



Configuring Server Boot

This chapter includes the following sections:

- [Boot Policy, page 1](#)
- [UEFI Boot Mode, page 2](#)
- [UEFI Secure Boot, page 3](#)
- [CIMC Secure Boot, page 3](#)
- [Creating a Boot Policy, page 5](#)
- [SAN Boot, page 8](#)
- [iSCSI Boot, page 10](#)
- [LAN Boot, page 42](#)
- [Local Devices Boot, page 43](#)
- [Configuring an EFI Shell Boot for a Boot Policy, page 50](#)
- [Deleting a Boot Policy, page 51](#)
- [UEFI Boot Parameters, page 51](#)

Boot Policy

The Cisco UCS Manager enables you to create a boot policy for blade servers and rack servers.

The Cisco UCS Manager boot policy overrides the boot order in the BIOS setup menu and determines the following:

- Selection of the boot device
- Location from which the server boots
- Order in which boot devices are invoked

For example, you can have associated servers boot from a local device, such as a local disk or CD-ROM (VMedia), or you can select a SAN boot or a LAN (PXE) boot.

You can either create a named boot policy to associate with one or more service profiles, or create a boot policy for a specific service profile. A boot policy must be included in a service profile, and that service profile must be associated with a server for it to take effect. If you do not include a boot policy in a service profile, Cisco UCS Manager applies the default boot policy.

**Note**

Changes to a boot policy might be propagated to all servers created with an updating service profile template that includes that boot policy. Re-association of the service profile with the server to rewrite the boot order information in the BIOS is automatically triggered.

You can also specify the following for the boot policy:

- Local LUN name. The name specified is the logical name in the storage profile, not the deployed name. Specify only a primary name. Specifying a secondary name results in a configuration error.
- Specific JBOD disk number for booting from JBOD disks.
- Any LUN for backward compatibility; however, we do not recommend this. Other devices must not have bootable images to ensure a successful boot.

UEFI Boot Mode

Unified Extensible Firmware Interface (UEFI) is a specification that defines a software interface between an operating system and platform firmware. Cisco UCS Manager uses UEFI to replace the BIOS firmware interfaces. This allows the BIOS to run in UEFI mode while still providing legacy support.

You can choose either legacy or UEFI boot mode when you create a boot policy. Legacy boot mode is supported for all Cisco UCS servers. UEFI boot mode is supported only on M3 and higher servers, and allows you to enable UEFI secure boot mode.

UEFI PXE boot is supported with all Cisco VIC adapters on Cisco UCS rack servers integrated with Cisco UCS Manager Release 2.2(4) and later releases. Beginning with Cisco UCS Manager Release 2.2(1), UEFI PXE boot is supported on all Cisco blade servers.

The following limitations apply to the UEFI boot mode:

- UEFI boot mode is not supported with the following combinations:
 - Gen-3 Emulex and QLogic adapters on Cisco UCS blade and rack servers integrated with Cisco UCS Manager.
 - iSCSI boot for all adapters on Cisco UCS rack servers integrated with Cisco UCS Manager.
- If you want to use UEFI boot mode with two iSCSI LUNs, you must manually specify a common iSCSI initiator name in the service profile that is applied to both underlying iSCSI eNICs rather than allowing Cisco UCS Manager to select the name from an IQN suffix pool. If you do not supply a common name, Cisco UCS Manager will not be able to detect the second iSCSI LUN.
- You cannot mix UEFI and legacy boot mode on the same server.
- The server will boot correctly in UEFI mode only if the boot devices configured in the boot policy have UEFI-aware operating systems installed. If a compatible OS is not present, the boot device is not displayed on the **Actual Boot Order** tab in the **Boot Order Details** area.

- In some corner cases, the UEFI boot may not succeed because the UEFI boot manager entry was not saved correctly in the BIOS NVRAM. You can use the UEFI shell to enter the UEFI boot manager entry manually. This situation could occur in the following situations:
 - If a blade server with UEFI boot mode enabled is disassociated from the service profile, and the blade is manually powered on using the **Equipment** tab or the front panel.
 - If a blade server with UEFI boot mode enabled is disassociated from the service profile, and a direct VIC firmware upgrade is attempted.
 - If a blade or rack server with UEFI boot mode enabled is booted off SAN LUN, and the service profile is migrated.

You can create UEFI boot parameters in Cisco UCS Manager. [UEFI Boot Parameters](#), on page 51 provides more information.

UEFI Secure Boot

Cisco UCS Manager supports UEFI secure boot on Cisco UCS B-Series M3 and higher Blade servers, Cisco UCS C-Series M3 and higher Rack servers, and Cisco UCS S-Series M4 Rack servers. When UEFI secure boot is enabled, all executables, such as boot loaders and adapter drivers, are authenticated by the BIOS before they can be loaded. To be authenticated, the images must be signed by either the Cisco Certificate Authority (CA) or a Microsoft CA.

The following limitations apply to UEFI secure boot:

- UEFI boot mode must be enabled in the boot policy.
- The Cisco UCS Manager software and the BIOS firmware must be at Release 2.2 or greater.



Note UEFI boot mode is supported on Cisco UCS C-Series and S-Series rack servers beginning with Release 2.2(3a).

- User-generated encryption keys are not supported.
- UEFI secure boot can only be controlled by Cisco UCS Manager.
- If you want to downgrade to an earlier version of Cisco UCS Manager, and you have a server in secure boot mode, you must disassociate, then re-associate the server before downgrading. Otherwise, server discovery is not successful.

CIMC Secure Boot

With CIMC secure boot, only Cisco signed firmware images can be installed and run on the servers. When the CIMC is updated, the image is certified before the firmware is flashed. If certification fails, the firmware is not flashed. This prevents unauthorized access to the CIMC firmware.

Guidelines and Limitations for CIMC Secure Boot

- CIMC secure boot is supported on Cisco UCS M3 rack servers.



Note CIMC secure boot is enabled by default on the Cisco UCS C220 M4, C240 M4 rack servers, and is automatically enabled on the Cisco UCS C460 M4 rack server after upgrading to CIMC firmware release 2.2(3) or higher.

- After CIMC secure boot is enabled, you cannot disable it.
- After CIMC secure boot is enabled on a server, you cannot downgrade to a CIMC firmware image prior to 2.1(3).

Determining the CIMC Secure Boot Status

Procedure

	Command or Action	Purpose
Step 1	UCS-A# scope server <i>server-num</i>	Enters server mode for the specified server.
Step 2	UCS-A /chassis/server # scope cimc	Enters server CIMC mode.
Step 3	UCS-A /server/cimc # show secure-boot	Displays the CIMC secure boot status for the specified server. This can be one of the following: <ul style="list-style-type: none"> • Unsupported—CIMC secure boot is not supported on the server. • Disabled—CIMC secure boot is supported, but is disabled on the server. • Enabling—CIMC secure boot has been enabled, and the operation is in process. • Enabled—CIMC secure boot is enabled on the server.

The following example shows how to display the CIMC secure boot status:

```
UCS-A# scope server 1
UCS-A /chassis/server # scope cimc
UCS-A /chassis/server/cimc # show secure-boot
Secure Boot: Disabled
UCS-A /chassis/server/cimc #
```

Enabling CIMC Secure Boot

Procedure

	Command or Action	Purpose
Step 1	UCS-A# scope server <i>server-num</i>	Enters server mode for the specified server.
Step 2	UCS-A /chassis/server # scope cimc	Enters server CIMC mode.
Step 3	UCS-A /server/cimc # enable secure-boot	Enables CIMC secure boot status for the specified server. CIMC secure boot is only supported on Cisco UCS M3 rack servers. Note Once enabled, CIMC secure boot cannot be disabled.
Step 4	UCS-A /server/cimc # commit-buffer	Commits the transaction to the system configuration.

The following example shows how to enable CIMC secure boot and commit the transaction:

```
UCS-A# scope server 1
UCS-A /chassis/server # scope cimc
UCS-A /chassis/server/cimc # enable secure-boot
Warning: When committed, CIMC Secure Boot and Installation Feature will be enabled for the
server.
This is an irreversible operation!!

UCS-A /chassis/server/cimc* # commit-buffer
UCS-A /chassis/server/cimc #
```

Creating a Boot Policy

You can also create a local boot policy that is restricted to a service profile or service profile template. However, Cisco recommends that you create a global boot policy that can be included in multiple service profiles or service profile templates.

Before You Begin

If you are creating a boot policy that boots the server from a SAN LUN and you require reliable SAN boot operations, you must first remove all local disks from servers associated with a service profile that includes the boot policy.



Note

This does not apply for Cisco UCS M3 and M4 servers.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# scope org <i>org-name</i>	Enters organization mode for the specified organization. To enter the root organization mode, type <i>/</i> as the <i>org-name</i> .
Step 2	UCS-A /org # create boot-policy <i>policy-name</i> [purpose { operational utility }]	Creates a boot policy with the specified policy name, and enters organization boot policy mode. When you create the boot policy, specify the operational option. This ensures that the server boots from the operating system installed on the server. The utility options is reserved and should only be used if instructed to do so by a Cisco representative.
Step 3	UCS-A /org/boot-policy # set descr <i>description</i>	(Optional) Provides a description for the boot policy. Note If your description includes spaces, special characters, or punctuation, you must begin and end your description with quotation marks. The quotation marks do not appear in the description field of any show command output.
Step 4	UCS-A /org/boot-policy # set reboot-on-update { no yes }	Specifies whether the servers using this boot policy are automatically rebooted after you make changes to the boot order.
Step 5	UCS-A /org/boot-policy # set enforce-vnic-name { no yes }	If you choose yes , Cisco UCS Manager displays a configuration error and reports whether one or more of the vNICs, vHBAs, or iSCSI vNICs listed in the Boot Order table match the server configuration in the service profile. If you choose no , Cisco UCS Manager uses the vNICs, vHBAs, or iSCSI vNICs (as appropriate for the boot option) from the service profile.
Step 6	UCS-A /org/boot-policy # set boot-mode { legacy uefi }	Specifies whether the servers using this boot policy are using UEFI or legacy boot mode.
Step 7	UCS-A /org/boot-policy # commit-buffer	Commits the transaction to the system configuration.
Step 8	UCS-A /org/boot-policy # create boot-security	Enters boot security mode for the specified boot policy.
Step 9	UCS-A /org/boot-policy/boot-security # set secure-boot { no yes }	Specifies whether secure boot is enabled for the boot policy.
Step 10	UCS-A /org/boot-policy/boot-security # commit-buffer	Commits the transaction to the system configuration.

The following example shows how to create a boot policy named boot-policy-LAN, specify that servers using this policy will not be automatically rebooted when the boot order is changed, set the UEFI boot mode, enable UEFI boot security, and commit the transaction:

```
UCS-A# scope org /
UCS-A /org* # create boot-policy boot-policy-LAN purpose operational
UCS-A /org/boot-policy* # set descr "Boot policy that boots from the LAN."
UCS-A /org/boot-policy* # set reboot-on-update no
UCS-A /org/boot-policy* # set boot-mode uefi
UCS-A /org/boot-policy* # commit-buffer
UCS-A /org/boot-policy # create boot-security
UCS-A /org/boot-policy/boot-security* # set secure-boot yes
UCS-A /org/boot-policy/boot-security* # commit-buffer
UCS-A /org/boot-policy/boot-security #
```

What to Do Next

Configure one or more of the following boot options for the boot policy and set their boot order:

- **LAN Boot** —Boots from a centralized provisioning server. It is frequently used to install operating systems on a server from that server.

If you choose the LAN Boot option, continue to [Configuring a LAN Boot for a Boot Policy](#), on page 42.

- **SAN Boot** —Boots from an operating system image on the SAN. You can specify a primary and a secondary SAN boot. If the primary boot fails, the server attempts to boot from the secondary.

We recommend that you use a SAN boot policy, because it offers the most service profile mobility within the system. If you boot from the SAN, when you move a service profile from one server to another, the new server boots from exactly the same operating system image. Therefore, the new server appears to be exactly the same server to the network.

If you choose the SAN Boot option, continue to [Configuring a SAN Boot for a Boot Policy](#), on page 8.

- **Virtual Media Boot** —Mimics the insertion of a physical CD into a server. It is typically used to manually install operating systems on a server.

If you choose the Virtual Media boot option, continue to [Configuring a Virtual Media Boot for a Boot Policy](#), on page 47.



Tip

If you configure a local disk and a SAN LUN for the boot order storage type and the operating system or logical volume manager (LVM) is configured incorrectly, the server might boot from the local disk rather than the SAN LUN.

For example, on a server with Red Hat Linux installed, where the LVM is configured with default LV names and the boot order is configured with a SAN LUN and a local disk, Linux reports that there are two LVs with the same name and boots from the LV with the lowest SCSI ID, which could be the local disk.

Include the boot policy in a service profile and template.

SAN Boot

You can configure a boot policy to boot one or more servers from an operating system image on the SAN. The boot policy can include a primary and a secondary SAN boot. If the primary boot fails, the server attempts to boot from the secondary.

Cisco recommends using a SAN boot, because it offers the most service profile mobility within the system. If you boot from the SAN when you move a service profile from one server to another, the new server boots from the same operating system image. Therefore, the new server appears as the same server to the network.

To use a SAN boot, ensure that the following is configured:

- The Cisco UCS domain must be able to communicate with the SAN storage device that hosts the operating system image.
- A boot target LUN (Logical Unit Number) on the device where the operating system image is located.

