Server Policies

This chapter includes the following sections:

Configuring Server Pools

Server Pools

A server pool contains a set of servers. These servers typically share the same characteristics. Those characteristics can be their location in the chassis, or an attribute such as server type, amount of memory, local storage, type of CPU, or local drive configuration. You can manually assign a server to a server pool, or use server pool policies and server pool policy qualifications to automate the assignment.

If your system implements multitenancy through organizations, you can designate one or more server pools to be used by a specific organization. For example, a pool that includes all servers with two CPUs could be assigned to the Marketing organization, while all servers with 64 GB memory could be assigned to the Finance organization.

A server pool can include servers from any chassis in the system. A given server can belong to multiple server pools.

Creating a Server Pool

Procedure

	Command or Action	Purpose
Step 1	UCSC# **connect resource-mgr**	Enters resource manager mode.
Step 2	UCSC(resource-mgr)# **scope org** *org-name*	Enters organization mode for the specified organization. To enter the root organization mode, type / as the *org-name*.
Step 3	UCSC(resource-mgr) /org # **create server-pool** *server-pool-name*	Creates a server pool with the specified name, and enters organization server pool mode.
Step 4	UCSC(resource-mgr) /org/server-pool # **create server** *chassis-num/slot-num*	Creates a server for the server pool. **Note** A server pool can contain more than one server. To create multiple servers for the pool, you must enter multiple create server commands from organization server pool mode.
Step 5	UCSC(resource-mgr) /org/server-pool # **commit-buffer**	Commits the transaction to the system configuration.

The following example shows how to create the server pool named ServPool2, create two servers for the server pool, and commit the transaction:

```
UCSC# connect resource-mgr
UCSC(resource-mgr)# scope org /
UCSC(resource-mgr) /org # create server-pool ServPool2
UCSC(resource-mgr) /org/server-pool* # create server 1/1
UCSC(resource-mgr) /org/server-pool* # create server 1/4
UCSC(resource-mgr) /org/server-pool* # commit-buffer
UCSC(resource-mgr) /org/server-pool #
```

Deleting a Server Pool

Procedure

	Command or Action	Purpose
Step 1	UCSC# **connect resource-mgr**	Enters resource manager mode.
Step 2	UCSC(resource-mgr)# **scope org** *org-name*	Enters organization mode for the specified organization. To enter the root organization mode, type / as the *org-name*.
Step 3	UCSC(resource-mgr) /org # **delete server-pool** *server-pool-name*	Deletes the specified server pool.

	Command or Action	Purpose
Step 4	UCSC(resource-mgr) /org # **commit-buffer**	Commits the transaction to the system configuration.

The following example shows how to delete the server pool named ServPool2 and commit the transaction:

```
UCSC# connect resource-mgr
UCSC(resource-mgr)# scope org /
UCSC(resource-mgr) /org # delete server-pool ServPool2
UCSC(resource-mgr) /org* # commit-buffer
UCSC(resource-mgr) /org #
```

Configuring IP Pools

IP Pools

IP pools are a collection of IP addresses. You can use IP pools in Cisco UCS Central in one of the following ways:

- For external management of Cisco UCS Manager servers.
- For iSCSI boot initiators.
- For both external management and iSCSI boot initiators in Cisco UCS Manager.

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

A fault is raised if the same IP address is assigned to two different Cisco UCS domains. If you want to use the same IP addresses, you can use the **scope** property to specify whether the IP addresses in the block are public or private:

- **public—**The IP addresses in the block can be assigned to one and only one registered Cisco UCS domain.
- **private—** The IP addresses in the block can be assigned to multiple Cisco UCS domains.

Cisco UCS Central creates public IP pools by default.

Global IP pools should be used for similar geographic locations. If the IP addressing schemes are different, the same IP pool can not be used for those sites.

Cisco UCS Central supports creating and deleting IPv4 and IPv6 blocks under IP pools. However, iSCSI boot initiators support only IPv4.

Creating an IP Pool

Procedure

	Command or Action	Purpose
Step 1	UCSC# **connect policy-mgr**	Enters policy manager mode.
Step 2	UCSC(policy-mgr) # **scope org** *org-name*	Enters organization mode for the specified organization. To enter the root organization mode, type / as the *org-name*.
Step 3	UCSC(policy-mgr) /org # **create ip-pool** *pool-name*	Creates an IP pool with the specified name, and enters organization IP pool mode.
Step 4	UCSC(policy-mgr) /org/ip-pool # **set descr** *description*	(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 5	UCSC(policy-mgr) /org/ip-pool # **create block** *first-ip-addr last-ip-addr gateway-ip-addr subnet-mask*	Creates a block (range) of IP addresses, and enters organization IP pool block mode. You must specify the first and last IP addresses in the address range, the gateway IP address, and subnet mask. **Note** An IP pool can contain more than one IP block. To create multiple blocks, enter multiple **create block** commands from organization IP pool mode.
Step 6	UCSC(policy-mgr) /org/ip-pool/block # **set primdns** *ip-address* **secdns** *ip-address*	Specifies the primary DNS and secondary DNS IP addresses.
Step 7	UCSC(policy-mgr) /org/ip-pool/block #**set scope** {**private** | **public**}	Specifies whether the IP addresses is private or public.
Step 8	UCSC(policy-mgr) /org/ip-pool/block # **commit-buffer**	Commits the transaction to the system configuration. **Note** If you plan to create another pool, wait at least 5 seconds.

The following example shows how to create an IP pool named GPool1, provide a description for the pool, specify a block of IP addresses and a primary and secondary IP address to be used for the pool, set the pool to private, and commit the transaction:

```
UCSC# connect policy-mgr
UCSC(policy-mgr)# scope org /
UCSC(policy-mgr) /org # create ip-pool GPool1
UCSC(policy-mgr) /org/ip-pool* # set descr "This is IP pool GPool1"
UCSC(policy-mgr) /org/ip-pool* # create block 192.168.100.1 192.168.100.200 192.168.100.10
```

```
 255.255.255.0
UCSC(policy-mgr) /org/ip-pool/block* # set primdns 192.168.100.1 secdns 192.168.100.20
UCSC(policy-mgr) /org/ip-pool/block* # set scope private
UCSC(policy-mgr) /org/ip-pool/block* # commit-buffer
UCSC(policy-mgr) /org/ip-pool/block #
```

What to Do Next

Include the IP pool in a service profile and/or template.

Creating an IP Pool with IPv6 Blocks

Procedure

	Command or Action	Purpose
Step 1	UCSC# **connect policy-mgr**	Enters policy manager mode.
Step 2	UCSC(policy-mgr) /org # **scope org** *org-name*	Enters organization mode for the specified organization. To enter the root organization mode, type/ as the *org-name*
Step 3	UCSC(policy-mgr) /org # **create ip-pool** *global-ip-pool*	Creates a global IP pool with the specified name, and enters the global IP pool mode.
Step 4	UCSC(policy-mgr) /org/ip-pool # **set descr** *description*	(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 5	UCSC(policy-mgr) /org/ip-pool # **create ipv6-block** *first-ip-addr last-ip-addr default-gateway ip address prefix*	Creates a block (range) of IP addresses, and enters organization IP pool block mode. You must specify the first and last IP addresses in the address range, the default gateway IP address, and the prefix. **Note** To create multiple blocks, enter multiple **create ipv6-block** commands.
Step 6	UCSC(policy-mgr) /org/ip-pool/ipv6-block # **set primdns** *ip-address* **secdns** *ip-address*	Specifies the primary DNS and secondary DNS IP addresses.
Step 7	UCSC(policy-mgr) /org/ip-pool/ipv6-block #**set qualifier** *word*	Sets the IPv6 block to an existing ID range qualifier name.
Step 8	UCSC(policy-mgr) /org/ip-pool/ipv6-block # **commit-buffer**	Commits the transaction to the system configuration. **Note** If you plan to create another pool, wait at least 5 seconds.

The following example shows how to create an IP pool with and IPv6 block:

```
UCSC# connect policy-mgr
UCSC(policy-mgr)# scope org org-name
UCSC(policy-mgr) /org # create ip-pool global-ip-pool
UCSC(policy-mgr) /org/ip-pool* # set descr "This is global-ip-pool gpool1"
UCSC(policy-mgr) /org/ip-pool* # create ipv6-block 2001:db8:111::a1 2001:db8:111::af
2001:db8:111::1 64
UCSC(policy-mgr) /org/ip-pool/ipv6-block* #  set primdns 2001:db8:111::FF secdns
2001:db8:111::FE
UCSC(policy-mgr) /org/ip-pool/ipv6-block* # set qualifier Q1
UCSC(policy-mgr) /org/ip-pool/ipv6-block* # commit-buffer
UCSC(policy-mgr) /org/ip-pool/ipv6-block #
```

Deleting an IP Pool

If you delete a pool, Cisco UCS Central does not reallocate any addresses from that pool that have been assigned to vNICs or vHBAs in Cisco UCS Manager. 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	UCSC# **connect policy-mgr**	Enters policy manager mode.
Step 2	UCSC(policy-mgr) # **scope org** *org-name*	Enters organization mode for the specified organization. To enter the root organization mode, type / as the *org-name*.
Step 3	UCSC(policy-mgr) /org # **delete ip-pool** *pool-name*	Deletes the specified IP pool.
Step 4	UCSC(policy-mgr) /org # **commit-buffer**	Commits the transaction to the system configuration. **Note** If you plan to delete another pool, wait at least 5 seconds.

The following example shows how to delete the IP pool named GPool1 and commit the transaction:

```
UCSC# connect policy-mgr
UCSC(policy-mgr)# scope org /
UCSC(policy-mgr) /org # delete ip-pool GPool1
UCSC(policy-mgr) /org* # commit-buffer
UCSC(policy-mgr) /org #
```

Configuring IQN Pools

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 pools created in Cisco UCS Central can be shared between Cisco UCS domains.

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	UCSC# **connect policy-mgr**	Enters policy manager mode.
Step 2	UCSC(policy-mgr) # **scope org** *org-name*	Enters organization mode for the specified organization. To enter the root organization mode, type / as the *org-name*.
Step 3	UCSC(policy-mgr) /org # **create iqn-pool** *pool-name*	Creates an IQN pool with the specified 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 has been saved.
Step 4	UCSC(policy-mgr) /org/iqn-pool # **set iqn-prefix** *prefix*	Specifies the prefix for the IQN block members. Unless limited by the adapter card, the prefix can contain up to 150 characters.
Step 5	UCSC(policy-mgr) /org/iqn-pool # **set descr** *description*	(Optional) Provides a description for the IQN pool. Enter up to 256 characters. You can use any characters or spaces except ` (accent mark), \ (backslash), ^ (carat), " (double quote), = (equal sign), > (greater than), < (less than), and ' (single quote). **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.

	Command or Action	Purpose
Step 6	UCSC(policy-mgr) /org/iqn-pool # **create block** *suffix from to*	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 *prefix*:*suffix*:*number*. 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	UCSC(policy-mgr) /org/iqn-pool/block # **commit-buffer**	Commits the transaction to the system configuration. **Note** If you plan to create another pool, wait at least 5 seconds.

The following example shows how to create an IQN pool named GPool1, provide a description for the pool, specify a prefix and a block of suffixes to be used for the pool, and commit the transaction:

```
UCSC# connect policy-mgr
UCSC(policy-mgr)# scope org /
UCSC(policy-mgr) /org # create iqn-pool GPool1
UCSC(policy-mgr) /org/iqn-pool* # set iqn-prefix iqn.alpha.com
UCSC(policy-mgr) /org/iqn-pool* # set descr "This is IQN pool GPool1"
UCSC(policy-mgr) /org/iqn-pool* # create block beta 3 5
UCSC(policy-mgr) /org/iqn-pool/block* # commit-buffer
UCSC(policy-mgr) /org/iqn-pool/block #
```

What to Do Next

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

Deleting an IQN Pool

If you delete a pool, Cisco UCS Central does not reallocate any addresses from that pool that have been assigned to vNICs or vHBAs in Cisco UCS Manager. 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	UCSC# **connect policy-mgr**	Enters policy manager mode.

	Command or Action	Purpose
Step 2	UCSC(policy-mgr) # **scope org** *org-name*	Enters organization mode for the specified organization. To enter the root organization mode, type / as the *org-name*.
Step 3	UCSC(policy-mgr) /org # **delete iqn-pool** *pool-name*	Deletes the specified IQN pool.
Step 4	UCSC(policy-mgr) /org # **commit-buffer**	Commits the transaction to the system configuration. **Note** If you plan to delete another pool, wait at least 5 seconds.

The following example shows how to delete the IQN pool named GPool1 and commit the transaction:

```
UCSC# connect policy-mgr
UCSC(policy-mgr)# scope org /
UCSC(policy-mgr) /org # delete iqn-pool GPool1
UCSC(policy-mgr) /org* # commit-buffer
UCSC(policy-mgr) /org #
```

Configuring UUID Suffix Pools

UUID Suffix Pools

A UUID suffix pool is a collection of SMBIOS UUIDs that are available to be assigned to servers. The first number of digits that constitute the prefix of the UUID are fixed. The remaining digits, the UUID suffix, are variable. A UUID suffix pool ensures that these variable values are unique for each server associated with a service profile which uses that particular pool to avoid conflicts.

If you use UUID suffix pools in service profiles, you do not have to manually configure the UUID of the server associated with the service profile. Assigning global UUID suffix pools from Cisco UCS Central to service profiles in Cisco UCS Central or Cisco UCS Manager allows them to be shared across Cisco UCS domains.

Creating a UUID Suffix Pool

Procedure

	Command or Action	Purpose
Step 1	UCSC# **connect policy-mgr**	Enters policy manager mode.
Step 2	UCSC(policy-mgr) # **scope org** *org-name*	Enters organization mode for the specified organization. To enter the root organization mode, type / as the *org-name*.

	Command or Action	Purpose
Step 3	UCSC(policy-mgr) /org # **create uuid-suffix-pool** *pool-name*	Creates a UUID suffix pool with the specified name, and enters organization UUID suffix pool mode.
Step 4	UCSC(policy-mgr) /org/uuid-suffix-pool # **set descr** *description*	(Optional) Provides a description for the UUID suffix 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 5	UCSC(policy-mgr) /org/uuid-suffix-pool # **create block** *first-uuid last-uuid*	Creates a block (range) of UUID suffixes, and enters organization UUID suffix pool block mode. You must specify the first and last UUID suffixes in the block using the form *nnnn-nnnnnnnnnnnn*, with the UUID suffixes separated by a space. **Note** A UUID suffix pool can contain more than one UUID suffix block. To create multiple UUID suffix blocks, you must enter multiple **create block** commands from organization UUID suffix pool mode.
Step 6	UCSC(policy-mgr) /org/uuid-suffix-pool/block # **commit-buffer**	Commits the transaction to the system configuration. **Note** If you plan to create another pool, wait at least 5 seconds.

The following example shows how to create a UUID suffix pool named GPool1, provide a description for the pool, specify a block of UUID suffixes to be used for the pool, and commit the transaction:

```
UCSC# connect policy-mgr
UCSC(policy-mgr)# scope org /
UCSC(policy-mgr) /org # create uuid-suffix-pool GPool1
UCSC(policy-mgr) /org/uuid-suffix-pool* # set descr "This is UUID suffix pool GPool1"
UCSC(policy-mgr) /org/uuid-suffix-pool* # create block 1000-000000000001 1000-000000000010
UCSC(policy-mgr) /org/uuid-suffix-pool/block* # commit-buffer
UCSC(policy-mgr) /org/uuid-suffix-pool/block #
```

What to Do Next

Include the UUID suffix pool in a service profile and/or template.

