by Cisco UCS Manager. |
| Cisco UCS Manager Firmware Management Guide | Discusses downloading and managing firmware, upgrading through Auto Install, upgrading through service profiles, directly upgrading at endpoints using firmware auto sync, managing the capability catalog, deployment scenarios, and troubleshooting. |
| Cisco UCS Manager Server Management Guide | Discusses the new licenses, registering Cisco UCS domains with Cisco UCS Central, power capping, server boot, server profiles and server-related policies. |
| Cisco UCS Manager Storage Management Guide | Discusses all aspects of storage management such as SAN and VSAN in Cisco UCS Manager. |
| Cisco UCS Manager Network Management Guide | Discusses all aspects of network management such as LAN and VLAN connectivity in Cisco UCS Manager. |
| Cisco UCS Manager System Monitoring Guide | Discusses all aspects of system and health monitoring including system statistics in Cisco UCS Manager. |
| Cisco UCS S3260 Server Integration with Cisco UCS Manager | Discusses all aspects of management of UCS S-Series servers that are managed through Cisco UCS Manager. |

# Storage Options

The following are the UCS Manager storage options and the benefits of each.

*Figure 1: Cisco UCS Manager Storage Options*




- **Direct Attached Storage (DAS)**—This is the storage available inside a server and is directly connected to the system through the motherboard within a parallel SCSI implementation. DAS is commonly described as captive storage. Devices in a captive storage topology do not have direct access to the storage network and do not support efficient sharing of storage. To access data with DAS, a user must go through a front-end network. DAS devices provide little or no mobility to other servers and little scalability.

  DAS devices limit file sharing and can be complex to implement and manage. For example, to support data backups, DAS devices require resources on the host and spare disk systems that other systems cannot use. The cost and performance of this storage depends upon the disks and RAID controller cards inside the servers. DAS is less expensive and is simple to configure; however, it lacks the scalability, performance, and advanced features provided by high-end storage.

- **Network Attached Storage (NAS)**—This storage is usually an appliance providing file system access. This storage could be as simple as an Network File System (NFS) or Common Internet File System (CIFS) share available to the servers. Typical NAS devices are cost-effective devices with not very high performance but have very high capacity with some redundancy for reliability. NAS is usually moderately expensive, simple to configure, and provides some advanced features; however, it also lacks scalability, performance, and advanced features provided by SAN.

- **Storage Area Network (SAN)**—A SAN is a specialized, high-speed network that attaches servers and storage devices. A SAN allows an any-to-any connection across the network by using interconnect elements, such as switches and directors. It eliminates the traditional dedicated connection between a server and storage, and the concept that the server effectively owns and manages the storage devices. It also eliminates any restriction to the amount of data that a server can access, currently limited by the number of storage devices that are attached to the individual server. Instead, a SAN introduces the flexibility of networking to enable one server or many heterogeneous servers to share a common storage utility. A network might include many storage devices, including disk, tape, and optical storage. Additionally, the storage utility might be located far from the servers that it uses. This type of storage

provides maximum reliability, expandability, and performance. The cost of SAN is also very high compared to other storage options.

SAN is the most resilient, highly scalable, and high performance storage; however, it is also the most expensive and complex to manage.

# Storage Design Considerations

UCS storage physical connectivity has a slightly different design consideration as compared to LAN physical connectivity. The following are some design considerations for SAN connectivity:

- Northbound storage physical connectivity does not support virtual port channels (vPCs) like LAN connectivity.
- Port channels or trunking is possible to combine multiple storage uplink ports that provide physical link redundancy.
- Redundancy of storage resources is handled by the storage itself and varies from vendor to vendor.
- Connect storage through northbound Cisco storage devices, such as Nexus or MDS Fabric Switches.
- It is possible to connect storage directly to UCS Fabric Interconnects, which is recommended for small implementations because of the fabric interconnect physical ports consumption and increased processing requirements.
- Software configuration including VSANs and zoning is required for providing access to storage resources.

# Storage Configuration Sequence

Follow the suggested sequence to configure a storage network:

1. Configure and enable server ports, uplink ports, and FC ports.
2. Create a management IP address pool (typically on the same subnet as the UCS Manager Admin IP address).
3. Create an UUID Pool, MAC Pool, WWNN Pool, WWPN Pool (or populate the corresponding "default" pools). Embed domain ID's. Use Fabric-specific Pools for MAC and WWPN (for example, Fabric-A, Fabric-B).
4. For SAN boot, create a unique "Boot Policy" for each storage array boot target.
5. Create VNIC templates (for example, eth0-A, eth1-B) that both draw from the above MAC Pool, and are associated with Fabric-A and Fabric-B respectively.
6. Create VHBA templates (for example, fc0-A, fc1-B) that both draw from the above WWPN Pool, and are associated with Fabric-A and Fabric-B respectively.
7. Create service profile templates that draw from all earlier established pools, policies and templates, as appropriate.
8. Instantiate the service profile from the template and associate the service profile to a given blade, or set the service profile template to associate with a particular server pool.

# Storage Protocols

Fiber Channel, iSCSI, and Fiber Channel over Ethernet are protocols for SAN connectivity.

- **iSCSI**—An industry standard protocol for attaching various I/O peripherals such as printers, scanners, tape drives, and storage devices. The most common SCSI devices are disks and tape libraries.

  SCSI is the core protocol to connect raw hard disk storage with the servers. To control remote storage with the SCSI protocol, different technologies are used as wrappers to encapsulate commands, such as FC and iSCSI.

  Fiber Channel protocol provides the infrastructure to encapsulate the SCSI traffic and provided connectivity between computers and storage. FC operates at speeds of 2, 4, 8, and 16 Gbps.

- **Fiber Channel (FC)** consists of the following:
  - Hard disk arrays that provide raw storage capacity.
  - Storage processors to manage hard disks and provide storage LUNs and masking for the servers.
  - Fiber Channel Switches (also known as Fabric) that provide connectivity between storage processors and server HBAs.
  - Fiber Channel Host Bus Adapters: They are installed in the computer and provide connectivity to the SAN.

  Fiber Channel identifies infrastructure components with World Wide Numbers (WWN) . WWNs are 64-bit addresses which uniquely identify the FC devices. Like MAC addresses, it has bits assigned to vendors to identify their devices. Each end device (like an HBA port) is given a World Wide Port Number (WWPN) and each connectivity device (like a Fabric switch) is given a World Wide Node Number (WWNN).

  A Fiber Chanel HBA used for connecting to a SAN is known as an initiator, and Fiber Channel SAN providing disks as LUNs is known as a target. The Fiber Channel protocol is different from Ethernet or TCP/IP protocols.

- **Fiber Channel over Ethernet (FCoE)** transport replaces the Fibre Channel cabling with 10 Gigabit Ethernet cables and provides lossless delivery over unified I/O. Ethernet is widely used in networking. With some advancement such as Data Center Ethernet (DCE) and Priority Flow Control (PFC) in Ethernet to make it more reliable for the datacenter, Fiber Channel is now also implemented on top of Ethernet. This implementation is known as FCoE.

# The Cisco UCS Manager SAN Tab

From the SAN tab, you the UCS administrator can create, modify, and delete configuration elements related to SANs (FC, iSCSI) or direct attached FC/FCoE, NAS appliances, and communications.

The major nodes in this tab are the following:

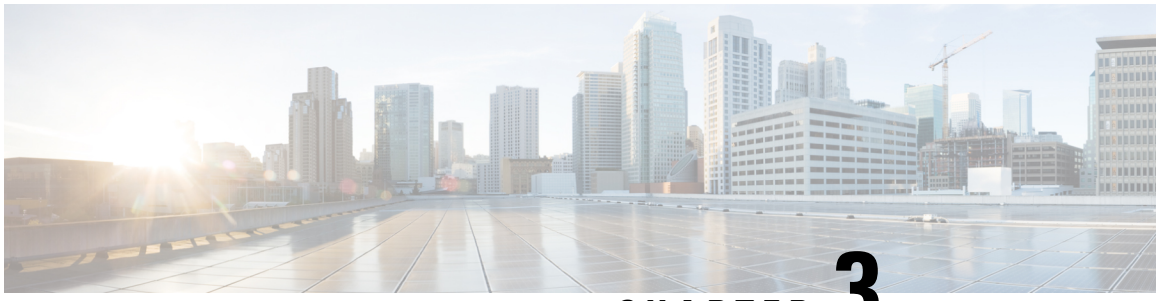
- **SAN Cloud**—This node allows you to:
  - Configure SAN uplinks, including storage ports and port channels and SAN pin groups.
  - View the FC identity assignment

- Configure WWN Pools, including WWPN, WWxN, and WWxN, and iSCSI Qualified Name (IQN), pools.
- View the FSM details for a particular end point to determine if a task succeeded or failed and use the FSM to troubleshoot any failures.
- Monitor storage events and faults for health management.

- **Storage Cloud**—This node allows you to:
  - Configure storage FC links and storage FCoE interfaces (using SAN Storage Manager.
  - Configure VSAN settings.
  - Monitor SAN cloud events for health management.

- **Policies**—This node allows you to:
  - Configure threshold policies, classes, and properties and monitor events.
  - Configure threshold organization and sub-organization storage policies, including default VHBA, behavior, FC adaptor, LACP, SAN connectivity, SAN connector, and VHBA templates.

- **Pools**—This node allows you to configure pools defined in the system, including IQN, IQN suffix, WWNN, WWPN, and WWxN.
- **Traffic Monitoring Sessions**—This node allows you to configure port traffic monitoring sessions defined in the system.



CHAPTER 3

# SAN Ports and Port Channels



## Port Modes

The port mode determines whether a unified port on the fabric interconnect is configured to carry Ethernet or Fibre Channel traffic. You configure the port mode in Cisco UCS Manager. However, the fabric interconnect does not automatically discover the port mode.

Changing the port mode deletes the existing port configuration and replaces it with a new logical port. Any objects associated with that port configuration, such as VLANs and VSANS, are also removed. There is no restriction on the number of times you can change the port mode for a unified port.

## Port Types

The port type defines the type of traffic carried over a unified port connection.

By default, unified ports changed to Ethernet port mode are set to the Ethernet uplink port type. Unified ports changed to Fibre Channel port mode are set to the Fibre Channel uplink port type. You cannot unconfigure Fibre Channel ports.

Changing the port type does not require a reboot.

**Ethernet Port Mode**

When you set the port mode to Ethernet, you can configure the following port types:

- Server ports
- Ethernet uplink ports
- Ethernet port channel members
- FCoE ports
- Appliance ports
- Appliance port channel members
- SPAN destination ports
- SPAN source ports

**Note** For SPAN source ports, configure one of the port types and then configure the port as SPAN source.

**Fibre Channel Port Mode**

When you set the port mode to Fibre Channel, you can configure the following port types:

- Fibre Channel uplink ports
- Fibre Channel port channel members
- Fibre Channel storage ports
- SPAN source ports

**Note** For SPAN source ports, configure one of the port types and then configure the port as SPAN source.

# Effect of Port Mode Changes on Data Traffic

Port mode changes can cause an interruption to the data traffic for the Cisco UCS domain. The length of the interruption and the traffic that is affected depend upon the configuration of the Cisco UCS domain and the module on which you made the port mode changes.

**Tip** To minimize the traffic disruption during system changes, form a Fibre Channel uplink port-channel across the fixed and expansion modules.

### Impact of Port Mode Changes on an Expansion Module

After you make port mode changes on an expansion module, the module reboots. All traffic through ports on the expansion module is interrupted for approximately one minute while the module reboots.

### Impact of Port Mode Changes on the Fixed Module in a Cluster Configuration

A cluster configuration has two fabric interconnects. After you make port changes to the fixed module, the fabric interconnect reboots. The impact on the data traffic depends upon whether or not you have configured the server vNICs to failover to the other fabric interconnect when one fails.

If you change the port modes on the expansion module of one fabric interconnect and then wait for that to reboot before changing the port modes on the second fabric interconnect, the following occurs:

- With server vNIC failover, traffic fails over to the other fabric interconnect and no interruption occurs.
- Without server vNIC failover, all data traffic through the fabric interconnect on which you changed the port modes is interrupted for approximately eight minutes while the fabric interconnect reboots.

If you change the port modes on the fixed modules of both fabric interconnects simultaneously, all data traffic through the fabric interconnects are interrupted for approximately eight minutes while the fabric interconnects reboot.

### Impact of Port Mode Changes on the Fixed Module in a Standalone Configuration

A standalone configuration has only one fabric interconnect. After you make port changes to the fixed module, the fabric interconnect reboots. All data traffic through the fabric interconnect is interrupted for approximately eight minutes while the fabric interconnect reboots.

# FC Links Rebalancing

The FC uplinks balance automatically when FC Port Channels are utilized. To create FC Port Channels, refer to Configuring a Fibre Channel Port Channel, on page 39.

For the FC uplinks that are not members of the Port Channels (Individual ISLs), load balancing is done according to the FC uplinks balancing algorithm. For a vHBA of a host or service profile to choose an available FC uplink, when FC uplink trunking is disabled, the uplink and vHBA must belong to the same VSAN

For each vHBA, the algorithm searches for an FC uplink in the following order:

1. Least used FC uplink based on the number of vHBAs currently bound to the uplink.
2. If FC uplinks are equally balanced, then round robin is used.

This process continues for all the other vHBAs. The algorithm also considers other parameters such as pre-fip/fip adapters and number of flogis. You may not see the least-used component when there are less than six flogis.

After a port configuration or any other uplink state changes, if the traffic passing through the FC uplinks is no longer balanced, you can re-balance the traffic by resetting the vHBA(s) on each adapter and allow the load balancing algorithm to evaluate for the current state of the FC uplinks.

# Configuring the Port Mode

**Caution** Changing the port mode can cause an interruption in data traffic because changes to the fixed module require a reboot of the fabric interconnect.

If the Cisco UCS domain has a cluster configuration that is set up for high availability and servers with service profiles that are configured for failover, traffic fails over to the other fabric interconnect and data traffic is not interrupted when the port mode is changed on the fixed module.

In the Cisco UCS Manager CLI, there are no new commands to support Unified Ports. Instead, you change the port mode by scoping to the mode for the desired port type and then creating a new interface. When you create a new interface for an already configured slot ID and port ID, UCS Manager deletes the previously configured interface and creates a new one. If a port mode change is required because you configure a port that previously operated in Ethernet port mode to a port type in Fibre Channel port mode, UCS Manager notes the change.

**Note** Expansions modules are not supported with Cisco UCS Mini.

### Procedure

**Step 1** UCS-A# **scope** *port-type-mode*

Enters the specified port type mode for one of the following port types:

**eth-server**

For configuring server ports.

**eth-storage**

For configuring Ethernet storage ports and Ethernet storage port channels.

**eth-traffic-mon**

For configuring Ethernet SPAN ports.

**eth-uplink**

For configuring Ethernet uplink ports.

**fc-storage**

For configuring Fibre Channel storage ports.

**fc-traffic-mon**

For configuring Fibre Channel SPAN ports.

**fc-uplink**

For configuring Fibre Channel uplink ports and Fibre Channel uplink port channels.

**Step 2** UCS-A /*port-type-mode* # **scope fabric** {**a** | **b**}

Enters the specified port type mode for the specified fabric.

**Step 3** UCS-A /*port-type-mode*/fabric # **create interface** *slot-id port-id*

Creates an interface for the specified port type.

If you are changing the port type from Ethernet port mode to Fibre Channel port mode, or vice-versa, the following warning appears:

```
Warning: This operation will change the port mode (from Ethernet to FC or vice-versa). When
committed, this change will require the module to restart.
```

**Step 4** Create new interfaces for other ports belonging to the Ethernet or Fibre Channel port block.

There are several restrictions that govern how Ethernet and Fibre Channel ports can be arranged on a fixed or expansion module. Among other restrictions, it is required that you change ports in groups of two. Violating any of the restrictions outlined in the Guidelines and Recommendations for Configuring Unified Ports section will result in an error.

**Step 5** UCS-A /*port-type-mode*/fabric/interface # **commit-buffer**

Commits the transaction to the system configuration.

---

Based on the module for which you configured the port modes, data traffic for the Cisco UCS domain is interrupted as follows:

- Fixed module—The fabric interconnect reboots. All data traffic through that fabric interconnect is interrupted. In a cluster configuration that provides high availability and includes servers with vNICs that are configured for failover, traffic fails over to the other fabric interconnect and no interruption occurs. Changing the port mode for both sides at once results in both fabric interconnects rebooting simultaneously and a complete loss of traffic until both fabric interconnects are brought back up.

  It takes about 8 minutes for the fixed module to reboot.

- Expansion module—The module reboots. All data traffic through ports in that module is interrupted.

  It takes about 1 minute for the expansion module to reboot.

### Example

The following example changes ports 3 and 4 on slot 1 from Ethernet uplink ports in Ethernet port mode to uplink Fibre Channel ports in Fibre Channel port mode:

```
UCS-A# scope fc-uplink
UCS-A /fc-uplink # scope fabric a
UCS-A /fc-uplink/fabric # create interface 1 3
Warning: This operation will change the port mode (from Ethernet to FC or vice-versa).
When committed, this change will require the fixed module to restart.
UCS-A /fc-uplink/fabric/interface* # up
UCS-A /fc-uplink/fabric* #create interface 1 4
Warning: This operation will change the port mode (from Ethernet to FC or vice-versa).
```

```
When committed, this change will require the fixed module to restart.
UCS-A /fc-uplink/fabric/interface* #commit-buffer
```

# Displaying Port Properties and Fibre Channel Statistics

### Procedure

**Step 1** UCS-A /fabric-interconnect # **connect nxos** {**a** | **b**

Enters NX-OS mode for the fabric interconnect.

**Step 2** UCS-A(nxos)# **show interface fc** *slot-id*/*port-id*

Displays the port properties and Fibre Channel statistics such as throughput rates and errors.

**Note**
The **receive B2B credit remaining** on a UCS 6400 Series fabric interconnect does not match the **transmit B2B credit remaining** on its peer switch. The **receive B2B credit remaining** parameter always remains at 64 because the credit is returned when the frame is released. On an MDS peer, the **transmit B2B credit remaining** parameter can go to 0. This results in the mismatch. You can run the **show interface fc** *slot-id*/*port-id* command on the peer switch to compare these parameters.

### Example

The following example displays port properties and Fibre Channel statistics for UCS 6400 Series fabric interconnects:

```
UCS-A /fabric-interconnect # connect nxos a
UCS-A(nxos)# show interface fc 1/6
fc1/6 is trunking
    Hardware is Fibre Channel, SFP is short wave laser w/o OFC (SN)
    Port WWN is 20:06:00:de:fb:21:77:00
    Admin port mode is NP, trunk mode is on
    snmp link state traps are enabled
    Port mode is TNP
    Port vsan is 8
    Speed is 16 Gbps
    Transmit B2B Credit is 32
    Receive B2B Credit is 64
    Receive data field Size is 2112
    Beacon is turned off
    Belongs to san-port-channel 32
    Trunk vsans (admin allowed and active) (1,5,7-8,40,120)
    Trunk vsans (up)                       (1,5,7-8,40,120)
    Trunk vsans (isolated)                 ()
    Trunk vsans (initializing)             ()
    5 minutes input rate 497578904 bits/sec,62197363 bytes/sec, 30981 frames/sec
    5 minutes output rate 501679056 bits/sec,62709882 bytes/sec, 30319 frames/sec
      430000799 frames input,863205473268 bytes
        0 discards,0 errors
        0 invalid CRC/FCS,0 unknown class
        0 too long,0 too short
```

```
  423530360 frames output,876009587416 bytes
    0 discards,0 errors
  1 input OLS,1  LRR,9 NOS,0 loop inits
  1 output OLS,0 LRR, 8 NOS, 0 loop inits
  64 receive B2B credit remaining
  32 transmit B2B credit remaining
  0 low priority transmit B2B credit remaining
Last clearing of "show interface" counters  :never
```

# Server Ports

## Configuring a Server Port

### Procedure

**Step 1** UCS-A# **scope eth-server**

Enters Ethernet server mode.

**Step 2** UCS-A /eth-server # **scope fabric** {**a** | **b**}

Enters Ethernet server fabric mode for the specified fabric.

**Step 3** UCS-A /eth-server/fabric # **create interface** *slot-num port-num*

Creates an interface for the specified Ethernet server port.

**Step 4** UCS-A /eth-server/fabric # **commit-buffer**

Commits the transaction to the system configuration.

### Example

The following example shows how to create an interface for Ethernet server port 4 on slot 1 of fabric B and commit the transaction:

```
UCS-A# scope eth-server
UCS-A /eth-server # scope fabric b
UCS-A /eth-server/fabric # create interface 1 4
UCS-A /eth-server/fabric* # commit-buffer
UCS-A /eth-server/fabric #
```

# Unconfiguring a Server Port

### Procedure

**Step 1** UCS-A# **scope eth-server**

Enters Ethernet server mode.

**Step 2** UCS-A /eth-server # **scope fabric** {**a** | **b**}

Enters Ethernet server fabric mode for the specified fabric.

**Step 3** UCS-A /eth-server/fabric # **delete interface** *slot-num* *port-num*

Deletes the interface for the specified Ethernet server port.

**Step 4** UCS-A /eth-server/fabric # **commit-buffer**

Commits the transaction to the system configuration.

### Example

The following example unconfigures Ethernet server port 12 on slot 1 of fabric B and commits the transaction:

```
UCS-A# scope eth-server
UCS-A /eth-server # scope fabric b
UCS-A /eth-server/fabric # delete interface 1 12
UCS-A /eth-server/fabric* # commit-buffer
UCS-A /eth-server/fabric #
```

# Unified Ports

## Guidelines for Configuring Unified Ports

Consider the following guidelines and restrictions when configuring unified ports:

### Hardware and Software Requirements

Unified ports are supported on the following:

- Cisco UCS 6664 Fabric Interconnect with Cisco UCS Manager Release 6.0(1b)and later releases
- Cisco UCS Fabric Interconnects 9108 100G (Cisco UCS X-Series Direct ) with Cisco UCS Manager Release 4.3(4b) and later releases
- Cisco UCS 6536 Fabric Interconnect with Cisco UCS Manager Release 4.2(3b) and later releases
- Cisco UCS 64108 Fabric Interconnect with Cisco UCS Manager Release 4.1 and later releases

- Cisco UCS 6454 Fabric Interconnect with Cisco UCS Manager Release 4.0 and later releases

### Port Mode Placement

Because the Cisco UCS Manager GUI interface uses a slider to configure the port mode for unified ports , it automatically enforces the following restrictions which limits how port modes can be assigned to unified ports. When using the Cisco UCS Manager CLI interface, these restrictions are enforced when you commit the transaction to the system configuration. If the port mode configuration violates any of the following restrictions, the Cisco UCS Manager CLI displays an error:

- Ethernet ports must be grouped together in a block.
- Fibre Channel ports must be grouped together in a block.

**Note**

- On the Cisco UCS Fabric Interconnects 9108 100G, the ports 1 & 2 are unified ports and can be configured as Ethernet or Fibre Channel ports.
- On the Cisco UCS 6536 Fabric Interconnect, the Unified Port capability is restricted to the first 16 ports. Only ports 1/1-1/16 can be configured as FC. The Fibre Channel ports must be contiguous, followed by contiguous Ethernet ports.
- On the Cisco UCS 6400 Series Fabric Interconnect, the Unified Port capability is restricted to first 16 ports. Only ports 1/1-1/16 can be configured as FC. The Fibre Channel ports must be contiguous, followed by contiguous Ethernet ports. The Cisco UCS 6400 Series Fabric Interconnect connected to a Cisco UCS server, connecting more than 16 ports will result in an error.
- Alternating Ethernet and Fibre Channel ports is not supported .

### Special Considerations for UCS Manager CLI Users

Because the Cisco UCS Manager CLI does not validate port mode changes until you commit the buffer to the system configuration, it is easy to violate the grouping restrictions if you attempt to commit the buffer before creating at least two new interfaces. To prevent errors, we recommend that you wait to commit your changes to the system configuration until you have created new interfaces for all of the unified ports changing from one port mode to another.

Commiting the buffer before configuring multiple interfaces will result in an error, but you do not need to start over. You can continue to configure unified ports until the configuration satisfies the aforementioned requirements.

# Cautions and Guidelines for Configuring Unified Uplink Ports and Unified Storage Ports

The following are cautions and guidelines to follow while working with unified uplink ports and unified storage ports:

- In an unified uplink port, if you enable one component as a SPAN source, the other component will automatically become a SPAN source.

**Note** If you create or delete a SPAN source under the Ethernet uplink port, Cisco UCS Manager automatically creates or deletes a SPAN source under the FCoE uplink port. The same happens when you create a SPAN source on the FCOE uplink port.

- You must configure a non default native VLAN on FCoE and unified uplink ports. This VLAN is not used for any traffic. Cisco UCS Manager will reuse an existing fcoe-storage-native-vlan for this purpose. This fcoe-storage-native-vlan will be used as a native VLAN on FCoE and unified uplinks.
- In an unified uplink port, if you do not specify a non default VLAN for the Ethernet uplink port the fcoe-storage-native-vlan will be assigned as the native VLAN on the unified uplink port. If the Ethernet port has a non default native VLAN specified as native VLAN, this will be assigned as the native VLAN for unified uplink port.
- When you create or delete a member port under an Ethernet port channel, Cisco UCS Manager automatically creates or deletes the member port under FCoE port channel. The same happens when you create or delete a member port in FCoE port channel.
- When you configure an Ethernet port as a standalone port, such as server port, Ethernet uplink, FCoE uplink or FCoE storage and make it as a member port for an Ethernet or FCOE port channel, Cisco UCS Manager automatically makes this port as a member of both Ethernet and FCoE port channels.
- When you remove the membership for a member port from being a member of server uplink, Ethernet uplink, FCoE uplink or FCoE storage, Cisco UCS Manager deletes the corresponding members ports from Ethernet port channel and FCoE port channel and creates a new standalone port.
- For unified uplink ports and unified storage ports, when you create two interfaces, only one license is checked out. As long as either interface is enabled, the license remains checked out. The license will be released only if both the interfaces are disabled for a unified uplink port or a unified storage port.
- In Cisco UCS 6536 Fabric Interconnect to configure FC breakout port, you have to configure ports from the sequence from 1/36 through 1/33. FC breakout ports (36 - 33) cannot be configured unless the previous ports are FC breakout ports. Also, configuring a single (individual) FC breakout port is supported.

  Ports 33-36 can be configured only as FC Uplink Port or FC Storage Port when it is configured as unified port.
- The Cisco UCS Fabric Interconnects 9108 100G (Cisco UCS X-Series Direct) supports port breakout for Ethernet Ports (1-8) and Unified Ports (1 and 2). These unified ports can function as Ethernet or Fibre Channel (FC) ports, accommodating up to 8 sub-ports configured in groups of four. The FC breakout ports can be configured as FC Uplink Port or FC Storage Port.

# Beacon LEDs for Unified Ports

Each port fabric interconnect has a corresponding beacon LED. When the **Beacon LED** property is configured, the beacon LEDs illuminate, showing you which ports are configured in a given port mode.

You can configure the **Beacon LED** property to show you which ports are grouped in one port mode: either Ethernet or Fibre Channel. By default, the Beacon LED property is set to Off.

**Note** For unified ports on the expansion module, you can reset the **Beacon LED** property to the default value of **Off** during expansion module reboot.

# Configuring the Beacon LEDs for Unified Ports

Complete the following task for each module for which you want to configure beacon LEDs.

### Procedure

**Step 1** UCS-A# **scope fabric-interconnect** {**a** | **b**}

Enters fabric interconnect mode for the specified fabric.

**Step 2** UCS-A /fabric # **scope card** *slot-id*

Enters card mode for the specified fixed or expansion module.

**Step 3** UCS-A /fabric/card # **scope beacon-led**

Enters beacon LED mode.

**Step 4** UCS-A /fabric/card/beacon-led # **set admin-state** {**eth** | **fc** | **off**}

Specifies which port mode is represented by illuminated beacon LED lights.

**eth**

All of the Unified Ports configured in Ethernet mode illuminate.

**fc**

All of the Unified Ports configured in Fibre Channel mode illuminate.

**off**

Beacon LED lights for all ports on the module are turned off.

**Step 5** UCS-A /fabric/card/beacon-led # **commit-buffer**

Commits the transaction to the system configuration.

### Example

The following example illuminates all of the beacon lights for Unified Ports in Ethernet port mode and commits the transaction:

```
UCS-A# scope fabric-interconnect a
UCS-A /fabric # scope card 1
UCS-A /fabric/card # scope beacon-led
UCS-A /fabric/card/beacon-led # set admin-state eth
UCS-A /fabric/card/beacon-led* # commit-buffer
UCS-A /fabric/card/beacon-led #
```

# Unified Ports on the Fabric Interconnect

Unified ports are ports on the fabric interconnect that can be configured to carry either Ethernet or Fibre Channel traffic. These ports are not reserved. A Cisco UCS domain cannot use these ports until you configure them.

**Note** When you configure a port on a fabric interconnect, the administrative state is automatically set to enabled. If the port is connected to another device, this may cause traffic disruption. You can disable the port after configuring it.

Configurable beacon LEDs indicate which unified ports are configured for the selected port mode.

# Unified Storage Ports

Unified storage involves configuring the same physical port as both an Ethernet storage interface and an FCoE storage interface. You can configure any appliance port or FCoE storage port as a unified storage port. To configure a unified storage port, you must have the fabric interconnect in Fibre Channel switching mode.

In a unified storage port, you can enable or disable individual FCoE storage or appliance interfaces.

- In an unified storage port, if you do not specify a non-default VLAN for the appliance port, the FCoE-storage-native-vlan will be assigned as the native VLAN on the unified storage port. If the appliance port has a non-default native VLAN specified as native VLAN, this will be assigned as the native VLAN for the unified storage port.
- When you enable or disable the appliance interface, the corresponding physical port is enabled or disabled. So when you disable the appliance interface in unified storage, even if the FCoE storage is enabled, it goes down with the physical port.
- When you enable or disable the FCoE storage interface, the corresponding VFC is enabled or disabled. So when the FCoE storage interface is disabled in a unified storage port, the appliance interface will continue to function normally.

## Configuring a Unified Storage Port

### Procedure

**Step 1** UCS-A# **scope eth-storage**

Enters Ethernet storage mode.

**Step 2** UCS-A /eth-storage # **scope fabric**{**a** | **b**}

Enters Ethernet storage mode for the specified fabric.

**Step 3** UCS-A /eth-storage/fabric # **create interface** *slot-num port-num*

Creates an interface for the specified appliance port.

**Step 4** UCS-A /eth-storage/fabric/interface* # **commit buffer**

Commits the transaction to the system configuration.

**Step 5** UCS-A /eth-storage/fabric/interface* # **scope fc-storage**

Enters FC storage mode.

**Step 6** UCS-A /fc-storage* # **scope fabric**{**a** | **b**}

Enters Ethernet storage mode for the specific appliance port.

**Step 7** UCS-A /fc-storage/fabric # **create interface fcoe** *slot-num port-num*

Adds FCoE storage port mode on the appliance port mode and creates a unified storage port..

### Example

The following example creates an interface for an appliance port 2 on slot 3 of fabric A, adds fc storage to the same port to convert it as an unified port , and commits the transaction:

```
UCS-A# scope eth-storage
UCS-A /eth-storage # scope fabric a
UCS-A /eth-storage/fabric # create interface 3 2
UCS-A /eth-storage/fabric* # commit-buffer
UCS-A /eth-storage/fabric* # scope fc-storage
UCS-A /fc-storage*# scope fabric a
UCS-A /fc-storage/fabric* # create interface fcoe 3 2
UCS-A /fc-storage/fabric* # commit-buffer
UCS-A /fc-storage/fabric*
```

# Unified Uplink Ports

When you configure an Ethernet uplink and an FCoE uplink on the same physical Ethernet port, it is called a unified uplink port. You can individually enable or disable either the FCoE or Ethernet interfaces independently.

- Enabling or disabling the FCoE uplink results in the corresponding VFC being enabled or disabled.
- Enabling or disabling an Ethernet uplink results in the corresponding physical port being enabled or disabled.

If you disable an Ethernet uplink, it disables the underlying physical port in a unified uplink. Therefore, even when the FCoE uplink is enabled, the FCoE uplink also goes down. But if you disable an FCoE uplink, only the VFC goes down. If the Ethernet uplink is enabled, it can still function properly in the unified uplink port.

## Configuring a Unified Uplink Port

To configure a unified uplink port, you will convert an existing FCoE uplink port as a unified port.

### Procedure

**Step 1** UCS-A# **scope eth-uplink**

Enters Ethernet uplink mode.

**Step 2** UCS-A /eth-uplink # **scope fabric** {**a** | **b**}

Enters Ethernet uplink fabric mode for the specified fabric.

**Step 3** UCS-A /eth-uplink/fabric # **create interface** *15*

Converts the FCoE uplink port as a unified port.

**Step 4** UCS-A /eth-uplink/fabric/port-channel # **commit-buffer**

Commits the transaction to the system configuration.

### Example

The following example creates a unified uplink port on an existing FCoE port:

```
UCS-A# scope eth-uplink
UCS-A /eth-uplink # scope fabric b
UCS-A /eth-uplink/fabric # create interface 1 5
UCS-A /eth-uplink/fabric/interface* # commit-buffer
UCS-A /eth-uplink/interface #
```

# Unified Uplink Port Channel

When you create an Ethernet port channel and an FCoE port channel with the same ID, it is called a unified uplink port channel. When the unified port channel is created, a physical Ethernet port channel and a VFC are created on the fabric interconnect with the specified members. The physical Ethernet port channel is used to carry both Ethernet and FCoE traffic. The VFC binds FCoE traffic to the Ethernet port channel.

The following rules will apply to the member port sets of the unified uplink port channel:

- The Ethernet port channel and FCoE port channel on the same ID, must have the same set of member ports.
- When you add a member port channel to the Ethernet port channel, Cisco UCS Manager adds the same port channel to FCoE port channel as well. Similarly, adding a member to the FCoE port channel adds the member port to the Ethernet port channel.
- When you delete a member port from one of the port channels, Cisco UCS Manager automatically deletes the member port from the other port channel.

If you disable an Ethernet uplink port channel, it disables the underlying physical port channel in a unified uplink port channel. Therefore, even when the FCoE uplink is enabled, the FCoE uplink port channel also goes down. If you disable an FCoE uplink port channel, only the VFC goes down. If the Ethernet uplink port channel is enabled, it can still function properly in the unified uplink port channel.

## Configuring a Unified Uplink Port Channel

To configure a unified uplink port channel, you will convert an existing FCoE uplink port channel as a unified port channel.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | UCS-A# **scope eth-uplink** | Enters Ethernet uplink mode. |
| **Step 2** | UCS-A /eth-uplink # **scope fabric** {**a** \| **b**} | Enters Ethernet uplink fabric mode for the specified fabric. |
| **Step 3** | UCS-A /eth-uplink/fabric # **create port-channel** *ID* | Creates a port channel for the specified Ethernet uplink port. |
| **Step 4** | UCS-A /eth-uplink/fabric/port-channel # **commit-buffer** | Commits the transaction to the system configuration. |

### Example

The following example creates a unified uplink port channel on an existing FCoE port channel:

```
UCS-A# scope eth-uplink
UCS-A /eth-uplink # scope fabric b
UCS-A /eth-uplink/fabric # create port-channel 2
UCS-A /eth-uplink/fabric/port-channel* # commit-buffer
UCS-A /eth-uplink/fabric #
```

# Cisco UCS Mini Scalability Ports

The Cisco UCS 6324 Fabric Interconnect contains a scalability port as well as four unified ports. The scalability port is a 40GB QSFP+ breakout port that, with proper cabling, can support four 1G or 10G SFP+ ports. The scalability ports can be used as a licensed server port for supported Cisco UCS rack servers, an appliance port, or a FCoE port.

In the Cisco UCS Manager GUI, the scalability port is displayed as **Scalability Port 5** below the **Ethernet Ports** node. The individual breakout ports are displayed as **Port 1** through **Port 4**.

In the Cisco UCS Manager CLI, the scalability port is not displayed, but the individual breakout ports are displayed as **Br-Eth1/5/1** through **Br-Eth1/5/4**.

## Configuring Scalability Ports

To configure ports, port channel members or SPAN members on the scalability port, scope into the scalability port first, then follow the steps for a standard unified port.

### Procedure

**Step 1** UCS-A# **scope eth-server**

Enters Ethernet server mode.

**Step 2** UCS-A /eth-server # **scope fabric** {**a** | **b**}

Enters Ethernet server fabric mode for the specified fabric.

**Step 3** UCS-A /eth-server/fabric # **scope aggr-interface** *slot-num port-num*

Enters ethernet server fabric aggregate interface mode for the scalability port.

**Step 4** UCS-A /eth-server/fabric/aggr-interface # **show interface**

Displays the interfaces on the scalability port.

**Step 5** UCS-A /eth-server/fabric/aggr-interface # **create interface** *slot-num port-num*

Creates an interface for the specified Ethernet server port.

**Step 6** UCS-A /eth-server/fabric/aggr-interface # **commit-buffer**

Commits the transaction to the system configuration.

### Example

The following example shows how to create an interface for Ethernet server port 3 on the fabric A scalability port and commit the transaction:

```
UCS-A# scope eth-server
UCS-A /eth-server # scope fabric a
UCS-A /eth-server/fabric # scope aggr-interface 1 5
UCS-A /eth-server/fabric/aggr-interface # show interface
Interface:

Slot Id Aggr-Port ID Port Id  Admin State Oper State    State Reason
------- ------------ -------- ----------- ------------- ------------
      1            5        1 Enabled     Up
      1            5        2 Enabled     Up
      1            5        3 Enabled     Admin Down    Administratively Down
      1            5        4 Enabled     Admin Down    Administratively Down

UCS-A /eth-server/fabric/aggr-interface # create interface 1 3
UCS-A /eth-server/fabric/aggr-interface* # commit-buffer
UCS-A /eth-server/fabric/aggr-interface #
```

# Appliance Ports

Appliance ports are only used to connect fabric interconnects to directly attached NFS storage.

**Note** When you create a new appliance VLAN, its IEEE VLAN ID is not added to the LAN Cloud. Therefore, appliance ports that are configured with the new VLAN remain down, by default, due to a pinning failure. To bring up these appliance ports, you have to configure a VLAN in the LAN Cloud with the same IEEE VLAN ID.

Cisco UCS Manager supports up to four appliance ports per fabric interconnect.

# Configuring an Appliance Port

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | UCS-A# **scope eth-storage** | Enters Ethernet storage mode. |
| **Step 2** | UCS-A /eth-storage # **scope fabric**{**a** \| **b**} | Enters Ethernet storage mode for the specified fabric. |
| **Step 3** | UCS-A /eth-storage/fabric # **create interface** *slot-num port-num* | Creates an interface for the specified appliance port. |
| **Step 4** | (Optional) UCS-A /eth-storage/fabric/interface # **set portmode** {**access** \| **trunk**} | Specifies whether the port mode is access or trunk. By default, the mode is set to trunk.<br>**Note**<br>If traffic for the appliance port needs to traverse the uplink ports, you must also define each VLAN used by this port in the LAN cloud. For example, you need the traffic to traverse the uplink ports if the storage is also used by other servers, or if you want to ensure that traffic fails over to the secondary fabric interconnect if the storage controller for the primary fabric interconnect fails. |
| **Step 5** | (Optional) UCS-A /eth-storage/fabric/interface # **set pingroupname** *pin-group name* | Specifies the appliance pin target to the specified fabric and port, or fabric and port channel. |
| **Step 6** | (Optional) UCS-A /eth-storage/fabric/interface # **set prio** *sys-class-name* | Specifies the QoS class for the appliance port. By default, the priority is set to best-effort.<br>The sys-class-name argument can be one of the following class keywords:<br>• **FC**—Use this priority for QoS policies that control vHBA traffic only.<br>• **Platinum**—Use this priority for QoS policies that control vNIC traffic only.<br>• **Gold**—Use this priority for QoS policies that control vNIC traffic only.<br>• **Silver**—Use this priority for QoS policies that control vNIC traffic only.<br>• **Bronze**—Use this priority for QoS policies that control vNIC traffic only. |

| | Command or Action | Purpose |
|---|---|---|
| | | • **Best Effort**—Do not use this priority. It is reserved for the Basic Ethernet traffic lane. If you assign this priority to a QoS policy and configure another system class as CoS 0, Cisco UCS Manager does not default to this system class. It defaults to the priority with CoS 0 for that traffic. |
| **Step 7** | (Optional) UCS-A /eth-storage/fabric/interface # **set adminspeed** {**10gbps** \| **1 gbps**} | Specifies the admin speed for the interface. By default, the admin speed is set to 10gbps. |
| **Step 8** | UCS-A /eth-storage/fabric/interface # **commit buffer** | Commits the transaction to the system configuration. |

### Example

The following example creates an interface for an appliance port 2 on slot 3 of fabric B, sets the port mode to access, pins the appliance port to a pin group called pingroup1, sets the QoS class to fc, sets the admin speed to 10 gbps, and commits the transaction:

```
UCS-A# scope eth-storage
UCS-A /eth-storage # scope fabric b
UCS-A /eth-storage/fabric # create interface 3 2
UCS-A /eth-storage/fabric* # set portmode access
UCS-A /eth-storage/fabric* # set pingroupname pingroup1
UCS-A /eth-storage/fabric* # set prio fc
UCS-A /eth-storage/fabric* # set adminspeed 10gbps
UCS-A /eth-storage/fabric* # commit-buffer
UCS-A /eth-storage/fabric #
```

### What to do next

Assign a VLAN or target MAC address for the appliance port.

# Assigning a Target MAC Address to an Appliance Port or Appliance Port Channel

The following procedure assigns a target MAC address to an appliance port. To assign a target MAC address to an appliance port channel, scope to the port channel instead of the interface.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | UCS-A# **scope eth-storage** | Enters Ethernet storage mode. |
| **Step 2** | UCS-A /eth-storage # **scope fabric**{**a** \| **b**} | Enters Ethernet storage mode for the specified fabric. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 3** | UCS-A /eth-storage/fabric # **scope interface** *slot-id port-id* | Enters Ethernet interface mode for the specified interface.<br>**Note** To assign a target MAC address to an appliance port channel, use the **scope port-channel** command instead of **scope interface** . |
| **Step 4** | UCS-A /eth-storage/fabric/interface # **create eth-target** *eth-target name* | Specifies the name for the specified MAC address target. |
| **Step 5** | UCS-A /eth-storage/fabric/interface/eth-target # **set mac-address** *mac-address* | Specifies the MAC address in nn:nn:nn:nn:nn:nn format. |

### Example

The following example assigns a target MAC address for an appliance device on port 3, slot 2 of fabric B and commits the transaction:

```
UCS-A# scope eth-storage
UCS-A /eth-storage* # scope fabric b
UCS-A /eth-storage/fabric* # scope interface 2 3
UCS-A /eth-storage/fabric/interface* # create eth-target macname
UCS-A /eth-storage/fabric/interface* # set mac-address 01:23:45:67:89:ab
UCS-A /eth-storage/fabric/interface* # commit-buffer
UCS-A /eth-storage/fabric #
```

The following example assigns a target MAC address for appliance devices on port channel 13 of fabric B and commits the transaction:

```
UCS-A# scope eth-storage
UCS-A /eth-storage* # scope fabric b
UCS-A /eth-storage/fabric* # scope port-channel 13
UCS-A /eth-storage/fabric/port-channel* # create eth-target macname
UCS-A /eth-storage/fabric/port-channel* # set mac-address 01:23:45:67:89:ab
UCS-A /eth-storage/fabric/port-channel* # commit-buffer
UCS-A /eth-storage/fabric #
```

# Creating an Appliance Port

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | UCS-A# **scope eth-storage** | Enters Ethernet storage mode. |
| **Step 2** | UCS-A/eth-storage# **create vlan** *vlan-name vlan-id* | Creates a named VLAN, specifies the VLAN name and VLAN ID, and enters Ethernet storage VLAN mode |
| **Step 3** | UCS-A/eth-storage/vlan# **set sharing primary** | Saves the changes. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 4** | UCS-A/eth-storage/vlan# **commit buffer** | Commits the transaction to the system configuration. |
| **Step 5** | UCS-A/eth-storage# **create vlan** *vlan-name vlan-id* | Creates a named VLAN, specifies the VLAN name and VLAN ID, and enters Ethernet storage VLAN mode . |
| **Step 6** | UCS-A/eth-storage/vlan# **set sharing community** | Associates the primary VLAN to the secondary VLAN that you are creating. |
| **Step 7** | UCS-A/eth-storage/vlan# **set pubnwname** *primary vlan-name* | Specifies the primary VLAN to be associated with this secondary VLAN. |
| **Step 8** | UCS-A/eth-storage/vlan# **commit buffer** | Commits the transaction to the system configuration. |

### Example

The following example creates an appliance port:

```
UCS-A# scope eth-storage
UCS-A/eth-storage# create vlan PRI600 600
UCS-A/eth-storage/vlan* # set sharing primary
UCS-A/eth-storage/vlan* # commit-buffer
UCS-A/eth-storage # create vlan COM602 602
UCS-A/eth-storage/vlan* # set sharing isolated
UCS-A/eth-storage/vlan* # set pubnwname PRI600
UCS-A/eth-storage/vlan* # commit-buffer
```

# Mapping an Appliance Port to a Community VLAN

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | UCS-A# **scope eth-storage** | Enters Ethernet storage mode. |
| **Step 2** | UCS-A/eth-storage# **scope fabric** {*a* \| *b*} | Enters Ethernet storage fabric interconnect mode for the specified fabric interconnect. |
| **Step 3** | UCS-A/eth-storage/fabric# **create interface** *slot-num port-num* | Creates an interface for the specified Ethernet server port. |
| **Step 4** | UCS-A/eth-storage/fabric/interface# **exit** | Exits from the interface.<br>**Note**<br>Ensure you commit the transaction after associating with the VLAN. |
| **Step 5** | UCS-A/eth-storage/fabric# **exit** | Exits from the fabric. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 6** | UCS-A/eth-storage# **scope vlan** *vlan-name* | Enters the specified VLAN.<br>**Note**<br>Ensure community VLAN is created in the appliance cloud. |
| **Step 7** | UCS-A/eth-storage/vlan# **create member-port** *fabric slot-num port-num* | Creates the member port for the specified fabric, assigns the slot number, and port number and enters member port configuration. |
| **Step 8** | UCS-A/eth-storage/vlan/member-port# **commit** | Commits the transaction to the system configuration. |

### Example

The following example maps an appliance port to an community VLAN:

```
UCS-A# scope eth-storage
UCS-A/eth-storage# scope fabric a
UCS-A/eth-storage/fabric# create interface 1 22
UCS-A/eth-storage/fabric/interface*# exit
UCS-A/eth-storage/fabric*# exit
UCS-A/eth-storage*# scope vlan COM602
UCS-A/eth-storage/vlan*# create member-port a 1 22
UCS-A/eth-storage/vlan/member-port* commit
```

# Unconfiguring an Appliance Port

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | UCS-A # **scope eth-storage** | Enters Ethernet storage mode. |
| **Step 2** | UCS-A /eth-storage # **scope fabric** {**a** \| **b**} | Enters Ethernet storage mode for the specified fabric. |
| **Step 3** | UCS-A /eth-storage/fabric # **delete eth-interface** *slot-num port-num* | Deletes the interface for the specified appliance port. |
| **Step 4** | UCS-A /eth-storage/fabric # **commit-buffer** | Commits the transaction to the system configuration. |

### Example

The following example unconfigures appliance port 3 on slot 2 of fabric B and commits the transaction:

```
UCS-A# scope eth-storage
UCS-A /eth-storage # scope fabric b
```

```
UCS-A /eth-storage/fabric # delete eth-interface 2 3
UCS-A /eth-storage/fabric* # commit-buffer
UCS-A /eth-storage/fabric #
```

# FCoE Uplink Ports

FCoE uplink ports are physical Ethernet interfaces between the fabric interconnects and the upstream Ethernet switch, used for carrying FCoE traffic. With this support the same physical Ethernet port can carry both Ethernet traffic and Fibre Channel traffic.

FCoE uplink ports connect to upstream Ethernet switches using the FCoE protocol for Fibre Channel traffic. This allows both the Fibre Channel traffic and Ethernet traffic to flow on the same physical Ethernet link.

**Note** FCoE uplinks and unified uplinks enable the multi-hop FCoE feature, by extending the unified fabric up to the distribution layer switch.

You can configure the same Ethernet port as any of the following:

- **FCoE uplink port**—As an FCoE uplink port for only Fibre Channel traffic.
- **Uplink port**—As an Ethernet port for only Ethernet traffic.
- **Unified uplink port**—As a unified uplink port to carry both Ethernet and Fibre Channel traffic.

# Configuring a FCoE Uplink Port

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | UCS-A# **scope fc-uplink** | Enters FC Uplink mode. |
| **Step 2** | UCS-A /fc-uplink # **scope fabric**{**a** \| **b**} | Enters FC - Uplink mode for the specific fabric. |
| **Step 3** | UCS-A /fc-uplink/fabric # **create fcoeinterface** *slot-numberport-number* | Creates interface for the specified FCoE uplink port. |
| **Step 4** | UCS-A /fc-uplink/fabric/fabricinterface # **commit-buffer** | Commits the transaction to the system configuration. |

### Example

The following example creates an interface for FCoE uplink port 1 on slot 8 of fabric A and commits the transaction:

```
UCS-A# scope fc-uplink
UCS-A /fc-uplink # scope fabric a
UCS-A /fc-uplink/fabric # create fcoeinterface 1 8
```

```
UCS-A /fc-uplink/fabric/fcoeinterface* # commit-buffer
UCS-A /fc-uplink/fabric/fcoeinterface #
```

# Viewing FCoE Uplink Ports

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | UCS-A# **scope fc-uplink** | Enters FC Uplink mode. |
| **Step 2** | UCS-A /fc-uplink # **scope fabric** {**a** \| **b**} | Enters FC - Uplink mode for the specific fabric. |
| **Step 3** | UCS-A /fc-uplink/fabric # **show fcoeinterface** | Lists the available interfaces. |

### Example

The following example displays the available FCoE uplink interfaces on fabric A:

```
UCS-A# scope fc-uplink
UCS-A /fc-uplink # scope fabric a
UCS-A /fc-uplink/fabric # show fcoeinterface
FCoE Interface:

Slot Id    Port Id    Admin State Operational State Operational State Reason  Li
c State            Grace Prd
---------- ---------- ----------- ----------------- ------------------------ --
------------------ ---------
         1         26 Enabled     Indeterminate                              Li
cense Ok                    0

Fcoe Member Port:

Port-channel Slot  Port  Oper State      State Reason
------------ ----- ----- --------------- ------------
1                1    10 Sfp Not Present Unknown
1                1     3 Sfp Not Present Unknown
1                1     4 Sfp Not Present Unknown
1                1     6 Sfp Not Present Unknown
1                1     8 Sfp Not Present Unknown
2                1     7 Sfp Not Present Unknown
UCS-A /fc-uplink/fabric #
```

# Unconfiguring a FCoE Uplink Port

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | UCS-A# **scope fc-uplink** | Enters FC Uplink mode. |
| **Step 2** | UCS-A /fc-uplink # **scope fabric** {**a** \| **b**} | Enters FC - Uplink mode for the specific fabric. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 3** | UCS-A /fc-uplink/fabric # **delete fcoeinterface** *slot-numberport-number* | Deletes the specified interface. |
| **Step 4** | UCS-A /fc-uplink/fabric/fabricinterface # **commit-buffer** | Commits the transaction to the system configuration. |

### Example

The following example deletes the FCoE uplink interface on port 1 on slot 8 of fabric A and commits the transaction:

```
UCS-A# scope fc-uplink
UCS-A /fc-uplink # scope fabric a
UCS-A /fc-uplink/fabric # delete fcoeinterface 1 8
UCS-A /fc-uplink/fabric/fcoeinterface* # commit-buffer
UCS-A /fc-uplink/fabric/fcoeinterface #
```

# FCoE and Fibre Channel Storage Ports

## Configuring a Fibre Channel Storage or FCoE Port

### Procedure

**Step 1** UCS-A# **scope fc-storage**

Enters Fibre Channel storage mode.

**Step 2** UCS-A /fc-storage # **scope fabric** {**a** | **b**}

Enters Fibre Channel storage mode for the specified fabric.

**Step 3** UCS-A /fc-storage/fabric # **create interface** {**fc** | **fcoe**} *slot-num port-num*

Creates an interface for the specified Fibre Channel storage port.

On Cisco UCS 6454 Fabric Interconnects, ports 49-54 cannot be configured as FCoE storage ports.

**Step 4** UCS-A /fc-storage/fabric # **commit-buffer**

Commits the transaction.

### Example

The following example creates an interface for Fibre Channel storage port 10 on slot 2 of fabric A and commits the transaction:

```
UCS-A# scope fc-storage
UCS-A /fc-storage # scope fabric a
UCS-A /fc-storage/fabric* # create interface fc 2 10
UCS-A /fc-storage/fabric # commit-buffer
```

### What to do next

Assign a VSAN.

# Unconfiguring a Fibre Channel Storage or FCoE Port

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | UCS-A# **scope fc-storage** | Enters Fibre Channel storage mode. |
| **Step 2** | UCS-A /fc-storage # **scope fabric** {**a** \| **b**} | Enters Fibre Channel storage mode for the specified fabric. |
| **Step 3** | UCS-A /fc-storage/fabric # **delete interface** {**fc** \| **fcoe**} *slot-num port-num* | Deletes the interface for the specified Fibre Channel or FCoE storage port. |
| **Step 4** | UCS-A /fc-storage/fabric # **commit-buffer** | Commits the transaction. |

### Example

The following example unconfigures Fibre Channel storage port 10 on slot 2 of fabric A and commits the transaction:

```
UCS-A# scope fc-storage
UCS-A /fc-storage # scope fabric a
UCS-A /fc-storage/fabric* # delete interface fc 2 10
UCS-A /fc-storage/fabric # commit-buffer
```

# Restoring a Fibre Channel Storage Port Back to an Uplink Fibre Channel Port

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | UCS-A# **scope fc-uplink** | Enters Fibre Channel uplink mode. |
| **Step 2** | UCS-A /fc-uplink # **scope fabric** {**a** \| **b**} | Enters Fibre Channel uplink mode for the specified fabric. |
| **Step 3** | UCS-A /fc-uplink/fabric # **create interface** *slot-num port-num* | Creates an interface for the specified Fibre Channel uplink port. |
| **Step 4** | UCS-A /fc-uplink/fabric # **commit-buffer** | Commits the transaction. |

| | Command or Action | Purpose |
|---|---|---|
| | | **Note** During the initial bring-up of a Fibre Channel (FC) uplink, such as when enabling a port or after a Fabric Interconnect reboot, it is normal to observe transient discards or cyclic redundancy check (CRC) errors on the peer interface, such as a Cisco MDS switch. These errors may occur during the link negotiation and stabilization process. If the error count stops incrementing once the uplink is operational and normal traffic is flowing, these transient errors are considered normal behavior and do not require further action. |

### Example

The following example creates an interface for Fibre Channel uplink port 10 on slot 2 of fabric A and commits the transaction:

```
UCS-A# scope fc-uplink
UCS-A /fc-uplink # scope fabric a
UCS-A /fc-uplink/fabric* # create interface 2 10
UCS-A /fc-uplink/fabric # commit-buffer
```

# Appliance Port Channels

An appliance port channel allows you to group several physical appliance ports to create one logical Ethernet storage link for the purpose of providing fault-tolerance and high-speed connectivity. In Cisco UCS Manager, you create a port channel first and then add appliance ports to the port channel. You can add up to eight appliance ports to a port channel.