**Note**

SAN boot is not supported on Gen-3 Emulex adapters on Cisco UCS blade and rack servers.

Configuring a SAN Boot for a Boot Policy

**Tip**

If you configure a local disk and a SAN LUN for the boot order storage type and the operating system or logical volume manager (LVM) is configured incorrectly, the server might boot from the local disk rather than the SAN LUN.

For example, on a server with Red Hat Linux installed, where the LVM is configured with default LV names and the boot order is configured with a SAN LUN and a local disk, Linux reports that there are two LVs with the same name and boots from the LV with the lowest SCSI ID, which could be the local disk.

This procedure continues directly from [Creating a Boot Policy](#), on page 5.

Before You Begin

Create a boot policy to contain the SAN boot configuration.

**Note**

Beginning with Release 2.2, all SAN boot-related CLI commands have been moved to the SAN scope. Any existing scripts from previous releases that use SAN boot under the storage scope instead of **org/boot-policy/san** or **org/service-profile/boot-definition/san** should be updated.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# scope org <i>org-name</i>	Enters organization mode for the specified organization. To enter the root organization mode, type / as the <i>org-name</i> .
Step 2	UCS-A /org # scope boot-policy <i>policy-name</i>	Enters organization boot policy mode for the specified boot policy.
Step 3	UCS-A /org/boot-policy # create san	Creates a SAN boot for the boot policy and enters organization boot policy storage mode.
Step 4	UCS-A /org/boot-policy/san # set order <i>order_number</i>	Sets the boot order for the SAN boot. Enter an integer between 1 and 16.
Step 5	UCS-A /org/boot-policy/san # create san-image { primary secondary }	Creates a SAN image location, and if the san-image option is specified, enters organization boot policy storage SAN image mode. When using the enhanced boot order on Cisco UCS M3 servers, or M4 servers, the boot order that you define is used. For standard boot mode using the terms "primary" or "secondary" do not imply a boot order. The effective order of boot devices within the same device class is determined by the PCIe bus scan order.
Step 6	UCS-A /org/boot-policy/ssn/san-image # set vhba <i>vhba-name</i>	Specifies the vHBA to be used for the SAN boot.
Step 7	UCS-A /org/boot-policy/san/san-image # create path { primary secondary }	Creates a primary or secondary SAN boot path and enters organization boot policy SAN path mode. When using the enhanced boot order on Cisco UCS M3 servers, or M4 servers, the boot order that you define is used. For standard boot mode using the terms "primary" or "secondary" do not imply a boot order. The effective order of boot devices within the same device class is determined by the PCIe bus scan order.
Step 8	UCS-A /org/boot-policy/san/san-image/path # set { lun <i>lun-id</i> wwn <i>wwn-num</i> }	Specifies the LUN or WWN to be used for the SAN path to the boot image.
Step 9	UCS-A /org/boot-policy/san/san-image/path # commit-buffer	Commits the transaction to the system configuration.

The following example shows how to enter the boot policy named lab1-boot-policy, create a SAN boot for the policy, set the boot order to 1, create a primary SAN image, use a vHBA named vHBA2, create primary path using LUN 0, and commit the transaction:

```
UCS-A# scope org /
UCS-A /org* # scope boot-policy lab1-boot-policy
UCS-A /org/boot-policy # create san
UCS-A /org/boot-policy/san* # set order 1
UCS-A /org/boot-policy/san* # create san-image primary
UCS-A /org/boot-policy/san/san-image* # set vhba vHBA2
UCS-A /org/boot-policy/san/san-image* # create path primary
UCS-A /org/boot-policy/san/san-image/path* # set lun 0
UCS-A /org/boot-policy/san/san-image/path* # commit-buffer
UCS-A /org/boot-policy/san/san-image/path #
```

The following example shows how to create a SAN boot for the service profile SP_lab1, set the boot order to 1, create a primary SAN image, use a vHBA named vHBA2, create primary path using LUN 0, and commit the transaction:

```
UCS-A# scope org /
UCS-A /org* # scope service-profile SP_lab1
UCS-A /org/service-profile # create boot-definition
UCS-A /org/service-profile/boot-definition* # create san
UCS-A /org/service-profile/boot-definition/san* # create san-image primary
UCS-A /org/service-profile/boot-definition/san/san-image* # set vhba vHBA2
UCS-A /org/service-profile/boot-definition/san/san-image* # create path primary
UCS-A /org/service-profile/boot-definition/san/san-image/path* # set lun 0
UCS-A /org/service-profile/boot-definition/san/san-image/path* # commit-buffer
UCS-A /org/service-profile/boot-definition/san/san-image/path #
```

What to Do Next

Include the boot policy in a service profile and template.

iSCSI Boot

iSCSI boot enables a server to boot its operating system from an iSCSI target machine located remotely over a network.

iSCSI boot is supported on the following Cisco UCS hardware:

- Cisco UCS blade servers that have the Cisco UCS M51KR-B Broadcom BCM57711 network adapter and use the default MAC address provided by Broadcom.
- Cisco UCS M81KR Virtual Interface Card
- Cisco UCS VIC-1240 Virtual Interface Card
- Cisco UCS VIC-1280 Virtual Interface Card
- Cisco UCS rack servers that have the Cisco UCS M61KR-B Broadcom BCM57712 network adapter.
- Cisco UCS P81E Virtual Interface Card
- Cisco UCS VIC 1225 Virtual Interface Card on Cisco UCS rack servers

There are prerequisites that must be met before you configure iSCSI boot. For a list of these prerequisites, see [iSCSI Boot Guidelines and Prerequisites](#), on page 11.

For a high-level procedure for implementing iSCSI boot, see [Configuring iSCSI Boot](#), on page 14.

iSCSI Boot Process

Cisco UCS Manager uses the iSCSI vNIC and iSCSI boot information created for the service profile in the association process to program the adapter, located on the server. After the adapter is programmed, the server reboots with the latest service profile values. After the power on self-test (POST), the adapter attempts to initialize using these service profile values. If the adapter can use the values and log in to its specified target, the adapter initializes and posts an iSCSI Boot Firmware Table (iBFT) to the host memory and a valid bootable LUN to the system BIOS. The iBFT that is posted to the host memory contains the initiator and target configuration that is programmed on the primary iSCSI vNIC.

**Note**

Previously, the host could see only one of the boot paths configured, depending on which path completed the LUN discovery first, and would boot from that path. Now, when there are two iSCSI boot vNICs configured, the host sees both of the boot paths. So for multipath configurations, a single IQN must be configured on both the boot vNICs. If there are different IQNs configured on the boot vNICs on a host, the host boots with the IQN that is configured on the boot vNIC with the lower PCI order.

The next step, which is the installation of the operating system (OS), requires an OS that is iBFT capable. During installation of the OS, the OS installer scans the host memory for the iBFT table and uses the information in the iBFT to discover the boot device and create an iSCSI path to the target LUN. Some OSs require a NIC driver to complete this path. If this step is successful, the OS installer finds the iSCSI target LUN on which to install the OS.

**Note**

The iBFT works at the OS installation software level and might not work with HBA mode (also known as TCP offload). Whether iBFT works with HBA mode depends on the OS capabilities during installation. Also, for a server that includes a Cisco UCS M51KR-B Broadcom BCM57711 adapter, the iBFT normally works at a maximum transmission unit (MTU) size of 1500, regardless of the MTU jumbo configuration. If the OS supports HBA mode, you might need to set HBA mode, dual-fabric support, and jumbo MTU size after the iSCSI installation process.

iSCSI Boot Guidelines and Prerequisites

These guidelines and prerequisites must be met before configuring iSCSI boot:

- After the iSCSI boot policies are 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 iSCSI boot policies.
- To set up iSCSI boot from a Windows 2008 server where the second vNIC (failover vNIC) must boot from an iSCSI LUN, consult Microsoft Knowledge Base Article 976042. Microsoft has a known issue where Windows might fail to boot from an iSCSI drive or cause a bugcheck error if the networking hardware is changed. To work around this issue, follow the resolution recommended by Microsoft.
- The storage array must be licensed for iSCSI boot and the array side LUN masking must be properly configured.

- Two IP addresses must be determined, one for each iSCSI initiator. If possible, the IP addresses should be on the same subnet as the storage array. The IP addresses are assigned statically or dynamically using the Dynamic Host Configuration Protocol (DHCP).
- You cannot configure boot parameters in the Global boot policy. Instead, after configuring boot parameters, include the boot policy in the appropriate service profile.
- The operating system (OS) must be iSCSI Boot Firmware Table (iBFT) compatible.
- For Cisco UCS M51KR-B Broadcom BCM57711 network adapters:
 - Servers that use iSCSI boot must contain the Cisco UCS M51KR-B Broadcom BCM57711 network adapter. For information on installing or replacing an adapter card, see the *Cisco UCS B250 Extended Memory Blade Server Installation and Service Note*. The service note is accessible from the *Cisco UCS B-Series Servers Documentation Roadmap* at <http://www.cisco.com/go/unifiedcomputing/b-series-doc>.
 - Set the MAC addresses on the iSCSI device.
 - If you are using the DHCP Vendor ID (Option 43), configure the MAC address of an iSCSI device in `/etc/dhcpd.conf`.
 - HBA mode (also known as TCP offload) and the boot to target setting are supported. However, only Windows OS supports HBA mode during installation.
 - Before installing the OS, disable the boot to target setting in the iSCSI adapter policy, then after installing the OS, re-enable the boot to target setting.



Note Each time you change an adapter policy setting, the adapter reboots to apply the new setting.

- When installing the OS on the iSCSI target, the iSCSI target must be ordered *before* the device where the OS image resides. For example, if you are installing the OS on the iSCSI target from a CD, the boot order should be the iSCSI target and then the CD.
 - After the server is iSCSI booted, do not modify the Initiator Name, Target name, LUN, iSCSI device IP, or Netmask/gateway using the Broadcom tool.
 - Do not interrupt the POST (power on self-test) process or the Cisco UCS M51KR-B Broadcom BCM57711 network adapter will fail to initialize.
- For Cisco UCS M81KR Virtual Interface Card and Cisco UCS VIC-1240 Virtual Interface Card:
For Cisco UCS VIC-1240 Virtual Interface Card:
 - Do not set MAC addresses on the iSCSI device.
 - HBA mode and the boot to target setting are *not* supported.
 - When installing the OS on the iSCSI target, the iSCSI target must be ordered *after* the device where the OS image resides. For example, if you are installing the OS on the iSCSI target from a CD, the boot order should be the CD and then the iSCSI target.
 - If you are using the DHCP Vendor ID (Option 43), the MAC address of the overlay vNIC must be configured in `/etc/dhcpd.conf`.
 - After the server is iSCSI booted, do not modify the IP details of the overlay vNIC.

- The VMware ESX/ESXi operating system does not support storing a core dump file to an iSCSI boot target LUN. Dump files must be written to a local disk.

Initiator IQN Configuration

Cisco UCS uses the following rules to determine the initiator IQN for an adaptor iSCSI vNIC at the time a service profile is associated with a physical server:

- An initiator IQN at the service profile level *and* at the iSCSI vNIC level cannot be used together in a service profile.
- If an initiator IQN is specified at the service profile level, all of the adaptor iSCSI vNICs are configured to use the same initiator IQN, except in the case of DHCP Option 43, where the initiator IQN is set to empty on the adapter iSCSI vNIC.
- When an initiator IQN is set at the iSCSI vNIC level, the initiator IQN at the service profile level is removed, if one is present.
- If there are two iSCSI vNIC in a service profile and only one of them has the initiator IQN set, the second one is configured with the default IQN pool. You can change this configuration later. The only exception is if DHCP Option 43 is configured. In this case, the initiator IQN on the second iSCSI vNIC is removed during service profile association.



Note If you change an iSCSI vNIC to use the DHCP Option 43 by setting the vendor ID, it does not remove the initiator IQN configured at the service profile level. The initiator IQN at the service profile level can still be used by another iSCSI vNIC which does not use the DHCP Option 43.

Enabling MPIO on Windows

You can enable (MPIO) to optimize connectivity with storage arrays.



Note If you change the networking hardware, Windows might fail to boot from an iSCSI drive. For more information, see [Microsoft support Article ID: 976042](#).

Before You Begin

The server on which you enable the Microsoft Multipath I/O (MPIO) must have a Cisco VIC driver.

If there are multiple paths configured to the boot LUN, only one path should be enabled when the LUN is installed.

Procedure

-
- Step 1** In the service profile associated with the server, configure the primary iSCSI vNIC. For more information, see [Creating an iSCSI vNIC in a Service Profile](#), on page 24.

- Step 2** Using the primary iSCSI vNIC, install the Windows operating system on the iSCSI target LUN.
- Step 3** After Windows installation completes, enable MPIO on the host.
- Step 4** In the service profile associated with the server, add the secondary iSCSI vNIC to the boot policy. For more information, see [Creating an iSCSI Adapter Policy](#), on page 15.

Configuring iSCSI Boot

When you configure an adapter or blade in Cisco UCS to iSCSI boot from a LUN target, complete all of the following steps.