Deleting a UUID Suffix Pool

If you delete a pool, Cisco UCS Central does not reallocate any addresses from that pool that have been assigned to vNICs or vHBAs in Cisco UCS Manager. 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	UCSC# **connect policy-mgr**	Enters policy manager mode.
Step 2	UCSC(policy-mgr) # **scope org** *org-name*	Enters organization mode for the specified organization. To enter the root organization mode, type / as the *org-name*.
Step 3	UCSC(policy-mgr) /org # **delete uuid-suffix-pool** *pool-name*	Deletes the specified UUID suffix pool.
Step 4	UCSC(policy-mgr) /org # **commit-buffer**	Commits the transaction to the system configuration. **Note** If you plan to delete another pool, wait at least 5 seconds.

The following example shows how to delete the UUID suffix pool named GPool1 and commit the transaction:

```
UCSC# connect policy-mgr
UCSC(policy-mgr)# scope org /
UCSC(policy-mgr) /org # delete uuid-suffix-pool GPool1
UCSC(policy-mgr) /org* # commit-buffer
UCSC(policy-mgr) /org #
```

Configuring Server-Related Policies

Ethernet and Fibre Channel Adapter Policies

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

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

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

- Max LUNs Per Target—SANsurfer has a maximum of 256 LUNs and does not display more than that number. Cisco UCS Central supports a higher maximum number of LUNs.
- Link Down Timeout—In SANsurfer, you configure the timeout threshold for link down in seconds. In Cisco UCS Central, you configure this value in milliseconds. Therefore, a value of 5500 ms in Cisco UCS Central displays as 5s in SANsurfer.
- Max Data Field Size—SANsurfer has allowed values of 512, 1024, and 2048. Cisco UCS Central allows you to set values of any size. Therefore, a value of 900 in Cisco UCS Central displays as 512 in SANsurfer.

Operating System Specific Adapter Policies

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

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

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

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

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

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

Accelerated Receive Flow Steering

Accelerated Receive Flow Steering (ARFS) is hardware-assisted receive flow steering that can increase CPU data cache hit rate by steering kernel level processing of packets to the CPU where the application thread consuming the packet is running. Using ARFS can improve CPU efficiency and reduce traffic latency.

ARFS is disabled by default and can be enabled through Cisco UCS Manager. To configure ARFS, do the following:

1. Create an adapter policy with ARFS enabled.
2. Associate the adapter policy with a service profile.
3. Enable ARFS on a host.

Configuring an Ethernet Adapter Policy

Procedure

	Command or Action	Purpose
Step 1	UCSC# **connect policy-mgr**	Enters policy manager mode.
Step 2	UCSC(policy-mgr)# **scope org** *org-name*	Enters organization mode for the specified organization. To enter the root organization mode, type / as the *org-name* .
Step 3	UCSC(policy-mgr) /org # **create eth-policy** *policy-name*	Creates the specified Ethernet adapter policy and enters organization Ethernet policy mode.
Step 4	UCSC(policy-mgr) /org/eth-policy # **set comp-queue count** *count*	(Optional) Configures the Ethernet completion queue.
Step 5	UCSC(policy-mgr) /org/eth-policy # **set descr** *description*	(Optional) Provides a description for the policy. **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 6	UCSC(policy-mgr) /org/eth-policy # **set failover timeout** *timeout-sec*	(Optional) Configures the Ethernet failover.
Step 7	UCSC(policy-mgr) /org/eth-policy # **set interrupt** {**coalescing-time** *sec* | **coalescing-type** {**idle** | **min**} | **count** *count* | **mode** {**intx** | **msi** | **msi-x**}}	(Optional) Configures the Ethernet interrupt.
Step 8	UCSC(policy-mgr) /org/eth-policy # **set offload** {**large-receive** | **tcp-rx-checksum** | **tcp-segment** | **tcp-tx-checksum**} {**disabled** | **enabled**}	(Optional) Configures the Ethernet offload.
Step 9	UCSC(policy-mgr) /org/eth-policy # **set recv-queue** {**count** *count* | **ring-size** *size-num*}	(Optional) Configures the Ethernet receive queue.
Step 10	UCSC(policy-mgr) /org/eth-policy # **set rss receivesidescaling** {**disabled** | **enabled**}	(Optional) Configures the RSS.
Step 11	UCSC(policy-mgr) /org/eth-policy # **set trans-queue** {**count** *count* | **ring-size** *size-num*}	(Optional) Configures the Ethernet transmit queue.
Step 12	UCSC(policy-mgr) /org/eth-policy # **commit-buffer**	Commits the transaction to the system configuration.

The following example configures an Ethernet adapter policy, and commits the transaction:

```
UCSC# connect policy-mgr
UCSC(policy-mgr)# scope org /
UCSC(policy-mgr) /org* # create eth-policy EthPolicy19
UCSC(policy-mgr) /org/eth-policy* # set comp-queue count 16
UCSC(policy-mgr) /org/eth-policy* # set descr "This is an Ethernet adapter policy example."
UCSC(policy-mgr) /org/eth-policy* # set failover timeout 300
UCSC(policy-mgr) /org/eth-policy* # set interrupt count 64
UCSC(policy-mgr) /org/eth-policy* # set offload large-receive disabled
UCSC(policy-mgr) /org/eth-policy* # set recv-queue count 32
UCSC(policy-mgr) /org/eth-policy* # set rss receivesidescaling enabled
UCSC(policy-mgr) /org/eth-policy* # set trans-queue
UCSC(policy-mgr) /org/eth-policy* # commit-buffer
UCSC(policy-mgr) /org/eth-policy #
```

Deleting an Ethernet Adapter Policy

Procedure

	Command or Action	Purpose
Step 1	UCSC# **connect policy-mgr**	Enters policy manager mode.
Step 2	UCSC(policy-mgr) # **scope org** *org-name*	Enters organization mode for the specified organization. To enter the root organization mode, type / as the *org-name*.
Step 3	UCSC(policy-mgr) /org # **delete eth-policy** *policy-name*	Deletes the specified Ethernet adapter policy.
Step 4	UCSC(policy-mgr) /org # **commit-buffer**	Commits the transaction to the system configuration.

The following example shows how to delete an Ethernet adapter policy, and commits the transaction:

```
UCSC# connect policy-mgr
UCSC(policy-mgr)# scope org /
UCSC(policy-mgr) /org # delete eth-policy EthPolicy19
UCSC(policy-mgr) /org* # commit-buffer
UCSC(policy-mgr) /org #
```

Server BIOS Settings

Cisco UCS provides two methods for making global modifications to the BIOS settings on servers in an Cisco UCS domain. You can create one or more BIOS policies that include a specific grouping of BIOS settings that match the needs of a server or set of servers, or you can use the default BIOS settings for a specific server platform.

Both the BIOS policy and the default BIOS settings for a server platform enable you to fine tune the BIOS settings for a server managed by Cisco UCS Central.

Depending upon the needs of the data center, you can configure BIOS policies for some service profiles and use the BIOS defaults in other service profiles in the same Cisco UCS domain, or you can use only one of them. You can also use Cisco UCS Central to view the actual BIOS settings on a server and determine whether they are meeting current needs.

Note Cisco UCS Central pushes BIOS configuration changes through a BIOS policy or default BIOS settings to the Cisco Integrated Management Controller (CIMC) buffer. These changes remain in the buffer and do not take effect until the server is rebooted.

We recommend that you verify the support for BIOS settings in the server that you want to configure. Some settings, such as Mirroring Mode for RAS Memory, are not supported by all Cisco UCS servers.

BIOS Policy

The BIOS policy is a policy that automates the configuration of BIOS settings for a server or group of servers. You can create global BIOS policies available to all servers in the root organization, or you can create BIOS policies in sub-organizations that are only available to that hierarchy.

To use a BIOS policy, do the following:

1. Create the BIOS policy in Cisco UCS Central.
2. Assign the BIOS policy to one or more service profiles.
3. Associate the service profile with a server.

During service profile association, Cisco UCS Central modifies the BIOS settings on the server to match the configuration in the BIOS policy. If you do not create and assign a BIOS policy to a service profile, the server uses the default BIOS settings for that server platform.

Default BIOS Settings

Cisco UCS Central includes a set of default BIOS settings for each type of server supported by Cisco UCS. The default BIOS settings are available only in the root organization and are global. Only one set of default BIOS settings can exist for each server platform supported by Cisco UCS. You can modify the default BIOS settings, but you cannot create an additional set of default BIOS settings.

Each set of default BIOS settings are designed for a particular type of supported server and are applied to all servers of that specific type which do not have a BIOS policy included in their service profiles.

Unless a Cisco UCS implementation has specific needs that are not met by the server-specific settings, we recommend that you use the default BIOS settings that are designed for each type of server in the Cisco UCS domain.

Cisco UCS Central applies these server platform-specific BIOS settings as follows:

- The service profile associated with a server does not include a BIOS policy.
- The BIOS policy is configured with the platform-default option for a specific setting.

You can modify the default BIOS settings provided by Cisco UCS Central. However, any changes to the default BIOS settings apply to all servers of that particular type or platform. If you want to modify the BIOS settings for only certain servers, we recommend that you use a BIOS policy.

Creating a BIOS Policy

Procedure

	Command or Action	Purpose
Step 1	UCSC# **connect policy-mgr**	Enters policy manager mode.
Step 2	UCSC(policy-mgr)# **scope org** *org-name*	Enters organization mode for the specified organization. To enter the root organization mode, type / as the *org-name* .
Step 3	Configure the BIOS settings	For the CLI commands, descriptions and information about the options for each BIOS setting, see the following topics: • Main BIOS Settings, on page 19 • Processor BIOS Settings, on page 21 • Intel Directed I/O BIOS Settings, on page 32 • RAS Memory BIOS Settings, on page 34 • Serial Port BIOS Settings, on page 36 • USB BIOS Settings, on page 37 • PCI Configuration BIOS Settings, on page 38 • Boot Options BIOS Settings, on page 40 • Server Management BIOS Settings, on page 41
Step 4	UCSC(policy-mgr) /org/bios-policy # **commit-buffer**	Commits the transaction to the system configuration.

The following example creates a BIOS policy under the root organization and commits the transaction:

```
UCSC# connect policy-mgr
UCSC(policy-mgr)# scope org /
UCSC(policy-mgr)/ org #create bios-policy biosPolicy3
UCSC(policy-mgr) /org/bios-policy* # set numa-config numa-optimization enabled
UCSC(policy-mgr) /org/bios-policy* # commit-buffer
UCSC(policy-mgr) /org/bios-policy #
```

Viewing the Actual BIOS Settings for a Server

Procedure

	Command or Action	Purpose
Step 1	UCSC# **connect policy-mgr**	Enters policy manager mode.

	Command or Action	Purpose
Step 2	UCSC(policy-mgr) # **scope org** *org-name*	Enters organization mode for the specified organization. To enter the root organization mode, type / as the *org-name*.
Step 3	UCSC(policy-mgr) /org # **scope server** *chassis-id* / *server-id*	Enters chassis server mode for the specified server.
Step 4	UCSC(policy-mgr) /org /chassis/server # **scope bios**	Enters BIOS mode for the specified server.
Step 5	UCSC(policy-mgr) /org /chassis/server/bios # **scope bios-settings**	Enters BIOS settings mode for the specified server.
Step 6	UCSC(policy-mgr) /org /chassis/server/bios/bios-settings # **show** *setting*	Displays the BIOS setting. Enter **show ?** to display a list of allowed values for *setting* .

The following example displays a BIOS setting for blade 3 in chassis 1:

```
UCSC# connect policy-mgr
UCSC(policy-mgr)# scope org /
UCSC(policy-mgr)/org # scope server 1/3
UCSC(policy-mgr)/org /chassis/server # scope bios
UCSC(policy-mgr)/org /chassis/server/bios # scope  bios-settings
UCSC(policy-mgr)/org /chassis/server/bios/bios-settings # show intel-vt-config

Intel Vt Config:
    Vt
    --
    Enabled

UCSC(policy-mgr)/org /chassis/server/bios/bios-settings #
```

Modifying BIOS Defaults

Procedure

	Command or Action	Purpose
Step 1	UCSC# **connect policy-mgr**	Enters policy manager mode.
Step 2	UCSC(policy-mgr)# **scope org** *org-name*	Enters organization mode for the specified organization. To enter the root organization mode, type / as the *org-name*.
Step 3	UCSC(policy-mgr) /org # **scope system**	Enters system mode.
Step 4	UCSC(policy-mgr) /org /system # **scope server-defaults**	Enters server defaults mode.
Step 5	UCSC(policy-mgr) /org /system/server-defaults # **show platform**	(Optional) Displays platform descriptions for all servers.

	Command or Action	Purpose
Step 6	UCSC(policy-mgr) /org /system/server-defaults # **scope platform** *platform-description*	Enters server defaults mode for the server specified. For the *platform-description* argument, enter the server description displayed by the **show platform** command using the following format: "*vendor*" *model revision*. **Tip** You must enter the vendor exactly as shown in the **show platform** command, including all punctuation marks.
Step 7	UCSC(policy-mgr) /org /system/server-defaults/platform # **scope bios-settings**	Enters server defaults BIOS settings mode for the server.
Step 8	Reconfigure the BIOS settings.	For the CLI commands, descriptions and information about the options for each BIOS setting, see the following topics: • Main BIOS Settings, on page 19 • Processor BIOS Settings, on page 21 • Intel Directed I/O BIOS Settings, on page 32 • RAS Memory BIOS Settings, on page 34 • Serial Port BIOS Settings, on page 36 • USB BIOS Settings, on page 37 • PCI Configuration BIOS Settings, on page 38 • Boot Options BIOS Settings, on page 40 • Server Management BIOS Settings, on page 41
Step 9	UCSC(policy-mgr) /org /system/server-defaults/platform/bios-settings # **commit-buffer**	Commits the transaction to the system configuration.

The following example shows how to change the NUMA default BIOS setting for a platform and commit the transaction:

```
UCSC# connect policy-mgr
UCSC(policy-mgr)# scope org
UCSC(policy-mgr)/org# scope system
UCSC(policy-mgr)/org /system # scope server-defaults
UCSC(policy-mgr)/org /system/server-defaults # show platform

Platform:
    Product Name Vendor     Model      Revision
    ------------ ---------- ---------- --------
    Cisco B200-M1
                 Cisco Systems, Inc.
                            N20-B6620-1
```

```
UCSC(policy-mgr)/org /system/server-defaults # scope platform "Cisco Systems, Inc."
N20-B6620-1 0
UCSC(policy-mgr)/org /system/server-defaults/platform # scope bios-settings
UCSC(policy-mgr)/org /system/server-defaults/platform/bios-settings # set numa-config
numa-optimization disabled
UCSC(policy-mgr)/org /system/server-defaults/platform/bios-settings* # commit-buffer
UCSC(policy-mgr)/org /system/server-defaults/platform/bios-settings #
```

Deleting a BIOS Policy

Procedure

	Command or Action	Purpose
Step 1	UCSC# **connect policy-mgr**	Enters policy manager mode.
Step 2	UCSC(policy-mgr) # **scope org** *org-name*	Enters organization mode for the specified organization. To enter the root organization mode, type / as the *org-name*.
Step 3	UCSC(policy-mgr) /org # **delete bios-policy** *policy-name*	Deletes the specified BIOS policy.
Step 4	UCSC(policy-mgr) /org # **commit-buffer**	Commits the transaction to the system configuration.

