Cisco UCS Manager Configuration Common Practices and Quick-Start Guide

Contents

Introduction to Cisco Unified Computing System

The introduction of the Cisco Unified Computing System™ (Cisco UCS™) in June 2009 presented a new model for data center and server management. As of August 2013, Cisco UCS is the #2 x86 blade server, worldwide,[1] and is used by more than 26,000 unique customers. While the model is no longer new, a growing number of customers are deploying Cisco UCS for the first time. This guide provides an overview of Cisco UCS essentials and best practices. This guide also covers the most direct path to working in a stateless-server SAN boot environment, upon which much of the Cisco UCS core value is based, with respect to service availability. In support of a utility or cloud computing model, this guide also presents a number of concepts and elements within the Cisco UCS Management Model that should help data center operators increase responsiveness and efficiency by improving data center automation.

A highly distilled summary can be found in the "Cisco UCS Quick-Start Guide" at the end of this document.

Cisco UCS Design Goals

Cisco UCS was architected with the following design goals:

1. To provide greater administrative and operational control, using fewer individual points of management, thereby allowing increased scalability while reducing complexity
2. To greatly reduce the time needed to commision new compute resources, and the time needed to deploy new servers (either bare-metal OS or hypervisor-based)
3. To improve service availability through the deepest possible abstraction of hardware state and I/O connectivity
4. To simplify server virtualization through converged physical infrastructure
5. To enable all the physical infrastructure and connectivity to be fully programmable, thereby reducing manual intervention, and enabling automation

In Cisco UCS, the notion of "server" and "chassis" take on meanings that are different from the traditional model. There is less focus on the "physical server" (blade or rack) and more focus on the "logical server", which is called a "Service Profile". The Service Profile contains all the server hardware identifiers, firmware, state, configuration, connectivity and behaviour, but is totally abstracted from the physical server. "Chassis" provide power and cooling, but no longer manage and control the servers within the enclosure.

Cisco UCS revolves around the Cisco UCS Manager, which acts as a server-domain controller—much in the same way that traditional blade-chassis are server-domain controllers for the 16 or so blades within. Cisco UCS provides a "virtual chassis" that can encompass up to 20 physical chassis.[2] The traditional configuration and management points are moved out from the individual chassis and promoted up to the datacenter access tier where the Cisco UCS Manager (UCSM) runs as an embedded device manager within a clustered pair of Cisco Fabric Interconnects (FI's). An instance of the UCS Manager (UCSM) is what defines a Cisco UCS domain. A domain can dynamically grow to 20 chassis without increasing points of management, since the chassis have no state nor configuration nor chassis-centric management points. All management and configuration is done at the domain level, not the chassis level.

[1] *Source: IDC Worldwide Quarterly Server Tracker, Q1 2013 Revenue Share, May 2013

[2] A UCS domain can grow to 160 servers, being a combination of blade or rack-mount

“Service Profiles” form the foundation for the stateless, utility computing model in Cisco UCS. While virtualization of MAC addresses and World Wide Port Name (WWPN) identifiers has been evolving through the industry for years, UCS goes ever further by extending the logical service profile definition to include the hardware, BIOS, CPU, and I/O adapter configuration, versions and settings.

Among the architectural goals for which Cisco UCS was not designed are:

- A server-provisioning framework
- A service orchestration or workflow engine
- A manager of upstream SAN and LAN devices
- A manager of virtual machines
- Visibility into adjacent UCS domain peers[3]

Cisco UCS Manager Initial Configuration

Bringing up Cisco UCS for the first time is done through the serial console. A first time wizard will guide the user through the standard questions (hostname, IP address, netmask, default gateway, etc).

These standard practices should be followed:

- Ensure the FI’s “L1” and “L2” ports are connected to form the cluster and that the chassis are cabled up to the FI’s before applying power for the first time.
- Allocate three IP addresses in the management and administrative subnet: one for each Fabric Interconnect, and one for the virtual IP interface that defines the Cisco UCS Manager instance and enables management, regardless of which FI is acting as the primary[4] device.
- The “-A” and “-B” strings are implicitly added to the end of hostname and should not be specified explicitly.

Most of the configuration is done on the “A” side. The configuration of the “B” side Fabric Interconnect is inherited from the “A” side, once a “cluster configuration” is detected.

After running the first time setup wizard, everything can be easily managed from the UCSM GUI by pointing a browser to the UCS management IP address.

The following sections describe the steps that must be performed prior to general operations.

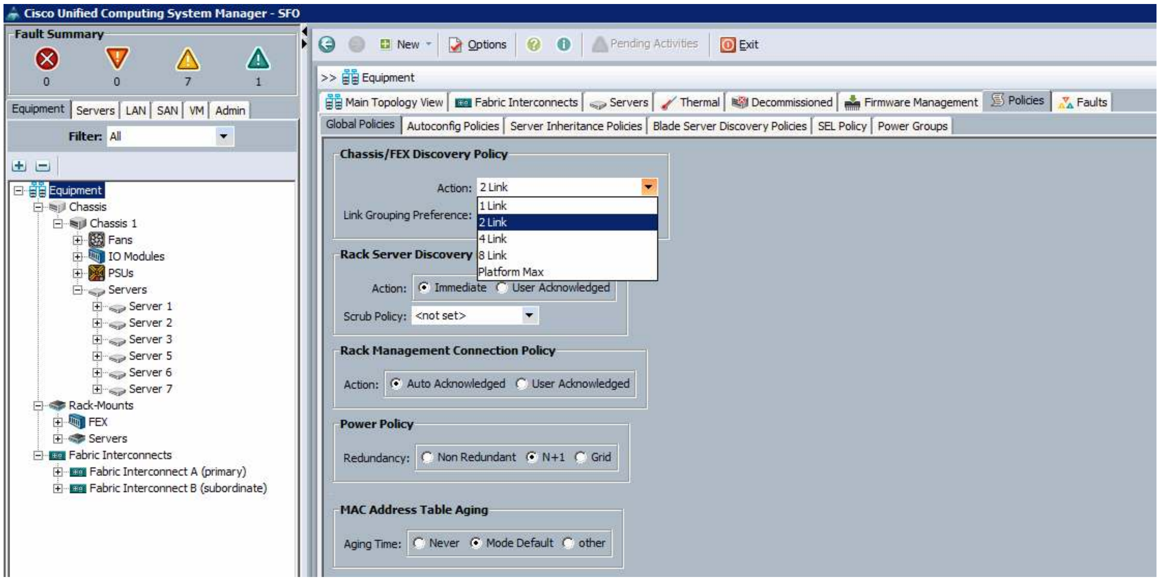
Setting the Equipment Policies

The “**Chassis Discovery Policy**” specifies the minimum number of connections between the I/O modules (IOM’s) and the FI’s. This value must be set explicitly. To set this value from the Equipment tab, select Equipment and then choose Policies > Global Policies and set the policy as shown in Figure 1.

[3] Visibility and configuration across multiple domains is achieved through Cisco UCS Central

[4] The Cisco UCS Manager Control Plane runs as “primary” and “subordinate”; the Data Plane is always active/active across both FI’s.

Figure 1. Chassis Discovery Policy



Chassis/FEX Discovery Policy

Action: 2 Link

1 Link
2 Link
4 Link
8 Link
Platform Max

Link Grouping Preference:

Rack Server Discovery

Action: ● Immediate ○ User Acknowledged

Scrub Policy: <not set>

Rack Management Connection Policy

Action: ● Auto Acknowledged ○ User Acknowledged

Power Policy

Redundancy: ○ Non Redundant ● N+1 ○ Grid

MAC Address Table Aging

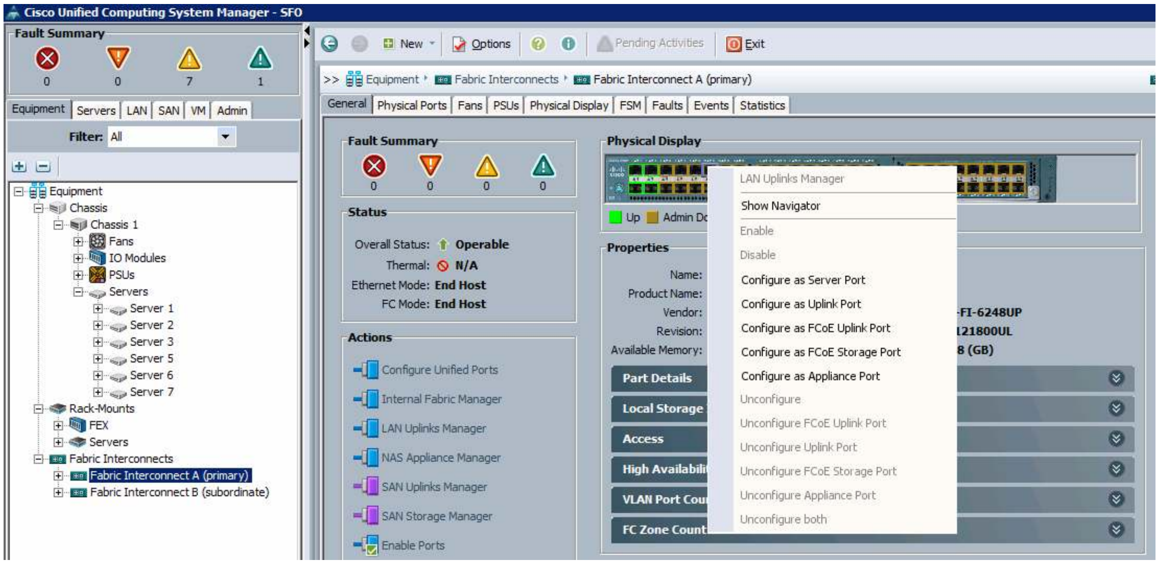
Aging Time: ○ Never ● Mode Default ○ other

The “**Link Grouping Preference**” determines whether all of the links from the IOM to the fabric interconnect are grouped into a fabric port channel during chassis discovery. If the link grouping preference is set to port channel, all of the links from the IOM to the fabric interconnect are grouped in a fabric port channel. The best practice is to set Link Grouping Preference to “Port Channel”, to which provides greater availability in the event of link failure.

Enable Server and Uplink Ports

The FI’s are essentially network access devices. They are oriented with southbound connections going to the chassis IOMs and with northbound connections going to the core LAN or SAN. Ports going south to the chassis IOMs are configured as “Server Ports” and ports going north to the core LAN are configured as “Uplink Ports”. Ports can be configured and enabled by right-clicking the desired ports when the FI is in scope, as shown in Figure 2.

Figure 2. Configuring and Enabling Fabric Interconnect Ports



Status

Overall Status: Operable
Thermal: N/A
Ethernet Mode: End Host
FC Mode: End Host

Properties

Name:
Product Name:
Vendor: -FI-6248UP
Revision: 121800UL
Available Memory: 8 (GB)

One of the main solution benefits is the reduction in server provisioning time. Whenever additional chassis are racked, stacked, and cabled and their server ports are configured, the Cisco UCS Manager will automatically perform an inventory and deep discovery of any subsequently attached equipment, without requiring manual intervention. Regardless of how many new chassis are connected, the discovery process would take about 10 minutes to discover all chassis in parallel and bring their physical resources in to the information tree under the Equipment tab.

Create Management IP Address Pool

Cisco UCS provides out-of-band access (for remote keyboard, video, and mouse [KVM] and remote CD, DVD, and USB access) for every blade. This access is made possible by associating a pool of IP addresses for the cut-through interfaces that correspond with each blade's Cisco® Integrated Management Controller (CIMC). Typically, these addresses are configured on the same administrative subnet as the UCS Manager IP address. This pool is created from the LAN tab under Pools, by selecting "IP Pool ext-mgmt" and creating a block of addresses. The binding of these addresses to the blades happens automatically, with no manual intervention required.