Procedure

	Command or Action	Purpose
Step 1	Configure the iSCSI boot adapter policy.	(Optional) For more information, see Creating an iSCSI Adapter Policy , on page 15.
Step 2	Configure the authentication profiles for the initiator and target.	(Optional) For more information, see Creating an Authentication Profile , on page 17.
Step 3	To configure the iSCSI initiator to use an IP address from a pool of IP addresses, add a block of IP addresses to the iSCSI initiator pool.	(Optional) For more information, see Adding a Block of IP Addresses to the Initiator Pool , on page 19.
Step 4	Create a boot policy that can be used in any service profile. Alternatively, you can create a local boot policy only for the specific service policy. However, Cisco recommends that you create a boot policy that can be shared with multiple service profiles.	For more information about creating a boot policy that can be used in any service profile, see Creating an iSCSI Adapter Policy , on page 15.
Step 5	If you created a boot policy that can be used in any service profile, assign it to the service profile. Otherwise, proceed to the next step.	For more information, see Creating a Service Profile Template .
Step 6	Configure an Ethernet vNIC in a service profile.	The Ethernet vNIC is used as the overlay vNIC for the iSCSI device. For more information, see Configuring a vNIC for a Service Profile .
Step 7	Create an iSCSI vNIC in a service profile.	For more information, see Creating an iSCSI vNIC in a Service Profile , on page 24.
Step 8	Set the iSCSI initiator to boot using a static IP Address, an IP address from an IP pool, or DHCP.	See either Creating an iSCSI Initiator that Boots Using a Static IP Address , on page 26, Creating an iSCSI Initiator that Boots Using an IP Address from

	Command or Action	Purpose
		an IP Pool , on page 28, or Creating an iSCSI Initiator that Boots Using DHCP , on page 30.
Step 9	Create an iSCSI static or auto target.	For more information, see either Creating an iSCSI Static Target , on page 37 or Creating an iSCSI Auto Target , on page 40.
Step 10	Associate the service profile with a server.	For more information, see Associating a Service Profile with a Blade Server or Server Pool .
Step 11	Verify the iSCSI boot operation.	For more information, see <i>Verifying iSCSI Boot</i> .
Step 12	Install the OS on the server.	For more information, see one of the following guides: <ul style="list-style-type: none"> • <i>Cisco UCS B-Series Blade Servers VMware Installation Guide</i> • <i>Cisco UCS B-Series Blade Servers Linux Installation Guide</i> • <i>Cisco UCS B-Series Blade Servers Windows Installation Guide</i>
Step 13	Boot the server.	

Creating an iSCSI Adapter Policy

Procedure

	Command or Action	Purpose
Step 1	UCS-A# scope org <i>org-name</i>	Enters organization mode for the specified organization. To enter the root organization mode, type <i>/</i> as the <i>org-name</i> .
Step 2	UCS-A /org # create iscsi-policy <i>policy-name</i>	Creates the iSCSI adapter policy.
Step 3	UCS-A /org/iscsi-policy # set descr <i>description</i>	(Optional) Provides a description for the iSCSI adapter policy.
Step 4	UCS-A /org/iscsi-policy # set iscsi-protocol-item connection-timeout <i>timeout-secs</i>	The number of seconds to wait until Cisco UCS assumes that the initial login has failed and the iSCSI adapter is unavailable. Enter an integer between 0 and 255. If you enter 0, Cisco UCS uses the value set in the adapter firmware (default: 15 seconds).

	Command or Action	Purpose
Step 5	UCS-A /org/iscsi-policy # set iscsi-protocol-item dhcp-timeout <i>timeout-secs</i>	The number of seconds to wait before the initiator assumes that the DHCP server is unavailable. Enter an integer between 60 and 300 (default: 60 seconds).
Step 6	UCS-A /org/iscsi-policy # set iscsi-protocol-item lun-busy-retry-count <i>num</i>	The number of times to retry the connection in case of a failure during iSCSI LUN discovery. Enter an integer between 0 and 60. If you enter 0, Cisco UCS uses the value set in the adapter firmware (default: 15 seconds).
Step 7	UCS-A /org/iscsi-policy # set iscsi-protocol-item tcp-time-stamp {no yes}	Specifies whether to apply a TCP timestamp. With this setting, transmitted packets are given a time stamp of when the packet was sent so that the packet's round-trip time can be calculated, when needed. This setting applies only to Cisco UCS M51KR-B Broadcom BCM57711 adapters.
Step 8	UCS-A /org/iscsi-policy # set iscsi-protocol-item hbamode {no yes}	Specifies whether to enable HBA mode. This option should only be enabled for servers with the Cisco UCS NIC M51KR-B adapter running the Windows operating system.
Step 9	UCS-A /org/iscsi-policy # set iscsi-protocol-item boottotarget {no yes}	Specifies whether to boot from the iSCSI target. This option only applies to servers with the Cisco UCS NIC M51KR-B adapter. It should be disabled until you have installed an operating system on the server.
Step 10	UCS-A /org/iscsi-policy # commit-buffer	Commits the transaction to the system configuration.

The following example shows how to create an iSCSI adapter policy called `iscsiboot`, set the connection timeout, DHCP timeout, and LUN busy retry count, apply a TCP timestamp, and commit the transaction:

```
UCS-A# scope org /
UCS-A /org # create iscsi-policy iscsiboot
UCS-A /org/iscsi-policy* # set iscsi-protocol-item connection-timeout 60
UCS-A /org/iscsi-policy* # set iscsi-protocol-item dhcp-timeout 200
UCS-A /org/iscsi-policy* # set iscsi-protocol-item lun-busy-retry-count 5
UCS-A /org/iscsi-policy* # set iscsi-protocol-item tcp-time-stamp yes
UCS-A /org/iscsi-policy* # set iscsi-protocol-item hbamode yes
UCS-A /org/iscsi-policy* # set iscsi-protocol-item boottotarget yes
UCS-A /org/iscsi-policy* # commit-buffer
UCS-A /org/iscsi-policy #
```

What to Do Next

Include the adapter policy in a service profile and template.

Deleting an iSCSI Adapter Policy

Procedure

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

The following example shows how to delete an iSCSI adapter policy named iscsi-adapter-pol and commit the transaction:

```
UCS-A# scope org /
UCS-A /org # delete iscsi-policy iscsi-adapter-pol
UCS-A /org* # commit-buffer
UCS-A /org #
```

Creating an Authentication Profile

If you use authentication for iSCSI boot, you need to create an authentication profile for both the initiator and target.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# scope org <i>org-name</i>	Enters organization mode for the specified organization. To enter the root organization mode, type / as the <i>org-name</i> .
Step 2	UCS-A /org # create auth-profile <i>profile-name</i>	Creates an authentication profile with the specified name. The name can be up to 16 alphanumeric characters.
Step 3	UCS-A /org/auth-profile* # set user-id <i>id-name</i>	Creates a log in for authentication.
Step 4	UCS-A /org/auth-profile* # set password	Creates a password for authentication.
Step 5	UCS-A /org/auth-profile* # commit-buffer	Commits the transaction to the system configuration.
Step 6	UCS-A /org/auth-profile* # exit	Exits the current mode.

	Command or Action	Purpose
Step 7	Repeat steps 2 through 6 to create an authentication profile for the target.	

The following example shows how to create an authentication profile for an initiator and target and commit the transaction:

```
UCS-A# scope org
UCS-A /org # create auth-profile InitAuth
UCS-A /org/auth-profile* # set user-id init
UCS-A /org/auth-profile* # set password
Enter a password:
Confirm the password:
UCS-A /org/auth-profile* # commit-buffer
UCS-A /org/auth-profile # exit
UCS-A /org # create auth-profile TargetAuth
UCS-A /org/auth-profile* # set user-id target
UCS-A /org/auth-profile* # set password
Enter a password:
Confirm the password:
UCS-A /org/auth-profile* # commit-buffer
UCS-A /org/auth-profile # exit
```

What to Do Next

Create an Ethernet vNIC to be used as the overlay vNIC for the iSCSI device, and then create an iSCSI vNIC.

Deleting an Authentication Profile

Procedure

	Command or Action	Purpose
Step 1	UCS-A# <code>scope org org-name</code>	Enters organization mode for the specified organization. To enter the root organization mode, type <code>/</code> as the <code>org-name</code> .
Step 2	UCS-A /org # <code>delete auth-profile auth-profile-name</code>	Deletes the specified authentication profile.
Step 3	UCS-A /org # <code>commit-buffer</code>	Commits the transaction to the system configuration.

The following example shows how to delete an authentication profile called `iscsi-auth` and commit the transaction:

```
UCS-A# scope org
UCS-A /org # delete auth-profile iscsi-auth
UCS-A /org* # commit-buffer
UCS-A /org #
```

Adding a Block of IP Addresses to the Initiator Pool

You can create a group of IP addresses to be used for iSCSI boot. Cisco UCS Manager reserves the block of IP addresses you specify.

The IP pool must not contain any IP addresses that were assigned as static IP addresses for a server or service profile.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# scope org <i>org-name</i>	Enters the organization mode for the specified organization. To enter the root organization mode, enter <i>/</i> as the <i>org-name</i> .
Step 2	UCS-A /org# scope ip-pool iscsi-initiator-pool	Enters the mode to specify an iSCSI initiator pool.
Step 3	UCS-A /org/ip-pool # set descr <i>description</i>	(Optional) Provides a description for the IP pool. 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	UCS-A /org/ip-pool # set assignmentorder { default sequential }	This can be one of the following: <ul style="list-style-type: none"> • default—Cisco UCS Manager selects a random identity from the pool. • sequential—Cisco UCS Manager selects the lowest available identity from the pool.
Step 5	UCS-A /org/ip-pool# create block <i>from_ip_address to_ip_address default_gateway subnet_mask</i>	Creates a block of IP addresses for the iSCSI initiator.
Step 6	UCS-A/org/ip-pool/block# show detail expand	(Optional) Shows the block of IP addresses that you have created.
Step 7	UCS-A /org/ip-pool/block # commit-buffer	Commits the transaction to the system configuration.

The following example shows how to create an IP initiator pool for the iSCSI vNIC and commit the transaction:

```
UCS-A # scope org /
UCS-A /org # scope ip-pool iscsi-initiator-pool
UCS-A /org/ip-pool # create block 40.40.40.10 40.40.40.50 40.40.40.1 255.0.0.0
UCS-A /org/ip-pool/block # show detail expand
Block of IP Addresses:
  From: 40.40.40.10
```

```
To: 40.40.40.50
Default Gateway: 40.40.40.1
Subnet Mask: 255.0.0.0
UCS-A /org/ip-pool/block # commit buffer
```

What to Do Next

Configure one or more service profiles or service profile templates to obtain the iSCSI initiator IP address from the iSCSI initiator IP pool.

Deleting a Block of IP Addresses from the Initiator Pool

Procedure

	Command or Action	Purpose
Step 1	UCS-A# scope org <i>org-name</i>	Enters the organization mode for the specified organization. To enter the root organization mode, enter / as the <i>org-name</i> .
Step 2	UCS-A /org# scope ip-pool iscsi-initiator-pool	Enters the mode to specify an iSCSI initiator pool.
Step 3	UCS-A /org/ip-pool# delete block <i>from_ip_address to_ip_address</i>	Deletes the specified block of IP addresses from the initiator pool.
Step 4	UCS-A/org/ip-pool/block# show detail expand	(Optional) Shows that the block of IP addresses has been deleted.
Step 5	UCS-A /org/ip-pool# commit buffer	Commits the transaction to the system configuration.

The following example shows how to delete a block of IP addresses from the initiator pool and commit the transaction:

```
UCS-A # scope org /
UCS-A /org # scope ip-pool iscsi-initiator-pool
UCS-A /org/ip-pool # delete block 40.40.40.10 40.40.40.50 40.40.40.1 255.0.0.0
UCS-A /org/ip-pool # show detail expand

IP Pool:
  Name: iscsi-initiator-pool
  Size: 0
  Assigned: 0
  Descr:
UCS-A /org/ip-pool # commit buffer
```

Creating an iSCSI Boot Policy

You can add up to two iSCSI vNICs per boot policy. One vNIC acts as the primary iSCSI boot source, and the other acts as the secondary iSCSI boot source.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# scope org <i>org-name</i>	Enters organization mode for the specified organization. To enter the root organization mode, type / as the <i>org-name</i> .
Step 2	UCS-A /org # create boot-policy <i>policy-name</i> [purpose { operational utility }]	<p>Creates a boot policy with the specified policy name, and enters organization boot policy mode.</p> <p>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.</p> <p>When you create the boot policy, specify the operational option. This ensures that the server boots from the operating system installed on the server. The utility options is reserved and should only be used if instructed to do so by a Cisco representative.</p>
Step 3	UCS-A /org/boot-policy # set descr <i>description</i>	<p>(Optional) Provides a description for the boot policy.</p> <p>Note If your description includes spaces, special characters, or punctuation, you must begin and end your description with quotation marks. The quotation marks do not appear in the description field of any show command output.</p>
Step 4	UCS-A /org/boot-policy # set enforce-vnic-name { no yes }	<p>(Optional) If you choose yes, Cisco UCS Manager reports whether the device name specified in the boot policy matches what is specified in the service profile.</p> <p>If you choose no, Cisco UCS Manager uses any vNIC, vHBA, or iSCSI device from the service profile and does not report whether the device name specified in the boot policy matches what is specified in the service profile.</p>
Step 5	UCS-A /org/boot-policy # set reboot-on-update { no yes }	<p>Specifies whether the servers using this boot policy are automatically rebooted after you make changes to the boot order.</p> <p>In the Cisco UCS Manager GUI, if the Reboot on Boot Order Change check box is checked for a boot policy, and if CD-ROM or Floppy is the last device in the boot order, deleting or adding the device does not directly affect the boot order and the server does not reboot.</p>
Step 6	UCS-A /org/boot-policy # create iscsi	Adds an iSCSI boot to the boot policy.
Step 7	UCS-A /org/boot-policy/iscsi # create path { primary secondary }	Specifies the primary and secondary paths that Cisco UCS Manager uses to reach the iSCSI target . With iSCSI boot, you

	Command or Action	Purpose
		set up two paths. Cisco UCS Manager uses the primary path first, and if that fails, then it uses the secondary path.
Step 8	UCS-A /org/boot-policy/iscsi/path # create iscsivnicname <i>iscsi-vnic-name</i>	Creates an iSCSI vNIC.
Step 9	UCS-A /org/boot-policy/iscsi/path # exit	Exits iSCSI path mode.
Step 10	UCS-A /org/boot-policy/iscsi/path # set order order-num	Specifies the order for the iSCSI boot in the boot order.
Step 11	Repeat steps 8-10 to create secondary iSCSI vNICs.	(Optional)
Step 12	UCS-A /org/boot-policy/iscsi # commit-buffer	Commits the transaction to the system configuration.