## Configuring an Appliance Port Channel

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | UCS-A# **scope eth-storage** | Enters Ethernet storage mode. |
| **Step 2** | UCS-A /eth-storage # **scope fabric** {**a** \| **b** } | Enters Ethernet storage fabric mode for the specified fabric. |
| **Step 3** | UCS-A /eth-storage/fabric # **create port-channel** *port-num* | Creates a port channel on the specified Ethernet storage port, and enters Ethernet storage fabric port channel mode. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 4** | (Optional) UCS-A /eth-storage/fabric/port-channel # {**enable** \| **disable**} | Enables or disables the administrative state of the port channel. The port channel is disabled by default. |
| **Step 5** | (Optional) UCS-A /eth-storage/fabric/port-channel # **set name** *port-chan-name* | Specifies the name for the port channel. |
| **Step 6** | (Optional) UCS-A /eth-storage/fabric/port-channel # **set pingroupname** *pin-group name* | Specifies the appliance pin target to the specified fabric and port, or fabric and port channel. |
| **Step 7** | (Optional) UCS-A /eth-storage/fabric/port-channel # **set portmode** {**access** \| **trunk**} | Specifies whether the port mode is access or trunk. By default, the mode is set to trunk. |
| **Step 8** | (Optional) UCS-A /eth-storage/fabric/port-channel # **set prio** *sys-class-name* | Specifies the QoS class for the appliance port. By default, the priority is set to best-effort.<br>The sys-class-name argument can be one of the following class keywords:<br>• **FC**—Use this priority for QoS policies that control vHBA traffic only.<br>• **Platinum**—Use this priority for QoS policies that control vNIC traffic only.<br>• **Gold**—Use this priority for QoS policies that control vNIC traffic only.<br>• **Silver**—Use this priority for QoS policies that control vNIC traffic only.<br>• **Bronze**—Use this priority for QoS policies that control vNIC traffic only.<br>• **Best Effort**—Do not use this priority. It is reserved for the Basic Ethernet traffic lane. If you assign this priority to a QoS policy and configure another system class as CoS 0, Cisco UCS Manager does not default to this system class. It defaults to the priority with CoS 0 for that traffic. |
| **Step 9** | (Optional) UCS-A /eth-storage/fabric/port-channel # **set speed** {**1gbps** \| **2gbps** \| **4gbps** \| **8gbps** \| **auto**} | Specifies the speed for the port channel. |
| **Step 10** | UCS-A /eth-storage/fabric/port-channel # **commit-buffer** | Commits the transaction to the system configuration. |

### Example

The following example creates a port channel on port 13 of fabric A and commits the transaction:

```
UCS-A# scope eth-storage
UCS-A /eth-storage # scope fabric a
UCS-A /eth-storage/fabric # create port-channel 13
UCS-A /eth-storage/fabric/port-channel* # enable
UCS-A /eth-storage/fabric/port-channel* # set name portchan13a
UCS-A /eth-storage/fabric/port-channel* # set pingroupname pingroup1
UCS-A /eth-storage/fabric/port-channel* # set portmode access
UCS-A /eth-storage/fabric/port-channel* # set prio fc
UCS-A /eth-storage/fabric/port-channel* # set speed 2gbps
UCS-A /eth-storage/fabric/port-channel* # commit-buffer
UCS-A /eth-storage/fabric/port-channel #
```

# Unconfiguring an Appliance Port Channel

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | UCS-A# **scope eth-storage** | Enters Ethernet storage mode. |
| **Step 2** | UCS-A /eth-storage # **scope fabric** {**a** \| **b** } | Enters Ethernet storage fabric mode for the specified fabric. |
| **Step 3** | UCS-A /eth-storage/fabric # **delete port-channel** *port-num* | Deletes the port channel from the specified Ethernet storage port. |
| **Step 4** | UCS-A /eth-storage/fabric # **commit-buffer** | Commits the transaction to the system configuration. |

### Example

The following example unconfigures the port channel on port 13 of fabric A and commits the transaction:

```
UCS-A# scope eth-storage
UCS-A /eth-storage # scope fabric a
UCS-A /eth-storage/fabric # delete port-channel 13
UCS-A /eth-storage/fabric* # commit-buffer
UCS-A /eth-storage/fabric #
```

# Enabling or Disabling an Appliance Port Channel

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | UCS-A# **scope eth-storage** | Enters Ethernet storage mode. |
| **Step 2** | UCS-A /eth-storage # **scope fabric** {**a** \| **b** } | Enters Ethernet storage mode for the specified fabric. |
| **Step 3** | UCS-A /eth-storage/fabric # **scope port-channel** *port-chan-name* | Enters Ethernet storage port channel mode. |
| **Step 4** | UCS-A /eth-storage/fabric/port-channel # {**enable** \| **disable** } | Enables or disables the administrative state of the port channel. The port channel is disabled by default. |
| **Step 5** | UCS-A /eth-storage/fabric/port-channel # **commit-buffer** | Commits the transaction to the system configuration. |

### Example

The following example enables port channel 13 on fabric A and commits the transaction:

```
UCS-A# scope eth-storage
UCS-A /eth-storage # scope fabric a
UCS-A /eth-storage/fabric # scope port-channel 13
UCS-A /eth-storage/fabric/port-channel* # enable
UCS-A /eth-storage/fabric/port-channel* # commit-buffer
UCS-A /eth-storage/fabric/port-channel #
```

# Adding a Member Port to an Appliance Port Channel

### Procedure

**Step 1** UCS-A# **scope eth-storage**

Enters Ethernet storage mode.

**Step 2** UCS-A /eth-storage # **scope fabric** {**a** | **b** }

Enters Ethernet storage fabric mode for the specified fabric.

**Step 3** UCS-A /eth-storage/fabric # **scope port-channel** *port-num*

Enters Ethernet storage fabric port channel mode for the specified port channel.

**Step 4** UCS-A /eth-storage/fabric/port-channel # **create member-port** *slot-num port-num*

Creates the specified member port from the port channel and enters Ethernet storage fabric port channel member port mode.

**Step 5** UCS-A /eth-storage/fabric/port-channel # **commit-buffer**

Commits the transaction to the system configuration.

### Example

The following example adds the member port on slot 1, port 7 to the port channel on port 13 of fabric A and commits the transaction.

```
UCS-A# scope eth-storage
UCS-A /eth-storage # scope fabric a
UCS-A /eth-storage/fabric # scope port-channel 13
UCS-A /eth-storage/fabric/port-channel # create member-port 1 7
UCS-A /eth-storage/fabric/port-channel* # commit-buffer
UCS-A /eth-storage/fabric/port-channel #
```

# Deleting a Member Port from an Appliance Port Channel

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | UCS-A# **scope eth-storage** | Enters Ethernet storage mode. |
| **Step 2** | UCS-A /eth-storage # **scope fabric** {**a** \| **b** } | Enters Ethernet storage fabric mode for the specified fabric. |
| **Step 3** | UCS-A /eth-storage/fabric # **scope port-channel** *port-num* | Enters Ethernet storage fabric port channel mode for the specified port channel. |
| **Step 4** | UCS-A /eth-storage/fabric/port-channel # **delete member-port** *slot-num port-num* | Deletes the specified member port from the port channel. |
| **Step 5** | UCS-A /eth-storage/fabric/port-channel # **commit-buffer** | Commits the transaction to the system configuration. |

### Example

The following example deletes a member port from the port channel on port 13 of fabric A and commits the transaction:

```
UCS-A# scope eth-storage
UCS-A /eth-storage # scope fabric a
UCS-A /eth-storage/fabric # scope port-channel 13
UCS-A /eth-storage/fabric/port-channel # delete member-port 1 7
UCS-A /eth-storage/fabric/port-channel* # commit-buffer
```

```
UCS-A /eth-storage/fabric/port-channel #
```

# Fibre Channel Port Channels

A Fibre Channel port channel allows you to group several physical Fibre Channel ports (link aggregation) to create one logical Fibre Channel link to provide fault-tolerance and high-speed connectivity. In Cisco UCS Manager, you create a port channel first and then add Fibre Channel ports to the port channel.

**Note** Fibre Channel port channels are not compatible with non-Cisco technology.

You can create up to four Fibre Channel port channels in each Cisco UCS domain with Cisco UCS 6200, 6300, and Cisco UCS 6454 Fabric Interconnect Series fabric interconnects. Each Fibre Channel port channel can include a maximum of 16 uplink Fibre Channel ports.

You can create up to two Fibre Channel port channels in each Cisco UCS domain with Cisco UCS 6324 fabric interconnects. Each Fibre Channel port channel can include a maximum of four uplink Fibre Channel ports.

Ensure that the Fibre Channel port channel on the upstream NPIV switch is configured with its channel mode as **active**. If both the member port(s) and peer port(s) do not have the same channel mode configured, the port channel will not come up. When the channel mode is configured as **active**, the member ports initiate port channel protocol negotiation with the peer port(s) regardless of the channel group mode of the peer port. If the peer port, while configured in a channel group, does not support the port channel protocol, or responds with a nonnegotiable status, it defaults to the On mode behavior. The **active** port channel mode allows automatic recovery without explicitly enabling and disabling the port channel member ports at either end.

This example shows how to configure channel mode as active:

```
switch(config)# int po114
switch(config-if)# channel mode active
```

# Configuring a Fibre Channel Port Channel

**Note** If you are connecting two Fibre Channel port channels, the admin speed for both port channels must match for the link to operate. If the admin speed for one or both of the Fibre Channel port channels is set to auto, Cisco UCS adjusts the admin speed automatically.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | UCS-A# **scope fc-uplink** | Enters Fibre Channel uplink mode. |
| **Step 2** | UCS-A /fc-uplink # **scope fabric** {**a** \| **b** } | Enters Fibre Channel uplink fabric mode for the specified fabric. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 3** | UCS-A /fc-uplink/fabric # **create port-channel** *port-num* | Creates a port channel on the specified Fibre Channel uplink port, and enters Fibre Channel uplink fabric port channel mode. |
| **Step 4** | (Optional) UCS-A /fc-uplink/fabric/port-channel # {**enable** \| **disable**} | Enables or disables the administrative state of the port channel. The port channel is disabled by default. |
| **Step 5** | (Optional) UCS-A /fc-uplink/fabric/port-channel # **set name** *port-chan-name* | Specifies the name for the port channel. |
| **Step 6** | (Optional) UCS-A /fc-uplink/fabric/port-channel # **set speed** {**1gbps** \| **2gbps** \| **4gbps** \| **8gbps** \| **auto**} | Specifies the speed for the port channel. |
| **Step 7** | UCS-A /fc-uplink/fabric/port-channel # **commit-buffer** | Commits the transaction to the system configuration. |

### Example

The following example creates port channel 13 on fabric A, sets the name to portchan13a, enables the administrative state, sets the speed to 2 Gbps, and commits the transaction:

```
UCS-A# scope fc-uplink
UCS-A /fc-uplink # scope fabric a
UCS-A /fc-uplink/fabric # create port-channel 13
UCS-A /fc-uplink/fabric/port-channel* # enable
UCS-A /fc-uplink/fabric/port-channel* # set name portchan13a
UCS-A /fc-uplink/fabric/port-channel* # set speed 2gbps
UCS-A /fc-uplink/fabric/port-channel* # commit-buffer
UCS-A /fc-uplink/fabric/port-channel #
```

# Unconfiguring a Fibre Channel Port Channel

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | UCS-A# **scope fc-uplink** | Enters Fibre Channel uplink mode. |
| **Step 2** | UCS-A /fc-uplink # **scope fabric** {**a** \| **b** } | Enters Fibre Channel uplink fabric mode for the specified fabric. |
| **Step 3** | UCS-A /fc-uplink/fabric # **delete port-channel** *port-num* | Deletes the port channel on the specified Fibre Channel uplink port. |
| **Step 4** | UCS-A /fc-uplink/fabric # **commit-buffer** | Commits the transaction to the system configuration. |

### Example

The following example unconfigures port channel 13 on fabric A and commits the transaction:

```
UCS-A# scope fc-uplink
UCS-A /fc-uplink # scope fabric a
UCS-A /fc-uplink/fabric # delete port-channel 13
UCS-A /fc-uplink/fabric* # commit-buffer
UCS-A /fc-uplink/fabric #
```

# Adding Channel Mode Active To The Upstream NPIV Fibre Channel Port Channel

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | UCS-A# **scope fc-uplink** | Enters Fibre Channel uplink mode. |
| **Step 2** | UCS-A /fc-uplink # **scope fabric** {**a** \| **b** } | Enters Fibre Channel uplink fabric mode for the specified fabric. |
| **Step 3** | UCS-A /fc-uplink/fabric # **create port-channel** *port-num* | Creates a port channel on the specified Fibre Channel uplink port, and enters Fibre Channel uplink fabric port channel mode. |
| **Step 4** | (Optional) UCS-A /fc-uplink/fabric/port-channel # {**enable** \| **disable**} | Enables or disables the administrative state of the port channel. The port channel is disabled by default. |
| **Step 5** | (Optional) UCS-A /fc-uplink/fabric/port-channel # **set name** *port-chan-name* | Specifies the name for the port channel. |
| **Step 6** | (Optional) UCS-A /fc-uplink/fabric/port-channel # **scope** *port-chan-name* | Specifies the name for the port channel. |
| **Step 7** | (Optional) UCS-A /fc-uplink/fabric/port-channel # **channel mode** {**active**} | Configures the channel-mode active on the upstream NPIV switch. |
| **Step 8** | UCS-A /fc-uplink/fabric/port-channel # **commit-buffer** | Commits the transaction to the system configuration. |

### Example

The following example enables channel mode to active:

```
UCS-A# scope fc-uplink
UCS-A /fc-uplink # scope fabric a
UCS-A /fc-uplink/fabric # create port-channel 13
UCS-A /fc-uplink/fabric/port-channel* # enable
UCS-A /fc-uplink/fabric/port-channel* # set name portchan13a
UCS-A /fc-uplink/fabric/port-channel* # channel mode active
UCS-A /fc-uplink/fabric/port-channel* # commit-buffer
UCS-A /fc-uplink/fabric/port-channel # exit
UCS-A /fc-uplink/fabric/ # show port-channel database

portchan13a
    Administrative channel mode is active
    Operational channel mode is active

UCS-A /fc-uplink/fabric/ #
```

# Enabling or Disabling a Fibre Channel Port Channel

### Procedure

**Step 1** UCS-A# **scope fc-uplink**

Enters Fibre Channel uplink mode.

**Step 2** UCS-A /fc-uplink # **scope fabric** {**a** | **b** }

Enters Fibre Channel uplink mode for the specified fabric.

**Step 3** UCS-A /fc-uplink/fabric # **scope port-channel** *port-chan-name*

Enters Fibre Channel uplink port channel mode.

**Step 4** UCS-A /fc-uplink/fabric/port-channel # {**enable** | **disable** }

Enables or disables the administrative state of the port channel. The port channel is disabled by default.

### Example

The following example enables port channel 13 on fabric A and commits the transaction:

```
UCS-A# scope fc-uplink
UCS-A /fc-uplink # scope fabric a
UCS-A /fc-uplink/fabric # scope port-channel 13
UCS-A /fc-uplink/fabric/port-channel* # enable
UCS-A /fc-uplink/fabric/port-channel* # commit-buffer
UCS-A /fc-uplink/fabric/port-channel #
```

# Adding a Member Port to a Fibre Channel Port Channel

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | UCS-A# **scope fc-uplink** | Enters Fibre Channel uplink mode. |
| **Step 2** | UCS-A /fc-uplink # **scope fabric** {**a** \| **b** } | Enters Fibre Channel uplink fabric mode for the specified fabric. |
| **Step 3** | UCS-A /fc-uplink/fabric # **scope port-channel** *port-num* | Enters Fibre Channel uplink fabric port channel mode for the specified port channel. |
| **Step 4** | UCS-A /fc-uplink/fabric/port-channel # **create member-port** *slot-num port-num* | Creates the specified member port from the port channel and enters Fibre Channel uplink fabric port channel member port mode. |
| **Step 5** | UCS-A /fc-uplink/fabric/port-channel # **commit-buffer** | Commits the transaction to the system configuration. |

### Example

The following example adds the member port on slot 1, port 7 to port channel 13 on fabric A and commits the transaction.

```
UCS-A# scope fc-uplink
UCS-A /fc-uplink # scope fabric a
UCS-A /fc-uplink/fabric # scope port-channel 13
UCS-A /fc-uplink/fabric # create member-port 1 7
UCS-A /fc-uplink/fabric/port-channel* # commit-buffer
UCS-A /fc-uplink/fabric/port-channel #
```

# Deleting a Member Port from a Fibre Channel Port Channel

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | UCS-A# **scope fc-uplink** | Enters Fibre Channel uplink mode. |
| **Step 2** | UCS-A /fc-uplink # **scope fabric** {**a** \| **b**} | Enters Fibre Channel uplink fabric mode for the specified fabric. |
| **Step 3** | UCS-A /fc-uplink/fabric # **scope port-channel** *port-num* | Enters Fibre Channel uplink fabric port channel mode for the specified port channel. |
| **Step 4** | UCS-A /fc-uplink/fabric/port-channel # **delete member-port** *slot-num port-num* | Deletes the specified member port from the port channel. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 5** | UCS-A /fc-uplink/fabric/port-channel # **commit-buffer** | Commits the transaction to the system configuration. |

### Example

The following example deletes a member port from port channel 13 on fabric A and commits the transaction:

```
UCS-A# scope fc-uplink
UCS-A /fc-uplink # scope fabric a
UCS-A /fc-uplink/fabric # scope port-channel 13
UCS-A /fc-uplink/fabric # delete member-port 1 7
UCS-A /fc-uplink/fabric/port-channel* # commit-buffer
UCS-A /fc-uplink/fabric/port-channel #
```

# Configuring Organizationally Unique Identifier

Cisco Fiber Channel (FC) port channels are dependent on the Organizationally Unique Identifiers (OUIs) of the devices at each end of the port channels for successful configuration. When new devices are released or new OUI ranges are assigned to existing devices due to OUI pool depletion, the new OUIs need to be added to their respective OUI tables for the port channel to be configured successfully.

### Adding an OUI

To establish FC port-channels with new Cisco FC devices or devices with newly assigned OUI ranges, you can manually add OUIs into the database using the following command through Cisco UCSM CLI:

```
FI-A # sc fabric-interconnect {a|b}
FI-A /fabric-interconnect # sc oui-pool default
FI-A /fabric-interconnect/oui-pool # sh oui
FI-A /fabric-interconnect/oui-pool # create oui [oui-id]
FI-A /fabric-interconnect/oui-pool/oui* # commit-buffer
```

Where, `oui-id` is the new OUI of the device that needs to be added. The device OUI must be an eight digit hexadecimal number. The valid range of OUI is from 0x000000 to 0xffffff. For example, 0xabcdef.

### Viewing OUIs

To view the list of OUIs, run the following command:

```
FI-A /fabric-interconnect/oui-pool# show oui
```

The following example shows the sample output of the `show oui` command:

```
FI-A /fabric-interconnect/oui-pool# show oui

OUI Entry:
Oui
---
0x0001ac
0x1b0000
0xaabbcc
0xddeeff
```

### Deleting an OUI

To delete an OUI, run the following command:

```
FI-A /fabric-interconnect/oui-pool# delete ouientry [oui-id]
```

Where, oui-id is the OUI of the device that needs to be deleted.

# FCoE Port Channels

An FCoE port channel allows you to group several physical FCoE ports to create one logical FCoE port channel. At a physical level, the FCoE port channel carries FCoE traffic over an Ethernet port channel. So an FCoE port channel with a set of members is essentially an Ethernet port channel with the same members. This Ethernet port channel is used as a physical transport for FCoE traffic.

For each FCoE port channel, Cisco UCS Manager creates a VFC internally and binds it to an Ethernet port channel. FCoE traffic received from the hosts is sent over the VFC the same way as the FCoE traffic is sent over Fibre Channel uplinks.

## Configuring a FCoE Port Channel

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | UCS-A# **scope fc-uplink** | Enters FC Uplink mode. |
| **Step 2** | UCS-A /fc-uplink # **scope fabric** {**a** \| **b**} | Enters FC - Uplink mode for the specific fabric. |
| **Step 3** | UCS-A /fc-uplink/fabric # **create fcoe-port-channel** *number* | Creates port channel for the specified FCoE uplink port. |
| **Step 4** | UCS-A /fc-uplink/fabric/fabricinterface # **commit-buffer** | Commits the transaction to the system configuration. |

### Example

The following example creates an interface for FCoE uplink port 1 on slot 4 of fabric A and commits the transaction:

```
UCS-A# scope fc-uplink
UCS-A /fc-uplink # scope fabric a
UCS-A /fc-uplink/fabric # create fcoe-port-channel 4
UCS-A /fc-uplink/fabric/fcoe-port-channel* # commit-buffer
UCS-A /fc-uplink/fabric/fcoe-port-channel #
```

# Adding a Member Port to a FCoE Uplink Port Channel

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | UCS-A# **scope fc-uplink** | Enters Fibre Channel uplink mode. |
| **Step 2** | UCS-A /fc-uplink # **scope fabric** {**a** \| **b** } | Enters Fibre Channel uplink fabric mode for the specified fabric. |
| **Step 3** | UCS-A /fc-uplink/fabric # **scope fcoe-port-channel** *ID* | Enters FCoE uplink port channel mode for the specified port channel. |
| **Step 4** | UCS-A /fc-uplink/fabric/fcoe-port-channel # **create member-port** *slot-num port-num* | Creates the specified member port from the port channel and enters FCoE uplink fabric port channel member port mode.<br>**Note**<br>If the FCoE uplink port channel is a unified uplink port channel, you will get the following message:<br>Warning: if this is a unified port channel then member will be added to the ethernet port channel of the same id as well. |
| **Step 5** | UCS-A /fc-uplink/fabric/fcoe-port-channel # **commit-buffer** | Commits the transaction to the system configuration. |

### Example

The following example adds the member port on slot 1, port 7 to FCoE port channel 13 on fabric A and commits the transaction.

```
UCS-A# scope fc-uplink
UCS-A /fc-uplink # scope fabric a
UCS-A /fc-uplink/fabric # scope fcoe-port-channel 13
UCS-A /fc-uplink/fabric # create member-port 1 7
UCS-A /fc-uplink/fabric/fcoe-port-channel* # commit-buffer
UCS-A /fc-uplink/fabric/fcoe-port-channel #
```

# Adapter Port Channels

An adapter port channel groups into one logical link all the physical links going from a Cisco UCS Virtual Interface Card (VIC) into an I/O.

Adapter port channels are created and managed internally by Cisco UCS Manager when it detects that the correct hardware is present. Adapter port channels cannot be configured manually. Adapter port channels are viewable using the Cisco UCS Manager GUI or the Cisco UCS Manager CLI.

# Viewing Adapter Port Channels

### Procedure

**Step 1** UCS-A# **scope chassis** *chassis-num*

Enters chassis mode for the specified chassis.

**Step 2** UCS-A /chassis # **scope iom** {**a** **b**}

Enters chassis IOM mode for the specified IOM.

**Step 3** UCS-A /chassis/iom # **scope port group**

Enters port group mode for the specified port group.

**Step 4** UCS-A /chassis/iom/port group # **show host-port-channel** [**detail** | **expand**]

Displays the adapter port channels on the specified chassis.

### Example

This following example shows how to display information on host port channels within a port group mode:

```
UCS-A # scope chassis 1
UCS-A /chassis # scope iom a
UCS-A /chassis/iom # scope port group
UCS-A /chassis/iom/port group # show host-port-channel

Host Port channel:

  Port Channel Id Fabric ID Oper State       State Reason
    --------------- --------- ---------------- ------------
                1289 B         Up
                1290 B         Up
                1306 B         Up
                1307 B         Up
                1309 B         Up
                1315 B         Up

UCS-A /chassis/iom/port group #
```

# Event Detection and Action

Cisco UCS Manager uses the statistics collection policy to monitor and trigger an alarm when there are faults in the network interface ports connected from the I/O module (IOM) to the fabric interconnect.

The error statistics for the network interface ports is called NiErrStats and consists of the following errors:

| NiErrStats Error Name | Description |
|---|---|
| frameTx | Collects the TX_FRM_ERROR counter values. |
| tooLong | Collects the RX_TOOLONG counter values. |
| tooShort | Collects the sum of RX_UNDERSIZE and RX_FRAGMENT counter values. |
| Crc | Collects the sum of RX_CRERR_NOT_STOMPED and RX_CRCERR_STOMPED counter values. |
| inRange | Collects the RX_INRANGEERR counter values. |

**Note** The network interface port statistics is collected only from active ports and the information is sent to Cisco UCS Manager.

# Policy-Based Port Error Handling

If Cisco UCS Manager detects any errors on active NI ports, and if the error-disable feature is enabled, Cisco UCS Manager automatically disables the respective FI port that is connected to the NI port that had errors. When a FI port is error disabled, it is effectively shut down and no traffic is sent or received on that port.

The error-disable function serves two purposes:

- It lets you know which FI port is error-disabled and that the connected NI Port has errors.
- It eliminates the possibility that this port can cause other ports, which are connected to the same Chassis/FEX, to fail. Such a failure can occur when the NI port has errors, which can ultimately cause serious network issues. The error-disable function helps prevent these situations.

# Creating Threshold Definition

**Procedure**

**Step 1** UCS-A # **scope eth-server**

Enters Ethernet storage mode.

**Step 2** UCS-A/eth-server # **scope stats-threshold-policy default**

Enters statistics threshold policy mode.

**Step 3** UCSA/eth-server/stats-threshold-policy # **create class** *class-name*

Creates the specified statistics threshold policy class and enters the organization statistics threshold policy class mode. To see a list of the available class name keywords, enter the **create class ?** command in organization threshold policy mode.

**Step 4** UCS-A/eth-server/stats-threshold-policy/class # **create property** *property-name*

Creates the specified statistics threshold policy class property and enters the organization statistics threshold policy class property mode. To see a list of the available property name keywords, enter the **create property ?** command in organization threshold policy class mode.

**Step 5** UCS-A/eth-server/stats-threshold-policy/class/property # **set normal-value** *value*

Specifies the normal value for the class property. The *value* format can vary depending on the class property being configured. To see the required format, enter the **set normal-value ?** command in organization statistics threshold policy class property mode.

**Step 6** UCS-A/eth-server/stats-threshold-policy/class/property # **create threshold-value** {*above-normal* | *below-normal*} {*cleared* | *condition* | *critical* | *info* | *major* | *minor* | *warning*}

Creates the specified threshold value for the class property and enters the organization statistics threshold policy class property threshold value mode.

**Step 7** UCS-A/eth-server/stats-threshold-policy/class/property/threshold-value # **set** {**deescalating** | **escalating**} *value*

Specifies the deescalating and escalating class property threshold value. The *value* format can vary depending on the class property threshold value being configured. To see the required format, enter the **set deescalating ?** or **set escalating ?** command in the organization statistics threshold policy class property threshold value mode.

**Step 8** UCS-A/eth-server/stats-threshold-policy/class/property/threshold-value # **commit-buffer**

Commits the transaction to the system configuration.

### Example

The following example shows how to create a threshold definition:

```
UCS-A # scope eth-server
UCS-A /eth-server # scope stats-threshold-policy default
UCS-A /eth-server/stats-threshold-policy # create class ni-ether-error-stats
UCS-A /eth-server/stats-threshold-policy/class* # create property crc-delta
UCS-A /eth-server/stats-threshold-policy/class/property* # set normal-value 0
UCS-A /eth-server/stats-threshold-policy/class/property* # create threshold-value above-normal
 major
UCS-A /eth-server/stats-threshold-policy/class/property/threshold-value* # set escalating
5
UCS-A /eth-server/stats-threshold-policy/class/property/threshold-value* # set deescalating
 3
UCS-A /eth-server/stats-threshold-policy/class/property/threshold-value* # commit-buffer
```

# Configuring Error Disable on a Fabric Interconnect Port

### Procedure

**Step 1** UCS-A # **scope eth-server**

Enters Ethernet storage mode.

**Step 2** UCS-A/eth-server # **scope stats-threshold-policy default**

Enters statistics threshold policy mode.

**Step 3** UCSA/eth-server/stats-threshold-policy # **scope class** *class-name*

Enters the organization statistics threshold policy class mode for the specified statistics threshold policy class.

**Step 4** UCS-A/eth-server/stats-threshold-policy/class # **scope property** *property-name*

Enters the organization statistics threshold policy class property mode for the specified statistics threshold policy class property.

**Step 5** UCS-A/eth-server/stats-threshold-policy/class/property # **set error-disable-fi-port** {**yes** | **no**}

Specifies the error disable state for the class property.

Use the **no** option to disable error disable for the class property.

**Step 6** UCS-A/eth-server/stats-threshold-policy/class/property* # **commit-buffer**

Commits the transaction to the system configuration.

### Example

The following example shows how to enable error disable on an FI port:

```
UCS-A # scope eth-server
UCS-A /eth-server # scope stats-threshold-policy default
UCS-A /eth-server/stats-threshold-policy # scope class ni-ether-error-stats
UCS-A /eth-server/stats-threshold-policy/class # scope property crc-delta
UCS-A /eth-server/stats-threshold-policy/class/property # set error-disable-fi-port yes
UCS-A /eth-server/stats-threshold-policy/class/property* # commit-buffer
```

# Configuring Auto Recovery on a Fabric Interconnect Port

### Procedure

**Step 1** UCS-A # **scope eth-server**

Enters Ethernet storage mode.

**Step 2** UCS-A/eth-server # **scope stats-threshold-policy default**

Enters statistics threshold policy mode.

**Step 3** UCSA/eth-server/stats-threshold-policy # **scope class** *class-name*

Enters the organization statistics threshold policy class mode for the specified statistics threshold policy class.

**Step 4** UCS-A/eth-server/stats-threshold-policy/class # **scope property** *property-name*

Enters the organization statistics threshold policy class property mode for the specified statistics threshold policy class property.

**Step 5** UCS-A/eth-server/stats-threshold-policy/class/property # **set auto-recovery** {**enabled** | **disabled**}

Specifies the auto recovery state for the class property.

Use the **disabled** option to disable auto recovery for the class property.

**Step 6** UCS-A/eth-server/stats-threshold-policy/class/property* # **set auto-recovery-time** *time*

Specifies the time in minutes after which the port is automatically re-enabled. The auto recovery time can range from 0 minutes to 4294967295 minutes.

**Step 7** UCS-A/eth-server/stats-threshold-policy/class/property* # **commit-buffer**

Commits the transaction to the system configuration.

### Example

The following example shows how to configure auto recovery on an FI port:

```
UCS-A # scope eth-server
UCS-A /eth-server # scope stats-threshold-policy default
UCS-A /eth-server/stats-threshold-policy # scope class ni-ether-error-stats
UCS-A /eth-server/stats-threshold-policy/class # scope property crc-delta
UCS-A /eth-server/stats-threshold-policy/class/property # set auto-recovery enabled
UCS-A /eth-server/stats-threshold-policy/class/property* # set auto-recovery-time 5
UCS-A /eth-server/stats-threshold-policy/class/property* # commit-buffer
```

# Viewing the Network Interface Port Error Counters

### Procedure

**Step 1** UCS-A # **scope chassis** *chassis-num*

Enters chassis mode for the specified chassis.

**Step 2** UCS-A/chassis # **scope iom** {**a** | **b**}

Enters chassis IOM mode for the specified IOM.

**Step 3** UCS-A/chassis/iom # **scope port-group fabric**

Enters the network interface port.

**Step 4** UCS-A/chassis/iom/port-group # **scope fabric-if** *fabric-if number*

Enters the specified network interface port number.

**Step 5** UCS-A/chassis/iom/port-group/fabric-if # **show stats**

Displays the error counters for the network interface port.

### Example

The following example shows how to display the statistics for the network interface ports:

```
UCS-A # scope chassis 1
UCS-A/chassis # scope iom a
UCS-A/chassis/iom # scope port-group fabric
UCS-A/chassis/iom/port-group # scope faric-if 1
UCS-A/chassis/iom/port-group/fabric-if # show stats
NI Ether Error Stats:
Time Collected: 2014-08-20T15:37:24:688
Monitored Object: sys/chassis-1/slot-1/fabric/port-1/ni-err-stats
Suspect: Yes
Crc (errors): 5000
Frame Tx (errors): 0
Too Long (errors): 0
Too Short (errors): 0
In Range (errors): 0
Thresholded: 0
```

# Fabric Port Channels

Fabric port channels allow you to group several of the physical links from an IOM and IFM (IOM for Cisco UCS X-Series Servers) to a fabric interconnect into one logical link for redundancy and bandwidth sharing. As long as one link in the fabric port channel remains active, the fabric port channel continues to operate.

If the correct hardware is connected, fabric port channels are created by Cisco UCS Manager in the following ways:

- During chassis discovery according to the settings configured in the chassis discovery policy.
- After chassis discovery according to the settings configured in the chassis connectivity policy for a specific chassis.

For each IOM and IFM (IOM for Cisco UCS X-Series Servers) there is a single fabric port channel. Each uplink connecting an IOM and IFM (IOM for Cisco UCS X-Series Servers) to a fabric interconnect can be configured as a discrete link or included in the port channel, but an uplink cannot belong to more than one fabric port channel. For example, if a chassis with two IOMs is discovered and the chassis discovery policy is configured to create fabric port channels, Cisco UCS Manager creates two separate fabric port channels: one for the uplinks connecting IOM-1 and another for the uplinks connecting IOM-2. No other chassis can

join these fabric port channels. Similarly, uplinks belonging to the fabric port channel for IOM-1 cannot join the fabric port channel for IOM-2.

# Load Balancing Over Ports

Load balancing traffic among ports between IOMs and fabric interconnects uses the following criteria for hashing.

- For Ethernet traffic:

  Layer 2 source and destination address

  Layer 3 source and destination address

  Layer 4 source and destination ports

- For FCoE traffic:

  Layer 2 source and destination address

  Source and destination IDs (SID and DID) and Originator Exchange ID (OXID)

In this example, a 2200 Series IOM module is verified by connecting iom *X* (where *X* is the chassis number).

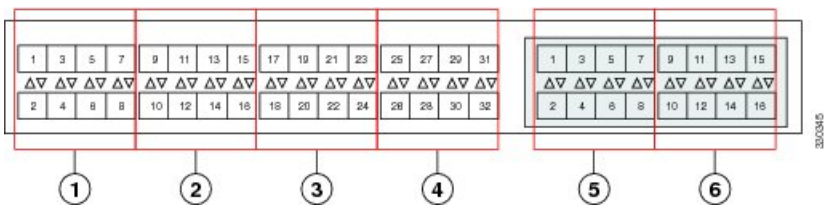
```
show platform software fwmctrl nifport
(....)
 Hash Parameters:
   l2_da: 1 l2_sa: 1 l2_vlan: 0
   l3_da: 1 l3_sa: 1
   l4_da: 1 l4_sa: 1
   FCoE l2_da: 1 l2_sa: 1 l2_vlan: 0
   FCoE l3_did: 1 l3_sid: 1 l3_oxid: 1
```

# Cabling Considerations for Fabric Port Channels

When you configure the links between the Cisco UCS 2200 Series FEX and a Cisco UCS 6200 series fabric interconnect in fabric port channel mode, the available virtual interface namespace (VIF) on the adapter varies depending on where the FEX uplinks are connected to the fabric interconnect ports.

Inside the 6248 fabric interconnect there are six sets of eight contiguous ports, with each set of ports managed by a single chip. When all uplinks from an FEX are connected to a set of ports managed by a single chip, Cisco UCS Manager maximizes the number of VIFs used in service profiles deployed on the blades in the chassis. If uplink connections from an IOM are distributed across ports managed by separate chips, the VIF count is decreased.

***Figure 2: Port Groups for Fabric Port Channels***



**Caution** Adding a second link to a fabric-port-channel port group is disruptive and will automatically increase the available amount of VIF namespace from 63 to 118. Adding further links is not disruptive and the VIF namespace stays at 118.

**Caution** Linking a chassis to two fabric-port-channel port groups does not affect the VIF namespace unless it is manually acknowledged. The VIF namespace is then automatically set to the smaller size fabric port-channel port group usage (either 63 or 118 VIFs) of the two groups.

For high availability cluster-mode applications, we strongly recommend symmetric cabling configurations. If the cabling is asymmetric, the maximum number of VIFs available is the smaller of the two cabling configurations.

For more information on the maximum number of VIFs for your Cisco UCS environment, see the Configuration Limits document for your hardware and software configuration.

# Viewing Fabric Port Channels

### Procedure

**Step 1** UCS-A# **scope eth-server**

Enters Ethernet server mode.

**Step 2** UCS-A /eth-server # **scope fabric** {**a** | **b**}

Enters Ethernet server fabric mode for the specified fabric.

**Step 3** UCS-A /eth-server/fabric # **show fabric-port-channel** [**detail** | **expand**]

Displays fabric port channels on the specified fabric interconnect.

### Example

The following example displays information about configured fabric port channels on fabric interconnect A:

```
UCS-A# scope eth-server
UCS-A /eth-server # scope fabric a
UCS-A /eth-server/fabric # show fabric-port-channel
Fabric Port Channel:
    Port Channel Id Chassis Id Admin State Oper State       State Reason
    --------------- ---------- ----------- ---------------- ------------
               1025 1          Enabled     Failed           No operational members
               1026 2          Enabled     Up

UCS-A /eth-server/fabric #
```

# Enabling or Disabling a Fabric Port Channel Member Port

### Procedure

**Step 1** UCS-A# **scope eth-server**

Enters Ethernet server mode.

**Step 2** UCS-A /eth-server # **scope fabric** {**a** | **b**}

Enters Ethernet server fabric mode for the specified fabric.

**Step 3** UCS-A /eth-server/fabric # **scope fabric-port-channel** *port-chan-id*

Enters Ethernet server fabric, fabric port channel mode for the specified fabric.

**Step 4** UCS-A /eth-server/fabric/fabric-port-channel # **scope member-port** *slot-id port-id*

Enters Ethernet server fabric, fabric port channel mode for the specified member port.

**Step 5** UCS-A /eth-server/fabric/fabric-port-channel # {**enable** | **disable**}

Enables or disables the specified member port.

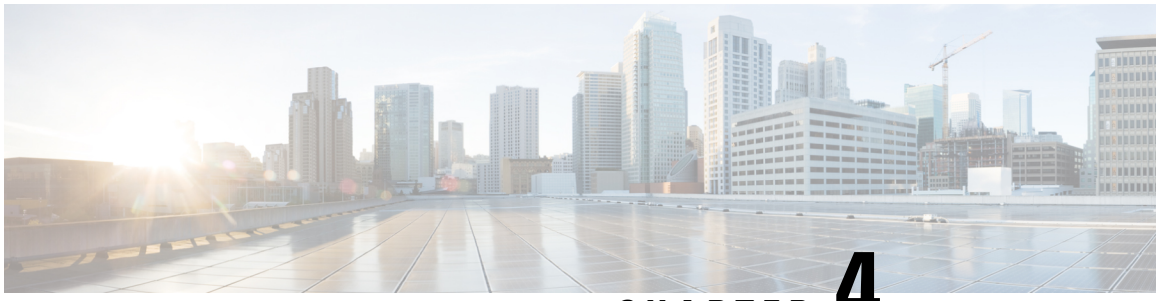
**Step 6** UCS-A /eth-server/fabric/fabric-port-channel # **commit-buffer**

Commits the transaction to the system configuration.

### Example

The following example disables fabric channel member port 1 31 on fabric port channel 1025 and commits the transaction:

```
UCS-A# scope eth-server
UCS-A /eth-server # scope fabric a
UCS-A /eth-server/fabric # scope fabric-port-channel 1025
UCS-A /eth-server/fabric/fabric-port-channel # scope member-port 1 31
UCS-A /eth-server/fabric/fabric-port-channel/member-port # disable
UCS-A /eth-server/fabric/fabric-port-channel/member-port* # commit-buffer
UCS-A /eth-server/fabric/fabric-port-channel/member-port #
```



CHAPTER 4

# Fibre Channel Zoning



# Information About Fibre Channel Zoning

Fibre Channel zoning allows you to partition the Fibre Channel fabric into one or more zones. Each zone defines the set of Fibre Channel initiators and Fibre Channel targets that can communicate with each other in a VSAN. Zoning also enables you to set up access control between hosts and storage devices or user groups.

**Note** Fibre Channel Zoning is not supported on Cisco UCS 6454 Fabric Interconnect

The access and data traffic control provided by zoning does the following:

- Enhances SAN network security
- Helps prevent data loss or corruption
- Reduces performance issues

## Information About Zones

A zone consists of multiple zone members and has the following characteristics:

- Members in a zone can access each other; members in different zones cannot access each other.

- Zones can vary in size.
- Devices can belong to more than one zone.
- A physical fabric can have a maximum of 8,000 zones.

## Information About Zone Sets

Each zone set consists of one or more zones. You can use zone sets to enforce access control within the Fibre Channel fabric. In addition, zone sets provide you with the following advantages:

- Only one zone set can be active at any time.
- All zones in a zone set can be activated or deactivated as a single entity across all switches in the fabric.
- Changes to a zone set are not applied until the zone set has been activated. If you make changes to the active zone set, you must reactivate that zone set to apply the changes.
- A zone can be a member of more than one zone set.
- A switch in a zone can have a maximum of 500 zone sets.

# Support for Fibre Channel Zoning in Cisco UCS Manager

Cisco UCS Manager supports switch-based Fibre Channel zoning and Cisco UCS Manager-based Fibre Channel zoning. You cannot configure a combination of zoning types in the same Cisco UCS domain. You can configure a Cisco UCS domain with one of the following types of zoning:

- Cisco UCS Manager-based Fibre Channel zoning—This configuration combines direct attach storage with local zoning. Fibre Channel or FCoE storage is directly connected to the fabric interconnects and zoning is performed in Cisco UCS Manager, using Cisco UCS local zoning. Any existing Fibre Channel or FCoE uplink connections need to be disabled. Cisco UCS does not currently support active Fibre Channel or FCoE uplink connections coexisting with the utilization of the UCS Local Zoning feature.
- Switch-based Fibre Channel zoning—This configuration combines direct attach storage with uplink zoning. The Fibre Channel or FCoE storage is directly connected to the fabric interconnects and zoning is performed externally to the Cisco UCS domain through an MDS or Nexus 5000 switch. This configuration does not support local zoning in the Cisco UCS domain.

**Note** Zoning is configured on a per-VSAN basis. You cannot enable zoning at the fabric level.

## Cisco UCS Manager-Based Fibre Channel Zoning

With Cisco UCS Manager-based zoning, Cisco UCS Manager controls the Fibre Channel zoning configuration for the Cisco UCS domain, including creating and activating zones for all VSANs that you set up with this type of zoning. This type of zoning is also know as local zoning or direct attach storage with local zoning.

**Note** You cannot implement Cisco UCS Manager-based zoning if the VSAN is also configured to communicate with a VSAN on an upstream switch and includes Fibre Channel or FCoE uplink ports.

#### Supported Fibre Channel Zoning Modes

Cisco UCS Manager-based zoning supports the following types of zoning:

- Single initiator single target—Cisco UCS Manager automatically creates one zone for each vHBA and storage port pair. Each zone has two members. We recommend that you configure this type of zoning unless you expect the number of zones to exceed the maximum supported.
- Single initiator multiple targets—Cisco UCS Manager automatically creates one zone for each vHBA. We recommend that you configure this type of zoning if you expect the number of zones to reach or exceed the maximum supported.

## vHBA Initiator Groups

vHBA initiator groups determine the Fibre Channel zoning configuration for all vHBAs in a service profile. Cisco UCS Manager does not include any default vHBA initiator groups. You must create vHBA initiator groups in any service profile that is to be assigned to servers included in a zone.

The configuration in a vHBA initiator group determines the following:

- The vHBAs included in the initiator group, which are sometimes referred to as vHBA initiators.
- A Fibre Channel storage connection policy, which includes the associated VSAN and the Fibre Channel target ports on the storage array.
- The type of Fibre Channel zoning to be configured for the vHBAs included in the group.

## Fibre Channel Storage Connection Policy

The Fibre Channel storage connection policy contains a collection of target storage ports on storage arrays that you use to configure Cisco UCS Manager-based Fibre Channel zoning. You can create this policy under an organization or an initiator group.

The storage arrays in these zones must be directly connected to the fabric interconnects. The target storage ports on these arrays that you include in the Fibre Channel storage connection policy can be either Fibre Channel storage ports or FCoE storage ports. You use the WWN of a port to add it to the policy and to identify the port for the Fibre Channel zone.

**Note** Cisco UCS Manager does not create default Fibre Channel storage.

## Fibre Channel Active Zone Set Configuration

In each VSAN that has been enabled for Fibre Channel zoning, Cisco UCS Manager automatically configures one zone set and multiple zones. The zone membership specifies the set of initiators and targets that are allowed to communicate with each other. Cisco UCS Manager automatically activates that zone set.

Cisco UCS Manager processes the user-configured vHBA initiator groups and their associated Fibre Channel storage connection policy to determine the desired connectivity between Fibre Channel initiators and targets. Cisco UCS Manager uses the following information to build pair-wise zone membership between initiators and targets:

- The port WWNs of the vHBA initiators derived from the vHBA initiator groups.
- The port WWNs of the storage array derived from the storage connection policy.

## Switch-Based Fibre Channel Zoning

With switch-based zoning, a Cisco UCS domain inherits the zoning configuration from the upstream switch. You cannot configure or view information about your zoning configuration in Cisco UCS Manager. You have to disable zoning on a VSAN in Cisco UCS Manager to use switch-based zoning for that VSAN.

# Guidelines and recommendations for Cisco UCS Manager-Based Fibre Channel Zoning

When you plan your configuration for Fibre Channel zoning, consider the following guidelines and recommendations:

#### Fibre Channel Switching Mode Must Be Switch Mode for Cisco UCS Manager Configurations

If you want Cisco UCS Manager to handle Fibre Channel zoning, the fabric interconnects must be in Fibre Channel Switch mode. You cannot configure Fibre Channel zoning in End-Host mode.

#### Symmetrical Configuration Is Recommended for High Availability

If a Cisco UCS domain is configured for high availability with two fabric interconnects, we recommend that both fabric interconnects are configured with the same set of VSANs.

# Configuring Cisco UCS Manager Fibre Channel Zoning

**Note** This procedure provides a high level overview of the steps required to configure a Cisco UCS domain for Fibre Channel zoning that is controlled by Cisco UCS Manager. You must ensure that you complete all of the following steps.

### Procedure

**Step 1** If you have not already done so, disconnect the fabric interconnects in the Cisco UCS domain from any external Fibre Channel switches, such as an MDS.

**Step 2** If the Cisco UCS domain still includes zones that were managed by the external Fibre Channel switch, run the **clear-unmanaged-fc-zone-all** command on every affected VSAN to remove those zones.

This functionality is not currently available in the Cisco UCS Manager GUI. You must perform this step in the Cisco UCS Manager CLI.

**Step 3** Configure the Fibre Channel switching mode for both fabric interconnects in Fibre Channel Switch mode.

You cannot configure Fibre Channel zoning in End-Host mode.

**Step 4** Configure the Fibre Channel and FCoE storage ports that you require to carry traffic for the Fibre Channel zones.

**Step 5** Create one or more VSANs and enable Fibre Channel zoning on all VSANs that you require to carry traffic for the Fibre Channel zones.

For a cluster configuration, we recommend that you create the VSANs that you intend to include in a Fibre Channel zone in Fibre Channel storage mode and accessible to both fabric interconnects.

**Step 6** Create one or more Fibre Channel storage connection policies.

You can perform this step when you configure Fibre Channel zoning in the service profiles, if you prefer.

**Step 7** Configure zoning in service profiles or service profile templates for servers that need to communicate through Fibre Channel zones.

Complete the following steps to complete this configuration:

- Enable zoning in the VSAN or VSANs assigned to the VHBAs.
- Configure one or more vHBA initiator groups.

# Creating a VSAN for Fibre Channel Zoning

### Procedure

**Step 1** UCS-A# **scope fc-uplink**

Enters Fibre Channel uplink mode.

**Step 2** UCS-A /fc-uplink #**create vsan** {*VSAN_Name*} {*VSAN_ID*} {*FCoE_VLAN_ID*}

Enter the following:

- VSAN_Name- The name assigned to the network. This name can be between 1 and 32 alphanumeric characters. You cannot use spaces or any special characters other than - (hyphen), _ (underscore), : (colon), and . (period), and you cannot change this name after the object is saved.
- VSAN_ID- The unique identifier assigned to the network. The ID can be between 1 and 4078, or between 4080 and 4093. 4079 is a reserved VSAN ID.
- FCoE_VLAN_ID- The unique identifier assigned to the VLAN used for Fibre Channel connections. The ID can be between 1 and 4029, or between 4048 and 4093. VLAN 4048 is user configurable. However, Cisco UCS Manager uses VLAN 4048 for the following default values. If you want to assign 4048 to a VLAN, you must reconfigure these values:

- After an upgrade to Cisco UCS, Release 2.0—The FCoE storage port native VLAN uses VLAN 4048 by default. If the default FCoE VSAN was set to use VLAN 1 before the upgrade, you must change it to a VLAN ID that is not used or reserved. For example, consider changing the default to 4049 if that VLAN ID is not in use.
- After a fresh install of Cisco UCS, Release 2.0—The FCoE VLANfor the default VSAN uses VLAN 4048 by default. The FCoEstorage port native VLAN uses VLAN 4049.

For FIP-capable, converged network adapters, such as the Cisco UCSCNA M72KR-Q and the Cisco UCS CNA M72KR-E, the named VSANmust be configured with a named VLAN that is not the native VLANfor the FCoE VLAN ID. This configuration ensures that FCoE trafficcan pass through these adapters.

**Step 3** UCS-A /fc-uplink #**commit-buffer**

### Example

The following examples creates a VSAN named TestVsan and commits the changes in the system:

```
UCS-A # scope fc-uplink
UCS-A /fc-uplink # create vsan TestVsan 2 30
UCS-A /fc-uplink/vsan* # commit-buffer
UCS-A /fc-uplink/vsan #
```

# Creating a New Fibre Channel Zone Profile

Perform the following procedure to create a new Fibre Channel Zone Profile.

### Before you begin

Ensure that the VSAN is created for the Fiber Channel Zoning.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | UCS-A # **scope fc-storage** | Enters Fibre Channel storage mode. |
| **Step 2** | UCS-A /fc-storage # **create fc-zone-profile** *Profile_Name* | Creates a Fibre Channel profile with the specified name. |
| **Step 3** | UCS-A /fc-storage/fc-zone-profile * # **create fc-user-zone** *Zone_Name* | Enters Fibre Channel zone profile mode and creates the specified Fibre Channel zone. |
| **Step 4** | UCS-A /fc-storage/fc-zone-profile/fc-user-zone* # **set path** {**A** \| **B**} | Sets the Fibre Channel zone path. |
| **Step 5** | UCS-A /fc-storage/fc-zone-profile/fc-user-zone* # **set vsan** *VSAN_Name* | Sets the Fibre Channel zone to the named VSAN. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 6** | UCS-A /fc-storage/fc-zone-profile/fc-user-zone* # **create member** *wwpn* | Creates the WWPN for the Fibre Channel zone profile. |
| **Step 7** | UCS-A /fc-storage/fc-zone-profile/fc-user-zone* # **commit-buffer** | Commits the transaction to the system configuration. |

### Example

The following example shows how to create FC Zoning Policy named myProfilie:

```
UCS-A# scope fc-storage
UCS-A /fc-storage # create fc-zone-profile myProfile
UCS-A /fc-storage/fc-zone-profile* # create fc-user-zone myZone
UCS-A /fc-storage/fc-zone-profile/fc-user-zone* # set path A
UCS-A /fc-storage/fc-zone-profile/fc-user-zone* # set vsan test
UCS-A /fc-storage/fc-zone-profile/fc-user-zone* # create member 20:c2:11:25:b5:00:00:7f
UCS-A /fc-storage/fc-zone-profile/fc-user-zone/member* # commit-buffer
```

# Deleting a Fibre Channel Zone Profile

Perform the following procedure to delete a Fibre Channel Zone Profile.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | UCS-A # **scope fc-storage** | Enters Fibre Channel storage mode. |
| **Step 2** | UCS-A /fc-storage # **delete fc-zone-profile** *Profile_Name* | Deletes a Fibre Channel profile with the specified name. |
| **Step 3** | UCS-A /fc-storage* # **commit-buffer** | Commits the transaction to the system configuration. |

### Example

The following example shows how to delete an FC Zone Profile named myProfile:

```
UCS-A # scope fc-storage
UCS-A /fc-storage # delete fc-zone-profile myProfile
UCS-A /fc-storage* # commit-buffer
UCS-A /fc-storage #
```

# Deleting a Fibre Channel User Zone

Perform the following procedure to delete a Fibre Channel User Zone.

### Procedure

**Step 1** UCS-A # **scope fc-storage**

Enters Fibre Channel storage mode.

**Step 2** UCS-A /fc-storage # **scope fc-zone-profile** *Profile_Name*

Enters the specified Fibre Channel profile.

**Step 3** UCS-A /fc-storage/fc-zone-profile # **delete fc-user-zone** *Userzone_Name*

Deletes the specified Fibre Channel User Zone.

**Step 4** UCS-A /fc-storage/fc-zone-profile* # **commit-buffer**

Commits the transaction to the system configuration.

### Example

The following example shows how to delete an FC User Zone Profile named myZone:

```
UCS-A # scope fc-storage
UCS-A /fc-storage # scope fc-zone-profile myProfile
UCS-A /fc-storage/fc-zone-profile # delete fc-user-zone myZone
UCS-A /fc-storage/fc-zone-profile* # commit-buffer
UCS-A /fc-storage #
```

# Removing Unmanaged Zones from a VSAN Accessible to Both Fabric Interconnects

After you disconnect the external Fibre Channel switch, the Fibre Channel zones that were managed by that switch might not been cleared from the Cisco UCS domain. This procedure removes those zones from each VSAN in the Cisco UCS domain so that you can configure Fibre Channel zoning in Cisco UCS.

### Before you begin

If you have not already done so, disconnect the fabric interconnects in the Cisco UCS domain from any external Fibre Channel switches, such as an MDS.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | UCS-A# **scope fc-uplink** | Enters Fibre Channel uplink mode. |
| **Step 2** | UCS-A /fc-uplink # **scope fabric** {**a** \| **b**} | Enters Fibre Channel uplink mode for the specified fabric interconnect. |
| **Step 3** | UCS-A /fc-uplink/fabric # **scope vsan** *vsan-name* | Enters VSAN mode for the specified named VSAN. |
| **Step 4** | UCS-A /fc-uplink/fabric/vsan # **clear-unmanaged-fc-zones-all** | Clears all unmanaged Fibre Channel zones from the specified named VSAN.<br>If desired, you can repeat Steps 2 through 4 to remove unmanaged zones from all VSANs that are accessible to the specified fabric interconnect before you commit the buffer. |
| **Step 5** | UCS-A /fc-uplink/fabric/vsan # **commit-buffer** | Commits the transaction to the system configuration. |

### Example

The following example shows how to remove unmanaged zones from a named VSAN accessible to fabric interconnect A and commit the transaction:

```
UCS-A# scope fc-uplink
UCS-A /fc-uplink # scope fabric a
UCS-A /fc-uplink/fabric # scope vsan finance
UCS-A /fc-uplink/fabric/vsan # clear-unmanaged-fc-zones-all
UCS-A /fc-uplink/fabric/vsan* # commit-buffer
UCS-A /fc-uplink #
```

# Removing Unmanaged Zones from a VSAN Accessible to One Fabric Interconnect

After you disconnect the external Fibre Channel switch, the Fibre Channel zones that were managed by that switch might not been cleared from the Cisco UCS domain. This procedure removes those zones from each VSAN in the Cisco UCS domain so that you can configure Fibre Channel zoning in Cisco UCS.