The following example shows how to delete a BIOS policy under the root organization and commits the transaction:

```
UCSC# connect policy-mgr
UCSC(policy-mgr)# scope org /
UCSC(policy-mgr)/ org #delete bios-policy biosPolicy3
UCSC(policy-mgr) /org* # commit-buffer
UCSC(policy-mgr) /org #
```

Main BIOS Settings

The following table lists the main server BIOS settings that you can configure through a BIOS policy or the default BIOS settings:

Name	Description
Reboot on BIOS Settings Change	When the server is rebooted after you change one or more BIOS settings. If you enable this setting, the server is rebooted according to the maintenance policy in the server's service profile. For example, if the maintenance policy requires user acknowledgment, the server is not rebooted and the BIOS changes are not applied until a user acknowledges the pending activity. If you do not enable this setting, the BIOS changes are not applied until the next time the server is rebooted, whether as a result of another server configuration change or a manual reboot.
Quiet Boot	What the BIOS displays during Power On Self-Test (POST). This can be one of the following: • **disabled—**The BIOS displays all messages and Option ROM information during boot. • **enabled—**The BIOS displays the logo screen, but does not display any messages or Option ROM information during boot. • **—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor.
Post Error Pause	What happens when the server encounters a critical error during POST. This can be one of the following: • **disabled—**The BIOS continues to attempt to boot the server. • **enabled—**The BIOS pauses the attempt to boot the server and opens the Error Manager when a critical error occurs during POST. • **—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor.
Resume Ac On Power Loss	How the server behaves when power is restored after an unexpected power loss. This can be one of the following: • **stay-off—**The server remains off until manually powered on. • **last-state—**The server is powered on and the system attempts to restore its last state. • **reset—**The server is powered on and automatically reset. • **—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor.

Name	Description
Front Panel Lockout	Whether the power and reset buttons on the front panel are ignored by the server. This can be one of the following: • **disabled**—The power and reset buttons on the front panel are active and can be used to affect the server. • **enabled**—The power and reset buttons are locked out. The server can only be reset or powered on or off from the CIMC GUI. • **—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor.
Consistent Device Naming	Consistent Device Naming allows Ethernet interfaces to be named in a consistent manner. This makes Ethernet interface names more uniform, easy to identify, and persistent when adapter or other configuration changes are made. Whether consistent device naming is enabled or not. This can be one of the following: • **disabled**—Consistent device naming is disabled for the BIOS policy. • **enabled**—Consistent device naming is enabled for the BIOS policy. This enables Ethernet interfaces to be named consistently. • **—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor.

Processor BIOS Settings

The following table lists the processor BIOS settings that you can configure through a BIOS policy or the default BIOS settings:

Name	Description
Turbo Boost	Whether the processor uses Intel Turbo Boost Technology, which allows the processor to automatically increase its frequency if it is running below power, temperature, or voltage specifications. This can be one of the following: • **disabled—**The processor does not increase its frequency automatically. • **enabled—**The processor uses Turbo Boost Technology if required. • **—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor.
Enhanced Intel Speedstep	Whether the processor uses Enhanced Intel SpeedStep Technology, which allows the system to dynamically adjust processor voltage and core frequency. This technology can result in decreased average power consumption and decreased average heat production. This can be one of the following: • **disabled—**The processor never dynamically adjusts its voltage or frequency. • **enabled—**The processor utilizes Enhanced Intel SpeedStep Technology and enables all supported processor sleep states to further conserve power. • **—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor. We recommend that you contact your operating system vendor to make sure your operating system supports this feature.
Hyper Threading	Whether the processor uses Intel Hyper-Threading Technology, which allows multithreaded software applications to execute threads in parallel within each processor. This can be one of the following: • **disabled—**The processor does not permit hyperthreading. • **enabled—**The processor allows for the parallel execution of multiple threads. • **—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor. We recommend that you contact your operating system vendor to make sure the operating system supports this feature.

Name	Description
Core Multi Processing	Sets the state of logical processor cores per CPU in a package. If you disable this setting, Intel Hyper Threading technology is also disabled. This can be one of the following: • **all—**Enables multiprocessing on all logical processor cores. • **1** through ***n*—**Specifies the number of logical processor cores per CPU that can run on the server. To disable multiprocessing and have only one logical processor core per CPU running on the server, choose 1. • **—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor. We recommend that you contact your operating system vendor to make sure your operating system supports this feature.
Execute Disabled Bit	Classifies memory areas on the server to specify where the application code can execute. As a result of this classification, the processor disables code execution if a malicious worm attempts to insert code in the buffer. This setting helps to prevent damage, worm propagation, and certain classes of malicious buffer overflow attacks. This can be one of the following: • **disabled—**The processor does not classify memory areas. • **enabled—**The processor classifies memory areas. • **—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor. We recommend that you contact your operating system vendor to make sure your operating system supports this feature.
Virtualization Technology (VT)	Whether the processor uses Intel Virtualization Technology, which allows a platform to run multiple operating systems and applications in independent partitions. This can be one of the following: • **disabled—**The processor does not permit virtualization. • **enabled—**The processor allows multiple operating systems in independent partitions. • **—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor. **Note** If you change this option, you must power cycle the server before the setting takes effect.

Name	Description
Hardware Pre-fetcher	Whether the processor allows the Intel hardware prefetcher to fetch streams of data and instruction from memory into the unified second-level cache when necessary. This can be one of the following: • **Disabled—**The hardware prefetcher is not used. • **Enabled—**The processor uses the hardware prefetcher when cache issues are detected. • **—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor. **Note** must be set to **Custom** in order to specify this value. For any value other than **Custom**, this option is overridden by the setting in the selected CPU performance profile.
Adjacent Cache Line Pre-fetcher	Whether the processor fetches cache lines in even/odd pairs instead of fetching just the required line. This can be one of the following: • **Disabled—**The processor only fetches the required line. • **Enabled—**The processor fetches both the required line and its paired line. • **—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor. **Note** must be set to **Custom** in order to specify this value. For any value other than **Custom**, this option is overridden by the setting in the selected CPU performance profile.
DCU Streamer Pre-fetch	Whether the processor uses the DCU IP Prefetch mechanism to analyze historical cache access patterns and preload the most relevant lines in the L1 cache. This can be one of the following: • **Disabled—**The processor does not try to anticipate cache read requirements and only fetches explicitly requested lines. • **Enabled—**The DCU prefetcher analyzes the cache read pattern and prefetches the next line in the cache if it determines that it may be needed. • **—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor.

Name	Description
DCU IP Pre-fetcher	Whether the processor uses the DCU IP Prefetch mechanism to analyze historical cache access patterns and preload the most relevant lines in the L1 cache. This can be one of the following: • **Disabled—**The processor does not preload any cache data. • **Enabled—**The DCU IP prefetcher preloads the L1 cache with the data it determines to be the most relevant. • **—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor.
Direct Cache Access	Allows processors to increase I/O performance by placing data from I/O devices directly into the processor cache. This setting helps to reduce cache misses. This can be one of the following: • **disabled—**Data from I/O devices is not placed directly into the processor cache. • **enabled—**Data from I/O devices is placed directly into the processor cache. • **—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor.
Processor C State	Whether the system can enter a power savings mode during idle periods. This can be one of the following: • **disabled—**The system remains in a high-performance state even when idle. • **enabled—**The system can reduce power to system components such as the DIMMs and CPUs. • **—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor. We recommend that you contact your operating system vendor to make sure your operating system supports this feature.
Processor C1E	Allows the processor to transition to its minimum frequency upon entering C1. This setting does not take effect until after you have rebooted the server. This can be one of the following: • **disabled—**The CPU continues to run at its maximum frequency in the C1 state. • **enabled—**The CPU transitions to its minimum frequency. This option saves the maximum amount of power in the C1 state. • **—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor.

Name	Description
Processor C3 Report	Whether the processor sends the C3 report to the operating system. This can be one of the following: • **disabled—**The processor does not send the C3 report. • **acpi-c2—**The processor sends the C3 report using the advanced configuration and power interface (ACPI) C2 format. • **acpi-c3—**The processor sends the C3 report using the ACPI C3 format. • **—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor. On the Cisco UCS B440 Server, the BIOS Setup menu uses enabled and disabled for these options. If you specify acpi-c2 or acpi-c2, the server sets the BIOS value for that option to enabled.
Processor C6 Report	Whether the processor sends the C6 report to the operating system. This can be one of the following: • **disabled—**The processor does not send the C6 report. • **enabled—**The processor sends the C6 report. • **—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor.
Processor C7 Report	Whether the processor sends the C7 report to the operating system. This can be one of the following: • **disabled—**The processor does not send the C7 report. • **enabled—**The processor sends the C7 report. • **—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor.

Name	Description
CPU Performance	Sets the CPU performance profile for the server. This can be one of the following: • **enterprise—**For M3 servers, all prefetchers and data reuse are enabled. For M1 and M2 servers, data reuse and the DCU IP prefetcher are enabled, and all other prefetchers are disabled. • **high-throughput—**Data reuse and the DCU IP prefetcher are enabled, and all other prefetchers are disabled. • **hpc—**All prefetchers are enabled and data reuse is disabled. This setting is also known as high-performance computing.
Max Variable MTRR Setting	Allows you to select the number of mean time to repair (MTRR) variables. This can be one of the following: • **auto-max—**BIOS uses the default value for the processor. • **8—**BIOS uses the number specified for the variable MTRR. • **—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor.
Local X2 APIC	Allows you to set the type of Application Policy Infrastructure Controller (APIC) architecture. This can be one of the following: • **xapic—**Uses the standard xAPIC architecture. • **x2apic—**Uses the enhanced x2APIC architecture to support 32 bit addressability of processors. • **auto—**Automatically uses the xAPIC architecture that is detected. • **—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor.

Name	Description
Power Technology	Enables you to configure the CPU power management settings for the following options: • Enhanced Intel Speedstep Technology • Intel Turbo Boost Technology • Processor Power State C6 Power Technology can be one of the following: • **Disabled—**The server does not perform any CPU power management and any settings for the BIOS parameters mentioned above are ignored. • **—**The server determines the best settings for the BIOS parameters mentioned above and ignores the individual settings for these parameters. • **Performance—**The server automatically optimizes the performance for the BIOS parameters mentioned above. • **Custom—**The server uses the individual settings for the BIOS parameters mentioned above. You must select this option if you want to change any of these BIOS parameters. • **—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor.
Energy Performance	Allows you to determine whether system performance or energy efficiency is more important on this server. This can be one of the following: • **performance** • **balanced-performance** • **balanced-energy** • **energy-efficient** • **—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor. **Note** must be set to **Custom** or the server ignores the setting for this parameter.

Name	Description
Frequency Floor Override	Whether the CPU is allowed to drop below the maximum non-turbo frequency when idle. This can be one of the following: • **Disabled—** The CPU can drop below the maximum non-turbo frequency when idle. This option decreases power consumption but may reduce system performance. • **Enabled—** The CPU cannot drop below the maximum non-turbo frequency when idle. This option improves system performance but may increase power consumption. • **—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor.
P-STATE Coordination	Allows you to define how BIOS communicates the P-state support model to the operating system. There are 3 models as defined by the Advanced Configuration and Power Interface (ACPI) specification. • **HW_ALL—**The processor hardware is responsible for coordinating the P-state among logical processors with dependencies (all logical processors in a package). • **SW_ALL—**The OS Power Manager (OSPM) is responsible for coordinating the P-state among logical processors with dependencies (all logical processors in a physical package), and must initiate the transition on all of the logical processors. • **SW_ANY—**The OS Power Manager (OSPM) is responsible for coordinating the P-state among logical processors with dependencies (all logical processors in a package), and may initiate the transition on any of the logical processors in the domain. • **—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor. **Note** must be set to **Custom** or the server ignores the setting for this parameter.

Name	Description
DRAM Clock Throttling	Allows you to tune the system settings between the memory bandwidth and power consumption. This can be one of the following: • **Balanced—** DRAM clock throttling is reduced, providing a balance between performance and power. • **Performance—**DRAM clock throttling is disabled, providing increased memory bandwidth at the cost of additional power. • **—**DRAM clock throttling is increased to improve energy efficiency. • **—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor.
Channel Interleaving	Whether the CPU divides memory blocks and spreads contiguous portions of data across interleaved channels to enable simultaneous read operations. This can be one of the following: • **Auto—**The CPU determines what interleaving is done. • **1-way—**Some channel interleaving is used. • **2-way** • **3-way** • **4-way—**The maximum amount of channel interleaving is used. • **—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor.
Rank Interleaving	Whether the CPU interleaves physical ranks of memory so that one rank can be accessed while another is being refreshed. This can be one of the following: • **Auto—**The CPU determines what interleaving is done. • **1-way—**Some rank interleaving is used. • **2-way** • **4-way** • **8-way—**The maximum amount of rank interleaving is used. • **—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor.

Name	Description
Demand Scrub	Whether the system corrects single bit memory errors encountered when the CPU or I/O makes a demand read. This can be one of the following: • **Disabled—** Single bit memory errors are not corrected. • **Enabled—** Single bit memory errors are corrected in memory and the corrected data is set in response to the demand read. • **—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor.
Patrol Scrub	Whether the system actively searches for, and corrects, single bit memory errors even in unused portions of the memory on the server. This can be one of the following: • **Disabled—**The system checks for memory ECC errors only when the CPU reads or writes a memory address. • **Enabled—**The system periodically reads and writes memory searching for ECC errors. If any errors are found, the system attempts to fix them. This option may correct single bit errors before they become multi-bit errors, but it may adversely affect performance when the patrol scrub is running. • **—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor.
Altitude	The approximate number of meters above sea level at which the physical server is installed. This can be one of the following: • **Auto—**The CPU determines the physical elevation. • **—**The server is approximately 300 meters above sea level. • **—**The server is approximately 900 meters above sea level. • **—**The server is approximately 1500 meters above sea level. • **—**The server is approximately 3000 meters above sea level. • **—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor.

Name	Description
Package C State Limit	The amount of power available to the server components when they are idle. This can be one of the following: • —The server may enter any available C state. • —The server provides all server components with full power at all times. This option maintains the highest level of performance and requires the greatest amount of power. • —When the CPU is idle, the system slightly reduces the power consumption. This option requires less power than C0 and allows the server to return quickly to high performance mode. • —When the CPU is idle, the system reduces the power consumption further than with the C1 option. This requires less power than C1 or C0, but it takes the server slightly longer to return to high performance mode. • —When the CPU is idle, the system reduces the power consumption further than with the C3 option. This option saves more power than C0, C1, or C3, but there may be performance issues until the server returns to full power. • —When the CPU is idle, the system reduces the power consumption further than with the C1 option. This requires less power than C1 or C0, but it takes the server slightly longer to return to high performance mode. • —When the CPU is idle, the server makes a minimal amount of power available to the components. This option saves the maximum amount of power but it also requires the longest time for the server to return to high performance mode. • —When the CPU is idle, the server makes a minimal amount of power available to the components. This option saves more power than C7, but it also requires the longest time for the server to return to high performance mode. • —The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor.