The following example shows how to create an iSCSI boot policy named `iscsi-boot-policy-LAN`, provide a description for the boot policy, specify that servers using this policy are not automatically rebooted when the boot order is changed, set the boot order for iSCSI boot to 2, create an iSCSI boot and associate it with a vNIC called `iscsienic1`, and commit the transaction:

```
UCS-A# scope org /
UCS-A /org* # create boot-policy iscsi-boot-policy-LAN purpose operational
UCS-A /org/boot-policy* # set descr "Boot policy that boots from iSCSI."
UCS-A /org/boot-policy* # set enforce-vnic-name yes
UCS-A /org/boot-policy* # set reboot-on-update no
UCS-A /org/boot-policy* # create iscsi
UCS-A /org/boot-policy/iscsi* # create path primary
UCS-A /org/boot-policy/iscsi/path* # set iscsivnicname iscsienic1
UCS-A /org/boot-policy/iscsi/path* # exit
UCS-A /org/boot-policy/iscsi* # set order 2
UCS-A /org/boot-policy/iscsi* # commit-buffer
UCS-A /org/boot-policy #
```

What to Do Next

Include the boot policy in a service profile and template.

After a server is associated with a service profile that includes this boot policy, you can verify the actual boot order in the **Boot Order Details** area on the **General** tab for the server.

Deleting iSCSI Devices from a Boot Policy

Procedure

	Command or Action	Purpose
Step 1	UCS-A# scope org <i>org-name</i>	Enters organization mode for the specified organization. To enter the root organization mode, type / as the <i>org-name</i> .
Step 2	UCS-A /org # scope boot-policy <i>boot-pol-name</i>	Enters boot policy organization mode for the specified boot policy.
Step 3	UCS-A /org/boot-policy # delete iscsi	Deletes the iSCSI boot from the boot policy.
Step 4	UCS-A /org/boot-policy # commit-buffer	Commits the transaction to the system configuration.

The following example shows how to delete an iSCSI boot from the boot policy named boot-policy-iscsi and commit the transaction:

```
UCS-A# scope org /
UCS-A /org # scope boot-policy boot-policy-iscsi
UCS-A /org/boot-policy # delete iscsi
UCS-A /org/boot-policy* # commit-buffer
UCS-A /org/boot-policy #
```

Setting an Initiator IQN at the Service Profile Level

In a service profile, you can create an initiator with a specific IQN or one that is derived from a pool of IQNs.

Before You Begin

You cannot delete an IQN using the CLI.

To understand the initiator IQN configuration guidelines, see [Initiator IQN Configuration](#), on page 13.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# scope org <i>org-name</i>	Enters organization mode for the specified organization. To enter the root organization mode, type / as the <i>org-name</i> .
Step 2	UCS-A /org # scope service-profile <i>profile-name</i>	Enters service profile organization mode for the service profile.

	Command or Action	Purpose
Step 3	UCS-A /org/service-profile# set iscsi-identity { initiator name <i>initiator-name</i> initiator-pool-name <i>pool-name</i> }	Creates an initiator with the specified name. The name can be up to 16 alphanumeric characters.
Step 4	UCS-A /org/service-profile* # commit buffer	Commits the transaction to the system configuration.
Step 5	UCS-A /org/auth-profile* # exit	Exits the current mode.

The following example shows how to create a specific name for an iSCSI initiator and commit the transaction:

```
UCS-A# scope org /
UCS-A /org # scope service-profile accounting
UCS-A /org/service-profile # set iscsi-identity initiator-name manual:IQN
UCS-A /org/service-profile* # commit-buffer
```

Creating an iSCSI vNIC in a Service Profile

You can create an iSCSI vNIC in a service profile.

Before You Begin

You must have an Ethernet vNIC in a service profile to be used as the overlay vNIC for the iSCSI device.

Procedure

	Command or Action	Purpose
Step 1	UCS-A # scope org <i>org-name</i>	Enters organization mode for the specified organization. To enter the root organization mode, type / as the <i>org-name</i> .
Step 2	UCS-A /org # scope service-profile <i>profile-name</i>	Enters service profile organization mode for the service profile.
Step 3	UCS-A /org/service-profile # create vnic-iscsi <i>iscsi-vnic-name</i> .	Specifies the iSCSI vNIC name.
Step 4	UCS-A /org/service-profile/vnic-iscsi* # set iscsi-adaptor-policy <i>iscsi-adaptor-name</i>	(Optional) Specifies the iSCSI adaptor policy that you have created for this iSCSI vNIC.
Step 5	UCS-A /org/service-profile/vnic-iscsi* # set auth-name <i>authentication-profile-name</i>	(Optional) Sets the authentication profile to be used by the iSCSI vNIC. The authentication profile must already exist for it to be set. For more information, see Creating an Authentication Profile , on page 17.

	Command or Action	Purpose
Step 6	UCS-A /org/service-profile/vnic-iscsi* # set identity { dynamic-mac { <i>dynamic-mac-address</i> derived } mac-pool <i>mac-pool-name</i> }	Specifies the MAC address for the iSCSI vNIC. Note The MAC address is only set for Cisco UCS NIC M51KR-B adapters.
Step 7	UCS-A /org/service-profile/vnic-iscsi* # set iscsi-identity { initiator-name <i>initiator-name</i> initiator-pool-name <i>iqn-pool-name</i> }	Specifies the name of the iSCSI initiator or the name of an IQN pool from which the iSCSI initiator name will be provided. The iSCSI initiator name can be up to 223 characters.
Step 8	UCS-A /org/service-profile/vnic-iscsi* # set overlay-vnic-name <i>overlay-vnic-name</i>	Specifies the Ethernet vNIC that is used by the iSCSI device as the overlay vNIC. For more information, see Configuring a vNIC for a Service Profile .
Step 9	UCS-A /org/service-profile/vnic-iscsi* # create eth-if	Creates an Ethernet interface for a VLAN assigned to the iSCSI vNIC.
Step 10	UCS-A /org/service-profile/vnic-iscsi/eth-if* # set vllanname <i>vlan-name</i> .	Specifies the VLAN name. The default VLAN is default. For the Cisco UCS M81KR Virtual Interface Card and the Cisco UCS VIC-1240 Virtual Interface Card, the VLAN that you specify must be the same as the native VLAN on the overlay vNIC. For the Cisco UCS M51KR-B Broadcom BCM57711 adapter, the VLAN that you specify can be any VLAN assigned to the overlay vNIC.
Step 11	UCS-A /org/service-profile/vnic-iscsi # commit-buffer	Commits the transaction to the system configuration.

The following example shows how to create an iSCSI vNIC called scsivnic1, add it to an existing service profile called accounting, and commit the transaction:

```
UCS-A# scope org /
UCS-A /org # scope service-profile accounting
UCS-A /org/service-profile # create vnic-iscsi iSCSI1
UCS-A /org/service-profile/vnic-iscsi* # set iscsi-adaptor-policy iscsiboot
UCS-A /org/service-profile/vnic-iscsi* # set auth-name initauth
UCS-A /org/service-profile/vnic-iscsi* # set identity dynamic-mac derived
UCS-A /org/service-profile/vnic-iscsi* # set iscsi-identity initiator-name iSCSI1
UCS-A /org/service-profile/vnic-iscsi* # set overlay-vnic-name eth1
UCS-A /org/service-profile/vnic-iscsi* # create eth-if
UCS-A /org/service-profile/vnic-iscsi/eth-if* # set vllanname default
UCS-A /org/service-profile/vnic-iscsi/eth-if* # commit buffer
```

What to Do Next

Configure an iSCSI initiator to boot using a static IP address, an IP address from a configured IP pool, or DHCP.

Deleting an iSCSI vNIC from a Service Profile

Procedure

	Command or Action	Purpose
Step 1	UCS-A# scope org <i>org-name</i>	Enters organization mode for the specified organization. To enter the root organization mode, type / as the <i>org-name</i> .
Step 2	UCS-A /org # scope service-profile <i>profile-name</i>	Enters service profile organization mode for the service profile.
Step 3	UCS-A /org/service-profile # delete vnic-iscsi <i>iscsi-vnic-name</i>	Deletes the specified iSCSI vNIC from the specified service profile.
Step 4	UCS-A /org/service-profile # commit-buffer	Commits the transaction to the system configuration.

The following example shows how to delete an iSCSI vNIC called scsivnic1 and commit the transaction:

```
UCS-A# scope org /
UCS-A /org # scope service-profile accounting
UCS-A /org/service-profile # delete vnic-iscsi scsivnic1
UCS-A /org/service-profile* # commit-buffer
UCS-A /org/service-profile #
```

Creating an iSCSI Initiator that Boots Using a Static IP Address

In a service profile, you can create an iSCSI initiator and configure it to boot using a static IP address.

Before You Begin

You have completed the following:

- Created iSCSI overlay vNICs in a service profile.
- Created an iSCSI vNIC in a service profile.

Procedure

	Command or Action	Purpose
Step 1	UCS-A # scope org <i>org-name</i>	Enters organization mode for the specified organization. To enter the root organization mode, type / as the <i>org-name</i> .

	Command or Action	Purpose
Step 2	UCS-A /org # scope service-profile <i>profile-name</i>	Enters service profile organization mode for the service profile.
Step 3	UCS-A /org/service-profile # scope vnic-iscsi <i>iscsi-vnic-name</i>	Enters the configuration mode for the specified iSCSI vNIC.
Step 4	UCS-A /org/service-profile/iscsi-boot/vnic-iscsi # create ip-if	Creates an IP interface.
Step 5	UCS-A /org/service-profile/iscsi-boot/vnic-iscsi/eth-if/ip-if* # enter static-ip-params	Specifies that you are entering static IP boot parameters.
Step 6	UCS-A /org/service-profile/iscsi-boot/vnic-iscsi/eth-if/ip-if/static-ip-params* # set addr <i>ip-address</i>	Specifies the static IP address.
Step 7	UCS-A /org/service-profile/iscsi-boot/vnic-iscsi/eth-if/ip-if/static-ip-params* # set default-gw <i>ip-address</i>	Specifies the default gateway IP address.
Step 8	UCS-A /org/service-profile/iscsi-boot/vnic-iscsi/eth-if/ip-if/static-ip-params* # set primary-dns <i>ip-address</i>	Specifies the primary DNS IP address.
Step 9	UCS-A /org/service-profile/iscsi-boot/vnic-iscsi/eth-if/ip-if/static-ip-params* # set secondary-dns <i>ip-address</i>	Specifies the secondary DNS IP address.
Step 10	UCS-A /org/service-profile/iscsi-boot/vnic-iscsi/eth-if/ip-if/static-ip-params* # set subnet <i>subnet-ip-address</i>	Specifies the subnet mask.
Step 11	UCS-A /org/service-profile/iscsi-boot/vnic-iscsi/eth-if/ip-if/static-ip-params* # commit buffer	Commits the transaction to the system configuration.

The following example shows how to configure the initiator to boot using a static IP address and commit the transaction:

```
UCS-A # scope org
UCS-A /org # scope service-profile accounting
UCS-A /org/service-profile # scope vnic-iscsi iSCSI1
UCS-A /org/service-profile/iscsi-boot/vnic-iscsi # create ip-if
UCS-A /org/service-profile/iscsi-boot/vnic-iscsi/ip-if* # enter static-ip-params
UCS-A /org/service-profile/iscsi-boot/vnic-iscsi/ip-if/static-ip-params* # set addr
10.104.105.193
UCS-A /org/service-profile/iscsi-boot/vnic-iscsi/ip-if/static-ip-params* # set default-gw
10.104.105.1
UCS-A /org/service-profile/iscsi-boot/vnic-iscsi/ip-if/static-ip-params* # set primary-dns
11.11.11.100
UCS-A /org/service-profile/iscsi-boot/vnic-iscsi/ip-if/static-ip-params* # set secondary-dns
11.11.11.100
UCS-A /org/service-profile/iscsi-boot/vnic-iscsi/ip-if/static-ip-params* # set subnet
```

```
255.255.255.0
```

```
UCS-A /org/service-profile/iscsi-boot/vnic-iscsi/ip-if/static-ip-params* # commit-buffer
```

What to Do Next

Create an iSCSI target.