### Before you begin

If you have not already done so, disconnect the fabric interconnects in the Cisco UCS domain from any external Fibre Channel switches, such as an MDS.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | UCS-A# **scope fc-uplink** | Enters Fibre Channel uplink mode. |
| **Step 2** | UCS-A /fc-uplink # **scope vsan** *vsan-name* | Enters VSAN mode for the specified named VSAN. |
| **Step 3** | UCS-A /fc-uplink/vsan # **clear-unmanaged-fc-zones-all** | Clears all unmanaged Fibre Channel zones from the specified named VSAN.<br>If desired, you can repeat steps 2 and 3 to remove unmanaged zones from all VSANs that are accessible to both fabric interconnects before you commit the buffer. |
| **Step 4** | UCS-A /fc-uplink/vsan # **commit-buffer** | Commits the transaction to the system configuration. |

### Example

The following example shows how to remove unmanaged zones from a named VSAN and commit the transaction:

```
UCS-A# scope fc-uplink
UCS-A /fc-uplink # scope vsan finance
UCS-A /fc-uplink/vsan # clear-unmanaged-fc-zones-all
UCS-A /fc-uplink/vsan* # commit-buffer
UCS-A /fc-uplink #
```

# Configuring Fibre Channel Storage Connection Policies

## Creating a Fibre Channel Storage Connection Policy

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | UCS-A# **scope org** *org-name* | Enters the organization mode for the specified organization. To enter the root organization mode, enter / as the *org-name*. |
| **Step 2** | UCS-A /org # **create storage-connection-policy** *policy-name* | Creates a storage connection policy with the specified policy name, and enters organization storage connection policy mode. |
| **Step 3** | UCS-A /org # **set zoning-type** {**none** \| **simt** \| **sist**} | • **None**—Cisco UCS Manager does not configure Fibre Channel zoning. |

| | Command or Action | Purpose |
|---|---|---|
| | | • **Single Initiator Single Target**—Cisco UCS Manager automatically creates one zone for each vHBA and storage port pair. Each zone has two members. We recommend that you configure this type of zoning unless you expect the number of zones to exceed the maximum supported.<br>• **Single Initiator Multiple Targets**—Cisco UCS Manager automatically creates one zone for each vHBA. We recommend that you configure this type of zoning if you expect the number of zones to reach or exceed the maximum supported. |
| **Step 4** | UCS-A /org/storage-connection-policy # **create storage-target** *wwpn* | Creates a storage target endpoint with the specified WWPN, and enters storage target mode. |
| **Step 5** | UCS-A /org/storage-connection-policy/storage-target # **set target-path** {**a** \| **b**} | Specifies which fabric interconnect is used for communications with the target endpoint. |
| **Step 6** | UCS-A /org/storage-connection-policy/storage-target # **set target-vsan** *vsan* | Specifies which VSAN is used for communications with the target endpoint. |
| **Step 7** | UCS-A /org/storage-connection-policy # **commit-buffer** | Commits the transaction to the system configuration. |

### Example

The following example configures a Fibre Channel storage connection policy in the root organization named scPolicyZone1, using fabric interconnect A and the default VSAN, and commits the transaction:

```
UCS-A# scope org /
UCS-A /org* # create storage-connection-policy scPolicyZone1
UCS-A /org/storage-connection-policy* set zoning-type sist
UCS-A /org/storage-connection-policy* # create storage-target 20:10:20:30:40:50:60:70
UCS-A /org/storage-connection-policy/storage-target* # set target-path a
UCS-A /org/storage-connection-policy/storage-target* # set target-vsan default
UCS-A /org/storage-connection-policy* # commit-buffer
UCS-A /org/storage-connection-policy #
```

# Deleting a Fibre Channel Storage Connection Policy

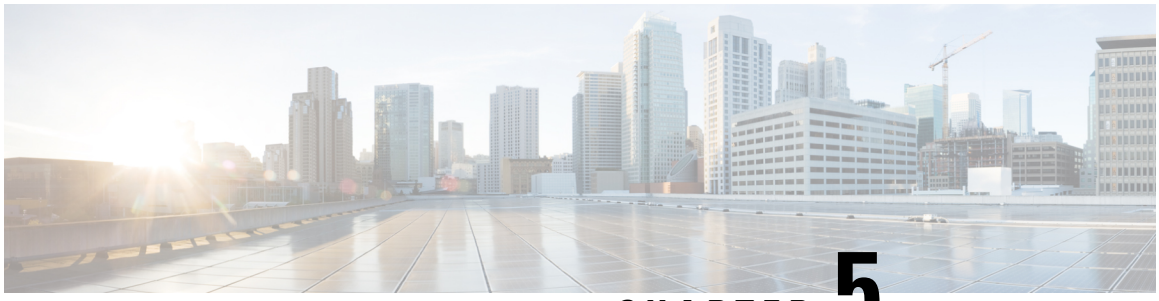
### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | UCS-A# **scope org** *org-name* | Enters the organization mode for the specified organization. To enter the root organization mode, enter / as the *org-name*. |
| **Step 2** | UCS-A /org # **delete storage-connection-policy** *policy-name* | Deletes the specified storage connection policy. |
| **Step 3** | UCS-A /org # **commit-buffer** | Commits the transaction to the system configuration. |

### Example

The following example deletes the storage connection policy named scPolicyZone1 from the root organization and commits the transaction:

```
UCS-A# scope org /
UCS-A /org # delete san-connectivity-policy scPolicyZone1
UCS-A /org* # commit-buffer
UCS-A /org #
```



CHAPTER 5

# Named VSANs



# Named VSANs

A named VSAN creates a connection to a specific external SAN. The VSAN isolates traffic to that external SAN, including broadcast traffic. The traffic on one named VSAN knows that the traffic on another named VSAN exists, but cannot read or access that traffic.

Like a named VLAN, the name that you assign to a VSAN ID adds a layer of abstraction that allows you to globally update all servers associated with service profiles that use the named VSAN. You do not need to reconfigure the servers individually to maintain communication with the external SAN. You can create more than one named VSAN with the same VSAN ID.

### Named VSANs in Cluster Configurations

In a cluster configuration, a named VSAN can be configured to be accessible only to the Fibre Channel uplink ports on one fabric interconnect or to the Fibre Channel uplink ports on both fabric interconnects.

### Named VSANs and the FCoE VLAN ID

You must configure each named VSAN with an FCoE VLAN ID. This property determines which VLAN is used for transporting the VSAN and its Fibre Channel packets.

For FIP-capable, converged network adapters, such as the Cisco UCS CNA M72KR-Q and the Cisco UCS CNA M72KR-E, the named VSAN must be configured with a named VLAN that is not the native VLAN for the FCoE VLAN ID. This configuration ensures that FCoE traffic can pass through these adapters.

In the following sample configuration, a service profile with a vNIC and vHBA mapped to fabric A is associated with a server that has FIP capable, converged network adapters:

- The vNIC is configured to use VLAN 10.
- VLAN 10 is also designated as the native VLAN for the vNIC.
- The vHBA is configured to use VSAN 2.
- Therefore, VSAN 2 cannot be configured with VLAN 10 as the FCoE VLAN ID. VSAN 2 can be mapped to any other VLAN configured on fabric A.

# Fibre Channel Uplink Trunking for Named VSANs

You can configure Fibre Channel uplink trunking for the named VSANs on each fabric interconnect. If you enable trunking on a fabric interconnect, all named VSANs in a Cisco UCS domain are allowed on all Fibre Channel uplink ports on that fabric interconnect.

# Guidelines and Recommendations for VSANs

The following guidelines and recommendations apply to all named VSANs, including storage VSANs.

### VSAN 4079 is a Reserved VSAN ID

Do not configure a VSAN as 4079. This VSAN is reserved and cannot be used in either FC switch mode or FC end-host mode.

If you create a named VSAN with ID 4079, Cisco UCS Manager marks that VSAN with an error and raises a fault.

### Reserved VSAN Range for Named VSANs in FC Switch Mode

If you plan to use FC switch mode in a Cisco UCS domain, do not configure VSANs with an ID in the range from 3040 to 4078.

VSANs in that range are not operational if the fabric interconnects are configured to operate in FC switch mode. Cisco UCS Manager marks that VSAN with an error and raises a fault.

### Reserved VSAN Range for Named VSANs in FC End-Host Mode

If you plan to use FC end-host mode in a Cisco UCS domain, do not configure VSANs with an ID in the range from 3840 to 4079.

VSANs in that range are not operational if the following conditions exist in a Cisco UCS domain:

- The fabric interconnects are configured to operate in FC end-host mode.
- The Cisco UCS domain is configured with Fibre Channel trunking or SAN port channels.

If these configurations exist, Cisco UCS Manager does the following:

1. Renders all VSANs with an ID in the range from 3840 to 4079 non-operational.
2. Raises a fault against the non-operational VSANs.
3. Transfers all non-operational VSANs to the default VSAN.
4. Transfers all vHBAs associated with the non-operational VSANs to the default VSAN.

If you disable Fibre Channel trunking and delete any existing SAN port channels, Cisco UCS Manager returns all VSANs in the range from 3840 to 4078 to an operational state and restores any associated vHBAs back to those VSANs.

### Range Restrictions for Named VSAN IDs in FC Switch Mode

If you plan to use FC switch mode in a Cisco UCS domain, do not configure VSANs in the range from 3040 to 4078.

When a fabric interconnect operating in FC switch mode is connected to MDS as the upstream switch, VSANs configured in Cisco UCS Manager in the range from 3040 to 4078 and assigned as port VSANs cannot be created in MDS. This configuration results in a possible port VSAN mismatch.

### Guidelines for FCoE VLAN IDs

**Note** FCoE VLANs in the SAN cloud and VLANs in the LAN cloud must have different IDs. Using the same ID for an FCoE VLAN in a VSAN and a VLAN results in a critical fault and traffic disruption for all vNICs and uplink ports using that FCoE VLAN. Ethernet traffic is dropped on any VLAN with an ID that overlaps with an FCoE VLAN ID.

VLAN 4048 is user configurable. However, Cisco UCS Manager uses VLAN 4048 for the following default values. If you want to assign 4048 to a VLAN, you must reconfigure these values:

- After an upgrade to Cisco UCS, Release 2.0—The FCoE storage port native VLAN uses VLAN 4048 by default. If the default FCoE VSAN was set to use VLAN 1 before the upgrade, you must change it to a VLAN ID that is not used or reserved. For example, consider changing the default to 4049 if that VLAN ID is not in use.
- After a fresh install of Cisco UCS, Release 2.0—The FCoE VLAN for the default VSAN uses VLAN 4048 by default. The FCoE storage port native VLAN uses VLAN 4049.

# Creating a Named VSAN Accessible to Both Fabric Interconnects (Fibre Channel Uplink Mode)

**Note** FCoE VLANs in the SAN cloud and VLANs in the LAN cloud must have different IDs. Using the same ID for an FCoE VLAN in a VSAN and a VLAN results in a critical fault and traffic disruption for all vNICs and uplink ports using that FCoE VLAN. Ethernet traffic is dropped on any VLAN with an ID that overlaps with an FCoE VLAN ID.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | UCS-A# **scope fc-uplink** | Enters Fibre Channel uplink mode. |
| **Step 2** | UCS-A /fc-uplink # **create vsan** *vsan-name vsan-id fcoe-id* | Creates the specified named VSAN, specifies the VSAN name, VSAN ID and FCoE VLAN ID, and enters Fibre Channel uplink VSAN mode.<br>• After an upgrade to Cisco UCS, Release 2.0—The FCoE storage port native VLAN uses VLAN 4048 by default. If the default FCoE VSAN was set to use VLAN 1 before the upgrade, you must change it to a VLAN ID that is not used or reserved. For example, consider changing the default to 4049 if that VLAN ID is not in use.<br>• After a fresh install of Cisco UCS, Release 2.0—The FCoE VLAN for the default VSAN uses VLAN 4048 by default. The FCoE storage port native VLAN uses VLAN 4049. |
| **Step 3** | UCS-A /fc-uplink/vsan # **set fc-zoning** {**disabled** \| **enabled**} | Configures Fibre Channel zoning for the VSAN, as follows:<br>• disabled—The upstream switch configures and controls the Fibre Channel zoning or Fibre Channel zoning is not implemented on this VSAN.<br>• enabled—Cisco UCS Manager configures and controls Fibre Channel zoning. |
| **Step 4** | UCS-A /fc-uplink/vsan # **commit-buffer** | Commits the transaction to the system configuration. |

### Example

The following example creates a named VSAN for both fabric interconnects, names the VSAN accounting, assigns the VSAN ID 2112, assigns the FCoE VLAN ID 4021, enables the VSAN for Cisco UCS Manager-based Fibre Channel zoning, and commits the transaction:

```
UCS-A# scope fc-uplink
UCS-A /fc-uplink* # create vsan accounting 2112 4021
UCS-A /fc-uplink/vsan # set fc-zoning enabled
UCS-A /fc-uplink/vsan* # commit-buffer
UCS-A /fc-uplink/vsan #
```

# Creating a Named VSAN Accessible to Both Fabric Interconnects (Fibre Channel Storage Mode)

**Note** FCoE VLANs in the SAN cloud and VLANs in the LAN cloud must have different IDs. Using the same ID for an FCoE VLAN in a VSAN and a VLAN results in a critical fault and traffic disruption for all vNICs and uplink ports using that FCoE VLAN. Ethernet traffic is dropped on any VLAN with an ID that overlaps with an FCoE VLAN ID.

### Procedure

**Step 1** UCS-A# **scope fc-storage**

Enters Fibre Channel storage mode.

**Step 2** UCS-A /fc-storage # **create vsan** *vsan-name vsan-id fcoe-id*

Creates the specified named VSAN, specifies the VSAN name, VSAN ID, and FCoE VLAN ID, and enters Fibre Channel storage VSAN mode.

- After an upgrade to Cisco UCS, Release 2.0—The FCoE storage port native VLAN uses VLAN 4048 by default. If the default FCoE VSAN was set to use VLAN 1 before the upgrade, you must change it to a VLAN ID that is not used or reserved. For example, consider changing the default to 4049 if that VLAN ID is not in use.
- After a fresh install of Cisco UCS, Release 2.0—The FCoE VLAN for the default VSAN uses VLAN 4048 by default. The FCoE storage port native VLAN uses VLAN 4049.

**Step 3** UCS-A /fc-storage/vsan # **create member-port** {**fc** | **fcoe**} {**a** | **b**} *slot-id port-id*

Creates a member port; specifies whether the port type, fabric, slot ID and port ID.

**Step 4** UCS-A /fc-storage/vsan # **set fc-zoning** {**disabled** | **enabled**}

Configures Fibre Channel zoning for the VSAN, as follows:

- disabled—The upstream switch configures and controls the Fibre Channel zoning or Fibre Channel zoning is not implemented on this VSAN.
- enabled—Cisco UCS Manager configures and controls Fibre Channel zoning.

**Step 5** UCS-A /fc-storage/vsan # **commit-buffer**

Commits the transaction to the system configuration.

### Example

The following example creates a named VSAN, names the VSAN finance, assigns the VSAN ID 3955, assigns the FCoE VLAN ID 4021, creates a member port and assigns it to member port A, slot 1 port 40, enables the VSAN for Cisco UCS Manager-based Fibre Channel zoning, and commits the transaction:

```
UCS-A# scope fc-storage
UCS-A /fc-storage/ # create VSAN finance 3955 4021
UCS-A /fc-storage/vsan # create member-port fcoe a 1 40
UCS-A /fc-storage/vsan # set fc-zoning enabled
UCS-A /fc-storage/vsan/member-port* # commit-buffer
UCS-A /fc-storage/vsan/member-port #
```

# Creating a Named VSAN Accessible to One Fabric Interconnect (Fibre Channel Uplink Mode)

**Note** FCoE VLANs in the SAN cloud and VLANs in the LAN cloud must have different IDs. Using the same ID for an FCoE VLAN in a VSAN and a VLAN results in a critical fault and traffic disruption for all vNICs and uplink ports using that FCoE VLAN. Ethernet traffic is dropped on any VLAN with an ID that overlaps with an FCoE VLAN ID.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | UCS-A# **scope fc-uplink** | Enters Fibre Channel uplink mode. |
| **Step 2** | UCS-A /fc-uplink # **scope fabric** {**a** \| **b**} | Enters Fibre Channel uplink fabric interconnect mode for the specified fabric interconnect (A or B). |
| **Step 3** | UCS-A /fc-uplink/fabric # **create vsan** *vsan-name vsan-id fcoe-id* | Creates the specified named VSAN, specifies the VSAN name, VSAN ID, and FCoE VLAN ID, and enters Fibre Channel uplink VSAN mode. |

| | Command or Action | Purpose |
|---|---|---|
| | | • After an upgrade to Cisco UCS, Release 2.0—The FCoE storage port native VLAN uses VLAN 4048 by default. If the default FCoE VSAN was set to use VLAN 1 before the upgrade, you must change it to a VLAN ID that is not used or reserved. For example, consider changing the default to 4049 if that VLAN ID is not in use.<br>• After a fresh install of Cisco UCS, Release 2.0—The FCoE VLAN for the default VSAN uses VLAN 4048 by default. The FCoE storage port native VLAN uses VLAN 4049. |
| **Step 4** | UCS-A /fc-uplink/vsan # **set fc-zoning** {**disabled** \| **enabled**} | Configures Fibre Channel zoning for the VSAN, as follows:<br>• disabled—The upstream switch configures and controls the Fibre Channel zoning or Fibre Channel zoning is not implemented on this VSAN.<br>• enabled—Cisco UCS Manager configures and controls Fibre Channel zoning. |
| **Step 5** | UCS-A /fc-uplink/fabric/vsan # **commit-buffer** | Commits the transaction to the system configuration. |

### Example

The following example creates a named VSAN for fabric interconnect A, names the VSAN finance, assigns the VSAN ID 3955, assigns the FCoE VLAN ID 2221, enables the VSAN for Cisco UCS Manager-based Fibre Channel zoning, and commits the transaction:

```
UCS-A# scope fc-uplink
UCS-A /fc-uplink # scope fabric a
UCS-A /fc-uplink/fabric # create vsan finance 3955 2221
UCS-A /fc-uplink/vsan # set fc-zoning enabled
UCS-A /fc-uplink/fabric/vsan* # commit-buffer
UCS-A /fc-uplink/fabric/vsan #
```

# Creating a Named VSAN Accessible to One Fabric Interconnect (Fibre Channel Storage Mode)

**Note** FCoE VLANs in the SAN cloud and VLANs in the LAN cloud must have different IDs. Using the same ID for an FCoE VLAN in a VSAN and a VLAN results in a critical fault and traffic disruption for all vNICs and uplink ports using that FCoE VLAN. Ethernet traffic is dropped on any VLAN with an ID that overlaps with an FCoE VLAN ID.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | UCS-A# **scope fc-storage** | Enters Fibre Channel storage mode. |
| **Step 2** | UCS-A /fc-storage # **scope fabric** {**a** \| **b**} | Enters Fibre Channel storage mode for the specified fabric interconnect. |
| **Step 3** | UCS-A /fc-storage/fabric # **create vsan** *vsan-name vsan-id fcoe-id* | Creates the specified named VSAN, specifies the VSAN name, VSAN ID, and FCoE VLAN ID, and enters Fibre Channel storage VSAN mode.<br>• After an upgrade to Cisco UCS, Release 2.0—The FCoE storage port native VLAN uses VLAN 4048 by default. If the default FCoE VSAN was set to use VLAN 1 before the upgrade, you must change it to a VLAN ID that is not used or reserved. For example, consider changing the default to 4049 if that VLAN ID is not in use.<br>• After a fresh install of Cisco UCS, Release 2.0—The FCoE VLAN for the default VSAN uses VLAN 4048 by default. The FCoE storage port native VLAN uses VLAN 4049. |
| **Step 4** | UCS-A /fc-storage/fabric/vsan # **create member-port** {**fc** \| **fcoe**} {**a** \| **b**} *slot-id port-id* | Creates a member port on the specified VSAN. |
| **Step 5** | UCS-A /fc-storage/vsan # **set fc-zoning** {**disabled** \| **enabled**} | Configures Fibre Channel zoning for the VSAN, as follows:<br>• disabled—The upstream switch configures and controls the Fibre Channel zoning or Fibre Channel zoning is not implemented on this VSAN. |

| | Command or Action | Purpose |
|---|---|---|
| | | • enabled—Cisco UCS Manager configures and controls Fibre Channel zoning. |
| **Step 6** | UCS-A /fc-storage/fabric/vsan # **commit-buffer** | Commits the transaction to the system configuration. |

### Example

The following example creates a named VSAN on fabric A, names the VSAN finance, assigns the VSAN ID 3955, assigns the FCoE VLAN ID 2221, creates a member port and assigns the it to member port A, slot 1 port 40, and commits the transaction:

```
UCS-A# scope fc-storage
UCS-A /fc-storage/ # scope fabric a
UCS-A /fc-storage/fabric # create VSAN finance 3955 2221
UCS-A /fc-storage/fabric/vsan # create member-port a 1 40
UCS-A /fc-storage/fabric/vsan # set fc-zoning enabled
UCS-A /fc-storage/fabric/vsan/member-port* # commit-buffer
UCS-A /fc-storage/fabric/vsan/member-port #
```

# Displaying a Named VSAN

The UCS Manager displays the named VSAN membership information.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | UCS-A# **scope fc-uplink** | Enters Fibre Channel uplink mode. |
| **Step 2** | UCS-A /fc-uplink # **scope vsan** *vsan-name* | Scopes the specified named VSAN.<br>You can proceed to Step 3 or Step 4 based on the Fabric Interconnect series. |
| **Step 3** | UCS-A /fc-uplink #/vsan # **show member-port** | Displays the named VSAN member details.<br>**Note**<br>This command is supported on Cisco UCS 6300, 6400, and 6500 Series Fabric Interconnects and Cisco UCS Fabric Interconnects 9108 100G. |
| **Step 4** | UCS-A /fc-uplink #/vsan # **show member-aggr-port** | Displays the named VSAN aggregate member port details.<br>**Note**<br>This command is supported only on Cisco UCS 6500 Series Fabric Interconnects. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 5** | (Optional) UCS-A /fc-uplink #/vsan # **show member-aggr-port expand** | Displays the expanded view of the named VSAN aggregate member port details.<br>**Note** This command is supported only on Cisco UCS 6500 Series Fabric Interconnects. |

### Example

#### Example 1

The following example shows how to view the details of a named VSAN on Cisco UCS 6300 and 6400 Series Fabric Interconnects.

```
UCS-A# scope fc-uplink
UCS-A /fc-uplink # scope vsan #vsan name#
UCS-A /fc-uplink #/vsan# show member port
UCS-A /fc-uplink #
```

#### Example 2

The following example shows how to view the details of a named VSAN aggregate member port details on Cisco UCS 6500 series Fabric Interconnect:

```
UCS-A# scope fc-uplink
UCS-A /fc-uplink # scope vsan #vsan name#
UCS-A /fc-uplink #/vsan# show member-aggr-port
UCS-A /fc-uplink #

Output:
Member Aggregate-Port:
    Fabric ID Slot Id Aggr Port Id
    --------- ------- ------------
    A               1 35
    A               1 36
```

#### Example 3

The following example shows how to view the details of a named VSAN aggregate member port details in an expanded view on Cisco UCS 6500 Series Fabric Interconnect.

```
UCS-A# scope fc-uplink
UCS-A /fc-uplink # scope vsan #vsan name#
UCS-A /fc-uplink #/vsan# show member-aggr-port expand
UCS-A /fc-uplink #

Output:
Member Aggregate-Port:
    Fabric ID: A
    Slot Id: 1
    Aggr Port Id: 35

    Breakout FC Member Port:
        Fabric ID Slot ID    Aggr-Port ID Port ID    Oper State       State Reason Oper
Speed
       --------- ---------- ------------ ---------- ---------------- ------------ ----------

        A                 1 35           2          Sfp Not Present  SFP not present
```

```
indeterminate
        A                   1 35           3          Sfp Not Present  SFP not present

indeterminate
        A                   1 35           4          Sfp Not Present  SFP not present

indeterminate
    Fabric ID: A
    Slot Id: 1
    Aggr Port Id: 36

    Breakout FC Member Port:
        Fabric ID Slot ID    Aggr-Port ID Port ID    Oper State       State Reason Oper
Speed
       --------- ---------- ------------ ---------- ---------------- ------------ ----------

        A                   1 36           2          Sfp Not Present  SFP not present

indeterminate
        A                   1 36           3          Sfp Not Present  SFP not present

indeterminate
        A                   1 36           4          Sfp Not Present  SFP not present

indeterminate
```

# Deleting a Named VSAN

If Cisco UCS Manager includes a named VSAN with the same VSAN ID as the one you delete, the VSAN is not removed from the fabric interconnect configuration until all named VSANs with that ID are deleted.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | UCS-A# **scope fc-uplink** | Enters Fibre Channel uplink mode. |
| **Step 2** | UCS-A /fc-uplink # **delete vsan** *vsan-name* | Deletes the specified named VSAN. |
| **Step 3** | UCS-A /fc-uplink # **commit-buffer** | Commits the transaction to the system configuration. |

### Example

The following example shows how to delete a named VSAN and commit the transaction:

```
UCS-A# scope fc-uplink
UCS-A /fc-uplink # delete vsan finance
UCS-A /fc-uplink* # commit-buffer
UCS-A /fc-uplink #
```

# Changing the VLAN ID for the FCoE Native VLAN for a Named VSAN

**Note** FCoE VLANs in the SAN cloud and VLANs in the LAN cloud must have different IDs. Using the same ID for an FCoE VLAN in a VSAN and a VLAN results in a critical fault and traffic disruption for all vNICs and uplink ports using that FCoE VLAN. Ethernet traffic is dropped on any VLAN with an ID that overlaps with an FCoE VLAN ID.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | UCS-A# **scope fc-uplink** | Enters Fibre Channel uplink mode. |
| **Step 2** | UCS-A /fc-uplink # **scope vsan** *vsan-name* | Enters VSAN mode for the specified named VSAN. |
| **Step 3** | UCS-A /fc-uplink/vsan # **set fcoe-vlan** *fcoe-vlan-id* | Sets the unique identifier assigned to the VLAN used for Fibre Channel connections. |
| **Step 4** | UCS-A /fc-uplink/vsan # **commit-buffer** | Commits the transaction to the system configuration. |

### Example

The following example changes the VLAN ID for the FCoE Native VLAN on a named VSAN called finance to 4000 and commits the transaction:

```
UCS-A# scope fc-uplink
UCS-A /fc-uplink # scope vsan finance
UCS-A /fc-uplink/vsan # set fcoe-vlan 4000
UCS-A /fc-uplink/vsan* # commit-buffer
UCS-A /fc-uplink/vsan #
```

# Changing the VLAN ID for the FCoE Native VLAN for a Storage VSAN

**Note** FCoE VLANs in the SAN cloud and VLANs in the LAN cloud must have different IDs. Using the same ID for an FCoE VLAN in a VSAN and a VLAN results in a critical fault and traffic disruption for all vNICs and uplink ports using that FCoE VLAN. Ethernet traffic is dropped on any VLAN with an ID that overlaps with an FCoE VLAN ID.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | UCS-A# **scope fc-storage** | Enters Fibre Channel storage mode. |
| **Step 2** | UCS-A /fc-storage # **set fcoe-storage-native-vlan** *fcoe-id* | Sets the unique identifier assigned to the VLAN used for Fibre Channel connections. |
| **Step 3** | UCS-A /fc-storage # **commit-buffer** | Commits the transaction to the system configuration. |

### Example

The following example changes the VLAN ID for the FCoE Native VLAN on a storage VSAN called finance to 4000 and commits the transaction:

```
UCS-A# scope fc-storage
UCS-A /fc-storage # set fcoe-storage-native-vlan 4000
UCS-A /fc-storage* # commit-buffer
UCS-A /fc-storage #
```

# Enabling or Disabling Fibre Channel Uplink Trunking

**Note** If the fabric interconnects are configured for Fibre Channel end-host mode, enabling Fibre Channel uplink trunking renders all VSANs with an ID in the range from 3840 to 4079 non-operational.

**Note** Before enabling VSAN trunking on a Fabric Interconnect, ensure that all host OS storage path redundancies are functioning. For more information on steps to monitor and ensure the Fibre Channel paths are recovered, see the Verification that the Data Path is Ready section. This should be followed to avoid an all paths down to the Fibre Channel Uplinks.

After confirmation, enable Fibre Channel Uplink Trunking on secondary Fabric Interconnect and wait until the secondary Fibre Channel VIF paths recover. Then move to enabling the primary Fabric Interconnect Fibre Channel Trunking after validating data paths.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | UCS-A# **scope fc-uplink** | Enters Fibre Channel uplink mode. |
| **Step 2** | UCS-A /fc-uplink # **scope fabric** {**a** \| **b** } | Enters Fibre Channel uplink mode for the specified fabric. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 3** | UCS-A /fc-uplink/fabric # **set uplink-trunking** {**enabled** \| **disabled** } | Enables or disables uplink trunking. |
| **Step 4** | UCS-A /fc-uplink/fabric # **commit-buffer** | Commits the transaction to the system configuration. |

### Example

The following example enables Fibre Channel uplink trunking for fabric A and commits the transaction:

```
UCS-A# scope fc-uplink
UCS-A /fc-uplink # scope fabric a
UCS-A /fc-uplink/fabric # set uplink-trunking enabled
UCS-A /fc-uplink/fabric* # commit-buffer
UCS-A /fc-uplink/fabric #
```

# Configuring Breakout VSAN and Member Port

The following steps describe about creating a breakout VSAN and adding a member to breakout VSAN:

### Procedure

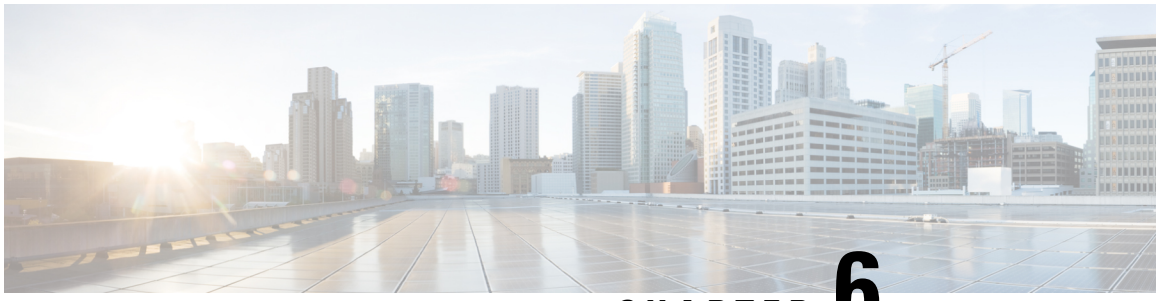
| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | UCS-A# **scope fc-uplink** | Enters Fibre Channel uplink mode. |
| **Step 2** | UCS-A /fc-uplink # **scope fabric a** | Enters Fibre Channel uplink fabric mode for the specified fabric. |
| **Step 3** | UCS-A /fc-uplink/fabric # **create vsan testVsan** *2020 2022* | Creates VSAN on the specified Fibre Channel uplink port, and enters Fibre Channel uplink fabric VSAN mode. |
| **Step 4** | UCS-A /fc-uplink/fabric/vsan* # **create member-aggr-port a***1 36* | Creates the interface for the specified aggregate (main) Fibre Channel uplink port. |
| **Step 5** | UCS-A /fc-uplink/fabric/vsan* # **create member-aggr-port a***1 36* | Creates the interface for the specified aggregate (main) Fibre Channel uplink port. |
| **Step 6** | UCS-A /fc-uplink/fabric/vsan/member-aggr-port* # **create br-member-portbr-fc** *3* | Creates a breakout member port for the specified Fibre Channel Uplink port. |
| **Step 7** | UCS-A /fc-uplink/fabric/vsan/member-aggr-port/br-fc* # **commit-buffer** | Commits the transaction to the system configuration. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 8** | UCS-A /fc-uplink/fabric/vsan/member-aggr-port/br-fc # **up** | |
| **Step 9** | UCS-A /fc-uplink/fabric/vsan/member-aggr-port # **show br-member-port br-fc br-fc** | Displays the output. |

### Example

The following is the example for creating and adding a member to a breakout VSAN:

```
Breakout FC Member Port:
    Fabric ID Slot ID    Aggr-Port ID Port ID    Oper State       State Reason Oper Speed
    --------- ---------- ------------ ---------- ---------------- ------------ ----------
    A                  1 36           3          Sfp Not Present  SFP not present
                                                                                indeterminate
UCS-A /fc-uplink/fabric/vsan/member-aggr-port # up
UCS-A /fc-uplink/fabric/vsan # show member-aggr-port

Member Aggregate-Port:
    Fabric ID Slot Id Aggr Port Id
    --------- ------- ------------
    A               1 36
UCS-A /fc-uplink/fabric/vsan #
```



CHAPTER 6

# SAN Pin Groups



## SAN Pin Groups

Cisco UCS uses SAN pin groups to pin Fibre Channel traffic from a vHBA on a server to an uplink Fibre Channel port on the fabric interconnect. You can use this pinning to manage the distribution of traffic from the servers.

**Note** In Fibre Channel switch mode, SAN pin groups are irrelevant. Any existing SAN pin groups will be ignored.

To configure pinning for a server, you must include the SAN pin group in a vHBA policy. The vHBA policy is then included in the service profile assigned to that server. All traffic from the vHBA will travel through the I/O module to the specified uplink Fibre Channel port.

You can assign the same pin group to multiple vHBA policies. As a result, you do not need to manually pin the traffic for each vHBA.

**Important** Changing the target interface for an existing SAN pin group disrupts traffic for all vHBAs which use that pin group. The fabric interconnect performs a log in and log out for the Fibre Channel protocols to re-pin the traffic.

## Configuring a SAN Pin Group

In a system with two fabric interconnects, you can associate the pin group with only one fabric interconnect or with both fabric interconnects.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | UCS-A# **scope fc-uplink** | Enters Fibre Channel uplink mode. |
| **Step 2** | UCS-A /fc-uplink # **create pin-group** *pin-group-name* | Creates a Fibre Channel (SAN) pin group with the specified name, and enters Fibre Channel uplink pin group mode. |
| **Step 3** | (Optional) UCS-A /fc-uplink/pin-group # **set descr** *description* | Provides a description for the pin group.<br>**Note** If your description includes spaces, special characters, or punctuation, you must begin and end your description with quotation marks. The quotation marks will not appear in the description field of any **show** command output. |
| **Step 4** | (Optional) UCS-A /fc-uplink/pin-group # **set target** {**a** \| **b** \| **dual**} **port** *slot-num* / *port-num* | Sets the Fibre Channel pin target to the specified fabric and port. |
| **Step 5** | UCS-A /fc-uplink/pin-group # **commit-buffer** | Commits the transaction to the system configuration. |

### Example

The following example creates a SAN pin group named fcpingroup12, provides a description for the pin group, sets the pin group target to slot 2, port 1, and commits the transaction:

```
UCS-A# scope fc-uplink
UCS-A /fc-uplink # create pin-group fcpingroup12
UCS-A /fc-uplink/pin-group* # set descr "This is my pin group #12"
UCS-A /fc-uplink/pin-group* # set target a port 2/1
UCS-A /fc-uplink/pin-group* # commit-buffer
UCS-A /fc-uplink/pin-group #
```

### What to do next

Include the pin group in a vHBA template.

# Configuring a FCoE Pin Group

You can create a FCoE pin group, and specify the FCoE uplink port as the pin group target.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | UCS-A# **scope fc-uplink** | Enters FC uplink mode. |
| **Step 2** | UCS-A /fc-uplink # **create pin-group fcoepingroup** | Creates a FCoE pin group with the specified name, and enters FCoE uplink pin group mode. |
| **Step 3** | UCS-A /fc-uplink/pin-group # **set target a fcoe-port 1/8** | Sets FCoE port 1/8 as the target port for this pin group. |
| **Step 4** | UCS-A /fc-uplink/pin-group # **commit-buffer** | Commits the transaction to the system configuration. |

### Example

```
UCS-A# scope fc-uplink
UCS-A /fc-uplink # create pin-group fcoepingroup
UCS-A /fc-uplink/pin-group* #set target a fcoe-port 1/8
UCS-A /fc-uplink/pin-group* # commit-buffer
UCS-A /fc-uplink/pin-group #
```

# Configuring SAN Pin Group

The following steps describe about creating a breakout SAN pin group:

### Procedure

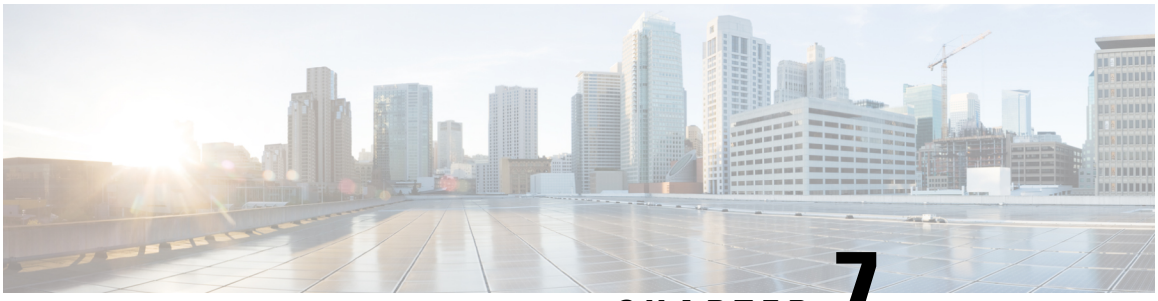
| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | UCS-A# **scope fc-uplink** | Enters Fibre Channel uplink mode. |
| **Step 2** | UCS-A /fc-uplink **create pin-group Test** | |
| **Step 3** | UCS-A /fc-uplink/pin-group* # **set target** *a***breakout-port***1 36 4* | Sets the Fibre Channel pin target to the specified fabric and port. |
| **Step 4** | UCS-A /fc-uplink/pin-group* # **commit-buffer** | Commits the transaction to the system configuration. |
| **Step 5** | UCS-A /fc-uplink/pin-group# **show target** | Displays the output. |

### Example

The following is the example for creating breakout SAN pin-group:

```
FC Pin Target:
    Fabric Endpoint
    ------ --------
```

```
    A      fabric/san/A/slot-1-aggr-port-36/phys-slot-1-port-4
UCS-A /fc-uplink/pin-group #
```



CHAPTER 7

# FC Identity Assignment



# Fibre Channel Identity

A Fibre Channel node and port must have a globally unique World Wide Number (WWN). In Cisco UCS WWNs are created as identity pools. A Fibre Channel node (a whole server, storage array) must have a World Wide Node Name (WWNN) and a Fibre Channel port must have a World Wide Port Name (WWPN). Both WWNNs and WWPNs are physical entities; hence, they have a 64-bit address.

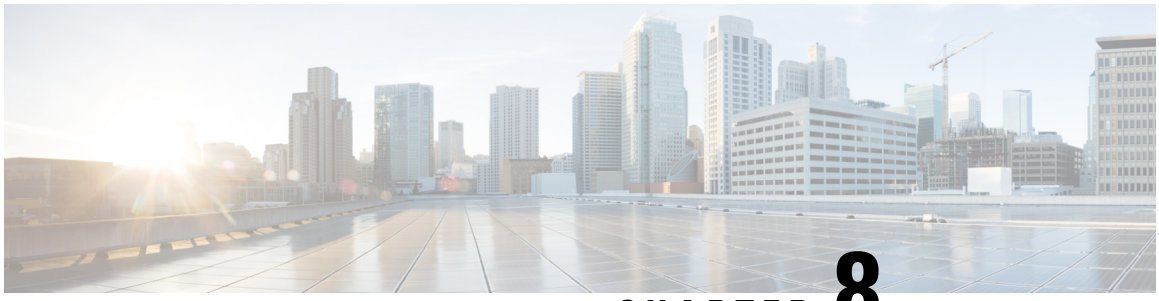
The WWNN pool is created as one large pool for the Cisco UCS domain. You can use the default pool in the Cisco UCS Manager SAN tab. However, it is recommended to create a custom WWNN pool for that UCS domain.

A communicating device is a node. A host bus adapter in a server constitutes a Fibre Channel node. For servers and hosts, WWNN is unique foe each host bus adapter (HBA). For a SAN switch, the WWNN is common for the chassis. For midrange storage, the WWNN is common for each controller unit. For enterprise storage, the WWNN is unique for the entire array.

Each server has a unique WWPN for each port of the HBA. For a SAN switch, the WWPN is available for each port in the chassis. For storage, each port has an individual.

The FC Identity Tab in Cisco UCS Manager displays the FC Identity of the devices in the Cisco UCS domain SAN cloud, including the:

- Selected device WWNN or WWPN identifier.
- Whether the identifier is assigned ti a vHBA.
- vHBA to which the identifier is assigned.



CHAPTER 8

# WWN Pools



# WWN Pools

A World Wide Name (WWN) pool is a collection of WWNs for use by the Fibre Channel vHBAs in a Cisco UCS domain. You create separate pools for the following:

- WW node names assigned to the vHBA
- WW port names assigned to the vHBA
- Both WW node names and WW port names

**Important** A WWN pool can include only WWNNs or WWPNs in the ranges from 20:00:00:00:00:00:00:00 to 20:FF:00:FF:FF:FF:FF:FF or from 50:00:00:00:00:00:00:00 to 5F:FF:00:FF:FF:FF:FF:FF. All other WWN ranges are reserved. When fibre channel traffic is sent through the Cisco UCS infrastructure, the source WWPN is converted to a MAC address. You cannot use WWPN pool which can translate to a source multicast MAC address. To ensure the uniqueness of the Cisco UCS WWNNs and WWPNs in the SAN fabric, Cisco recommends using the following WWN prefix for all blocks in a pool: 20:00:00:25:B5:XX:XX:XX

If you use WWN pools in service profiles, you do not have to manually configure the WWNs that will be used by the server associated with the service profile. In a system that implements multi-tenancy, you can use a WWN pool to control the WWNs used by each organization.

You assign WWNs to pools in blocks.

### WWNN Pools

A WWNN pool is a WWN pool that contains only WW node names. If you include a pool of WWNNs in a service profile, the associated server is assigned a WWNN from that pool.

### WWPN Pools

A WWPN pool is a WWN pool that contains only WW port names. If you include a pool of WWPNs in a service profile, the port on each vHBA of the associated server is assigned a WWPN from that pool.

### WWxN Pools

A WWxN pool is a WWN pool that contains both WW node names and WW port names. You can specify how many ports per node are created with WWxN pools. The pool size must be a multiple of *ports-per-node* + 1. For example, if you specify 7 ports per node, the pool size must be a multiple of 8. If you specify 63 ports per node, the pool size must be a multiple of 64.

You can use a WWxN pool whenever you select a WWNN or WWPN pool. The WWxN pool must be created before it can be assigned.

- For WWNN pools, the WWxN pool is displayed as an option in the **WWNN Assignment** drop-down list.
- For WWPN pools, choose **Derived** in the **WWPN Assignment** drop-down list.

# Creating a WWN Pool

**Important** A WWN pool can include only WWNNs or WWPNs in the ranges from 20:00:00:00:00:00:00:00 to 20:FF:00:FF:FF:FF:FF:FF or from 50:00:00:00:00:00:00:00 to 5F:FF:00:FF:FF:FF:FF:FF. All other WWN ranges are reserved. When fibre channel traffic is sent through the Cisco UCS infrastructure, the source WWPN is converted to a MAC address. You cannot use WWPN pool which can translate to a source multicast MAC address. To ensure the uniqueness of the Cisco UCS WWNNs and WWPNs in the SAN fabric, Cisco recommends using the following WWN prefix for all blocks in a pool: 20:00:00:25:B5:XX:XX:XX

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | UCS-A# **scope org** *org-name* | Enters the organization mode for the specified organization. To enter the root organization mode, enter / as the *org-name*. |
| **Step 2** | UCS-A /org # **create wwn-pool** *wwn-pool-name* {**node-and-port-wwn-assignment** \| **node-wwn-assignment** \| **port-wwn-assignment**} | Creates a WWN pool with the specified name and purpose, and enters organization WWN pool mode. This can be one of the following:<br>• **node-and-port-wwn-assignment**—Creates a WWxN pool that includes both world wide node names (WWNNs) and world wide port names (WWPNs).<br>• **node-wwn-assignment**—Creates a WWNN pool that includes only WWNNs. |

| | Command or Action | Purpose |
|---|---|---|
| | | • **port-wwn-assignment**—Creates a WWPN pool that includes only WWPNs.<br><br>This name can be between 1 and 32 alphanumeric characters. You cannot use spaces or any special characters other than - (hyphen), _ (underscore), : (colon), and . (period), and you cannot change this name after the object is saved. |
| **Step 3** | (Optional) UCS-A /org/wwn-pool # **set descr** *description* | Provides a description for the WWN pool.<br><br>**Note**<br>If your description includes spaces, special characters, or punctuation, you must begin and end your description with quotation marks. The quotation marks will not appear in the description field of any **show** command output. |
| **Step 4** | UCS-A /org/wwn-pool # **set assignmentorder** {**default** \| **sequential**} | This can be one of the following:<br><br>• **default**—Cisco UCS Manager selects a random identity from the pool.<br><br>• **sequential**—Cisco UCS Manager selects the lowest available identity from the pool. |
| **Step 5** | UCS-A /org/wwn-pool # **set max-ports-per-node** {**15-ports-per-node** \| **3-ports-per-node** \| **31-ports-per-node** \| **63-ports-per-node** \| **7-ports-per-node**} | For WWxN pools, specify the maximum number of ports that can be assigned to each node name in this pool. The default value is **3-ports-per-node**.<br><br>**Note**<br>The pool size for WWxN pools must be a multiple of *ports-per-node* + 1. For example, if you specify **7-ports-per-node**, the pool size must be a multiple of 8. If you specify **63-ports-per-node**, the pool size must be a multiple of 64. |
| **Step 6** | UCS-A /org/wwn-pool # **create block** *first-wwn last-wwn* | Creates a block (range) of WWNs, and enters organization WWN pool block mode. You must specify the first and last WWN in the block using the form *nn***:***nn***:***nn***:***nn***:***nn***:***nn***:***nn***:***nn*, with the WWNs separated by a space.<br><br>**Note**<br>A WWN pool can contain more than one WWN block. To create multiple WWN blocks, you must enter multiple **create block** commands from organization WWN pool mode. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 7** | UCS-A /org/wwn-pool/block # **exit** | Exits organization WWN pool block mode. |
| **Step 8** | UCS-A /org/wwn-pool # **create initiator** *wwn wwn* | Creates a single initiator for a WWNN or WWPN pool, and enters organization WWN pool initiator mode. You must specify the initiator using the form *nn*:*nn*:*nn*:*nn*:*nn*:*nn*:*nn*:*nn*.<br>**Note**<br>A WWNN or WWPN pool can contain more than one initiator. To create multiple initiators, you must enter multiple **create initiator** commands from organization WWN pool mode. |
| **Step 9** | UCS-A /org/wwn-pool/initiator # **commit-buffer** | Commits the transaction to the system configuration. |

### Example

The following example shows how to create a WWNN pool named sanpool, provide a description for the pool, specify a block of WWNs and an initiator to be used for the pool, and commit the transaction:

```
UCS-A# scope org /
UCS-A /org # create wwn-pool sanpool node-wwn-assignment
UCS-A /org/wwn-pool* # set descr "This is my WWNN pool"
UCS-A /org/wwn-pool* # create block 20:00:00:25:B5:00:00:00 20:00:00:25:B5:00:00:01
UCS-A /org/wwn-pool/block* # exit
UCS-A /org/wwn-pool* # create initiator 23:00:00:05:AD:1E:02:00
UCS-A /org/wwn-pool/initiator* # commit-buffer
UCS-A /org/wwn-pool/initiator #
```

The following example shows how to create a WWxN pool named sanpool, provide a description for the pool, specify seven ports per node, specify a block of eight WWNs to be used for the pool, and commit the transaction:

```
UCS-A# scope org /
UCS-A /org # create wwn-pool sanpool node-and-port-wwn-assignment
UCS-A /org/wwn-pool* # set descr "This is my WWxN pool"
UCS-A /org/wwn-pool* # set max-ports-per-node 7-ports-per-node
UCS-A /org/wwn-pool* # create block 20:00:00:25:B5:00:00:00 20:00:00:25:B5:00:00:08
UCS-A /org/wwn-pool/block* # commit-buffer
UCS-A /org/wwn-pool/block #
```

### What to do next

- Include the WWPN pool in a vHBA template.
- Include the WWNN pool in a service profile and template.
- Include the WWxN pool in a service profile and template.

# Deleting a WWN Pool

If you delete a pool, Cisco UCS Manager does not reallocate any addresses from that pool that were assigned to vNICs or vHBAs. All assigned addresses from a deleted pool remain with the vNIC or vHBA to which they are assigned until one of the following occurs:

- The associated service profiles are deleted.
- The vNIC or vHBA to which the address is assigned is deleted.
- The vNIC or vHBA is assigned to a different pool.

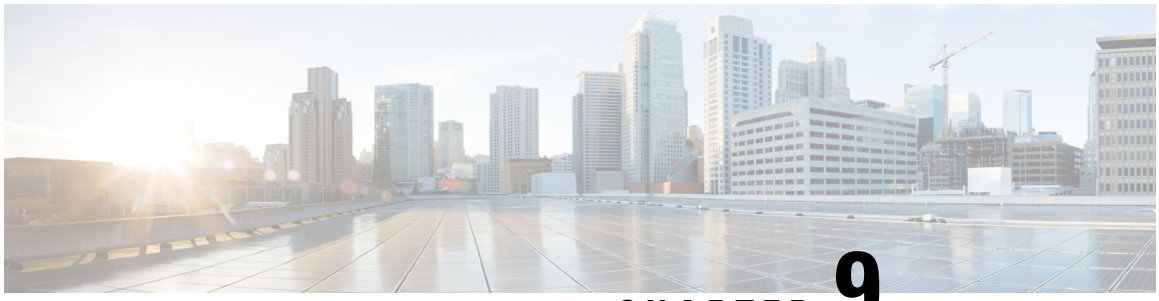
### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | UCS-A# **scope org** *org-name* | Enters the organization mode for the specified organization. To enter the root organization mode, enter **/** as the *org-name*. |
| **Step 2** | UCS-A /org # **delete wwn-pool** *pool-name* | Deletes the specified WWN pool. |
| **Step 3** | UCS-A /org # **commit-buffer** | Commits the transaction to the system configuration. |

### Example

The following example shows how to delete the WWN pool named pool4 and commit the transaction:

```
UCS-A# scope org /
UCS-A /org # delete wwn-pool pool4
UCS-A /org* # commit-buffer
UCS-A /org #
```



CHAPTER 9

# Storage-Related Policies



# Configuring vHBA Templates

## vHBA Template

This template is a policy that defines how a vHBA on a server connects to the SAN. It is also referred to as a vHBA SAN connectivity template.

You must include this policy in a service profile for it to take effect.

## Configuring a vHBA Template

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | UCS-A# **scope org** *org-name* | Enters organization mode for the specified organization. To enter the root organization mode, type / as the *org-name* . |
| **Step 2** | UCS-A /org # **create vhba-templ** *vhba-templ-name* [**fabric** {**a** \| **b**}] [**fc-if** *vsan-name*] | Creates a vHBA template and enters organization vHBA template mode. |
| **Step 3** | (Optional) UCS-A /org/vhba-templ # **set descr** *description* | Provides a description for the vHBA template. |
| **Step 4** | (Optional) UCS-A /org/vhba-templ # **set fabric** {**a** \| **b**} | Specifies the fabric to use for the vHBA. If you did not specify the fabric when creating the vHBA template in Step 2, then you have the option to specify it with this command. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 5** | (Optional) UCS-A /org/vhba-templ # **set fc-if** *vsan-name* | Specifies the Fibre Channel interface (named VSAN) to use for the vHBA template. If you did not specify the Fibre Channel interface when creating the vHBA template in Step 2, you have the option to specify it with this command. |
| **Step 6** | UCS-A /org/vhba-templ # **set max-field-size** *size-num* | Specifies the maximum size of the Fibre Channel frame payload (in bytes) that the vHBA supports. |
| **Step 7** | UCS-A /org/vhba-templ # **set pin-group** *group-name* | Specifies the pin group to use for the vHBA template. |
| **Step 8** | UCS-A /org/vhba-templ # **set qos-policy** *mac-pool-name* | Specifies the QoS policy to use for the vHBA template. |
| **Step 9** | UCS-A /org/vhba-templ # **set stats-policy** *policy-name* | Specifies the server and server component statistics threshold policy to use for the vHBA template. |
| **Step 10** | UCS-A /org/vhba-templ # **set type** {**initial-template** \| **updating-template**} | Specifies the vHBA template update type. If you do not want vHBA instances created from this template to be automatically updated when the template is updated, use the **initial-template** keyword; otherwise, use the **updating-template** keyword to ensure that all vHBA instances are updated when the vHBA template is updated. |
| **Step 11** | UCS-A /org/vhba-templ # **set wwpn-pool** *pool-name* | Specifies the WWPN pool to use for the vHBA template. |
| **Step 12** | UCS-A /org/vhba-templ # **commit-buffer** | Commits the transaction to the system configuration. |

### Example

The following example configures a vHBA template and commits the transaction:

```
UCS-A# scope org /
UCS-A /org* # create vhba template VhbaTempFoo
UCS-A /org/vhba-templ* # set descr "This is a vHBA template example."
UCS-A /org/vhba-templ* # set fabric a
UCS-A /org/vhba-templ* # set fc-if accounting
UCS-A /org/vhba-templ* # set max-field-size 2112
UCS-A /org/vhba-templ* # set pin-group FcPinGroup12
UCS-A /org/vhba-templ* # set qos-policy policy34foo
UCS-A /org/vhba-templ* # set stats-policy ServStatsPolicy
UCS-A /org/vhba-templ* # set type updating-template
UCS-A /org/vhba-templ* # set wwpn-pool SanPool7
UCS-A /org/vhba-templ* # commit-buffer
UCS-A /org/vhba-templ #
```

# Deleting a vHBA Template

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | UCS-A# **scope org** *org-name* | Enters organization mode for the specified organization. To enter the root organization mode, type **/** as the *org-name* . |
| **Step 2** | UCS-A /org # **delete vhba-templ** *vhba-templ-name* | Deletes the specified vHBA template. |
| **Step 3** | UCS-A /org # **commit-buffer** | Commits the transaction to the system configuration. |

### Example

The following example deletes the vHBA template named VhbaTempFoo and commits the transaction:

```
UCS-A# scope org /
UCS-A /org # delete vhba template VhbaTempFoo
UCS-A /org* # commit-buffer
UCS-A /org #
```

# Configuring Fibre Channel Adapter Policies

## Ethernet and Fibre Channel Adapter Policies

These policies govern the host-side behavior of the adapter, including how the adapter handles traffic. For example, you can use these policies to change default settings for the following:

- Queues
- Interrupt handling
- Performance enhancement
- RSS hash
- Failover in a cluster configuration with two fabric interconnects

**Note** For Fibre Channel adapter policies, the values displayed by Cisco UCS Manager may not match those displayed by applications such as QLogic SANsurfer. For example, the following values may result in an apparent mismatch between SANsurfer and Cisco UCS Manager:

- Max LUNs Per Target—SANsurfer has a maximum of 256 LUNs and does not display more than that number. Cisco UCS Manager supports a higher maximum number of LUNs. This parameter is applicable only for FC-Initiator.
- Link Down Timeout—In SANsurfer, you configure the timeout threshold for link down in seconds. In Cisco UCS Manager, you configure this value in milliseconds. Therefore, a value of 5500 ms in Cisco UCS Manager displays as 5s in SANsurfer.
- Max Data Field Size—SANsurfer has allowed values of 512, 1024, and 2048. Cisco UCS Manager allows you to set values of any size. Therefore, a value of 900 in Cisco UCS Manager displays as 512 in SANsurfer.
- LUN Queue Depth—The LUN queue depth setting is available for Windows system FC adapter policies. Queue depth is the number of commands that the HBA can send and receive in a single transmission per LUN. Windows Storport driver sets this to a default value of 20 for physical miniports and to 250 for virtual miniports. This setting adjusts the initial queue depth for all LUNs on the adapter. Valid range for this value is 1 - 254. The default LUN queue depth is 20. This feature only works with Cisco UCS Manager version 3.1(2) and higher. This parameter is applicable only for FC-Initiator.
- IO TimeOut Retry—When the target device does not respond to an IO request within the specified timeout, the FC adapter cancels the pending command then resends the same IO after the timer expires. The FC adapter valid range for this value is 1 - 59 seconds. The default IO retry timeout is 5 seconds. This feature only works with Cisco UCS Manager version 3.1(2) and higher.

From Cisco UCS Manager 4.3(4a), the adapter settings are optimized for Windows, Linux, and VMware for Cisco UCS VIC 1400 Series adapters, Cisco UCS VIC 14000 Series adapters, and Cisco UCS VIC 15000 Series adapters.

### Operating System Specific Adapter Policies

By default, Cisco UCS provides a set of Ethernet adapter policies and Fibre Channel adapter policies. These policies include the recommended settings for each supported server operating system. Operating systems are sensitive to the settings in these policies. Storage vendors typically require non-default adapter settings. You can find the details of these required settings on the support list provided by those vendors.

**Important** We recommend that you use the values in these policies for the applicable operating system. Do not modify any of the values in the default policies unless directed to do so by Cisco Technical Support.

However, if you are creating an Ethernet adapter policy for an OS (instead of using the default adapter policy), you must use the following formulas to calculate values that work for that OS.

Depending on the UCS firmware, your driver interrupt calculations may be different. Newer UCS firmware uses a calculation that differs from previous versions. Later driver release versions on Linux operating systems now use a different formula to calculate the Interrupt Count. In this formula, the Interrupt Count is the maximum of either the Transmit Queue or the Receive Queue plus 2.