Create Host Firmware Package (Best Practice)

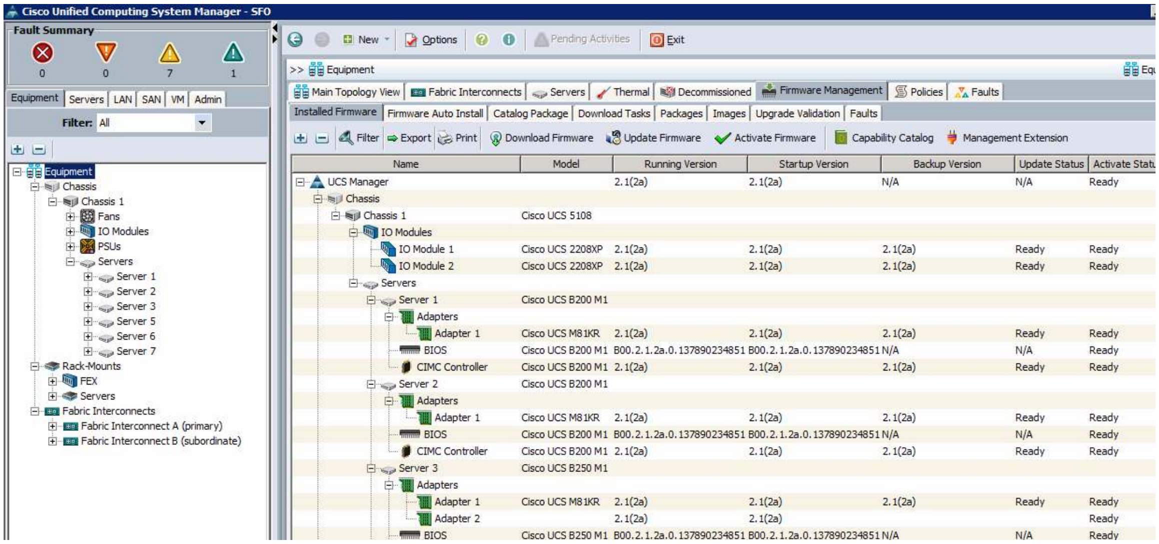
With Cisco UCS, the lowest-level BIOS and adapter firmware can be grouped into "Host Firmware Packages", which can then be associated with the logical server (Service Profiles). This model represents a major shift from traditional server models: the version of the BIOS and adapter firmware become properties that are prescribed and associated as part of the logical server (Service Profile) definition. This model stands in contrast with that of other so-called "stateless server" models that have not abstracted to this level of detail, and have no control over the physical server's BIOS and adapter firmware versions. Furthermore, the Host Firmware Package is not architecturally related to the UCSM firmware version and does not need to be synchronized with that version.[5]

In the initial configuration and setup (or for any UCSM firmware upgrade), a best practice is to create a corresponding Host Firmware Package with the corresponding UCSM version.

[5] Unless directed so by the product release notes.

Cisco UCS Manager offers the capability to quickly and easily report on the firmware versions for all major hardware components. From the Equipment tab, select Equipment and choose Firmware Management > Installed Firmware, as shown in Figure 3. In general, the IOMs, FIs, and UCS Manager should all have the same version; the version on the adapter cards, BIOS, and CIMC will generally be dictated by the Host Firmware Package from the associated Service Profile.

Figure 3. Firmware Reporting for all Major Hardware Components



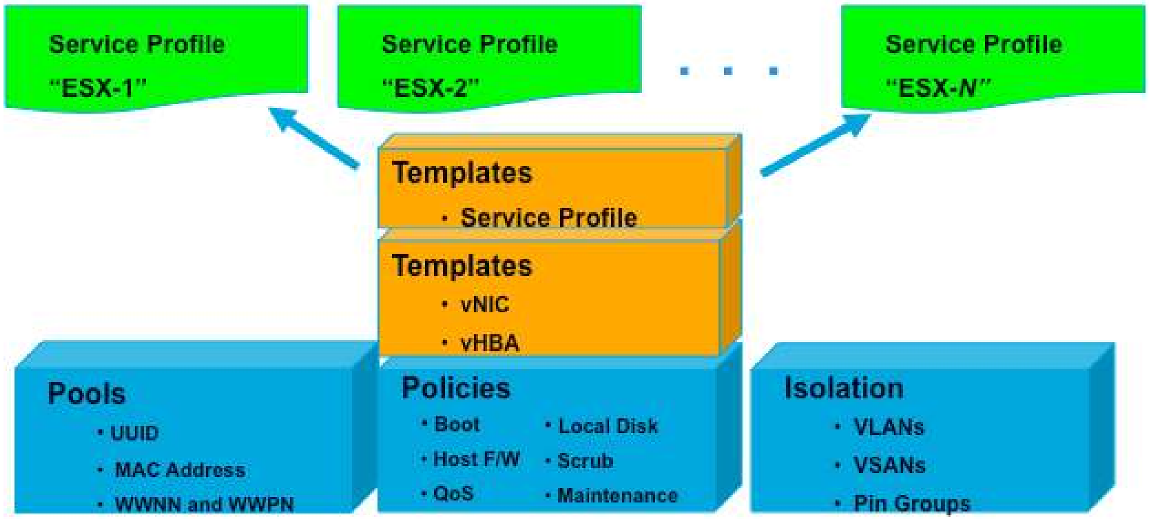
Name	Model	Running Version	Startup Version	Backup Version	Update Status	Activate Statu
UCS Manager		2.1(2a)	2.1(2a)	N/A	N/A	Ready
Chassis						
Chassis 1	Cisco UCS 5108					
IO Modules						
IO Module 1	Cisco UCS 2208XP	2.1(2a)	2.1(2a)	2.1(2a)	Ready	Ready
IO Module 2	Cisco UCS 2208XP	2.1(2a)	2.1(2a)	2.1(2a)	Ready	Ready
Servers						
Server 1	Cisco UCS B200 M1					
Adapters						
Adapter 1	Cisco UCS M81KR	2.1(2a)	2.1(2a)	2.1(2a)	Ready	Ready
BIOS	Cisco UCS B200 M1	B00.2.1.2a.0.137890234851	B00.2.1.2a.0.137890234851	N/A	N/A	Ready
CIMC Controller	Cisco UCS B200 M1	2.1(2a)	2.1(2a)	2.1(2a)	Ready	Ready
Server 2	Cisco UCS B200 M1					
Adapters						
Adapter 1	Cisco UCS M81KR	2.1(2a)	2.1(2a)	2.1(2a)	Ready	Ready
BIOS	Cisco UCS B200 M1	B00.2.1.2a.0.137890234851	B00.2.1.2a.0.137890234851	N/A	N/A	Ready
CIMC Controller	Cisco UCS B200 M1	2.1(2a)	2.1(2a)	2.1(2a)	Ready	Ready
Server 3	Cisco UCS B250 M1					
Adapters						
Adapter 1	Cisco UCS M81KR	2.1(2a)	2.1(2a)	2.1(2a)	Ready	Ready
Adapter 2		2.1(2a)	2.1(2a)			Ready
BIOS	Cisco UCS B250 M1	B00.2.1.2a.0.137890234851	B00.2.1.2a.0.137890234851	N/A	N/A	Ready

At this point, all the hardware should be available and ready to use as physical equipment. The real power of Cisco UCS, though, is in the way it creates and configures application-level servers (Service Profiles) and has them “associate” on to the physical hardware.

Improved Availability with Stateless Servers

Cisco UCS provides the infrastructure to support the highest levels of service availability through a stateless server environment. Stateless servers are logical servers (OS and application images) that can run on any physical server or blade in a way that encapsulates traditional hardware identifiers (World Wide Node Names [WWNNs],World Wide Port Names [WWPNs], MAC addresses, etc.), as integral parts of the Service Profile identity (Figure 4).

Figure 4. Cisco UCS Logical Building Blocks



The foundations of Service Profiles are these logical building blocks (for example, pools and policies) that can then be captured for reuse. Furthermore, virtual network interface card (vNIC) and virtual host bus adapter (vHBA) templates can be referenced for use in higher-level Service-Profile templates. To reduce server provisioning time, Service Profile templates can then be used to rapidly instantiate actual Service Profiles (possibly automatically associating those instantiated service profiles with actual physical servers and blades).

The relationships between these internal building blocks are all rationalized and normalized. Higher-level objects (such as vNICs) can be created by referencing lower-level objects (such as pools and policies). This normalized data model promotes reuse and removes the need for duplication of common objects.

Pools

Pools are the base building blocks for uniquely identifying hardware resources. As the basis for the UCS management model, they allow Service Profiles to be associated with any blade, while still providing the exact same ID and presentation to the upstream LAN or SAN. There are three sets of pools used part of best practices:

- WWNN and WWPN pools: Provide unique IDs for Fibre Channel resources on a server (Fibre Channel nodes and ports)
- MAC address pools: Provide unique IDs for network interface ports
- UUID pools: Provide IDs similar to a serial number or service tag

In the GUI, these pools are all functionally organized, with UUID pools maintained from the Server tab, WWNN and WWPN pools maintained from the SAN tab, and MAC address pools maintained from the LAN tab.

Best Practice: ID Conventions

Cisco UCS Management domains can coexist along with many other UCSM domains and with servers other than Cisco UCS servers, all of which can have their own sets of unique hardware identifiers and pools. The presentation of duplicate WWNs and MAC addresses to the LAN or SAN could naturally be a major source of complications.

To avoid these issues, adopt an enumeration scheme for domains, such that domain ID's are embedded in the high-order byte range of all pools, including MAC, WWNN, WWPN and UUID. Best practices are to embed either a simple domain ID, or a site/domain pair, along with a fabric side indicator to guarantee uniqueness and identify fabric source. For example, a MAC pool block would take the form **00:25:B5:23:BX:YY**, where **00:25:B5** designates Cisco UCS, **23** indicates site **2**, domain **3**, and **B** indicates the B-side fabric. Smaller environments could shorten the encoding to just domain and fabric side, as in **00:25:B5:1A:XX:YY**.

As organizations grow their UCS infrastructure, the use of UCS Central[6] is strongly recommended for managing global ID pools.

Best Practice: Pools

Define and use Pools as a standard practice. Make sure that:

- UUID pools are referenced when you create Service Profiles
- MAC address pools are referenced when you create vNICs
- WWNN pools are referenced when you create Service Profiles
- WWPN pools are referenced when you create vHBAs

Similarly, Pools should also be referenced when you create any corresponding template objects (vNICs, vHBAs, and Service Profiles).

Trade-offs exist when considering pool management. For many environments, populating and using the respective default pools may be simplest and sufficient, or by creating domain-wide pools, such as "MAC-A", or "WWPN-B". This approach reduces the number of objects that need to be configured and managed. Alternatively, operators are free to configure different pools on a per-tenant or per-application basis. This approach can provide more specific identity management and more granular traffic monitoring of tenants, applications, etc.

Policies

Policies are a primary mechanism for enforcing rules, which helps ensure consistency and repeatability. Defining and using a comprehensive set of policies enables greater consistency, control, predictability and automation. The most common policies that should be used regularly are presented here.

Boot Policy

Boot Policy determines how a server boots, specifying the boot devices, the method, and the boot order.

Traditional use of SAN boot requires manual configuration for each server performing SAN boot. Typically, having 100 servers SAN-boot would require configuring 100 servers manually and individually. Cisco UCS inverts this unwieldy model, and instead requires configuring only in proportion to the number of storage arrays serving SAN-boot images, regardless of the number of servers doing SAN-boot. A single boot policy, with the WWPNs of a single storage array can be referenced and reused by any number of servers, without additional manual configuration.

Much of the Cisco UCS core value around availability is predicated on SAN boot. Therefore, the use of SAN boot within a Boot policy is a most highly recommended best practice to improve service availability.

[6] http://www.cisco.com/en/US/products/ps12502/index.html

Suggested Practices: Boot Policy

- Have a CD-ROM as the first in the boot order, for emergencies and for booting in recovery mode.
- For SAN boot, define separate boot policies for each storage array that would be serving boot LUNs.
- For network boot, define the vNIC device as last in the boot order, following either SAN or local boot. This allows for a network boot and installation, only if the OS had not previously been installed.