Deleting the Static IP Address Boot Parameters from an iSCSI Initiator

In a service profile, you can delete the static IP address boot parameters from an iSCSI initiator.

Procedure

	Command or Action	Purpose
Step 1	UCS-A # scope org <i>org-name</i>	Enters organization mode for the specified organization. To enter the root organization mode, type / as the <i>org-name</i> .
Step 2	UCS-A /org # scope service-profile <i>profile-name</i>	Enters service profile organization mode for the service profile.
Step 3	UCS-A /org/service-profile # scope vnic-iscsi <i>iscsi-vnic-name</i>	Enters the configuration mode for the specified iSCSI vNIC.
Step 4	UCS-A /org/service-profile/iscsi-boot/vnic-iscsi # scope ip-if	Enters the configuration mode for an IP interface.
Step 5	UCS-A /org/service-profile/iscsi-boot/vnic-iscsi/ip-if* # delete static-ip-params	Deletes the static IP boot parameters from an initiator.
Step 6	UCS-A /org/service-profile/iscsi-boot/vnic-iscsi/ip-if/static-ip-params* # commit buffer	Commits the transaction to the system configuration.

The following example shows how to delete the static IP address boot parameters from the initiator and commit the transaction:

```
UCS-A # scope org
UCS-A /org # scope service-profile accounting
UCS-A /org/service-profile # scope vnic-iscsi iSCSI1
UCS-A /org/service-profile/iscsi-boot/vnic-iscsi # scope ip-if
UCS-A /org/service-profile/iscsi-boot/vnic-iscsi/ip-if # delete static-ip-params
UCS-A /org/service-profile/iscsi-boot/vnic-iscsi/ip-if* # commit-buffer
```

Creating an iSCSI Initiator that Boots Using an IP Address from an IP Pool

In a service profile, you can create an iSCSI initiator and configure it to boot using an IP address from an IP pool that you have created.

Before You Begin

You have completed the following:

- Created an overlay vNIC in a service profile
- Created an iSCSI vNIC in a service profile.

Procedure

	Command or Action	Purpose
Step 1	UCS-A # scope org <i>org-name</i>	Enters organization mode for the specified organization. To enter the root organization mode, type / as the <i>org-name</i> .
Step 2	UCS-A /org # scope service-profile <i>profile-name</i>	Enters service profile organization mode for the service profile.
Step 3	UCS-A /org/service-profile # scope iscsi-boot	Enters the configuration mode for configuring iSCSI boot parameters.
Step 4	UCS-A /org/service-profile/iscsi-boot # scope vnic-iscsi <i>iscsi-vnic-name</i>	Enters the configuration mode for the specified iSCSI vNIC.
Step 5	UCS-A /org/service-profile/iscsi-boot/vnic-iscsi* # scope ip-if	Enters the configuration mode for the iSCSI Ethernet interface.
Step 6	UCS-A /org/service-profile/iscsi-boot/vnic-iscsi/ip-if* # enter pooled-ip-params	Specifies that the iSCSI initiator boot using one of the IP addresses from the previously created iSCSI initiator IP pool.
Step 7	UCS-A /org/service-profile/iscsi-boot/vnic-iscsi/ip-if/pooled-ip-params* # commit buffer	Commits the transaction to the system configuration.

The following example shows how to create an iSCSI initiator and configure it to boot using an IP address from an IP pool:

```
UCS-A # scope org
UCS-A /org # scope service-profile accounting
UCS-A /org/service-profile/iscsi-boot # scope vnic-iscsi iSCSI1
UCS-A /org/service-profile/iscsi-boot/vnic-iscsi # scope ip-if
UCS-A /org/service-profile/iscsi-boot/vnic-iscsi/ip-if* # enter pooled-ip-params
UCS-A /org/service-profile/iscsi-boot/vnic-iscsi/ip-if/pooled-ip-params* # commit buffer
```

What to Do Next

Create an iSCSI target.

Deleting the IP Pool Boot Parameter from an iSCSI Initiator

In a service profile, you can create an iSCSI initiator and configure it to boot using an IP address from an IP pool that you have created.

Procedure

	Command or Action	Purpose
Step 1	UCS-A # scope org <i>org-name</i>	Enters organization mode for the specified organization. To enter the root organization mode, type <i>/</i> as the <i>org-name</i> .
Step 2	UCS-A /org # scope service-profile <i>profile-name</i>	Enters service profile organization mode for the service profile.
Step 3	UCS-A /org/service-profile # scope iscsi-boot	Enters the configuration mode for configuring the iSCSI boot parameters.
Step 4	UCS-A /org/service-profile/iscsi-boot/ # scope vnic-iscsi <i>iscsi-vnic-name</i>	Enters the configuration mode for the specified iSCSI vNIC.
Step 5	UCS-A /org/service-profile/iscsi-boot/vnic-iscsi # enter ip-if	Enters the configuration mode for an IP interface.
Step 6	UCS-A /org/service-profile/iscsi-boot/vnic-iscsi/ip-if* # delete pooled-ip-params	Specifies that the iSCSI initiator does not use an IP address from an IP pool to boot.
Step 7	UCS-A /org/service-profile/iscsi-boot/vnic-iscsi/ip-if/pooled-ip-params* # commit buffer	Commits the transaction to the system configuration.

The following example shows how to delete the boot using an IP address from an IP pool parameter and commit the transaction:

```
UCS-A # scope org
UCS-A /org # scope service-profile accounting
UCS-A /org/service-profile # scope iscsi-boot
UCS-A /org/service-profile/iscsi-boot # scope vnic-iscsi iSCSI1
UCS-A /org/service-profile/iscsi-boot/vnic-iscsi # enter ip-if
UCS-A /org/service-profile/iscsi-boot/vnic-iscsi/ip-if* # delete pooled-ip-params
UCS-A /org/service-profile/iscsi-boot/vnic-iscsi/ip-if/pooled-ip-params* # commit buffer
```

Creating an iSCSI Initiator that Boots Using DHCP

In a service profile, you can create an iSCSI initiator and configure it to boot using DHCP.

Before You Begin

You have completed the following:

- Created iSCSI overlay vNICs in a service profile.
- Created an iSCSI vNIC in a service profile.

Procedure

	Command or Action	Purpose
Step 1	UCS-A # scope org <i>org-name</i>	Enters organization mode for the specified organization. To enter the root organization mode, type / as the <i>org-name</i> .
Step 2	UCS-A /org # scope service-profile <i>profile-name</i>	Enters service profile organization mode for the service profile.
Step 3	UCS-A /org/service-profile # scope iscsi-boot	Enters the configuration mode for configuring iSCSI boot parameters.
Step 4	UCS-A /org/service-profile/iscsi-boot # scope vnic-iscsi <i>iscsi-vnic-name</i>	Enters the configuration mode for the specified iSCSI vNIC.
Step 5	UCS-A /org/service-profile/iscsi-boot/vnic-iscsi # create ip-if	Creates an IP interface.
Step 6	UCS-A /org/service-profile/iscsi-boot/vnic-iscsi/ip-if* # create dhcp-ip-params	Specifies that you are setting the initiator to boot using DHCP.
Step 7	UCS-A /org/service-profile/iscsi-boot/vnic-iscsi/ip-if/dhcp-ip-params* # commit buffer	Commits the transaction to the system configuration.

The following example shows how to configure the initiator to boot using DHCP and commit the transaction:

```
UCS-A # scope org
UCS-A /org # scope service-profile accounting
UCS-A /org/service-profile # scope iscsi-boot
UCS-A /org/service-profile/iscsi-boot # scope vnic-iscsi iSCSI1
UCS-A /org/service-profile/iscsi-boot/vnic-iscsi # create ip-if
UCS-A /org/service-profile/iscsi-boot/vnic-iscsi/ip-if* # create dhcp-ip-params
UCS-A /org/service-profile/iscsi-boot/vnic-iscsi/ip-if/dhcp-ip-params* # commit-buffer
```

What to Do Next

Create an iSCSI target.

Deleting the DHCP Boot Parameter from an iSCSI Initiator

In a service profile, you can remove the DHCP boot parameter from an iSCSI initiator.

Procedure

	Command or Action	Purpose
Step 1	UCS-A # scope org <i>org-name</i>	Enters organization mode for the specified organization. To enter the root organization mode, type / as the <i>org-name</i> .
Step 2	UCS-A /org # scope service-profile <i>profile-name</i>	Enters service profile organization mode for the service profile.
Step 3	UCS-A /org/service-profile # scope iscsi-boot	Enters the configuration mode for configuring iSCSI boot parameters.
Step 4	UCS-A /org/service-profile/iscsi-boot # scope vnic-iscsi <i>iscsi-vnic-name</i>	Enters the configuration mode for the specified iSCSI vNIC.
Step 5	UCS-A /org/service-profile/iscsi-boot/vnic-iscsi # enter ip-if	Enters the configuration mode for an IP interface.
Step 6	UCS-A /org/service-profile/iscsi-boot/vnic-iscsi/ip-if* # delete dhcp-ip-params	Specifies that the initiator does not use DHCP to boot.
Step 7	UCS-A /org/service-profile/iscsi-boot/vnic-iscsi/ip-if/dhcp-ip-params* # commit buffer	Commits the transaction to the system configuration.

The following example shows how to delete the boot using DHCP parameter and commit the transaction:

```
UCS-A # scope org
UCS-A /org # scope service-profile accounting
UCS-A /org/service-profile # scope iscsi-boot
UCS-A /org/service-profile/iscsi-boot # scope vnic-iscsi iSCSI1
UCS-A /org/service-profile/iscsi-boot/vnic-iscsi # enter ip-if
UCS-A /org/service-profile/iscsi-boot/vnic-iscsi/ip-if* # delete dhcp-ip-params
UCS-A /org/service-profile/iscsi-boot/vnic-iscsi/ip-if/dhcp-ip-params* # commit-buffer
```

IQN Pools

An IQN pool is a collection of iSCSI Qualified Names (IQNs) for use as initiator identifiers by iSCSI vNICs in a Cisco UCS domain.

IQN pool members are of the form *prefix:suffix:number*, where you can specify the prefix, suffix, and a block (range) of numbers.

An IQN pool can contain more than one IQN block, with different number ranges and different suffixes, but sharing the same prefix.

Creating an IQN Pool


Note

In most cases, the maximum IQN size (prefix + suffix + additional characters) is 223 characters. When using the Cisco UCS NIC M51KR-B adapter, you must limit the IQN size to 128 characters.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# scope org <i>org-name</i>	Enters the organization mode for the specified organization. To enter the root organization mode, enter / as the <i>org-name</i> .
Step 2	UCS-A /org # create iqn-pool <i>pool-name</i>	Creates an IQN pool with the specified pool name and enters organization IQN pool mode. 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	UCS-A /org/iqn-pool # set iqn-prefix <i>prefix</i>	Specifies the prefix for the IQN block members. Unless limited by the adapter card, the prefix can contain up to 150 characters.
Step 4	UCS-A /org/iqn-pool # set descr <i>description</i>	(Optional) Provides a description for the IQN pool. Enter up to 256 characters. 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/iqn-pool # set assignmentorder { default sequential }	This can be one of the following: <ul style="list-style-type: none"> • default—Cisco UCS Manager selects a random identity from the pool. • sequential—Cisco UCS Manager selects the lowest available identity from the pool.
Step 6	UCS-A /org/iqn-pool # create block <i>suffix from to</i>	Creates a block (range) of IQNs, and enters organization IQN pool block mode. You must specify the base suffix, the starting suffix number, and the ending suffix number. The resulting IQN pool members are of the form <i>prefix:suffix:number</i> . The suffix can be up to 64 characters. Note An IQN pool can contain more than one IQN block. To create multiple blocks, enter multiple create block commands from organization IQN pool mode.
Step 7	UCS-A /org/iqn-pool/block # commit-buffer	Commits the transaction to the system configuration.

The following example shows how to create an IQN pool named pool4, provide a description for the pool, specify a prefix and a block of suffixes to be used for the pool, and commit the transaction:

```
UCS-A# scope org /
UCS-A /org # create iqn-pool pool4
UCS-A /org/iqn-pool* # set iqn-prefix iqn.alpha.com
UCS-A /org/iqn-pool* # set descr "This is IQN pool 4"
UCS-A /org/iqn-pool* # create block beta 3 5
UCS-A /org/iqn-pool/block* # commit-buffer
UCS-A /org/iqn-pool/block #
```

What to Do Next

Include the IQN suffix pool in a service profile and template.

Adding a Block to an IQN Pool

Procedure

	Command or Action	Purpose
Step 1	UCS-A# scope org <i>org-name</i>	Enters organization mode for the specified organization. To enter the root organization mode, type / as the <i>org-name</i> .
Step 2	UCS-A /org # scope iqn-pool <i>pool-name</i>	Enters organization IQN pool mode for the specified pool.
Step 3	UCS-A /org/iqn-pool # create block <i>suffix from to</i>	Creates a block (range) of IQN suffixes, and enters organization IQN pool block mode. You must specify the base suffix, the starting suffix number, and the ending suffix number. The resulting IQN pool members are of the form <i>prefix:suffix:number</i> . Note An IQN pool can contain more than one IQN block. To create multiple blocks, enter multiple create block commands from organization IQN pool mode.
Step 4	UCS-A /org/iqn-pool/block # commit-buffer	Commits the transaction to the system configuration.
Step 5	UCS-A /org/iqn-pool/block # exit	(Optional) Returns to organization IQN pool mode.
Step 6	UCS-A /org/iqn-pool # show block	(Optional) Displays the blocks of suffixes.