### Interrupt Count in Linux Adapter Policies

Drivers on Linux operating systems use differing formulas to calculate the Interrupt Count, depending on the eNIC driver version. The UCS 3.2 release increased the number of Tx and Rx queues for the eNIC driver from 8 to 256 each.

Use one of the following strategies, according to your driver version.

For Linux drivers before the UCS 3.2 firmware release, use the following formula to calculate the Interrupt Count.

Completion Queues = Transmit Queues + Receive Queues
Interrupt Count = (Completion Queues + 2) rounded up to nearest power of 2

For example, if Transmit Queues = 1 and Receive Queues = 8 then:

Completion Queues = 1 + 8 = 9
Interrupt Count = (9 + 2) rounded up to the nearest power of 2 = 16

On drivers for UCS firmware release 3.2 and higher, the Linux eNIC drivers use the following formula to calculate the Interrupt Count.

Interrupt Count = Max(Tx, Rx) + 2

For example:

Interrupt Count wq = 32, rq = 32, cq = 64 - then Interrupt Count = Max(32, 32) + 2 = 34
Interrupt Count wq = 64, rq = 8, cq = 72 – then Interrupt Count = Max(64, 8) + 2 = 66
Interrupt Count wq = 1, rq = 16, cq = 17 - then Interrupt count = Max(1, 16) + 2 = 18

### Interrupt Count in Windows Adapter Policies

For Windows OS, the recommended adapter policy in UCS Manager for VIC 1400 series and above adapters is Win-HPN and if RDMA is used, the recommended policy is Win-HPN-SMB. For VIC 1400 series and above adapters, the recommended interrupt value setting is 512 and the Windows VIC driver takes care of allocating the required number of Interrupts.

For VIC 1300 and VIC 1200 series adapters, the recommended UCS Manager adapter policy is Windows and the Interrupt would be TX + RX + 2, rounded to closest power of 2. The maximum supported Windows queues is 8 for Rx Queues and 1 for Tx Queues.

Example for VIC 1200 and VIC 1300 series adapters:

Tx = 1, Rx = 4, CQ = 5, Interrupt = 8 ( 1 + 4 rounded to nearest power of 2), Enable RSS

Example for VIC 1400 series , 14000 series and 15000 series adapters and above adapters:

Tx = 1, Rx = 4, CQ = 5, Interrupt = 512 , Enable RSS

### NVMe over Fabrics using Fibre Channel

The NVM Express (NVMe) interface allows host software to communicate with a non-volatile memory subsystem. This interface is optimized for Enterprise non-volatile storage, which is typically attached as a register level interface to the PCI Express (PCIe) interface.

NVMe over Fabrics using Fibre Channel (FC-NVMe) defines a mapping protocol for applying the NVMe interface to Fibre Channel. This protocol defines how Fibre Channel services and specified Information Units (IUs) are used to perform the services defined by NVMe over a Fibre Channel fabric. NVMe initiators can access and transfer information to NVMe targets over Fibre Channel.

FC-NVMe combines the advantages of Fibre Channel and NVMe. You get the improved performance of NVMe along with the flexibility and the scalability of the shared storage architecture. Cisco UCS Manager Release 4.0(2) supports NVMe over Fabrics using Fibre Channel on UCS VIC 1400 Series adapters.

Starting with UCS Manager release 4.3(2b), NVMeoF using RDMA is supported on Cisco UCS VIC 14000 series adapters.

Starting with UCS Manager release 4.2(2), NVMeoF using Fibre Channel is supported on Cisco UCS VIC 15000 series adapters.

Cisco UCS Manager provides the recommended FC NVME Initiator adapter policies in the list of pre-configured adapter policies. To create a new FC-NVMe adapter policy, follow the steps in the *Creating a Fibre Channel Adapter Policy* section.

### NVMe over Fabrics Using RDMA

NVMe over Fabrics (NVMeoF) is a communication protocol that allows one computer to access NVMe namespaces available on another computer. NVMeoF is similar to NVMe, but differs in the network-related steps involved in using the NVMeoF storage devices. The commands for discovering, connecting, and disconnecting a NVMeoF storage device are integrated into the **nvme** utility provided in Linux..

The NVMeoF fabric that Cisco supports is RDMA over Converged Ethernet version 2 (RoCEv2). RoCEv2 is a fabric protocol that runs over UDP. It requires a no-drop policy.

The eNIC RDMA driver works in conjunction with the eNIC driver, which must be loaded first when configuring NVMeoF.

Cisco UCS Manager provides the default Linux-NVMe-RoCE adapter policy for creating NVMe RoCEv2 interfaces. Do not use the default Linux adapter policy. For complete information on configuring RoCEv2 over NVMeoF, refer to the *Cisco UCS Manager Configuration Guide for RDMA over Converged Ethernet (RoCE) v2*.

NVMeoF using RDMA is supported on M5 B-Series or C-Series Servers with Cisco UCS VIC 1400 Series adapters.

Starting with UCS Manager release 4.3(2b), NVMeOF using RDMA is supported on Cisco UCS VIC 14000 series adapters.

Starting with UCS Manager release 4.2(2), NVMeOF using RDMA is supported on Cisco UCS VIC 15000 series adapters.

# Configuring a Fibre Channel Adapter Policy

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | UCS-A# **scope org** *org-name* | Enters organization mode for the specified organization. To enter the root organization mode, type / as the *org-name* . |
| **Step 2** | UCS-A /org # **create fc-policy** *policy-name* | Creates the specified Fibre Channel adapter policy and enters organization Fibre Channel policy mode. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 3** | (Optional) UCS-A /org/fc-policy # **set descr** *description* | Provides a description for the policy.<br>**Note** If your description includes spaces, special characters, or punctuation, you must begin and end your description with quotation marks. The quotation marks will not appear in the description field of any **show** command output. |
| **Step 4** | (Optional) UCS-A /org/fc-policy # **set error-recovery** {**fcp-error-recovery** {**disabled** \| **enabled**} \| **link-down-timeout** *timeout-msec* \| **port-down-io-retry-count** *retry-count* \| **port-down-timeout** *timeout-msec*} | Configures the Fibre Channel error recovery. |
| **Step 5** | (Optional) UCS-A /org/fc-policy # **set interrupt mode** {**intx** \| **msi** \| **msi-x**}} | Configures the driver interrupt mode. |
| **Step 6** | (Optional) UCS-A /org/fc-policy # **set port** {**io-throttle-count** *throttle-count* \| **max-luns** *max-num*} | Configures the Fibre Channel port.<br>**Note** The **max-luns** option is applicable only to the **fc-initiator** vHBA type. |
| **Step 7** | (Optional) UCS-A /org/fc-policy # **set port-f-logi** {**retries** *retry-count* \| **timeout** *timeout-msec*} | Configures the Fibre Channel port fabric login (FLOGI). |
| **Step 8** | (Optional) UCS-A /org/fc-policy # **set port-p-logi** {**retries** *retry-count* \| **timeout** *timeout-msec*} | Configures the Fibre Channel port-to-port login (PLOGI). |
| **Step 9** | (Optional) UCS-A /org/fc-policy # **set recv-queue** {**count** *count* \| **ring-size** *size-num*} | Configures the Fibre Channel receive queue. |
| **Step 10** | (Optional) UCS-A /org/fc-policy # **set scsi-io** {**count** *count* \| **ring-size** *size-num*} | Configures the Fibre Channel I/O. |
| **Step 11** | (Optional) UCS-A /org/fc-policy # **set trans-queue ring-size** *size-num*} | Configures the Fibre Channel transmit queue. |
| **Step 12** | (Optional) UCS-A /org/fc-policy # **set vhbatype mode** {**fc-initiator** \| **fc-nvme-initiator** \| **fc-nvme-target** \| **fc-target**} | The vHBA type used in this policy. vHBAs supporting FC and FC-NVMe can now be created on the same adapter.<br>**Note** **fc-nvme-target** and **fc-target** are available as Tech Preview options. |
| **Step 13** | UCS-A /org/fc-policy # **commit-buffer** | Commits the transaction to the system configuration. |

### Example

The following example configures a Fibre Channel adapter policy and commits the transaction:

```
UCS-A# scope org /
UCS-A /org* # create fc-policy FcPolicy42
UCS-A /org/fc-policy* # set descr "This is a Fibre Channel adapter policy example."
UCS-A /org/fc-policy* # set error-recovery error-detect-timeout 2500
UCS-A /org/fc-policy* # set port max-luns 4
UCS-A /org/fc-policy* # set port-f-logi retries 250
UCS-A /org/fc-policy* # set port-p-logi timeout 5000
UCS-A /org/fc-policy* # set recv-queue count 1
UCS-A /org/fc-policy* # set scsi-io ring-size 256
UCS-A /org/fc-policy* # set trans-queue ring-size 256
UCS-A /org/fc-policy* # commit-buffer
UCS-A /org/fc-policy #
```

The following example configures a Fibre Channel adapter policy with the vHBA type set to FC NVME Initiator and commits the transaction:

```
UCS-A# scope org /
UCS-A /org* # create fc-policy FcPolicy42
UCS-A /org/fc-policy* # set descr "This is a Fibre Channel adapter policy example."
UCS-A /org/fc-policy* # set error-recovery error-detect-timeout 2500
UCS-A /org/fc-policy* # set port-f-logi retries 250
UCS-A /org/fc-policy* # set port-p-logi timeout 5000
UCS-A /org/fc-policy* # set recv-queue count 1
UCS-A /org/fc-policy* # set scsi-io ring-size 256
UCS-A /org/fc-policy* # set trans-queue ring-size 256
UCS-A /org/fc-policy* # set vhbatype mode fc-nvme-initiator
UCS-A /org/fc-policy* # commit-buffer
UCS-A /org/fc-policy #
```

# Deleting a Fibre Channel Adapter Policy

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | UCS-A# **scope org** *org-name* | Enters organization mode for the specified organization. To enter the root organization mode, type **/** as the *org-name* . |
| **Step 2** | UCS-A /org # **delete fc-policy** *policy-name* | Deletes the specified Fibre Channel adapter policy. |
| **Step 3** | UCS-A /org # **commit-buffer** | Commits the transaction to the system configuration. |

### Example

The following example deletes the Fibre Channel adapter policy named FcPolicy42 and commits the transaction:

```
UCS-A# scope org /
UCS-A /org # delete fc-policy FcPolicy42
UCS-A /org* # commit-buffer
UCS-A /org #
```

# Configuring the Default vHBA Behavior Policy

## Default vHBA Behavior Policy

Default vHBA behavior policy allow you to configure how vHBAs are created for a service profile. You can choose to create vHBAs manually, or you can allow them to be created automatically.

You can configure the default vHBA behavior policy to define how vHBAs are created. This can be one of the following:

- **None**—Cisco UCS Manager does not create default vHBAs for a service profile. All vHBAs must be explicitly created.
- **HW Inherit**—If a service profile requires vHBAs and none have been explicitly defined, Cisco UCS Manager creates the required vHBAs based on the adapter installed in the server associated with the service profile.

**Note** If you do not specify a default behavior policy for vHBAs, **none** is used by default.

## Configuring a Default vHBA Behavior Policy

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | UCS-A# **scope org /** | Enters the root organization mode. |
| **Step 2** | UCS-A/org # **scope vhba-beh-policy** | Enters default vHBA behavior policy mode. |
| **Step 3** | UCS-A/org/vhba-beh-policy # **set action** {**hw-inherit** [**template_name** *name*] \| **none**} | Specifies the default vHBA behavior policy. This can be one of the following:<br>• **hw-inherit**—If a service profile requires vHBAs and none have been explicitly defined, Cisco UCS Manager creates the required vHBAs based on the adapter installed in the server associated with the service profile.<br>If you specify **hw-inherit**, you can also specify a vHBA template to create the vHBAs. |

| | Command or Action | Purpose |
|---|---|---|
| | | • **none**—Cisco UCS Manager does not create default vHBAs for a service profile. All vHBAs must be explicitly created. |
| **Step 4** | UCS-A/org/vhba-beh-policy # **commit-buffer** | Commits the transaction to the system configuration. |

### Example

This example shows how to set the default vHBA behavior policy to **hw-inherit**.

```
UCS-A # scope org /
UCS-A/org # scope vhba-beh-policy
UCS-A/org/vhba-beh-policy # set action hw-inherit
UCS-A/org/vhba-beh-policy* # commit-buffer
UCS-A/org/vhba-beh-policy #
```

# Configuring SAN Connectivity Policies

## About the LAN and SAN Connectivity Policies

Connectivity policies determine the connections and the network communication resources between the server and the LAN or SAN on the network. These policies use pools to assign MAC addresses, WWNs, and WWPNs to servers and to identify the vNICs and vHBAs that the servers use to communicate with the network.

**Note** We do not recommend that you use static IDs in connectivity policies, because these policies are included in service profiles and service profile templates and can be used to configure multiple servers.

## Privileges Required for LAN and SAN Connectivity Policies

Connectivity policies enable users without network or storage privileges to create and modify service profiles and service profile templates with network and storage connections. However, users must have the appropriate network and storage privileges to create connectivity policies.

### Privileges Required to Create Connectivity Policies

Connectivity policies require the same privileges as other network and storage configurations. For example, you must have at least one of the following privileges to create connectivity policies:

- admin—Can create LAN and SAN connectivity policies
- ls-server—Can create LAN and SAN connectivity policies
- ls-network—Can create LAN connectivity policies

- ls-storage—Can create SAN connectivity policies

### Privileges Required to Add Connectivity Policies to Service Profiles

After the connectivity policies have been created, a user with ls-compute privileges can include them in a service profile or service profile template. However, a user with only ls-compute privileges cannot create connectivity policies.

# Interactions between Service Profiles and Connectivity Policies

You can configure the LAN and SAN connectivity for a service profile through either of the following methods:

- LAN and SAN connectivity policies that are referenced in the service profile
- Local vNICs and vHBAs that are created in the service profile
- Local vNICs and a SAN connectivity policy
- Local vHBAs and a LAN connectivity policy

Cisco UCS maintains mutual exclusivity between connectivity policies and local vNIC and vHBA configuration in the service profile. You cannot have a combination of connectivity policies and locally created vNICs or vHBAs. When you include a LAN connectivity policy in a service profile, all existing vNIC configuration is erased, and when you include a SAN connectivity policy, all existing vHBA configuration in that service profile is erased.

# Creating a SAN Connectivity Policy

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | UCS-A# **scope org** *org-name* | Enters the organization mode for the specified organization. To enter the root organization mode, enter / as the *org-name*. |
| **Step 2** | UCS-A /org # **create san-connectivity-policy** *policy-name* | Creates the specified SAN connectivity policy, and enters organization network control policy mode.<br>This name can be between 1 and 16 alphanumeric characters. You cannot use spaces or any special characters other than - (hyphen), _ (underscore), : (colon), and . (period), and you cannot change this name after the object is saved. |
| **Step 3** | (Optional) UCS-A /org/lan-connectivity-policy # **set descr** *policy-name* | Adds a description to the policy. We recommend that you include information about where and how the policy should be used. |

| | Command or Action | Purpose |
|---|---|---|
| | | Enter up to 256 characters. You can use any characters or spaces except ` (accent mark), \ (backslash), ^ (carat), " (double quote), = (equal sign), > (greater than), < (less than), or ' (single quote). |
| **Step 4** | UCS-A /org/service-profile # **set identity** {**dynamic-uuid** {*uuid* \| **derived**} \| **dynamic-wwnn** {*wwnn* \| **derived**} \| **uuid-pool** *pool-name* \| **wwnn-pool** *pool-name*} | Specifies how the server acquires a UUID or WWNN. You can do one of the following:<br>• Create a unique UUID in the form *nnnnnnnn-nnnn-nnnn-nnnnnnnnnnnn*<br>• Derive the UUID from the one burned into the hardware at manufacture<br>• Use a UUID pool<br>• Create a unique WWNN in the form *hh* **:** *hh* **:** *hh* **:** *hh* **:** *hh* **:** *hh* **:** *hh* **:** *hh*<br>• Derive the WWNN from one burned into the hardware at manufacture<br>• Use a WWNN pool |
| **Step 5** | UCS-A /org/lan-connectivity-policy # **commit-buffer** | Commits the transaction to the system configuration. |

### Example

The following example shows how to create a SAN connectivity policy named SanConnect242 and commit the transaction:

```
UCS-A# scope org /
UCS-A /org* # create san-connectivity-policy SanConnect242
UCS-A /org/san-connectivity-policy* # set descr "SAN connectivity policy"
UCS-A /org/san-connectivity-policy* # set identity wwnn-pool SanPool7
UCS-A /org/san-connectivity-policy* # commit-buffer
UCS-A /org/san-connectivity-policy #
```

### What to do next

Add one or more vHBAs and/or initiator groups to this SAN connectivity policy.

# Deleting a SAN Connectivity Policy

If you delete a SAN connectivity policy that is included in a service profile, it also deletes all vHBAs from that service profile and disrupts SAN data traffic for the server associated with the service profile.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | UCS-A# **scope org** *org-name* | Enters the organization mode for the specified organization. To enter the root organization mode, enter / as the *org-name*. |
| **Step 2** | UCS-A /org # **delete san-connectivity-policy** *policy-name* | Deletes the specified SAN connectivity policy. |
| **Step 3** | UCS-A /org # **commit-buffer** | Commits the transaction to the system configuration. |

### Example

The following example shows how to delete a SAN connectivity policy named SanConnect52 from the root organization and commit the transaction:

```
UCS-A# scope org /
UCS-A /org # delete san-connectivity-policy SanConnect52
UCS-A /org* # commit-buffer
UCS-A /org #
```

# Creating a vHBA for a SAN Connectivity Policy

If you are continuing from Creating a SAN Connectivity Policy, on page 107, begin this procedure at Step 3.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | UCS-A# **scope org** *org-name* | Enters the organization mode for the specified organization. To enter the root organization mode, enter / as the *org-name*. |
| **Step 2** | UCS-A /org # **scope san-connectivity-policy** *policy-name* | Enters SAN connectivity policy mode for the specified SAN connectivity policy. |
| **Step 3** | UCS-A /org/san-connectivity-policy # **create vhba** *vhba-name* [**fabric** {**a** \| **b**}] [**fc-if** *fc-if-name*] | Creates a vHBA for the specified SAN connectivity policy and enters vHBA mode.<br>This name can be between 1 and 16 alphanumeric characters. You cannot use spaces or any special characters other than - (hyphen), _ (underscore), : (colon), and . (period), and you cannot change this name after the object is saved. |
| **Step 4** | UCS-A /org/san-connectivity-policy/vhba # **set adapter-policy** *policy-name* | Specifies the adapter policy to use for the vHBA. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 5** | UCS-A /org/san-connectivity-policy/vhba # **set identity** {**dynamic-wwpn** {*wwpn* \| **derived**} \| **wwpn-pool** *wwn-pool-name*} | Specifies the WWPN for the vHBA.<br>You can set the storage identity using one of the following options:<br>• Create a unique WWPN in the form *hh*:*hh*:*hh*:*hh*:*hh*:*hh*:*hh*:*hh*.<br>You can specify a WWPN in the range from 20:00:00:00:00:00:00:00 to 20:FF:FF:FF:FF:FF:FF:FF or from 50:00:00:00:00:00:00:00 to 5F:FF:FF:FF:FF:FF:FF:FF.<br>If you want the WWPN to be compatible with Cisco MDS Fibre Channel switches, use the WWPN template **20:00:00:25:B5:XX:XX:XX**.<br>• Derive the WWPN from one burned into the hardware at manufacture.<br>• Assign a WWPN from a WWN pool. |
| **Step 6** | UCS-A /org/san-connectivity-policy/vhba # **set max-field-size** *size-num* | Specifies the maximum size of the Fibre Channel frame payload (in bytes) that the vHBA supports.<br>Enter an integer between 256 and 2112. The default is 2048. |
| **Step 7** | UCS-A /org/san-connectivity-policy/vhba # **set order** {*order-num* \| **unspecified**} | Specifies the PCI scan order for the vHBA. |
| **Step 8** | UCS-A /org/san-connectivity-policy/vhba # **set pers-bind** {**disabled** \| **enabled**} | Disables or enables persistent binding to Fibre Channel targets. |
| **Step 9** | UCS-A /org/san-connectivity-policy/vhba # **set pin-group** *group-name* | Specifies the SAN pin group to use for the vHBA. |
| **Step 10** | UCS-A /org/san-connectivity-policy/vhba # **set qos-policy** *policy-name* | Specifies the QoS policy to use for the vHBA. |
| **Step 11** | UCS-A /org/san-connectivity-policy/vhba # **set stats-policy** *policy-name* | Specifies the statistics threshold policy to use for the vHBA. |
| **Step 12** | UCS-A /org/san-connectivity-policy/vhba # **set template-name** *policy-name* | Specifies the vHBA template to use for the vHBA. If you choose to use a vHBA template for the vHBA, you must still complete all of the configuration not included in the vHBA template, including Steps 4, 7, and 8. |
| **Step 13** | UCS-A /org/san-connectivity-policy/vhba # **set vcon** {**1** \| **2** \| **3** \| **4** \| **any**} | Assigns the vHBA to one or all virtual network interface connections. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 14** | UCS-A /org/san-connectivity-policy/vhba # **commit-buffer** | Commits the transaction to the system configuration. |

### Example

The following example shows how to configure a vHBA for a SAN connectivity policy named SanConnect242 and commit the transaction:

```
UCS-A# scope org /
UCS-A /org* # scope san-connectivity-policy SanConnect242
UCS-A /org/san-connectivity-policy* # create vhba vhba3 fabric a
UCS-A /org/san-connectivity-policy/vhba* # set adapter-policy AdaptPol2
UCS-A /org/san-connectivity-policy/vhba* # set identity wwpn-pool SanPool7
UCS-A /org/san-connectivity-policy/vhba* # set max-field-size 2112
UCS-A /org/san-connectivity-policy/vhba* # set order 0
UCS-A /org/san-connectivity-policy/vhba* # set pers-bind enabled
UCS-A /org/san-connectivity-policy/vhba* # set pin-group FcPinGroup12
UCS-A /org/san-connectivity-policy/vhba* # set qos-policy QosPol5
UCS-A /org/san-connectivity-policy/vhba* # set stats-policy StatsPol2
UCS-A /org/san-connectivity-policy/vhba* # set template-name SanConnPol3
UCS-A /org/san-connectivity-policy/vhba* # set vcon any
UCS-A /org/san-connectivity-policy/vhba* # commit-buffer
UCS-A /org/san-connectivity-policy/vhba #
```

### What to do next

If desired, add another vHBA or an initiator group to the SAN connectivity policy. If not, include the policy in a service profile or service profile template.

# Deleting a vHBA from a SAN Connectivity Policy

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | UCS-A# **scope org** *org-name* | Enters the organization mode for the specified organization. To enter the root organization mode, enter **/** as the *org-name*. |
| **Step 2** | UCS-A /org # **scope san-connectivity-policy** *policy-name* | Enters SAN connectivity policy mode for the specified SAN connectivity policy. |
| **Step 3** | UCS-A /org/san-connectivity-policy # **delete vHBA** *vhba-name* | Deletes the specified vHBA from the SAN connectivity policy. |
| **Step 4** | UCS-A /org/san-connectivity-policy # **commit-buffer** | Commits the transaction to the system configuration. |

### Example

The following example shows how to delete a vHBA named vHBA3 from a SAN connectivity policy named SanConnect242 and commit the transaction:

```
UCS-A# scope org /
UCS-A /org # scope san-connectivity-policy SanConnect242
UCS-A /org/san-connectivity-policy # delete vHBA vHBA3
UCS-A /org/san-connectivity-policy* # commit-buffer
UCS-A /org/san-connectivity-policy #
```

# Creating an Initiator Group for a SAN Connectivity Policy

If you are continuing from Creating a SAN Connectivity Policy, on page 107, begin this procedure at Step 3.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | UCS-A# **scope org** *org-name* | Enters the organization mode for the specified organization. To enter the root organization mode, enter **/** as the *org-name*. |
| **Step 2** | UCS-A /org # **scope san-connectivity-policy** *policy-name* | Enters SAN connectivity policy mode for the specified SAN connectivity policy. |
| **Step 3** | UCS-A /org/san-connectivity-policy # **create initiator-group** *group-name* **fc** | Creates the specified initiator group for Fibre Channel zoning and enters initiator group mode.<br>This name can be between 1 and 16 alphanumeric characters. You cannot use spaces or any special characters other than - (hyphen), _ (underscore), : (colon), and . (period), and you cannot change this name after the object is saved. |
| **Step 4** | UCS-A /org/san-connectivity-policy/initiator-group # **create initiator** *vhba-name* | Creates the specified vHBA initiator in the initiator group.<br>If desired, repeat this step to add a second vHBA initiator to the group. |
| **Step 5** | UCS-A /org/san-connectivity-policy/initiator-group # **set storage-connection-policy** *policy-name* | Associates the specified storage connection policy with the SAN connectivity policy.<br>**Note**<br>This step assumes that you want to associate an existing storage connection policy to associate with the SAN connectivity policy. If you do, continue with Step 10. If you want to create a local storage definition for this policy instead, continue with Step 6. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 6** | UCS-A /org/san-connectivity-policy/initiator-group/storage-connection-def # **create storage-target** *wwpn* | Creates a storage target endpoint with the specified WWPN, and enters storage target mode. |
| **Step 7** | UCS-A /org/san-connectivity-policy/initiator-group/storage-connection-def/storage-target # **set target-path** {**a** \| **b**} | Specifies which fabric interconnect is used for communications with the target endpoint. |
| **Step 8** | UCS-A /org/san-connectivity-policy/initiator-group/storage-connection-def/storage-target # **set target-vsan** *vsan* | Specifies which VSAN is used for communications with the target endpoint. |
| **Step 9** | UCS-A /org/san-connectivity-policy/initiator-group # **commit-buffer** | Commits the transaction to the system configuration. |

### Example

The following example shows how to configure an initiator group named initGroupZone1 with two initiators for a a SAN connectivity policy named SanConnect242, configure a local storage connection policy definition named scPolicyZone1, and commit the transaction:

```
UCS-A# scope org /
UCS-A /org* # scope san-connectivity-policy SanConnect242
UCS-A /org/san-connectivity-policy # create initiator-group initGroupZone1 fc
UCS-A /org/san-connectivity-policy/initiator-group* # set zoning-type sist
UCS-A /org/san-connectivity-policy/initiator-group* # create initiator vhba1
UCS-A /org/san-connectivity-policy/initiator-group* # create initiator vhba2
UCS-A /org/san-connectivity-policy/initiator-group* # create storage-connection-def
scPolicyZone1
UCS-A /org/san-connectivity-policy/initiator-group/storage-connection-def* # create
storage-target
20:10:20:30:40:50:60:70
UCS-A /org/san-connectivity-policy/initiator-group/storage-connection-def/storage-target*
# set
target-path a
UCS-A /org/san-connectivity-policy/initiator-group/storage-connection-def/storage-target*
# set
target-vsan default
UCS-A /org/san-connectivity-policy/initiator-group* # commit-buffer
UCS-A /org/san-connectivity-policy/initiator-group #
```

### What to do next

If desired, add another initiator group or a vHBA to the SAN connectivity policy. If not, include the policy in a service profile or service profile template.

# Creating an SPDM Security Policy

## SPDM Security

Cisco UCS M6, M7, and M8 servers can contain mutable components that could provide vectors for attack against a device itself or use of a device to attack another device within the system. To defend against these attacks, the Security Protocol and Data Model (SPDM) Specification enables a secure transport implementation that challenges a device to prove its identity and the correctness of its mutable component configuration.

SPDM defines messages, data objects, and sequences for performing message exchanges between devices over a variety of transport and physical media. It orchestrates message exchanges between Baseboard Management Controllers (BMC) and end-point devices over a Management Component Transport Protocol (MCTP). Message exchanges include authentication of hardware identities accessing the BMC. The SPDM enables access to low-level security capabilities and operations by specifying a managed level for device authentication, firmware measurement, and certificate management. Endpoint devices are challenged to provide authentication. and BMC authenticates the endpoints and only allows access for trusted entities.

The UCS Manager optionally allows uploads of external security certificates to BMC. A maximum of 40 SPDM certificates is allowed, including native internal certificates. Once the limit is reached, no more certificates can be uploaded. User uploaded certificates can be deleted but internal/default certificates cannot.

A SPDM security policy allows you to specify one of three Security level settings. Security can be set at one of the three levels listed below:

- Full Security:

  This is the highest MCTP security setting. When you select this setting, a fault is generated when any endpoint authentication failure or firmware measurement failure is detected. A fault will also be generated if any of the endpoints do not support either endpoint authentication or firmware measurements.

- Partial Security (default):

  When you select this setting, a fault is generated when any endpoint authentication failure or firmware measurement failure is detected. There will NOT be a fault generated when the endpoint doesn’t support endpoint authentication or firmware measurements.

- No Security

  When you select this setting, there will NOT be a fault generated for any failure (either endpoint measurement or firmware measurement failures).

You can also upload the content of one or more external/device certificates into BMC. Using a SPDM policy allows you to change or delete security certificates or settings as desired. Certificates can be deleted or replaced when no longer needed.

Certificates are listed in all user interfaces on a system.

## SPDM Authentication

The Security Protocol and Data Model (SPDM) is used by the BMC for authentication with the storage controller. It requires that the storage controller firmware is secure booted as well as having a Broadcom certificate chain installed in the slot0. During a firmware update, the Broadcom firmware will retain the older measurements for the storage firmware until the OCR or host reboots. If device authentication fails, the firmware will allow only inventory related commands and no set operations can be performed.

# Creating a SPDM Security Policy

A Security Protocol and Data Model (SPDM) policy can be created to present security alert-level and certificate contents to BMC for authentication.

- UCS-A# **scope org**

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | UCS-A# **scope org** *org-name* | Enters organization mode for the specified organization. To enter the root organization mode, type **/** as the *org-name* . |
| **Step 2** | UCS-A /org # **create spdm-certificate-policy** *policy-name* | Creates a SPDM security certificate policy with the specified policy name, and enters organization SPDM certificate policy mode. |
| **Step 3** | UCS-A /org/spdm-certificate-policy* # **set fault-alert** {**full** \| **partial** \| **no**} | Configures the fault alert level for this policy. |
| **Step 4** | (Optional) UCS-A /org/spdm-certificate-policy* # **set descr** *description* | Provides a description for the SPDM security certificate policy.<br>**Note** If your description includes spaces, special characters, or punctuation, you must begin and end your description with quotation marks. The quotation marks will not appear in the description field of any **show** command output. |
| **Step 5** | UCS-A /org/spdm-certificate-policy # **commit-buffer** | Commits the transaction to the system configuration. |

### Example

The following example shows creating a policy called "test" using an alert level of Partial Security (fault generated when an endpoint authentication or firmware measurement failure is detected). The default policy owner is Local.

```
UCS-A-FI-A /org #create spdm-certificate-policy test
UCS-A-FI-A /org /spdm-certificate-policy* # set?
fault-alert - Configure fault alert setting
desc - Description of policy
policy-owner  - Change ownership of policies
UCS-A-FI-A /org /spdm-certificate-policy* # set fault-alert partial
UCS-A-FI-A /org/spdm-certificate-policy* #commit-buffer
UCS-A-FI-A /org/spdm-certificate-policy# show details

SPDM Certificate Profile:
Name: test
Fault Alert Setting: partial
```

```
Description:
Policy Owner: Local
```

### What to do next

Assign outside security certificates, if desired.

## Loading an Outside SPDM Security Certificate Policy

The SPDM allows you to download an outside security certificate.

### Before you begin

Create a SPDM security certificate policy.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | UCS-A /org # **scope spdm-certificate-policy** | Enters SPDM security certificate policy mode. |
| **Step 2** | UCS-A org/spdm-certificate-policy# **create spdm-cert** *Certificate name* | Creates a SPDM security certificate policy for the specified external certificate,. |
| **Step 3** | UCS-A /org/spdm-certificate-policy* # **set** {*certificate* } | Specifying certificate prompts for the content of the outside certificate. The only supported certificate type is **pem**. |
| **Step 4** | UCS-A /org/spdm-certificate-policy # **commit-buffer** | Commits the transaction to the system configuration. |

The following example shows loading a certificate for Broadcom of type PEM.

### Example

```
UCS-A-FI-A /org/spdm-certificate-policy# create spdm-cert?
Name - Certificate name

UCS-A-FI-A /org/spdm-certificate-policy# create spdm-cert Broadcom
UCS-A-FI-A /org/spdm-certificate-policy/spdm-cert* # set?
certificate - Certificate content

UCS-A-FI-A /org/spdm-certificate-policy/spdm-cert* # set certificate
{enter certificate content}
UCS-A-FI-A /org/spdm-certificate-policy/spdm-cert* # commit-buffer
UCS-A-FI-A /org/spdm-certificate-policy/spdm-cert# show detail
SPDM Certificate:
Name: Broadcom
Certificate Type: pem
Certificate Content:
```

## Displaying the Security Policy Fault Alert Level

After the policy is created, you can check the alert level for the SPDM policy.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | UCS-A /org/spdm-certificate-policy # **show fault-alert**<br>**Example:**<br>`UCS-A /server/cimc/spdm-certificate #show fault-alert` | The returned result shows that the setting for this SPDM policy is Partial, the default.<br>`SPDM Fault Alert Setting: Partial` |

## Viewing the Certificate Inventory

You can view what SPDM certificates have been uploaded and also request further details for a specified certificate.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | UCS-A # **scope server** *server* | |
| **Step 2** | UCS-A/server # **scope cimc** *server* | |
| **Step 3** | UCS-A/server/cimc # **scope spdm** *server* | |
| **Step 4** | UCS-A/server/cimc/spdm # **show certificate** | The returned result shows the certificate inventory. |
| **Step 5** | UCS-A/server/cimc/spdm # **show certificate** *certificate-id***detail**<br>**Example:**<br>`UCS-A /server/cimc/spdm-certificate #show certificate 3 detail`<br>`Certificate Information`<br>`Certificate Id : 3`<br>`Subject Country Code (C) : US`<br>`Subject State (ST) : Colorado`<br>`Subject Organization (O) : Broadcom Inc.`<br>`Subject Organization Unit(OU) : NA`<br>`Subject Common Name (CN) : NA`<br>`Issuer Country Code (C) : US`<br>`Issuer State (ST) : Colorado`<br>`Issuer City (L) : Colorado Springs`<br>`Issuer Organization (O) : Broadcom Inc.`<br>`Issuer Organization Unit(OU) : NA`<br>`Issuer Common Name (CN) : NA` | The returned result shows the certificate ID, identifiers, and expiration date. |

| | Command or Action | Purpose |
|---|---|---|
| | Valid From : Oct 23 00:25:13 2019 GMT<br>Valid To : Apr 8 10:36:14 2021 GMT<br>UserUploaded : Yes<br>Certificate Content : <Certificate String><br>Certificate Type : PEM | |
| **Step 6** | UCS-A /org/spdm-certificate-policy/certificate # **show**<br>**Example:**<br>SPDM Certificate:<br>Name SPDM Certificate Type<br>---<br>cert1 Pem<br>**Example:**<br>UCS-A /server/cimc/spdm-certificate/certificate #up<br>UCS-A /server/cimc/spdm-certificate #show<br>SPDM Certificate Policy:<br>Name Fault Alert Setting<br>---<br>Broadcom Full | The returned result shows the type of certificate details.<br>The returned result shows the fault alert setting. |

# Deleting a SPDM Policy

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | UCS-A# **scope org** *org-name* | Enters organization mode for the specified organization. To enter the root organization mode, type / as the org-name. |
| **Step 2** | UCS-A /org # **delete spdm-certificate-policy** *policy-name* | Deletes the specified SPDM control policy. |
| **Step 3** | UCS-A /org # **commit-buffer** | Commits the transaction to the system configuration. |

### Example

The following example deletes a power control policy called VendorPolicy2 and commits the transaction:

```
UCS-A# scope org /
UCS-A /org # delete spdm-certificate-policy VendorPolicy2
UCS-A /org* # commit-buffer
UCS-A /org #
```

# Deleting an Initiator Group from a SAN Connectivity Policy

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | UCS-A# **scope org** *org-name* | Enters the organization mode for the specified organization. To enter the root organization mode, enter **/** as the *org-name*. |
| **Step 2** | UCS-A /org # **scope san-connectivity-policy** *policy-name* | Enters SAN connectivity policy mode for the specified SAN connectivity policy. |
| **Step 3** | UCS-A /org/san-connectivity-policy # **delete initiator-group** *group-name* | Deletes the specified initiator group from the SAN connectivity policy. |
| **Step 4** | UCS-A /org/san-connectivity-policy # **commit-buffer** | Commits the transaction to the system configuration. |

### Example

The following example shows how to delete an initiator group named initGroup3 from a SAN connectivity policy named SanConnect242 and commit the transaction:

```
UCS-A# scope org /
UCS-A /org # scope san-connectivity-policy SanConnect242
UCS-A /org/san-connectivity-policy # delete initiator-group initGroup3
UCS-A /org/san-connectivity-policy* # commit-buffer
UCS-A /org/san-connectivity-policy #
```

# Configuring an Aero Controller Storage Profile

## Autoconfiguration Mode for Storage Controllers

Autoconfiguration mode for storage controllers are supported on Cisco UCS C-Series Rack servers (M6, M7, and M8), B-Series Blade servers (M6), and X-Series Compute Nodes (M6, M7, and M8).

C-series M6 servers support PCIe SAS316-port storage controllers for Direct Attached Storage. Controllers support an Autoconfiguration mode in which the state of a newly inserted disk is automatically moved to the Unconfigured-Good state.

Because of this, you can choose whether or not to use Autoconfiguration by creating a Storage Profile and associating it with the server. The default is that the automatic configuration feature is disabled, which retains the drive state when the server is rebooted.

If Autoconfiguration is used, you must select a drive state from one of the following:

- Unconfigured-Good
- JBOD
- RAID0 (RAID0 WriteBack)

This is because the controller firmware changes the behavior of systemPD to EPD-PT. EPD-PT is internally a RAID0 volume without any drive DDF metadata. The controller stores the metadata for identifying it as a RAID0 volume. The EPD-PT drives are considered as JBOD drives so the drive status is reported as JBOD and online.

Controller supports the following models:

- UCSC-RAID-M6T
- UCSC-RAID-M6HD
- UCSC-RAID-M6SD
- UCSX-X10C-RAIDF
- UCSC-RAID-HP

The table below shows the behavior of Autoconfiguration in different scenarios.

| Autoconfig Mode | Reboot/OCR | Hotplug | User Action |
|---|---|---|---|
| Unconfigured-Good (OFF) | • All Unconfigured-Good drives remain Unconfigured-Good.<br>• All previously configured JBOD remain JBOD. | • Inserted drive remains Unconfigured-Good.<br>• JBOD from a different server remains Unconfigured-Good on this controller. | Disabling Autoconfig has no impact on the existing configuration<br>Any JBOD device remains as JBOD across controller boot.<br>Any Unconfigured-Good remains unconfiguredgood across controller boot. |

| Autoconfig Mode | Reboot/OCR | Hotplug | User Action |
|---|---|---|---|
| JBOD | • All Unconfigured-Good are converted to JBOD. | Newly inserted unconfigured device is converted to JBOD. | All Unconfigured-Good drives (non-user created) on the controller while running Autoconfig is converted to JBOD.<br>User created Unconfigured-Good drive remains Unconfigured-Good until next reboot. During reboot Unconfigured-Good gets converted to JBOD. |
| RAID0 (RAID0 WriteBack) | • All Unconfigured-Good converted to RAID0 WriteBack. | Newly inserted unconfigured device is converted to RAID0 WriteBack. | All Unconfigured-Good drives (non-user created) on the controller while running Autoconfig is converted to RAID0 WriteBack.<br>User created Unconfigured-Good remains Unconfigured-Good across controller reboot.<br>Any RAID0 WriteBack device remains as RAID0 WriteBack across controller reboot. |

Selecting EPD-PT (JBOD) as the default configuration does not retain the Unconfigured-Good state across host reboot. The drive state can be retained by disabling the automatic configuration feature. If the Autoconfig option is used, the default automatic configuration will always mark a drive as Unconfigured-Good.

When Autoconfig is selected, then the drive is configured to the desired drive state, the JBOD and unconfigured drives will set the drive state accordingly on the next controller boot or OCR,

The following table shows sample use cases for different Autoconfig scenarios.

| Use Case Scenario | Autoconfig Option |
|---|---|
| Using the server for JBOD Only (for example: Hyper converged, Hadoop data node etc ) | JBOD |
| Using the server for RAID volume (for example: SAP HANA database) | Unconfigured-Good |
| Using the server for Mixed JBOD and RAID volume | Unconfigured-Good |
| Using the server for per drive RAID0 WriteBack (for example: Hadoop data node) | RAID0 WriteBack |

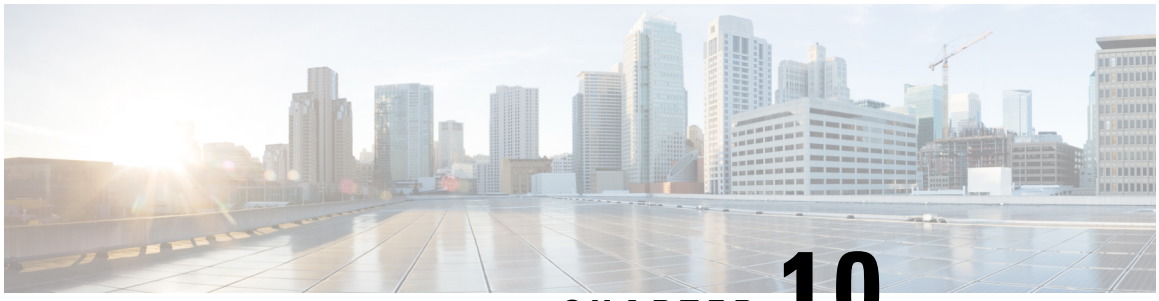
**Note** When using UCSX-X10C-RAIDF or UCSC-RAID-M6T controllers with Autoconfiguration Mode set to RAID0, drives in *Unconfigured Good* state may not transition to *Online* and RAID0 LUNs will not be created after a storage profile redeploy and server reboot. This is a known behavior and differs from UCSC-RAID-HP controllers, where drives transition to *Online* and LUNs are created as expected.

## Creating an Autoconfiguation Profile

You can include the storage Autoconfiguration (Auto Config) mode option in your storage profile and unconfigure it when no longer needed. Changes will take effect on the next system boot. Auto Config for storage is only available on Cisco UCS M6 servers with Aero controllers.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | UCS-A# **scope org** *org-name* | Enters organization mode for the specified organization. To enter the root organization mode, type / as the *org-name* . |
| **Step 2** | UCS-A/org# **scope storage-profile** *profile-name* | Enters the storage profile for the specified profile. |
| **Step 3** | UCS-A/org/storage-profile# **show detail expand** | Shows a detailed view of the Storage Profile. If Auto Config Mode has not been enabled for this storage profile, or no Aero controller is present, you should not see an entry for Auto Config Mode. If Auto Config is not configured, inserted devices will retain their state on system reboot. |
| **Step 4** | UCS-A/org/storage-profile# **set auto-config-mode** jbod \| raid-0 \| unconfigured-good \| unspecified | Enables Auto Config Mode and sets the disk configuration mode to the desired state. If no further parameters are specified, all inserted devices will be tagged as Unconfigured Good on reboot. Enter unconfigured if you wish to disable Auto Config mode. |
| **Step 5** | UCS-A/org/storage-profile# **commit-buffer** | Commits the transaction to the system configuration. |



CHAPTER 10

# Storage Profiles



# Storage Profiles

To allow flexibility in defining the number of storage disks, roles and usage of these disks, and other storage parameters, you can create and use storage profiles. A storage profile encapsulates the storage requirements for one or more service profiles. LUNs configured in a storage profile can be used as boot LUNs or data LUNs, and can be dedicated to a specific server. You can also specify a local LUN as a boot device. However, LUN resizing is not supported. The introduction of storage profiles allows you to do the following:

- Configure multiple virtual drives and select the physical drives that are used by a virtual drive. You can also configure the storage capacity of a virtual drive.
- Configure the number, type and role of disks in a disk group.
- Associate a storage profile with a service profile.

You can create a storage profile both at an org level and at a service-profile level. A service profile can have a dedicated storage profile as well as a storage profile at an org level.

# Cisco 24G Tri-Mode RAID and HBA Controllers

Beginning with 4.3(5a), Cisco UCS Manager supports the Cisco Tri-Mode RAID and HBA controllers. These controllers can be plugged directly into a dedicated slot and support RAID (RAID-0, RAID-1, RAID-5, RAID-6, RAID-10, RAID-50, and RAID-60) and JBOD mode configurations.

***Table 2: Controller Support Details for X-Series Servers***

| Controller (PID) | Product Name | Supported Servers | Maximum Number of Drives Supported | Supported Controller Type |
|---|---|---|---|---|
| UCSX-RAID-M1L6 | 24G Tri-Mode M1 RAID Controller w/4GB FBWC 16D | UCS-X210c-M8 | 6 | RAID (RAID0, 1, 5, 6, 10, 50) and JBOD |

***Table 3: Controller Support Details for C-Series Servers***

| Controller (PID) | Product Name | Supported Servers | Maximum Number of Drives Supported | Supported Controller Type |
|---|---|---|---|---|
| UCSC-RAID-M1L16 | Cisco 24G Tri-Mode M1 RAID 4GB Flash Backed Write Cache (FBWC) 16D | UCSC-C220-M7S<br>UCSC-C225-M8S<br>UCSC-C220-M8S | 16 | RAID (RAID0, 1, 5, 6, 10, 50, 60) and JBOD |
| UCSC-RAID-M1L32 | Cisco 24G Tri-Mode M1 RAID 4GB Flash Backed Write Cache (FBWC) 32D | UCSC-C240-M8SX | 32 | RAID (RAID0, 1, 5, 6, 10, 50, 60) and JBOD |
| UCSC-RAID-MP1L32 | Cisco 24G Tri-Mode MP1 RAID 4GB Flash Backed Write Cache (FBWC) 32D | UCSC-C240-M7SX<br>UCSC-C245-M8SX<br>UCSC-C240-M8SX | 32 | RAID (RAID0, 1, 5, 6, 10, 50, 60) and JBOD |
| UCSC-RAID-MP1LL32 | Cisco 24G Tri-Mode MP1 RAID 4GB Flash Backed Write Cache (FBWC) 32D | UCSC-C240-M8SX | 32 | RAID (RAID0, 1, 5, 6, 10, 50, 60) and JBOD |

| Controller (PID) | Product Name | Supported Servers | Maximum Number of Drives Supported | Supported Controller Type |
|---|---|---|---|---|
| UCSC-RAID-MP1LL32 | Cisco 24G Trimode MP1 RAID Controller with 8GB FBWC LFF 32D | UCSC-C240-M8L | 32 | RAID (RAID0, 1, 5, 6, 10, 50, 60) and JBOD |
| UCSC-HBA-MP1LL32 | Cisco 24G Tri-Mode MP1 HBA LFF 32D | UCSC-C240-M8SX | 32 | JBOD |
| UCSC-HBA-M1EXJBOD | Cisco 24G Tri-Mode M1 HBA EXJBOD | UCSC-C240-M8SX | 32 | JBOD |
| UCSC-HBA-M1L16 | Cisco 24G Tri-Mode M1 HBA | UCSC-C240-M8SX<br>UCSC-C240-M7SX<br>UCSC-C245-M8SX | 32 | JBOD |
| | | UCSC-C220-M7S<br>UCSC-C225-M8S<br>UCSC-C220-M8S | 16 | |
| | | UCSC-C240-M8SX | 14 | |

### Advantages of Cisco 24G Tri-Mode M1 RAID and HBA Controllers

- Uses Enterprise Key Management (EKMS) for remote key management, enhancing the physical security of data.
- Allows quick integration of new vendors and adaptors via Out-Of-Band management.
- 5% of maximum drive space is reserved to allow slight variance in drive sizes over time.

### Expected Behaviors, Recommendations, and Known Limitations of Cisco 24G Tri-Mode M1 RAID Controllers

- In a RAID-5 setup, hot plugging a drive with 3 HDDs degrades the virtual drive. Inserting a JBOD SSD into the same slot marks the drive as failed due to a mismatch, even though the drive is not faulty.
- In a RAID-1 setup, the Virtual Drive creation supports a maximum of two physical disks. Configuring RAID-1 with more than two physical disks will result in a configuration failure.
- Self-encrypting drives (SEDs) and non-SEDs cannot be mixed when creating a virtual drive using the Local Disk Configuration Policy. This limitation applies specifically to the UCSC-RAID-M1L16 and UCSC-RAID-M1L32 controllers.
- Secure erase operations on remotely secured drives is blocked on servers that have no connections to a Key Management Interoperability Protocol (KMIP) server.

- Physically removing a drive from a volume with 3 HDDs to a different slot on the same server results in it being inventoried as a foreign configuration drive due to tight coupling with the original slot.
- Physical disks are always in **JBOD** mode, so the **Unconfigured Good** state is irrelevant.
- There will be some unused free space on the disk when virtual drives are created using the **Expand To Available** option.
- It is recommended to limit the number of virtual drives in a drive group on these RAID controllers to a maximum of 8.
- Ensuring the supercap is fully charged when creating RAID volumes with caching enabled is necessary; otherwise, the creation of virtual drives will fail, resulting in a service profile association failure. However, re-associating the service profile will create the virtual drives, and the association will be successful.
- Each virtual drive (VD) must be deleted one at a time, ensuring the completion of each deletion before starting another. No other operations, including VD deletions, can be performed during an ongoing transformation. Associating a service profile will lead to a configuration failure until the VD transformation is complete. The server must be powered on, for the transformation to complete before associating the service profile. It is recommended to use a **Scrub Policy** to delete VDs.
- When moving RAID groups from one server to another, the RAID configuration is slot-aware, and the drive slots must remain the same.
- During drive migration between servers, ensure both servers have the same controller type (UCSC-RAID-M1L16/UCSC-RAID-M1L32). Power down server 1 before migration and place the entire set of drives in the same slots in server 2, ensuring that server 2 has all empty slots to accommodate the drives. It is recommended to move the entire set of drives from server 1 to server 2.
- Only one Dedicated Hot Spare (DHS) per drive group array is allowed. An attempt to increase the DHS count to two or more to an existing Disk Group Policy results in an error.
- The Programmable System-on-Chip (PSoC) is integrated into the controller firmware package, eliminating the need for separate PSoC firmware updates.
- The *Use JBOD* option in drive group creation is redundant since drives default to JBOD mode when no volumes are configured.
- The controllers support both read and write cache policies and supports the two configurations: Cache enabled for both or Cache disabled for both.
- The following policy or settings are not supported:
  - I/O policy, Drive cache policy, and Access policy
  - Auto Configuration Mode
  - Online Insertion and Removal (OIR)
  - Global Hot Spare
  - Import Foreign Configuration
  - Clear Foreign Configuration
- Importing or clearing configuration options for controllers should be disabled when a foreign configuration drive is placed on the server. Currently, these options are enabled but result in a warning/error stating that the operation is not supported.

**Note** For more information, see the **Guidelines** section in Security Policies for Self-Encrypting Drives, on page 213.

# Servers and Storage Support

*Table 4: Cisco UCS C240 M8 Server*

| C-Series Servers | Drives/Storage Controllers |
|---|---|
| Cisco UCS C240 M8 Server | |

| C-Series Servers | Drives/Storage Controllers |
|---|---|
| | • UCSC-C240-M8SX<br>• Up to 28 front-facing and 4 rear facing SFF SAS/SATA HDDs or SAS/SATA SSDs or NVMe SSDs<br>• Up to 8 of the front slots can be direct-attach NVMe<br>• Up to 4 rear drives (SAS/SATA/NVME), the rear slots can be moved to direct-attach<br>• 24G Tri-Mode M1 RAID Controller 32 Drvies (UCSC-RAID-MP1L32)<br>• Cisco 24G Trimode M1 HBA Controller 16 Drives (UCSC-HBA-M1L16)<br>• Cisco 24G Tri-Mode M1 HBA EXJBOD (UCSC-HBA-M1EXJBOD). For supported configurations, see Hardware Compatibility List (HCL).<br>• Cisco 6Gb/s M.2 SATA RAID Controller (UCS-M2-HWRAID2)<br>• Rear Hot-plug M.2 module (Riser 3) (UCSC-M2RR-240M8)<br>• Rear Hot-plug M.2 module (MLOM) (UCSC-M2RM-M8)<br>• UCSC-C240-M8L<br>• Up to 4 rear drives (SAS/SATA/NVME) support is available. The rear slots can be moved to direct-attach<br>• Cisco 24G Trimode M1 RAID Controller with 8GB FBWC LFF 32D (UCSC-RAID-MP1LL32)<br>• Cisco 24G Trimode M1 HBA LFF Controller (32 Drives) (UCSC-HBAMP1LL32)<br>• Cisco 24G Tri-Mode M1 HBA EXJBOD (UCSC-HBA-M1EXJBOD). For supported configurations, see Hardware Compatibility List (HCL).<br>• Cisco 6Gb/s M.2 SATA RAID Controller (UCS-M2-HWRAID2)<br>• Rear Hot-plug M.2 module (Riser 3) |

| C-Series Servers | Drives/Storage Controllers |
| --- | --- |
| | (UCSC-M2RR-240M8)<br>• Rear Hot-plug M.2 module (MLOM) (UCSC-M2RM-M8)<br>• UCSC-C240-M8E3S<br>• Supports up to 32 E3.S drives.<br>• Supports the following M.2 controllers:<br>• UCS-M2-HWRAID2<br>• UCSC-M2RR-240M8<br>• UCSC-M2RM-M8<br>• Up to 4 rear E3.S drive support is available. The rear slots can be moved to direct-attach. |

*Table 5: Cisco UCS C220 M8 Server*

| Servers | Drives/Storage Controllers |
|---|---|
| Cisco UCS C220 M8 Server | • UCSC-C220-M8S<br>• Up to 10 drives with a mix of SAS/SATA/NVMe U.3<br>• Upto 8 direct-attach support for slots 1-4 and 6-9 NVMe SSDs. No rear slot support.<br>• Cisco 24 Gbps Tri-Mode M1 RAID 4GB Flash Backed Write Cache (FBWC) 16 drives (UCSC-RAID-M1L16)<br>• Cisco 24 Gbps Tri-mode Host Bus Adapter (HBA) controller (UCSC-HBA-M1L16)<br>• Cisco 24G Tri-Mode M1 HBA EXJBOD (UCSC-HBA-M1EXJBOD). For supported configurations, see Hardware Compatibility List (HCL).<br>• Cisco 6Gb/s M.2 SATA RAID Controller (UCS-M2-HWRAID2)<br>• Rear Hot-plug M.2 module (MLOM) (UCSC-M2RM-M8)<br>• UCSC-C220-M8E3S<br>• Supports up to 16 E3.S drives<br>• Supports the following M.2 controllers:<br>• UCS-M2-HWRAID2<br>• UCSC-M2RM-M8 |

*Table 6: Cisco UCS C225 M8 Server*

| Servers | Drives/Storage Controllers |
|---|---|
| Cisco UCS C225 M8 Server | • UCSC-C225-M8S<br>• Up to 10 SAS/SATA or NVMe disk drives<br>• Cisco 24 Gbps Tri-mode RAID controller that supports SAS4 or NVMe hardware RAID (UCSC-RAID-HP)<br>• Cisco 24 Gbps Tri-Mode M1 RAID 4GB Flash Backed Write Cache (FBWC) 16 drives (UCSC-RAID-M1L16)<br>• Cisco 24 Gbps Tri-mode Host Bus Adapter (HBA) controller (UCSC-HBA-M1L16)<br>• Up to 4 direct-attach NVMe SSDs<br>• UCSC-C225-M8N<br>• Up to 10 direct attach NVMe SSDs<br>• All 10 NVMe drives connected at PCIe Gen4 x4 |

*Table 7: Cisco UCS C245 M8 Server*

| Servers | Drives/Storage Controllers |
|---|---|
| Cisco UCS C245 M8 Server | • UCSC-C245-M8SX<br>• Cisco 24 Gbps Tri-mode RAID controller that supports SAS4 or NVMe hardware RAID (UCSC-RAID-HP)<br>• Cisco 24 Gbps Tri-mode RAID controller with 4GB Flash Backed Write Cache (FBWC) 32 drives (UCSC-RAID-MP1L32)<br>• Cisco 24G Tri-mode M1 (16-port) HBA controllers (UCSC-HBA-M1L16)<br>• 24 front facing SFF SAS/SATA HDDs or SAS/SATA SSDs or NVMe SSDs<br>• Optionally, up to four of the slots can be direct-attach NVMe. These drives must be placed in front drive bays 1, 2, 3, and 4 only. The rest of the bays (5 - 24) can be populated with SAS/SATA/NVMe SSDs or HDDs.<br>• Optionally, up to four direct-attach SFF rear-facing SAS/SATA/NVMe drives<br>• A mini-storage module connector on the motherboard supports a boot-optimized RAID controller carrier that holds up to two SATA M.2 SSDs. Mixing different capacity SATA M.2 SSDs is not supported.<br>• 8GB FlexMMC utility storage for staging of firmware and other user data. 8GB FlexMMC storage is built into the motherboard |

*Table 8: Cisco UCS C240 M7 Server*

| Server | Drives/Storage Controllers |
|---|---|
| Cisco UCS C240 M7 Server | **Drives**<br>All RAID controllers are only supported on UCSC-C240-M7SX. For UCSC-C240-M7SN, drives are controlled directly from the CPU.<br>• **UCSC-C240-M7SX**<br>  • Up to 24 front facing SFF SAS/SATA HDDs or SAS/SATA SSDs or NVMe SSDs<br>  • Optionally, up to four of the slots can be direct-attach NVMe.<br>• **UCSC-C240-M7SN**<br>  • Up to 24 front NVMe drives (only).<br>  • Optionally, up to 4 rear NVMe drives (only)<br>  • Two CPUs are required when choosing NVMe SSDs<br>• **Controllers**<br>  • UCSC-RAID-MP1L32 — 24G Tri-Mode MP1 RAID Controller<br>  • UCSC-RAID-HP — Cisco Tri-Mode 24G SAS RAID Controller<br>  • UCSC-RAID-SD-D — Cisco 12G SAS RAID Controller<br>  • UCSC-HBA-M1L16 — 24G Tri-Mode M1 HBA for 16 Drives (SAS/SATA/U.3 NVMe)<br>  • UCSC-SAS-T-D — Cisco M6 12G SAS HBA for (16 Drives) |

*Table 9: Cisco UCS C220 M7 Server*

| Server | Drives/Storage Controllers |
|---|---|
| Cisco UCS C220 M7 Server | **Drives**<br>• **UCSC-C220-M7S** - Up to 10 SFF SAS/SATA hard drives (HDDs) or SAS/SATA/NVMe solid state drives (SSDs).<br>• **UCSC-C220-M7N**Up to 10 2.5-inch direct-attach NVMe SSDs<br>• **Controllers**<br>  • UCSC-RAID-M1L16 — 24G Tri-Mode M1 RAID Controller<br>  • UCSC-RAID-HP — Cisco Tri-Mode 24G SAS RAID Controller<br>  • UCSC-RAID-T-D — Cisco M6 12G SAS RAID Controller<br>  • UCSC-HBA-M1L16 — 24G Tri-Mode M1 HBA for 16 Drives (SAS/SATA/U.3 NVMe)<br>  • UCSC-SAS-T-D — Cisco M6 12G SAS HBA for (16 Drives) |

*Table 10: Cisco UCS C225 M6 Server*

| Servers | Drives/Storage Controllers |
|---|---|
| Cisco UCS C225 M6 Server | • UCS C225 M6SX and UCS C245 M6SX in C225-SFF (10 front SAS/SATA drives)<br>• 2 M.2 2280 Drives on UCS-M2-HWRAID<br>• Direct Attached NVMe drives (10 NVMe drives in the front) |

*Table 11: Cisco UCS C245 M6 Server*

| Servers | Drives/Storage Controllers |
|---|---|
| Cisco UCS C245 M6 | • Dual UCS C245 M6SX<br>16 SAS/SATA HDD<br>• UCS C245 M6SX Plus<br>28 SAS/SATA HDD<br>• 2 M.2 2280 Drives on UCS-M2-HWRAID<br>• Directly Attached NVMe on rear risers(up to 4 NVMe SSD) |

**Note** It is recommended not to move drives between SAS3 and SAS4 controllers to avoid data loss.

# Cisco M.2 Controller on Cisco UCS C-Series M8 and X-Series M8 Servers

Beginning with 4.3(6a), Cisco UCS Manager introduces support for the following controllers on Cisco UCS M8 server models.