Host Firmware Policy

Use Host Firmware Policy to associate qualified or well-known versions of the BIOS, adapter ROM, or local disk controller with logical Service Profiles, as described earlier. A best practice is to create one policy, based on the latest packages that correspond with the Cisco UCS Manager infrastructure and server software release, and to reference that Host Firmware Package for all Service Profiles and templates created. This best practice will help ensure version consistency of a server's lowest-level firmware, regardless of physical server failures that may cause re-association of Service Profiles on other blades.

Maintenance Policy

Use the Maintenance Policy to specify how Cisco UCS Manager should proceed for configuration changes that will have a service impact or require a server reboot. Values for the Maintenance Policy can be “immediate”, “user-ack”, or “timer automatic”. The best practice is to not use the “default” policy, and instead to create and use Maintenance Policies for either “user-ack” or “timer automatic”, and to always have these as elements of the Service Profile or Service Profile Template definition.

Local Disk Policy

Local disk policy specifies how to configure any local disks on the blade. A best practice is to specify no local storage for SAN boot environments, thereby precluding any problems at Service Profile association time, when local disks may present themselves the host OS during installation. For additional assurance, you can remove or unseat local disks from blades completely, especially blades used for OS installation.

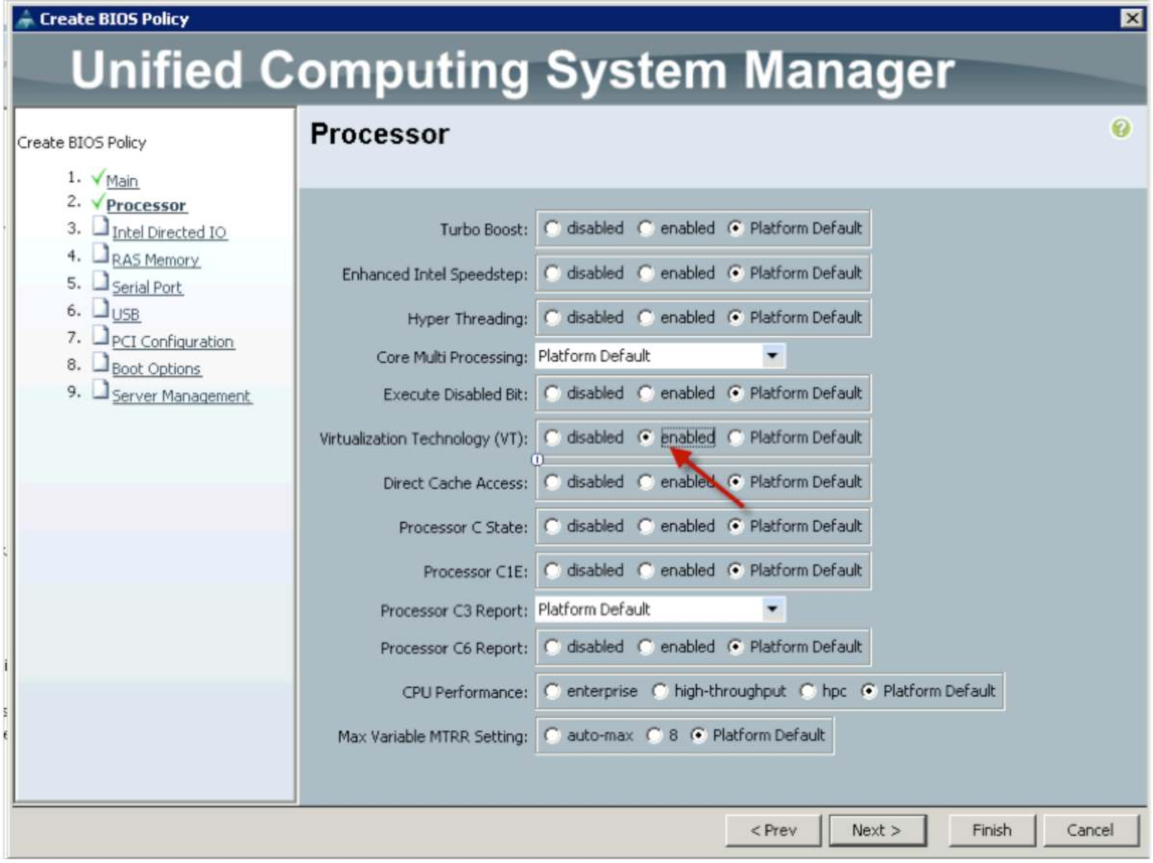
Scrub Policy

Scrub policy determines what happens to local disks and BIOS upon Service Profile disassociation. The default policy is no scrubbing. A best practice is to set the policy to scrub the local disk, especially for service providers, multi-tenant customers, and environments in which network installation to a local disk is used.

BIOS Policy

BIOS policy enables very specific control of CPU settings that are normally accessible only through the console during startup. For VMware and virtual environments that depend on CPU support for Intel Virtualization Technology, a corresponding policy can be created, removing any requirement for manual intervention during server provisioning. Similarly, applications that are sensitive to Intel Turbo Boost or Hyper-Threading can could have dedicated BIOS policies referenced, as shown in Figure 2. Also, setting “Quiet Boot” to “disabled” allows diagnostic message visibility, which may be helpful in troubleshooting situations.

Figure 5. BIOS Policies



Processor

Turbo Boost: ○ disabled ○ enabled ● Platform Default
Enhanced Intel Speedstep: ○ disabled ○ enabled ● Platform Default
Hyper Threading: ○ disabled ○ enabled ● Platform Default
Core Multi Processing: Platform Default
Execute Disabled Bit: ○ disabled ○ enabled ● Platform Default
Virtualization Technology (VT): ○ disabled ● enabled ○ Platform Default
Direct Cache Access: ○ disabled ○ enabled ● Platform Default
Processor C State: ○ disabled ○ enabled ● Platform Default
Processor C1E: ○ disabled ○ enabled ● Platform Default
Processor C3 Report: Platform Default
Processor C6 Report: ○ disabled ○ enabled ● Platform Default
CPU Performance: ○ enterprise ○ high-throughput ○ hpc ● Platform Default
Max Variable MTRR Setting: ○ auto-max ○ 8 ● Platform Default

Isolation

Cisco UCS addresses isolation requirements with the following objects:

- **VLANs:** These provide the foundation for network-based isolation. VLANs are created on the northbound LAN switch and then declared as available, since UCS does not create VLANs. Any declared VLANs can then be referenced when vNICs or vNIC templates are created.
- **VSANs:** These provide corresponding storage-based isolation capabilities. VSANs are created on the northbound SAN switch and then declared as available, since UCS does not create VSANs. Any declared VSAN can then be referenced when you create vHBAs or vHBA templates. Unlike with VLANs, vHBAs can associate with only a single VSAN.
- **Pin groups:** These provide isolation to specific northbound interfaces for both network and storage traffic. After pin groups are defined, they can be referenced as the target data path for any given vNIC or vHBA (or template) to help ensure that all traffic associated with a given vNIC or vHBA is isolated to the prescribed physical uplink ports.

Templates

Cisco UCS Manager provides templates for the primary objects (vNICs, vHBAs, and Service Profiles) to facilitate reuse and rapid deployment.

Best Practices: Templates

- In the GUI, use expert mode when creating Service Profile templates to achieve the optimal level of control and definition.
- When creating templates, reference the subordinate Pools and Policies that have been previously defined.

vNIC and vHBA Templates

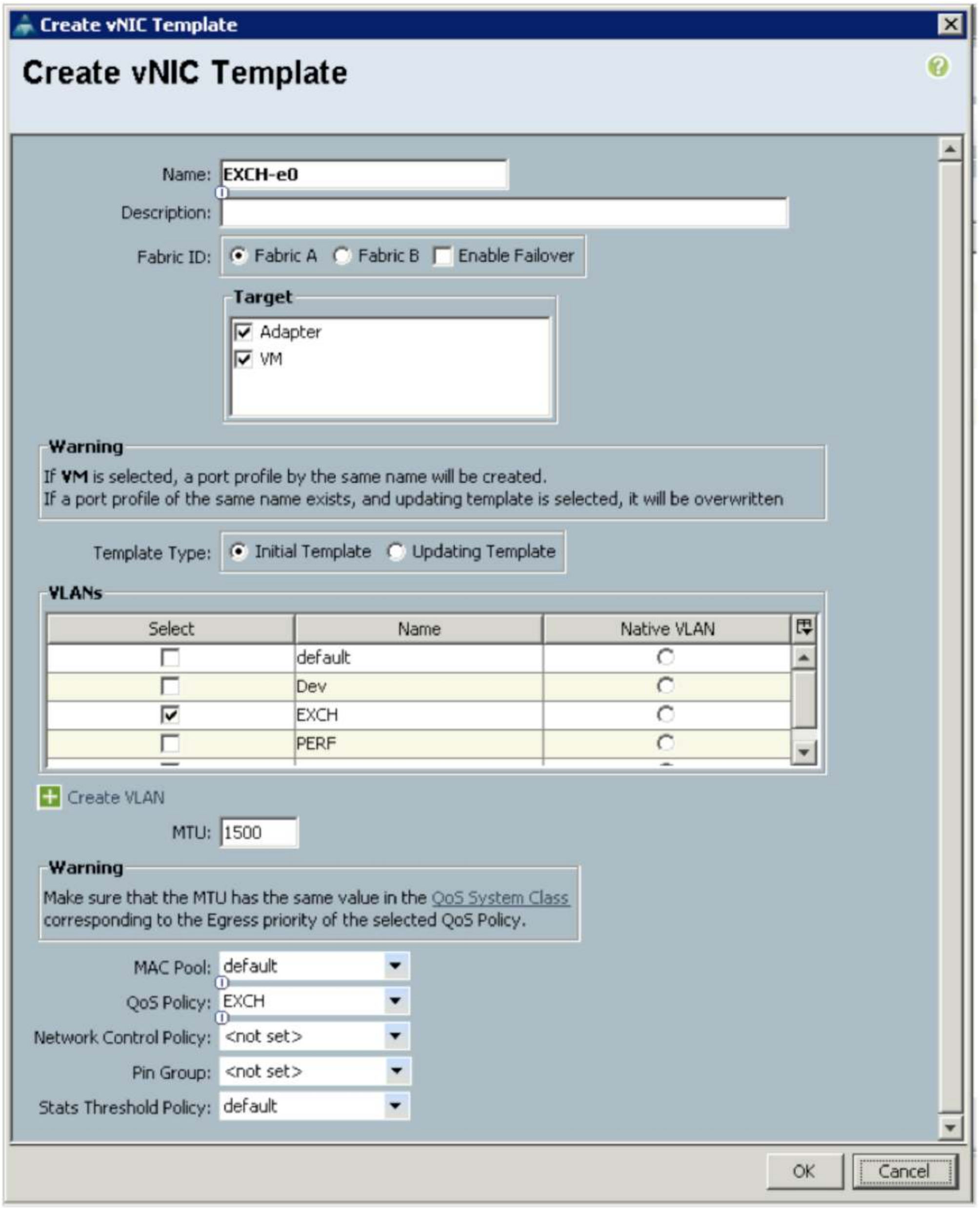
vNIC and vHBA resources are always associated with specific FIs (A-side or B-side fabric interconnect). A typical Service Profile has at least two vNICs and two vHBAs: one bound to each side. vNIC (or vHBA) templates can be used to encapsulate both the MAC address pool (or WWPN pool) association as well as the fabric ID.

Best Practice: vNIC and vHBA Templates

Create reusable vNIC and vHBA templates in which termination is either reflected in the name (e.g., “fc0-A”) or through well-accepted conventions (e.g., even interface to A side, and odd interface to B side).

vNIC templates should be thought of as application-specific network profiles that include important security definitions, such as VLAN mappings, as shown in Figure 6.

Figure 6. vNIC Template



Create vNIC Template

Create vNIC Template

Name: EXCH-e0

Description:

Fabric ID: ● Fabric A ○ Fabric B □ Enable Failover

Target

☑ Adapter
☑ VM

Warning

If **VM** is selected, a port profile by the same name will be created.
If a port profile of the same name exists, and updating template is selected, it will be overwritten

Template Type: ● Initial Template ○ Updating Template

VLANs

Select	Name	Native VLAN
□	default	○
□	Dev	○
☑	EXCH	○
□	PERF	○

Create VLAN

MTU: 1500

Warning

Make sure that the MTU has the same value in the QoS System Class corresponding to the Egress priority of the selected QoS Policy.