This example shows how to add a block of IQN suffixes to an IQN pool named pool4 and commit the transaction:

```
UCS-A# scope org /
UCS-A /org # scope iqn-pool pool4
UCS-A /org/iqn-pool # create block beta 3 5
```

```

UCS-A /org/iqn-pool/block* # commit-buffer
UCS-A /org/iqn-pool/block # exit
UCS-A /org/iqn-pool # show block
Block of IQN Names:
  Suffix      From  To
  -----
  beta                3   5
UCS-A /org/iqn-pool #

```

Deleting a Block from an IQN Pool

If you delete an address block from a pool, Cisco UCS Manager does not reallocate any addresses in that block that were assigned to vNICs or vHBAs. All assigned addresses from a deleted block 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 <i>org-name</i>	Enters organization mode for the specified organization. To enter the root organization mode, type / as the <i>org-name</i> .
Step 2	UCS-A /org # scope iqn-pool <i>pool-name</i>	Enters organization IQN pool mode for the specified pool.
Step 3	UCS-A /org/iqn-pool # delete block <i>suffix from to</i>	Deletes a block (range) of IQNs. You must specify the base suffix and the first and last numbers in the block to be deleted.
Step 4	UCS-A /org/iqn-pool # commit-buffer	Commits the transaction to the system configuration.

This example shows how to delete a block of suffixes from an IQN pool named pool4 and commit the transaction:

```

UCS-A# scope org /
UCS-A /org # scope iqn-pool pool4
UCS-A /org/iqn-pool # delete block beta 0 12
UCS-A /org/iqn-pool* # commit-buffer
UCS-A /org/iqn-pool #

```

Deleting an IQN 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 <i>org-name</i>	Enters the organization mode for the specified organization. To enter the root organization mode, enter / as the <i>org-name</i> .
Step 2	UCS-A /org # delete iqn-pool <i>pool-name</i>	Deletes the specified IQN pool.
Step 3	UCS-A /org # commit-buffer	Commits the transaction to the system configuration.

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

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

Viewing IQN Pool Usage

Procedure

	Command or Action	Purpose
Step 1	UCS-A# scope org <i>org-name</i>	Enters the organization mode for the specified organization. To enter the root organization mode, enter / as the <i>org-name</i> .
Step 2	UCS-A /org # scope iqn-pool <i>pool-name</i>	Enters organization IQN pool mode for the specified pool.
Step 3	UCS-A /org/iqn-pool # show pooled	Displays the assignments of the IQN block members.

The following example shows how to display the assignments of suffixes in the IQN pool named pool4:

```
UCS-A# scope org /
UCS-A /org # scope iqn-pool pool4
UCS-A /org/iqn-pool # show pooled
Pooled:
  Name           Assigned Assigned To Dn
  -----
  beta:3         No
  beta:4         No
  beta:5         No
```

UCS-A /org/iqn-pool #

Creating an iSCSI Static Target

You can create a static target.

Before You Begin

You have already created an iSCSI vNIC.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# scope org <i>org-name</i>	Enters organization mode for the specified organization. To enter the root organization mode, type / as the organization name.
Step 2	UCS-A /org # scope service-profile <i>profile-name</i>	Enters service profile organization mode for the service profile to which you want to add an iSCSI target.
Step 3	UCS-A /org/service-profile # scope iscsi-boot	Enters the mode for configuring iSCSI boot parameters.
Step 4	UCS-A /org/service-profile/iscsi-boot # scope vnic-iscsi <i>iscsi-vnic-name</i>	Enters the iSCSI vNIC mode for the specified vNIC.
Step 5	UCS-A /org/service-profile/iscsi-boot/vnic-iscsi # create static-target-if {1 2}	Creates a static target for the iSCSI vNIC and assigns a priority level to it. Valid priority levels are 1 or 2.
Step 6	UCS-A /org/service-profile/iscsi-boot/vnic-iscsi/static-target-if # set name <i>name</i>	A regular expression that defines the iSCSI Qualified Name (IQN) or Extended Unique Identifier (EUI) name for the iSCSI target. You can enter any alphanumeric characters as well as the following special characters: <ul style="list-style-type: none"> • . (period) • : (colon) • - (dash) <p>Important This name must be properly formatted according to standard IQN or EUI guidelines. The following examples show properly formatted target names:</p> <ul style="list-style-type: none"> • iqn.2001-04.com.example • iqn.2001-04.com.example:storage.diskarrays-sn • iqn.2001-04.com.example:storage.tape1.sy • iqn.2001-04.com.example:storage.disk2.sy • eui.02004567A425678D

	Command or Action	Purpose
Step 7	UCS-A /org/service-profile/iscsi-boot/vnic-iscsi/static-target-if # set port <i>port-num</i>	The port associated with the iSCSI target. Enter an integer between 1 and 65535. The default is
Step 8	UCS-A /org/service-profile/iscsi-boot/vnic-iscsi/static-target-if # set auth-name <i>auth-profile</i>	(Optional) If you need the target to authenticate itself and have an authentication profile, you need to specify the name of the authentication profile. The name of the associated iSCSI authentication profile.
Step 9	UCS-A /org/service-profile/iscsi-boot/vnic-iscsi/static-target-if # set ipaddress <i>ipv4-address</i>	The IPv4 address assigned to the iSCSI target.
Step 10	UCS-A /org/service-profile/iscsi-boot/vnic-iscsi/static-target-if # create lun	Creates the LUN that corresponds to the location of the interface.
Step 11	UCS-A /org/service-profile/iscsi-boot/vnic-iscsi/static-target-if/lun* # set id <i>id-number</i>	Specifies the target LUN id. Valid values are from 0 to 65535.
Step 12	UCS-A /org/service-profile/iscsi-boot/vnic-iscsi/static-target-if/lun* # exit	Exits the current configuration mode.
Step 13	UCS-A /org/service-profile/iscsi-boot/vnic-iscsi/static-target-if # exit	Exits the current configuration mode.
Step 14	UCS-A /org/service-profile/iscsi-boot/vnic-iscsi # commit-buffer	Commits the transaction to the system configuration database.
Step 15	Repeat steps 5 through 14 to create a second static target.	(Optional)

The following example shows how to create two iSCSI static target interfaces and commit the transaction:

```
UCS-A # scope org test
UCS-A /org # scope service-profile accounting
UCS-A /org/service-profile # scope iscsi-boot
UCS-A /org/service-profile/iscsi-boot # scope vnic-iscsi iSCSI1
UCS-A /org/service-profile/iscsi-boot/vnic-iscsi # create static-target-if 1
UCS-A /org/service-profile/iscsi-boot/vnic-iscsi/static-target-if* # set name statictarget1
UCS-A /org/service-profile/iscsi-boot/vnic-iscsi/static-target-if* # set port 3260
UCS-A /org/service-profile/iscsi-boot/vnic-iscsi/static-target-if* # set auth-name
authprofile1
UCS-A /org/service-profile/iscsi-boot/vnic-iscsi/static-target-if* # set ip-address
192.168.10.10
UCS-A /org/service-profile/iscsi-boot/vnic-iscsi/static-target-if* # create lun
UCS-A /org/service-profile/iscsi-boot/vnic-iscsi/static-target-if/lun* # set id 1
UCS-A /org/service-profile/iscsi-boot/vnic-iscsi/static-target-if/lun* # exit
UCS-A /org/service-profile/iscsi-boot/vnic-iscsi/static-target-if* # exit
UCS-A /org/service-profile/iscsi-boot/vnic-iscsi # commit-buffer
UCS-A /org/service-profile/iscsi-boot/vnic-iscsi # create static-target-if 2
UCS-A /org/service-profile/iscsi-boot/vnic-iscsi/static-target-if* # set ipaddress
192.168.10.11
UCS-A /org/service-profile/iscsi-boot/vnic-iscsi/static-target-if* # set name statictarget2
```

```

UCS-A /org/service-profile/iscsi-boot/vnic-iscsi/static-target-if* # set port 3260
UCS-A /org/service-profile/iscsi-boot/vnic-iscsi/static-target-if* # set auth-name
authprofile1
UCS-A /org/service-profile/iscsi-boot/vnic-iscsi/static-target-if* # create lun
UCS-A /org/service-profile/iscsi-boot/vnic-iscsi/static-target-if/lun* # set id 1
UCS-A /org/service-profile/iscsi-boot/vnic-iscsi/static-target-if/lun* # exit
UCS-A /org/service-profile/iscsi-boot/vnic-iscsi/static-target-if* # exit
UCS-A /org/service-profile/iscsi-boot/vnic-iscsi # commit-buffer

```

What to Do Next

To configure a second iSCSI device, repeat the steps for creating an iSCSI vNIC, initiator, and target.

Deleting an iSCSI Static Target

You can delete an iSCSI static target. However, you must have at least one iSCSI static target remaining after you delete one. Therefore, you must have two iSCSI static targets in order to delete one of them.



Note

If you have two iSCSI targets and you delete the first priority target, the second priority target becomes the first priority target, although the Cisco UCS Manager still shows it as the second priority target.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# scope org <i>org-name</i>	Enters organization mode for the specified organization. To enter the root organization mode, type / as the <i>org-name</i> .
Step 2	UCS-A /org # scope service-profile <i>profile-name</i>	Enters service profile organization mode for the service profile to which you want to add an iSCSI target.
Step 3	UCS-A /org/service-profile # scope iscsi-boot	Enters the mode for configuring iSCSI boot parameters.
Step 4	UCS-A /org/service-profile/iscsi-boot # scope vnic-iscsi <i>iscsi-vnic-name</i>	Enters the iSCSI vNIC mode for the specified vNIC name.
Step 5	UCS-A /org/service-profile/iscsi-boot/vnic-iscsi # delete static-target-if	Deletes the static target for the iSCSI vNIC.
Step 6	UCS-A /org/service-profile/iscsi-boot/vnic-iscsi # commit-buffer	Commits the transaction to the system configuration.

The following example shows how to delete an iSCSI static target and commit the transaction:

```

UCS-A # scope org test
UCS-A /org # scope service-profile sample
UCS-A /org # scope iscsi-boot

```

```
UCS-A /org/service-profile/iscsi-boot # scope vnic-iscsi trial
UCS-A /org/service-profile/iscsi-boot/vnic-iscsi # delete static-target-if 1
UCS-A /org/service-profile/iscsi-boot/vnic-iscsi # commit-buffer
UCS-A /org/service-profile/iscsi-boot/vnic-iscsi #
```

Creating an iSCSI Auto Target

You can create an iSCSI auto target with or without the vendor IDs.

Before You Begin

These prerequisites must be met before creating iSCSI auto target:

- You have already created an iSCSI vNIC in a service profile.
- You have considered the prerequisites for the VIC that you are using. For more information, see [iSCSI Boot Guidelines and Prerequisites](#), on page 11

Procedure

	Command or Action	Purpose
Step 1	UCS-A# scope org <i>org-name</i>	Enters organization mode for the specified organization. To enter the root organization mode, type / as the <i>org-name</i> .
Step 2	UCS-A /org # scope service-profile <i>profile-name</i>	Enters service profile organization mode for the service profile that you want to add an iSCSI target interface to.
Step 3	UCS-A /org # scope iscsi-boot Example:	Enters the mode for configuring iSCSI boot parameters.
Step 4	UCS-A /org/service-profile/iscsi-boot # scope vnic-iscsi <i>iscsi-vnic-name</i>	Enters iSCSI vNIC service profile organization mode for the specified vNIC name.
Step 5	UCS-A /org/service-profile/iscsi-boot/vnic-iscsi/ # create auto-target-if	Creates an auto target for the iSCSI vNIC. If you plan to use an auto target without the vendor ID, you must configure an initiator name. For more information, see Creating an iSCSI vNIC in a Service Profile , on page 24.
Step 6	UCS-A /org/service-profile/iscsi-boot/vnic-iscsi/auto-target-if* # set dhcp-vendor-id <i>vendor-id</i>	(Optional) Sets a vendor ID for the auto target. The vendor ID can be up to 32 alphanumeric characters.
Step 7	UCS-A /org/service-profile/iscsi-boot/vnic-iscsi/auto-target-if* # exit	Exists the current configuration mode.

	Command or Action	Purpose
Step 8	UCS-A /org/service-profile/iscsi-boot/vnic-iscsi # commit-buffer	Commits the transaction to the system configuration.

The following example shows how to create an iSCSI auto target *without* a vendor ID and commit the transaction:

```
UCS-A # scope org
UCS-A /org # scope service-profile accounting
UCS-A /org/service-profile # scope iscsi-boot
UCS-A /org/service-profile/iscsi-boot # scope vnic-iscsi iSCSI1
UCS-A /org/service-profile/iscsi-boot/vnic-iscsi # create auto-target-if
UCS-A /org/service-profile/iscsi-boot/vnic-iscsi/auto-target-if* # exit
UCS-A /org/service-profile/iscsi-boot/vnic-iscsi # commit-buffer
```

The following example shows how to create an iSCSI auto target *with* a vendor ID and commit the transaction:

```
UCS-A # scope org
UCS-A /org # scope service-profile accounting
UCS-A /org/service-profile # scope iscsi-boot
UCS-A /org/service-profile/iscsi-boot # scope vnic-iscsi iSCSI1
UCS-A /org/service-profile/iscsi-boot/vnic-iscsi # create auto-target-if
UCS-A /org/service-profile/iscsi-boot/vnic-iscsi/auto-target-if* # set dhcp-vendor-id
iSCSI_Vendor
UCS-A /org/service-profile/iscsi-boot/vnic-iscsi/auto-target-if* # exit
UCS-A /org/service-profile/iscsi-boot/vnic-iscsi # commit-buffer
```

What to Do Next

To configure a second iSCSI device, repeat the steps for creating an iSCSI vNIC, initiator, and target.