Intel Directed I/O BIOS Settings

The following table lists the Intel Directed I/O BIOS settings that you can configure through a BIOS policy or the default BIOS settings:

Name	Description
VT for Directed IO	Whether the processor uses Intel Virtualization Technology for Directed I/O (VT-d). This can be one of the following: • **disabled—**The processor does not use virtualization technology. • **enabled—**The processor uses virtualization technology. • **—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor. **Note** This option must be enabled if you want to change any of the other Intel Directed I/O BIOS settings.
Interrupt Remap	Whether the processor supports Intel VT-d Interrupt Remapping. This can be one of the following: • **disabled—**The processor does not support remapping. • **enabled—**The processor uses VT-d Interrupt Remapping as required. • **—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor.
Coherency Support	Whether the processor supports Intel VT-d Coherency. This can be one of the following: • **disabled—**The processor does not support coherency. • **enabled—**The processor uses VT-d Coherency as required. • **—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor.
ATS Support	Whether the processor supports Intel VT-d Address Translation Services (ATS). This can be one of the following: • **disabled—**The processor does not support ATS. • **enabled—**The processor uses VT-d ATS as required. • **—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor.

Name	Description
Pass Through DMA Support	Whether the processor supports Intel VT-d Pass-through DMA. This can be one of the following: • **disabled—**The processor does not support pass-through DMA. • **enabled—**The processor uses VT-d Pass-through DMA as required. • **—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor.

RAS Memory BIOS Settings

The following table lists the RAS memory BIOS settings that you can configure through a BIOS policy or the default BIOS settings:

Name	Description
Memory RAS Config	How the memory reliability, availability, and serviceability (RAS) is configured for the server. This can be one of the following: • **maximum performance—**System performance is optimized. • **mirroring—**System reliability is optimized by using half the system memory as backup. • **lockstep—**If the DIMM pairs in the server have an identical type, size, and organization and are populated across the SMI channels, you can enable lockstep mode to minimize memory access latency and provide better performance. Lockstep is enabled by default for B440 servers. • **—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor.

Name	Description
NUMA	Whether the BIOS supports NUMA. This can be one of the following: • **disabled—**The BIOS does not support NUMA. • **enabled—**The BIOS includes the ACPI tables that are required for NUMA-aware operating systems. If you enable this option, the system must disable Inter-Socket Memory interleaving on some platforms. • **—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor.
Mirroring Mode	Memory mirroring enhances system reliability by keeping two identical data images in memory. This option is only available if you choose the **mirroring** option for **Memory RAS Config**. It can be one of the following: • **inter-socket—**Memory is mirrored between two Integrated Memory Controllers (IMCs) across CPU sockets. • **intra-socket—**One IMC is mirrored with another IMC in the same socket. • **—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor.
Sparing Mode	Sparing optimizes reliability by holding memory in reserve so that it can be used in case other DIMMs fail. This option provides some memory redundancy, but does not provide as much redundancy as mirroring. The available sparing modes depend on the current memory population. This option is only available if you choose **sparing** option for **Memory RAS Config**. It can be one of the following: • **dimm-sparing—**One DIMM is held in reserve. If a DIMM fails, the contents of a failing DIMM are transferred to the spare DIMM. • **rank-sparing—**A spare rank of DIMMs is held in reserve. If a rank of DIMMs fails, the contents of the failing rank are transferred to the spare rank. • **—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor.

Name	Description
LV DDR Mode	Whether the system prioritizes low voltage or high frequency memory operations. This can be one of the following: • **power-saving-mode—**The system prioritizes low voltage memory operations over high frequency memory operations. This mode may lower memory frequency in order to keep the voltage low. • **performance-mode—**The system prioritizes high frequency operations over low voltage operations. • **—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor.
DRAM Refresh Rate	The refresh interval rate for internal memory. This can be one of the following: • **1x** • **2x** • **3x** • **4x** • **auto** • **—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor.
DDR3 Voltage Selection	The voltage to be used by the dual-voltage RAM. This can be one of the following: • **DDR3-1500mv** • **DDR3-1350mv** • **—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor.

Serial Port BIOS Settings

The following table lists the serial port BIOS settings that you can configure through a BIOS policy or the default BIOS settings:

Name	Description
Serial Port A	Whether serial port A is enabled or disabled. This can be one of the following: • **disabled—**The serial port is disabled. • **enabled—**The serial port is enabled. • **—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor.

USB BIOS Settings

The following table lists the USB BIOS settings that you can configure through a BIOS policy or the default BIOS settings:

Name	Description
Make Device Non Bootable	Whether the server can boot from a USB device. This can be one of the following: • **disabled—**The server can boot from a USB device. • **enabled—**The server cannot boot from a USB device. • **—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor.
Legacy USB Support	Whether the system supports legacy USB devices. This can be one of the following: • **disabled—**USB devices are only available to EFI applications. • **enabled—**Legacy USB support is always available. • **auto—**Disables legacy USB support if no USB devices are connected. • **—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor.

Name	Description
USB System Idle Power Optimizing Setting	Whether the USB System Idle Power Optimizing setting is used to reduce USB EHCI idle power consumption. Depending upon the value you choose, this setting can have an impact on performance. This can be one of the following: • **high-performance—**The USB System Idle Power Optimizing setting is disabled, because optimal performance is preferred over power savings. Selecting this option can significantly improve performance. We recommend you select this option unless your site has server power restrictions. • **lower-idle-power—**The USB System Idle Power Optimizing setting is enabled, because power savings are preferred over optimal performance. • **—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor.
USB Front Panel Access Lock	USB front panel lock is configured to enable or disable the front panel access to USB ports. This can be one of the following: • **disabled** • **enabled** • **—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor.

PCI Configuration BIOS Settings

The following table lists the PCI configuration BIOS settings that you can configure through a BIOS policy or the default BIOS settings:

Name	Description
Max Memory Below 4G	Whether the BIOS maximizes memory usage below 4GB for an operating system without PAE support, depending on the system configuration. This can be one of the following: • **disabled—**Does not maximize memory usage. Choose this option for all operating systems with PAE support. • **enabled—**Maximizes memory usage below 4GB for an operating system without PAE support. • **—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor.

Name	Description
Memory Mapped IO Above 4Gb Config	Whether to enable or disable memory mapped I/O of 64-bit PCI devices to 4GB or greater address space. Legacy option ROMs are not able to access addresses above 4GB. PCI devices that are 64-bit compliant but use a legacy option ROM may not function correctly with this setting enabled. This can be one of the following: • **disabled—**Does not map I/O of 64-bit PCI devices to 4GB or greater address space. • **enabled—**Maps I/O of 64-bit PCI devices to 4GB or greater address space. • **—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor.
VGA Priority	Allows you to set the priority for VGA graphics devices if multiple VGA devices are found in the system. This can be one of the following: • **onboard—**Priority is given to the onboard VGA device. BIOS post screen and OS boot are driven through the onboard VGA port. • **offboard—**Priority is given to the PCIE Graphics adapter. BIOS post screen and OS boot are driven through the external graphics adapter port. • **onboard-vga-disabled—**Priority is given to the PCIE Graphics adapter, and the onboard VGA device is disabled. **Note** The vKVM does not function when the onboard VGA is disabled. • **—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor. **Note** Only onboard VGA devices are supported with Cisco UCS B-Series servers.
ASPM Support	Allows you to set the level of ASPM (Active Power State Management) support in the BIOS. This can be one of the following: • **disabled—**ASPM support is disabled in the BIOS. • **auto—**The CPU determines the power state. • **force l0—**Force all links to L0 standby (L0s) state. • **—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor.

Boot Options BIOS Settings

The following table lists the boot options BIOS settings that you can configure through a BIOS policy or the default BIOS settings:

Name	Description
Boot Option Retry	Whether the BIOS retries NON-EFI based boot options without waiting for user input. This can be one of the following: • **disabled—**Waits for user input before retrying NON-EFI based boot options. • **enabled—**Continually retries NON-EFI based boot options without waiting for user input. • **—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor.
Intel Entry SAS RAID	Whether the Intel SAS Entry RAID Module is enabled. This can be one of the following: • **disabled—**The Intel SAS Entry RAID Module is disabled. • **enabled—**The Intel SAS Entry RAID Module is enabled. • **—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor.
Intel Entry SAS RAID Module	How the Intel SAS Entry RAID Module is configured. This can be one of the following: • **it-ir-raid—**Configures the RAID module to use Intel IT/IR RAID. • **intel-esrtii—**Configures the RAID module to use Intel Embedded Server RAID Technology II. • **—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor.
Onboard SCU Storage Support	Whether the onboard software RAID controller is available to the server. This can be one of the following: • **disabled—**The software RAID controller is not available. • **enabled—**The software RAID controller is available. • **—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor.

Server Management BIOS Settings

The following tables list the server management BIOS settings that you can configure through a BIOS policy or the default BIOS settings:

General Settings

Name	Description
Assert Nmi on Serr	Whether the BIOS generates a non-maskable interrupt (NMI) and logs an error when a system error (SERR) occurs. This can be one of the following: • **disabled**—The BIOS does not generate an NMI or log an error when a SERR occurs. • **enabled**—The BIOS generates an NMI and logs an error when a SERR occurs. You must enable this setting if you want to enable **Assert Nmi on Perr**. • **—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor.
Assert Nmi on Perr	Whether the BIOS generates a non-maskable interrupt (NMI) and logs an error when a processor bus parity error (PERR) occurs. This can be one of the following: • **disabled**—The BIOS does not generate an NMI or log an error when a PERR occurs. • **enabled**—The BIOS generates an NMI and logs an error when a PERR occurs. You must enable **Assert Nmi on Serr** to use this setting. • **—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor.

Name	Description
OS Boot Watchdog Timer	Whether the BIOS programs the watchdog timer with a predefined timeout value. If the operating system does not complete booting before the timer expires, the CIMC resets the system and an error is logged. This can be one of the following: • **disabled—**The watchdog timer is not used to track how long the server takes to boot. • **enabled—**The watchdog timer tracks how long the server takes to boot. If the server does not boot within the predefined length of time, the CIMC resets the system and logs an error. • **—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor. This feature requires either operating system support or Intel Management software.
OS Boot Watchdog Timer Timeout Policy	What action the system takes if the watchdog timer expires. This can be one of the following: • **power-off—**The server is powered off if the watchdog timer expires during OS boot. • **reset—**The server is reset if the watchdog timer expires during OS boot. • **—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor. This option is only available if you enable the OS Boot Watchdog Timer.
OS Boot Watchdog Timer Timeout	What timeout value the BIOS uses to configure the watchdog timer. This can be one of the following: • **5-minutes—**The watchdog timer expires 5 minutes after the OS begins to boot. • **10-minutes—**The watchdog timer expires 10 minutes after the OS begins to boot. • **15-minutes—**The watchdog timer expires 15 minutes after the OS begins to boot. • **20-minutes—**The watchdog timer expires 20 minutes after the OS begins to boot. • **—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor. This option is only available if you enable the OS Boot Watchdog Timer.

Name	Description
FRB-2 Timer	Whether the FRB-2 timer is used to recover the system if it hangs during POST. This can be one of the following: • **Disabled—**The FRB-2 timer is not used. • **Enabled—**The FRB-2 timer is started during POST and used to recover the system if necessary. • **—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor.

Console Redirection Settings

Name	Description
Console Redirection	Allows a serial port to be used for console redirection during POST and BIOS booting. After the BIOS has booted and the operating system is responsible for the server, console redirection is irrelevant and has no effect. This can be one of the following: • **disabled—**No console redirection occurs during POST. • **serial-port-a—**Enables serial port A for console redirection during POST. This option is valid for blade servers and rack-mount servers. • **serial-port-b—**Enables serial port B for console redirection and allows it to perform server management tasks. This option is only valid for rack-mount servers. • **—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor. **Note** If you enable this option, you also disable the display of the Quiet Boot logo screen during POST.
Flow Control	Whether a handshake protocol is used for flow control. Request to Send / Clear to Send (RTS/CTS) helps to reduce frame collisions that can be introduced by a hidden terminal problem. This can be one of the following: • **none—**No flow control is used. • **rts-cts—**RTS/CTS is used for flow control. • **—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor. **Note** This setting must match the setting on the remote terminal application.

Name	Description
BAUD Rate	What BAUD rate is used for the serial port transmission speed. If you disable Console Redirection, this option is not available. This can be one of the following: • **9600—**A 9600 BAUD rate is used. • **19200—**A 19200 BAUD rate is used. • **38400—**A 38400 BAUD rate is used. • **57600—**A 57600 BAUD rate is used. • **115200—**A 115200 BAUD rate is used. • **—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor. **Note** This setting must match the setting on the remote terminal application.
Terminal Type	What type of character formatting is used for console redirection. This can be one of the following: • **pc-ansi—**The PC-ANSI terminal font is used. • **vt100—**A supported vt100 video terminal and its character set are used. • **vt100-plus—**A supported vt100-plus video terminal and its character set are used. • **vt-utf8—**A video terminal with the UTF-8 character set is used. • **—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor. **Note** This setting must match the setting on the remote terminal application.
Legacy OS Redirect	Whether redirection from a legacy operating system, such as DOS, is enabled on the serial port. This can be one of the following: • **disabled—**The serial port enabled for console redirection is hidden from the legacy operating system. • **enabled—** The serial port enabled for console redirection is visible to the legacy operating system. • **—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor.

Name	Description
Putty KeyPad **set PuttyFunctionKeyPad**	Allows you to change the action of the PuTTY function keys and the top row of the numeric keypad. This can be one of the following: • **VT100—**The function keys generate ESC OP through ESC O[. • **LINUX—**Mimics the Linux virtual console. Function keys F6 to F12 behave like the default mode, but F1 to F5 generate ESC [[A through ESC [[E. • **XTERMR6—**Function keys F5 to F12 behave like the default mode. Function keys F1 to F4 generate **ESC OP** through ESC OS, which are the sequences produced by the top row of the keypad on Digital terminals. • **SCO—**The function keys F1 to F12 generate ESC [M through ESC [X. The function and shift keys generate ESC [Y through ESC [j. The control and function keys generate ESC [k through ESC [v. The shift, control and function keys generate ESC [w through ESC [{. • **ESCN—**The default mode. The function keys match the general behavior of Digital terminals. The function keys generate sequences such as ESC [11~ and ESC [12~. • **VT400—**The function keys behave like the default mode. The top row of the numeric keypad generates ESC OP through ESC OS. • **—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor.

IPMI Access Profile

This policy allows you to determine whether IPMI commands can be sent directly to the server, using the IP address. For example, you can send commands to retrieve sensor data from the CIMC. This policy defines the IPMI access, including a username and password that can be authenticated locally on the server, and whether the access is read-only or read-write.

You must include this policy in a service profile and that service profile must be associated with a server for it to take effect.