MAC Pool: default

QoS Policy: EXCH

Network Control Policy: <not set>

Pin Group: <not set>

Stats Threshold Policy: default

Availability Options

Network availability can be provided by either:

- Selecting Enable Failover, which provides availability at the hardware adapter level and helps ensure that service is not interrupted if one side of the fabric (FI or IOM) is down
- Using network interface card (NIC) teaming or NIC bonding, which provides availability at the host OS level

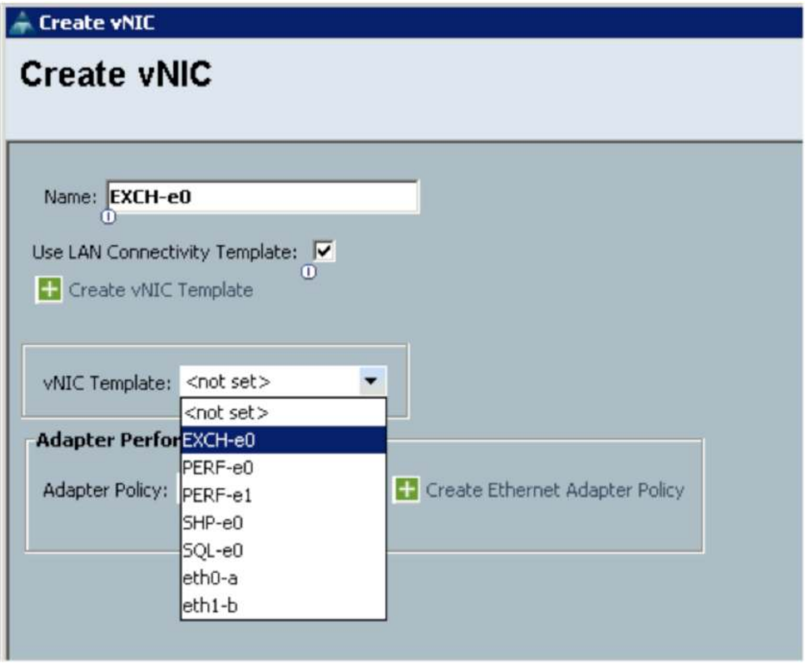
Storage availability can be provided only by host-side multi-pathing. Hardware failover is not an option for vHBAs.

Best Practice: Network Availability

For network availability, either use hardware failover or use NIC teaming (or bonding), but do not use both concurrently.

After a vNIC and vHBA template is defined, it can be referenced through expert-mode service-profile creation by selecting Use LAN (or SAN) Connectivity Template, as in Figure 7.

Figure 7. LAN Connectivity Template



Create vNIC

Name: EXCH-e0

Use LAN Connectivity Template: ☑

vNIC Template: <not set>

Service Profile Templates

Service Profile templates provide the means for associating all subordinate elements together (Pools, Policies, and vNIC and vHBA templates). Service Profile templates enables easy and rapid instantiation of Service Profiles, and rapid server provisioning.

Cisco UCS has two types of templates: “initial” (the default) and “updating”. After a template has been created and a Service Profile has been instantiated, the template type governs subsequent updating behavior and capabilities. Updating templates allow configuration changes to be reflected immediately across existing Service Profiles, while changes to initial templates are reflected only in newly instantiated (but not previously instantiated) Service Profiles.

Updating Templates: Use with Caution

Updating templates provide a very powerful feature that preserves the relationship between the template and the instantiated object (vNIC, vHBA, or Service Profile). The purpose of this feature is to enforce consistency and allow configuration changes to propagate at scale when, for example:

- Updating Host Firmware Package versions (BIOS included) for a large number of servers
- Mapping a new VLAN to all servers in a hypervisor cluster

Certain changes, such as updating VLAN mappings or QoS settings will not cause a service interruption. However, some changes reflected using updating templates may require a server reboot: for example, updates to BIOS policies or changes to versions of Host Firmware Packages. Service interruptions should always be governed through maintenance policies, so that all service interruptions only happen upon user acknowledgment or within scheduled maintenance windows.

Updating templates should be used with great awareness and caution. Updating templates can save considerable time during a scheduled maintenance window. However, they could also have unfortunate results if configuration changes are made at scale during normal operations.

While updating templates are designed to enforce consistency, exceptions are allowed. Service profiles created from an updating template can be changed individually, but must first unbind from the template.

Service Profile templates are best-suited for encapsulating and formalizing all service-level attributes for a given service or application without regard to physical connectivity constraints.

Best Practice: Service-Profile Templates

Use a Service Profile template as a definition for a class or type or version of an application, service, or operating system.

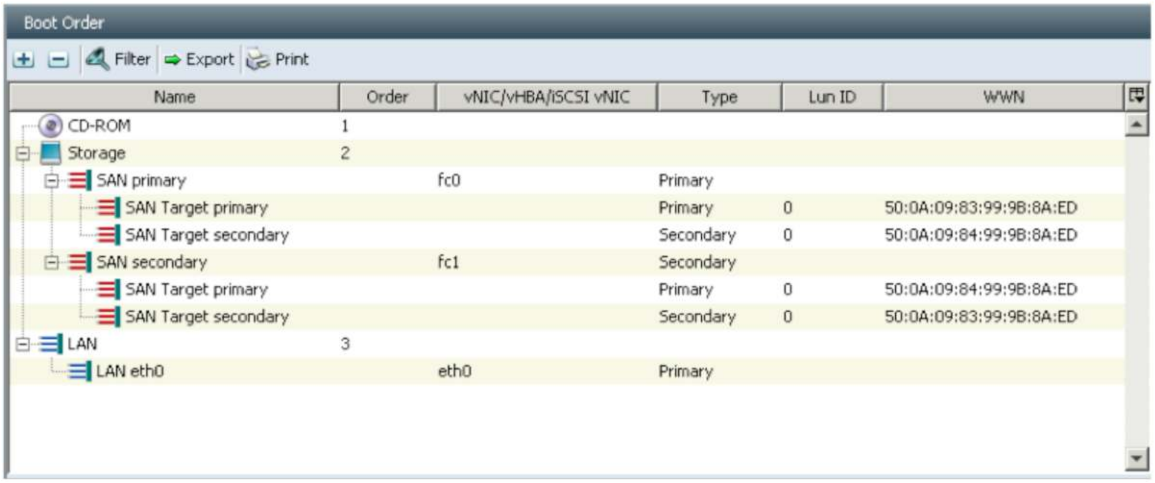
SAN-Boot Essentials

One of the core Cisco UCS values— service availability, provided by the stateless-server model—is predicated on SAN-boot. Server-image mobility cannot be achieved when booting from local disk. There are several considerations for preparing a reliable, consistent, and repeatable SAN-boot environment. Following are some best practices.

Configure Boot Policy for Availability

Boot policies allow booting from primary and secondary vHBAs as well as primary and secondary storage paths, as shown in Figure 8.

Figure 8. Boot Policies

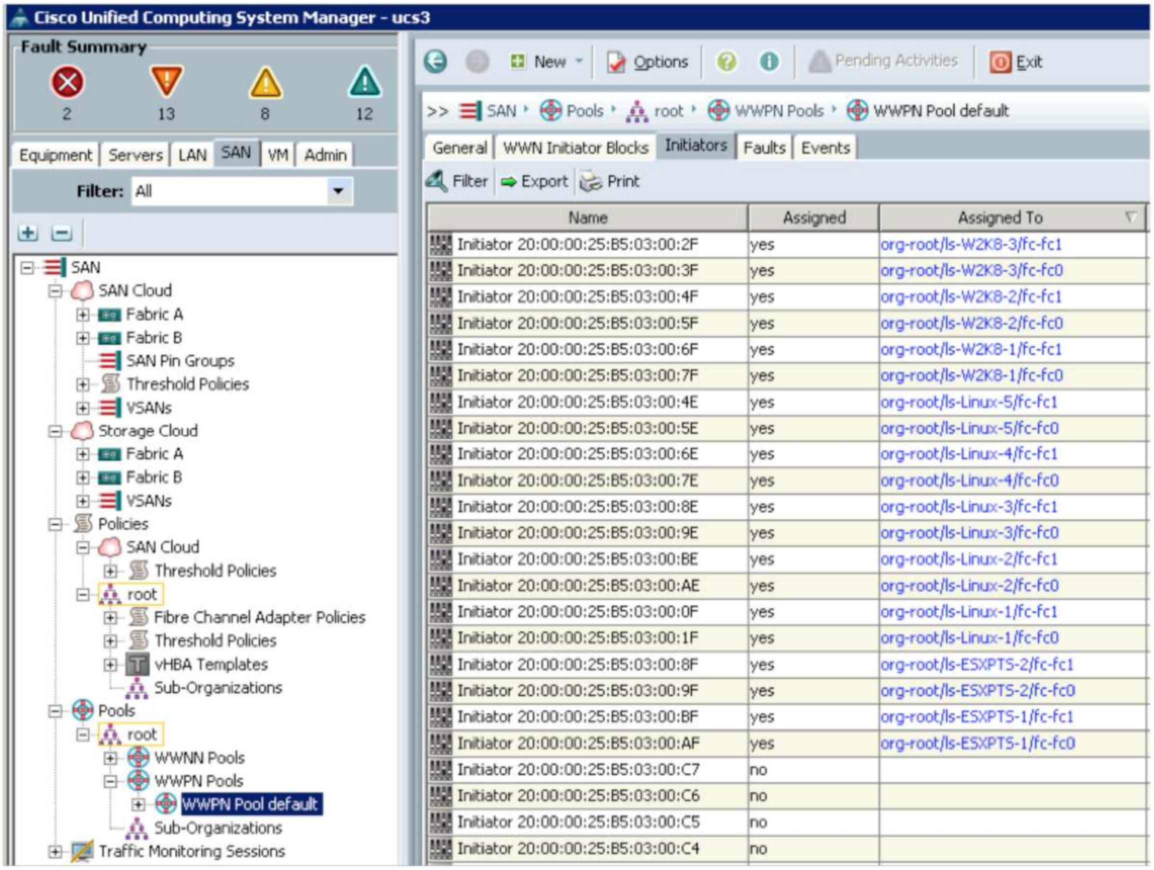


Name	Order	vNIC/vHBA/iSCSI vNIC	Type	Lun ID	WWN
CD-ROM	1				
Storage	2				
SAN primary		fc0	Primary		
SAN Target primary			Primary	0	50:0A:09:83:99:9B:8A:ED
SAN Target secondary			Secondary	0	50:0A:09:84:99:9B:8A:ED
SAN secondary		fc1	Secondary		
SAN Target primary			Primary	0	50:0A:09:84:99:9B:8A:ED
SAN Target secondary			Secondary	0	50:0A:09:83:99:9B:8A:ED
LAN	3				
LAN eth0		eth0	Primary		

Focus on the Service-Profile WWPNs

The WWPN of the Service Profile is the most significant key for integrating with both the SAN switch (zoning) and the SAN storage array (LUN masking). Figure 9 illustrates the mapping between WWPN pool elements and their corresponding service profiles.