Deleting an iSCSI Auto Target

You can delete an auto target only if you have a static target set.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# scope org <i>org-name</i>	Enters organization mode for the specified organization. To enter the root organization mode, type / as the <i>org-name</i> .
Step 2	UCS-A /org # scope service-profile <i>profile-name</i>	Enters the service profile mode for the service profile to which you want to add an iSCSI target.
Step 3	UCS-A /org/service-profile # scope iscsi-boot	Enters the mode for configuring iSCSI boot parameters.
Step 4	UCS-A /org/service-profile/iscsi-boot # scope vnic-iscsi <i>iscsi-vnic-name</i>	Enters the iSCSI vNIC mode for the specified vNIC name.

	Command or Action	Purpose
Step 5	UCS-A /org/service-profile/iscsi-boot/vnic-iscsi # delete auto-target-if	Deletes the auto target.
Step 6	UCS-A /org/service-profile/iscsi-boot/vnic-iscsi # commit-buffer	Commits the transaction to the system configuration.

The following example shows how to delete an iSCSI auto target and commit the transaction:

```
UCS-A # scope org test
UCS-A /org # scope service-profile accounting
UCS-A /org/service-profile # scope iscsi-boot
UCS-A /org/service-profile/iscsi-boot # scope vnic-iscsi iSCSI1
UCS-A /org/service-profile/iscsi-boot/vnic-iscsi # delete auto-target-if
UCS-A /org/service-profile/iscsi-boot/vnic-iscsi # commit-buffer
```

Verifying iSCSI Boot

Use the KVM console to view the boot up messages as the adapter is booting. For information on how to access the KVM console, see the *Starting the KVM Console* chapter.

This step can only be performed using the Cisco UCS Manager GUI. For more information, see the *Starting the KVM Console* chapter in the *UCS Manager GUI Configuration Guide*.

- For the Cisco UCS M51KR-B Broadcom BCM57711, the following message appears:
Logging in the 1st iSCSI Target.... Succeeded.
- For the Cisco UCS M81KR Virtual Interface Card, the following message appears:
Option ROM installed successfully.

LAN Boot

You can configure a boot policy to boot one or more servers from a centralized provisioning server on the LAN. A LAN (or PXE) boot is frequently used to install operating systems on a server from that LAN server.

You can add more than one type of boot device to a LAN boot policy. For example, you could add a local disk or virtual media boot as a secondary boot device.

Configuring a LAN Boot for a Boot Policy

Before You Begin

Create a boot policy to contain the LAN boot configuration.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# scope org <i>org-name</i>	Enters organization mode for the specified organization. To enter the root organization mode, type / as the <i>org-name</i> .
Step 2	UCS-A /org # scope boot-policy <i>policy-name</i>	Enters organization boot policy mode for the specified boot policy.
Step 3	UCS-A /org/boot-policy # create lan	Creates a LAN boot for the boot policy and enters organization boot policy LAN mode.
Step 4	UCS-A /org/boot-policy/lan # set order {1 2 3 4}	Specifies the boot order for the LAN boot.
Step 5	UCS-A /org/boot-policy/lan # create path {primary secondary}	Creates a primary or secondary LAN boot path and enters organization boot policy LAN path mode.
Step 6	UCS-A /org/boot-policy/lan/path # set vnic <i>vnic-name</i>	Specifies the vNIC to use for the LAN path to the boot image.
Step 7	UCS-A /org/boot-policy/lan/path # commit-buffer	Commits the transaction to the system configuration.

The following example enters the boot policy named lab2-boot-policy, creates a LAN boot for the policy, sets the boot order to 2, creates primary and secondary paths using the vNICs named vNIC1 and vNIC2 , and commits the transaction:

```
UCS-A# scope org /
UCS-A /org* # scope boot-policy lab2-boot-policy
UCS-A /org/boot-policy* # create lan
UCS-A /org/boot-policy/lan* # set order 2
UCS-A /org/boot-policy/lan* # create path primary
UCS-A /org/boot-policy/lan/path* # set vnic vNIC1
UCS-A /org/boot-policy/lan/path* # exit
UCS-A /org/boot-policy/lan* # create path secondary
UCS-A /org/boot-policy/lan/path* # set vnic vNIC2
UCS-A /org/boot-policy/lan/path* # commit-buffer
UCS-A /org/boot-policy/lan/path #
```

What to Do Next

Include the boot policy in a service profile and template.

Local Devices Boot

Cisco UCS Manager allows you to boot from different local devices.

**Note**

For Cisco UCS M3 and higher blade and rack servers using enhanced boot order, you can select both top-level and second-level boot devices. For Cisco UCS M1 and M2 blade and rack servers using standard boot order, you can only select a top-level device.

Local Disk Boot

If a server has a local drive, you can configure a boot policy to boot the server from the top-level local disk device or from any of the following second-level devices:

- Local LUN—Enables boot from local disk or local LUN.
- Local JBOD—Enables boot from a bootable JBOD.
- SD card—Enables boot from SD card.
- Internal USB—Enables boot for internal USB.
- External USB—Enables boot from external USB.
- Embedded Local LUN—Enables boot from the embedded local LUN on the Cisco UCS 240 M4 server.
- Embedded Local Disk—Enables boot from the embedded local disk on the Cisco UCS C240 M4SX and the M4L servers.

**Note**

Second-level devices are only available for Cisco UCS M3 and higher blade and rack servers using enhanced boot order. For Cisco UCS M1 and M2 blade and rack servers using standard boot order, you can choose only the top-level **Add Local Disk**.

Virtual Media Boot

You can configure a boot policy to boot one or more servers from a virtual media device that is accessible from the server. A virtual media device mimics the insertion of a physical CD/DVD disk (read-only) or floppy disk (read-write) into a server. This type of server boot is typically used to manually install operating systems on a server.

**Note**

Second-level devices are only available for Cisco UCS M3 and higher blade and rack servers using enhanced boot order. For Cisco UCS M1 and M2 blade and rack servers using standard boot order, you can choose only the top-level **Add CD/DVD** or **Add Floppy**.

Remote Virtual Drive Boot

You can configure a boot policy to boot one or more servers from a remote virtual drive that is accessible from the server.

Configuring a Local Disk Boot for a Boot Policy

You can also create a local boot policy that is restricted to a service profile or service profile template. However, Cisco recommends that you create a global boot policy that can be included in multiple service profiles or service profile templates.

You can add more than one type of boot device to a boot policy. For example, you could add a virtual media boot as a secondary boot device.



Note

Beginning with Release 2.2, if you want to add any top-level local storage device to the boot order, you must use **create local-any** after the **create local** command. If you have any policies from previous releases that contain a local storage device, they will be modified to use local-any during upgrade.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# scope org <i>org-name</i>	Enters organization mode for the specified organization. To enter the root organization mode, type / as the <i>org-name</i> .
Step 2	UCS-A /org # scope boot-policy <i>policy-name</i>	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-any local-lun sd-card usb-extern usb-intern }	<p>Specifies the type of local storage. This can be one of the following:</p> <ul style="list-style-type: none"> • local-any—Any type of local storage device. This option can be used in either legacy or UEFI boot mode. <p>Note Cisco UCS M1 and M2 blade and rack servers using standard boot order can only use local-any.</p> <ul style="list-style-type: none"> • local-lun—A local hard disk drive. • sd-card—An SD card. • usb-extern—An external USB card. • usb-intern—An internal USB card. <p>For Cisco UCS M3 and higher blade and rack servers using enhanced boot order, you can select both top-level and second-level boot devices. For Cisco UCS M1 and M2 blade and rack</p>

	Command or Action	Purpose
		servers using standard boot order, you can only select a top-level device.
Step 6	UCS-A /org/boot-policy/storage/local/local-storage-device # set order order_number	Sets the boot order for the specified local storage device. Enter an integer between 1 and 16. When using the enhanced boot order on Cisco UCS M3 servers, or M4 servers, the boot order that you define is used. For standard boot mode using the terms "primary" or "secondary" do not imply a boot order. The effective order of boot devices within the same device class is determined by the PCIe bus scan order.
Step 7	UCS-A /org/boot-policy/storage/local/local-storage-device # commit-buffer	Commits the transaction to the system configuration.

The following example shows how to create a boot policy named lab1-boot-policy, create a local hard disk drive boot for the policy, set the boot order to 3, 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/sd-card* # set order 3
UCS-A /org/boot-policy/storage/local/sd-card* # commit-buffer
UCS-A /org/boot-policy/storage/local/sd-card #
```

The following example shows how to create a local SD card boot for the service profile SP_lab1, set the boot order to 3, and commit the transaction:

```
UCS-A# scope org /
UCS-A /org* # scope service-profile SP_lab1
UCS-A /org/service-profile # create boot-definition
UCS-A /org/service-profile/boot-definition* # create storage
UCS-A /org/service-profile/boot-definition/storage* # create local
UCS-A /org/service-profile/boot-definition/storage/local* # create sd-card
UCS-A /org/service-profile/boot-definition/storage/local* # set order 3
UCS-A /org/service-profile/boot-definition/storage/local* # commit-buffer
UCS-A /org/service-profile/boot-definition/storage/local #
```

The following example shows how to create any top-level local device boot for the service profile SP_lab1, set the boot order to 3, and commit the transaction:

```
UCS-A# scope org /
UCS-A /org* # scope service-profile SP_lab1
UCS-A /org/service-profile # create boot-definition
UCS-A /org/service-profile/boot-definition* # create storage
UCS-A /org/service-profile/boot-definition/storage* # create local
UCS-A /org/service-profile/boot-definition/storage/local* # create local-any
UCS-A /org/service-profile/boot-definition/storage/local/local-any* # set order 3
UCS-A /org/service-profile/boot-definition/storage/local/local-any* # commit-buffer
UCS-A /org/service-profile/boot-definition/storage/local/local-any #
```

What to Do Next

Include the boot policy in a service profile and template.

Configuring a Virtual Media Boot for a Boot Policy



Note Virtual Media requires the USB to be enabled. If you modify the BIOS settings that affect the USB functionality, you also affect the Virtual Media. Therefore, Cisco recommends that you leave the following USB BIOS defaults for best performance:

- Make Device Non Bootable—set to **disabled**
- USB Idle Power Optimizing Setting—set to **high-performance**

Before You Begin

Create a boot policy to contain the virtual media boot configuration.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# scope org <i>org-name</i>	Enters organization mode for the specified organization. To enter the root organization mode, type / as the <i>org-name</i> .
Step 2	UCS-A /org # scope boot-policy <i>policy-name</i>	Enters organization boot policy mode for the specified boot policy.
Step 3	UCS-A /org/boot-policy # create virtual-media { read-only read-only-local read-only-remote read-write read-write-drive read-write-local read-write-remote }	<p>Creates the specified virtual media boot for the boot policy and enters organization boot policy virtual media mode. This can be one of the following:</p> <ul style="list-style-type: none"> • read-only—Local or remote CD/DVD. This option can be used in either legacy or UEFI boot mode. • read-only-local—Local CD/DVD. • read-only-remote—Remote CD/DVD. • read-write—Local or remote floppy disk drive. This option can be used in either legacy or UEFI boot mode. • read-write-drive—Remote USB drive. • read-write-local—Local floppy disk drive. • read-write-remote—Remote floppy disk drive. <p>Note For Cisco UCS M3 and higher blade and rack servers using enhanced boot order, you can select both top-level and second-level boot devices. For Cisco UCS M1 and M2 blade and rack servers using standard boot order, you can only select a top-level device.</p>

	Command or Action	Purpose
Step 4	UCS-A /org/boot-policy/virtual-media # set order order_number	Sets the boot order for the virtual-media boot. Enter an integer between 1 and 16.
Step 5	UCS-A /org/boot-policy/virtual-media # commit-buffer	Commits the transaction to the system configuration.

The following example shows how to enter the boot policy named lab3-boot-policy, create a CD/DVD virtual media boot, set the boot order to 3, and commit the transaction:

```
UCS-A# scope org /
UCS-A /org* # scope boot-policy lab3-boot-policy
UCS-A /org/boot-policy* # create virtual-media read-only-local
UCS-A /org/boot-policy/virtual-media* # set order 3
UCS-A /org/boot-policy/virtual-media* # commit-buffer
```

What to Do Next

Include the boot policy in a service profile and template.

Creating a CIMC vMedia Boot Policy

You can also create a local boot policy that is restricted to a service profile or service profile template. However, Cisco recommends that you create a global boot policy that can be included in multiple service profiles or service profile templates.

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 <i>org-name</i> .
Step 2	UCS-A /org # create boot-policy policy-name	Creates a boot policy with the specified policy name, and enters organization boot policy mode.
Step 3	UCS-A /org/boot-policy* # create virtual-media ?	Displays a list of local and remote devices to your can access and boot.
Step 4	UCS-A /org/boot-policy* # create virtual-media {access vMediaMappingName}	Displays a list of local and remote devices to your can access and boot.
Step 5	UCS-A /org/boot-policy* # create virtual-media read-write-remote-drive vMediaMap0}	Creates vMedia Boot Device configuration for specified vMedia.
Step 6	UCS-A /org/boot-policy/virtual-media* # commit-buffer	Commits the transaction to the system configuration.

	Command or Action	Purpose
Step 7	UCS-A /org/boot-policy/virtual-media* # show detail expand	Displays the following boot order. Boot virtual media: Order: 1 Access: Read Write Remote vMedia Drive Name: vmediaMap0

The following example creates a CIMC vMedia boot policy.