Configuring an IPMI Access Profile

Before You Begin

Obtain the following:

- Username with appropriate permission that can be authenticated by the operating system of the server
- Password for the username
- Permission associated with the username

Procedure

	Command or Action	Purpose
Step 1	UCSC# **connect policy-mgr**	Enters policy manager mode.
Step 2	UCSC(policy-mgr)# **scope org** *org-name*	Enters organization mode for the specified organization. To enter the root organization mode, type / as the *org-name* .
Step 3	UCSC(policy-mgr) /org # **create ipmi-access-profile** *profile-name*	Creates the specified IPMI access profile and enters organization IPMI access profile mode.
Step 4	UCSC(policy-mgr) /org/ipmi-access-profile # **create ipmi-user** *ipmi-user-name*	Creates the specified endpoint user and enters organization IPMI access profile endpoint user mode. **Note** More than one endpoint user can be created within an IPMI access profile, with each endpoint user having its own password and privileges.
Step 5	UCSC(policy-mgr) /org/ipmi-access-profile/ipmi-user # **set password**	Sets the password for the endpoint user. After entering the **set password** command, you are prompted to enter and confirm the password. For security purposes, the password that you type does not appear in the CLI.
Step 6	UCSC(policy-mgr) /org/ipmi-access-profile/ipmi-user # **set privilege** {**admin** | **readonly**}	Specifies whether the endpoint user has administrative or read-only privileges.
Step 7	UCSC(policy-mgr) /org/ipmi-access-profile/ipmi-user # **commit-buffer**	Commits the transaction to the system configuration.

The following example creates an IPMI access profile named ReadOnly, creates an endpoint user named bob, sets the password and the privileges for bob, and commits the transaction:

```
UCSC# connect policy-mgr
UCSC(policy-mgr)# scope org /
UCSC(policy-mgr) /org # create ipmi-access-profile ReadOnly
UCSC(policy-mgr) /org/ipmi-access-profile* # create ipmi-user bob
UCSC(policy-mgr) /org/ipmi-access-profile/ipmi-user* # set password
Enter a password:
Confirm the password:
UCSC(policy-mgr) /org/ipmi-access-profile/ipmi-user* # set privilege readonly
UCSC(policy-mgr) /org/ipmi-access-profile/ipmi-user* # commit-buffer
UCSC(policy-mgr) /org/ipmi-access-profile/ipmi-user #
```

What to Do Next

Include the IPMI profile in a service profile and/or template.

Deleting an IPMI Access Profile

Procedure

	Command or Action	Purpose
Step 1	UCSC# **connect policy-mgr**	Enters policy manager mode.
Step 2	UCSC(policy-mgr)# **scope org** *org-name*	Enters organization mode for the specified organization. To enter the root organization mode, type / as the *org-name*.
Step 3	UCSC(policy-mgr) /org # **delete ipmi-access-profile** *profile-name*	Deletes the specified IPMI access profile.
Step 4	UCSC(policy-mgr) /org # **commit-buffer**	Commits the transaction to the system configuration.

The following example deletes the IPMI access profile named ReadOnly and commits the transaction:

```
UCSC# connect policy-mgr
UCSC(policy-mgr)# scope org /
UCSC(policy-mgr) /org # delete ipmi-access-profile ReadOnly
UCSC(policy-mgr) /org* # commit-buffer
UCSC(policy-mgr) /org #
```

Adding an Endpoint User to an IPMI Access Profile

Procedure

	Command or Action	Purpose
Step 1	UCSC# **connect policy-mgr**	Enters policy manager mode.
Step 2	UCSC(policy-mgr)# **scope org** *org-name*	Enters organization mode for the specified organization. To enter the root organization mode, type / as the *org-name* .
Step 3	UCSC(policy-mgr) /org # **scope ipmi-access-profile** *profile-name*	Enters organization IPMI access profile mode for the specified IPMI access profile.
Step 4	UCSC(policy-mgr) /org/ipmi-access-profile # **create ipmi-user** *ipmi-user-name*	Creates the specified endpoint user and enters organization IPMI access profile endpoint user mode. **Note** More than one endpoint user can be created within an IPMI access profile, with each endpoint user having its own password and privileges.

	Command or Action	Purpose
Step 5	UCSC(policy-mgr) /org/ipmi-access-profile/ipmi-user # **set password**	Sets the password for the endpoint user. After entering the **set password** command, you are prompted to enter and confirm the password. For security purposes, the password that you type does not appear in the CLI.
Step 6	UCSC(policy-mgr) /org/ipmi-access-profile/ipmi-user # **set privilege** {**admin** | **readonly**}	Specifies whether the endpoint user has administrative or read-only privileges.
Step 7	UCSC(policy-mgr) /org/ipmi-access-profile/ipmi-user # **commit-buffer**	Commits the transaction to the system configuration.

The following example adds an endpoint user named alice to the IPMI access profile named ReadOnly and commits the transaction:

```
UCSC# connect policy-mgr
UCSC(policy-mgr)# scope org /
UCSC(policy-mgr) /org* # scope ipmi-access-profile ReadOnly
UCSC(policy-mgr) /org/ipmi-access-profile* # create ipmi-user alice
UCSC(policy-mgr) /org/ipmi-access-profile/ipmi-user* # set password
Enter a password:
Confirm the password:
UCSC(policy-mgr) /org/ipmi-access-profile/ipmi-user* # set privilege readonly
UCSC(policy-mgr) /org/ipmi-access-profile/ipmi-user* # commit-buffer
UCSC(policy-mgr) /org/ipmi-access-profile/ipmi-user #
```

Deleting an Endpoint User from an IPMI Access Profile

Procedure

	Command or Action	Purpose
Step 1	UCSC# **connect policy-mgr**	Enters policy manager mode.
Step 2	UCSC(policy-mgr)# **scope org** *org-name*	Enters organization mode for the specified organization. To enter the root organization mode, type / as the *org-name*.
Step 3	UCSC(policy-mgr) /org # **scope ipmi-access-profile** *profile-name*	Enters organization IPMI access profile mode for the specified IPMI access profile.
Step 4	UCSC(policy-mgr) /org/ipmi-access-profile # **delete ipmi-user** *epuser-name*	Deletes the specified endpoint user from the IPMI access profile.
Step 5	UCSC(policy-mgr) /org/ipmi-access-profile # **commit-buffer**	Commits the transaction to the system configuration.

The following example deletes the endpoint user named alice from the IPMI access profile named ReadOnly and commits the transaction:

```
UCSC# connect policy-mgr
UCSC(policy-mgr)# scope org /
UCSC(policy-mgr) /org # scope ipmi-access-profile ReadOnly
UCSC(policy-mgr) /org/ipmi-access-profile # delete ipmi-user alice
UCSC(policy-mgr) /org/ipmi-access-profile* # commit-buffer
UCSC(policy-mgr) /org/ipmi-access-profile #
```

Boot Policy

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 choose to 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 that can be associated 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. Reassociation of the service profile with the server to rewrite the boot order information in the BIOS is automatically triggered.

Creating a Boot Policy

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.

Procedure

	Command or Action	Purpose
Step 1	UCSC# **connect policy-mgr**	Enters policy manager mode.
Step 2	UCSC(policy-mgr)# **scope org** *org-name*	Enters organization mode for the specified organization. To enter the root organization mode, type / as the *org-name* .

	Command or Action	Purpose
Step 3	UCSC(policy-mgr) /org # **create boot-policy** *policy-name* [**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 4	UCSC(policy-mgr) /org/boot-policy # **set descr** *description*	(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 5	UCSC(policy-mgr) /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 6	UCSC(policy-mgr) /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	UCSC(policy-mgr) /org/boot-policy # **set enforce-vnic-name** {**no** | **yes**}	If you choose **yes**, Cisco UCS Central uses any vNICs or vHBAs defined in the **Boot Order**. If you choose **no**, Cisco UCS Central uses the priority specified in the vNIC or vHBA.
Step 8	UCSC(policy-mgr) /org/boot-policy # **commit-buffer**	Commits the transaction to the system configuration.
Step 9	UCSC(policy-mgr) /org/boot-policy # **create boot-security**	Enters boot security mode for the specified boot policy.
Step 10	UCSC(policy-mgr) /org/boot-policy/boot-security # **set secure-boot** {**no** | **yes**}	Specifies whether secure boot is enabled for the boot policy.
Step 11	UCSC(policy-mgr) /org/boot-policy/boot-security # **commit-buffer**	Commits the transaction to the system configuration.

The following example creates a boot policy named boot-policy-LAN, provides a description for the boot policy, specifies that servers using this policy will not be automatically rebooted when the boot order is changed, and commits the transaction:

```
UCSC# connect policy-mgr
UCSC(policy-mgr)# scope org /
UCSC(policy-mgr) /org* # create boot-policy boot-policy-LAN purpose operational
UCSC(policy-mgr) /org/boot-policy* # set descr "Boot policy that boots from the LAN."
```

```
UCSC(policy-mgr) /org/boot-policy* # set reboot-on-update no
UCSC(policy-mgr) /org/boot-policy* # set boot-mode uefi
UCSC(policy-mgr) /org/boot-policy* # commit-buffer
UCSC(policy-mgr) /org/boot-policy*# create boot-security
UCSC(policy-mgr) /org/boot-policy* # set secure-boot yes
UCSC(policy-mgr) /org/boot-policy* # commit-buffer
UCSC(policy-mgr) /org/boot-policy* #
UCSC(policy-mgr) /org/boot-policy #
```

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 51.

- **Storage 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, 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 Storage Boot option, continue to Configuring a SAN Boot for a Boot Policy, on page 53.

- **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 60.

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	UCSC# **connect policy-mgr**	Enters policy manager mode.

	Command or Action	Purpose
Step 2	UCSC(policy-mgr)# **scope org** *org-name*	Enters organization mode for the specified organization. To enter the root organization mode, type / as the *org-name* .
Step 3	UCSC(policy-mgr) /org # **scope boot-policy** *policy-name*	Enters organization boot policy mode for the specified boot policy.
Step 4	UCSC(policy-mgr) /org/boot-policy # **create lan**	Creates a LAN boot for the boot policy and enters organization boot policy LAN mode.
Step 5	UCSC(policy-mgr) /org/boot-policy/lan # **set order** {**1** | **2** | **3** | **4**}	Specifies the boot order for the LAN boot.
Step 6	UCSC(policy-mgr) /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 7	UCSC(policy-mgr) /org/boot-policy/lan/path # **set vnic** *vnic-name*	Specifies the vNIC to use for the LAN path to the boot image.
Step 8	UCSC(policy-mgr) /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:

```
UCSC# connect policy-mgr
UCSC(policy-mgr)# scope org /
UCSC(policy-mgr) /org* # scope boot-policy lab2-boot-policy
UCSC(policy-mgr) /org/boot-policy* # create lan
UCSC(policy-mgr) /org/boot-policy/lan* # set order 2
UCSC(policy-mgr) /org/boot-policy/lan* # create path primary
UCSC(policy-mgr) /org/boot-policy/lan/path* # set vnic vNIC1
UCSC(policy-mgr) /org/boot-policy/lan/path* # exit
UCSC(policy-mgr) /org/boot-policy/lan* # create path secondary
UCSC(policy-mgr) /org/boot-policy/lan/path* # set vnic vNIC2
UCSC(policy-mgr) /org/boot-policy/lan/path* # commit-buffer
UCSC(policy-mgr) /org/boot-policy/lan/path #
```

What to Do Next

Include the boot policy in a service profile and/or 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.

We recommend that you use 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 exact same operating system image. Therefore, the new server appears to be the exact 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 & rack servers.

Configuring a SAN Boot for a Boot Policy

Note We recommend that the boot order in a boot policy include either a local disk or a SUN LUN, but not both, to avoid the possibility of the server booting from the wrong storage type. If you configure a local disk and a SUN 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 configures with default LV names and the boot order is configured with a SUN 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 49.

Before You Begin

Create a boot policy to contain the SAN boot configuration.

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

Procedure

	Command or Action	Purpose
Step 1	UCSC# **connect policy-mgr**	Enters policy manager mode.
Step 2	UCSC(policy-mgr)# **scope org** *org-name*	Enters organization mode for the specified organization. To enter the root organization mode, type / as the *org-name* .
Step 3	UCSC(policy-mgr) /org # **scope boot-policy** *policy-name*	Enters organization boot policy mode for the specified boot policy.

	Command or Action	Purpose
Step 4	UCSC(policy-mgr) /org/boot-policy # **create storage**	Creates a SAN boot for the boot policy and enters organization boot policy storage mode.
Step 5	UCSC(policy-mgr) /org/boot-policy/storage # **set order** {**1** | **2** | **3** | **4**}	Sets the boot order for the SAN boot.
Step 6	UCSC(policy-mgr) /org/boot-policy/storage # **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. The use of the terms primary or secondary boot devices does not imply a boot order. The effective order of boot devices within the same device class is determined by PCIe bus scan order.
Step 7	UCSC(policy-mgr) /org/boot-policy/storage/san-image # **set vhba** *vhba-name*	Specifies the vHBA to be used for the SAN boot.
Step 8	UCSC(policy-mgr) /org/boot-policy/storage/san-image # **create path** {**primary** | **secondary**}	Creates a primary or secondary SAN boot path and enters organization boot policy SAN path mode. The use of the terms primary or secondary boot devices does not imply a boot order. The effective order of boot devices within the same device class is determined by PCIe bus scan order.
Step 9	UCSC(policy-mgr) /org/boot-policy/storage/san-image/path # **set** {**lun** *lun-id* | **wwn** *wwn-num*}	Specifies the LUN or WWN to be used for the SAN path to the boot image.
Step 10	UCSC(policy-mgr) /org/boot-policy/storage/san-image/path # **commit-buffer**	Commits the transaction to the system configuration.

The following example enters the boot policy named lab1-boot-policy, creates a SAN boot for the policy, sets the boot order to 1, creates a primary SAN image, uses a vHBA named vHBA2, creates primary path using LUN 967295200, and commits the transaction:

```
UCSC# connect policy-mgr
UCSC(policy-mgr)# scope org /
UCSC(policy-mgr) /org* # scope boot-policy lab1-boot-policy
UCSC(policy-mgr) /org/boot-policy* # create storage
UCSC(policy-mgr) /org/boot-policy/storage* # set order 1
UCSC(policy-mgr) /org/boot-policy/storage* # create san-image primary
UCSC(policy-mgr) /org/boot-policy/storage* # set vhba vHBA2
UCSC(policy-mgr) /org/boot-policy/storage/san-image* # create path primary
UCSC(policy-mgr) /org/boot-policy/storage/san-image/path* # set lun 967295200
UCSC(policy-mgr) /org/boot-policy/storage/san-image/path* # commit-buffer
UCSC(policy-mgr) /org/boot-policy/storage/san-image/path #
```

What to Do Next

Include the boot policy in a service profile and/or 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

There are prerequisites that must be met before you configure iSCSI boot. For a list of these prerequisites, see iSCSI Boot Guidelines and Prerequisites.

Configuring an iSCSI Boot for a Boot Policy

Before You Begin

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	UCSC# **connect policy-mgr**	Enters policy manager mode.
Step 2	UCSC(policy-mgr)# **scope org** *org-name*	Enters organization mode for the specified organization. To enter the root organization mode, type / as the *org-name* .
Step 3	UCSC(policy-mgr) /org # **scope boot-policy** *policy-name*	Enters organization boot policy mode for the specified boot policy.
Step 4	UCSC(policy-mgr) /org/boot-policy # **create iscsi**	Adds an iSCSI boot to the boot policy and enters iSCSI mode.
Step 5	UCSC(policy-mgr) /org/boot-policy/iscsi # **create path** {**primary** | **secondary**}	Specifies the primary and secondary paths that Cisco UCS Central uses to reach the iSCSI target. With iSCSI boot, you set up two paths. Cisco UCS Central uses the primary path first, and if that fails, then it uses the secondary path.