Figure 9. WWPN Pool Elements and Service Profile Mapping



Name	Assigned	Assigned To
Initiator 20:00:00:25:B5:03:00:2F	yes	org-root/ls-W2K8-3/fc-fc1
Initiator 20:00:00:25:B5:03:00:3F	yes	org-root/ls-W2K8-3/fc-fc0
Initiator 20:00:00:25:B5:03:00:4F	yes	org-root/ls-W2K8-2/fc-fc1
Initiator 20:00:00:25:B5:03:00:5F	yes	org-root/ls-W2K8-2/fc-fc0
Initiator 20:00:00:25:B5:03:00:6F	yes	org-root/ls-W2K8-1/fc-fc1
Initiator 20:00:00:25:B5:03:00:7F	yes	org-root/ls-W2K8-1/fc-fc0
Initiator 20:00:00:25:B5:03:00:4E	yes	org-root/ls-Linux-5/fc-fc1
Initiator 20:00:00:25:B5:03:00:5E	yes	org-root/ls-Linux-5/fc-fc0
Initiator 20:00:00:25:B5:03:00:6E	yes	org-root/ls-Linux-4/fc-fc1
Initiator 20:00:00:25:B5:03:00:7E	yes	org-root/ls-Linux-4/fc-fc0
Initiator 20:00:00:25:B5:03:00:8E	yes	org-root/ls-Linux-3/fc-fc1
Initiator 20:00:00:25:B5:03:00:9E	yes	org-root/ls-Linux-3/fc-fc0
Initiator 20:00:00:25:B5:03:00:BE	yes	org-root/ls-Linux-2/fc-fc1
Initiator 20:00:00:25:B5:03:00:AE	yes	org-root/ls-Linux-2/fc-fc0
Initiator 20:00:00:25:B5:03:00:0F	yes	org-root/ls-Linux-1/fc-fc1
Initiator 20:00:00:25:B5:03:00:1F	yes	org-root/ls-Linux-1/fc-fc0
Initiator 20:00:00:25:B5:03:00:8F	yes	org-root/ls-ESXPTS-2/fc-fc1
Initiator 20:00:00:25:B5:03:00:9F	yes	org-root/ls-ESXPTS-2/fc-fc0
Initiator 20:00:00:25:B5:03:00:BF	yes	org-root/ls-ESXPTS-1/fc-fc1
Initiator 20:00:00:25:B5:03:00:AF	yes	org-root/ls-ESXPTS-1/fc-fc0
Initiator 20:00:00:25:B5:03:00:C7	no	
Initiator 20:00:00:25:B5:03:00:C6	no	
Initiator 20:00:00:25:B5:03:00:C5	no	
Initiator 20:00:00:25:B5:03:00:C4	no	

Note that there is no dependency on physical server association. In fact, pre-provisioning is a common use case, where the SAN infrastructure (zoning and LUN masking) can be completely preconfigured before any corresponding physical servers are present.

Use Single-Initiator Zoning—Never Open Zoning

Make sure that all storage array ports and the vHBA WWPNs are on the same VSAN and are zoned together. Use single-initiator zoning as a best practice. Never place all vHBA WWPNs and storage array ports in a single open zone.

Use the Fibre Channel Switch Name Server to Verify Connectivity

After a Service Profile has associated with a blade, the SAN name server can be viewed to verify proper connectivity between the Cisco UCS FIs and the SAN. On Cisco Nexus® and Cisco MDS 9000 Family switches, this is done with a **show flogi database** command from the command-line interface (CLI). All ports that have logged on to the Fibre Channel fabric will be visible. If intended vHBAs and WWPNs are not present in this table, then there likely is a UCSM configuration problem (the Service profile did not associate properly, the intended vHBA or WWPN was not referenced by the service profile, the FIs and the SAN switch were not cabled correctly, etc.).

Avoid Mixing Heterogeneous Storage Array Types

Mixing storage array types in a SAN-boot environment can be problematic, especially if the storage arrays are not capable of providing precise LUN mapping: specifying which LUN number gets presented to the host. Servers will typically boot from LUN 0. Results may be indeterminate if multiple storage arrays present multiple instances of LUN 0.

SAN-Boot Troubleshooting

Resolving SAN-boot problems typically require diagnostic examination from all 3 participating elements: the SAN switch, the storage array, and the server. Following are some common helpful techniques:

- Disable Quiet Boot in the BIOS policy to view boot-time diagnostics.
- Verify that the upstream SAN switch is enabled and configured for NPIV.
- Remove or unseat any local disks from blades during the OS installation phase.

Table 1. Common SAN-boot Problems

Observation	Likely Problem Causes
vHBA or storage array not present in the Fibre Channel name server	• Physical cabling is incorrect. • Ports on FI or SAN switch are not enabled.
vHBA visible from the name server but not from the storage array	• Zone or VSAN is misconfigured. • Storage array is not properly cabled or configured in the zone or VSAN.
Storage array visible from vHBA option ROM, but LUNs are not present or show up as “LUNZ” for EMC	LUN masking is not configured on the storage array.
vHBA or host shown as registered on a Clariion device, but other hosts with same WWPN appear as logged in	In multipath environments, all paths must be explicitly registered for EMC arrays.[7]
SAN installation proceeded correctly, but system will not subsequently boot	• Local disks are plugged into blade. • Host multipath drivers may be needed during installation phase.

Imaging and Provisioning

Cisco UCS provides two mechanisms for imaging servers: virtual media (from the KVM console) and network installation. Installation over the network is the best practice, because installations can be managed through automation with data transferred over 10 Gigabit Ethernet connections, whereas virtual media installations proceed manually over the 1 Gigabit Ethernet management interface.

[7] Multi-path registration is done automatically by NetApp arrays.

Installation of some versions of Microsoft Windows using virtual media may require manual mapping and unmapping between the ISO installation image and storage/network device drivers. For any errors encountered during the disk formatting phase, be sure to remap the original ISO installation image.

Provisioning boot LUNs for several similar servers is best accomplished by first creating a “golden image” boot LUN and then using intelligent storage array features such as LUN or volume cloning to replicate the boot image.

Note: Cloned LUNs works well for Linux and Microsoft Windows, but not for VMware.

IP address assignment is best managed centrally by a Domain Host Configuration Protocol (DHCP) server; hostnames are best set based on the result of a reverse Domain Name System (DNS) lookup.

Microsoft Windows activation is best managed through volume licensing.

Advanced Topics

Server Pools

Server Pools provide a means for partitioning and segregating physical blades into different groups. The grouping criteria are left to the administrator. Possible criteria may focus on physical server capabilities (CPU speed, memory size, use of local disks or not, etc.), logical business divisions (marketing, finance, etc.), specific customers being hosted, or specific geographies being served. Server Pools can be built manually from individual blades, or they can be populated automatically through Server Pool Policy Qualifiers. After Server Pools have been defined, they can be associated with individual Service Profiles or with a Service Profile template.

You can use Service Profile templates together with qualified Server Pools as a way of providing and enforcing a hardware-based service-level agreement for applications, as appropriate.

Pool-Policy Qualifiers and Pool Policies

Pool-policy Qualifiers and Pool Policies work in conjunction with Server Pools. Use Pool-Policy Qualifiers to separate blades into different sets or classes. Common qualifiers include:

- Memory size
- Number of CPU cores
- Blade model or type
- Physical chassis or slot
- Power group

Pool Policies form an association between Pool Policy Qualifiers and Server Pools. Server Pools can be populated immediately as Pool Policies are created, with any newly discovered blades automatically sorted into the respective Server Pools. Individual blades can be present in multiple, overlapping Server Pools at the same time.

As a best practice, the same name should be used for Pool-Policy Qualifiers and Pool Policies (and optionally for the Server Pool as well) for ease of correlation.

Configuration Backup

The Cisco UCS configuration can be backed up easily and should be backed up regularly either through the GUI or automated scripts.

There are four types of backups, summarized in Table 2.

Table 2. Configuration Backup Types

Type	Format	Description	Size
Full State	Binary	Used for full system restore as part of disaster recovery	~ 1 to 10 MB
System Configuration	XML	Roles, Call Home, communication services, distributed virtual switch, etc.	~100 to 500 KB
Logical Configuration	XML	Service profiles, VLANs, VSANs, pools, policies, templates, etc.	~100 to 500 KB
All Configurations	XML	Both Logical and System configurations	~200 to 1000 KB

For the Logical Configuration and All Configurations backups, select the Preserve Identities feature to preserve the actual MAC address, WWN, and UUID values; otherwise, the backup references only the logical pool names, but not the actual identities.

“Gold configurations” can be used at the domain level, similar to the way that “gold images” can be used at the server level. This approach involves developing a standard configuration that serves as a UCS domain template, which implements an All Configurations backup without preserving identities. In an environment with multiple UCS domains, the golden configuration could be imported to subsequent domains. Identity pools would use the same names across all domains, but they would be populated with domain-specific values to avoid collisions. This approach allows standardization of common objects (policies, service profiles, templates, etc.) across multiple domains.

Best Practices: Preserve Identities

- Use the Preserve Identities feature when backing up individual domains for prescribed restoration (same site or domain or exact recovery site or domain).
- Do not use Preserve Identities when creating “gold UCSM domain configuration” templates.

Power Groups and Power-Control Policies

Power Groups and Power-Control Policies work together to help ensure that power limitation thresholds are not crossed and that workloads are appropriately prioritized after any power thresholds are approached. Power Groups are defined by a number of chassis that share a common power limitation dependency: for example, all chassis on the same power strip, or in the same physical rack, or in a set of racks that share a common power circuit.

The Power-Cap value for a Power Group provides a power ceiling threshold that will not be crossed. The Power-Control Policies are referenced from a Service Profile (or template) and provide prioritization or relative weighting among workloads. If the Power-Cap value is reached for a given Power Group, then the power draw (and performance) for all servers in that group will be diminished in proportion to their relative weighting, so that the Power-Cap threshold is not crossed. Servers with a lower priority will be slowed more than servers with a higher priority. Referencing a Power-Control Policy with no cap will exempt that server from the power capping feature.

Organizations

“Organization” is a hierarchical construct that allows administrators to partition resources logically, to scale management more effectively, and to support multi-tenancy. Organizations exist hierarchically, with root as the default top level. Any sub-organizations created will have a corresponding set of Pools, Policies, and templates within their management scope.

Two important properties apply to the management of hierarchical organizations: inheritance and override. For inheritance, any Pools, Policies, and templates that are defined at a parent level in the hierarchy (for example, at the root level) can be referenced by objects in subordinate child organizations in the hierarchy (for example, by Service Profiles in sub-organizations under root). Override characteristics give autonomy to local organization administrators. Any local changes or overrides to Pools, Policies, and templates within a child organization will not be shared by or visible to peers or parents.

Although identify pools can be maintained at local organizational levels, a best practice is be to have a single set of UUID, MAC address, WWNN and WWPN pools maintained exclusively at the root level and created in close coordination with the data center's site-wide catalog.

Monitoring

Cisco UCS provides the standard set of health and monitoring methods such as syslog and Simple Network Management Protocol (SNMP) with its associated MIBs[8] (**get** and **fault traps** only; no **set**). The best practice for UCS monitoring is to use existing methods and frameworks that are already familiar and well understood, such as SCOM,[9] OpenView, or BPPM.[10]

Lightweight Directory Access Protocol (LDAP)

Cisco UCS was designed to integrate seamlessly with existing authentication frameworks, such as LDAP and Active Directory. While the basic configuration has already been well documented,[11] here are some of best practices to keep in mind:

- Maintain matching role names between Active Directory and Cisco UCS.
- Use non-expiring passwords for the non-administrative bind user account (which periodically verifies group membership).
- Test or verify all LDAP providers as they are added.

Cisco Fabric Extender Link (FEX-Link), Data Center VM-FEX, and VN-Tag

Cisco FEX-Link is a foundational Cisco networking innovation to extend and scale network configuration, management, and monitoring without adding proportional management and administrative overhead, as exemplified by the Cisco Nexus architecture.[12] Additional network ports can be provided by adding fabric extenders, which act as "remote line cards", while still maintaining centralized administration through a single controlling switch.

Cisco FEX-Link is fundamental in the architecture of Cisco UCS. The internal architecture of the FIs are essentially the same as for the Nexus 5K,[13] and the internal architecture of the chassis IOMs are essentially the same as for the Nexus 2K.[14] This architecture is what allows a Cisco UCS domain to add chassis without increasing management overhead.