```
UCS-A# scope org /
UCS-A /org* # create boot-policy boot-policy vm-vmediamap-boot
UCS-A /org/boot-policy* # create virtual-media
```

Viewing a CIMC vMedia Mount

Procedure

	Command or Action	Purpose
Step 1	UCS-A# scope server chassis_id/blade_id	Enters chassis server mode for the specified server.
Step 2	UCS-A# /chassis/server # scope cimc	Enters CIMC mode.
Step 3	UCS-A /chassis/server/cimc # show vmedia-mapping-list detail expand	Displays the vMedia mapping details.

The following example shows how to view a CIMC vMedia mount.

```
UCS-A# scope server 1/2
UCS-A /chassis/server # scope cimc
UCS-A /chassis/server/cimc # show vmedia-mapping-list detail expand

vMedia Mapping List:
vMedia Mapping:
Disk Id: 1
Mapping Name: cdd
Device Type: Cdd
Remote IP: 172.31.1.167
Image Path: cifs
Image File Name: ubuntu-14.11-desktop-i386.iso
Mount Protocol: Cifs
Mount Status: Mounted
Error: None
Password:
User ID: Administrator

UCS-A /chassis/server/cimc #
```

Configuring an EFI Shell Boot for a Boot Policy

You can create a boot policy with an EFI Shell as the boot device. Booting from an EFI Shell prevents loss of data and provides more options to script, debug, and control various booting scenarios. EFI Shell is supported as a boot device only in the **Uefi** boot mode.

Before You Begin

To configure EFI Shell as a boot device, ensure that the boot mode is set to **Uefi**.



Important In an EFI Shell boot policy, If you edit the boot mode to **Legacy**, Cisco UCS Manager removes the EFI Shell boot device and sets the boot policy to default.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# scope org <i>org-name</i>	Enters the organization mode for the specified organization. To enter the root organization mode, enter / as the <i>org-name</i> .
Step 2	UCS-A /org # create boot-policy <i>policy-name</i>	Creates a boot policy with the specified policy name, and enters organization boot policy mode.
Step 3	UCS-A /org/boot-policy* # set boot-mode { legacy uefi }	Specifies whether the servers using this boot policy are using UEFI or legacy boot mode. Note To configure EFI Shell as a boot device, ensure that the boot mode is set to Uefi
Step 4	UCS-A /org/boot-policy* # create efi-shell	Creates an EFI Shell boot for the boot policy and enters organization boot policy mode.
Step 5	UCS-A /org/boot-policy/efi-shell* # commit-buffer	Commits the transaction to the system configuration.

```
UCS-A# scope org
UCS-A /org # create boot-policy efi_shell
UCS-A /org/boot-policy* # set boot-mode uefi
UCS-A /org/boot-policy* # create efi-shell
UCS-A /org/boot-policy/efi-shell* # commit-buffer
```

What to Do Next

Include the boot policy in a service profile and template.

Deleting a Boot Policy

Procedure

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

The following example deletes the boot policy named boot-policy-LAN and commits the transaction:

```
UCS-A# scope org /
UCS-A /org # delete boot-policy boot-policy-LAN
UCS-A /org* # commit-buffer
UCS-A /org #
```

UEFI Boot Parameters

UEFI boot mode for servers is dependent on information that is stored on the platform hardware. The boot entry, which contains information about the UEFI OS boot loader, is stored in the BIOS flash of the server. In Cisco UCS Manager releases earlier than Release 2.2(4), when a service profile is migrated from one server to another server, the boot loader information is not available on the destination server. Hence, the BIOS cannot load the boot loader information for the server to boot in UEFI boot mode.

Cisco UCSM Release 2.2(4) introduces UEFI boot parameters to provide the BIOS with information about the location of the UEFI OS boot loader on the destination server from where the BIOS loads it. Now, the server can use the boot loader information and boot in UEFI boot mode.

Guidelines and Limitations for UEFI Boot Parameters

- You can configure UEFI boot parameters only if the boot mode is UEFI.
- When you upgrade Cisco UCS Manager to Release 2.2(4), UEFI boot failure during service profile migration is not handled automatically. You must explicitly create the UEFI boot parameters in the target device to successfully boot to the UEFI-capable OS.
- UEFI boot parameters are supported on all M3 and higher servers that support second-level boot order.
- You can specify UEFI boot parameters for the following device types:
 - SAN LUN
 - iSCSI LUN
 - Local LUN

- UEFI boot parameters are specific to each operating system. You can specify UEFI boot parameters for the following operating systems:
 - VMware ESX
 - SuSE Linux
 - Microsoft Windows
 - Red Hat Enterprise Linux 7

Configuring UEFI Boot Parameters for a Local LUN

Before You Begin

Ensure that the boot mode for the local LUN is set to UEFI.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# scope org <i>org-name</i>	Enters organization mode for the specified organization. To enter the root organization mode, type / as the <i>org-name</i> .
Step 2	UCS-A /org # scope boot-policy <i>policy-name</i>	Enters organization boot policy mode for the specified boot policy.
Step 3	UCS-A /org/boot-policy # scope storage	Enters organization boot policy storage mode for the boot policy.
Step 4	UCS-A /org/boot-policy/storage # scope local	Enters the boot policy local storage mode.
Step 5	UCS-A /org/boot-policy/storage/local/ # scope { local-any local-lun sd-card usb-extern usb-intern }	Specifies the type of local storage. This can be one of the following: <ul style="list-style-type: none"> • local-any—Any type of local storage device. This option

Command or Action	Purpose
	<p>can be used in either legacy or UEFI boot mode.</p> <p>Note Cisco UCS M1 and M2 blade and rack servers using standard boot order can only use <code>local-lun</code>.</p> <ul style="list-style-type: none"> • <code>local-lun</code>—A local hard disk drive. • <code>sd-card</code>—An SD card. • <code>usb-extern</code>—An external USB card. • <code>usb-intern</code>—An internal USB card. <p>Important The only type of local storage for which you can configure UEFI boot parameters is <code>local-lun</code>.</p>

	Command or Action	Purpose
Step 6	UCS-A /org/boot-policy/storage/local/local-lun # scope local-lun-image-path {primary secondary}	Enters the image path for the local LUN.
Step 7	UCS-A /org/boot-policy/storage/local/local-lun/local-lun-image-path # create uefi-boot-param	Creates UEFI boot parameters and enters UEFI boot parameter mode.
Step 8	UCS-A /org/boot-policy/storage/local/local-lun/local-lun-image-path/uefi-boot-param* # set bootloader-name <i>name</i>	Sets the name of the boot loader.
Step 9	UCS-A /org/boot-policy/storage/local/local-lun/local-lun-image-path/uefi-boot-param* # set bootloader-path <i>path</i>	Sets the path of the boot loader.
Step 10	UCS-A /org/boot-policy/storage/local/local-lun/local-lun-image-path/uefi-boot-param* # set boot-description " <i>description</i> "	Sets a description for the boot loader.
Step 11	UCS-A /org/boot-policy/storage/local/local-lun/local-lun-image-path/uefi-boot-param* # commit-buffer	Commits the transaction to the system configuration.

The following example shows how to create UEFI boot parameters for a local LUN, and commit the transaction:

```
UCS-A# scope org /
UCS-A /org* # scope boot-policy bpl
UCS-A /org/boot-policy* # scope storage
UCS-A /org/boot-policy/storage* # scope local
UCS-A /org/boot-policy/storage/local* # scope local-lun
UCS-A /org/boot-policy/storage/local/local-lun # scope local-lun-image-path primary
UCS-A /org/boot-policy/storage/local/local-lun/local-lun-image-path # create uefi-boot-param
UCS-A /org/boot-policy/storage/local/local-lun/local-lun-image-path/uefi-boot-param* # set
  bootloader-name grub.efi
UCS-A /org/boot-policy/storage/local/local-lun/local-lun-image-path/uefi-boot-param* # set
  bootloader-path EFI\redhat
UCS-A /org/boot-policy/storage/local/local-lun/local-lun-image-path/uefi-boot-param* # set
  boot-description "Red Hat Enterprise Linux"
UCS-A /org/boot-policy/storage/local/local-lun/local-lun-image-path/uefi-boot-param* #
commit-buffer
```

Configuring UEFI Boot Parameters for an iSCSI LUN

Before You Begin

Ensure that the boot mode for the iSCSI LUN is set to UEFI.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# scope org <i>org-name</i>	Enters organization mode for the specified organization. To enter the root organization mode, type / as the <i>org-name</i> .
Step 2	UCS-A /org # scope boot-policy <i>policy-name</i>	Enters organization boot policy mode for the specified boot policy.
Step 3	UCS-A /org/boot-policy # scope iscsi	Enters organization boot policy iSCSI mode for the boot policy.
Step 4	UCS-A /org/boot-policy/iscsi # scope path { primary secondary }	Enters the image path for the iSCSI LUN.
Step 5	UCS-A /org/boot-policy/iscsi/path # create uefi-boot-param	Creates UEFI boot parameters and enters UEFI boot parameter mode.
Step 6	UCS-A /org/boot-policy/iscsi/path/uefi-boot-param* # set bootloader-name <i>name</i>	Sets the name of the boot loader.
Step 7	UCS-A /org/boot-policy/iscsi/path/uefi-boot-param* # set bootloader-path <i>path</i>	Sets the path of the boot loader.
Step 8	UCS-A /org/boot-policy/iscsi/path/uefi-boot-param* # set boot-description " <i>description</i> "	Sets a description for the boot loader.
Step 9	UCS-A /org/boot-policy/iscsi/path/uefi-boot-param* # commit-buffer	Commits the transaction to the system configuration.

The following example shows how to create UEFI boot parameters for an iSCSI LUN, and commit the transaction:

```
UCS-A# scope org /
UCS-A /org* # scope boot-policy bp2
UCS-A /org/boot-policy* # scope iscsi
UCS-A /org/boot-policy/iscsi # scope path primary
UCS-A /org/boot-policy/iscsi/path # create uefi-boot-param
UCS-A /org/boot-policy/iscsi/path/uefi-boot-param* # set bootloader-name grub.efi
UCS-A /org/boot-policy/iscsi/path/uefi-boot-param* # set bootloader-path EFI\redhat
UCS-A /org/boot-policy/iscsi/path/uefi-boot-param* # set boot-description "Red Hat Enterprise Linux"
UCS-A /org/boot-policy/iscsi/path/uefi-boot-param* # commit-buffer
```

Configuring UEFI Boot Parameters for a SAN LUN

Before You Begin

Ensure that the boot mode for the SAN LUN is set to UEFI.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# scope org <i>org-name</i>	Enters organization mode for the specified organization. To enter the root organization mode, type / as the <i>org-name</i> .
Step 2	UCS-A /org # scope boot-policy <i>policy-name</i>	Enters organization boot policy mode for the specified boot policy.
Step 3	UCS-A /org/boot-policy # scope san	Enters organization boot policy SAN mode for the boot policy.
Step 4	UCS-A /org/boot-policy/san # scope san-image { primary secondary }	Enters the SAN image.
Step 5	UCS-A /org/boot-policy/san/san-image # scope path { primary secondary }	Enters the image path for the SAN LUN.
Step 6	UCS-A /org/boot-policy/san/san-image/path # create uefi-boot-param	Creates UEFI boot parameters and enters UEFI boot parameter mode.
Step 7	UCS-A /org/boot-policy/san/san-image/path/uefi-boot-param* # set bootloader-name <i>name</i>	Sets the name of the boot loader.
Step 8	UCS-A /org/boot-policy/san/san-image/path/uefi-boot-param* # set bootloader-path <i>path</i>	Sets the path of the boot loader.
Step 9	UCS-A /org/boot-policy/san/san-image/path/uefi-boot-param* # set boot-description " <i>description</i> "	Sets a description for the boot loader.
Step 10	UCS-A /org/boot-policy/san/san-image/path/uefi-boot-param* # commit-buffer	Commits the transaction to the system configuration.

The following example shows how to create UEFI boot parameters for a SAN LUN, and commit the transaction:

```
UCS-A# scope org /
UCS-A /org* # scope boot-policy bp3
UCS-A /org/boot-policy* # scope san
UCS-A /org/boot-policy/san # scope san-image primary
UCS-A /org/boot-policy/san/san-image # scope path primary
UCS-A /org/boot-policy/san/san-image/path # create uefi-boot-param
UCS-A /org/boot-policy/san/san-image/path/uefi-boot-param* # set bootloader-name grub.efi
```



```
UCS-A /org/boot-policy/san/san-image/path/uefi-boot-param* # set bootloader-path EFI\redhat
UCS-A /org/boot-policy/san/san-image/path/uefi-boot-param* # set boot-description "Red Hat
Enterprise Linux"
UCS-A /org/boot-policy/san/san-image/path/uefi-boot-param* # commit-buffer
```