Supported Controllers and Servers:

- UCSX-M2I-HWRD-FPS: Cisco UCS X210c M8 Compute Node
- UCS-M2-HWRAID2: Cisco UCS C240 M8 Server and UCS C220 M8 Server
- UCSC-M2RR-240M8: Cisco UCS C240 M8 Server
- UCSC-M2RM-M8: Cisco UCS C240 M8 Server and UCS C220 M8 Server

# Cisco Boot Optimized M.2 RAID Controller

Beginning with 4.0(4a), Cisco UCS Manager supports Cisco boot optimized M.2 RAID controller (UCS-M2-HWRAID), which is based on Marvell® 88SE92xx PCIe to SATA 6Gb/s controller. It is supported on the following servers:

- Cisco UCS C240 M7 Server
- Cisco UCS C220 M7 Server
- Cisco UCS C225 M6 Server
- Cisco UCS C245 M6 Server
- Cisco UCS C220 M6 Server

- Cisco UCS C240 M6 Server
- Cisco UCS C220 M5 Server
- Cisco UCS C240 M5 Server
- Cisco UCS C480 M5 Server
- Cisco UCS B200 M5 Server
- Cisco UCS B480 M5 Server

The following M.2 drives are managed by the Cisco boot optimized M.2 RAID controller:

- 240GB M.2 6G SATA SSD
- 960GB M.2 6G SATA SSD

The Cisco boot optimized M.2 RAID controller supports only RAID1/JBOD (default - JBOD) mode and only UEFI boot mode.

### Limitations of Cisco boot optimized M.2 RAID controller

- Existing LUN migration is not supported.
- **Local Disk Configuration** policy is not supported.
- The number of LUNs that can be created is limited to one because creating a single LUN uses the entire disk capacity.
- LUN is created using the **Local LUN** tab (see Creating Local LUNs, on page 165) under storage profile and not using the controller definitions.
- You cannot mix different capacity M.2 drives.
- You cannot rename an orphan virtual drive on a blade or a rack server.

# Cisco Boot Optimized M.2 NVMe RAID Controller

Beginning with 4.3(4a), Cisco UCS Manager supports Cisco boot optimized M.2 NVMe RAID controller (UCS-M2-NVRAID), which is based on a Marvell® 88SE92xx PCIe to SATA 6Gb/s controller. It is supported on the following servers:

- Cisco UCS C220 M7 Server
- Cisco UCS C240 M7 Server

The following M.2 drives are managed by the Cisco boot optimized M.2 RAID controller:

- 960GB M.2 6G SATA SSD
- 400GB M.2 6G SATA SSD

The Cisco boot optimized M.2 RAID controller supports only RAID0, RAID1, or JBOD (default - JBOD) mode and only UEFI boot mode.

### Limitations of Cisco Boot Optimized M.2 NVMe RAID Controller

- Existing LUN migration is not supported.
- **Local Disk Configuration** policy is not supported.
- The number of LUNs that can be created is limited to one because creating a single LUN uses the entire disk capacity.
- LUN is created using the **Local LUN** tab (see Creating Local LUNs, on page 165) under storage profile and not using the controller definitions.
- You cannot mix different capacity M.2 drives.
- You cannot rename an orphan virtual drive on a blade or a rack server.

# Disk Groups and Disk Group Configuration Policies

You can select and configure the disks to be used for storage. A logical collection of these physical disks is called a disk group. Disk groups allow you to organize local disks. The storage controller controls the creation and configuration of disk groups.

A disk group configuration policy defines how a disk group is created and configured. The policy specifies the RAID level to be used for the disk group. It also specifies either a manual or an automatic selection of disks for the disk group, and roles for disks. You can use a disk group policy to manage multiple disk groups. However, a single disk group can be managed only by one disk group policy.

A hot spare is an unused extra disk that can be used by a disk group in the case of failure of a disk in the disk group. Hot spares can be used only in disk groups that support a fault-tolerant RAID level.

## Virtual Drives

A disk group can be partitioned into virtual drives. Each virtual drive appears as an individual physical device to the Operating System.

All virtual drives in a disk group must be managed by using a single disk group policy.

### Configuration States

Indicates the configuration states of a virtual drive. Virtual drives can have the following configuration states:

- Applying—Creation of the virtual drive is in progress.
- Applied—Creation of the virtual drive is complete, or virtual disk policy changes are configured and applied successfully.
- Failed to apply—Creation, deletion, or renaming of a virtual drive has failed due to errors in the underlying storage subsystem.
- Orphaned—The service profile that contained this virtual drive is deleted or the service profile is no longer associated with a storage profile.

**Note** Orphaned LUNs cannot be used for booting OS. Although an image can be installed on these LUNs, booting from these drives will fail. To use any specific orphaned LUN, you must reassociate the storage profile, which will return it to the "Equipped" presence state.

- Not in use—The service profile that contained this virtual drive is in the disassociated state.

### Deployment States

Indicates the actions that you are performing on virtual drives. Virtual drives can have the following deployment states:

- No action—No pending work items for the virtual drive.
- Creating—Creation of the virtual drive is in progress.
- Deleting—Deletion of the virtual drive is in progress.
- Modifying—Modification of the virtual drive is in progress.
- Apply-Failed—Creation or modification of the virtual drive has failed.

### Operability States

Indicates the operating condition of a virtual drive. Virtual drives can have the following operability states:

- Optimal—The virtual drive operating condition is good. All configured drives are online.
- Degraded—The virtual drive operating condition is not optimal. One of the configured drives has failed or is offline.
- Cache-degraded—The virtual drive has been created with a write policy of **write back** mode, but the BBU has failed, or there is no BBU.

**Note** This state does not occur if you select the **always write back** mode.

- Partially degraded—The operating condition in a RAID 6 virtual drive is not optimal. One of the configured drives has failed or is offline. RAID 6 can tolerate up to two drive failures.
- Offline—The virtual drive is not available to the RAID controller. This is essentially a failed state.
- Unknown—The state of the virtual drive is not known.

### Presence States

Indicates the presence of virtual drive components. Virtual drives have the following presence states:

- Equipped—The virtual drive is available.
- Mismatched—A virtual drive deployed state is different from its configured state.

- Missing—Virtual drive is missing.

# RAID Levels

The RAID level of a disk group describes how the data is organized on the disk group for the purpose of ensuring availability, redundancy of data, and I/O performance.

The following are features provided by RAID:

- Striping—Segmenting data across multiple physical devices. This improves performance by increasing throughput due to simultaneous device access.
- Mirroring—Writing the same data to multiple devices to accomplish data redundancy.
- Parity—Storing of redundant data on an additional device for the purpose of error correction in the event of device failure. Parity does not provide full redundancy, but it allows for error recovery in some scenarios.
- Spanning—Allows multiple drives to function like a larger one. For example, four 20 GB drives can be combined to appear as a single 80 GB drive.

The supported RAID levels include the following:

- Disable Local Storage—(Supported for PCH SSD Controller Definition) This disk policy mode is to disable the SATA AHCI Controller. This mode can be set only when disks are not present under the SATA AHCI controller. To re-enable this controller and to bring the controller back to its default value (AHCI), you can select No RAID mode or No Local Storage mode.
- No Local Storage—(Supported for PCH SSD Controller Definition) For a diskless server or a SAN only configuration. If you select this option, you cannot associate any service profile which uses this policy with a server that has a local disk.
- RAID 0 Striped— (Supported for PCH SSD Controller Definition) Data is striped across all disks in the array, providing fast throughput. There is no data redundancy, and all data is lost if any disk fails.
- RAID 1 Mirrored— (Supported for PCH SSD Controller Definition) Data is written to two disks, providing complete data redundancy if one disk fails. The maximum array size is equal to the available space on the smaller of the two drives.
- Any Configuration—(Supported for PCH SSD Controller Definition) For a server configuration that carries forward the local disk configuration without any changes.
- No RAID—(Supported for PCH SSD Controller Definition) Allthe disks can be used individually without interdependency similarto JBOD disks. If you choose No RAID and you apply this policyto a server that already has an operating system with RAID storageconfigured, the system does not remove the disk contents.Therefore, there may be no visible differences on the server afteryou apply the No RAID mode. This can lead to a mismatchbetween the RAID configuration in the policy and the actual diskconfiguration shown in the Inventory > Storage tab for the server.To make sure that any previous RAID configuration informationis removed from a disk, apply a scrub policy that removes all diskinformation after you apply the No RAID configuration mode.
- RAID 5 Striped Parity— (Not supported for PCH SSD Controller Definition) Data is striped across all disks in the array. Part of the capacity of each disk stores parity information that can be used to reconstruct data if a disk fails. RAID 5 provides good data throughput for applications with high read request rates.

RAID 5 distributes parity data blocks among the disks that are part of a RAID-5 group and requires a minimum of three disks.

- RAID 6 Striped Dual Parity— (Not supported for PCH SSD Controller Definition) Data is striped across all disks in the array and two sets of parity data are used to provide protection against failure of up to two physical disks. In each row of data blocks, two sets of parity data are stored.

  Other than addition of a second parity block, RAID 6 is identical to RAID 5 . A minimum of four disks are required for RAID 6.

- RAID 10 Mirrored and Striped— (Not supported for PCH SSD Controller Definition) RAID 10 uses mirrored pairs of disks to provide complete data redundancy and high throughput rates through block-level striping. RAID 10 is mirroring without parity and block-level striping. A minimum of four disks are required for RAID 10.
- RAID 50 Striped Parity and Striped— (Not supported for PCH SSD Controller Definition) Data is striped across multiple striped parity disk sets to provide high throughput and multiple disk failure tolerance.
- RAID 60 Striped Dual Parity and Striped— (Not supported for PCH SSD Controller Definition) Data is striped across multiple striped dual parity disk sets to provide high throughput and greater disk failure tolerance.

**Note** Some Cisco UCS servers require a license for certainRAID configuration options. When Cisco UCS Managerassociates a service profile containing this local diskpolicy with a server, Cisco UCS Manager verifies thatthe selected RAID option is properly licensed. If thereare issues, Cisco UCS Manager displays a configurationerror during the service profile association. For RAID license information for a specific Cisco UCSserver, see the Hardware Installation Guide for that server.

# Automatic Disk Selection

When you specify a disk group configuration, and do not specify the local disks in it, Cisco UCS Manager determines the disks to be used based on the criteria specified in the disk group configuration policy. Cisco UCS Manager can make this selection of disks in multiple ways.

When all qualifiers match for a set of disks, then disks are selected sequentially according to their slot number. Regular disks and dedicated hot spares are selected by using the lowest numbered slot.

The following is the disk selection process:

1. Iterate over all local LUNs that require the creation of a new virtual drive. Iteration is based on the following criteria, in order:
   a. Disk type
   b. Minimum disk size from highest to lowest
   c. Space required from highest to lowest
   d. Disk group qualifier name, in alphabetical order
   e. Local LUN name, in alphabetical order

2. Select regular disks depending on the minimum number of disks and minimum disk size. Disks are selected sequentially starting from the lowest numbered disk slot that satisfies the search criteria.

**Note** If you specify **Any** as the type of drive, the first available drive is selected. After this drive is selected, subsequent drives will be of a compatible type. For example, if the first drive was SATA, all subsequent drives would be SATA. Cisco UCS Manager Release 2.5 supports only SATA and SAS.

Cisco UCS Manager Release 2.5 does not support RAID migration.

3. Select dedicated hot spares by using the same method as normal disks. Disks are only selected if they are in an **Unconfigured Good** state.
4. If a provisioned LUN has the same disk group policy as a deployed virtual drive, then try to deploy the new virtual drive in the same disk group. Otherwise, try to find new disks for deployment.

# Supported LUN Modifications

Some modifications that are made to the LUN configuration when LUNs are already deployed on an associated server are supported.

The following are the types of modifications that can be performed:

- Creation of a new virtual drive.
- Deletion of an existing virtual drive, which is in the orphaned state.
- Non-disruptive changes to an existing virtual drive. These changes can be made on an existing virtual drive without loss of data, and without performance degradation:
  - Policy changes. For example, changing the write cache policy.
  - Modification of boot parameters

The removal of a LUN will cause a warning to be displayed. Ensure that you take action to avoid loss of data.

# Unsupported LUN Modifications

Some modifications to existing LUNs are not possible without destroying the original virtual drive and creating a new one. All data is lost in these types of modification, and these modifications are not supported.

Disruptive modifications to an existing virtual drive are not supported. The following are unsupported disruptive changes:

- Any supported RAID level change that can be handled through reconstruction. For example, RAID0 to RAID1.
- Increasing the size of a virtual drive through reconstruction.
- Addition and removal of disks through reconstruction.

Destructive modifications are also not supported. The following are unsupported destructive modifications:

- RAID-level changes that do not support reconstruction. For example, RAID5 to RAID1.
- Shrinking the size of a virtual drive.
- RAID-level changes that support reconstruction, but where there are other virtual drives present on the same drive group.
- Disk removal when there is not enough space left on the disk group to accommodate the virtual drive.
- Explicit change in the set of disks used by the virtual drive.

# Disk Insertion Handling

When the following sequence of events takes place:

1. The LUN is created in one of the following ways:
   a. You specify the slot specifically by using a local disk reference
   b. The system selects the slot based on criteria specified by you
2. The LUN is successfully deployed, which means that a virtual drive is created, which uses the slot.
3. You remove a disk from the slot, possibly because the disk failed.
4. You insert a new working disk into the same slot.

The following scenarios are possible:

- Non-Redundant Virtual Drives, on page 143
- Redundant Virtual Drives with No Hot Spare Drives, on page 143
- Redundant Virtual Drives with Hot Spare Drives, on page 144
- Replacing Hot Spare Drives, on page 144
- Inserting Physical Drives into Unused Slots, on page 144

## Non-Redundant Virtual Drives

For non-redundant virtual drives (RAID 0), when a physical drive is removed, the state of the virtual drive is **Inoperable**. When a new working drive is inserted, the new physical drive goes to an **Unconfigured Good** state.

For non-redundant virtual drives, there is no way to recover the virtual drive. You must delete the virtual drive and re-create it.

## Redundant Virtual Drives with No Hot Spare Drives

For redundant virtual drives (RAID 1, RAID 5, RAID 6, RAID 10, RAID 50, RAID 60) with no hot spare drives assigned, virtual drive mismatch, virtual drive member missing, and local disk missing faults appear until you insert a working physical drive into the same slot from which the old physical drive was removed.

If the physical drive size is greater than or equal to that of the old drive, the storage controller automatically uses the new drive for the virtual drive. The new drive goes into the **Rebuilding** state. After rebuild is complete, the virtual drive goes back into the **Online** state.

## Redundant Virtual Drives with Hot Spare Drives

For redundant virtual drives (RAID 1, RAID 5, RAID 6, RAID 10, RAID 50, RAID 60) with hot spare drives assigned, when a drive fails, or when you remove a drive, the dedicated hot spare drive, if available, goes into the **Rebuilding** state with the virtual drive in the **Degraded** state. After rebuilding is complete, that drive goes to the **Online** state.

Cisco UCSM raises a disk missing and virtual drive mismatch fault because although the virtual drive is operational, it does not match the physical configuration that Cisco UCSM expects.

if you insert a new disk in the slot with the disk missing, automatic copy back starts from the earlier hot spare disk to the newly inserted disk. After copy back, the hot spare disk is restored. In this state all faults are cleared.

If automatic copy back does not start, and the newly inserted disk remains in the **Unconfigured Good**, **JBOD**, or **Foreign Configuration** state, remove the new disk from the slot, reinsert the earlier hot spare disk into the slot, and import foreign configuration. This initiates the rebuilding process and the drive state becomes **Online**. Now, insert the new disk in the hot spare slot and mark it as hot spare to match it exactly with the information available in Cisco UCSM.

## Replacing Hot Spare Drives

If a hot spare drive is replaced, the new hot spare drive will go to the **Unconfigured Good**, **Unconfigured Bad**, **JBOD**, or **Foreign Configuration** state.

Cisco UCSM will raise a virtual drive mismatch or virtual drive member mismatch fault because the hot spare drive is in a state different from the state configured in Cisco UCSM.

You must manually clear the fault. To do this, you must perform the following actions:

1. Clear the state on the newly inserted drive to **Unconfigured Good**.
2. Configure the newly inserted drive as a hot spare drive to match what is expected by Cisco UCSM.

## Inserting Physical Drives into Unused Slots

If you insert new physical drives into unused slots, neither the storage controller nor Cisco UCSM will make use of the new drive even if the drive is in the **Unconfigured Good** state and there are virtual drives that are missing good physical drives.

The drive will simply go into the **Unconfigured Good** state. To make use of the new drive, you will need to modify or create LUNs to reference the newly inserted drive.

# Virtual Drive Naming

When you use UCSM to create a virtual drive, UCSM assigns a unique ID that can be used to reliably identify the virtual drive for further operations. UCSM also provides the flexibility to provide a name to the virtual

drive at the time of service profile association. Any virtual drive without a service profile or a server reference is marked as an orphan virtual drive.

In addition to a unique ID, a name is assigned to the drive. Names can be assigned in two ways:

- When configuring a virtual drive, you can explicitly assign a name that can be referenced in storage profiles.
- If you have not preprovisioned a name for the virtual drive, UCSM generates a unique name for the virtual drive.

You can rename an orphan virtual drive drives on a blade or a rack server that are not referenced by any service profile or server.

**Note** The renaming an orphan virtual drive is not supported on the Cisco boot optimized M.2 Raid controller (UCS-M2-HWRAID).

# LUN Dereferencing

A LUN is dereferenced when it is no longer used by any service profile. This can occur as part of the following scenarios:

- The LUN is no longer referenced from the storage profile
- The storage profile is no longer referenced from the service profile
- The server is disassociated from the service profile
- The server is decommissioned

When the LUN is no longer referenced, but the server is still associated, re-association occurs.

When the service profile that contained the LUN is disassociated, the LUN state is changed to **Not in use**.

When the service profile that contained the LUN is deleted, the LUN state is changed to **Orphaned**.

# Controller Limits

| Servers/Storage Controllers | Maximum Virtual Drives |
|---|---|
| UCS-M2-HWRAID2 | 1 |
| UCSC-M2RM-M8, UCSC-M2RR-240M8 | 1 |
| UCSC-C245-M8, UCSC-C225-M8, UCSC-C240-M8, and UCSC-C220-M8 | 32 |
| UCSC-C240-M7, UCSC-C220-M7 | 32 |
| UCSB-MRAID12G-M6 | 16 |

| Servers/Storage Controllers | Maximum Virtual Drives |
|---|---|
| UCSC-C220-M6, UCSC-C240-M6, UCSC-C225-M6, UCSC-C245-M6 | 32 |
| UCSC-C240-M5, UCSC-C480-M5 | 32 |
| UCS-S3260-M5 | 64 |
| UCSB-MRAID12G | 16 |
| UCS-M2-HWRAID | 2 |
| For all other servers. | 18 |

**Note**

- UCS-M2-HWRAID2 is supported only on Cisco UCS C220 M8 and Cisco UCS C240 M8 servers.
- Storage controllers support the check max feature.
- When servers with multiple storage controllers are managed by the same storage profile, the maximum virtual drives are limited to the maximum value supported by the server.
- UCS-MSTOR-M2 and UCS-MSTOR-SD controllers are not supported on M6 servers.

# Servers and Storage Support

*Table 12: Cisco UCS C240 M8 Server*

| C-Series Servers | Drives/Storage Controllers |
|---|---|
| Cisco UCS C240 M8 Server | |

| C-Series Servers | Drives/Storage Controllers |
|---|---|
| | • UCSC-C240-M8SX<br>• Up to 28 front-facing and 4 rear facing SFF SAS/SATA HDDs or SAS/SATA SSDs or NVMe SSDs<br>• Up to 8 of the front slots can be direct-attach NVMe<br>• Up to 4 rear drives (SAS/SATA/NVME), the rear slots can be moved to direct-attach<br>• 24G Tri-Mode M1 RAID Controller 32 Drvies (UCSC-RAID-MP1L32)<br>• Cisco 24G Trimode M1 HBA Controller 16 Drives (UCSC-HBA-M1L16)<br>• Cisco 24G Tri-Mode M1 HBA EXJBOD (UCSC-HBA-M1EXJBOD). For supported configurations, see Hardware Compatibility List (HCL).<br>• Cisco 6Gb/s M.2 SATA RAID Controller (UCS-M2-HWRAID2)<br>• Rear Hot-plug M.2 module (Riser 3) (UCSC-M2RR-240M8)<br>• Rear Hot-plug M.2 module (MLOM) (UCSC-M2RM-M8)<br>• UCSC-C240-M8L<br>• Up to 4 rear drives (SAS/SATA/NVME) support is available. The rear slots can be moved to direct-attach<br>• Cisco 24G Trimode M1 RAID Controller with 8GB FBWC LFF 32D (UCSC-RAID-MP1LL32)<br>• Cisco 24G Trimode M1 HBA LFF Controller (32 Drives) (UCSC-HBAMP1LL32)<br>• Cisco 24G Tri-Mode M1 HBA EXJBOD (UCSC-HBA-M1EXJBOD). For supported configurations, see Hardware Compatibility List (HCL).<br>• Cisco 6Gb/s M.2 SATA RAID Controller (UCS-M2-HWRAID2)<br>• Rear Hot-plug M.2 module (Riser 3) |

| C-Series Servers | Drives/Storage Controllers |
| --- | --- |
| | (UCSC-M2RR-240M8)<br>• Rear Hot-plug M.2 module (MLOM) (UCSC-M2RM-M8)<br>• UCSC-C240-M8E3S<br>• Supports up to 32 E3.S drives.<br>• Supports the following M.2 controllers:<br>• UCS-M2-HWRAID2<br>• UCSC-M2RR-240M8<br>• UCSC-M2RM-M8<br>• Up to 4 rear E3.S drive support is available. The rear slots can be moved to direct-attach. |

*Table 13: Cisco UCS C220 M8 Server*

| Servers | Drives/Storage Controllers |
|---|---|
| Cisco UCS C220 M8 Server | • UCSC-C220-M8S<br>• Up to 10 drives with a mix of SAS/SATA/NVMe U.3<br>• Upto 8 direct-attach support for slots 1-4 and 6-9 NVMe SSDs. No rear slot support.<br>• Cisco 24 Gbps Tri-Mode M1 RAID 4GB Flash Backed Write Cache (FBWC) 16 drives (UCSC-RAID-M1L16)<br>• Cisco 24 Gbps Tri-mode Host Bus Adapter (HBA) controller (UCSC-HBA-M1L16)<br>• Cisco 24G Tri-Mode M1 HBA EXJBOD (UCSC-HBA-M1EXJBOD). For supported configurations, see Hardware Compatibility List (HCL).<br>• Cisco 6Gb/s M.2 SATA RAID Controller (UCS-M2-HWRAID2)<br>• Rear Hot-plug M.2 module (MLOM) (UCSC-M2RM-M8)<br>• UCSC-C220-M8E3S<br>• Supports up to 16 E3.S drives<br>• Supports the following M.2 controllers:<br>• UCS-M2-HWRAID2<br>• UCSC-M2RM-M8 |

*Table 14: Cisco UCS C225 M8 Server*

| Servers | Drives/Storage Controllers |
|---|---|
| Cisco UCS C225 M8 Server | • UCSC-C225-M8S<br>• Up to 10 SAS/SATA or NVMe disk drives<br>• Cisco 24 Gbps Tri-mode RAID controller that supports SAS4 or NVMe hardware RAID (UCSC-RAID-HP)<br>• Cisco 24 Gbps Tri-Mode M1 RAID 4GB Flash Backed Write Cache (FBWC) 16 drives (UCSC-RAID-M1L16)<br>• Cisco 24 Gbps Tri-mode Host Bus Adapter (HBA) controller (UCSC-HBA-M1L16)<br>• Up to 4 direct-attach NVMe SSDs<br>• UCSC-C225-M8N<br>• Up to 10 direct attach NVMe SSDs<br>• All 10 NVMe drives connected at PCIe Gen4 x4 |

*Table 15: Cisco UCS C245 M8 Server*

| Servers | Drives/Storage Controllers |
|---|---|
| Cisco UCS C245 M8 Server | • UCSC-C245-M8SX<br>• Cisco 24 Gbps Tri-mode RAID controller that supports SAS4 or NVMe hardware RAID (UCSC-RAID-HP)<br>• Cisco 24 Gbps Tri-mode RAID controller with 4GB Flash Backed Write Cache (FBWC) 32 drives (UCSC-RAID-MP1L32)<br>• Cisco 24G Tri-mode M1 (16-port) HBA controllers (UCSC-HBA-M1L16)<br>• 24 front facing SFF SAS/SATA HDDs or SAS/SATA SSDs or NVMe SSDs<br>• Optionally, up to four of the slots can be direct-attach NVMe. These drives must be placed in front drive bays 1, 2, 3, and 4 only. The rest of the bays (5 - 24) can be populated with SAS/SATA/NVMe SSDs or HDDs.<br>• Optionally, up to four direct-attach SFF rear-facing SAS/SATA/NVMe drives<br>• A mini-storage module connector on the motherboard supports a boot-optimized RAID controller carrier that holds up to two SATA M.2 SSDs. Mixing different capacity SATA M.2 SSDs is not supported.<br>• 8GB FlexMMC utility storage for staging of firmware and other user data. 8GB FlexMMC storage is built into the motherboard |

*Table 16: Cisco UCS C240 M7 Server*

| Server | Drives/Storage Controllers |
|---|---|
| Cisco UCS C240 M7 Server | **Drives**<br>All RAID controllers are only supported on UCSC-C240-M7SX. For UCSC-C240-M7SN, drives are controlled directly from the CPU.<br>• **UCSC-C240-M7SX**<br>  • Up to 24 front facing SFF SAS/SATA HDDs or SAS/SATA SSDs or NVMe SSDs<br>  • Optionally, up to four of the slots can be direct-attach NVMe.<br>• **UCSC-C240-M7SN**<br>  • Up to 24 front NVMe drives (only).<br>  • Optionally, up to 4 rear NVMe drives (only)<br>  • Two CPUs are required when choosing NVMe SSDs<br>• **Controllers**<br>  • UCSC-RAID-MP1L32 — 24G Tri-Mode MP1 RAID Controller<br>  • UCSC-RAID-HP — Cisco Tri-Mode 24G SAS RAID Controller<br>  • UCSC-RAID-SD-D — Cisco 12G SAS RAID Controller<br>  • UCSC-HBA-M1L16 — 24G Tri-Mode M1 HBA for 16 Drives (SAS/SATA/U.3 NVMe)<br>  • UCSC-SAS-T-D — Cisco M6 12G SAS HBA for (16 Drives) |

*Table 17: Cisco UCS C220 M7 Server*

| Server | Drives/Storage Controllers |
|---|---|
| Cisco UCS C220 M7 Server | **Drives**<br>• **UCSC-C220-M7S** - Up to 10 SFF SAS/SATA hard drives (HDDs) or SAS/SATA/NVMe solid state drives (SSDs).<br>• **UCSC-C220-M7N**Up to 10 2.5-inch direct-attach NVMe SSDs<br>• **Controllers**<br>  • UCSC-RAID-M1L16 — 24G Tri-Mode M1 RAID Controller<br>  • UCSC-RAID-HP — Cisco Tri-Mode 24G SAS RAID Controller<br>  • UCSC-RAID-T-D — Cisco M6 12G SAS RAID Controller<br>  • UCSC-HBA-M1L16 — 24G Tri-Mode M1 HBA for 16 Drives (SAS/SATA/U.3 NVMe)<br>  • UCSC-SAS-T-D — Cisco M6 12G SAS HBA for (16 Drives) |

*Table 18: Cisco UCS C225 M6 Server*

| Servers | Drives/Storage Controllers |
|---|---|
| Cisco UCS C225 M6 Server | • UCS C225 M6SX and UCS C245 M6SX in C225-SFF (10 front SAS/SATA drives)<br>• 2 M.2 2280 Drives on UCS-M2-HWRAID<br>• Direct Attached NVMe drives (10 NVMe drives in the front) |

*Table 19: Cisco UCS C245 M6 Server*

| Servers | Drives/Storage Controllers |
|---|---|
| Cisco UCS C245 M6 | • Dual UCS C245 M6SX<br>16 SAS/SATA HDD<br>• UCS C245 M6SX Plus<br>28 SAS/SATA HDD<br>• 2 M.2 2280 Drives on UCS-M2-HWRAID<br>• Directly Attached NVMe on rear risers(up to 4 NVMe SSD) |

**Note** It is recommended not to move drives between SAS3 and SAS4 controllers to avoid data loss.

# Configuring Storage Profiles

## Configuring a Disk Group Policy

You can choose to configure a disk group policy through automatic or manual disk selection. Configuring a disk group involves the following:

1. Setting the RAID Level, on page 156
2. Automatically Configuring Disks in a Disk Group, on page 157 or Manually Configuring Disks in a Disk Group, on page 159

**Note** If you have a setup with the Cisco Boot Optimized M.2 Raid Controller (UCS-M2-HWRAID), then you can configure the disk only manually.

3. Configuring Virtual Drive Properties, on page 161

**Guidelines for sucessful creation of LUNs using JBOD or UG drives:**

1. When the drive state is UG and is in the disk group policy, and if Use JBOD is set to:
   - Yes—Both JBOD and UG drives can be used based on the drive slot ordering.
   - No—Only UG drives can be used.
2. When drive state is JBOD and is in the disk group policy, and if Use JBOD is set to:
   - Yes—Both JBOD and UG drives can be used based on the drive slot ordering.
   - No—Only UG drives can be used.

3. When the drive state is JBOD or UG and is in the disk group policy, and if Use JBOD is set to:
   - Yes—Both JBOD and UG drives can be used.
   - No—Only UG drives can be used.

**Note** The UCS Manager disk selection is based on the sequential slot number, irrespective of the drive state.

## Setting the RAID Level

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | UCS-A# **scope org** *org-name* | Enters the organization mode for the specified organization. To enter the root organization mode, enter / as the *org-name*. |
| **Step 2** | UCS-A /org# **create disk-group-config-policy** *disk-group-name* | Creates a disk group configuration policy with the specified name and enters disk group configuration policy mode. |
| **Step 3** | UCS-A /org/disk-group-config-policy* # **set raid-level** *raid-level* | Specifies the RAID level for the disk group configuration policy. The RAID levels that you can specify are:<br>• raid-0-striped<br>• raid-1-mirrored<br>**Note** The Cisco boot optimized M.2 RAID controller (UCS-M2-HWRAID) supports only RAID1.<br>**Note** The Cisco boot optimized M.2 NVMe RAID controller (UCS-M2-NVRAID) supports only RAID 0 and RAID 1. |
| **Step 4** | UCS-A /org/disk-group-config-policy* # **commit-buffer** | Commits the transaction to the system configuration. |

### Example

This example shows how to set the RAID level for a disk group configuration policy.

```
UCS-A# scope org
UCS-A /org # create disk-group-config-policy raid5policy
```

```
UCS-A /org/disk-group-config-policy* # set raid-level raid-5-striped-parity
UCS-A /org/disk-group-config-policy* # commit-buffer
```

### What to do next

Automatically or manually configure disks as part of the disk group configuration policy.

## Automatically Configuring Disks in a Disk Group

You can allow UCSM to automatically select and configure disks in a disk group.

When you create a disk group with RAID 1 policy and configure four disks for it, a RAID1E configuration is created internally by the storage controller.

When you create a disk group with RAID 0, RAID 1 policy and configure four disks for it, a RAID1E configuration is created internally by the storage controller.

If you have a set-up with the Cisco Boot Optimized M.2 Raid Controller (UCS-M2-HWRAID) and Cisco Boot Optimized M.2 Raid Controller (UCS-M2-NVRAID), then go to Manually Configuring Disks in a Disk Group, on page 159.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | UCS-A# **scope org** *org-name* | Enters the organization mode for the specified organization. To enter the root organization mode, enter / as the *org-name*. |
| **Step 2** | UCS-A /org# **enter disk-group-config-policy** *disk-group-name* | Enters disk group configuration policy mode for the specified disk group name. |
| **Step 3** | UCS-A /org/disk-group-config-policy* # **enter disk-group-qual** | Enters disk group qualification mode. In this mode, UCSM automatically configures disks as part of the specified disk group. |
| **Step 4** | UCS-A /org/disk-group-config-policy/disk-group-qual* # **set drive-type** *drive-type* | Specifies the drive type for the disk group. You can select:<br>• HDD<br>• SSD<br>• Unspecified<br>**Note**<br>If you specify **Unspecified** as the type of drive, the first available drive is selected. After this drive is selected, subsequent drives will be of a compatible type. For example, if the first was SSD, all subsequent drives would be SSD. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 5** | UCS-A /org/disk-group-config-policy/disk-group-qual* # **set min-drive-size** *drive-size* | Specifies the minimum drive size for the disk group. Only disks that match this criteria will be available for selection.<br><br>The range for minimum drive size is from 0 to10240 GB. You can also set the minimum drive size as **Unspecified**. If you set the minimum drive size as **Unspecified**, drives of all sizes will be available for selection. |
| **Step 6** | UCS-A /org/disk-group-config-policy/disk-group-qual* # **set num-ded-hot-spares** *hot-spare-num* | Specifies the number of dedicated hot spares for the disk group.<br><br>The range for dedicated hot spares is from 0 to 24 hot spares. You can also set the number of dedicated hot spares as **Unspecified**. If you set the number of dedicated hot spares as **Unspecified**, the hot spares will be selected according to the disk selection process. |
| **Step 7** | UCS-A /org/disk-group-config-policy/disk-group-qual* # **set num-drives** *drive-num* | Specifies the number of drives for the disk group.<br><br>The range for drives is from 0 to 24 drives for Cisco UCS C240, C220, C24, and C22 servers. For all other servers, the limit is 16 drives per server.. You can also set the number of drives as **Unspecified**. If you set the number of drives as **Unspecified**, the number of drives will be selected according to the disk selection process. |
| **Step 8** | UCS-A /org/disk-group-config-policy/disk-group-qual* # **set num-glob-hot-spares** *hot-spare-num* | Specifies the number of global hot spares for the disk group.<br><br>The range for global hot spares is from 0 to 24 hot spares. You can also set the number of global hot spares as **Unspecified**. If you set the number of global hot spares as **Unspecified**, the global hot spares will be selected according to the disk selection process. |
| **Step 9** | UCS-A /org/disk-group-config-policy/disk-group-qual* # **set use-remaining-disks** {**no** \| **yes**} | Specifies whether the remaining disks in the disk group policy should be used or not.<br><br>The default value for this command is **no**. |
| **Step 10** | UCS-A /org/disk-group-config-policy/disk-group-qual* # **commit-buffer** | Commits the transaction to the system configuration. |

### Example

This example shows how to automatically configure disks for a disk group configuration policy.

```
UCS-A# scope org
UCS-A /org # enter disk-group-config-policy raid5policy
UCS-A /org/disk-group-config-policy* # enter disk-group-qual
UCS-A /org/disk-group-config-policy/disk-group-qual* # set drive-type hdd
UCS-A /org/disk-group-config-policy/disk-group-qual* # set min-drive-size 1000
UCS-A /org/disk-group-config-policy/disk-group-qual* # set num-ded-hot-spares 2
UCS-A /org/disk-group-config-policy/disk-group-qual* # set num-drives 7
UCS-A /org/disk-group-config-policy/disk-group-qual* # set num-glob-hot-spares 2
UCS-A /org/disk-group-config-policy/disk-group-qual* # set use-remaining-disks no
UCS-A /org/disk-group-config-policy/disk-group-qual* # commit-buffer
```

```
UCS-A# scope org
UCS-A /org # enter disk-group-config-policy raid5policy
UCS-A /org/disk-group-config-policy* # enter disk-group-qual
UCS-A /org/disk-group-config-policy/disk-group-qual* # set drive-type ssd
UCS-A /org/disk-group-config-policy/disk-group-qual* # set min-drive-size 1000
UCS-A /org/disk-group-config-policy/disk-group-qual* # set num-ded-hot-spares 2
UCS-A /org/disk-group-config-policy/disk-group-qual* # set num-drives 7
UCS-A /org/disk-group-config-policy/disk-group-qual* # commit-buffer
```

### What to do next

Configure Virtual Drives.

## Manually Configuring Disks in a Disk Group

You can manually configure disks for a disk group.

When you create a disk group with RAID 1 policy and configure four disks for it, a RAID 1E configuration is created internally by the storage controller.

Cisco boot optimized M.2 RAID controller (UCS-M2-HWRAID) supports only RAID1.

Cisco boot optimized M.2 NVMe RAID controller (UCS-M2-NVRAID) supports RAID 0 and RAID 1.

When you create a disk group with RAID 0, RAID 1 policy and configure four disks for it, a RAID1E configuration is created internally by the storage controller.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | UCS-A# **scope org** *org-name* | Enters the organization mode for the specified organization. To enter the root organization mode, enter **/** as the *org-name*. |
| **Step 2** | UCS-A /org# **enter disk-group-config-policy** *disk-group-name* | Enters disk group configuration policy mode for the specified disk group name. |
| **Step 3** | UCS-A /org/disk-group-config-policy* # **create local-disk-config-ref** *slot-num* | Creates a local disk configuration reference for the specified slot and enters local disk configuration reference mode. |

| | Command or Action | Purpose |
|---|---|---|
| | | **Note**<br>M.2 drives typically have Slot IDs = 253, 254.<br><br>**Note**<br>M.2 drives for M.2 NVMe RAID controller (UCS-M2-NVRAID) typically have Slot IDs = 251, 252. |
| **Step 4** | UCS-A<br>/org/disk-group-config-policy/local-disk-config-ref *# **set role** *role* | Specifies the role of the local disk in the disk group. You can select:<br>• ded-hot-spare: Dedicated hot spare<br>• glob-hot-spare: Global hot spare<br>• normal<br><br>**Note**<br>If you have a setup with the Cisco Boot Optimized M.2 Raid Controller (UCS-M2-HWRAID), then select normal. Selecting any other value results in configuration error. |
| **Step 5** | UCS-A<br>/org/disk-group-config-policy/local-disk-config-ref *# **set span-id** *span-id* | Specifies the ID of the span group to which the disk belongs. Disks belonging to a single span group can be treated as a single disk with a larger size. The values range from 0 to 8. For RAID-10, RAID-50, and RAID-60, minimum 2 spans are required and maximum 8 spans are supported. You can also set the Span ID as **Unspecified** when spanning information is not required.<br><br>**Note**<br>• In Cisco UCS Release 2.5, you can have a maximum of 4 span groups.<br>• If you have a setup with the Cisco Boot Optimized M.2 Raid Controller (UCS-M2-HWRAID), then this field does not apply. Select the Span ID field as **Unspecified**. Selecting any value results in configuration error. |
| **Step 6** | UCS-A<br>/org/disk-group-config-policy/local-disk-config-ref *# **commit-buffer** | Commits the transaction to the system configuration. |

### Example

This example shows how to manually configure disks for a disk group configuration policy.

```
UCS-A# scope org
UCS-A /org # enter disk-group-config-policy raid5policy
UCS-A /org/disk-group-config-policy* # create local-disk-config-ref 1
UCS-A /org/disk-group-config-policy/local-disk-config-ref *# set role ded-hot-spare
UCS-A /org/disk-group-config-policy/local-disk-config-ref* # set span-id 1
UCS-A /org/disk-group-config-policy/local-disk-config-ref *# commit-buffer
```

### What to do next

Configure Virtual Drive Properties.

## Configuring Virtual Drive Properties

All virtual drives in a disk group must be managed by using a single disk group policy.

If you try to associate to a server that does not support these properties, a configuration error will be generated.

Only the following storage controllers support these properties:

- LSI 6G MegaRAID SAS 9266-8i
- LSI 6G MegaRAID SAS 9271-8i
- LSI 6G MegaRAID 9265-8i
- LSI MegaRAID SAS 2208 ROMB
- LSI MegaRAID SAS 9361-8i

For the LSI MegaRAID SAS 2208 ROMB controller, these properties are supported only in the B420-M3 blade server. For the other controllers, these properties are supported in multiple rack servers.

**Note** If you have a setup with the Cisco Boot Optimized M.2 Raid Controller (UCS-M2-HWRAID) or Cisco boot optimized M.2 NVMe RAID controller (UCS-M2-NVRAID), then:

- You can create only one virtual drive
- For **strip-size**, select **64KB** or **32KB**. Selecting any other value results in configuration error.

  For Cisco boot optimized M.2 NVMe RAID controller (UCS-M2-NVRAID), select **64KB** or platform default as the **Strip-Size**. Selecting any other value results in configuration error.
- For **access-policy**, **read-policy**, **write-cache-policy**, **io-policy**, and **drive-cache**, select **platform-default**. Selecting any other value results in configuration error.
- The **Access Policy** for the Raid controller (UCSC-RAID-HP) supports only the **Read Write** option.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | UCS-A# **scope org** *org-name* | Enters the organization mode for the specified organization. To enter the root organization mode, enter / as the *org-name*. |
| **Step 2** | UCS-A /org# **scope disk-group-config-policy** *disk-group-name* | Enters disk group configuration policy mode for the specified disk group name. |
| **Step 3** | UCS-A /org/disk-group-config-policy* # **create virtual-drive-def** | Creates a virtual drive definition and enters the virtual drive definition mode. |
| **Step 4** | UCS-A /org/disk-group-config-policy/virtual-drive-def* # **set access-policy** *policy-type* | Specifies the access policy. This can be one of the following:<br>• blocked<br>• platform-default<br>• read-only:<br>• read-write |
| **Step 5** | UCS-A /org/disk-group-config-policy/virtual-drive-def* # **set drive-cache** *state* | Specifies the state of the drive cache. This can be one of the following:<br>• enable<br>• disable<br>• no-change<br>• platform-default<br>**Important**<br>In Cisco UCS Release 2.5, the drive cache state cannot be changed. It will remain as **platform-default**, irrespective of the drive cache state that you select. |
| **Step 6** | UCS-A /org/disk-group-config-policy/virtual-drive-def* # **set io-policy** *policy-type* | Specifies the I/O policy. This can be one of the following:<br>• cached<br>• direct<br>• platform-default |
| **Step 7** | UCS-A /org/disk-group-config-policy/virtual-drive-def* # **set read-policy** *policy-type* | Specifies the read policy. This can be one of the following:<br>• normal |

| | Command or Action | Purpose |
|---|---|---|
| | | • platform-default<br>• read-ahead |
| **Step 8** | UCS-A /org/disk-group-config-policy/virtual-drive-def* # **set strip-size** *strip-size* | Specifies the strip size. This can be one of the following:<br>• 64 KB<br>• 128 KB<br>• 256 KB<br>• 512 KB<br>• 1024 KB<br>• platform-default |
| **Step 9** | UCS-A /org/disk-group-config-policy/virtual-drive-def* # **set write-cache-policy** *policy-type* | Specifies the write-cache-policy. This can be one of the following:<br>• always-write-back<br>• platform-default<br>• write-back-good-bbu<br>• write-through |
| **Step 10** | UCS-A /org/disk-group-config-policy/virtual-drive-def* # **commit-buffer** | Commits the transaction to the system configuration. |
| **Step 11** | UCS-A /org/disk-group-config-policy/virtual-drive-def* # **show** | Displays the configured virtual drive properties. |

### Example

This example shows how to configure virtual disk properties:

```
UCS-A# scope org
UCS-A /org # scope disk-group-config-policy raid0policy
UCS-A /org/disk-group-config-policy # create virtual-drive-def
UCS-A /org/disk-group-config-policy/virtual-drive-def* # set access-policy read-write
UCS-A /org/disk-group-config-policy/virtual-drive-def* # set drive-cache enable
UCS-A /org/disk-group-config-policy/virtual-drive-def* # set io-policy cached
UCS-A /org/disk-group-config-policy/virtual-drive-def* # set read-policy normal
UCS-A /org/disk-group-config-policy/virtual-drive-def* # set strip-size 1024
UCS-A /org/disk-group-config-policy/virtual-drive-def* # set write-cache-policy write-through
UCS-A /org/disk-group-config-policy/virtual-drive-def* # commit-buffer
UCS-A /org/disk-group-config-policy/virtual-drive-def # show

Virtual Drive Def:
    Strip Size (KB): 1024KB
```

```
    Access Policy: Read Write
    Read Policy: Normal
    Configured Write Cache Policy: Write Through
    IO Policy: Cached
    Drive Cache: Enable
UCS-A /org/disk-group-config-policy/virtual-drive-def #
```

### What to do next

Create a Storage Profile

# Creating a Storage Profile

You can create a storage profile at the org level and at the service-profile level.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | UCS-A# **scope org** *org-name* | Enters the organization mode for the specified organization. To enter the root organization mode, enter **/** as the *org-name*. |
| **Step 2** | UCS-A /org # **create storage-profile** *storage-profile-name* | Creates a storage profile with the specified name at the org level and enters storage-profile configuration mode. |
| **Step 3** | UCS-A /org/storage-profile* # **commit-buffer** | Commits the transaction to the system configuration. |
| **Step 4** | (Optional) UCS-A /org* # **enter service-profile** *service-profile-name* | Enters the specified service profile. |
| **Step 5** | (Optional) UCS-A /org/service-profile* # **create storage-profile-def** | Creates a storage profile at the service-profile level. |
| **Step 6** | UCS-A /org/service-profile/storage-profile-def* # **commit-buffer** | Commits the transaction to the system configuration. |

### Example

This example shows how to create a storage profile at the org level.

```
UCS-A# scope org
UCS-A /org # create storage-profile stp2
UCS-A /org/storage-profile* # commit-buffer
```

This example shows how to create a storage profile at the service-profile level.

```
UCS-A# scope org
UCS-A /org* # enter service-profile sp1
UCS-A /org/service-profile* # create storage-profile-def
UCS-A /org/service-profile/storage-profile-def* # commit-buffer
```

### What to do next

Create Local LUNs

# Deleting a Storage Profile

You can delete a storage profile that was created at the org level or at the service-profile level.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | UCS-A# **scope org** *org-name* | Enters the organization mode for the specified organization. To enter the root organization mode, enter **/** as the *org-name*. |
| **Step 2** | UCS-A /org # **delete storage-profile** *storage-profile-name* | Deletes the storage profile with the specified name at the org level. |
| **Step 3** | (Optional) UCS-A /org # **scope service-profile** *service-profile-name* | Enters the specified service profile. |
| **Step 4** | (Optional) UCS-A /org/service-profile # **delete storage-profile-def** | Deletes the dedicated storage profile at the service-profile level. |

### Example

This example shows how to delete a storage profile at the org level.

```
UCS-A # scope org
UCS-A /org #  delete storage-profile stor1
```

This example shows how to delete a storage profile at the service-profile level.

```
UCS-A # scope org
UCS-A /org # scope service-profile sp1
UCS-A /org/service-profile # delete storage-profile-def
```

# Local LUNs

## Creating Local LUNs

You can create local LUNs within a storage profile at the org level and within a dedicated storage profile at the service-profile level.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | UCS-A# **scope org** *org-name* | Enters the organization mode for the specified organization. To enter the root organization mode, enter **/** as the *org-name*. |
| **Step 2** | UCS-A /org # **enter storage-profile** *storage-profile-name* | Enters storage-profile mode for the specified storage profile. |
| **Step 3** | UCS-A /org/storage-profile* # **create local-lun** *lun-name* | Creates a local LUN with the specified name. |
| **Step 4** | UCS-A /org/storage-profile/local-lun* # **set auto-deploy** {**auto-deploy** \| **no-auto-deploy**} | Specifies whether the LUN should be auto-deployed or not. |
| **Step 5** | UCS-A /org/storage-profile/local-lun* # **set disk-policy-name** *disk-policy-name* | Specifies the name of the disk policy name for this LUN. |
| **Step 6** | UCS-A /org/storage-profile/local-lun* # **set size** *size* | Specifies the size of this LUN in GB.<br>**Note**<br>In a setup with the Cisco boot optimized M.2 Raid controller, you do not have to specify a size. The system uses the full disk capacity to create the LUN, irrespective of the size specified.<br>**Note**<br>You do not need to specify a LUN size while claiming an orphaned LUN. |
| **Step 7** | UCS-A/org/storage-profile/local-lun*#/ **set fractional size 100*** | Specifies the fractional size of the LUN. This command allows you to set a fractional size, which can be useful for specific storage configurations where fractional allocation is required. |
| **Step 8** | UCS-A /org/storage-profile/local-lun* # **set expand-to-avail** {**no** \| **yes**} | Specifies whether the LUN should be expanded to the entire available disk group.<br>For each service profile, only one LUN can be configured to use this option. |
| **Step 9** | UCS-A /org/storage-profile/local-lun* # **commit-buffer** | Commits the transaction to the system configuration. |

### Example

This example shows how to configure a local LUN within a storage profile at the org level.

```
UCS-A# scope org
UCS-A /org # enter storage-profile stp2
UCS-A /org/storage-profile* # create local-lun lun2
UCS-A /org/storage-profile/local-lun* # set disk-policy-name dpn2
UCS-A /org/storage-profile/local-lun* # set size 1000
UCS-A /org/storage-profile/local-lun* # set fractional size 100
UCS-A /org/storage-profile/local-lun* # set expand-to-avail yes
UCS-A /org/storage-profile/local-lun* # commit-buffer

CiscoHH-A /org/storage-profile # show local-lun

Local SCSI LUN:
LUN Name         Size (GB)  Fractional Size (MB) Expand To Available Disk Policy Name
Auto Deploy
---------------- ---------- -------------------- ------------------- --------------------
-----------
lun2             1000       100                  Yes                 dpn2
Auto Deploy


UCS-A# scope org
UCS-A /org # enter storage-profile stp2
UCS-A /org/storage-profile* # create local-lun lun2
UCS-A /org/storage-profile/local-lun* # set disk-policy-name dpn2
UCS-A /org/storage-profile/local-lun* # set expand-to-avail yes
UCS-A /org/storage-profile/local-lun* # set size 1000
UCS-A /org/storage-profile/local-lun* # set fractional size 100
UCS-A /org/storage-profile/local-lun* # set expand-to-avail yes
UCS-A /org/storage-profile/local-lun* # commit-buffer
```

This example shows how to configure a local LUN within a dedicated storage profile at the service-profile level.

```
UCS-A# scope org
UCS-A /org* # enter service-profile stp1
UCS-A /org/service-profile* # enter storage-profile-def
UCS-A /org/service-profile/storage-profile-def # create local-lun lun1
UCS-A /org/service-profile/storage-profile-def/local-lun* # set disk-policy-name dpn1

UCS-A /org/service-profile/storage-profile-def/local-lun* # set size 1000
UCS-A /org/service-profile/storage-profile-def/local-lun* # set fractional size 100
UCS-A /org/service-profile/storage-profile-def/local-lun* # set expand-to-avail yes
UCS-A /org/service-profile/storage-profile-def/local-lun* # commit-buffer

UCS-A# scope org
UCS-A /org # enter service-profile sp1
UCS-A /org/service-profile* # enter storage-profile-def
UCS-A /org/service-profile/storage-profile-def # create local-lun lun1
UCS-A /org/service-profile/storage-profile-def/local-lun* # set auto-deploy no-auto-deploy
UCS-A /org/service-profile/storage-profile-def/local-lun* # set disk-policy-name dpn1
UCS-A /org/service-profile/storage-profile-def/local-lun* # set size 1000
UCS-A /org/service-profile/storage-profile-def/local-lun* # set fractional size 100
UCS-A /org/service-profile/storage-profile-def/local-lun* # set expand-to-avail yes
UCS-A /org/service-profile/storage-profile-def/local-lun* # commit-buffer
```

### What to do next

Associate a Storage Profile with a Service Profile

# Deleting Local LUNs In a Storage Profile

When a LUN is deleted, the corresponding virtual drive is marked as orphan after the virtual drive reference is removed from the server.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | UCS-A# **scope org** *org-name* | Enters the organization mode for the specified organization. To enter the root organization mode, enter **/** as the *org-name*. |
| **Step 2** | UCS-A /org # **enter storage-profile** *storage-profile-name* | Enters storage-profile mode for the specified storage profile. |
| **Step 3** | (Optional) UCS-A /org/storage-profile* # **show local-lun** | Displays the local LUNs in the specified storage profile. |
| **Step 4** | UCS-A /org/storage-profile* # **delete local-lun** *lun-name* | Deletes the specified LUN. |
| **Step 5** | UCS-A /org/storage-profile* # **commit-buffer** | Commits the transaction to the system configuration. |

### Example

This example shows how to delete a LUN in a storage profile.

```
UCS-A # scope org
UCS-A /org # enter storage-profile stp2
UCS-A /org/storage-profile # show local-lun

Local SCSI LUN:
    LUN Name         Size (GB)  Fractional Size (MB) Expand To Available Disk Policy Name
    Auto Deploy
---------------- ---------- -------------------- ------------------- --------------------
-----------
    lun2             1000       100                  Yes                 dpn2
    No Auto Deploy

UCS-A /org/storage-profile # delete local-lun luna
UCS-A /org/storage-profile* # commit-buffer
UCS-A /org/storage-profile* # show local-lun


Local SCSI LUN:
LUN Name         Size (GB)  Fractional Size (MB) Expand To Available Disk Policy Name
Auto Deploy
---------------- ---------- -------------------- ------------------- --------------------
-----------
lun2             1000       100                  Yes                 dpn2
No Auto Deploy
```

# LUN Set

## LUN Set

Beginning with release 4.0(2a), Cisco UCS Manager provides the ability to configure a range of disk slots into individual RAID0 LUNs using LUN Set option.