[8] ftp://ftp.cisco.com/pub/mibs/supportlists/ucs/ucs-manager-supportlist.html
[9] http://www.cisco.com/en/US/docs/unified_computing/ucs/sw/scom/quick/start/guide/ucsMPQS.html
[10] http://cisco.com/go/bmc
[11] http://www.cisco.com/en/US/docs/unified_computing/ucs/sw/sample_configurations/UCSM_1_4_LDAP_with_AD/b_Sample_Configuration_LDAP_with_AD.html
[12] Cisco Nexus 5000 and Nexus 7000 Series Switches, and Cisco Nexus 2000 Series Fabric Extenders.
[13] Cisco Nexus 5000 Series Switch
[14] Cisco Nexus 2000 Series Fabric Extender

Manageability problems can be exacerbated by the pervasive adoption of virtualization and the consolidation of systems and fabrics. Consolidation pushes the industry toward a smaller number of physical ports, but configuration, management, and monitoring still require detailed control over the virtual endpoints.

Cisco FEX-Link solves this problem at the network transport layer, by injecting a new industry-standard[15] tag that maintains a context for the flows corresponding to virtual network endpoints or “virtual ports”. In this way, a single physical port can scale to serve a large number of individual (virtual) network interfaces (analogous to the way that NPIV solves the problem with Fibre Channel ports). Cisco FEX-Link treats network endpoints consistently from a management standpoint, regardless of whether the endpoints correspond to a physical server’s network port or a virtual machine’s virtual network port.

The main benefits of Cisco FEX-Link include the following:

- Network policy configuration is maintained centrally by the central controlling switch.
- Network policy configuration is applied consistently, regardless of whether the endpoint is a physical or a virtual port.

Cisco VM-FEX[16] extends the FEX-Link model in to the realm of virtualization by creating a virtual network link (“VN-Link”[17]). Network management, configuration, and monitoring are still maintained centrally, but with a “virtual fabric extender” getting embedded in the hypervisor. With the Cisco Nexus 1000V Series Switches, VN-Link is implemented in software; with the Cisco UCS virtual interface card (VIC), VN-Link is implemented in hardware through VM-FEX.

In the virtualization realm, all virtual machine network traffic typically is proxied through the hypervisor host, and as a virtual machine migrates dynamically from host to host (for example, using VMware vMotion or DRS), all the virtual machine’s network context is lost. VM-FEX allows network administrators to maintain context and affinity on a per–virtual machine basis for each virtual network interface. Networking traffic for a virtual machine will be connected with a virtual Ethernet port that is associated with the virtual machine itself, not with the hypervisor host. Through this virtual port, the network administrator can apply policy, configure security, and monitor traffic precisely for an individual virtual machine’s vNIC, regardless of the hypervisor.

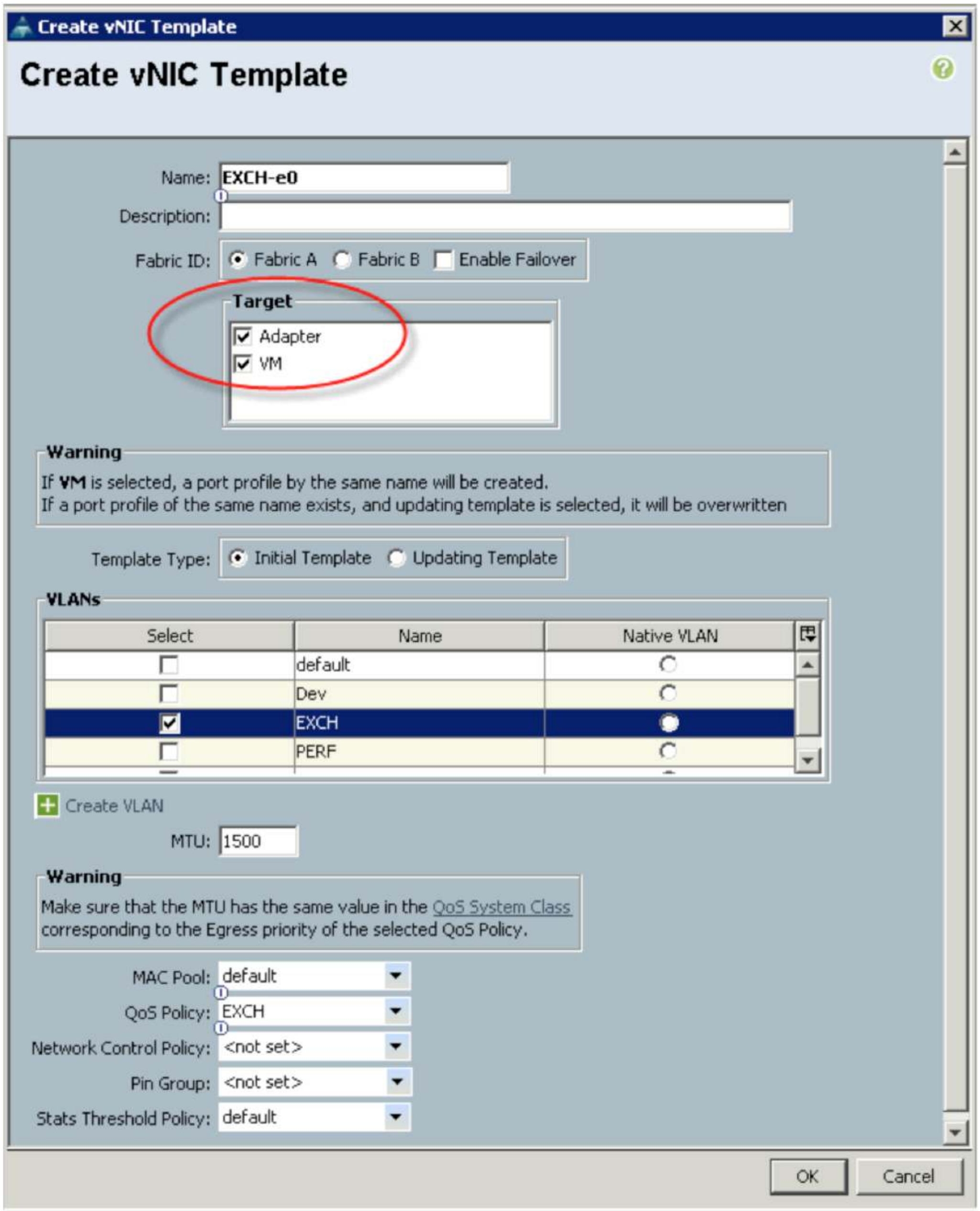
Consider a server physical-to-virtual migration use case as an example. From the networking standpoint, the goal is to have the exact same network configuration and policy provided—such as VLAN-based isolation and QoS–based SLAs—regardless of whether the deployment is physical or virtual machine. Figure 10 of the GUI shows how vNIC templates are created.

[15] VN-Tag: http://standards.ieee.org/regauth/ethertype/eth.txt

[16] Cisco Data Center Virtual Machine Fabric Extender

[17] http://www.cisco.com/en/US/solutions/collateral/ns340/ns517/ns224/ns892/ns894/white_paper_c11-525307.html

Figure 10. vNIC Template Creation Wizard



Create vNIC Template

Create vNIC Template

Name: EXCH-e0

Description:

Fabric ID: ● Fabric A ○ Fabric B □ Enable Failover

Target

☑ Adapter

☑ VM

Warning

If **VM** is selected, a port profile by the same name will be created.
If a port profile of the same name exists, and updating template is selected, it will be overwritten

Template Type: ● Initial Template ○ Updating Template

VLANs

Select	Name	Native VLAN
□	default	○
□	Dev	○
☑	EXCH	●
□	PERF	○

Create VLAN

MTU: 1500

Warning

Make sure that the MTU has the same value in the QoS System Class corresponding to the Egress priority of the selected QoS Policy.

MAC Pool: default

QoS Policy: EXCH

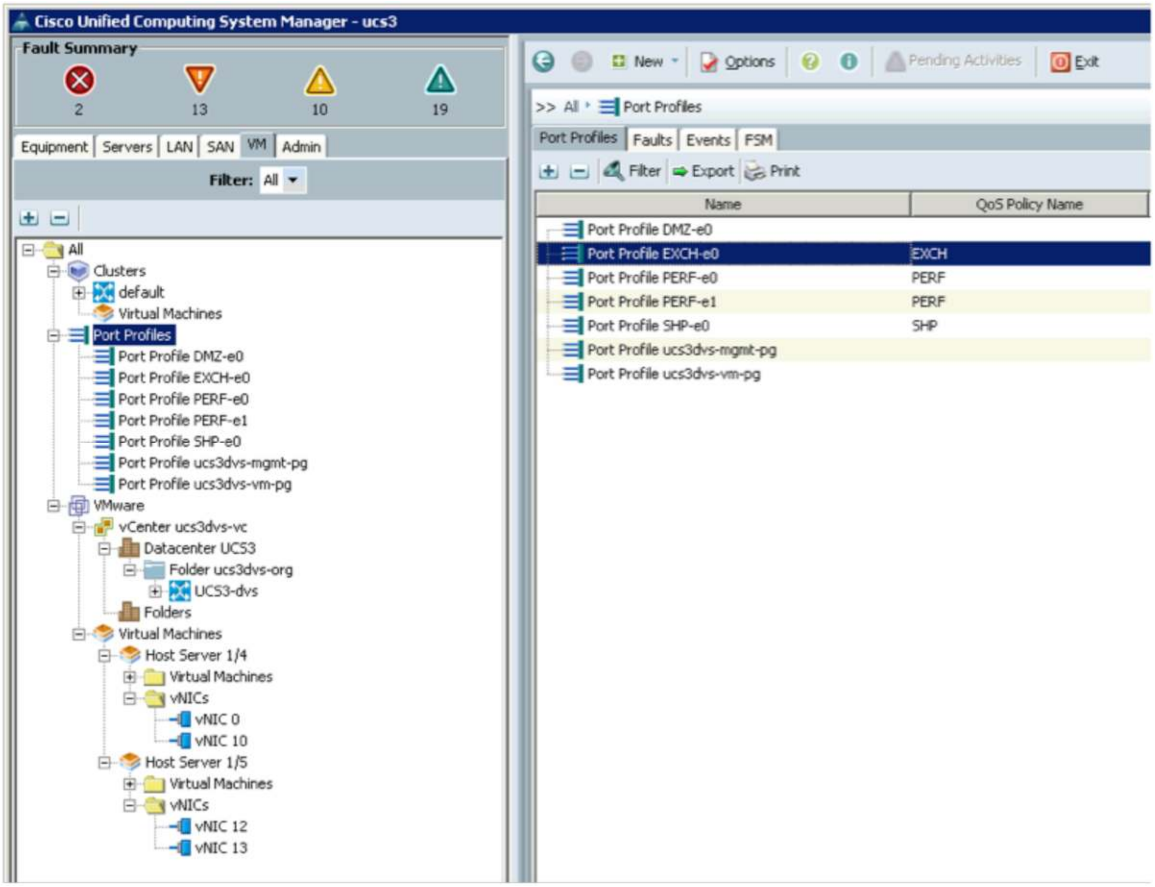
Network Control Policy: <not set>

Pin Group: <not set>

Stats Threshold Policy: default

The Target box presents one of the more profound capabilities in the Cisco UCS model: uniform network policy configuration regardless of physical or virtual machine boundaries. When the EXCH-e0 network profile is created (with all associated VLANs and QoS policies), the Target box allows this network policy to be applied to either a physical machine (Adapter), a virtual machine (VM), or both. The physical machine (Adapter) would apply to a Service Profile. However, if VM is included as the target, this vNIC template becomes a network port profile that can then be applied and exported to a distributed virtual switch, as illustrated in GUI through the VM tab, shown in Figure 11.