	Command or Action	Purpose
Step 6	UCSC(policy-mgr) /org/boot-policy/iscsi/path # **set iscsivnicname** *vnic-name*	Specifies the vNIC to use for the iSCSI path to the boot image.
Step 7	UCSC(policy-mgr) /org/boot-policy/iscsi/path # **exit**	Exits iSCSI path mode.
Step 8	UCSC(policy-mgr) /org/boot-policy/iscsi # **set order** *ordernum*	Specifies the order for the iSCSI boot in the boot order.
Step 9	UCSC(policy-mgr) /org/boot-policy/iscsi # **commit-buffer**	Commits the transaction to the system configuration.

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

```
UCSC# connect policy-mgr
UCSC(policy-mgr)# scope org /
UCSC(policy-mgr) /org* # scope boot-policy lab2-boot-policy
UCSC(policy-mgr) /org/boot-policy # create iscsi
UCSC(policy-mgr) /org/boot-policy/iscsi* # create path primary
UCSC(policy-mgr) /org/boot-policy/iscsi/path* # set vnic vNIC1
UCSC(policy-mgr) /org/boot-policy/iscsi/path* # exit
UCSC(policy-mgr) /org/boot-policy/iscsi* # set order 2
UCSC(policy-mgr) /org/boot-policy/iscsi* # create path secondary
UCSC(policy-mgr) /org/boot-policy/iscsi/path* # set vnic vNIC2
UCSC(policy-mgr) /org/boot-policy/iscsi/path* # commit-buffer
UCSC(policy-mgr) /org/boot-policy/iscsi/path #
```

What to Do Next

Include the boot policy in a service profile and/or template.

Creating an iSCSI Adapter Policy

Procedure

	Command or Action	Purpose
Step 1	UCSC# **connect policy-mgr**	Enters policy manager mode.
Step 2	UCSC(policy-mgr) # **scope org** *org-name*	Enters organization mode for the specified organization. To enter the root organization mode, type / as the *org-name*.
Step 3	UCSC(policy-mgr) /org # **create iscsi-policy** *policy-name*	Creates the iSCSI adapter policy.
Step 4	UCSC(policy-mgr) /org/iscsi-policy # **set descr** *description*	(Optional) Provides a description for the iSCSI adapter policy.

	Command or Action	Purpose
Step 5	UCSC(policy-mgr) /org/iscsi-policy # **set iscsi-protocol-item connection-timeout** *timeout-secs*	The number of seconds until Cisco UCS Central 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 Central uses the value set in the adapter firmware (default: 15 seconds).
Step 6	UCSC(policy-mgr) /org/iscsi-policy # **set iscsi-protocol-item dhcp-timeout** *timeout-secs*	The number of seconds to wait before the initiator assumes that the DHCP server in unavailable. Enter an integer between 60 and 300 (default: 60 seconds).
Step 7	UCSC(policy-mgr) /org/iscsi-policy # **set iscsi-protocol-item lun-busy-retry-count** *num*	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 Central uses the value set in the adapter firmware (default: 15 seconds).
Step 8	UCSC(policy-mgr) /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 9	UCSC(policy-mgr) /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 10	UCSC(policy-mgr) /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 11	UCSC(policy-mgr) /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:

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

```
UCSC(policy-mgr) /org/iscsi-policy* # commit-buffer
UCSC(policy-mgr) /org/iscsi-policy #
```

What to Do Next

Include the adapter policy in a service profile and/or template.

Deleting an iSCSI Adapter Policy

Procedure

	Command or Action	Purpose
Step 1	UCSC# **connect policy-mgr**	Enters policy manager mode.
Step 2	UCSC(policy-mgr)# **scope org** *org-name*	Enters organization mode for the specified organization. To enter the root organization mode, type / as the *org-name*.
Step 3	UCSC(policy-mgr) /org # **delete iscsi-policy** *policy-name*	Deletes the iSCSI adapter policy.
Step 4	UCSC(policy-mgr) /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:

```
UCSC# connect policy-mgr
UCSC(policy-mgr)# scope org /
UCSC(policy-mgr) /org # delete iscsi-policy iscsi-adapter-pol
UCSC(policy-mgr) /org* # commit-buffer
UCSC(policy-mgr) /org #
```

Creating an iSCSI 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	UCSC# **connect policy-mgr**	Enters policy manager mode.
Step 2	UCSC(policy-mgr) # **scope org** *org-name*	Enters organization mode for the specified organization. To enter the root organization mode, type / as the *org-name*.
Step 3	UCSC(policy-mgr) /org # **create auth-profile** *profile-name*	Creates an authentication profile with the specified name. The name can be up to 16 alphanumeric characters.

	Command or Action	Purpose
Step 4	UCSC(policy-mgr) /org/auth-profile # **set user-id** *id-name*	Creates a log in for authentication.
Step 5	UCSC(policy-mgr) /org/auth-profile # **set password**	Creates a password for authentication.
Step 6	UCSC(policy-mgr) /org/auth-profile # **commit-buffer**	Commits the transaction to the system configuration.
Step 7	UCSC(policy-mgr) /org/auth-profile # **exit**	Exits the current mode.
Step 8	Repeat steps 3 through 7 to create an authentication profile for the target.	
Step 9	UCSC(policy-mgr) /org/auth-profile # **commit-buffer**	Commits the transaction to the system configuration.

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

```
UCSC# connect policy-mgr
UCSC(policy-mgr)# scope org
UCSC(policy-mgr) /org # create auth-profile InitAuth
UCSC(policy-mgr) /org/auth-profile* # set user-id init
UCSC(policy-mgr) /org/auth-profile* # set  password
Enter a password:
Confirm the password:
UCSC(policy-mgr) /org/auth-profile* # commit-buffer
UCSC(policy-mgr) /org/auth-profile # exit
UCSC(policy-mgr) /org # create auth-profile TargetAuth
UCSC(policy-mgr) /org/auth-profile* # set user-id target
UCSC(policy-mgr) /org/auth-profile* # set  password
Enter a password:
Confirm the password:
UCSC(policy-mgr) /org/auth-profile* # commit-buffer
UCSC(policy-mgr) /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 iSCSI Authentication Profile

Procedure

	Command or Action	Purpose
Step 1	UCSC# **connect policy-mgr**	Enters policy manager mode.
Step 2	UCSC(policy-mgr) # **scope org** *org-name*	Enters organization mode for the specified organization. To enter the root organization mode, type / as the *org-name*.

	Command or Action	Purpose
Step 3	UCSC(policy-mgr) /org # **delete auth-profile** *profile-name*	Deletes the specified iSCSI authentication profile.
Step 4	UCSC(policy-mgr) /org # **commit-buffer**	Commits the transaction to the system configuration.

The following example shows how to delete an iSCSI authentication profile and commit the transaction:

```
UCSC# connect policy-mgr
UCSC(policy-mgr)# scope org
UCSC(policy-mgr) /org # delete auth-profile InitAuth
UCSC(policy-mgr) /org* # commit-buffer
UCSC(policy-mgr) /org #
```

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.

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, we recommend 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	UCSC# **connect policy-mgr**	Enters policy manager mode.
Step 2	UCSC(policy-mgr)# **scope org** *org-name*	Enters organization mode for the specified organization. To enter the root organization mode, type / as the *org-name* .

	Command or Action	Purpose
Step 3	UCSC(policy-mgr) /org # **scope boot-policy** *policy-name*	Enters organization boot policy mode for the specified boot policy.
Step 4	UCSC(policy-mgr) /org/boot-policy # **create virtual-media** {**read-only** | **read-write**}	Creates a virtual media boot for the boot policy, specifies whether the virtual media is has read-only or read-write privileges, and enters organization boot policy virtual media mode.
Step 5	UCSC(policy-mgr) /org/boot-policy/virtual-media # **set order** {**1** | **2** | **3** | **4**}	Sets the boot order for the virtual-media boot.
Step 6	UCSC(policy-mgr) /org/boot-policy/virtual-media # **commit-buffer**	Commits the transaction to the system configuration.

The following example enters the boot policy named lab3-boot-policy, creates a virtual media boot with read-only privileges for the policy, sets the boot order to 3, and commits the transaction:

```
UCSC# connect policy-mgr
UCSC(policy-mgr)# scope org /
UCSC(policy-mgr) /org* # scope boot-policy lab3-boot-policy
UCSC(policy-mgr) /org/boot-policy* # create virtual-media read-only
UCSC(policy-mgr) /org/boot-policy/virtual-media* # set order 3
UCSC(policy-mgr) /org/boot-policy/virtual-media* # commit-buffer
```

What to Do Next

Include the boot policy in a service profile and/or template.

Deleting a Boot Policy

Procedure

	Command or Action	Purpose
Step 1	UCSC# **connect policy-mgr**	Enters policy manager mode.
Step 2	UCSC(policy-mgr) # **scope org** *org-name*	Enters organization mode for the specified organization. To enter the root organization mode, type / as the *org-name*.
Step 3	UCSC(policy-mgr) /org # **delete boot-policy** *policy-name*	Deletes the specified boot policy.
Step 4	UCSC(policy-mgr) /org # **commit-buffer**	Commits the transaction to the system configuration.

The following example shows how to delete a boot policy and commits the transaction:

```
UCSC# connect policy-mgr
UCSC(policy-mgr)# scope org /
UCSC(policy-mgr) /org # delete boot-policy boot-policy-LAN
UCSC(policy-mgr) /org* # commit-buffer
UCSC(policy-mgr) /org #
```

Local Disk Configuration Policy

This policy configures any optional SAS local drives that have been installed on a server through the onboard RAID controller of the local drive. This policy enables you to set a local disk mode for all servers that are associated with a service profile that includes the local disk configuration policy.

The local disk modes include the following:

- **No Local Storage—**For a diskless server or a SAN only configuration. If you select this option, you cannot associate any service profile which uses this policy with a server that has a local disk.
- **RAID 0 Striped—**Data is striped across all disks in the array, providing fast throughput. There is no data redundancy, and all data is lost if any disk fails.
- **RAID 1 Mirrored—**Data is written to two disks, providing complete data redundancy if one disk fails. The maximum array size is equal to the available space on the smaller of the two drives.
- **Any Configuration—**For a server configuration that carries forward the local disk configuration without any changes.
- **No RAID—**For a server configuration that removes the RAID and leaves the disk MBR and payload unaltered.

  If you choose **No RAID** and you apply this policy to a server that already has an operating system with RAID storage configured, the system does not remove the disk contents. Therefore, there may be no visible differences on the server after you apply the **No RAID** mode. This can lead to a mismatch between the RAID configuration in the policy and the actual disk configuration shown in the **Inventory** > **Storage** tab for the server.

  To make sure that any previous RAID configuration information is removed from a disk, apply a scrub policy that removes all disk information after you apply the **No RAID** configuration mode.
- **RAID 5 Striped Parity—**Data is striped across all disks in the array. Part of the capacity of each disk stores parity information that can be used to reconstruct data if a disk fails. RAID 5 provides good data throughput for applications with high read request rates.
- **RAID 6 Striped Dual Parity—**Data is striped across all disks in the array and two parity disks are used to provide protection against the failure of up to two physical disks. In each row of data blocks, two sets of parity data are stored.
- **RAID 10 Mirrored and Striped—**RAID 10 uses mirrored pairs of disks to provide complete data redundancy and high throughput rates.
- **RAID 50 Striped Parity and Striped —**Data is striped across multiple striped parity disk sets to provide high throughput and multiple disk failure tolerance.
- **RAID 60 Striped Dual Parity and Striped —**Data is striped across multiple striped dual parity disk sets to provide high throughput and greater disk failure tolerance.

You must include this policy in a service profile and that service profile must be associated with a server for the policy to take effect.

Note For a Cisco UCS C-Series server integrated with Cisco UCS Manager, with an embedded on-board RAID controller, the local disk mode should always be **Any Configuration**, and the RAID must be configured directly on the controller.

Guidelines for all Local Disk Configuration Policies

Before you create a local disk configuration policy, consider the following guidelines:

No Mixed HDDs and SSDs

Do not include HDDs and SSDs in a single server or RAID configuration.

Do Not Assign a Service Profile with the Default Local Disk Configuration Policy from a B200 M1 or M2 to a B200 M3

Due to the differences in the RAID/JBOD support provided by the storage controllers of B200 M1 and M2 servers and those of the B200 M3 server, you cannot assign or re-assign a service profile that includes the default local disk configuration policy from a B200M1 or M2 server to a B200 M3 server. The default local disk configuration policy includes those with Any Configuration or JBOD configuration.

JBOD Mode Support

Note Only B200 M1, B200 M2, B200 M3, B250 M1, B250 M2 and B22 M3 blade servers support the JBOD mode for local disks.

Guidelines for Local Disk Configuration Policies Configured for RAID

Configure RAID Settings in Local Disk Configuration Policy for Servers with MegaRAID Storage Controllers

If a blade server or integrated rack-mount server has a MegaRAID controller, you must configure RAID settings for the drives in the Local Disk Configuration policy included in the service profile for that server. You can do this either by configuring the local disk configuration policy in the service profile using one of the defined RAID modes for that server, or you can use the **Any Configuration** mode with the LSI Utilities toolset to create the RAID volumes.

If you do not configure your RAID LUNs before installing the OS, disk discovery failures might occur during the installation and you might see error messages such as “No Device Found.”

Server May Not Boot After RAID1 Cluster Migration if Any Configuration Mode Specified in Service Profile

After RAID1 clusters are migrated, you need to associate a service profile with the server. If the local disk configuration policy in the service profile is configured with **Any Configuration** mode rather than **RAID1**, the RAID LUN remains in "inactive" state during and after association. As a result, the server cannot boot.

To avoid this issue, ensure that the service profile you associate with the server contains the identical local disk configuration policy as the original service profile before the migration and does not include the **Any Configuration** mode.

Do Not Use JBOD Mode on Servers with MegaRAID Storage Controllers

Do not configure or use JBOD mode or JBOD operations on any blade server or integrated rack-mount server with a MegaRAID storage controllers. JBOD mode and operations are not intended for nor are they fully functional on these servers.

Maximum of One RAID Volume and One RAID Controller in Integrated Rack-Mount Servers

A rack-mount server that has been integrated with Cisco UCS Manager and is registered with Cisco UCS Central can have a maximum of one RAID volume irrespective of how many hard drives are present on the server.

All the local hard drives in an integrated rack-mount server must be connected to only one RAID Controller. Integration with Cisco UCS Manager does not support the connection of local hard drives to multiple RAID Controllers in a single rack-mount server. We therefore recommend that you request a single RAID Controller configuration when you order rack-mount servers to be integrated with Cisco UCS Manager.

In addition, do not use third party tools to create multiple RAID LUNs on rack-mount servers. Cisco UCS Manager does not support that configuration.

Maximum of One RAID Volume and One RAID Controller in Blade Servers

A blade server can have a maximum of one RAID volume irrespective of how many drives are present in the server. All the local hard drives must be connected to only one RAID controller. For example, a B200 M3 server has an LSI controller and an Intel Patsburg controller, but only the LSI controller can be used as a RAID controller.

In addition, do not use third party tools to create multiple RAID LUNs on blade servers. Cisco UCS Central does not support that configuration.

Number of Disks Selected in Mirrored RAID Should Not Exceed Two

If the number of disks selected in the Mirrored RAID exceed two, RAID 1 is created as a RAID 10 LUN. This issue can occur with the Cisco UCS B440 M1 and B440 M2 servers.

License Required for Certain RAID Configuration Options on Some Servers

Some Cisco UCS servers require a license for certain RAID configuration options. When Cisco UCS Central associates a service profile containing this local disk policy with a server, Cisco UCS Central verifies that the selected RAID option is properly licensed. If there are issues, Cisco UCS Central displays a configuration error during the service profile association.

For RAID license information for a specific Cisco UCS server, see the *Hardware Installation Guide* for that server.

B420 M3 Server Does Not Support All Configuration Modes

The B420 M3 server does not support the following configuration modes in a local disk configuration policy:

- No RAID
- RAID 6 Striped Dual Parity

In addition, the B420 M3 does not support JBOD modes or operations.

Single-Disk RAID 0 Configurations Not Supported on Some Blade Servers

A single-disk RAID 0 configuration is not supported in the following blade servers:

- Cisco UCS B200 M1
- Cisco UCS B200 M2
- Cisco UCS B250 M1
- Cisco UCS B250 M2

Creating a Local Disk Configuration Policy

Procedure

	Command or Action	Purpose
Step 1	UCSC# **connect policy-mgr**	Enters policy manager mode.
Step 2	UCSC(policy-mgr)# **scope org** *org-name*	Enters organization mode for the specified organization. To enter the root organization mode, type / as the *org-name* .
Step 3	UCSC(policy-mgr) /org # **create local-disk-config-policy** *policy-name*	Creates a local disk configuration policy and enters local disk configuration policy mode.
Step 4	UCSC(policy-mgr) /org/local-disk-config-policy # **set descr** *description*	(Optional) Provides a description for the local disk configuration policy.
Step 5	UCSC(policy-mgr) /org/local-disk-config-policy # **set mode** {**any-configuration** | **no-local-storage** | **no-raid** | **raid-0-striped** | **raid-1-mirrored** | **raid-5-striped-parity** | **raid-6-striped-dual-parity** | **raid-10-mirrored-and-striped**}	Specifies the mode for the local disk configuration policy.
Step 6	UCSC(policy-mgr) /org/local-disk-config-policy # **set protect** {**yes** | **no**}	Specifies whether the server retains the configuration in the local disk configuration policy even if the server is disassociated from the service profile. **Caution** Protect Configuration becomes non-functional if one or more disks in the server are defective or faulty. When a service profile is disassociated from a server and a new service profile associated, the setting for the Protect Configuration property in the new service profile takes precedence and overwrites the setting in the previous service profile.

	Command or Action	Purpose
		Note If you disassociate the server from a service profile with this option enabled and then associate it with a new service profile that includes a local disk configuration policy with different properties, the server returns a configuration mismatch error and the association fails.
Step 7	UCSC(policy-mgr) /org/local-disk-config-policy # **commit-buffer**	Commits the transaction to the system configuration.

The following example configures a local disk configuration policy and commits the transaction:

```
UCSC# connect policy-mgr
UCSC(policy-mgr)# scope org /
UCSC(policy-mgr) /org # create local-disk-config-policy DiskPolicy7
UCSC(policy-mgr) /org/local-disk-config-policy* # set mode raid-1-mirrored
UCSC(policy-mgr) /org/local-disk-config-policy* # set protect yes
UCSC(policy-mgr) /org/local-disk-config-policy* # commit-buffer
UCSC(policy-mgr) /org/local-disk-config-policy #
```

Viewing a Local Disk Configuration Policy

Procedure

	Command or Action	Purpose
Step 1	UCSC# **connect policy-mgr**	Enters policy manager mode.
Step 2	UCSC(policy-mgr)# **scope org** *org-name*	Enters organization mode for the specified organization. To enter the root organization mode, type / as the *org-name* .
Step 3	UCSC(policy-mgr) /org # **show local-disk-config-policy** *policy-name*	Displays the local disk policy. If you have not configured a local disk policy, the local disk configuration (created by the create local-disk-config command) displays. Displays the local disk definition (set by the create local-disk-config command). If the serial over LAN definition is not set, and if a policy is set (using the set local-disk-config-policy command), then the policy will be displayed.

The following example shows how to display local disk policy information for a local disk configuration policy called DiskPolicy7:

```
UCSC# connect policy-mgr
UCSC(policy-mgr)# scope org /
UCSC(policy-mgr) /org # show local-disk-config-policy DiskPolicy7

Local Disk Config Policy:
```

```
Name: DiskPolicy7
Mode: Raid 1 Mirrored
Description:
Protect Configuration: Yes
```

Deleting a Local Disk Configuration Policy

Procedure

	Command or Action	Purpose
Step 1	UCSC# **connect policy-mgr**	Enters policy manager mode.
Step 2	UCSC(policy-mgr)# **scope org** *org-name*	Enters organization mode for the specified organization. To enter the root organization mode, type / as the *org-name*.
Step 3	UCSC(policy-mgr) /org # **delete local-disk-config-policy** *policy-name*	Deletes the specified local disk configuration policy.
Step 4	UCSC(policy-mgr) /org # **commit-buffer**	Commits the transaction to the system configuration.

The following example deletes the local disk configuration policy named DiskPolicy7 and commits the transaction:

```
UCSC# connect policy-mgr
UCSC(policy-mgr)# scope org /
UCSC(policy-mgr) /org # delete local-disk-config-policy DiskPolicy7
UCSC(policy-mgr) /org* # commit-buffer
UCSC(policy-mgr) /org #
```

Power Control Policy

Cisco UCS uses the priority set in the power control policy, along with the blade type and configuration, to calculate the initial power allocation for each blade within a chassis. During normal operation, the active blades within a chassis can borrow power from idle blades within the same chassis. If all blades are active and reach the power cap, service profiles with higher priority power control policies take precedence over service profiles with lower priority power control policies.

Priority is ranked on a scale of 1-10, where 1 indicates the highest priority and 10 indicates lowest priority. The default priority is 5.

For mission-critical application a special priority called no-cap is also available. Setting the priority to no-cap prevents Cisco UCS from leveraging unused power from a particular server. With this setting, the server is allocated the maximum amount of power possible for that type of server.

Note You must include this policy in a service profile and that service profile must be associated with a server for it to take effect.

Creating a Power Control Policy

Procedure

	Command or Action	Purpose
Step 1	UCSC# **connect policy-mgr**	Enters policy manager mode.
Step 2	UCSC(policy-mgr) # **scope org** *org-name*	Enters organization mode for the specified organization. To enter the root organization mode, type / as the *org-name*.
Step 3	UCSC(policy-mgr) /org # **create power-control-policy** *policy-name*	Creates a power control policy and enters power control policy mode.
Step 4	UCSC(policy-mgr) /org/power-control-policy # **set priority** {*priority-num* | **no-cap**}	Specifies the priority for the power control policy.
Step 5	UCSC(policy-mgr) /org/power-control-policy # **commit-buffer**	Commits the transaction to the system configuration.

The following example shows how to create a power control policy and commits the transaction:

```
UCSC# connect policy-mgr
UCSC(policy-mgr)# scope org /
UCSC(policy-mgr) /org # create power-control-policy PCP-1
UCSC(policy-mgr) /org/power-control-policy* # set priority 1
UCSC(policy-mgr) /org/power-control-policy* # commit-buffer
UCSC(policy-mgr) /org/power-control-policy #
```

Deleting a Power-Control-Policy

Procedure

	Command or Action	Purpose
Step 1	UCSC# **connect policy-mgr**	Enters policy manager mode.
Step 2	UCSC(policy-mgr) # **scope org** *org-name*	Enters organization mode for the specified organization. To enter the root organization mode, type / as the *org-name*.
Step 3	UCSC(policy-mgr) /org # **delete power-control-policy** *policy-name*	Deletes the specified power control policy.
Step 4	UCSC(policy-mgr) /org # **commit-buffer**	Commits the transaction to the system configuration.

The following example shows how to delete a power control policy and commits the transaction:

```
UCSC# connect policy-mgr
UCSC(policy-mgr)# scope org /
UCSC(policy-mgr) /org # delete power-control-policy PCP-1
UCSC(policy-mgr) /org* # commit-buffer
UCSC(policy-mgr) /org #
```

Scrub Policy

This policy determines what happens to local data and to the BIOS settings on a server during the discovery process, when the server is reacknowledged, or when the server is disassociated from a service profile.

Note Local disk scrub policies only apply to hard drives that are managed by Cisco UCS Manager and do not apply to other devices such as USB drives.

Depending upon how you configure a scrub policy, the following can occur at those times:

Disk scrub

One of the following occurs to the data on any local drives on disassociation:

- If enabled, destroys all data on any local drives.
- If disabled, preserves all data on any local drives, including local storage configuration.

BIOS Settings Scrub

One of the following occurs to the BIOS settings when a service profile containing the scrub policy is disassociated from a server:

- If enabled, erases all BIOS settings for the server and resets them to the BIOS defaults for that server type and vendor.
- If disabled, preserves the existing BIOS settings on the server.

FlexFlash Scrub

FlexFlash Scrub enables you to pair new or degraded SD cards, resolve FlexFlash metadata configuration failures, and migrate older SD cards with 4 partitions to single partition SD cards. One of the following occurs to the SD card when a service profile containing the scrub policy is disassociated from a server, or when the server is reacknowledged:

- If enabled, the HV partition on the SD card is formatted using the PNUOS formatting utility. If two SD cards are present, the cards are RAID-1 paired, and the HV partitions in both cards are marked as valid. The card in slot 1 is marked as primary, and the card in slot 2 is marked as secondary.
- If disabled, preserves the existing SD card settings.

Note

- Because the FlexFlash scrub erases the HV partition on the SD cards, we recommend that you take a full backup of the SD card(s) using your preferred host operating system utilities before performing the FlexFlash Scrub.
- To resolve metadata config failures in a service profile, you need to disable FlexFlash in the local disk config policy before you run the FlexFlash scrub, then enable FlexFlash after the server is reacknowledged.
- Disable the scrub policy as soon as the pairing is complete or the metadata failures are resolved.

Creating a Scrub Policy

Procedure

	Command or Action	Purpose
Step 1	UCSC# **connect policy-mgr**	Enters policy manager mode.
Step 2	UCSC(policy-mgr)# **scope org** *org-name*	Enters organization mode for the specified organization. To enter the root organization mode, type **/** as the *org-name* .
Step 3	UCSC(policy-mgr) /org # **create scrub-policy** *policy-name*	Creates a scrub policy with the specified policy name, and enters organization scrub policy mode.
Step 4	UCSC(policy-mgr) /org/scrub-policy # **set descr** *description*	(Optional) Provides a description for the scrub policy. **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	UCSC(policy-mgr) /org/scrub-policy # **set disk-scrub** {**no** | **yes**}	Disables or enables disk scrubbing on servers using this scrub policy as follows: • If enabled, destroys all data on any local drives • If disabled, preserves all data on any local drives, including local storage configuration
Step 6	UCSC(policy-mgr) /org/scrub-policy # **set bios-settings-scrub** {**no** | **yes**}	Disables or enables BIOS settings scrubbing on servers using this scrub policy as follows: • If enabled, erases all BIOS settings for the server and resets them to the BIOS defaults for that server type and vendor • If disabled, preserves the existing BIOS settings on the server

	Command or Action	Purpose
Step 7	UCSC(policy-mgr) /org/scrub-policy # **commit-buffer**	Commits the transaction to the system configuration.

The following example creates a scrub policy named ScrubPolicy2, enables disk scrubbing on servers using the scrub policy, and commits the transaction:

```
UCSC# connect policy-mgr
UCSC(policy-mgr)# scope org /
UCSC(policy-mgr) /org # create scrub-policy ScrubPolicy2
UCSC(policy-mgr) /org/scrub-policy* # set descr "Scrub disk but not BIOS."
UCSC(policy-mgr) /org/scrub-policy* # set disk-scrub yes
UCSC(policy-mgr) /org/scrub-policy* # set bios-settings-scrub no
UCSC(policy-mgr) /org/scrub-policy* # commit-buffer
UCSC(policy-mgr) /org/scrub-policy #
```

Deleting a Scrub Policy

Procedure

	Command or Action	Purpose
Step 1	UCSC# **connect policy-mgr**	Enters policy manager mode.
Step 2	UCSC(policy-mgr)# **scope org** *org-name*	Enters organization mode for the specified organization. To enter the root organization mode, type / as the *org-name*.
Step 3	UCSC(policy-mgr) /org # **delete scrub-policy** *policy-name*	Deletes the specified scrub policy.
Step 4	UCSC(policy-mgr) /org # **commit-buffer**	Commits the transaction to the system configuration.

The following example deletes the scrub policy named ScrubPolicy2 and commits the transaction:

```
UCSC# connect policy-mgr
UCSC(policy-mgr)# scope org /
UCSC(policy-mgr) /org # delete scrub-policy ScrubPolicy2
UCSC(policy-mgr) /org* # commit-buffer
UCSC(policy-mgr) /org #
```

Serial over LAN Policy

This policy sets the configuration for the serial over LAN connection for all servers associated with service profiles that use the policy. By default, the serial over LAN connection is disabled.

If you implement a serial over LAN policy, we recommend that you also create an IPMI profile.

You must include this policy in a service profile and that service profile must be associated with a server for it to take effect.

Configuring a Serial over LAN Policy

Procedure

	Command or Action	Purpose
Step 1	UCSC# **connect policy-mgr**	Enters policy manager mode.
Step 2	UCSC(policy-mgr) # **scope org** *org-name*	Enters organization mode for the specified organization. To enter the root organization mode, type / as the *org-name*.
Step 3	UCSC(policy-mgr) /org # **create sol-policy** *policy-name*	Creates a serial over LAN policy and enters organization serial over LAN policy mode.
Step 4	UCSC(policy-mgr) /org/sol-policy # **set descr** *description*	(Optional) Provides a description for the policy. **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	UCSC(policy-mgr) /org/sol-policy # **set speed** {**115200** | **19200** | **38400** | **57600** | **9600**}	Specifies the serial baud rate.
Step 6	UCSC(policy-mgr) /org/sol-policy # {**disable** | **enable**}	Disables or enables the serial over LAN policy. By default, the serial over LAN policy is disabled; you must enable it before it can be applied.
Step 7	UCSC(policy-mgr) /org/sol-policy # **commit-buffer**	Commits the transaction to the system configuration.

The following example creates a serial over LAN policy named Sol9600, provides a description for the policy, sets the speed to 9,600 baud, enables the policy, and commits the transaction:

```
UCSC# connect policy-mgr
UCSC(policy-mgr)# scope org /
UCSC(policy-mgr) /org # create sol-policy Sol9600
UCSC(policy-mgr) /org/sol-policy* # set descr "Sets serial over LAN policy to 9600 baud."
UCSC(policy-mgr) /org/sol-policy* # set speed 9600
UCSC(policy-mgr) /org/sol-policy* # enable
UCSC(policy-mgr) /org/sol-policy* # commit-buffer
UCSC(policy-mgr) /org/sol-policy #
```

Viewing a Serial over LAN Policy

Procedure

	Command or Action	Purpose
Step 1	UCSC# **connect policy-mgr**	Enters policy manager mode.
Step 2	UCSC(policy-mgr)# **scope org** *org-name*	Enters organization mode for the specified organization. To enter the root organization mode, type / as the *org-name* .
Step 3	UCSC(policy-mgr) /org # **show sol-policy** *policy-name*	Displays the serial over LAN definition (set by the create sol-config command). If the serial over LAN definition is not set, and if a policy is set (using the set sol-policy command), then the policy will be displayed.

The following example shows how to display serial over LAN information for a serial over LAN policy called Sol9600:

```
UCSC# connect policy-mgr
UCSC(policy-mgr)# scope org /
UCSC(policy-mgr) /org # show sol-policy Sol9600

SOL Policy:
Full Name: Sol9600
SOL State: Enable
Speed: 9600
Description:
```

Server Pool Policy Qualifications

This policy qualifies servers based on the inventory of a server conducted during the discovery process. The qualifications are individual rules that you configure in the policy to determine whether a server meets the selection criteria. For example, you can create a rule that specifies the minimum memory capacity for servers in a data center pool.

Qualifications are used in other policies to place servers, not just by the server pool policies. For example, if a server meets the criteria in a qualification policy, it can be added to one or more server pools or have a service profile automatically associated with it.

You can use the server pool policy qualifications to qualify servers according to the following criteria:

- Adapter type
- Chassis location
- Memory type and configuration
- Power group
- CPU cores, type, and configuration
- Storage configuration and capacity

- Server model

Depending upon the implementation, you might need to configure several policies with server pool policy qualifications including the following:

- Autoconfiguration policy
- Chassis discovery policy
- Server discovery policy
- Server inheritance policy
- Server pool policy

Creating a Server Pool Qualification Policy

Procedure

	Command or Action	Purpose
Step 1	UCSC# **connect policy-mgr**	Enters policy manager mode.
Step 2	UCSC(policy-mgr) # **scope org** *org-name*	Enters organization mode for the specified organization. To enter the root organization mode, type / as the *org-name*.
Step 3	UCSC(policy-mgr) /org # **create server-qual** *server-qual-name*	Creates a server pool qualification with the specified name, and enters organization server qualification mode.
Step 4	UCSC(policy-mgr) /org/server-qual # **commit-buffer**	Commits the transaction to the system configuration.

The following example shows how to create a server pool qualification named ServPoolQual22 and commit the transaction:

```
UCSC# connect policy-mgr
UCSC(policy-mgr)# scope org /
UCSC(policy-mgr) /org* # create server-qual ServPoolQual22
UCSC(policy-mgr) /org/server-qual* # commit-buffer
UCSC(policy-mgr) /org/server-qual #
```

Creating a Domain Qualification for a Policy Qualification

Procedure

	Command or Action	Purpose
Step 1	UCSC# **connect policy-mgr**	Enters policy manager mode.

	Command or Action	Purpose
Step 2	UCSC(policy-mgr) # **scope org** *org-name*	Enters organization mode for the specified organization. To enter the root organization mode, type / as the *org-name*.
Step 3	UCSC(policy-mgr) /org # **scope server-qual** *server-qual-name*	Enters server qualification mode for the specified server pool policy qualification.
Step 4	UCSC(policy-mgr) /org/server-qual # **create domain-qual** *domain-qual-name*	Creates the specified domain qualification and enters domain qualification mode.
Step 5	UCSC(policy-mgr) /org/server-qual/domain-qual # **commit-buffer**	Commits the transaction to the system configuration.

The following example shows how to add a domain qualification to a server pool policy qualification and commit the transaction:

```
UCSC# connect policy-mgr
UCSC(policy-mgr)# scope org /
UCSC(policy-mgr) /org # scope server-qual ServPoolQual22
UCSC(policy-mgr) /org/server-qual # create domain-qual TestDomain
UCSC(policy-mgr) /org/server-qual/domain-qual* # commit-buffer
UCSC(policy-mgr) /org/server-qual/domain-qual #
```

Creating an Adapter Qualification for a Policy Qualification

Procedure

	Command or Action	Purpose
Step 1	UCSC# **connect policy-mgr**	Enters policy manager mode.
Step 2	UCSC(policy-mgr) # **scope org** *org-name*	Enters organization mode for the specified organization. To enter the root organization mode, type / as the *org-name*.
Step 3	UCSC(policy-mgr) /org # **scope server-qual** *server-qual-name*	Enters server qualification mode for the specified server pool policy qualification.
Step 4	UCSC(policy-mgr) /org/server-qual # **create adapter**	Creates the specified adapter qualification and enters adapter qualification mode.
Step 5	UCSC(policy-mgr) /org/server-qual/adapter # **create cap-qual** *adapter-type*	Creates an adapter capacity qualification for the specified adapter type and enters organization server qualification adapter capacity qualification mode. The *adapter-type* argument can be any of the following values: • **fcoe —**Fibre Channel over Ethernet • **non-virtualized-eth-if —**Non-virtualized Ethernet interface

	Command or Action	Purpose
		• **non-virtualized-fc-if —**Non-virtualized Fibre Channel interface • **path-encap-consolidated —**Path encapsulation consolidated • **path-encap-virtual —**Path encapsulation virtual • **protected-eth-if —**Protected Ethernet interface • **protected-fc-if —**Protected Fibre Channel interface • **protected-fcoe —**Protected Fibre Channel over Ethernet • **uplink-aggregation —**Uplink Aggregation • **virtualized-eth-if —**Virtualized Ethernet interface • **virtualized-eth-sriov —**Virtualized Ethernet SRIOV • **virtualized-fc-if —**Virtualized Fibre Channel interface • **virtualized-fc-sriov —**Virtualized Fibre Channel SRIOV • **virtualized-scsi-if —**Virtualized SCSI interface
Step 6	UCSC(policy-mgr) /org/server-qual/adapter/cap-qual # **set maximum** {*max-cap* | **unspecified**}	Specifies the maximum capacity for the selected adapter type.
Step 7	UCSC(policy-mgr) /org/server-qual/adapter/cap-qual # **set pid-regex** *regex*	Specifies the regular expression that the PID must match.
Step 8	UCSC(policy-mgr) /org/server-qual/adapter/cap-qual # **commit-buffer**	Commits the transaction to the system configuration.

The following example shows how to add a domain qualification to a server pool policy qualification and commit the transaction:

```
UCSC# connect policy-mgr
UCSC(policy-mgr)# scope org /
UCSC(policy-mgr) /org # scope server-qual ServPoolQual22
UCSC(policy-mgr) /org/server-qual # create adapter TestAdapter
UCSC(policy-mgr) /org/server-qual/adapter* # create cap-qual non-virtualized-eth-if
UCSC(policy-mgr) /org/server-qual/adapter/cap-qual* # set maximum unspecified
UCSC(policy-mgr) /org/server-qual/adapter/cap-qual* # commit-buffer
UCSC(policy-mgr) /org/server-qual/adapter/cap-qual #
```

Deleting a Server Pool Policy Qualification

Procedure

	Command or Action	Purpose
Step 1	UCSC# **connect policy-mgr**	Enters policy manager mode.
Step 2	UCSC(policy-mgr)# **scope org** *org-name*	Enters organization mode for the specified organization. To enter the root organization mode, type / as the *org-name* .
Step 3	UCSC(policy-mgr) /org # **delete server-qual** *server-qual-name*	Deletes the specified server pool qualification.
Step 4	UCSC(policy-mgr) /org/server-qual # **commit-buffer**	Commits the transaction to the system configuration.

The following example deletes the server pool qualification named ServPoolQual22 and commits the transaction:

```
UCSC# connect policy-mgr
UCSC(policy-mgr)# scope org /
UCSC(policy-mgr) /org* # delete server-qual ServPoolQual22
UCSC(policy-mgr) /org* # commit-buffer
UCSC(policy-mgr) /org #
```

vNIC/vHBA Placement Policies

vNIC/vHBA placement policies are used to determine the following:

- How the virtual network interface connections (vCons) are mapped to the physical adapters on a server.
- What types of vNICs or vHBAs can be assigned to each vCon.

Each vNIC/vHBA placement policy contains four vCons that are virtual representations of the physical adapters. When a vNIC/vHBA placement policy is assigned to a service profile, and the service profile is associated with a server, the vCons in the vNIC/vHBA placement policy are assigned to the physical adapters and the vNICs and vHBAs are assigned to those vCons.

For blade or rack servers that contain one adapter, Cisco UCS assigns all vCons to that adapter. For servers that contain four adapters, Cisco UCS assigns vCon1 to Adapter1, vCon2 to Adapter2, vCon3 to Adapter3, and vCon4 to Adapter4.

For blade or rack servers that contain two or three adapters, Cisco UCS assigns the vCons based on the type of server and the selected virtual slot mapping scheme, which can be **Round Robin** or **Linear Ordered**. For details about the available mapping schemes, see vCon to Adapter Placement.

After Cisco UCS assigns the vCons, it assigns the vNICs and vHBAs based on the **Selection Preference** for each vCon. This can be one of the following:

- —All configured vNICs and vHBAs can be assigned to the vCon, whether they are explicitly assigned to it, unassigned, or dynamic. This is the default.

- —vNICs and vHBAs must be explicitly assigned to the vCon. You can assign them explicitly through the service profile or the properties of the vNIC or vHBA.
- —Dynamic vNICs and vHBAs cannot be assigned to the vCon. The vCon can be used for all static vNICs and vHBAs, whether they are unassigned or explicitly assigned to it.
- —Unassigned vNICs and vHBAs cannot be assigned to the vCon. The vCon can be used for dynamic vNICs and vHBAs and for static vNICs and vHBAs that are explicitly assigned to it.
- —Cisco usNICs cannot be assigned to the vCon. The vCon can be used for all other configured vNICs and vHBAs, whether they are explicitly assigned to it, unassigned, or dynamic.

Note An SRIOV usNIC that is explicitly assigned to a vCon set to will remain assigned to that vCon.

If you do not include a vNIC/vHBA placement policy in the service profile, Cisco UCS Central defaults to the **Round Robin** vCon mapping scheme and the **All** vNIC/vHBA selection preference, distributing the vNICs and vHBAs between the adapters based on the capabilities and relative capacities of each adapter.

Configuring a vNIC/vHBA Placement Policy

Procedure

	Command or Action	Purpose
Step 1	UCSC# **connect policy-mgr**	Enters policy manager mode.
Step 2	UCSC(policy-mgr)# **scope org** *org-name*	Enters organization mode for the specified organization. To enter the root organization mode, type / as the *org-name*.
Step 3	UCSC(policy-mgr) /org # **create vcon-policy** *policy-name*	Creates the specified vNIC/vHBA placement profile and enters organization vcon policy mode.
Step 4	UCSC(policy-mgr) /org/vcon-policy # **set descr** *description*	(Optional) Provides a description for the vNIC/vHBA Placement Profile. Enter up to 256 characters. You can use any characters or spaces except ` (accent mark), \ (backslash), ^ (carat), " (double quote), = (equal sign), > (greater than), < (less than), or ' (single quote). **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	UCSC(policy-mgr) /org/vcon-policy # **set mapping-scheme** {**round-robin** | **linear-ordered**}	(Optional) For blade or rack servers that contain one adapter, Cisco UCS Central assign all vCons to that adapter. For servers that contain four adapters, Cisco UCS Central assign vCon1 to Adapter1, vCon2 to Adapter2, vCon3 to Adapter3, and vCon4 to Adapter4.

	Command or Action	Purpose
		For blade or rack servers that contain two or three adapters, Cisco UCS Central assigns vCons based on the selected virtual slot mapping scheme. This can be one of the following: • **Round Robin round-robin—** In a server with two adapter cards, Cisco UCS Central assigns vCon1 and vCon3 to Adapter1, then assigns vCon2 and vCon4 to Adapter2. In a server with three adapter cards, Cisco UCS Central assigns vCon1 to Adapter1, vCon2 and vCon4 to Adapter2, and vCon3 to Adapter3. This is the default scheme. • **Linear Ordered Linear-ordered—** In a server with two adapter cards, Cisco UCS Central assigns vCon1 and vCon2 to Adapter1, then assigns vCon3 and vCon4 to Adapter2. In a server with three adapter cards, Cisco UCS Central assigns vCon1 to Adapter1 and vCon2 to Adapter2, then assigns vCon3 and vCon4 to Adapter3. In N20-B6620 and N20-B6625-2 blade servers, the two adapters are numbered left to right while vCons are numbered right to left. If one of these blade servers has a single adapter, Cisco UCS Central assigns all vCons to that adapter. If the server has two adapters, the vCon assignment depends upon the virtual slot mapping scheme: • **Round Robin round-robin—**Cisco UCS Central assigns vCon2 and vCon4 to Adapter1 and vCon1 and vCon3 to Adapter2. This is the default. • **Linear Ordered linear-ordered—**Cisco UCS Central assigns vCon3 and vCon4 to Adapter1 and vCon1 and vCon2 to Adapter2.
Step 6	UCSC(policy-mgr) /org/vcon-policy # **set vcon** {**1** | **2** | **3** | **4**} **selection** {**all** | **assigned-only** | **exclude-dynamic** | **exclude-unassigned**}	Specifies the selection preference for the specified vCon. The options are: • **All all—**All configured vNICs and vHBAs can be assigned to the vCon, whether they are explicitly assigned to it, unassigned, or dynamic. This is the default. • **Assigned Only assigned-only—**vNICs and vHBAs must be explicitly assigned to the vCon. You can assign them explicitly through the service profile or the properties of the vNIC or vHBA. • **Exclude Dynamic exclude-dynamic —**Dynamic vNICs and vHBAs cannot be assigned to the vCon. The vCon can be used for all static vNICs and vHBAs, whether they are unassigned or explicitly assigned to it. • **Exclude Unassigned exclude-unassigned—**Unassigned vNICs and vHBAs cannot be assigned to the vCon. The vCon can be used

	Command or Action	Purpose
		for dynamic vNICs and vHBAs and for static vNICs and vHBAs that are explicitly assigned to it.
Step 7	UCSC(policy-mgr) /org/vcon-policy # **commit-buffer**	Commits the transaction.

The following example creates a vNIC/vHBA placement policy named Adapter1All, sets the vCon mapping scheme to Linear Ordered, specifies that only assigned vNICs and vHBAs can be placed on adapter 1, and commits the transaction:

```
UCSC# connect policy-mgr
UCSC(policy-mgr)# scope org /
UCSC(policy-mgr) /org # create vcon-policy Adapter1
UCSC(policy-mgr) /org/vcon-policy* # set descr "This profile places all vNICs and vHBAs on
 adapter 1."
UCSC(policy-mgr) /org/vcon-policy* # set mapping-scheme linear-ordered
UCSC(policy-mgr) /org/vcon-policy* # set vcon 1 selection assigned-only
UCSC(policy-mgr) /org/vcon-policy* # commit-buffer
UCSC(policy-mgr) /org/vcon-policy* #
UCSC(policy-mgr) /org #
```

Deleting a vNIC/vHBA Placement Policy

Procedure

	Command or Action	Purpose
Step 1	UCSC# **connect policy-mgr**	Enters policy manager mode.
Step 2	UCSC(policy-mgr)# **scope org** *org-name*	Enters organization mode for the specified organization. To enter the root organization mode, type / as the *org-name* .
Step 3	UCSC(policy-mgr) /org # **delete vcon-policy** *policy-name*	Deletes the specified vNIC/vHBA placement profile.
Step 4	UCSC(policy-mgr) /org # **commit-buffer**	Commits the transaction.

The following example deletes the vNIC/vHBA placement profile named Adapter1All and commits the transaction:

```
UCSC# connect policy-mgr
UCSC(policy-mgr)scope org /
UCSC(policy-mgr) /org # delete vcon-policy Adapter1All
UCSC(policy-mgr) /org* # commit-buffer
UCSC(policy-mgr) /org #
```