The following guidelines should be considered while creating a LUN Set:

- Only SSD and HDD types of disks are allowed.
- Upto 60 disks are allowed in one range.
- You cannot add the same set of disks in range under two different LUN Set configurations.
- If a disk is set in the disk slot range of LUN Set, then you cannot configure the same disk set in Local LUN configuration under the same storage policy. Similarly, if a disk is set in Local LUN configuration, then you cannot use the same disk in the disk slot range of LUN Set.
- The server, in which the LUN Set is configured, should support OOB storage operations.
- You cannot configure a Local Disk Policy along with a Storage Policy in the same Service Profile.
- You cannot have the same name for a Local LUN and LUN Set.
- In S-series server PCH controllers, slots 201 and 202 do not support LUN Set.

### Limitations of LUN Set

Cisco UCS Manager has the following limitations with LUN Set:

- You cannot claim orphaned Local LUNs into a LUN Set.
- Once created, you cannot modify a LUN Set. You should delete and create a new LUN Set with desired parameters.
- OS boot is not supported from LUN Set.

## Creating a LUN Set

You can create a LUN Set within a storage profile at the org level and within a dedicated storage profile at the service-profile level.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | UCS-A# **scope org** *org-name* | Enters the organization mode for the specified organization. To enter the root organization mode, enter / as the *org-name*. |
| **Step 2** | UCS-A /org # **enter storage-profile** *storage-profile-name* | Enters storage-profile mode for the specified storage profile. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 3** | UCS-A /org/storage-profile* # **create lun-set** *lun-set-name* | Creates a LUN Set with the specified name. |
| **Step 4** | UCS-A /org/storage-profile/lun-set* # **set disk-slot-range** *disk-slot-range* | Specifies the slot range for the disk. |
| **Step 5** | UCS-A /org/storage-profile/lun-set* # **create virtual-drive-def** | Enters the virtual drive configuration settings mode. |
| **Step 6** | UCS-A /org/storage-profile/lun-set/virtual-drive-def* # **set access-policy** {**blocked** \| **platform-default** \| **read-only** \| **read-write**} | Specifies the type of access allowed. |
| **Step 7** | UCS-A /org/storage-profile/lun-set/virtual-drive-def* # **set drive-cache** {**disable** \| **enable** \| **no-change** \| **platform-default**} | Specifies the type of drive cache. |
| **Step 8** | UCS-A /org/storage-profile/lun-set/virtual-drive-def* # **set io-policy** {**cached** \| **direct** \| **platform-default**} | Specifies the type of Input/Output Policy. |
| **Step 9** | UCS-A /org/storage-profile/lun-set/virtual-drive-def* # **set read-policy** {**normal** \| **platform-default** \| **read-ahead**} | Specifies the read-ahead cache mode. |
| **Step 10** | UCS-A /org/storage-profile/lun-set/virtual-drive-def* # **set security** {**no** \| **yes**} | Configure this option to secure a virtual drive. |
| **Step 11** | UCS-A /org/storage-profile/lun-set/virtual-drive-def* # **set strip-size** {**1024kb** \| **128kb** \| **16kb** \| **256kb** \| **32kb** \| **512kb** \| **64kb** \| **8kb** \| **platform-default**} | Specifies the portion of the striped data segment that resides on each physical disk. |
| **Step 12** | UCS-A /org/storage-profile/lun-set/virtual-drive-def* # **set write-cache-policy** {**always-write-back** \| **platform-default** \| **write-back-good-bbu** \| **write-through**} | Specifies the type of write policy. |
| **Step 13** | UCS-A /org/storage-profile/lun-set/virtual-drive-def* # **commit-buffer** | Commits the transaction to the system configuration. |

### Example

The following example creates a LUN set and configure the virtual drive settings:

```
UCS-A# scope org
UCS-A/org # enter storage-profile stroageprofile1
UCS-A/org/storage-profile # create lun-set lunset1
UCS-A/org/storage-profile/lun-set* # set disk-slot-range 2
UCS-A/org/storage-profile/lun-set* # create virtual-drive-def
UCS-A/org/storage-profile/lun-set/virtual-drive-def* # set access-policy read-write
UCS-A/org/storage-profile/lun-set/virtual-drive-def* # set drive-cache enable
UCS-A/org/storage-profile/lun-set/virtual-drive-def* # set io-policy direct
UCS-A/org/storage-profile/lun-set/virtual-drive-def* # set read-policy read-ahead
UCS-A/org/storage-profile/lun-set/virtual-drive-def* # set security yes
UCS-A/org/storage-profile/lun-set/virtual-drive-def* # set strip-size 512kb
UCS-A/org/storage-profile/lun-set/virtual-drive-def* # set write-cache-policy platform-default
UCS-A/org/storage-profile/lun-set/virtual-drive-def* # commit-buffer
```

### What to do next

Associate the Storage Profile with a Service Profile

## Deleting a LUN Set

You can delete a LUN Set within a storage profile at the org level and within a dedicated storage profile at the service-profile level.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | UCS-A# **scope org** *org-name* | Enters the organization mode for the specified organization. To enter the root organization mode, enter **/** as the *org-name*. |
| **Step 2** | UCS-A /org # **enter storage-profile** *storage-profile-name* | Enters storage-profile mode for the specified storage profile. |
| **Step 3** | UCS-A /org/storage-profile* # **delete lun-set** *lun-set-name* | Deletes the LUN Set with the specified name. |
| **Step 4** | UCS-A /org/storage-profile* # **commit-buffer** | Commits the transaction to the system configuration. |

### Example

The following example deletes a LUN set:

```
UCS-A# scope org
UCS-A/org # enter storage-profile stroageprofile1
UCS-A/org/storage-profile # delete lun-set lunset1
UCS-A/org/storage-profile* # commit-buffer
```

# Configuring Aero Controllers

## Automatic Configuration Mode for Aero Controllers

### Autoconfiguration Mode for Storage Controllers

Autoconfiguration mode for storage controllers are supported on Cisco UCS C-Series Rack servers (M6, M7, and M8), B-Series Blade servers (M6), and X-Series Compute Nodes (M6, M7, and M8).

C-series M6 servers support PCIe SAS316-port storage controllers for Direct Attached Storage. Controllers support an Autoconfiguration mode in which the state of a newly inserted disk is automatically moved to the Unconfigured-Good state.

Because of this, you can choose whether or not to use Autoconfiguration by creating a Storage Profile and associating it with the server. The default is that the automatic configuration feature is disabled, which retains the drive state when the server is rebooted.

If Autoconfiguration is used, you must select a drive state from one of the following:

- Unconfigured-Good
- JBOD
- RAID0 (RAID0 WriteBack)

This is because the controller firmware changes the behavior of systemPD to EPD-PT. EPD-PT is internally a RAID0 volume without any drive DDF metadata. The controller stores the metadata for identifying it as a RAID0 volume. The EPD-PT drives are considered as JBOD drives so the drive status is reported as JBOD and online.

Controller supports the following models:

- UCSC-RAID-M6T
- UCSC-RAID-M6HD
- UCSC-RAID-M6SD
- UCSX-X10C-RAIDF
- UCSC-RAID-HP

The table below shows the behavior of Autoconfiguration in different scenarios.

| Autoconfig Mode | Reboot/OCR | Hotplug | User Action |
|---|---|---|---|
| Unconfigured-Good (OFF) | • All Unconfigured-Good drives remain Unconfigured-Good.<br>• All previously configured JBOD remain JBOD. | • Inserted drive remains Unconfigured-Good.<br>• JBOD from a different server remains Unconfigured-Good on this controller. | Disabling Autoconfig has no impact on the existing configuration<br>Any JBOD device remains as JBOD across controller boot.<br>Any Unconfigured-Good remains unconfiguredgood across controller boot. |
| JBOD | • All Unconfigured-Good are converted to JBOD. | Newly inserted unconfigured device is converted to JBOD. | All Unconfigured-Good drives (non-user created) on the controller while running Autoconfig is converted to JBOD.<br>User created Unconfigured-Good drive remains Unconfigured-Good until next reboot. During reboot Unconfigured-Good gets converted to JBOD. |
| RAID0 (RAID0 WriteBack) | • All Unconfigured-Good converted to RAID0 WriteBack. | Newly inserted unconfigured device is converted to RAID0 WriteBack. | All Unconfigured-Good drives (non-user created) on the controller while running Autoconfig is converted to RAID0 WriteBack.<br>User created Unconfigured-Good remains Unconfigured-Good across controller reboot.<br>Any RAID0 WriteBack device remains as RAID0 WriteBack across controller reboot. |

Selecting EPD-PT (JBOD) as the default configuration does not retain the Unconfigured-Good state across host reboot. The drive state can be retained by disabling the automatic configuration feature. If the Autoconfig option is used, the default automatic configuration will always mark a drive as Unconfigured-Good.

When Autoconfig is selected, then the drive is configured to the desired drive state, the JBOD and unconfigured drives will set the drive state accordingly on the next controller boot or OCR,

The following table shows sample use cases for different Autoconfig scenarios.

| Use Case Scenario | Autoconfig Option |
|---|---|
| Using the server for JBOD Only (for example: Hyper converged, Hadoop data node etc ) | JBOD |
| Using the server for RAID volume (for example: SAP HANA database) | Unconfigured-Good |
| Using the server for Mixed JBOD and RAID volume | Unconfigured-Good |
| Using the server for per drive RAID0 WriteBack (for example: Hadoop data node) | RAID0 WriteBack |

**Note** When using UCSX-X10C-RAIDF or UCSC-RAID-M6T controllers with Autoconfiguration Mode set to RAID0, drives in *Unconfigured Good* state may not transition to *Online* and RAID0 LUNs will not be created after a storage profile redeploy and server reboot. This is a known behavior and differs from UCSC-RAID-HP controllers, where drives transition to *Online* and LUNs are created as expected.

## Creating an Autoconfiguation Profile

You can include the storage Autoconfiguration (Auto Config) mode option in your storage profile and unconfigure it when no longer needed. Changes will take effect on the next system boot. Auto Config for storage is only available on Cisco UCS M6 servers with Aero controllers.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | UCS-A# **scope org** *org-name* | Enters organization mode for the specified organization. To enter the root organization mode, type / as the *org-name* . |
| **Step 2** | UCS-A/org# **scope storage-profile** *profile-name* | Enters the storage profile for the specified profile. |
| **Step 3** | UCS-A/org/storage-profile# **show detail expand** | Shows a detailed view of the Storage Profile. If Auto Config Mode has not been enabled for this storage profile, or no Aero controller is present, you should not see an entry for Auto Config Mode. If Auto Config is not configured, inserted devices will retain their state on system reboot. |
| **Step 4** | UCS-A/org/storage-profile# **set auto-config-mode** jbod \| raid-0 \| unconfigured-good \| unspecified | Enables Auto Config Mode and sets the disk configuration mode to the desired state. If no further parameters are specified, all inserted devices will be tagged as Unconfigured Good on reboot. Enter unconfigured if you wish to disable Auto Config mode. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 5** | UCS-A/org/storage-profile# **commit-buffer** | Commits the transaction to the system configuration. |

# PCH Controller Definitions

## PCH SSD Controller Definition

Cisco UCS Manager Platform Controller Hub (PCH) Solid State Drive (SSD) Controller Definition provides a local storage configuration in storage profiles where you can configure all the disks in a single RAID or in a JBOD disk array.

The PCH Controller Definition configuration provides the following features:

- Ability to configure a single LUN RAID across two internal SSDs connected to the onboard PCH controller
- A way to configure the controller in two modes: AHCI (JBOD) and SWRAID (RAID).
- Ability to configure the PCH storage device in an Embedded Local LUN and Embedded Local Disk boot policy so precision control for boot order is achieved even with the presence of other bootable local storage devices in the server. Do not use the Local LUN or the Local JBOD options to boot from PCH disks
- Scrub policy support for the internal SSD drives. This is applicable only for the SWRAID mode. This does not apply for the AHCI and NORAID of PCH Controller modes.*See the UCS Manager Server Management Guide.*
- Firmware upgrade support for the internal SSD drives. Disk firmware upgrade is supported only when the PCH Controller is in SWRAID mode. It is not supported for AHCI mode.

You can configure PCH controller SSDs in a storage profile policy. You can enable or disable protect configuration which saves the LUN configuration even after a service profile disassociation. You choose a controller mode. The PCH controller configuration supports only these two RAID options: RAID0 and RAID1. Use No RAID configuration option for AHCI mode where all the disks connected to the controller configured as JBOD disks. The configuration deployment happens as part of the storage profile association to a service profile process.

Cisco UCS Manager enables you to scrub the disks for internal SSDs drives on M5 for blade and rack servers. From Cisco UCS Manager 4.3(4a) onwards, the AHCI controller mode and SWRAID controller-based configurations are supported to scrub the sSATA controlled two internal SATA M.2 drives from UCS Manager interface.

For the PCH Controller Definition configuration in a Cisco UCS Manager boot policy two new devices exist to select: PCH LUN and PCH Disk. EmbeddedLocalLun represents the boot device in SWRAID mode and EmbeddedLocalDisk represent the boot devices in AHCI mode.

The system uses the same scrub policy is used to scrub supported SSDs. If the scrub is Yes, configured LUNs are destroyed as part of disassociation or re-discovery. If the scrub is No, configured LUNs are saved during disassociation and re-discovery.

Cisco UCS Manager supports firmware upgrade for the internal SSDs only when the PCH Controller is in SWRAID mode. It is not supported in AHCI mode.

### FCH Controller Configuration

Fusion Controller Hub (FCH) SSD Controller Definition provides a local storage configuration in storage profiles for AMD based Cisco UCS C125 M5 Server. For AMD processor based servers, the PCH controller is referred to as FCH controller. The controller type remains as PCH in the Cisco UCS Manager GUI.

The FCH Controller works the same as PCH Controller except for the following differences:

- FCH is only in AHCI (JBOD) mode.

**Note** Cisco UCS Manager GUI shows the RAID support as **RAID0, RAID1**, however, Cisco UCS C125 M5 Server supports only AHCI mode.

**Note** You must re-acknowledge the server if you remove or insert disks managed by the PCH Controller.

- There are two FCH controllers:
  - First PCH controller manages SATA disks in the front panel (in the absence of a separate PCIe storage controller)
  - Second PCH controller manages the M.2 SSDs

**Note** For Cisco UCS C125 M5 Server, the PCH IDs are 3 and 4.

**Note** Further information and procedures related to PCH controller in this document are applicable to both Intel and AMD based servers.

## Creating a Storage Profile PCH Controller Definition

You can create a PCH controller definition under a storage profile at the org level or at the service profile level.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | UCS-A# **scope org** *org-name* | Enters the organization mode for the specified organization. To enter the root organization mode, enter / as the *org-name*.<br>**Note**<br>This task assumes the storage profile is at the org level. If the storage profile is at the service |

| | Command or Action | Purpose |
|---|---|---|
| | | profile level, see the example below for the steps to scope to the storage profile definition under the service profile. |
| **Step 2** | UCS-A /org # **scope storage-profile** *storage-profile-name* | Enters storage-profile configuration mode for the selected storage profile. |
| **Step 3** | UCS-A /org/storage-profile # **create controller-def** *controller-definition-name* | Creates a PCH controller definition with the specified name and enters controller-definition configuration mode. |
| **Step 4** | UCS-A /org/storage-profile/controller-def* # **create controller-mode-config** | Creates a PCH controller configuration and enters controller-mode configuration mode. |
| **Step 5** | UCS-A /org/storage-profile/controller-def/controller-mode-config* # **set protect-config** {**yes**\|**no**} | Specifies whether the server retains the configuration in the PCH controller even if the server is disassociated from the service profile. |
| **Step 6** | UCS-A /org/storage-profile/controller-def/controller-mode-config* # **set raid-mode** {**any-configuration** \| **disable-local-storage** \|**no-local-storage** \| **no-raid** \| **raid-0-striped** \| **raid-1-mirrored** \| **raid-5-striped-parity** \| **raid-50--striped-parity-and-striped** \| **raid-6-striped-dual-parity** \| **raid-60-striped-dual-parity-and-striped** \| **raid-10-mirrored-and-striped**} | Specifies the raid mode for the PCH controller. |
| **Step 7** | UCS-A /org/storage-profile/controller-def/controller-mode-config* # **commit-buffer** | Commits the transaction to the system configuration. |

### Example

This example shows how to add a PCH controller definition called "raid1-controller" with raid mode set to RAID 1 Mirrored to the org-level storage profile named "storage-profile-A".

```
UCS-A# scope org /
UCS-A /org # scope storage-profile storage-profile-A
UCS-A /org/storage-profile # create controller-def raid1-controller
UCS-A /org/storage-profile/controller-def* # create controller-mode-config
UCS-A /org/storage-profile/controller-def/controller-mode-config* # set protect-config yes
UCS-A /org/storage-profile/controller-def/controller-mode-config* # set raid-mode
raid-1-mirrored
UCS-A /org/storage-profile/controller-def/controller-mode-config* # commit buffer
```

This example shows how to scope to the service profile called "Service-Profile1", create a storage profile, then create a PCH controller definition called "Raid60Ctrlr" within that storage profile. The controller definition has protection mode off and uses RAID 60 Striped Dual Parity and Striped.

```
UCS-A /org/service-profile # scope org /
UCS-A /org # scope service-profile Service-Profile1
UCS-A /org/service-profile # create storage-profile-def
UCS-A /org/service-profile/storage-profile-def* # create controller-def Raid60Ctrlr
UCS-A /org/service-profile/storage-profile-def/controller-def* # create controller-mode-config
UCS-A /org/service-profile/storage-profile-def/controller-def/controller-mode-config* # set
 protect-config no
UCS-A /org/service-profile/storage-profile-def/controller-def/controller-mode-config* # set
 raid-mode raid-60-striped-dual-parity-and-striped
UCS-A /org/service-profile/storage-profile-def/controller-def/controller-mode-config* #
commit-buffer
```

# Deleting a Storage Profile PCH Controller Definition

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | UCS-A# **scope org** *org-name* | Enters the organization mode for the specified organization. To enter the root organization mode, enter **/** as the *org-name*.<br>**Note**<br>This task assumes the storage profile is at the org level. If the storage profile is at the service profile level, see the example below for the steps to scope to the storage profile definition under the service profile. |
| **Step 2** | UCS-A /org # **scope storage-profile** *storage-profile-name* | Enters storage-profile configuration mode for the selected storage profile. |
| **Step 3** | UCS-A /org/storage-profile # **delete controller-def** *controller-definition-name* | Deletes a PCH controller definition with the specified name. |
| **Step 4** | UCS-A /org/storage-profile* # **commit-buffer** | Commits the transaction to the system configuration. |

### Example

This example shows how to delete a PCH controller definition called "raid1-controller" from the org-level storage profile named "storage-profile-A".

```
UCS-A# scope org
UCS-A /org # scope storage-profile storage-profile-A
UCS-A /org/storage-profile # delete controller-def raid1-controller
UCS-A /org/storage-profile* # commit-buffer
```

# Migrating an M.2 Module

## Migrating an M.2 module in SWRAID

Perform this procedure to migrate an M.2 module in SWRAID mode to a destination server:

### Before you begin

Only UEFI boot mode is supported with software RAID configuration in the controller definition. This condition is applicable even if the drives are not used as boot drive. Ensure that the source and destination server boot mode is set to UEFI and controller definition is configured as same SWRAID (R0/R1).

### Procedure

**Step 1** Gracefully shut down the server.

**Step 2** Physically remove the M.2 module.

The boot mode in the source server for SWRAID M.2 controller configuration in the source server has to be UEFI. Configure the boot policy of destination server with UEFI boot parameters under embedded disk.

**Step 3** Insert the disk in the M.2 module in the destination server.

**Step 4** Power on the server.

**Step 5** Re-acknowledge the server.

## Migrating an M.2 Module in AHCI Mode

Perform this procedure to migrate an M.2 module in NORAID mode to a destination server:

### Before you begin

- If the source server is in legacy boot mode, ensure that the destination server is also in legacy boot mode and controller definition is configured as **NORAID**.
- If the source server is in UEFI boot mode, ensure that the destination server is also in UEFI boot mode and controller definition is configured as **NORAID**.

### Procedure

**Step 1** Gracefully shut down the server.

**Step 2** Physically remove the M.2 module.

**Step 3** Do one of the following:

- If the disk under M.2 controller had boot mode as UEFI on the source server, configure the boot policy of the destination server with UEFI boot parameters.
- If the disk under M.2 controller had boot mode as legacy on the source server, configure the boot policy of the destination server as legacy mode

**Step 4** Insert the M.2 module in the destination server.

**Step 5** Power on the server.

**Step 6** Re-acknowledge the server.

**Note**
If the disk is faulty, the server shows the disk status as **Not Detected**. Perform Replacing a Faulty M.2 Disk, on page 184 to replace the faulty disk.

## Migrating a SWRAID Disk

Perform this procedure to migrate a M.2 disk in SWRAID mode to a destination server:

### Before you begin

Only UEFI boot mode is supported with software RAID configuration in the controller definition. This condition is applicable even if the drives are not used as boot drive. Ensure that the source and destination server boot mode is set to UEFI and controller definition is configured as same SWRAID (R0/R1).

### Procedure

**Step 1** Gracefully shut down the server.

**Step 2** Physically remove the M.2 module and extract the disk.

If the disk is used as SWRAID in the source server the boot mode has to be UEFI and configure boot policy of destination server with UEFI boot parameters under embedded disk.

**Step 3** Insert the disk in the M.2 module in the destination server.

**Step 4** Power on the server.

**Step 5** Re-acknowledge the server.

**Note**
The **Drive State** of the disk should show as **Online**. If the disk is faulty, the sever fails to detect the disk or the **Drive State** shows as **BAD** (or **FAILED**) instead of **Online**. Perform Replacing a Faulty M.2 Disk, on page 184 to replace the faulty disk.

## Migrating a JBOD Disk in AHCI Mode

Perform this procedure to migrate a JBOD disk in NORAID mode to a destination server:

### Before you begin

- If the source server is in legacy boot mode, ensure that the destination server is also in legacy boot mode and controller definition is configured as **NORAID**.
- If the source server is in UEFI boot mode, ensure that the destination server is also in UEFI boot mode and controller definition is configured as **NORAID**.

**Procedure**

**Step 1** Gracefully shut down the server.

**Step 2** Physically remove the module and extract the M.2 disk.

**Step 3** Do one of the following:

- If the disk under M.2 controller had boot mode as UEFI on the source server, configure the boot policy of the destination server with UEFI boot parameters.
- If the disk under M.2 controller had boot mode as legacy on the source server, configure the boot policy of the destination server as legacy mode

**Step 4** Insert the M.2 disk in the M.2 module on the destination server.

**Step 5** Power on the server.

**Step 6** Re-acknowledge the server.

# Hybrid Slot Configuration

## Creating a Hybrid Slot Configuration Policy

Hybrid Slots indicate whether the RAID controller can handle U.3 drives in RAID or Direct attached mode. You can view the Name, Hybrid Slot ID, Drive Type, Current Mode, and Preferred Mode. The applicable values are RAID and Direct.

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | UCS-A# **scope org** *org-name* | Enters the organization mode for the specified organization. To enter the root organization mode, enter / as the *org-name*.<br>**Note**<br>This task assumes the storage profile is at the org level. |
| **Step 2** | UCS-A /org # **scope storage-profile** <*storage-profile-name*> | Enters storage-profile configuration mode for the selected storage profile. |
| **Step 3** | UCS-A /org/storage-profile/ # **set auto-config-mode** <*descr*> | Sets the auto configuration mode. |
| **Step 4** | UCS-A /org/storage-profile* # **create hybrid-drive-slot** <*hybrid-drive-slot-name*> | Creates a hybrid drive slot with the specified name. |
| **Step 5** | UCS-A /org/storage-profile/hybrid-drive-slot* # **set direct-attached-slots** <*slot-num*> | Sets the direct attached slots and enters the hybrid drive slot configuration mode. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 6** | UCS-A /org/storage-profile/hybrid-drive-slot* # **set raid-attached-slots** <*slot-num*> | Sets the Raid attached slots and enters the hybrid drive slot configuration mode. |
| **Step 7** | UCS-A /org/storage-profile/controller-def/hybrid-drive-slot* # **commit-buffer** | Commits the transaction to the system configuration. |

### Example

The following example shows how to create a hybrid slot configuration policy and commits the transaction:

```
UCS-A /server # scope org
UCS-A /org # scope storage-profile
UCS-A /org/storage-profile # set auto-config-mode descr
UCS-A /org/storage-profile # create hybrid-drive-slot
UCS-A /org/storage-profile/hybrid-drive-slot* # set direct-attached-slots 1,2
UCS-A /org/storage-profile/hybrid-drive-slot* # set raid-attached-slots 3,4
UCS-A /org/storage-profile/hybrid-drive-slot* # commit-buffer
UCS-A /org/storage-profile/hybrid-drive-slot
```

The following example shows the details of hybrid slot configuration:

```
UCS-A /org/storage-profile/hybrid-drive-slot # show detail
Hybrid Drive Slot Config:
Direct Attached Drive Slots: 1,2
RAID Attached Drive Slots: 3,4
UCS-A /org/storage-profile/hybrid-drive-slot #
```

The following example shows the detailed view of the hybrid slot configuration:

```
UCS-A /server # show hybrid-slot detail
Hybrid Slot Info:
Hybrid Slot Id: 1
Preferred Mode: Cpu Direct Attached
Current Mode: Controller Attached
Drive Type: Bay Empty
Hybrid Slot Id: 2
Preferred Mode: Cpu Direct Attached
Current Mode: Controller Attached
Drive Type: Bay Empty

Hybrid Slot Id: 3
Preferred Mode: Cpu Direct Attached
Current Mode: Controller Attached
Drive Type: Bay Empty

Hybrid Slot Id: 4
Preferred Mode: Controller Attached
Current Mode: Controller Attached
Drive Type: SAS

Hybrid Slot Id: 101
Preferred Mode: Controller Attached
Current Mode: Controller Attached
Drive Type: Bay Empty
```

```
Hybrid Slot Id: 102
Preferred Mode: Controller Attached
Current Mode: Controller Attached
Drive Type: Bay Empty

Hybrid Slot Id: 103
Preferred Mode: Controller Attached
Current Mode: Controller Attached
Drive Type: Bay Empty

Hybrid Slot Id: 104
Preferred Mode: Controller Attached
Current Mode: Controller Attached
Drive Type: Bay Empty


Drive Type: Bay Empty
```

## Viewing or Modifying a Hybrid Slot Configuration Policy

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | UCS-A# **scope org** *org-name* | Enters the organization mode for the specified organization. To enter the root organization mode, enter **/** as the *org-name*.<br>**Note**<br>This task assumes the storage profile is at the org level. |
| **Step 2** | UCS-A /org # **scope storage-profile** <*storage-profile-name*> | Enters storage-profile configuration mode for the selected storage profile. |
| **Step 3** | UCS-A /org/storage-profile/ # **scope auto-config-mode** <*descr*> | Enters the auto configuration mode. |
| **Step 4** | UCS-A /org/storage-profile* # **scope hybrid-drive-slot** <*hybrid-drive-slot-name*> | Enters a hybrid drive slot with the specified name. |
| **Step 5** | UCS-A /org/storage-profile/hybrid-drive-slot* # **scope direct-attached-slots** <*slot-num*> | Enters the hybrid drive slot configuration mode for directly attached slot. |
| **Step 6** | UCS-A /org/storage-profile/hybrid-drive-slot* # **scope raid-attached-slots** <*slot-num*> | Enters the hybrid drive slot configuration mode for Raid attached slot. |
| **Step 7** | UCS-A /org/storage-profile/controller-def/hybrid-drive-slot* # **commit-buffer** | Commits the transaction to the system configuration. |

### Example

The following example shows how to view a hybrid slot configuration policy and commits the transaction:

```
UCS-A /server # scope org
UCS-A /org # scope storage-profile
UCS-A /org/storage-profile/hybrid-drive-slot* # scope direct-attached-slots 1,2
UCS-A /org/storage-profile/hybrid-drive-slot* # scope raid-attached-slots 3,4
UCS-A /org/storage-profile/hybrid-drive-slot* # commit-buffer
UCS-A /org/storage-profile/hybrid-drive-slot #
```

## Deleting a Hybrid Slot Configuration Policy

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | UCS-A# **scope org** *org-name* | Enters the organization mode for the specified organization. To enter the root organization mode, enter / as the *org-name*.<br>**Note** This task assumes the storage profile is at the org level. |
| **Step 2** | UCS-A /org # **scope storage-profile** <*storage-profile-name*> | Enters the storage profile configuration mode for the selected storage profile. |
| **Step 3** | UCS-A /org/storage-profile* # **delete hybrid-drive-slot** <*hybrid-drive-slot-name*> | Deletes the hybrid drive slot configuration. |
| **Step 4** | UCS-A /org/storage-profile * # **commit-buffer** | Commits the transaction to the system configuration. |

### Example

The following example shows how to create a hybrid slot configuration policy and commits the transaction:

```
UCS-A# scope org
UCS-A /org # scope storage-profile test1
UCS-A /org/storage-profile # delete hybrid-drive-slot test1
UCS-A /org/storage-profile* #  commit-buffer
UCS-A /org/storage-profile #
```

# Replacing a Faulty M.2 Disk

Perform this procedure to replace a faulty M.2 disk.

### Before you begin

Ensure that the SWRAID controller definition is configured and the replacement disk formatted empty drive.

### Procedure

**Step 1** Gracefully power down the server.

**Step 2** Physically remove the faulty M.2 drive. Use the **Serial Number** and **Disk Slot** to identity the faulty disk.

**Step 3** Insert the replacement M.2 drive.

**Step 4** Power on the server.

**Step 5** Wait for the disk to rebuild and then re-acknowledge the server.

**Note**
SWRAID rebuild may take anywhere between 35 to 75 minutes depending on the disk size, disk speed, OS content, and other parameters.

AHCI is a NORAID configuration and hence rebuild is not applicable.

**Note**
After replacing the faulty M.2 drive, the operability state and drive-state of the drive in other slot change to Degraded and Rebuilding. To bring back the drive to normal state, decommission and recommission the blade.

# Associating a Storage Profile with a Service Profile

A storage profile created under org can be referred by multiple service profiles, and a name reference in service profile is needed to associate the storage profile with a service profile.

**Important** Storage profiles can be defined under org and under service profile (dedicated). Hence, a service profile inherits local LUNs from both possible storage profiles. A service profile can have a maximum of two such local LUNs.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | UCS-A# **scope org** *org-name* | Enters the organization mode for the specified organization. To enter the root organization mode, enter / as the *org-name*. |
| **Step 2** | UCS-A /org # **scope service-profile** *service-profile-name* | Enters the specified service profile mode. |
| **Step 3** | UCS-A /org/service-profile # **set storage-profile-name** *storage-profile-name* | Associates the specified storage profile with the service profile.<br>**Note** |

| | Command or Action | Purpose |
|---|---|---|
| | | To dissociate the service profile from a storage profile, use the **set storage-profile-name** command and specify "" as the storage profile name. |
| **Step 4** | UCS-A /org/service-profile* # **commit-buffer** | Commits the transaction to the system configuration. |

### Example

This example shows how to associate a storage profile with a service profile.

```
UCS-A# scope org
UCS-A /org # scope service-profile sp1
UCS-A /org/service-profile # set storage-profile-name stp2
```

This example shows how to dissociate a service profile from a storage profile.

```
UCS-A# scope org
UCS-A /org # scope service-profile sp1
UCS-A /org/service-profile # set storage-profile-name ""
```

# Displaying Details of All Local LUNs Inherited By a Service Profile

Storage profiles can be defined under org and as a dedicated storage profile under service profile. Thus, a service profile inherits local LUNs from both possible storage profiles. It can have a maximum of 2 such local LUNs. You can display the details of all local LUNs inherited by a service profile by using the following command:

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | UCS-A /org/service-profile # **show local-lun-ref** | Displays the following detailed information about all the local LUNs inherited by the specified service profile:<br>• **Name**—LUN name in the storage profile.<br>• **Admin State**—Specifies whether a local LUN should be deployed or not. Admin state can be **Online** or **Undeployed**.<br>When the local LUN is being referenced by a service profile, if the auto-deploy status is **no-auto-deploy** then the admin state will be **Undeployed**, else it will be **Online**. After the local LUN is referenced by a service profile, any change made to |

| | Command or Action | Purpose |
|---|---|---|
| | | this local LUN's auto-deploy status is not reflected in the admin state of the LUN inherited by the service profile.<br>• **RAID Level**—Summary of the RAID level of the disk group used.<br>• **Provisioned Size (GB)**—Size, in GB, of the LUN specified in the storage profile.<br>• **Assigned Size (MB)**—Size, in MB, assigned by UCSM.<br>• **Config State**—State of LUN configuration. The states can be one of the following:<br>  • **Applying**—Admin state is online, the LUN is associated with a server, and the virtual drive is being created.<br>  • **Applied**—Admin state is online, the LUN is associated with a server, and the virtual drive is created.<br>  • **Apply Failed**—Admin stage is online, the LUN is associated with a server, but the virtual drive creation failed.<br>  • **Not Applied**—The LUN is not associated with a server, or the LUN is associated with a service profile, but admin state is undeployed.<br>  • **Not In Use**—Service profile is using the virtual drive, but the virtual drive is not associated with a server.<br>• **Reference LUN**—The preprovisioned virtual drive name, or UCSM-generated virtual drive name.<br>• **Deploy Name**—The virtual drive name after deployment.<br>• **ID**—Virtual drive ID.<br>• **Drive State**—State of the virtual drive. The states are:<br>  • **Unknown**<br>  • **Optimal** |

| | Command or Action | Purpose |
|---|---|---|
| | | • **Degraded**<br>• **Inoperable**<br>• **Partially Degraded**<br>• **Self Test Failed**<br>**Note** The *Self Test Failed* drive state enables you to monitor the health and performance of the virtual drive. In this drive state:<br>• The existing virtual drive operation or a new virtual drive creation works normally, unless the storage controller fails the virtual drive for any of the legitimate faults.<br>• The degree of the virtual drive failure is not displayed. However, most of the operations such as participation in Boot Order Policy, Secure Erase, and LED are still supported, except for the drive state modification.<br>• The drive can soon become unusable and can result in loss of information. |

### Example

```
UCS-A /org/service-profile # show local-lun-ref


Local LUN Ref:

    Profile LUN Name Admin State RAID Level             Provisioned Size (GB)  Assigned
Size (MB)   Config State Referenced Lun Deploy Name ID          Drive State

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

    luna             Online      RAID 0 Striped         1
     1024 Applied      luna-1          luna-1      1003        Optimal

    lunb             Online      RAID 0 Striped         1
     1024 Applied      lunb-1          lunb-1      1004        Optimal
```

```
UCS-A /org/service-profile #
```

```
Local LUN Ref:
    Name             Admin State RAID Level             Provisioned Size (GB)  Assigned
Size (MB)   Config State Referenced Lun Deploy Name ID          Drive State
    ---------------- ----------- ---------------------- ----------------------
-------------------- ------------ -------------- ----------- ------------ ------------
    lun111           Online      RAID 0 Striped         30                     30720
          Applied      lun111-1       lun111-1    1001        Optimal
    lun201           Online      Unspecified            1                      0
          Not Applied
```

# Importing Foreign Configurations for a RAID Controller

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | UCS-A# **scope chassis** *chassis-num* | Enters chassis mode for the specified chassis. |
| **Step 2** | UCS-A /chassis # **scope raid-controller** *raid-contr-id* {**sas** \| **sata**} | Enters RAID controller chassis mode. |
| **Step 3** | UCS-A /chassis/raid-controller # **set admin-state import-foreign-configuration** | Allows import of configurations from local disks that are in the **Foreign Configuration** state. |

### Example

This example shows how to import foreign configurations from local disks that are in the **Foreign Configuration** state:

```
UCS-A# scope chassis 1
UCS-A /chassis # scope raid-controller 1 sas
UCS-A /chassis/raid-controller # set admin-state import-foreign-configuration
UCS-A /chassis/raid-controller* #
```

# Configuring Local Disk Operations

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | UCS-A# **scope chassis** *chassis-num* | Enters chassis mode for the specified chassis. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 2** | UCS-A /chassis # **scope raid-controller** *raid-contr-id* {**sas** \| **sata**} | Enters RAID controller chassis mode. |
| **Step 3** | UCS-A /chassis/raid-controller # **scope local-disk** *local-disk-id* | Enters local disk configuration mode. |
| **Step 4** | UCS-A /chassis/raid-controller/local-disk # **set admin-state** {**clear-foreign-configuration** \| **dedicated-hot-spare** [*admin-vd-id*] \| **prepare-for-removal** \| **remove-hot-spare** \| **unconfigured-good** \| **undo-prepare-for-removal**} | Configures the local disk to one of the following states:<br>• **clear-foreign-configuration**—Clears any foreign configuration that exists in a local disk when it is introduced into a new configuration.<br>• **dedicated-hot-spare**—Specifies the local disk as a dedicated hot spare. The admin virtual drive ID that you can assign ranges from 0 to 4294967295.<br>• **prepare-for-removal**—Specifies that the local disk is marked for removal from the chassis.<br>• **remove-hot-spare**—Specifies that the local disk is no longer a hot spare. Use this only to clear any mismatch faults.<br>• **unconfigured-good**—Specifies that the local disk can be configured.<br>• **undo-prepare-for-removal**—Specifies that the local disk is no longer marked for removal from the chassis. |

### Example

This example shows how to clear any foreign configuration from a local disk:

```
UCS-A /chassis/raid-controller/local-disk # set admin-state clear-foreign-configuration
```

This example shows how to specify a local disk as a dedicated hot spare:

```
UCS-A /chassis/raid-controller/local-disk* # set admin-state dedicated-hot-spare 1001
```

This example shows how to specify that a local disk is marked for removal from the chassis:

```
UCS-A /chassis/raid-controller/local-disk* # set admin-state prepare-for-removal
```

This example shows how to specify that a local disk is marked for removal as a hot spare:

```
UCS-A /chassis/raid-controller/local-disk* # set admin-state remove-hot-spare
```

This example shows how to specify that a local disk is working, but is unconfigured for use:

```
UCS-A /chassis/raid-controller/local-disk* # set admin-state unconfigured-good
```

This example shows how to specify that a local disk is no longer marked for removal from the chassis:

```
UCS-A /chassis/raid-controller/local-disk* # set admin-state undo-prepare-for-removal
```

# Configuring Virtual Drive Properties

All virtual drives in a disk group must be managed by using a single disk group policy.

If you try to associate to a server that does not support these properties, a configuration error will be generated.

Only the following storage controllers support these properties:

- LSI 6G MegaRAID SAS 9266-8i
- LSI 6G MegaRAID SAS 9271-8i
- LSI 6G MegaRAID 9265-8i
- LSI MegaRAID SAS 2208 ROMB
- LSI MegaRAID SAS 9361-8i

For the LSI MegaRAID SAS 2208 ROMB controller, these properties are supported only in the B420-M3 blade server. For the other controllers, these properties are supported in multiple rack servers.

**Note** If you have a setup with the Cisco Boot Optimized M.2 Raid Controller (UCS-M2-HWRAID) or Cisco boot optimized M.2 NVMe RAID controller (UCS-M2-NVRAID), then:

- You can create only one virtual drive
- For **strip-size**, select **64KB** or **32KB**. Selecting any other value results in configuration error.

  For Cisco boot optimized M.2 NVMe RAID controller (UCS-M2-NVRAID), select **64KB** or platform default as the **Strip-Size**. Selecting any other value results in configuration error.
- For **access-policy**, **read-policy**, **write-cache-policy**, **io-policy**, and **drive-cache**, select **platform-default**. Selecting any other value results in configuration error.
- The **Access Policy** for the Raid controller (UCSC-RAID-HP) supports only the **Read Write** option.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | UCS-A# **scope org** *org-name* | Enters the organization mode for the specified organization. To enter the root organization mode, enter / as the *org-name*. |
| **Step 2** | UCS-A /org# **scope disk-group-config-policy** *disk-group-name* | Enters disk group configuration policy mode for the specified disk group name. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 3** | UCS-A /org/disk-group-config-policy* # **create virtual-drive-def** | Creates a virtual drive definition and enters the virtual drive definition mode. |
| **Step 4** | UCS-A /org/disk-group-config-policy/virtual-drive-def* # **set access-policy** *policy-type* | Specifies the access policy. This can be one of the following:<br>• blocked<br>• platform-default<br>• read-only:<br>• read-write |
| **Step 5** | UCS-A /org/disk-group-config-policy/virtual-drive-def* # **set drive-cache** *state* | Specifies the state of the drive cache. This can be one of the following:<br>• enable<br>• disable<br>• no-change<br>• platform-default<br>**Important**<br>In Cisco UCS Release 2.5, the drive cache state cannot be changed. It will remain as **platform-default**, irrespective of the drive cache state that you select. |
| **Step 6** | UCS-A /org/disk-group-config-policy/virtual-drive-def* # **set io-policy** *policy-type* | Specifies the I/O policy. This can be one of the following:<br>• cached<br>• direct<br>• platform-default |
| **Step 7** | UCS-A /org/disk-group-config-policy/virtual-drive-def* # **set read-policy** *policy-type* | Specifies the read policy. This can be one of the following:<br>• normal<br>• platform-default<br>• read-ahead |
| **Step 8** | UCS-A /org/disk-group-config-policy/virtual-drive-def* # **set strip-size** *strip-size* | Specifies the strip size. This can be one of the following:<br>• 64 KB<br>• 128 KB |

| | Command or Action | Purpose |
|---|---|---|
| | | • 256 KB<br>• 512 KB<br>• 1024 KB<br>• platform-default |
| **Step 9** | UCS-A /org/disk-group-config-policy/virtual-drive-def* # **set write-cache-policy** *policy-type* | Specifies the write-cache-policy. This can be one of the following:<br>• always-write-back<br>• platform-default<br>• write-back-good-bbu<br>• write-through |
| **Step 10** | UCS-A /org/disk-group-config-policy/virtual-drive-def* # **commit-buffer** | Commits the transaction to the system configuration. |
| **Step 11** | UCS-A /org/disk-group-config-policy/virtual-drive-def* # **show** | Displays the configured virtual drive properties. |

### Example

This example shows how to configure virtual disk properties:

```
UCS-A# scope org
UCS-A /org # scope disk-group-config-policy raid0policy
UCS-A /org/disk-group-config-policy # create virtual-drive-def
UCS-A /org/disk-group-config-policy/virtual-drive-def* # set access-policy read-write
UCS-A /org/disk-group-config-policy/virtual-drive-def* # set drive-cache enable
UCS-A /org/disk-group-config-policy/virtual-drive-def* # set io-policy cached
UCS-A /org/disk-group-config-policy/virtual-drive-def* # set read-policy normal
UCS-A /org/disk-group-config-policy/virtual-drive-def* # set strip-size 1024
UCS-A /org/disk-group-config-policy/virtual-drive-def* # set write-cache-policy write-through
UCS-A /org/disk-group-config-policy/virtual-drive-def* # commit-buffer
UCS-A /org/disk-group-config-policy/virtual-drive-def # show

Virtual Drive Def:
    Strip Size (KB): 1024KB
    Access Policy: Read Write
    Read Policy: Normal
    Configured Write Cache Policy: Write Through
    IO Policy: Cached
    Drive Cache: Enable
UCS-A /org/disk-group-config-policy/virtual-drive-def #
```

### What to do next

Create a Storage Profile

# Deleting an Orphaned Virtual Drive

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | UCS-A# **scope chassis** *chassis-num* | Enters chassis mode for the specified chassis. |
| **Step 2** | UCS-A /chassis # **scope raid-controller** *raid-contr-id* {**sas** \| **sata**} | Enters RAID controller chassis mode. |
| **Step 3** | (Optional) UCS-A /chassis/raid-controller # **delete virtual-drive id** *virtual-drive-id* | Deletes the orphaned virtual drive with the specified virtual drive ID. |
| **Step 4** | (Optional) UCS-A /chassis/raid-controller # **delete virtual-drive name** *virtual-drive-name* | Deletes the orphaned virtual drive with the specified virtual drive name. |
| **Step 5** | (Optional) UCS-A /chassis/raid-controller # **scope virtual-drive** *virtual-drive-id* | Enters virtual drive mode for the specified orphaned virtual drive. |
| **Step 6** | UCS-A /chassis/raid-controller/virtual-drive # **set admin-state delete** | Deletes the orphaned virtual drive. |
| **Step 7** | UCS-A /chassis/raid-controller/virtual-drive # **commit-buffer** | Commits the transaction to the system configuration. |

### Example

This example shows how to delete an orphan virtual drive by specifying the virtual drive ID.

```
UCS-A# scope chassis 1
UCS-A /chassis # scope raid-controller 1 sas
UCS-A /chassis/raid-controller # show virtual-drive

Virtual Drive:
    ID: 1001
    Name: lun111-1
    Block Size: 512
    Blocks: 62914560
    Size (MB): 30720
    Operability: Operable
    Presence: Equipped
    Oper Device ID: 0
    Change Qualifier: No Change
    Config State: Applied
    Deploy Action: No Action

    ID: 1002
    Name: luna-1
    Block Size: 512
    Blocks: 2097152
    Size (MB): 1024
    Operability: Operable
    Presence: Equipped
    Oper Device ID: 1
    Change Qualifier: No Change
    Config State: Orphaned
```

```
    Deploy Action: No Action

    ID: 1003
    Name: lunb-1
    Block Size: 512
    Blocks: 2097152
    Size (MB): 1024
    Operability: Operable
    Presence: Equipped
    Oper Device ID: 2
    Change Qualifier: No Change
    Config State: Orphaned
    Deploy Action: No Action

    ID: 1004
    Name: lunb-2
    Block Size: 512
    Blocks: 2097152
    Size (MB): 1024
    Operability: Operable
    Presence: Equipped
    Oper Device ID: 3
    Change Qualifier: No Change
    Config State: Orphaned
    Deploy Action: No Action

    ID: 1005
    Name: luna-2
    Block Size: 512
    Blocks: 2097152
    Size (MB): 1024
    Operability: Operable
    Presence: Equipped
    Oper Device ID: 4
    Change Qualifier: No Change
    Config State: Orphaned
    Deploy Action: No Action

...

UCS-A /chassis/raid-controller # delete virtual-drive id 1002
Warning: When committed, the virtual drive will be deleted, which may result in data loss.

UCS-A /chassis/raid-controller # commit-buffer
```

This example shows how to delete an orphan virtual drive by specifying the virtual drive name.

```
UCS-A# scope chassis 1
UCS-A /chassis # scope raid-controller 1 sas
UCS-A /chassis/raid-controller # show virtual-drive

Virtual Drive:
    ID: 1001
    Name: lun111-1
    Block Size: 512
    Blocks: 62914560
    Size (MB): 30720
    Operability: Operable
    Presence: Equipped
    Oper Device ID: 0
    Change Qualifier: No Change
    Config State: Applied
    Deploy Action: No Action

    ID: 1003
```

```
    Name: lunb-1
    Block Size: 512
    Blocks: 2097152
    Size (MB): 1024
    Operability: Operable
    Presence: Equipped
    Oper Device ID: 2
    Change Qualifier: No Change
    Config State: Orphaned
    Deploy Action: No Action

    ID: 1004
    Name: lunb-2
    Block Size: 512
    Blocks: 2097152
    Size (MB): 1024
    Operability: Operable
    Presence: Equipped
    Oper Device ID: 3
    Change Qualifier: No Change
    Config State: Orphaned
    Deploy Action: No Action

    ID: 1005
    Name: luna-2
    Block Size: 512
    Blocks: 2097152
    Size (MB): 1024
    Operability: Operable
    Presence: Equipped
    Oper Device ID: 4
    Change Qualifier: No Change
    Config State: Orphaned
    Deploy Action: No Action

...

UCS-A /chassis/raid-controller # delete virtual-drive name lunb-1
Warning: When committed, the virtual drive will be deleted, which may result in data loss.

UCS-A /chassis/raid-controller # commit-buffer
```

This example shows how to delete an orphan virtual drive by setting the admin-state.

```
UCS-A# scope chassis 1
UCS-A /chassis # scope raid-controller 1 sas
UCS-A /chassis/raid-controller # scope virtual-drive 1004
UCS-A /chassis/raid-controller/virtual-drive # set admin-state delete

Warning: When committed, the virtual drive will be deleted, which may result in data loss.

UCS-A /chassis/raid-controller/virtual-drive # commit-buffer
```

## Renaming an Orphaned Virtual Drive

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | UCS-A# **scope chassis** *chassis-num* | Enters chassis mode for the specified chassis. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 2** | UCS-A /chassis # **scope raid-controller** *raid-contr-id* {**sas** \| **sata**} | Enters RAID controller chassis mode. |
| **Step 3** | UCS-A /chassis/raid-controller # **scope virtual-drive** *virtual-drive-id* | Enters virtual drive mode for the specified virtual drive. |
| **Step 4** | UCS-A /chassis/raid-controller/virtual-drive # **set name** *virtual-drive-name* | Specifies a name for the orphan virtual drive. |
| **Step 5** | UCS-A /chassis/raid-controller/virtual-drive # **commit-buffer** | Commits the transaction to the system configuration. |

### Example

This example shows how to specify a name for an orphan virtual drive.

```
UCS-A /chassis # scope raid-controller 1 sas
UCS-A /chassis/raid-controller # scope virtual-drive 1060
UCS-A /chassis/raid-controller/virtual-drive* # set name vd1
UCS-A /chassis/raid-controller/virtual-drive* # commit-buffer
```

# Boot Policy for Local Storage

You can specify the primary boot device for a storage controller as a local LUN or a JBOD disk. Each storage controller can have one primary boot device. However, in a storage profile, you can set only one device as the primary boot LUN.

Beginning with 4.0(4a), Cisco UCS Manager supports Cisco boot optimized M.2 Raid controller based off Marvell 88SE92xx PCIe to SATA 6Gb/s controller (UCS-M2-HWRAID). The controller supports only UEFI boot mode.

Local storage option in the boot policy supports the boot from the SATA drives in the Cisco boot optimized M.2 Raid controller.

Also, embedded local storage option in the boot policy supports the boot from the SATA drives in the Cisco boot optimized M.2 Raid controller. The primary and the secondary type boot specifically from the M.2 SATA drives.

## Configuring the Boot Policy for a Local LUN

**Note** In Cisco UCS Manager Release 2.5, you cannot configure JBOD as a boot device.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | UCS-A# **scope org** *org-name* | Enters organization mode for the specified organization. To enter the root organization mode, type / as the *org-name* . |
| **Step 2** | UCS-A /org # **scope boot-policy** *policy-name* | Enters organization boot policy mode for the specified boot policy. |
| **Step 3** | UCS-A /org/boot-policy # **create storage** | Creates a storage boot for the boot policy and enters organization boot policy storage mode. |
| **Step 4** | UCS-A /org/boot-policy/storage # **create local** | Creates a local storage location and enters the boot policy local storage mode. |
| **Step 5** | UCS-A /org/boot-policy/storage/local/ # **create local-lun** | Specifies a local hard disk drive as the local storage. |
| **Step 6** | UCS-A /org/boot-policy/storage/local/local-lun # **create local-lun-image-path** {**primary** \| **secondary**} | Specifies the boot order for the LUN that you specify.<br>**Important**<br>Cisco UCS Manager Release 2.2(4) does not support **secondary** boot order. |
| **Step 7** | UCS-A /org/boot-policy/storage/local/local-lun/local-lun-image-path # **set lunname** *lun_name* | Specifies the name of the LUN that you want to boot from. |
| **Step 8** | UCS-A /org/boot-policy/storage/local/*local-storage-device* # **commit-buffer** | Commits the transaction to the system configuration. |

### Example

The following example shows how to create a boot policy named lab1-boot-policy, create a local hard disk drive boot for the policy, specify a boot order and a LUN to boot from, and commit the transaction:

```
UCS-A# scope org /
UCS-A /org* # scope boot-policy lab1-boot-policy
UCS-A /org/boot-policy* # create storage
UCS-A /org/boot-policy/storage* # create local
UCS-A /org/boot-policy/storage/local* # create local-lun
UCS-A /org/boot-policy/storage/local/local-lun # create local-lun-image-path primary
UCS-A /org/boot-policy/storage/local/local-lun/local-lun-image-path # set lunname luna
UCS-A /org/boot-policy/storage/local/local-lun/local-lun-image-path # commit-buffer
UCS-A /org/boot-policy/storage/local/local-lun/local-lun-image-path #
```

### What to do next

Include the boot policy in a service profile and template.

## Configuring the Boot Policy for a Local JBOD Disk

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | UCS-A# **scope org** *org-name* | Enters organization mode for the specified organization. To enter the root organization mode, type / as the *org-name* . |
| **Step 2** | UCS-A /org # **scope boot-policy** *policy-name* | Enters organization boot policy mode for the specified boot policy. |
| **Step 3** | UCS-A /org/boot-policy # **create storage** | Creates a storage boot for the boot policy and enters organization boot policy storage mode. |
| **Step 4** | UCS-A /org/boot-policy/storage # **create local** | Creates a local storage location and enters the boot policy local storage mode. |
| **Step 5** | UCS-A /org/boot-policy/storage/local/ # **create local-jbod** | Specifies the local JBOD as the local storage. |
| **Step 6** | UCS-A /org/boot-policy/storage/local/local-jbod # **create local-lun-image-path***{primary \| secondary}* | |
| **Step 7** | UCS-A /org/boot-policy/storage/local/local-jbod/local-disk-image-path # **set slotnumber** *slotnumber* | |
| **Step 8** | UCS-A /org/boot-policy/storage/local/local-jbod/local-disk-image-path* # **commit-buffer** | Commits the transaction to the system configuration. |

### Example

The following example shows how to create a boot policy named lab1-boot-policy, create an local JBOD disk drive boot for the policy, specify a boot order and a JBOD to boot from, and commit the transaction:

```
UCS-A# scope org /
UCS-A /org* # scope boot-policy lab1-boot-policy
UCS-A /org/boot-policy* # create storage
UCS-A /org/boot-policy/storage* # create local
UCS-A /org/boot-policy/storage/local/ # create local-jbod
UCS-A /org/boot-policy/storage/local/local-jbod* # create local-disk-image-path primary
UCS-A /org/boot-policy/storage/local/local-jbod/local-disk-image-path # set slotnumber 1
UCS-A /org/boot-policy/storage/local/local-jbod/local-disk-image-path* # commit-buffer
UCS-A /org/boot-policy/storage/local/local-jbod/local-disk-image-path #
```

### What to do next

Include the boot policy in a service profile and template.