Figure 11. Port Profiles



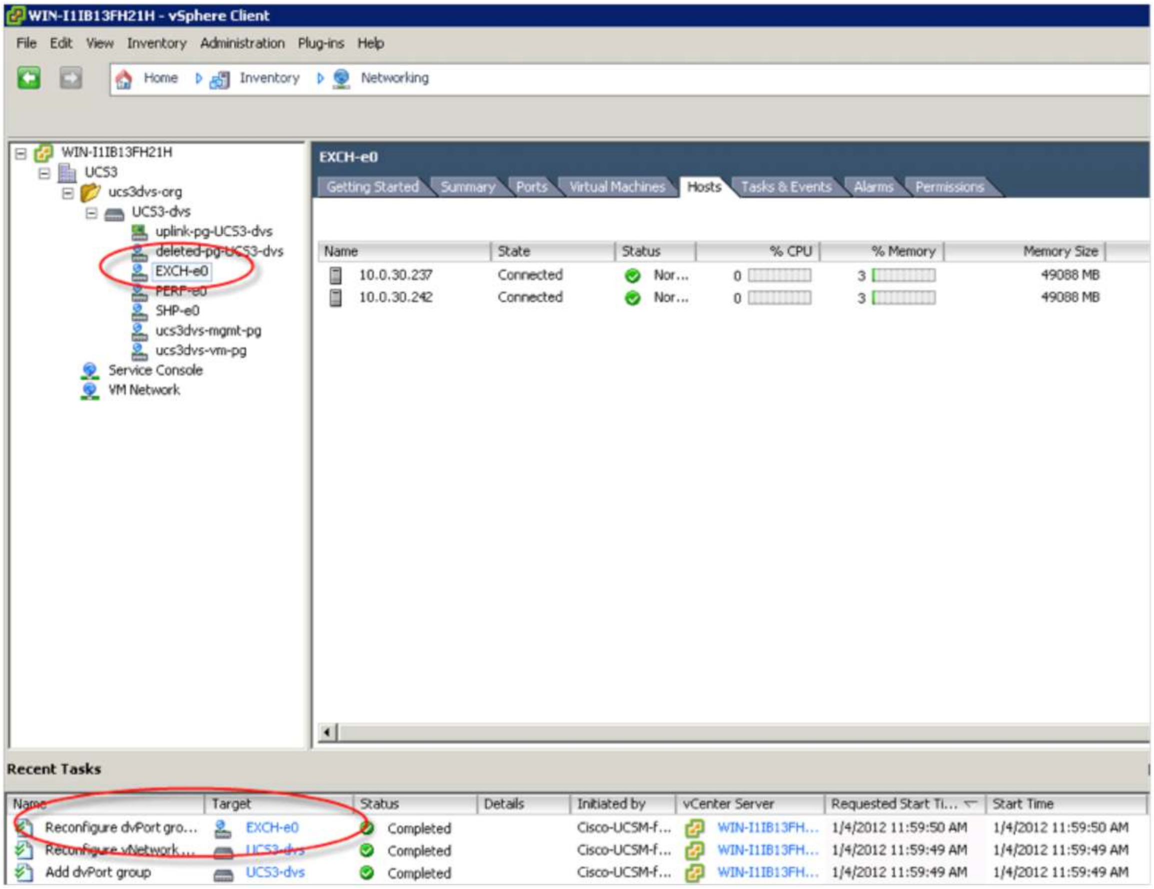
Name	QoS Policy Name
Port Profile DMZ-e0	
Port Profile EXCH-e0	EXCH
Port Profile PERF-e0	PERF
Port Profile PERF-e1	PERF
Port Profile SHP-e0	SHP
Port Profile ucs3dvs-mgmt-pg	
Port Profile ucs3dvs-vm-pg	

Using VMware as an example, assume that:

- The Cisco UCS Manager and VMware vCenter Server (VCS) instances have both performed proper security handshakes.
- A distributed virtual switch has been configured on the VMware VCS instance.
- The port profile has been created and exported through UCSM.

The port profile will appear as a network port profile in VMware VCS, as shown in Figure 12.

Figure 12. VMware Network Port Profile

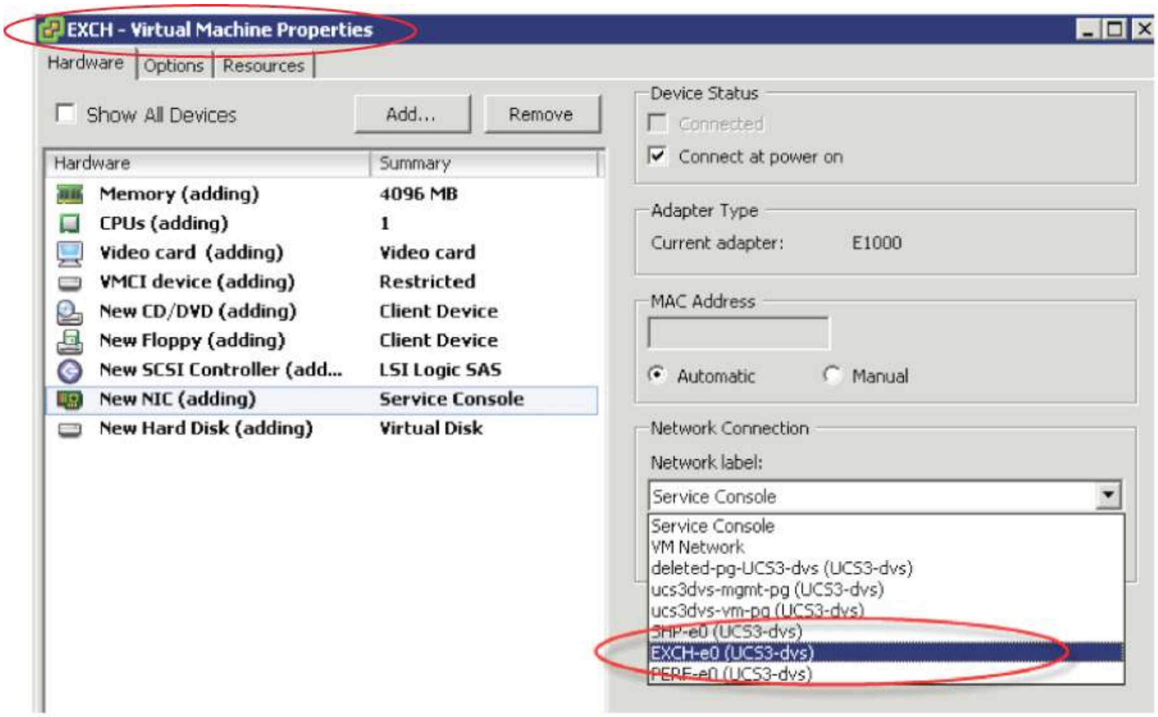


Name	State	Status	% CPU	% Memory	Memory Size
10.0.30.237	Connected	Nor...	0	3	49088 MB
10.0.30.242	Connected	Nor...	0	3	49088 MB

Name	Target	Status	Details	Initiated by	vCenter Server	Requested Start Ti...	Start Time
Reconfigure dvPort gro...	EXCH-e0	Completed		Cisco-UCSM-f...	WIN-I1IB13FH...	1/4/2012 11:59:50 AM	1/4/2012 11:59:50 AM
Reconfigure vNetwork...	UCS3-dvs	Completed		Cisco-UCSM-f...	WIN-I1IB13FH...	1/4/2012 11:59:49 AM	1/4/2012 11:59:49 AM
Add dvPort group	UCS3-dvs	Completed		Cisco-UCSM-f...	WIN-I1IB13FH...	1/4/2012 11:59:49 AM	1/4/2012 11:59:49 AM

Server and networking administrative roles can then be better segregated and normalized. As virtual machine administrators configure virtual machines, they are no longer required to be network administrators as well. The network administrator is responsible for creating network policy (for example, EXCH-e0), and the virtual machine administrator simply consumes the appropriately prescribed network port policy, as shown in Figure 13.

Figure 13. Associating Network Port Policy to a VM


EXCH - Virtual Machine Properties

□ Show All Devices

Hardware	Summary
Memory (adding)	4096 MB
CPUs (adding)	1
Video card (adding)	Video card
VMCI device (adding)	Restricted
New CD/DVD (adding)	Client Device
New Floppy (adding)	Client Device
New SCSI Controller (add...	LSI Logic SAS
New NIC (adding)	Service Console
New Hard Disk (adding)	Virtual Disk

Device Status
□ Connected
☑ Connect at power on

Adapter Type
Current adapter: E1000

MAC Address
● Automatic ○ Manual

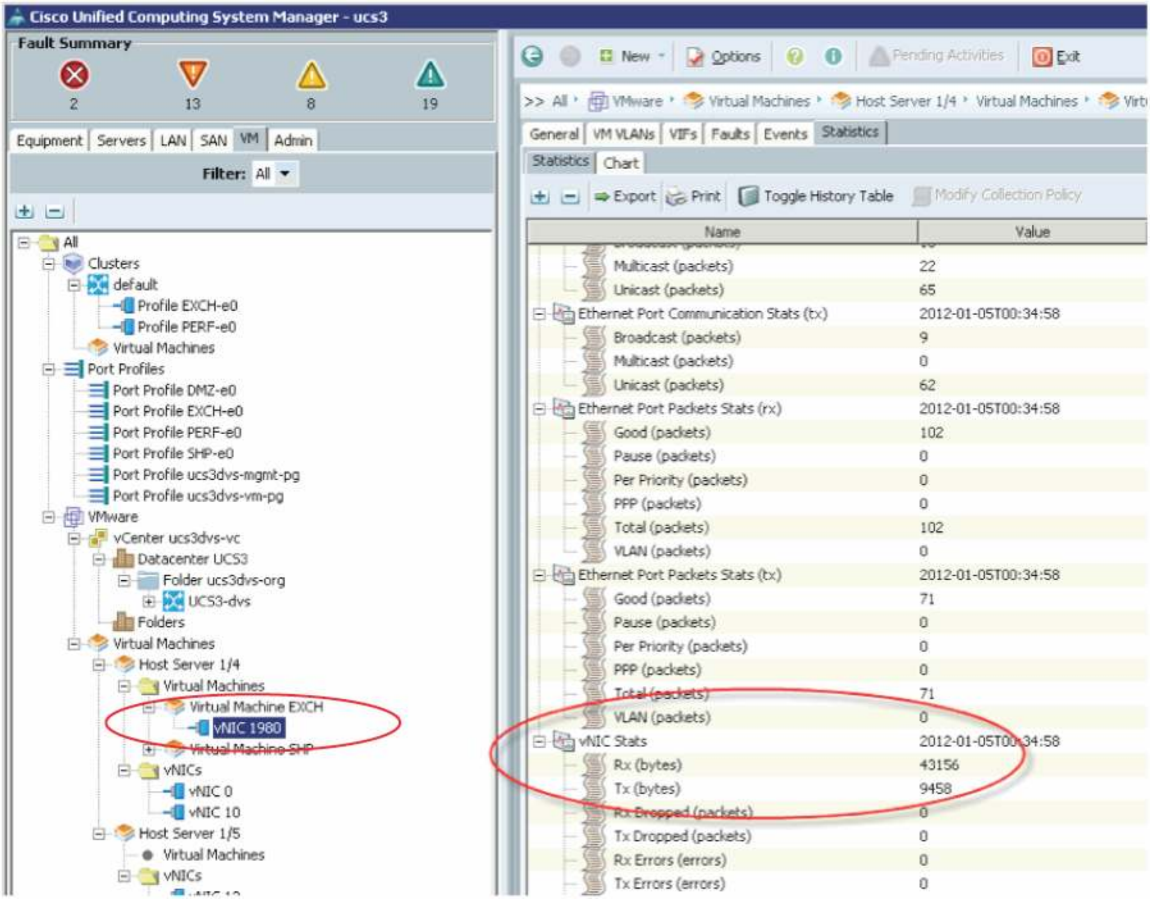
Network Connection
Network label:
Service Console

Service Console
VM Network
deleted-pg-UCS3-dvs (UCS3-dvs)
ucs3dvs-mgmt-pg (UCS3-dvs)
ucs3dvs-vm-pg (UCS3-dvs)
SHP-e0 (UCS3-dvs)
EXCH-e0 (UCS3-dvs)
PERF-e0 (UCS3-dvs)

As the virtual machine powers up and begins passing network traffic, all traffic flows across its virtual port. Instead of traffic being switched locally by the host-based virtual-switch, the virtual link provided by VM-FEX allows traffic to bypass the hypervisor and be switched at the physical switch layer (for example, the FIs).

The virtual ethernet port provides consistent network context for all traffic associated with a VM's virtual network port, regardless of the host on which the VM is running. All VM network traffic is tagged with the industry standard VN-Tag and bypasses any local host-based switching. The network administrator can then maintain a consistent and persistent view into the management, monitoring, state, and statistics of the network flows, even as VMs move between different hosts, as shown in Figure 14.