## Configuring the Boot Policy for an Embedded Local LUN

**Note** Specify one bootable LUN as either primary or secondary boot device. If you specify the bootable LUN as both primary and secondary boot devices, the boot policy will result in the service profile configuration error.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | UCS-A# **scope org** *org-name* | Enters organization mode for the specified organization. To enter the root organization mode, type **/** as the *org-name* . |
| **Step 2** | UCS-A /org # **scope boot-policy** *policy-name* | Enters organization boot policy mode for the specified boot policy. |
| **Step 3** | UCS-A /org/boot-policy # **create storage** | Creates a storage boot for the boot policy and enters organization boot policy storage mode. |
| **Step 4** | UCS-A /org/boot-policy/storage # **create local** | Creates a local storage location and enters the boot policy local storage mode. |
| **Step 5** | UCS-A /org/boot-policy/storage/local/ # **create embedded-local-lun** | Specifies the embedded local LUN as the local storage. |
| **Step 6** | UCS-A /org/boot-policy/storage/local/embedded-local-lun* # **commit-buffer** | Commits the transaction to the system configuration. |

### Example

The following example shows how to create a boot policy named lab1-boot-policy, create an embedded LUN boot for the policy, specify a boot order and a LUN to boot from, and commit the transaction:

```
UCS-A# scope org /
UCS-A /org* # scope boot-policy lab1-boot-policy
UCS-A /org/boot-policy* # create storage
UCS-A /org/boot-policy/storage* # create local
UCS-A /org/boot-policy/storage/local/ # create embedded-local-lun
UCS-A /org/boot-policy/storage/local/embedded-local-lun* # commit-buffer
UCS-A /org/boot-policy/storage/local/embedded-local-lun #
```

### What to do next

Include the boot policy in a service profile and template.

# Configuring the Boot Policy for an Embedded Local Disk

**Note** For Cisco UCS C125 M5 Server, if there is no separate PCIe storage controller, then do not configure boot policy for embedded local disk. Instead, use **Add Local Disk** option.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | UCS-A# **scope org** *org-name* | Enters organization mode for the specified organization. To enter the root organization mode, type **/** as the *org-name* . |
| **Step 2** | UCS-A /org # **scope boot-policy** *policy-name* | Enters organization boot policy mode for the specified boot policy. |
| **Step 3** | UCS-A /org/boot-policy # **create storage** | Creates a storage boot for the boot policy and enters organization boot policy storage mode. |
| **Step 4** | UCS-A /org/boot-policy/storage # **create local** | Creates a local storage location and enters the boot policy local storage mode. |
| **Step 5** | UCS-A /org/boot-policy/storage/local/ # **create embedded-local-jbod** | Specifies the embedded local JBOD as the local storage. |
| **Step 6** | UCS-A /org/boot-policy/storage/local/embedded-local-jbod* # **commit-buffer** | Commits the transaction to the system configuration. |

### Example

The following example shows how to create a boot policy named lab1-boot-policy, create an embedded JBOD disk drive boot for the policy, specify a boot order and a JBOD to boot from, and commit the transaction:

```
UCS-A# scope org /
UCS-A /org* # scope boot-policy lab1-boot-policy
UCS-A /org/boot-policy* # create storage
UCS-A /org/boot-policy/storage* # create local
UCS-A /org/boot-policy/storage/local/ # create embedded-local-jbod
UCS-A /org/boot-policy/storage/local/embedded-local-jbod* # commit-buffer
UCS-A /org/boot-policy/storage/local/embedded-local-jbod #
```

### What to do next

Include the boot policy in a service profile and template.

# Local LUN Operations in a Service Profile

Although a service profile is derived from a service profile template, the following operations can be performed for each local LUN at the individual service profile level:

- Preprovisioning a LUN Name or Claiming an Orphan LUN, on page 202
- Deploying and Undeploying a LUN, on page 203
- Renaming a Service Profile Referenced LUN, on page 204

**Note** Preprovisioning a LUN name, claiming an orphan LUN, and deploying or undeploying a LUN result in server reboot.

## Preprovisioning a LUN Name or Claiming an Orphan LUN

You can preprovision a LUN name or claim an orphan LUN by using the **set ref-name** command. Preprovisioning a LUN name or claiming an orphan LUN can be done only when the admin state of the LUN is **Undeployed**. You can also manually change the admin state of the LUN to **Undeployed** and claim an orphan LUN.

**Important** This operation will reboot the server.

If the LUN name is empty, set a LUN name before claiming it.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | UCS-A# **scope org** *org-name* | Enters the organization mode for the specified organization. To enter the root organization mode, enter / as the *org-name*. |
| **Step 2** | UCS-A /org# **scope service-profile** *service-profile-name* | Enters the specified service profile mode. |
| **Step 3** | UCS-A /org/service-profile# **scope local-lun-ref** *lun-name* | Enters the specified LUN. |
| **Step 4** | UCS-A /org/service-profile/local-lun-ref# **set ref-name** *ref-lun-name* | Sets the referenced LUN name.<br>If this LUN name exists and the LUN is orphaned, its is claimed by the service profile. If this LUN does not exist, a new LUN is created with the specified name. |

- If the LUN exists and is not orphaned, a configuration failure occurs.

- If a LUN is already referred to and the ref-name is changed, it will release the old LUN and will claim or create a LUN with the ref-name. The old LUN is marked as an orphan after the LUN reference is removed from the server.

### Example

This examples shows how to preprovision a LUN name.

```
UCS-A# scope org
UCS-A /org # scope service-profile sp1
UCS-A /org/service-profile* # scope local-lun-ref lun1
UCS-A /org/service-profile/local-lun-ref* # set ref-name lun2
```

## Deploying and Undeploying a LUN

You can deploy or undeploy a LUN by using the **admin-state** command. If the admin state of a local LUN is **Undeployed**, the reference of that LUN is removed and the LUN is not deployed.

**Important** This operation will reboot the server.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | UCS-A# **scope org** *org-name* | Enters the organization mode for the specified organization. To enter the root organization mode, enter / as the *org-name*. |
| **Step 2** | UCS-A /org# **scope service-profile** *service-profile-name* | Enters the specified service profile mode. |
| **Step 3** | UCS-A /org/service-profile# **scope local-lun-ref** *lun-name* | Enters the specified LUN. |
| **Step 4** | UCS-A /org/service-profile/local-lun-ref# **set admin-state** {**online** \| **undeployed**} | Sets the admin state of the specified LUN to **online** or **undeployed**.<br>If a LUN is already referred to and the admin state is set to **undeployed**, it will release the old LUN. The old LUN is marked as orphan after the LUN reference is removed from the server. |

### Example

This examples shows how to deploy a LUN.

```
UCS-A# scope org
UCS-A /org # scope service-profile sp1
UCS-A /org/service-profile* # scope local-lun-ref lun1
UCS-A /org/service-profile/local-lun-ref* # set admin-state online
```

This examples shows how to undeploy a LUN.

```
UCS-A# scope org
UCS-A /org # scope service-profile sp1
UCS-A /org/service-profile* # scope local-lun-ref lun1
UCS-A /org/service-profile/local-lun-ref* # set admin-state undeployed
```

## Renaming a Service Profile Referenced LUN

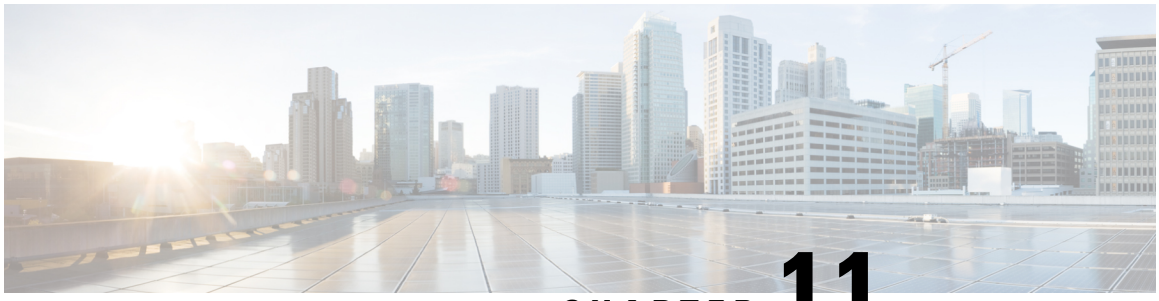
### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | UCS-A# **scope org** *org-name* | Enters the organization mode for the specified organization. To enter the root organization mode, enter **/** as the *org-name*. |
| **Step 2** | UCS-A /org# **scope service-profile** *service-profile-name* | Enters the specified service profile mode. |
| **Step 3** | UCS-A /org/service-profile# **scope local-lun-ref** *lun-name* | Enters the specified LUN. |
| **Step 4** | UCS-A /org/service-profile/local-lun-ref# **set name** | Renames the referenced LUN. |

### Example

This examples shows how to rename a LUN referenced by a service profile.

```
UCS-A# scope org
UCS-A /org # scope service-profile sp1
UCS-A /org/service-profile* # scope local-lun-ref lun1
UCS-A /org/service-profile/local-lun-ref* # set name lun11
```



CHAPTER 11

# Configuring SD Card Support



# FlexFlash Secure Digital Card Support

### Overview

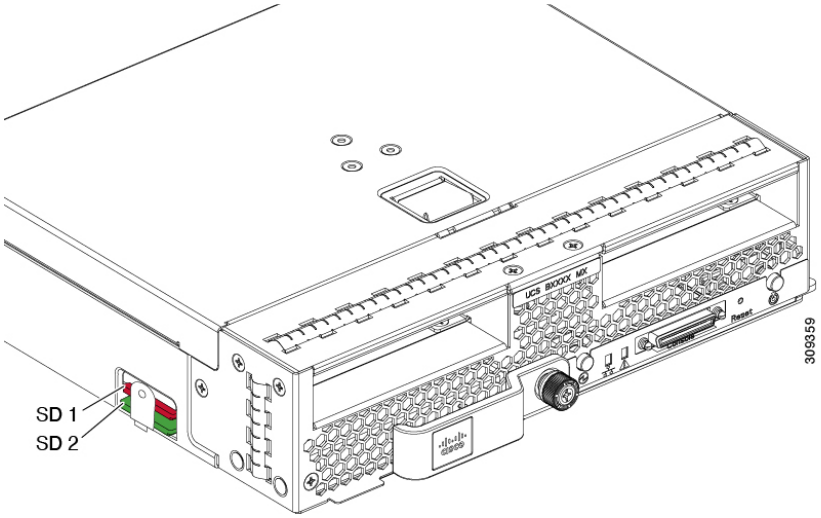
The SD cards are hosted by the Cisco Flexible Flash storage controller, a PCI-based controller which has two slots for SD cards. The cards contain a single partition called HV. When FlexFlash is enabled, Cisco UCS Manager displays the HV partition as a USB drive to both the BIOS and the host operating system.

You can populate one or both the SD card slots that are provided. If two SD cards are populated, you can use them in a mirrored mode.

**Note** Do not mix different capacity cards in the same server.

The SD cards can be used to store operating system boot images or other information. The following figure illustrates the SD card slots.

*Figure 3: SD Card Slots*




FlexFlash is disabled by default. You can enable FlexFlash in a local disk policy used in a service profile. When FlexFlash is enabled in a local disk policy, and the server is capable of supporting SD cards, the FlexFlash controller is enabled during service profile association. If a server is not capable of supporting SD cards or has an older CIMC version, a config failure message is displayed.

If you disable FlexFlash in a supported server, the Hypervisor or HV partition is immediately disconnected from the host. The FlexFlash controller will also be disabled as part of a related service profile disassociation.

The FlexFlash controller supports RAID-1 for dual SD cards. The FlexFlash scrub policy erases the HV partition in both cards, and brings the cards to a healthy RAID state.

You can configure new SD cards in a RAID pair and format them using one of the following methods:

- Format the SD cards.
- For an associated server, create a FlexFlash scrub policy and disassociate the service profile from the server. For an unassociated server, create a FlexFlash scrub policy and reacknowledge the server after modifying the default scrub policy.

  The *Scrub Policy Settings* section in the *Cisco UCS Manager Server Management Guide* provides more details about the usage of the scrub policy.

**Note** Disable the scrub policy as soon as the pairing is complete.

To boot from the HV partition, the SD card must be present in the boot policy used in the service profile.

### FlexFlash Firmware Management

The FlexFlash controller firmware is bundled as part of the CIMC image. When you upgrade the CIMC, if a newer firmware version is available for the FlexFlash controller, the controller can no longer be managed, and the FlexFlash inventory displays the **Controller State** as **Waiting For User Action** and the **Controller Health** as **Old Firmware Running**. To upgrade the FlexFlash controller firmware, you need to perform a board controller update. For more information, see the appropriate *Cisco UCS B-Series Firmware Management*

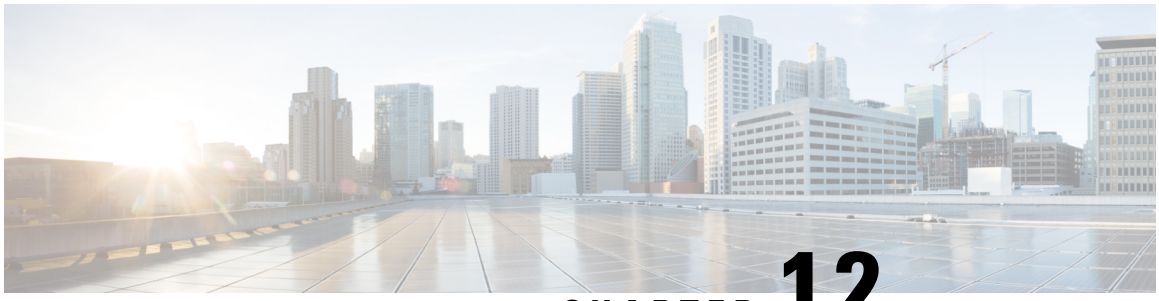
*Guide*, available at the following URL: http://www.cisco.com/en/US/products/ps10281/products_installation_and_configuration_guides_list.html.

**Limitations for the Cisco Flexible Flash Storage Controller:**

- The Cisco Flexible Flash storage controller only supports 16 GB, 32 GB, and 64 GB SD cards.
- We do not recommend using an SD card from a rack server in a blade server, or using an SD card from a blade server in a rack server. Switching SD cards between server types might result in data loss from the SD card.
- Some Cisco UCS C-Series rack-mount servers have SD cards with four partitions: HV, HUU, SCU, and Drivers. Only the HV partition is visible in Cisco UCS Manager. You can migrate a four-partition SD card to a single HV partition card with a FlexFlash scrub policy.
- The FlexFlash controller does not support RAID-1 sync (mirror rebuild). If the SD cards are in a degraded RAID state, or if any metadata errors are reported by the controller, you must run the FlexFlash scrub policy to pair the cards for RAID. For more information about the FlexFlash scrub policy, see Server-Related Policies. The following conditions might result in degraded RAID or metadata errors:
  - Inserting a new or used SD card in one slot, when the server already has an SD card populated in the second slot.
  - Inserting two SD cards from different servers.
- The server firmware version must be at 2.2(1a) or higher.

# FlexUtil Secure Digital Card Support

The C-Series M5 Rack-Mount servers support a Micro-SD (FlexUtil) memory card for storage. UCS Manager however does not provide management support for Micro-SD card.



CHAPTER 12

# Mini Storage



# Mini Storage

The mini storage slot is a new slot that is present on the Cisco UCS M5 blade and rack servers. This slot can be empty, populated with an SD storage module, or populated with an M.2 SATA module.

**Note** Cisco UCS Manager does not support micro-SD card.

The mini storage SD module consists of an in-built SD controller and two SD cardslots. These cards have RAID 1 capability.

The mini M.2 SATA module consists of two SATA slots. The PCH controller present on the server controls the SATA drives on this module.

Starting with 4.0(4a), Cisco UCS Manager supports Cisco boot optimized M.2 Raid controller based off Marvell 88SE92xx PCIe to SATA 6Gb/s controller (UCS-M2-HWRAID) for mini storage.

You can use Cisco UCS Manager to inventory and manage the mini storage modules.

## Viewing Mini Storage Properties

Mini storage modules are supported only on M5 and higher servers.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | UCS-A# **scope server** [*chassis-num*/*server-num* \| *dynamic-uuid*] | Enters server mode for the specified server. |
| **Step 2** | UCS-A /chassis/server # **show mini-storage** [**detail**] | Displays detailed information about the mini storage module for the specified server. |

### Example

This example displays detailed information about the mini storage module for server 6:

```
UCS-A# scope server 1/6
UCS-A /chassis/server # show mini-storage detail

Mini Storage Module:
    ID: 1
    Type: M2
    Model: UCS-MSTOR-M2
    Vendor: Cisco Systems Inc
    HW Rev: 0
    Serial: FCH2050JDHM
    VID: V00
    Part Number: 73-17926-04
    Product Name: Cisco UCS Mini-Storage Carrier for M.2
    Caption: Cisco UCS Mini-Storage Carrier for M.2 (holds up to 2)
    Description: Dual M.2 Mini-Storage Carrier (holds up to 2 M.2 modules)
```

# Viewing the Storage Controller for the Mini Storage

Mini storage modules are supported only on M5 and higher servers.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | UCS-A# **scope server** [*chassis-num*/*server-num* \| *dynamic-uuid*] | Enters server mode for the specified server. |
| **Step 2** | UCS-A /chassis/server # **scope mini-storage** *id* **m2** \| **sd** | Enters mini-storage mode for the specified server and mini storage card type. |
| **Step 3** | UCS-A /chassis/server/mini-storage # **show referenced-controller** [**detail**] | Displays information about the storage controller referenced by the mini storage module for the specified server. |

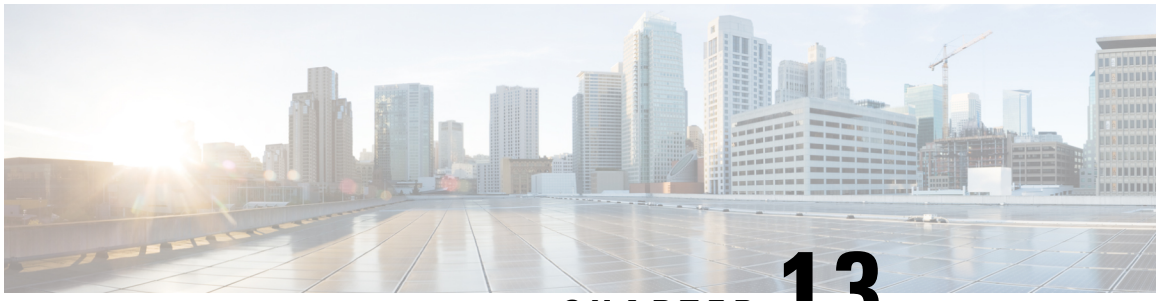
### Example

This example displays information about the storage controller for the M.2 mini storage card in server 6:

```
UCS-A# scope server 1/6
UCS-A /chassis/server # scope mini-storage 1 m2
UCS-A /chassis/server/mini-storage # show referenced-controller detail

Referenced Controller:
    ID: 1
    Type: PCH
```

This example displays information about the storage controller for the SD mini storage card in server 3:

```
UCS-A# scope server 1/3
UCS-A /chassis/server # scope mini-storage 1 sd
UCS-A /chassis/server/mini-storage # show referenced-controller detail

Referenced Controller:
    ID: 1
    Type: PCH
```



CHAPTER 13

# SED Security Policies



# Security Policies for Self-Encrypting Drives

Self-Encrypting Drives (SEDs) have special hardware that encrypts incoming data and decrypts outgoing data in real-time. The data on the disk is always encrypted in the disk and stored in the encrypted form. The encrypted data is always decrypted on the way out of the disk. A media encryption key controls this encryption and decryption. This key is never stored in the processor or memory. Cisco UCS Manager supports SED security policies on Cisco UCS C-Series servers, B-Series servers, , X-Series servers, and S-Series servers.

SEDs must be locked by providing a security key. The security key, which is also known as Key-Encryption Key or an authentication passphrase, is used to encrypt the media encryption key. If the disk is not locked, no key is required to fetch the data.

Cisco UCS Manager enables you to configure security keys locally or remotely. When you configure the key locally, you must remember the key. If you forget the key, it cannot be retrieved, and the data is lost. You can configure the key remotely by using a key management server (also known as KMIP server). This method addresses the issues related to safe-keeping and retrieval of the keys in the local management.

The encryption and decryption for SEDs is done through the hardware. Thus, it does not affect the overall system performance. SEDs reduce the disk retirement and redeployment costs through instantaneous cryptographic erasure. Cryptographic erasure is done by changing the media encryption key. When the media

encryption key of a disk is changed, the data on the disk cannot be decrypted, and is immediately rendered unusable.

**Guidelines**

To ensure secure and efficient management of Self-Encrypting Drives (SEDs) in Cisco UCS Manager, remember these guidelines:

- The deletion of secured Logical Unit Numbers (LUNs) is only possible using a scrub policy.
- Reconfiguration and deletion of secured LUNs are not allowed on a disassociated server.
- Data sanitization is not permitted until security is enabled.
- If incorrect credentials are provided, the Finite State Machine (FSM) completes without any error, but the LUNs become inoperable, and the drives get locked.
- A power cycle of the server is triggered if any changes are made to the security settings in the storage profile due to the Enterprise Key Management System (EKMS).
- When secured drives are moved between setups, the first association should occur only with security details and no LUN configuration to unlock the drives.
- Changes to login details do not trigger a change. A fresh storage profile association or modification along with other properties, is required.

# Security Flags of the Controller and Disk

Security flags indicate the current security status of the storage controller and disks.

The storage controller and disks have the following security flags:

- Security Capable—Indicates that the controller or disk is capable of supporting SED management.
- Security Enable—Indicates that the security-key is programmed on the controller or disk, and security is enabled on the device. This flag is set when you configure a security policy and associate it to a server, making the controller and disk secure. This flag is not set on an HX device.
- Secured—Indicates that the security-key is programmed on the controller or disk, and security is enabled on the HX device.

The following security flags are exclusive to storage disks:

- Locked—Indicates that the disk key does not match the key on the controller. This happens when you move disks across servers that are programmed with different keys. The data on a locked disk is inaccessible and the operating system cannot use the disk. To use this disk, you must either unlock the disk or secure erase the foreign configuration.
- Foreign Secured—Indicates that a secure disk is in foreign configuration. This happens when you unlock a locked disk with the right key, but the disk is in a foreign configuration state and the data on it is encrypted. To use this disk, you can either import the foreign configuration or clear the foreign config.

# Secure Data Deletion

The Commission Regulation (EU) 2019/424 requires that data be securely disposed of.

Secure data disposal is accomplished by using commonly available tools that erase the data from the various/drives, memory, and storage in the Cisco UCS servers and reset them to factory settings.

Secure data deletion for compliance with Commission Regulation (EU) 2019/424 is supported for the followingCisco UCS servers:

- Cisco UCS B200
- Cisco UCS B480
- Cisco UCS C125
- Cisco UCS C220
- Cisco UCS C240
- Cisco UCS C480
- Cisco UCS S3260

You must be familiar with what devices are present in your UCS server and run the appropriate tools for secure data deletion. In some cases, you may need to run multiple tools.

Full instructions on how to securely erase data are available at: https://www.cisco.com/web/dofc/18794277.pdf.

# Managing Local Security Policies

## Creating a Local Security Policy

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | UCS-A # **scope org** | Enters the root organization mode. |
| **Step 2** | UCS-A /org # **create storage-profile** *storage-profile-name* | Creates a storage profile with the specified name at the organization level and enters the storage-profile configuration mode. |
| **Step 3** | UCS-A /org/storage-profile* # **create security** | Creates a security policy for the specified storage profile and enters the security policy mode. |
| **Step 4** | UCS-A /org/storage-profile/security* # **create drive-security** | Creates a drive security policy for the specified storage profile security and enters the drive security policy mode. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 5** | UCS-A /org/storage-profile/security/drive-security* # **create local** | Creates a local security policy for the specified storage profile and enters the local policy mode. |
| **Step 6** | UCS-A /org/storage-profile/security/drive-security/local* # **set security-key** *security-key* | Sets the specified security key for the local policy. The security key must have 32 characters. |
| **Step 7** | UCS-A /org/storage-profile/security/drive-security/local* # **commit-buffer** | Commits the transaction to the system configuration. |

### Example

This example show how to create a local security policy with a security key:

```
UCS-A# scope org
UCS-A /org # create storage-profile stp-demo
UCS-A /org/storage-profile* # create security
UCS-A /org/storage-profile/security* # create drive-security
UCS-A /org/storage-profile/security/drive-security* # create local
UCS-A /org/storage-profile/security/drive-security/local* # set security-key
thereare32charactersinthisseckey
UCS-A /org/storage-profile/security/drive-security/local* # commit-buffer
UCS-A /org/storage-profile/security/drive-security/local #
```

# Modifying the Security Key of a Local Security Policy

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | UCS-A # **scope org** | Enters the root organization mode. |
| **Step 2** | UCS-A /org # **scope storage-profile** *storage-profile-name* | Enters the storage-profile configuration mode for the specified storage profile. |
| **Step 3** | UCS-A /org/storage-profile # **scope security** | Enters the security policy mode for the specified storage profile. |
| **Step 4** | UCS-A /org/storage-profile/security # **scope drive-security** | Enters the drive security policy mode for the specified storage profile security. |
| **Step 5** | UCS-A /org/storage-profile/security/drive-security # **scope local** | Enters the local policy mode for the the specified storage profile. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 6** | UCS-A /org/storage-profile/security/drive-security/local # **set deployed-security-key** *existing-security-key* | Specifies the existing key deployed on the server to configure a new key. |
| **Step 7** | UCS-A /org/storage-profile/security/drive-security/local* # **set security-key** *new-security-key* | Sets the new security key for the local policy. |
| **Step 8** | UCS-A /org/storage-profile/security/drive-security/local* # **commit-buffer** | Commits the transaction to the system configuration |

### Example

This example shows how to modify the security key of a local security policy:

```
UCS-A# scope org
UCS-A /org # scope storage-profile stp-demo
UCS-A /org/storage-profile # scope security
UCS-A /org/storage-profile/security # scope drive-security
UCS-A /org/storage-profile/security/drive-security # scope local
UCS-A /org/storage-profile/security/drive-security/local # set deployed-security-key
thereare32charactersinthisseckey
UCS-A /org/storage-profile/security/drive-security/local* # set security-key
thereare32charactersinthisnewkey
UCS-A /org/storage-profile/security/drive-security/local* # commit-buffer
UCS-A /org/storage-profile/security/drive-security/local #
```

# Modifying the Security Policy from Local to Remote

### Before you begin

Ensure that you have created a KMIP client certificate policy.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | UCS-A # **scope org** | Enters the root organization mode. |
| **Step 2** | UCS-A /org # **scope storage-profile** *storage-profile-name* | Enters the storage-profile configuration mode for the selected storage profile. |
| **Step 3** | UCS-A /org/storage-profile # **scope security** | Enters the security policy mode for the specified storage profile. |
| **Step 4** | UCS-A /org/storage-profile/security # **scope drive-security** | Enters the drive security policy mode for the specified storage profile security. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 5** | UCS-A /org/storage-profile/security/drive-security # **create remote** | Creates and enters the remote policy mode. |
| **Step 6** | UCS-A /org/storage-profile/security/drive-security/remote* # **set deployed-security-key** *existing-security-key* | Specifies the existing key deployed on the server. |
| **Step 7** | UCS-A /org/storage-profile/security/drive-security/remote* # **set primary-server** *primary-server-name* | Sets the primary server hostname or IP address. |
| **Step 8** | (Optional) UCS-A /org/storage-profile/security/drive-security/remote* # **set secondary-server** *secondary-server-name* | Sets the secondary server hostname or IP address. |
| **Step 9** | (Optional) UCS-A /org/storage-profile/security/drive-security/remote* # **set port** *kmip-server-port-number* | Sets the port number of the KMIP server. KMIP server port numbers can range from 1024 to 65535. |
| **Step 10** | UCS-A /org/storage-profile/security/drive-security/remote* # **set server-certificate** | Sets the KMIP certificate to the remote security policy. |
| **Step 11** | (Optional) UCS-A /org/storage-profile/security/drive-security/remote* # **set timeout** *timeout-seconds* | Sets the number of seconds in which communication between the storage and the KMIP server times out. Timeout can range from 5 seconds to 20 seconds. |
| **Step 12** | UCS-A /org/storage-profile/security/drive-security/remote* # **commit-buffer** | Commits the transaction to the system configuration. |
| **Step 13** | UCS-A /org/storage-profile/security/drive-security/remote # **exit** | Enters the drive security policy mode. |
| **Step 14** | UCS-A /org/storage-profile/security/drive-security # **delete local** | Deletes the existing local security policy. |
| **Step 15** | UCS-A /org/storage-profile/security/drive-security* # **commit-buffer** | Commits the transaction to the system configuration. |

### Example

This example shows how to modify a security policy from local to remote:

```
UCS-A # scope org
UCS-A /org # scope storage-profile stp-demo
```

```
UCS-A /org/storage-profile # scope security
UCS-A /org/storage-profile/security # scope drive-security
UCS-A /org/storage-profile/security/drive-security # create remote
UCS-A /org/storage-profile/security/drive-security/remote* # set deployed-security-key
thereare32charactersinthisseckey
UCS-A /org/storage-profile/security/drive-security/remote* # set primary-server 10.10.10.1
UCS-A /org/storage-profile/security/drive-security/remote* # set secondary-server 10.10.10.2
UCS-A /org/storage-profile/security/drive-security/remote* # set port 5696
UCS-A /org/storage-profile/security/drive-security/remote* # set server-certificate
Enter lines one at a time. Enter ENDOFBUF to finish. Press ^C to abort.
Prompt Certificate:
>-----BEGIN CERTIFICATE-----
    MIIEEDCCAvigAwIBAgIGALOfZVDsMA0GCSqGSIb3DQEBCwUAMIGQMSowKAYDVQQD
    EyFDRyBDQSBTIG9uIHZvcm1ldHJpY2RzbS5jaXNjby5jb20xFTATBgNVBAsTDFNh
    dmJ1U3RvcmRldjEWMBQGA1UEChMNQ2lzY28gU3lzdGVtczERMA8GA1UEBxMIU2Fu
    IEpvc2UxEzARBgNVBAgTCkNhbGlmb3JuaWExCzAJBgNVBAYTAlVTMB4XDTE2MDkw
    NzE5MzMwMVoXDTI2MDkwOTE5MzMwMVowgZAxKjAoBgNVBAMTIUNHIENBIFMgb24g
    dm9ybWV0cmljZHNtLmNpc2NvLmNvbTEVMBMGA1UECxMMU2F2YnVTdG9yZGV2MRYw
    FAYDVQQKEw1DaXNjbyBTeXN0ZW1zMREwDwYDVQQHEwhTYW4gSm9zZTETMB>EGA1UE
    CBMKQ2FsaWZvcm5pYTELMAkGA1UEBhMCVVMwggEiMA0GCSqGSIb3DQEBAQUAA4IB
    DwAwggEKAoIBAQDhX2UdIV>TQTchGo1FjAc5u1W9zAo/YkjD22ANpbEPiAmgWL97c
    Xwj7yzArflrZ2kWvQCm4f6AdLOFUWzbuo+Fxd3rurd>w6BhJXdLj8Piq8094PqCLp
    qdUF83SsRVVbCXHxOqdk9jsSQrvTcV4PloNrelMLq/mOqsaODs+us4ng7sMDtGXv
    LeKFC8DUEm0GlGQACwiJ3s9O4+P2CI/d4P/>EyWwqABf3YJmAI1EQyUnoTwrg6EgY
    ZvcpHsmjXnbBZrL+ON7FBcbrTanvjyJxE6tFf5cRPGhymfna7Fd3lfVwZCcGIoR+
    EOIAwgetzIRM6FzMiV2/tDT8STo/oo5Tg3dDAgMBAAGjbj>BsMBIGA1UdEwEB/wQI
    MAYBAf8CAQAwDgYDVR0PAQH/>>>>>>>>>BAQDAgEGMB0GA1UdDgQWBBRnYyFiAK2lEDZJNC0Y
    VlIqMgiUJDAnBgNVHSMEIDAegBRnYyFiAK2lEDZJNC0YVlIqMgiUJIIGALOfZVDs
    MA0GCSqGSIb3DQEBCwUAA4IBAQAfhB2+Ft8V2ELAFa7PcG/rU09ux7LYcCjt3STa
    mzKdZ7Rn5COvknKrJX+EefT7xl03CQXT9aeSAddQUOCy8fhiPoaMFrlTgs1hdS0p
    NJvfxV6QCun2UMRSuxWfG>0QFfofnXeIGkAmEYOpUdArSOTbtt4v6Lja1A+KEsvWW
    5KaVemo2nsd+iD0IPCOhpShAgaAwpnYUq9mLfVgvV07Z+hmkuOIQTZ2+h+pJQtE0
    +U5qaTts4pMXpqQPjlid0NMuaPug1SpSD7KBsjwR1SzehzPdns16uprmvWa3VBk3
    OK6y55FoIu+Wg9i/8kmfkghyGwTfo6weEKbleuVwupvpriMF>
    -----END CERTIFICATE-----

UCS-A /org/storage-profile/security/drive-security/remote* # commit-buffer
UCS-A /org/storage-profile/security/drive-security/remote # exit
UCS-A /org/storage-profile/security/drive-security # delete local
UCS-A /org/storage-profile/security/drive-security* # commit-buffer
UCS-A /org/storage-profile/security/drive-security #
```

# Inserting a Secured Disk into a Server with a Local Security Policy

When you insert a secured disk into a server, one of the following will occur:

- The security-key on the drive matches that of the server and it automatically gets unlocked.
- The security-key on the disk is different from the security-key on the server. The disk will appear as a locked disk. You can do one of the following on a locked disk:
  - Erase the secure foreign configuration to delete all data on the disk.
  - Unlock the disk by providing the correct key of the disk. After unlocking the disk, the disk will be in the Foreign Secured state. You must immediately import or clear the foreign configuration for these disks.

**Note** If you unlock another set of disks before importing the foreign configuration for the current set of disks, the current set of disks become locked again and go in to the Locked state.

# KMIP Client Certificate Policy

You can configure the key remotely by using a key management server, which is also known as KMIP server. You must create a KMIP client certificate policy before creating a remote policy. The hostname that is used for generating the certificate is the serial number of the KMIP server.

You can create a certificate policy from two separate scopes:

- Global scope—You can initially create a global certificate policy in this scope. Any modification of the certificate in this scope will not result in the regeneration of the certificate.
- Server scope—You can create or modify a certificate policy in this scope. This will result in a regeneration of the certificates. Such a certificate is specific to the server, and, for this server, overrides the global certificate.

After you create a KMIP client certificate policy, do one of the following:

- Copy the generated certificate to the KMIP Server.
- Use the generated Certificate Signing Request to get a CA-signed certificate. Copy this CA-signed certificate to the CIMC.

## Creating a Global KMIP Client Certificate Policy

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | UCS-A # **scope security** | Enters the security mode. |
| **Step 2** | UCS-A /security # **create kmip-client-cert-policy** | Creates the KMIP certificate policy and enters the KMIP client certificate policy mode. |
| **Step 3** | UCS-A /security/kmip-client-cert-policy* # **set country** *country-code* | Specifies the country code for the KMIP certificate policy. The country code must contain 2 letters in upper case. |
| **Step 4** | UCS-A /security/kmip-client-cert-policy* # **set locality** *locality-code* | Specifies the name of the locality or city for the KMIP certificate policy. Enter up to 32 characters for the locality name. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 5** | UCS-A /security/kmip-client-cert-policy* # **set org-name** *org-name* | Specifies the organization name requesting the KMIP certificate policy. Enter up to 32 characters for the organization name. |
| **Step 6** | UCS-A /security/kmip-client-cert-policy* # **set org-unit-name** *unit-name* | Specifies the organizational unit name requesting the KMIP certificate policy. Enter up to 64 characters for the organizational unit name. |
| **Step 7** | UCS-A /security/kmip-client-cert-policy* # **set state** *state-code* | Specifies the name of the state, province, or county for the KMIP certificate policy. Enter up to 32 characters for the state name. |
| **Step 8** | (Optional) UCS-A /security/kmip-client-cert-policy* # **set email** *email-address* | Specifies the email address associated with the request. |
| **Step 9** | (Optional) UCS-A /security/kmip-client-cert-policy* # **set validity** *days* | Specifies the validity of the certificate in number of days. The validity can range between 365 days and 3650 days. |
| **Step 10** | UCS-A /security/kmip-client-cert-policy* # **commit-buffer** | Commits the transaction to the system configuration. |
| **Step 11** | UCS-A /security/kmip-client-cert-policy # **show** | Displays details of the KMIP certificate policy. |

### Example

This example shows how to create a KMIP certificate policy.

```
UCS-A# scope security
UCS-A /security # create kmip-client-cert-policy
UCS-A /security/kmip-client-cert-policy* # set country IN
UCS-A /security/kmip-client-cert-policy* # set locality BLR
UCS-A /security/kmip-client-cert-policy* # set org-name XYZ
UCS-A /security/kmip-client-cert-policy* # set org-unit-name Ops
UCS-A /security/kmip-client-cert-policy* # set state KA
UCS-A /security/kmip-client-cert-policy* # commit-buffer
UCS-A /security/kmip-client-cert-policy # show

KMIP Client certificate policy:
Certificate request country name: IN
State, province or county (full name): KA
Locality name (eg, city): BLR
Organisation name (eg, company): XYZ
Organisational Unit Name (eg, section): Ops
Certificate request e-mail name:
Validity of certificate in number of days: 1095
UCS-A /security/kmip-client-cert-policy #
```

# Creating a KMIP Client Certificate for a Server

You can create a KMIP client certificate policy for a server. This certificate is applicable only to the specific server, and overrides the global KMIP client certificate.

The hostname that used to create the certificate when using this policy is the serial number of the server.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | UCS-A # **scope server** *server-number* | Enters the server configuration mode for the specified server. |
| **Step 2** | UCS-A /server # **create kmip-client-cert-policy** | Creates the KMIP certificate policy and enters the KMIP client certificate policy mode. |
| **Step 3** | UCS-A /server/kmip-client-cert-policy* # **set country** *country-code* | Specifies the country code for the KMIP certificate policy. The country code must contain 2 letters in upper case. |
| **Step 4** | UCS-A /server/kmip-client-cert-policy* # **set locality** *locality-code* | Specifies the name of the locality or city for the KMIP certificate policy. Enter up to 32 characters for the locality name. |
| **Step 5** | UCS-A /server/kmip-client-cert-policy* # **set org-name** *org-name* | Specifies the organization name requesting the KMIP certificate policy. Enter up to 32 characters for the organization name. |
| **Step 6** | UCS-A /server/kmip-client-cert-policy* # **set org-unit-name** *unit-name* | Specifies the organizational unit name requesting the KMIP certificate policy. Enter up to 64 characters for the organizational unit name. |
| **Step 7** | UCS-A /server/kmip-client-cert-policy* # **set state** *state-code* | Specifies the name of the state, province, or county for the KMIP certificate policy. Enter up to 32 characters for the state name. |
| **Step 8** | (Optional) UCS-A /server/kmip-client-cert-policy* # **set email** *email-address* | Specifies the email address associated with the request. |
| **Step 9** | (Optional) UCS-A /server/kmip-client-cert-policy* # **set validity** *days* | Specifies the validity of the certificate in number of days. The validity can range between 365 days and 3650 days. |
| **Step 10** | UCS-A /server/kmip-client-cert-policy* # **commit-buffer** | Commits the transaction to the system configuration. |
| **Step 11** | UCS-A /server/kmip-client-cert-policy # **show** | Displays details of the KMIP certificate. |

### Example

This example shows how to create a KMIP certificate on a rack-mount server.

```
UCS-A# scope server 5
UCS-A /server # create kmip-client-cert-policy
UCS-A /server/kmip-client-cert-policy* # set country IN
UCS-A /server/kmip-client-cert-policy* # set locality BLR
UCS-A /server/kmip-client-cert-policy* # set org-name XYZ
UCS-A /server/kmip-client-cert-policy* # set org-unit-name Ops
UCS-A /server/kmip-client-cert-policy* # set state KA
UCS-A /server/kmip-client-cert-policy* # commit-buffer
UCS-A /server/kmip-client-cert-policy* # show

KMIP Client certificate policy:
Certificate request country name: IN
State, province or county (full name): KA
Locality name (eg, city): BLR
Organisation name (eg, company): XYZ
Organisational Unit Name (eg, section): Ops
Certificate request e-mail name:
Validity of certificate in number of days: 1095
UCS-A /server/kmip-client-cert-policy #
```

This example shows how to create a KMIP certificate on a blade server.

```
UCS-A# scope server 1/5
UCS-A chassis/server # create kmip-client-cert-policy
UCS-A chassis/server/kmip-client-cert-policy* # set country IN
UCS-A chassis/server/kmip-client-cert-policy* # set locality BLR
UCS-A chassis/server/kmip-client-cert-policy* # set org-name XYZ
UCS-A chassis/server/kmip-client-cert-policy* # set org-unit-name Ops
UCS-A chassis/server/kmip-client-cert-policy* # set state KA
UCS-A chassis/server/kmip-client-cert-policy* # commit-buffer
UCS-A chassis/server/kmip-client-cert-policy* # show

KMIP Client certificate policy:
Certificate request country name: IN
State, province or county (full name): KA
Locality name (eg, city): BLR
Organisation name (eg, company): XYZ
Organisational Unit Name (eg, section): Ops
Certificate request e-mail name:
Validity of certificate in number of days: 1095
UCS-A /server/kmip-client-cert-policy #
```

# Managing Remote Security Policies

## Creating a Remote Security Policy

### Before you begin

Ensure that you have created a KMIP client certificate policy.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | UCS-A # **scope org** | Enters the root organization mode. |
| **Step 2** | UCS-A /org # **scope storage-profile** *storage-profile-name* | Enters the storage-profile configuration mode for the selected storage profile. |
| **Step 3** | UCS-A /org/storage-profile # **create security** | Creates and enters the security mode. |
| **Step 4** | UCS-A /org/storage-profile/security* # **create drive-security** | Creates and enters the drive-security mode. |
| **Step 5** | UCS-A /org/storage-profile/security/drive-security* # **create remote** | Creates and enters the remote policy mode. |
| **Step 6** | UCS-A /org/storage-profile/security/drive-security/remote* # **set primary-server** *primary-server-name* | Sets the primary server hostname or IP address. |
| **Step 7** | (Optional) UCS-A /org/storage-profile/security/drive-security/remote* # **set secondary-server** *secondary-server-name* | Sets the secondary server hostname or IP address. |
| **Step 8** | (Optional) UCS-A /org/storage-profile/security/drive-security/remote* # **set port** *kmip-server-port-number* | Sets the port number of the KMIP server. KMIP server port numbers can range from 1024 to 65535. |
| **Step 9** | UCS-A /org/storage-profile/security/drive-security/remote* # **set server-certificate** | Sets the KMIP certificate to the remote security policy. |
| **Step 10** | (Optional) UCS-A /org/storage-profile/security/drive-security/remote* # **set timeout** *timeout-seconds* | Sets the number of seconds in which communication between the storage and the KMIP server times out. Timeout can range from 5 seconds to 20 seconds. |
| **Step 11** | (Optional) UCS-A /org/storage-profile/security/drive-security/remote* # **create login** | Creates the login details for the KMIP server and enters the login mode. |
| **Step 12** | (Optional) UCS-A /org/storage-profile/security/drive-security/remote/login* # **set username** *username* | Sets the username to log into the KMIP server. |
| **Step 13** | (Optional) UCS-A /org/storage-profile/security/drive-security/remote/login* # **set password** *password* | Sets the password to log into the KMIP server. |
| **Step 14** | UCS-A /org/storage-profile/security/drive-security/remote/login* # **commit-buffer** | Commits the transaction to the system configuration. |

## Example

```
UCS-A # scope org
UCS-A /org # scope storage-profile stp-demo
UCS-A /org/storage-profile # create security
UCS-A /org/storage-profile/security* # create drive-security
UCS-A /org/storage-profile/security/drive-security* # create remote
UCS-A /org/storage-profile/security/drive-security/remote* # set primary-server 10.10.10.1
UCS-A /org/storage-profile/security/drive-security/remote* # set secondary-server 10.10.10.2
UCS-A /org/storage-profile/security/drive-security/remote* # set port 5696
UCS-A /org/storage-profile/security/drive-security/remote* # set server-certificate
Enter lines one at a time. Enter ENDOFBUF to finish. Press ^C to abort.
Prompt Certificate:
>-----BEGIN CERTIFICATE-----
    MIIEEDCCAvigAwIBAgIGALOfZVDsMA0GCSqGSIb3DQEBCwUAMIGQMSowKAYDVQQD
    EyFDRyBDQSBTIG9uIHZvcm1ldHJpY2RzbS5jaXNjby5jb20xFTATBgNVBAsTDFNh
    dmJ1U3RvcmRldjEWMBQGA1UEChMNQ2lzY28gU3lzdGVtczERMA8GA1UEBxMIU2Fu
    IEpvc2UxEzARBgNVBAgTCkNhbGlmb3JuaWExCzAJBgNVBAYTAlVTMB4XDTE2MDkw
    NzE5MzMwMVoXDTI2MDkwOTE5MzMwMVowgZAxKjAoBgNVBAMTIUNHIENBIFMgb24g
    dm9ybWV0cmljZHNtLmNpc2NvLmNvbTEVMBMGA1UECxMMU2F2YnVTdG9yZGV2MRYw
    FAYDVQQKEw1DaXNjbyBTeXN0ZW1zMREwDwYDVQQHEwhTYW4gSm9zZTETMB>EGA1UE
    CBMKQ2FsaWZvcm5pYTELMAkGA1UEBhMCVVMwggEiMA0GCSqGSIb3DQEBAQUAA4IB
    DwAwggEKAoIBAQDhX2UdIV>TQTchGo1FjAc5u1W9zAo/YkjD22ANpbEPiAmgWL97c
    Xwj7yzArflrZ2kWvQCm4f6AdLOFUWzbuo+Fxd3rurd>w6BhJXdLj8Piq8094PqCLp
    qdUF83SsRVVbCXHxOqdk9jsSQrvTcV4PloNrelMLq/mOqsaODs+us4ng7sMDtGXv
    LeKFC8DUEm0GlGQACwiJ3s9O4+P2CI/d4P/>EyWwqABf3YJmAI1EQyUnoTwrg6EgY
    ZvcpHsmjXnbBZrL+ON7FBcbrTanvjyJxE6tFf5cRPGhymfna7Fd3lfVwZCcGIoR+
>   EOIAwgetzIRM6FzMiV2/tDT8STo/oo5Tg3dDAgMBAAGjbj>BsMBIGA1UdEwEB/wQI
    MAYBAf8CAQAwDgYDVR0PAQH/>>>>>>>>>BAQDAgEGMB0GA1UdDgQWBBRnYyFiAK2lEDZJNC0Y
    VlIqMgiUJDAnBgNVHSMEIDAegBRnYyFiAK2lEDZJNC0YVlIqMgiUJIIGALOfZVDs
    MA0GCSqGSIb3DQEBCwUAA4IBAQAfhB2+Ft8V2ELAFa7PcG/rU09ux7LYcCjt3STa
    mzKdZ7Rn5COvknKrJX+EefT7xl03CQXT9aeSAddQUOCy8fhiPoaMFrlTgs1hdS0p
    NJvfxV6QCun2UMRSuxWfG>0QFfofnXeIGkAmEYOpUdArSOTbtt4v6Lja1A+KEsvWW
    5KaVemo2nsd+iD0IPCOhpShAgaAwpnYUq9mLfVgvV07Z+hmkuOIQTZ2+h+pJQtE0
    +U5qaTts4pMXpqQPjlid0NMuaPug1SpSD7KBsjwR1SzehzPdns16uprmvWa3VBk3
    OK6y55FoIu+Wg9i/8kmfkghyGwTfo6weEKbleuVwupvpriMF>
    -----END CERTIFICATE-----

UCS-A /org/storage-profile/security/drive-security/remote* # create login
UCS-A /org/storage-profile/security/drive-security/remote/login* # set username user1
UCS-A /org/storage-profile/security/drive-security/remote/login* # set password Password
UCS-A /org/storage-profile/security/drive-security/remote/login* # exit
UCS-A /org/storage-profile/security/drive-security/remote # exit
UCS-A /org/storage-profile/security/drive-security # show detail expand

Drive Security:
    Remote:
        Primary Server Name: 10.10.10.1
        Secondary Server Name:10.10.10.2
        KMIP Server Port: 5696
        Deployed Security Key:
        KMIP Server Certificate: -----BEGIN CERTIFICATE-----
    MIIEEDCCAvigAwIBAgIGALOfZVDsMA0GCSqGSIb3DQEBCwUAMIGQMSowKAYDVQQD
    EyFDRyBDQSBTIG9uIHZvcm1ldHJpY2RzbS5jaXNjby5jb20xFTATBgNVBAsTDFNh
    dmJ1U3RvcmRldjEWMBQGA1UEChMNQ2lzY28gU3lzdGVtczERMA8GA1UEBxMIU2Fu
    IEpvc2UxEzARBgNVBAgTCkNhbGlmb3JuaWExCzAJBgNVBAYTAlVTMB4XDTE2MDkw
    NzE5MzMwMVoXDTI2MDkwOTE5MzMwMVowgZAxKjAoBgNVBAMTIUNHIENBIFMgb24g
    dm9ybWV0cmljZHNtLmNpc2NvLmNvbTEVMBMGA1UECxMMU2F2YnVTdG9yZGV2MRYw
    FAYDVQQKEw1DaXNjbyBTeXN0ZW1zMREwDwYDVQQHEwhTYW4gSm9zZTETMBEGA1UE
    CBMKQ2FsaWZvcm5pYTELMAkGA1UEBhMCVVMwggEiMA0GCSqGSIb3DQEBAQUAA4IB
    DwAwggEKAoIBAQDhX2UdIVTQTchGo1FjAc5u1W9zAo/YkjD22ANpbEPiAmgWL97c
```

```
Xwj7yzArflrZ2kWvQCm4f6AdLOFUWzbuo+Fxd3rurdw6BhJXdLj8Piq8094PqCLp
qdUF83SsRVVbCXHxOqdk9jsSQrvTcV4PloNrelMLq/mOqsaODs+us4ng7sMDtGXv
LeKFC8DUEm0GlGQACwiJ3s9O4+P2CI/d4P/EyWwqABf3YJmAI1EQyUnoTwrg6EgY
ZvcpHsmjXnbBZrL+ON7FBcbrTanvjyJxE6tFf5cRPGhymfna7Fd3lfVwZCcGIoR+
EOIAwgetzIRM6FzMiV2/tDT8STo/oo5Tg3dDAgMBAAGjbjBsMBIGA1UdEwEB/wQI
MAYBAf8CAQAwDgYDVR0PAQH/BAQDAgEGMB0GA1UdDgQWBBRnYyFiAK2lEDZJNC0Y
VlIqMgiUJDAnBgNVHSMEIDAegBRnYyFiAK2lEDZJNC0YVlIqMgiUJIIGALOfZVDs
MA0GCSqGSIb3DQEBCwUAA4IBAQAfhB2+Ft8V2ELAFa7PcG/rU09ux7LYcCjt3STa
mzKdZ7Rn5COvknKrJX+EefT7xl03CQXT9aeSAddQUOCy8fhiPoaMFrlTgs1hdS0p
NJvfxV6QCun2UMRSuxWfG0QFfofnXeIGkAmEYOpUdArSOTbtt4v6Lja1A+KEsvWW
5KaVemo2nsd+iD0IPCOhpShAgaAwpnYUq9mLfVgvV07Z+hmkuOIQTZ2+h+pJQtE0
+U5qaTts4pMXpqQPjlid0NMuaPug1SpSD7KBsjwR1SzehzPdns16uprmvWa3VBk3
OK6y55FoIu+Wg9i/8kmfkghyGwTfo6weEKbleuVwupvpriMF
-----END CERTIFICATE-----
```

# Modifying a Remote Security Key

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | UCS-A # **scope server** *server-id* | Enters the server mode for the specified server. |
| **Step 2** | UCS-A /server # **scope raid-controller** *raid-controller-id {SAS \| SAT}* | Enters the RAID controller mode. Currently, Cisco UCS Manager supports SED only on SAS controllers. |
| **Step 3** | UCS-A /server/raid-controller # **set admin-state modify-remote-key** | Modifies the security key of a remote security policy. |
| **Step 4** | UCS-A /server/raid-controller # **commit-buffer** | Commits the transaction to the system configuration. |

### Example

This example shows how to modify the remote security key on the controller for a rack-mount server:

```
UCS-A# scope server 3
UCS-A /server # scope raid-controller 1 sas
UCS-A /server/raid-controller # set admin-state modify-remote-key
UCS-A /server/raid-controller* # commit-buffer
UCS-A /server/raid-controller #
```

This example shows how to modify the remote security key on the controller for a blade server:

```
UCS-A# scope server 1/3
UCS-A chassis/server # scope raid-controller 1 sas
UCS-A chassis/server/raid-controller # set admin-state modify-remote-key
UCS-A chassis/server/raid-controller* # commit-buffer
UCS-A chassis/server/raid-controller #
```

# Modifying the Security Policy from Remote to Local

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | UCS-A # **scope org** | Enters the root organization mode. |
| **Step 2** | UCS-A /org # **scope storage-profile** *storage-profile-name* | Enters the storage-profile configuration mode for the specified storage profile. |
| **Step 3** | UCS-A /org/storage-profile # **scope security** | Enters the security policy mode for the specified storage profile. |
| **Step 4** | UCS-A /org/storage-profile/security # **scope drive-security** | Enters the drive security policy mode for the specified storage profile security. |
| **Step 5** | UCS-A /org/storage-profile/security/drive-security # **delete remote** | Deletes the existing remote security policy. |
| **Step 6** | UCS-A /org/storage-profile/security/drive-security* # **commit-buffer** | Commits the transaction to the system configuration. |
| **Step 7** | UCS-A /org/storage-profile/security/drive-security # **create local** | Creates and enters the local policy mode. |
| **Step 8** | UCS-A /org/storage-profile/security/drive-security/local* # **set security-key** *security-key* | Sets the security key for the local policy. |
| **Step 9** | UCS-A /org/storage-profile/security/drive-security/local* # **commit-buffer** | Commits the transaction to the system configuration |
| **Step 10** | | |

### Example

This example shows how to modify a security policy from remote to local:

```
UCS-A# scope org
UCS-A /org # scope storage-profile stp-demo
UCS-A /org/storage-profile # scope security
UCS-A /org/storage-profile/security # scope drive-security
UCS-A /org/storage-profile/security/drive-security # delete remote
UCS-A /org/storage-profile/security/drive-security* # commit-buffer
UCS-A /org/storage-profile/security/drive-security # create local
UCS-A /org/storage-profile/security/drive-security/local* # set security-key
thereare32charactersinthisseckey
UCS-A /org/storage-profile/security/drive-security/local* # commit-buffer
UCS-A /org/storage-profile/security/drive-security/local #
```

# Inserting a Secured Disk into a Server with a Remote Security Policy

When you insert a secured disk into a server with a remote security policy, the storage disk will appear as a locked disk. Do one of the following:

- Unlock the disk manually with the local key if the disk was previously locked using the local key.
- Unlock using the remote KMIP server.

When you move a secured disk from a server with a local security policy to a server with a remote security policy, the disk will come up as locked. Unlock the disk manually with the local key.

# Securing an Existing Virtual Drive

### Before you begin

- The controller must be secure.
- The virtual drive must be in the **Orphaned** state.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | UCS-A# **scope server** *server-id* | Enters the server mode for the specified server. |
| **Step 2** | UCS-A /server# **scope raid-controller** *raid-controller-id {SAS \| SAT}* | Enters the RAID controller mode.<br>Currently, Cisco UCS Manager supports SED only on SAS controllers. |
| **Step 3** | UCS-A /server/raid-controller# **scope virtual-drive** *virtual-drive-id* | Enters the virtual drive mode for the specified orphaned virtual drive. |
| **Step 4** | UCS-A /server/raid-controller/virtual-drive# **set admin-state secure-drive-group** | Secures the existing virtual drive. |
| **Step 5** | UCS-A /server/raid-controller/virtual-drive*# **commit-buffer** | Commits the transaction to the system configuration. |

### Example

This example shows how to secure an existing virtual drive for a rack-mount server:

```
UCS-A# scope server 1
UCS-A /server# scope raid-controller 3 sas
UCS-A /server/raid-controller# scope virtual-drive 1000
UCS-A /server/raid-controller/virtual-drive # set admin-state secure-drive-group
UCS-A /server/raid-controller/virtual-drive*# commit-buffer
UCS-A /server/raid-controller/virtual-drive#
```

This example shows how to secure an existing virtual drive for a blade server:

```
UCS-A# scope server 1/4
UCS-A chassis/server# scope raid-controller 3 sas
UCS-A chassis/server/raid-controller# scope virtual-drive 1000
UCS-A chassis/server/raid-controller/virtual-drive # set admin-state secure-drive-group
UCS-A chassis/server/raid-controller/virtual-drive*# commit-buffer
UCS-A chassis/server/raid-controller/virtual-drive#
```

# Enabling Security on a Disk

### Before you begin

Ensure that the disk is a JBOD.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | UCS-A# **scope server** *server-id* | Enters the server mode for the specified server. |
| **Step 2** | UCS-A /server # **scope raid-controller** *raid-controller-id {SAS \| SAT}* | Enters the RAID controller mode.<br>Currently, Cisco UCS Manager supports SED only on SAS controllers. |
| **Step 3** | UCS-A /server/raid-controller # **scope local-disk** *local-disk-id* | Enters the local disk configuration mode |
| **Step 4** | UCS-A /server/raid-controller/local-disk # **set admin-state enable-security** | Enables security on a JBOD. |
| **Step 5** | UCS-A /server/raid-controller/local-disk* # **commit-buffer** | Commits the transaction to the system configuration. |

### Example

The example shows how to enable security on a JBOD for a rack-mount server:

```
UCS-A# scope server 1
UCS-A /server # scope raid-controller 3 sas
UCS-A /server/raid-controller # scope local-disk 2
UCS-A /server/raid-controller/local-disk # set admin-state enable-security
UCS-A /server/raid-controller/local-disk* # commit-buffer
UCS-A /server/raid-controller/local-disk #
```

The example shows how to enable security on a JBOD for a blade server:

```
UCS-A# scope server 1/3
UCS-A chassis/server # scope raid-controller 3 sas
```

```
UCS-A chassis/server/raid-controller # scope local-disk 2
UCS-A chassis/server/raid-controller/local-disk # set admin-state enable-security
UCS-A chassis/server/raid-controller/local-disk* # commit-buffer
UCS-A chassis/server/raid-controller/local-disk #
```

# Erasing a Secure Disk

### Before you begin

Ensure that the disk is in the **Unconfigured Good** state.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | UCS-A # **scope server** *server-id* | Enters the server mode for the specified server. |
| **Step 2** | UCS-A /server # **scope raid-controller** *raid-controller-id {SAS \| SAT}* | Enters the RAID controller mode.<br>Currently, Cisco UCS Manager supports SED only on SAS controllers. |
| **Step 3** | UCS-A /server/raid-controller # **scope local-disk** *local-disk-id* | Enters the local disk configuration mode. |
| **Step 4** | UCS-A /server/raid-controller/local-disk # **set admin-state clear secure-drive** | Erases the secured disk and clears the security on the disk. |
| **Step 5** | UCS-A /server/raid-controller/local-disk* # **commit-buffer** | Commits the transaction to the system configuration. |

### Example

This example shows how to erase a secure disk on a rack-mount server:

```
UCS-A # scope server 1
UCS-A /server # scope raid-controller 3 sas
UCS-A /server/raid-controller # scope local-disk 2
UCS-A /server/raid-controller/local-disk # set admin-state clear secure-drive
UCS-A /server/raid-controller/local-disk* # commit-buffer
UCS-A /server/raid-controller/local-disk #
```

This example shows how to erase a secure disk on a blade server:

```
UCS-A # scope server 1/3
UCS-A chassis/server # scope raid-controller 3 sas
UCS-A chassis/server/raid-controller # scope local-disk 2
UCS-A chassis/server/raid-controller/local-disk # set admin-state clear secure-drive
UCS-A chassis/server/raid-controller/local-disk* # commit-buffer
UCS-A chassis/server/raid-controller/local-disk #
```

# Disabling Security on a Controller

### Before you begin

You can disable security only on SAS controllers. To disable security on a controller, you must first disable security on all the secure disks and delete all the secure virtual drives under the controller.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | UCS-A # **scope server** *server-id* | Enters the server mode for the specified server. |
| **Step 2** | UCS-A /server # **scope raid-controller** *raid-controller-id {SAS \| SAT}* | Enters the RAID controller mode. Currently, Cisco UCS Manager supports SED only on SAS controllers. |
| **Step 3** | UCS-A /server/raid-controller # **set admin-state disable-security** | Disables security key on the controller. |
| **Step 4** | UCS-A /server/raid-controller # **commit-buffer** | Commits the transaction to the system configuration. |

### Example

This example shows how to disable security on the controller for a rack-mount server:

```
UCS-A# scope server 1
UCS-A /server # scope raid-controller 3 sas
UCS-A /server/raid-controller # set admin-state disable-security
UCS-A /server/raid-controller* # commit-buffer
UCS-A /server/raid-controller #
```

This example shows how to disable security on the controller for a blade server:

```
UCS-A# scope server 1/3
UCS-A chassis/server # scope raid-controller 3 sas
UCS-A chassis/server/raid-controller # set admin-state disable-security
UCS-A chassis/server/raid-controller* # commit-buffer
UCS-A chassis/server/raid-controller #
```

# Unlocking a Locked Disk

When the key of an SED does not match the key on the controller, it shows the disk as Locked, Foreign Secure. You must unlock the disks either by providing the security-key for that disk, or by using the remote KMIP server. After unlocking the disk, import or clear the foreign configuration.

After you unlock a locked disk, the security status of the disk will show as Foreign Secure.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | UCS-A # **scope server** *server-id* | Enters server mode for the specified server. |
| **Step 2** | UCS-A /server # **scope raid-controller** *raid-controller-id {SAS \| SAT}* | Enters the RAID controller mode. Currently, Cisco UCS Manager supports SEDs only on SAS controllers. |
| **Step 3** | UCS-A /server/raid-controller # **set admin-state unlock-disk** [*security-key*] | Unlocks the locked disks.<br>If the security-key is set, this key is used to unlock disks that are in the locked state.<br>If the security-key is not set, Cisco UCS Manager tries to unlock the disks by using the KMIP server. Setting the security-key is optional only if remote security is configured on the server. |
| **Step 4** | UCS-A /server/raid-controller* # **commit-buffer** | Commits the transaction to the system configuration. |

### Example

This example shows how to unlock a locked disk on a rack-mount server with a local security policy by using a security-key:

```
UCS-A # scope server 1
UCS-A /server # scope raid-controller 3 sas
UCS-A /server/raid-controller # set admin-state unlock-disk thisisastring
UCS-A /server/raid-controller* # commit-buffer
UCS-A /server/raid-controller #
```

This example shows how to unlock a locked disk on a rack-mount server with a remote security policy by using the KMIP server:

```
UCS-A # scope server 1
UCS-A /server # scope raid-controller 3 sas
UCS-A /server/raid-controller # set admin-state unlock-disk
UCS-A /server/raid-controller* # commit-buffer
UCS-A /server/raid-controller #
```

This example shows how to unlock a locked disk on a blade server with a local security policy by using a security-key:

```
UCS-A # scope server 1/2
UCS-A chassis/server # scope raid-controller 3 sas
UCS-A chassis/server/raid-controller # set admin-state unlock-disk thisisastring
UCS-A chassis/server/raid-controller* # commit-buffer
UCS-A chassis/server/raid-controller #
```

This example shows how to unlock a locked disk on a blade server with a remote security policy by using the KMIP server:

```
UCS-A # scope server 1/2
UCS-A chassis/server # scope raid-controller 3 sas
UCS-A chassis/server/raid-controller # set admin-state unlock-disk
UCS-A chassis/server/raid-controller* # commit-buffer
UCS-A chassis/server/raid-controller #
```

# Erasing a Secure Foreign Configuration Disk

You can erase a secure foreign configuration disk when you have a disk in locked state and you want to use the disk without accessing the existing data.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | UCS-A # **scope server** *server-id* | Enters the server mode for the specified server. |
| **Step 2** | UCS-A /server # **scope raid-controller** *raid-controller-id {SAS \| SAT}* | Enters the RAID controller mode.<br>Currently, Cisco UCS Manager supports SED only on SAS controllers. |
| **Step 3** | UCS-A /server/raid-controller # **scope local-disk** *local-disk-id* | Enters the local disk configuration mode. |
| **Step 4** | UCS-A /server/raid-controller/local-disk # **set admin-state clear secure-foreign-config-drive** | Clears the secure foreign configuration drive. |
| **Step 5** | UCS-A /server/raid-controller/local-disk* # **commit-buffer** | Commits the transaction to the system configuration. |

### Example

This example shows how to clear a foreign configuration disk on a rack-mount server:

```
UCS-A# scope server 1
UCS-A /server # scope raid-controller 3 sas
UCS-A /server/raid-controller # scope local-disk 2
UCS-A /server/raid-controller/local-disk # set admin-state clear secure-foreign-config-drive

UCS-A /server/raid-controller/local-disk* # commit-buffer
UCS-A /server/raid-controller/local-disk #
```

This example shows how to clear a foreign configuration disk on a blade server:

```
UCS-A# scope server 1/3
UCS-A chassis/server # scope raid-controller 3 sas
```

```
UCS-A chassis/server/raid-controller # scope local-disk 2
UCS-A chassis/server/raid-controller/local-disk # set admin-state clear
secure-foreign-config-drive
UCS-A chassis/server/raid-controller/local-disk* # commit-buffer
UCS-A chassis/server/raid-controller/local-disk #
```

# Displaying the Security Flags of a Controller

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | UCS-A # **scope server** *server-id* | Enters the server mode for the specified server. |
| **Step 2** | UCS-A /server # **scope raid-controller** *raid-controller-id {SAS \| SAT}* | Enters the RAID controller mode.<br>Currently, Cisco UCS Manager supports SED only on SAS controllers. |
| **Step 3** | UCS-A /server/raid-controller # **show detail** | Displays details of the RAID controller. |

### Example

This example shows to how to check if the security flag of controller is enabled on a rack-mount server:

```
UCS-A # scope server 1
UCS-A /server # scope raid-controller 3 sas
UCS-A /server/raid-controller # show detail

RAID Controller:
    ID: 3
    Type: SAS
    PCI Addr: 03:00.0
    Vendor: LSI Corp.
    Model: LSI MegaRAID SAS 3108
    Serial: SV55346948
    HW Rev: C0
    Raid Support: RAID0, RAID1, RAID5, RAID6, RAID10, RAID50, RAID60
    OOB Interface Supported: Yes
    Mode: RAID
    Rebuild Rate: 30
    Controller Status: Optimal
    Config State: Applied
    Pinned Cache Status: Disabled
    Sub OEM ID: 0
    Supported Strip Sizes: 1MB,64KB,256KB,512KB,128KB
    Default Strip Size: 64KB
    PCI Slot: HBA
    Controller Flags: Drive Security Capable
```

This example shows to how to check if the security flag of controller is enabled on a blade server:

```
UCS-A # scope server 1/2
UCS-A chassis/server # scope raid-controller 3 sas
UCS-A chassis/server/raid-controller # show detail

RAID Controller:
    ID: 3
    Type: SAS
    PCI Addr: 03:00.0
    Vendor: LSI Corp.
    Model: LSI MegaRAID SAS 3108
    Serial: SV55346948
    HW Rev: C0
    Raid Support: RAID0, RAID1, RAID5, RAID6, RAID10, RAID50, RAID60
    OOB Interface Supported: Yes
    Mode: RAID
    Rebuild Rate: 30
    Controller Status: Optimal
    Config State: Applied
    Pinned Cache Status: Disabled
    Sub OEM ID: 0
    Supported Strip Sizes: 1MB,64KB,256KB,512KB,128KB
    Default Strip Size: 64KB
    PCI Slot: HBA
    Controller Flags: Drive Security Capable
```

# Displaying the Security Flags of a Local Disk

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | UCS-A # **scope server** *server-id* | Enters the server mode for the specified server. |
| **Step 2** | UCS-A /server # **scope raid-controller** *raid-controller-id {SAS \| SAT}* | Enters the RAID controller mode.<br>Currently, Cisco UCS Manager supports SED only on SAS controllers. |
| **Step 3** | UCS-A /server/raid-controller # **scope local-disk** *local-disk-id* | Enters the local disk configuration mode. |
| **Step 4** | UCS-A /server/raid-controller/local-disk # **show detail** | Displays details of the local disk. |

### Example

This example shows to how to display the security flag of a local disk on a rack-mount server:

```
UCS-A # scope server 1
UCS-A /server # scope raid-controller 3 sas
UCS-A /server/raid-controller #scope local-disk 2
UCS-A /server/raid-controller/local-disk # show detail

Local Disk:
```

```
    ID: 4
    Block Size: 512
    Physical Block Size: 4096
    Blocks: 1560545280
    Raw Size: 763097
    Size: 761985
    Technology: SSD
    Operability: Operable
    Oper Qualifier Reason: N/A
    Presence: Equipped
    Connection Protocol: SAS
    Product Variant: default
    Product Name: 800GB Enterprise performance SAS SED SSD (10 FWPD) - MTFDJAK800MBS
    PID: UCS-SD800GBEK9
    VID: V01
    Vendor: MICRON
    Model: S650DC-800FIPS
    Vendor Description: Micron
    Serial: ZAZ090VD000082215OZ3
    HW Rev: 0
    Running-Vers: MB13
    Average Seek Time (R/W): N/A
    Track to Track Seek Time (R/W): 115ms
    Part Number: 16-100911-01
    SKU: UCS-SD800GBEK9
    Drive State: Online
    Power State: Active
    Link Speed: 12 Gbps
    Enclosure Association Type: Direct Attached
    Device Version: MB13
    Drive Security Flags: Secured,Security Enabled,Security Capable
```

This example shows to how to display the security flag of a local disk on a blade server:

```
UCS-A # scope server 1/2
UCS-A chassis/server # scope raid-controller 3 sas
UCS-A chassis/server/raid-controller #scope local-disk 2
UCS-A chassis/server/raid-controller/local-disk # show detail

Local Disk:
    ID: 4
    Block Size: 512
    Physical Block Size: 4096
    Blocks: 1560545280
    Raw Size: 763097
    Size: 761985
    Technology: SSD
    Operability: Operable
    Oper Qualifier Reason: N/A
    Presence: Equipped
    Connection Protocol: SAS
    Product Variant: default
    Product Name: 800GB Enterprise performance SAS SED SSD (10 FWPD) - MTFDJAK800MBS
    PID: UCS-SD800GBEK9
    VID: V01
    Vendor: MICRON
    Model: S650DC-800FIPS
    Vendor Description: Micron
    Serial: ZAZ090VD000082215OZ3
    HW Rev: 0
    Running-Vers: MB13
```

```
    Average Seek Time (R/W): N/A
    Track to Track Seek Time (R/W): 115ms
    Part Number: 16-100911-01
    SKU: UCS-SD800GBEK9
    Drive State: Online
    Power State: Active
    Link Speed: 12 Gbps
    Enclosure Association Type: Direct Attached
    Device Version: MB13
    Drive Security Flags: Secured,Security Enabled,Security Capable
```

# Displaying the Security Flags of a Virtual Drive

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | UCS-A # **scope server** *server-id* | Enters the server mode for the specified server. |
| **Step 2** | UCS-A /server # **scope raid-controller** *raid-controller-id {SAS \| SAT}* | Enters the RAID controller mode.<br>Currently, Cisco UCS Manager supports SED only on SAS controllers. |
| **Step 3** | UCS-A /server/raid-controller # **scope virtual-drive** *virtual-drive-id* | Enters the virtual drive mode. |
| **Step 4** | UCS-A /server/raid-controller/virtual-drive # **show detail** | Displays details of the virtual drive. |

### Example

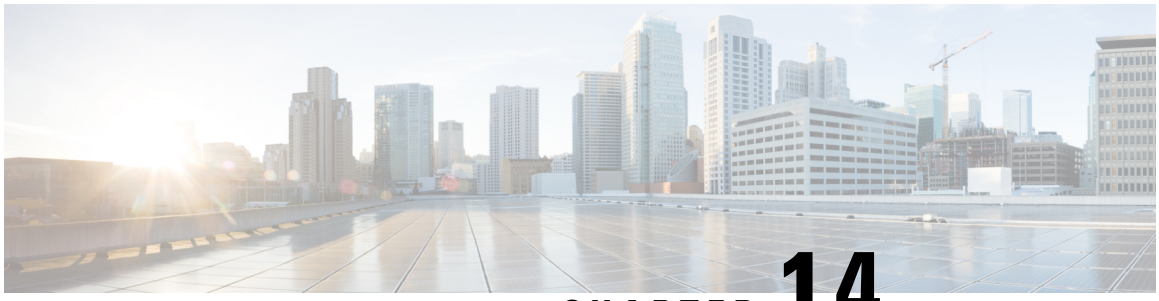
This example shows to how to display the security flag of a virtual disk for a rack-mount server:

```
UCS-A # scope server 1
UCS-A /server # scope raid-controller 3 sas
UCS-A /server/raid-controller # scope virtual-drive 1000
UCS-A /server/raid-controller/virtual-drive # show detail

Virtual Drive:
    ID: 1000
    Name: luna
    Block Size: 512
    Blocks: 20971520
    Size: 10240
    Operability: Operable
    Presence: Equipped
    Lifecycle: Allocated
    Drive State: Optimal
    Type: RAID 0 Striped
    Strip Size (KB): 64
    Access Policy: Read Write
    Read Policy: Normal
    Configured Write Cache Policy: Write Through
    Actual Write Cache Policy: Write Through
```

```
    IO Policy: Direct
    Drive Cache: No Change
    Bootable: False
    Oper Device ID: 0
    Change Qualifier: No Change
    Config State: Applied
    Deploy Action: No Action
    Service Profile Lun Reference: org-root/ls-sp1/vdrive-ref-lun-1
    Assigned To Server: sys/rack-unit-1
    Available Size on Disk Group (MB): 751745
    Unique Identifier: 90ae6ea0-6a39-49e1-9c0d-0f3e2e9ecfce
    Vendor Unique Identifier: 678da6e7-15b2-9c20-2011-c4f60c40e57a
    Security Flags: Drive Security Enable,Drive Security Capable
```

This example shows to how to display the security flag of a virtual disk for a blade server:

```
UCS-A # scope server 1/2
UCS-A chassis/server # scope raid-controller 3 sas
UCS-A chassis/server/raid-controller # scope virtual-drive 1000
UCS-A chassis/server/raid-controller/virtual-drive # show detail

Virtual Drive:
    ID: 1000
    Name: luna
    Block Size: 512
    Blocks: 20971520
    Size: 10240
    Operability: Operable
    Presence: Equipped
    Lifecycle: Allocated
    Drive State: Optimal
    Type: RAID 0 Striped
    Strip Size (KB): 64
    Access Policy: Read Write
    Read Policy: Normal
    Configured Write Cache Policy: Write Through
    Actual Write Cache Policy: Write Through
    IO Policy: Direct
    Drive Cache: No Change
    Bootable: False
    Oper Device ID: 0
    Change Qualifier: No Change
    Config State: Applied
    Deploy Action: No Action
    Service Profile Lun Reference: org-root/ls-sp1/vdrive-ref-lun-1
    Assigned To Server: sys/rack-unit-1
    Available Size on Disk Group (MB): 751745
    Unique Identifier: 90ae6ea0-6a39-49e1-9c0d-0f3e2e9ecfce
    Vendor Unique Identifier: 678da6e7-15b2-9c20-2011-c4f60c40e57a
    Security Flags: Drive Security Enable,Drive Security Capable
```



CHAPTER 14

# Storage Inventory



# NVMe-optimized M5 Servers

Beginning with 3.2(3a), Cisco UCS Manager supports the following NVMe-optimized M5 servers:

- UCSC-C220-M5SN—The PCIe MSwitch is placed in the dedicated MRAID slot for UCS C220 M5 servers. This setup supports up to 10 NVMe drives. The first two drives are direct-attached through the riser. The remaining eight drives are connected and managed by the MSwitch. This setup does not support any SAS/SATA drive combinations.
- UCSC-C240-M5SN—The PCIe MSwitch is placed in the riser-2 at slot-4 for UCS C240 M5 servers. The servers support up to 24 drives. Slots 1-8 are the NVMe drives connected and managed by the MSwitch. The servers also support up to two NVMe drives in the rear and are direct-attached through the riser. This setup supports SAS/SATA combination with the SAS/SATA drives from slots 9-24. These drives are managed by the SAS controller placed in the dedicated MRAID PCIe slot.
- UCS-C480-M5—UCS C480 M5 servers support up to three front NVMe drive cages, each supporting up to eight NVMe drives. Each cage has an interposer card, which contains the MSwitch. Each server can support up to 24 NVMe drives (3 NVMe drive cages x 8 NVMe drives). The servers also support a rear PCIe Aux drive cage, which can contain up to eight NVMe drives managed by an MSwitch placed in PCIe slot-10.

  This setup does not support:

  - a combination of NVMe drive cages and HDD drive cages
  - a combination of the Cisco 12G 9460-8i RAID controller and NVMe drive cages, irrespective of the rear Auxiliary drive cage

**Note** The UCS C480 M5 PID remains same as in earlier release.

**Note** On B200 and B480 M5 blade servers, NVMe drives cannot be used directly with SAS controllers. Use an LSTOR-PT pass-through controller instead.

The following MSwitch cards are supported in NVMe optimized M5 servers:

- UCS-C480-M5 HDD Ext NVMe Card (UCSC-C480-8NVME)—Front NVMe drive cage with an attached interposer card containing the PCIe MSwitch. Each server supports up to three front NVMe drive cages and each cage supports up to 8 NVMe drives. Each server can support up to 24 NVMe drives (3 NVMe drive cages x 8 NVMe drives).
- UCS-C480-M5 PCIe NVMe Switch Card (UCSC-NVME-SC)—PCIe MSwitch card to support up to eight NVMe drives in the rear auxiliary drive cage inserted in PCIe slot 10.

**Note** Cisco UCS-C480-M5 servers support a maximum of 32 NVMe drives (24 NVMe drives in the front + 8 NVMe drives in the rear auxiliary drive cage)

- UCSC-C220-M5SN and UCSC-C240-M5SN do not have separate MSwitch PIDs. MSwitch cards for these servers are part of the corresponding NVMe optimized server.

**Note** The UCS Manager does not receive any missing details on fault or alert during NVMe drive pull. It is applicable to NVMe drives behind the passthrough and the storage controller that are passthroughs for the NVMe drives.

# MSwitch Disaster Recovery

You can recover a corrupted MSwitch and roll back to a previous working firmware.

**Note** If you have a setup with Cisco UCS C480 M5 Server, then MSwitch disaster recovery process can be performed only on one MSwitch at a time. If the disaster recovery process is already running for one MSwitch, then wait for it to complete. You can monitor the recovery status from FSM.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | UCS-A# **scope server** [*chassis-num*/*server-num* \| *dynamic-uuid*] | Enters server mode for the specified server. |
| **Step 2** | UCS-A /server # **scope nvme-swtich** *nvme_switch* | Enters the specified NVMe swtich. |
| **Step 3** | UCS-A /server/nvme-switch # **set recover-nvme-switch** | Deletes the LUN Set with the specified name. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 4** | UCS-A /server/nvme-switch* # **commit-buffer** | Commits the transaction to the system configuration. |
| **Step 5** | UCS-A /server/nvme-switch # **exit** | Exits the MSwitch mode. |
| **Step 6** | UCS-A /server # **ack-nvme-switch-recovery acknowledge** | Acknowledges the MSwitch recovery. |
| **Step 7** | UCS-A /server* # **commit-buffer** | Commits the transaction to the system configuration.<br>**Note**<br>Do not reset the server during the disaster recovery process. |

### Example

The following example recovers the MSwitch on server1:

```
UCS-A# scope server 1
UCS-A/server # scope nvme-switch 1
UCS-A/server/nvme-switch # set recover-nvme-switch
UCS-A/server/nvme-switch* # commit-buffer
UCS-A/server/nvme-switch # exit
UCS-A/server # ack-nvme-switch-recovery acknowledge
UCS-A/server* # commit-buffer
```

# NVMe Replacement Considerations for B-Series M6 and X-Series Servers

Swapping or replacing NVMe storage devices on any of the below mentioned servers while the system is powered off can result in an error condition:

- Cisco UCS B200 M6 Server
- Cisco UCS X210C M6 Compute Node
- Cisco UCS X210c M7 Compute Node
- Cisco UCS X410c M7 Compute Node
- Cisco UCS X215c M8 Compute Node
- Cisco UCS X210c M8 Compute Node

To avoid encountering this error, use the following precautions:

- Replace or hot-swap NVMe SSD storage devices without powering off the server.

- If it is necessary to replace NVMe storage with the server powered off, decommission the server and remove or replace the hardware, then reboot the server. This will recommission the server and NVMe storage will be correctly discovered.

If NVMe storage is replaced when the system is powered off, the controller will be marked as unresponsive. To recover from this condition, re-acknowledge the server.

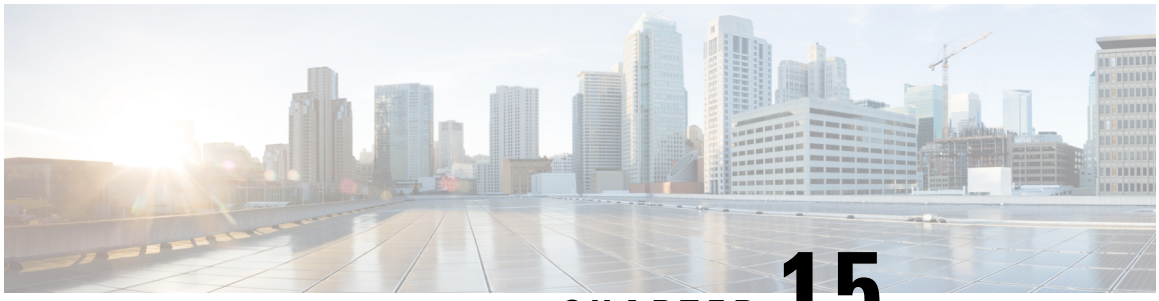
# Volume Management Device (VMD) Setup

The Intel® Volume Management Device (VMD) is a tool that provides NVMe drivers to manage PCIe Solid State Drives attached to VMD-enabled domains. This includes Surprise hot-plug of PCIe drives and configuring blinking patterns to report status. PCIe Solid State Drive (SSD) storage lacks a standardized method to blink LEDs to represent the status of the device. With VMD, you can control LED indicators on both direct attached and switch attached PCIe storage using a simple command-line tool.

To use VMD, you must first enable VMD through a UCS Manager BIOS policy and set the UEFI boot options. Enabling VMD provides Surprise hot plug and optional LED status management for PCIe SSD storage that is attached to the root port. VMD Passthrough mode provides the ability to manage drives on guest VMs.

Enabling VMD also allows configuration of Intel® Virtual RAID on CPU (VRoC), a hybrid RAID architecture on Intel® Xeon® Scalable Processors. Documentation on the use and configuration of VRoC can be found at the Intel website.

**IMPORTANT**: VMD must be enabled in the UCS Manager BIOS settings before Operating System install. If enabled after OS installation, the server will fail to boot. This restriction applies to both standard VMD and VMD Passthrough. Likewise, once enabled, you cannot disable VMD without a loss of system function.



CHAPTER 15

# Drive Diagnostics



## Overview of Drive Diagnostics

Beginning from release 4.2(2a), Drive Diagnostics feature supports running diagnostics on HDD/SSD and SAS/SATA drive types. This feature allows you to determine the device health by obtaining information from the device to determine usage, Operability, etc.

Cisco UCS Manager does not support on demand diagnostics. This feature checks the drive status automatically and provides a view only status. In case the self test fails, Cisco UCS Manager also raises a major fault.

## Viewing the Status of the Drive Self-test

### Procedure

**Step 1** In the **Navigation** pane, click the Equipment tab.

**Step 2** Expand **Equipment** > **Rack Mounts** > **Servers** > .

**Note**
For Cisco UCS C125 M5 Servers, expand **Equipment** > **Rack Mounts** > **Enclosures** > **Rack Enclosure rack_enclosure_number** > **Servers**.

**Step 3** Choose the server that you want to check the drive status.

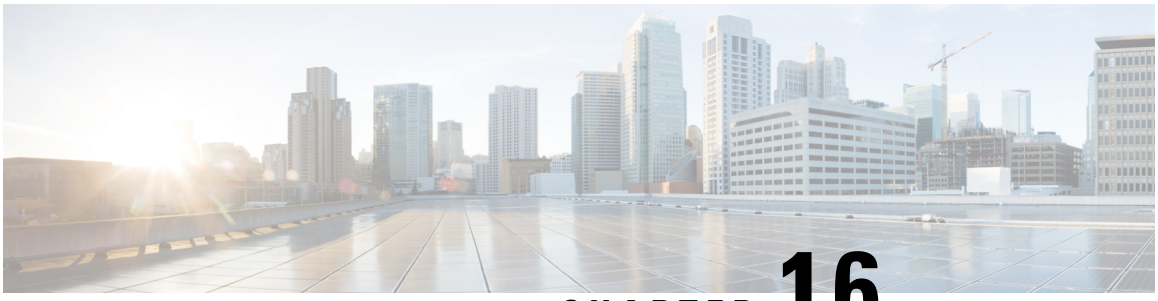
**Step 4** In the Work pane, click the **Inventory** > **Storage** > **Disks** tabs.

The Storage Controller inventory appears.

**Step 5** Click the **Storage** sub-tab.

**Step 6** If the **Drive State** shows **Self Test Failed**, drive may become unusable resulting in loss of information. Cisco recommends to back up data and replace the drive.

Cisco UCS Manager raises a major fault to when dirve goes into **Self Test Failed** state. In **Self Test Failed** state, normal functions continue to work.



CHAPTER 16

# Cisco UCS S3260 System Storage Management



## Storage Server Features and Components Overview

### Storage Server Features

The following table summarizes the Cisco UCS S3260 system features:

*Table 20: Cisco UCS S3260 System Features*

| Feature | Description |
|---|---|
| Chassis | Four rack unit (4RU) chassis |
| Processors | • Cisco UCS S3260 M5 server nodes: Two Intel Skylake 2S-EP processors inside each server node. |
| Memory | Up to 16 DIMMs inside each server node. |
| Multi-bit error protection | This system supports multi-bit error protection. |

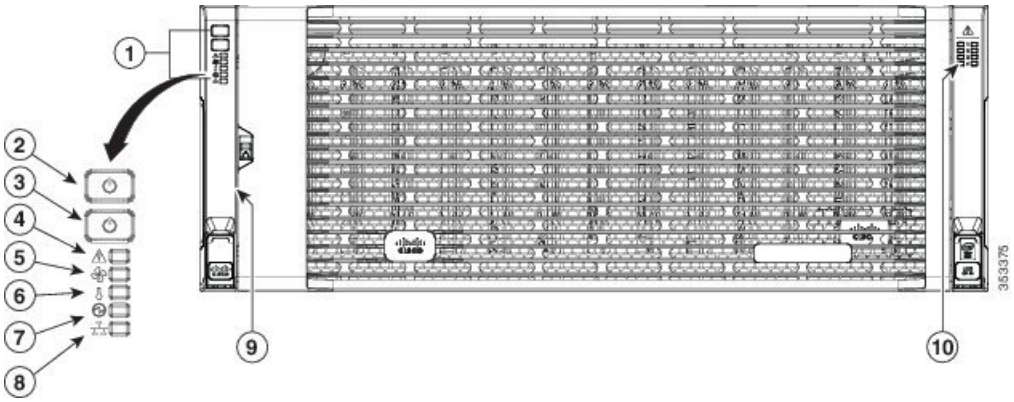
| Feature | Description |
|---|---|
| Storage | The system has the following storage options:<br>• Up to 56 top-loading 3.5-inch drives<br>• Up to four 3.5-inch, rear-loading drives in the optional drive expander module<br>• Up to four 2.5-inch, rear-loading SAS solid state drives (SSDs)<br>• Two 7 mm NVMe drive inside the server node<br>**Note** This is applicable for S3260 M5 servers only.<br>• Two 15 mm NVMe drive supported for IO Expander |
| Disk Management | The system supports up to two storage controllers:<br>• One dedicated mezzanine-style socket for a Cisco storage controller card inside each server node |
| RAID Backup | The supercap power module (SCPM) mounts to the RAID controller card. |
| PCIe I/O | The optional I/O expander provides two 8x Gen 3 PCIe expansion slots.<br>Release 3.2(3) and later supports the following for S3260 M5 servers:<br>• Intel X550 dual-port 10GBase-T<br>• Qlogic QLE2692 dual-port 16G Fiber Channel HBA<br>• N2XX-AIPCI01 Intel X520 Dual Port 10Gb SFP+ Adapter |
| Network and Management I/O | The system can have one or two system I/O controllers (SIOCs). These provide rear-panel management and data connectivity.<br>• Two SFP+ 40 Gb ports each SIOC.<br>• One 10/100/1000 Ethernet dedicated management port on each SIOC.<br>The server nodes each have one rear-panel KVM connector that can be used with a KVM cable, which provides two USB, one VGA DB-15, and one serial DB-9 connector. |

| Feature | Description |
|---|---|
| Power | Two or four power supplies, 1050 W each (hot-swappable and redundant as 2+2). |
| Cooling | Four internal fan modules that pull front-to-rear cooling, hot-swappable. Each fan module contains two fans.<br>In addition, there is one fan in each power supply. |

### Front Panel Features

The following image shows the front panel features for the Cisco UCS S3260 system:

*Figure 4: Front Panel Features*

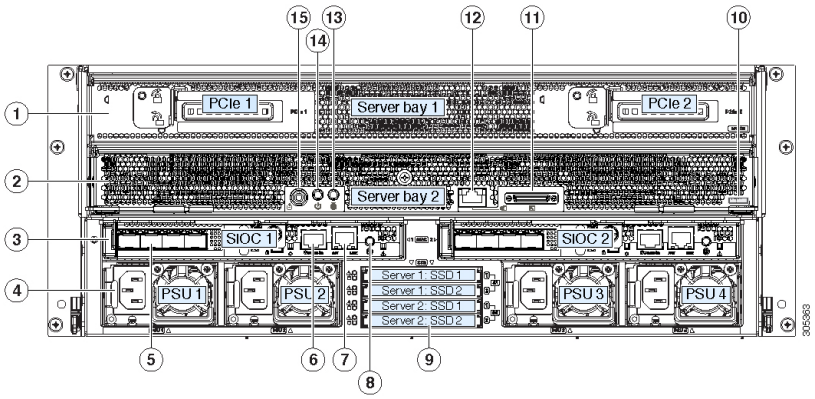



| | | | |
|---|---|---|---|
| 1 | Operations panel | 6 | Temperature status LED |
| 2 | System Power button/LED | 7 | Power supply status LED |
| 3 | System unit identification button/LED | 8 | Network link activity LED |
| 4 | System status LED | 9 | Pull-out asset tag (not visible under front bezel) |
| 5 | Fan status LED | 10 | Internal-drive status LEDs |

### Rear Panel Features

The following image shows the rear panel features for the Cisco UCS S3260 system:

*Figure 5: Front Panel Features*




Disk Slots

| | | | |
|---|---|---|---|
| 1 | Server bay 1<br>• (Optional) I/O expander, as shown (with Cisco UCS S3260 M5 server node only)<br>• (Optional) server node<br>• (Optional) drive expansion module | 8 | Not used at this time |
| 2 | Server bay 2<br>• (Optional) server node (Cisco UCS S3260 M5 shown)<br>(Optional) drive expansion module | 9 | Not used at this time |

| | | | |
|---|---|---|---|
| 3 | System I/O controller (SIOC)<br>• SIOC 1 is required if you have a server node in server bay 1<br>• SIOC 2 is required if you have server node in server bay 2 | 10 | Solid state drive bays (up to four 2.5-inch SAS SSDs)<br>• SSDs in bays 1 and 2 require a server node in server bay 1<br>• SSDs in bays 3 and 4 require a server node in server bay 2 |
| 4 | Power supplies (four, redundant as 2+2) | 11 | **Note**<br>This label identifies a Cisco UCS S3260 M5 server node. |
| 5 | 40-Gb SFP+ ports (two on each SIOC) | 12 | KVM console connector (one each server node).<br>Used with a KVM cable that provides two USB, one VGA, and one serial connector |
| 6 | Chassis Management Controller (CMS) Debug Firmware Utility port (one each SIOC) | 13 | Server node unit identification button/LED |
| 7 | 10/100/1000 dedicated management port, RJ-45 connector (one each SIOC) | 14 | Server node power button |
| | | 15 | Server node reset button (resets chipset in the server node |

## Storage Server Components

### Server Nodes

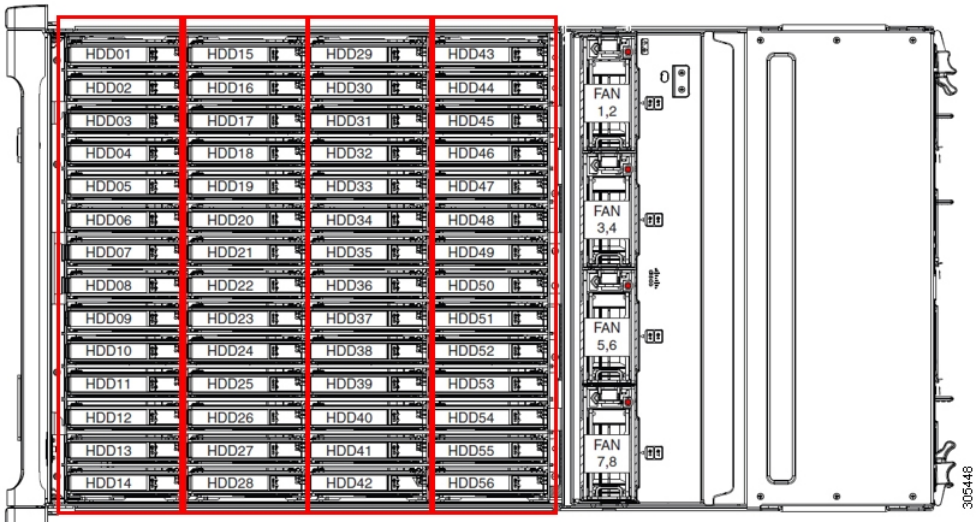
The Cisco UCS S3260 system consists of one or two server nodes, each with two CPUs, DIMM memory of 128, 256, or 512 GB, and a RAID card up to 4 GB cache or a pass-through controller. The server nodes can be one of the following:

- Cisco UCS S3260 M5 Server Node—This node might include an optional I/O expander module that attaches to the top of the server node.

### Disk Slots

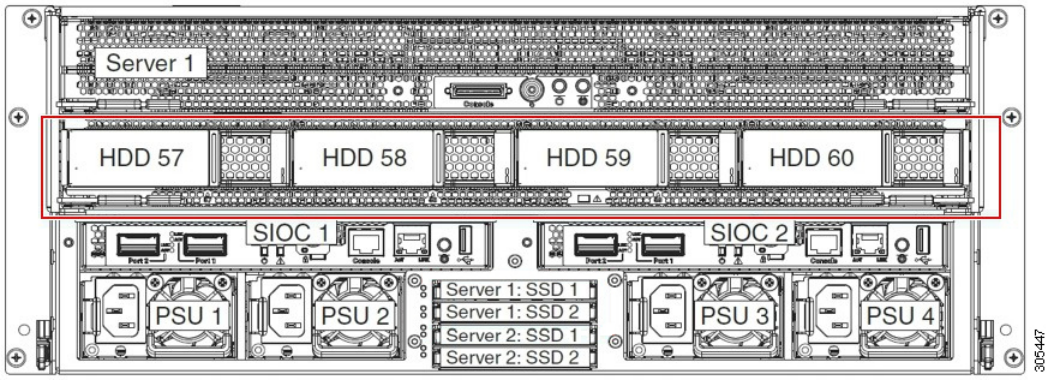
The Cisco UCS S3260 chassis has 4 rows of 14 disk slots on the HDD motherboard and 4 additional disk slots on the HDD expansion tray. The following image shows the disk arrangement for the 56 top-accessible, hot swappable 3.5-inch 6 TB or 4 TB 7200 rpm NL-SAS HDD drives. A disk slot has two SAS ports and each is connected a SAS expander in the chassis.

*Figure 6: Cisco UCS S3260 Top View*




The following image shows the Cisco UCS S3260 chassis with the 4 additional disk slots on the HDD expansion tray.

*Figure 7: Cisco UCS 3260 with the HDD expansion tray (Rear View)*




If you have two server nodes with two SIOCs, you will have the following functionality:

1. The top server node works with the left SIOC (Server Slot1 with SIOC1).
2. The bottom server works with the right SIOC (Sever Slot 2 with SIOC2).

If you have one server node with two SIOCs, you can enable Server SIOC Connectivity functionality. Beginning with release 3.1(3), Cisco UCS S3260 system supports Server SIOC Connectivity functionality. Using this functionality, you can configure the data path through both the primary and auxiliary SIOCs when the chassis has single server and dual SIOCs set up.

### SAS Expanders

The Cisco UCS S3260 system has two SAS expanders that run in redundant mode and connect the disks at the chassis level to storage controllers on the servers. The SAS expanders provide two paths between a storage controller, and hence enable high availability. They provide the following functionality:

- Manage the pool of hard drives.
- Disk zone configuration of the hard drives to storage controllers on the servers.

Beginning with release 3.2(3a), Cisco UCS Manager can enable single path access to disk by configuring single DiskPort per disk slot. This ensures that the server discovers only a single device and avoid a multi-path configuration.

The following table describes how the ports in each SAS expander are connected to the disks based on the type of deployment.

| Port range | Connectivity |
|---|---|
| 1-56 | Top accessible disks |
| 57-60 | Disks in the HDD expansion tray. |

**Note** The number of SAS uplinks between storage controller and SAS expander can vary based on the type of controller equipped in the server.

### Storage Enclosures

A Cisco UCS S3260 system has the following types of storage enclosures:

#### Chassis Level Storage Enclosures

- **HDD motherboard enclosure**—The 56 dual port disk slots in the chassis comprise the HDD motherboard enclosure.
- **HDD expansion tray**—The 4 additional dual disk slots in the Cisco UCS S3260 system comprise the HDD expansion tray.

**Note** The HDD expansion tray is a field replaceable unit (FRU). The disks will remain unassigned upon insertion, and can be assigned to storage controllers. For detailed steps on how to perform disk zoning, see Disk Zoning Policies, on page 253

#### Server level Storage Enclosures

Server level storage enclosures are pre-assigned dedicated enclosures to the server. These can be one of the following:

- **Rear Boot SSD enclosure**—This enclosure contains two 2.5 inch disk slots on the rear panel of the Cisco UCS S3260 system. Each server has two dedicated disk slots. These disk slots support SATA SSDs.
- **Server board NVMe enclosure**—This enclosure contains one PCIe NVMe controller.

**Note** In the Cisco UCS S3260 system, even though disks can be physically present on the two types of enclosures described above, from the host OS all the disks are viewed as part of one SCSI enclosure. They are connected to SAS expanders that are configured to run as single SES enclosure.

### Storage Controllers

#### Mezzanine Storage Controllers

The following table lists the storage controller type, firmware type, modes, sharing and OOB support for the various storage controllers.

*Table 21:*

| Storage Controller Type | Firmware type | Modes | Sharing | OOB Support |
|---|---|---|---|---|
| UCSC-S3X60-R1GB | Mega RAID | HW RAID, JBOD | No | Yes |
| UCSC-S3X60-HBA | Initiator Target | Pass through | Yes | Yes |
| UCS-S3260-DHBA | Initiator Target | Pass through | Yes | Yes |
| UCS-S3260-DRAID | Mega RAID | HW RAID, JBOD | No | Yes |

#### Other storage controllers

**SW RAID Controller**—The servers in the Cisco UCS S3260 system support two dedicated internal SSDs embedded into the PCIe riser that is connected to the SW RAID Controller.

**NVMe Controller**—This controller is used by servers in the Cisco UCS S3260 system for inventory and firmware updates of NVMe disks.

For more details about the storage controllers supported in the various server nodes, see the related service note:

- Cisco UCS S3260 M5 Server Node For Cisco UCS S3260 Storage Server Service Note

# Cisco UCS S3260 Storage Management Operations

The following table summarizes the various storage management operations that you can perform with the Cisco UCS Manager integrated Cisco UCS S3260 system.

| Operation | Description | See: |
|---|---|---|
| Disk Sharing for High Availability | The SAS expanders in the Cisco UCS S3260 system can manage the pool of drives at the chassis level. To share disks for high availability, perform the following:<br>1. Creating disk zoning policies.<br>2. Creating disk slots and assigning ownership.<br>3. Associating disks to chassis profile. | "Disk Zoning Policies" section in this guide. |
| Storage Profiles, Disk Groups and Disk Group Configuration Policies | You can utilize Cisco UCS Manager's Storage Profile and Disk Group Policies for defining storage disks, disk allocation and management in the Cisco UCS S3260 system. | "Storage Profiles" section in the |
| Storage Enclosure Operations | You can swap the HDD expansion tray with a server, or remove the tray if it was previously inserted. | "Removing Chassis Level Storage Enclosures" section in this guide. |

# Disk Sharing for High Availability

## Disk Zoning Policies

You can assign disk drives to the server nodes using disk zoning. Disk zoning can be performed on the controllers in the same server or on the controllers on different servers. Disk ownership can be one of the following:

**Unassigned**

Unassigned disks are those not visible to the server nodes.

**Dedicated**

If this option is selected, you will need to set the values for the **Server**, **Controller**, **Drive Path**, and **Slot Range** for the disk slot.

**Note** A disk is visible only to the assigned controller.

Beginning with release 3.2(3a), Cisco UCS Manager can enable single path access to disk by configuring single DiskPort per disk slot for Cisco UCS S3260 M5 and higher servers. Setting single path configuration ensures that the server discovers the disk drive only through a single drive path chosen in the configuration. Single path access is supported only for **Cisco UCS S3260 Dual Pass Through Controller** (UCS-S3260-DHBA)

Once single path access is enabled, you cannot downgrade to any release earlier than 3.2(3a). To downgrade, disable this feature and assign all the disk slots to both the disk ports by configuring disk path of the disk slots to **Path Both** in disk zoning policy.

**Shared**

Shared disks are those assigned to more than one controller. They are specifically used when the servers are running in a cluster configuration, and each server has its storage controllers in HBA mode.

**Note** Shared mode cannot be used under certain conditions when dual HBA controllers are used.

**Chassis Global Hot Spare**

If this option is selected, you will need to set the value for the **Slot Range** for the disk.

**Important** Disk migration and claiming orphan LUNs: To migrate a disk zoned to a server (Server 1) to another server (Server 2), you must mark the virtual drive (LUN) as transport ready or perform a hide virtual drive operation. You can then change the disk zoning policy assigned for that disk. For more information on virtual drive management, see the *Disk Groups and Disk Configuration Policies* section of the Cisco UCS Manager Storage Management Guide.

# Creating a Disk Zoning Policy

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | UCS-A# **scope org** *org-name* | Enters organization mode for the specified organization. To enter the root organization mode, type / as the *org-name* . |
| **Step 2** | UCS-A org/ # **create disk-zoning-policy** *diskzoning policy-name* | Creates a disk zoning policy name with the specified disk zoning policy name. |
| **Step 3** | UCS-A /org/disk-zoning-policy* # **commit-buffer** | Commits the transaction to the system configuration. |

### Example

The following example creates the dzp1 disk zoning policy:

```
UCS-A# scope org
UCS-A /org # create disk-zoning-policy dzp1
UCS-A /org/disk-zoning-policy*# commit-buffer
UCS-A /org/disk-zoning-policy#
```

# Creating Disk Slots and Assigning Ownership

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | UCS-A# **scope org** *org-name* | Enters organization mode for the specified organization. To enter the root organization mode, type **/** as the *org-name* . |
| **Step 2** | UCS-A org/ # **disk-zoning-policy** *disk-zoning-policy-name* | Enters the disk zoning policy. |
| **Step 3** | UCS-A org/disk-zoning-policy # **create disk-slot** *slot-id* | Creates disk slot with the specified slot number. |
| **Step 4** | UCS-A org/disk-zoning-policy/disk-slot* # **set ownership** *ownership-type {chassis-global-host-spare\|dedicated\|shared\|unassigned}* | Specifies the disk ownership to be one of the following:<br>• **chassis-global-hot-spare**—Chassis Global Hot Spare<br>• **dedicated**—Dedicated<br>Beginning with release 3.2(3a), Cisco UCS Manager can enable single path access to disk by configuring single DiskPort per disk slot. This ensures that the server discovers only a single device and avoid a multi-path configuration.<br>Drive Path options are:<br>• **path-both (Default)** - Drive path is zoned to both the SAS expanders.<br>• **path-0** - Drive path is zoned to SAS expander 1.<br>• **path-1** - Drive path is zoned to SAS expander 2.<br>Use the following command to set the drivepath:<br>**set drivepath *drivepath{path-0/path-1/path-both}***<br>• **shared**—Shared |

| | Command or Action | Purpose |
|---|---|---|
| | | **Note** Shared mode cannot be used under certain conditions when dual HBA controllers are used. To view the conditions for **Shared** mode for Dual HBA controller, see Table 22: Limitations for Shared Mode for Dual HBA Controller, on page 256.<br>• **unassigned**—Unassigned |
| **Step 5** | UCS-A org/disk-zoning-policy/disk-slot* # **create controller-ref** *server-id* **sas** *controller-id* | Creates controller reference for the specified server slot. |
| **Step 6** | UCS-A org/disk-zoning-policy/disk-slot # **commit-buffer** | Commits the transaction. |

***Table 22: Limitations for Shared Mode for Dual HBA Controller***

| Server | HDD Tray | Controller | Shared mode Support |
|---|---|---|---|
| Cisco UCS S3260 | No | Dual HBA | Not Supported |
| Cisco UCS S3260 | HDD Tray | Dual HBA | Not Supported |
| Pre-Provisioned | HDD Tray | Dual HBA | Not Supported |

### Example

The following example creates disk slot 1, sets the ownership as shared, creates a controller reference for the server slot 1, and commits the transaction:

```
UCS-A# scope org
UCS-A /org # scope disk-zoning-policy test
UCS-A /org/disk-zoning-policy* # create disk-slot 1
UCS-A /org/disk-zoning-policy/disk-slot* # set ownership shared
UCS-A /org/disk-zoning-policy/disk-slot* # create controller-ref 1 sas 1
UCS-A /org/disk-zoning-policy/disk-slot* # create controller-ref 2 sas 1
UCS-A /org/disk-zoning-policy/disk-slot* #commit-buffer
UCS-A /org/disk-zoning-policy/disk-slot #
```

# Associating Disk Zoning Policies to Chassis Profile

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | UCS-A# **scope org** *org-name* | Enters organization mode for the specified organization. To enter the root organization mode, type / as the *org-name* . |
| **Step 2** | UCS-A org/ # **create chassis-profile** *chassis-profile-name* | Creates a chassis profile with the specified name. |
| **Step 3** | UCS-A org/chassis-profile* # **set disk-zoning-policy** *disk-zoning-policy* | Sets the specified disk-zoning-policy. |
| **Step 4** | UCS-A org/chassis-profile* # **commit-buffer** | Commits the transaction. |
| **Step 5** | UCS-A org/chassis-profile # **associate chassis** *chassis-id* | Associates the disks in the disk zoning policy to the chassis with the specified chassis number. |

### Example

The following example creates the ch1 chassis profile, sets the disk zoning policy all56shared, commits the transaction and associates the disk in the all56shared policy with chassis 3:

```
UCS-A# scope org
UCS-A /org # create chassis-profile ch1
UCS-A /org/chassis-profile* # set disk-zoning-policy all56shared
UCS-A /org/chassis-profile* # commit-buffer
UCS-A /org/chassis-profile # associate chassis 3
UCS-A /org/fw-chassis-pack/pack-image #
```

# Disk Migration

Before you can migrate a disk zoned from one server to another, you must mark the virtual drive(LUN) as transport ready or perform a hide virtual drive operation. This will ensure that all references from the service profile have been removed prior to disk migration. For more information on virtual drives, please refer to the "virtual drives" section in the

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | UCS-A# **scope chassis** *chassis-num* | Enters chassis mode for the specified chassis. |
| **Step 2** | UCS-A /chassis# **scope virtual-drive-container** *virtual-drive-container-num* | Enters the virtual drive container with the specified number. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 3** | UCS-A /chassis/virtual-drive-container# **scope virtual-drive** *virtual-drive--num* | Enters the virtual drive for the specified virtual drive container. |
| **Step 4** | UCS-A /chassis/virtual-drive-container/virtual-drive# **scope virtual-drive** *virtual-drive--num* **set admin-state** *admin-state* | Specifies one of the following admin states for the virtual drive:<br>• **clear-transport-ready** — Sets the state of the virtual drive to no longer be transport ready.<br>• **delete** — Deletes the virtual drive.<br>• **hide**— Choose this option for the safe migration of the virtual drive from one server to another.<br>**Note** All virtual drives on a disk group must be marked as hidden before migrating or unassigning the disks from a server node.<br>• **transport-ready** — Choose this option for the safe migration of the virtual drive from one server to another.<br>**Note** When a virtual drive is marked as transport ready, the storage controller will disable all IO operations on the drive. In addition, after zoning the virtual drive and importing the foreign configuration, the virtual drive will be operational. |
| **Step 5** | UCS-A /chassis/virtual-drive-container/virtual-drive# **commit-buffer** | Commits the transaction to the system configuration. |

### Example

The following example sets the state of the virtual drive 1001 in the virtual drive container 1 to transport ready:

```
UCS-A# scope chassis
UCS-A /chassis# scope virtual-drive-container 1
UCS-A /chassis/virtual-drive-container# scope virtual-drive 1001
UCS-A /chassis/virtual-drive-container/virtual-drive# set admin-state transport-ready
UCS-A /chassis/virtual-drive-container/virtual-drive# commit-buffer
```

# Storage Enclosure Operations

## Removing Chassis Level Storage Enclosures

You can remove the storage enclosure corresponding to HDD expansion tray in Cisco UCS Manager after it is physically removed. You cannot remove server level or any other chassis level storage enclosures.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | UCS-A# **scope chassis** *chassis-id* | Enters chassis mode for the specified chassis. |
| **Step 2** | UCS-A /chassis # **remove storage-enclosure** *storage-enclosure-name* | Removes the chassis level storage enclosure with the specified name. |

### Example

The following example removes storage enclosure 25 from chassis 2:

```
UCS-A# scope chassis 2
UCS-A /chassis# remove storage-enclosure 25
UCS-A /chassis#
```

# SAS Expander Configuration Policy

## Creating SAS Expander Configuration Policy

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | UCS-A# **scope org** *org-name* | Enters organization mode for the specified organization. To enter the root organization mode, type **/** as the *org-name* . |
| **Step 2** | UCS-A org/ # **create sas-expander-configuration-policy** *sas-expander-configuration-policy-name* | Creates a SAS expander configuration policy with the specified policy name. |
| **Step 3** | (Optional) UCS-A /org/sas-expander-configuration-policy* # **set descr** *description* | Provides a description for the policy. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 4** | (Optional) UCS-A /org/sas-expander-configuration-policy* # **set 6g-12g-mixed-mode** *disabled\|enabled\|no-change* | **Note** Enabling or disabling 6G-12G Mixed Mode causes system reboot.<br>• **Disabled**—Connection Management is disabled in this policy and the Sas Expander uses only 6G speeds even if 12G is available.<br>• **Enabled**—Connection Management is enabled in this policy and it intelligently shifts between 6G and 12 G speeds based on availability.<br>• **No Change** (Default) —Pre-existing configuration is retained. |
| **Step 5** | UCS-A /org/sas-expander-configuration-policy* # **commit-buffer** | Commits the transaction to the system configuration. |

### Example

The following example creates the secp1 SAS expander configuration policy:

```
UCS-A# scope org
UCS-A /org # create sas-expander-configuration-policy secp1
UCS-A /org/sas-expander-configuration-policy*# set 6g-12g-mixed-mode enabled
UCS-A /org/sas-expander-configuration-policy*# commit-buffer
UCS-A /org/sas-expander-configuration-policy#
```

# Deleting a SAS Expander Configuration Policy

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | UCS-A# **scope org** *org-name* | Enters organization mode for the specified organization. To enter the root organization mode, type / as the *org-name* . |
| **Step 2** | UCS-A org/ # **delete sas-expander-configuration-policy** *sas-expander-configuration-policy-name* | Deletes a SAS expander configuration policy with the specified policy name. |
| **Step 3** | UCS-A /org* # **commit-buffer** | Commits the transaction to the system configuration. |

### Example

The following example deletes the secp1 SAS expander configuration policy:

```
UCS-A# scope org
UCS-A /org # delete create sas-expander-configuration-policy secp1
UCS-A /org*# commit-buffer
UCS-A /org/#
```