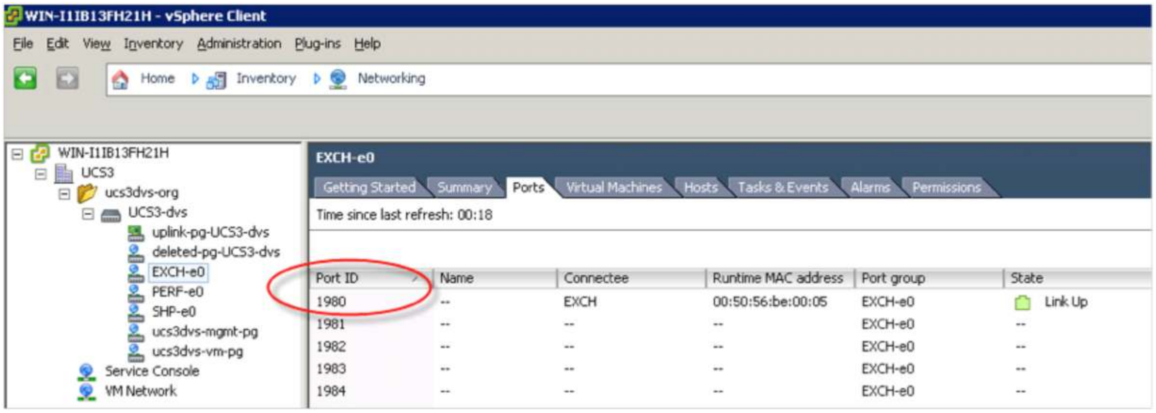
Figure 14. View of Management, Monitoring, State, and Statistics of Network Flows



Name	Value
Multicast (packets)	22
Unicast (packets)	65
Ethernet Port Communication Stats (tx)	2012-01-05T00:34:58
Broadcast (packets)	9
Multicast (packets)	0
Unicast (packets)	62
Ethernet Port Packets Stats (rx)	2012-01-05T00:34:58
Good (packets)	102
Pause (packets)	0
Per Priority (packets)	0
PPP (packets)	0
Total (packets)	102
VLAN (packets)	0
Ethernet Port Packets Stats (tx)	2012-01-05T00:34:58
Good (packets)	71
Pause (packets)	0
Per Priority (packets)	0
PPP (packets)	0
Total (packets)	71
VLAN (packets)	0
vNIC Stats	2012-01-05T00:34:58
Rx (bytes)	43156
Tx (bytes)	9458
Rx Dropped (packets)	0
Tx Dropped (packets)	0
Rx Errors (errors)	0
Tx Errors (errors)	0

The vNIC virtual port number presented by UCSM can be easily correlated with the virtual port number presented by VCS, as a way of maintaining context for the network activity.

Figure 15. vNIC virtual port number as context



Time since last refresh: 00:18

Port ID	Name	Connectee	Runtime MAC address	Port group	State
1980	--	EXCH	00:50:56:be:00:05	EXCH-e0	Link Up
1981	--	--	--	EXCH-e0	--
1982	--	--	--	EXCH-e0	--
1983	--	--	--	EXCH-e0	--
1984	--	--	--	EXCH-e0	--

Although these examples were based on VMware, VM-FEX is a transport innovation that is completely hypervisor independent. VM-FEX capabilities are also supported with KVM[18] under Red Hat Linux 6.1, and with Microsoft Hyper-V[19] under Microsoft Windows Server 2012.

Cisco FEX-Link and VN-Tag port-extension capabilities form the foundation upon which Cisco UCS is built. Many data center operators grapple with distinct and differing methods for managing physical servers and virtual machines. Cisco FEX-Link provides a paradigm for unification: allow network traffic to be managed, configured, and monitored in exactly the same way for both physical and virtual machines.

Best Practice: Configuring Network Policy

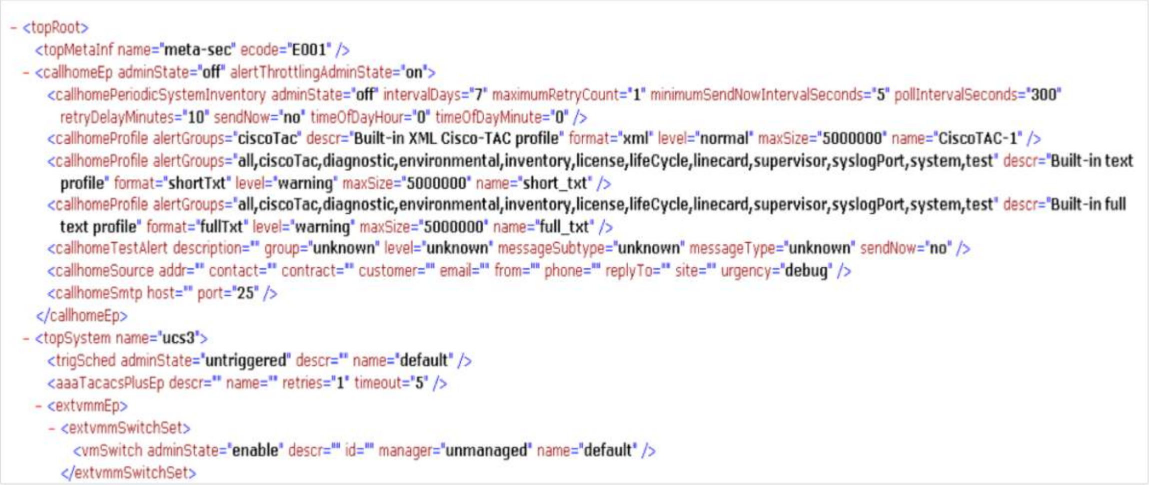
Configure network policy based on isolation and security requirements and SLAs, not based on virtual or physical boundaries, and not based on physical connectivity constraints.

Access Through the XML-Based API

As administrators become familiar with Cisco UCS (typically through the GUI) remember that the GUI is not the Cisco UCS Manager. The Cisco UCS Manager is the management engine that runs as a privileged guest of Cisco NX-OS within a clustered pair of fabric interconnects.

The only way to access the Cisco UCS Manager is through the open, published, and documented XML-based API.[20] In fact, the managed-objects model and the XML API were among the first elements designed in the Cisco Unified Computing System, well before any physical objects had been engineered.

To gain valuable insight into the nature of the managed-objects model, simply view the contents of a configuration backup:



```
- <topRoot>
    <topMetaInf name="meta-sec" ecode="E001" />
  - <callhomeEp adminState="off" alertThrottlingAdminState="on">
      <callhomePeriodicSystemInventory adminState="off" intervalDays="7" maximumRetryCount="1" minimumSendNowIntervalSeconds="5" pollIntervalSeconds="300"
        retryDelayMinutes="10" sendNow="no" timeOfDayHour="0" timeOfDayMinute="0" />
      <callhomeProfile alertGroups="ciscoTac" descr="Built-in XML Cisco-TAC profile" format="xml" level="normal" maxSize="5000000" name="CiscoTAC-1" />
      <callhomeProfile alertGroups="all,ciscoTac,diagnostic,environmental,inventory,license,lifeCycle,linecard,supervisor,syslogPort,system,test" descr="Built-in text
        profile" format="shortTxt" level="warning" maxSize="5000000" name="short_txt" />
      <callhomeProfile alertGroups="all,ciscoTac,diagnostic,environmental,inventory,license,lifeCycle,linecard,supervisor,syslogPort,system,test" descr="Built-in full
        text profile" format="fullTxt" level="warning" maxSize="5000000" name="full_txt" />
      <callhomeTestAlert description="" group="unknown" level="unknown" messageSubtype="unknown" messageType="unknown" sendNow="no" />
      <callhomeSource addr="" contact="" contract="" customer="" email="" from="" phone="" replyTo="" site="" urgency="debug" />
      <callhomeSmtp host="" port="25" />
    </callhomeEp>
  - <topSystem name="ucs3">
      <trigSched adminState="untriggered" descr="" name="default" />
      <aaaTacacsPlusEp descr="" name="" retries="1" timeout="5" />
    - <extvmmEp>
      - <extvmmSwitchSet>
          <vmSwitch adminState="enable" descr="" id="" manager="unmanaged" name="default" />
        </extvmmSwitchSet>
```

(…)

The backup looks like a hierarchical XML database, which conveys much of the essence of Cisco UCS Manager. Through the lens of the GUI, all these objects and attributes are projected through the various tabs and views.

[18] http://www.cisco.com/en/US/docs/unified_computing/ucs/sw/vm_fex/kvm/gui/config_guide/GUI_KVM_VM-FEX_UCSM_Configuration_Guide_chapter4.html
[19] http://www.cisco.com/en/US/solutions/collateral/ns340/ns517/ns224/ns955/ns963/solution_overview_c22-687087.html
[20] http://www.cisco.com/en/US/partner/docs/unified_computing/ucs/sw/api/ucs_api_book.html

To illustrate that even the GUI can gain access only through the XML API, you can examine the diagnostic trace file[21] to find the API calls and responses:

```
[------------- Sending Request to Server ------------
  <configConfMos
  inHierarchical="true">
    <inConfigs>
  <pair key="org-root/ls-EXCH-2">
    <lsServer
    agentPolicyName=""
    biosProfileName=""
    bootPolicyName="SAN-Boot"
    descr=""
    dn="org-root/ls-EXCH-2"
  (…)
    </lsServer>
  </pair>
    </inConfigs>
  </configConfMos>
-----------------------------------------------------]
[---------- Received Response from Server -----------
HTML Headers:
  Response: HTTP/1.1 200 OK
  Date: Fri, 06 Jan 2012 04:16:57 GMT
  Server: Apache/2.2.17 (Unix) mod_ssl/2.2.17 OpenSSL/FIPS
  Content-Length: 3385
  Connection: close
  Content-Type: application/soap+xml
[----------debugBuffer----------------]
  <configConfMos cookie="[hidden]" response="yes"> <outConfigs> <pair key="org-
root/ls-EXCH-2"> <lsServer agentPolicyName="" assignState="unassigned"
assocState="unassociated" biosProfileName="" bootPolicyName="SAN-Boot"
childAction="deleteNonPresent" configQualifier="" configState="not-applied"
descr="" dn="org-root/ls-EXCH-2" dynamicConPolicyName="" extIPState="none"
fltAggr="0" fsmDescr="" (…)
intId="975398" modified="1970-01-01T00:00:00.000" name="" oldPnDn=""
operState="untriggered" rn="ack" scheduler="" status="created"/> </lsServer>
</pair> </outConfigs> </configConfMos>
```

This trace from an active GUI session highlights an important point: **all** aspects of configuring, managing, and monitoring Cisco UCS can all be performed over HTTP or HTTPS. There is nothing proprietary in the access path for controlling all aspects of the Cisco UCS environment.

[21] "C:\Documents and Settings\<UserID>\AppData\LocalLow\Sun\Java\Deployment\log\.ucsm" in Windows 7

The XML API is so central and essential that many software development kits (SDK's), tools and utilities have been developed to help facilitate use and integration:

- Cisco UCS PowerTool[22] is a flexible and powerful command line toolkit that includes more than 1700 PowerShell cmdlets, to provide an easy way to interface, integrate and automate UCS management for Microsoft Windows-based administration. UCS PowerTool can be easily combined with many other 3rd party products that take advantage of the flexible and powerful scripting environment offered by Microsoft PowerShell.
- Cisco UCS Python SDK[23] is a Python module which helps automate all aspects of Cisco UCS management including server, network, storage and management for Linux-based administration.
- UCSPE is the Cisco UCS Manager Platform Emulator[24] that runs as a virtual machine. UCSPE provides the full capabilities of Cisco UCS Manager, through emulation of the underlying hardware. UCSPE exports an active instance of Cisco UCS Manager, including the XML API and GUI. Both Cisco UCS PowerTool and Cisco UCS Python SDK can be run against a UCSPE instance. Furthermore, UCSPE has the ability to import both hardware configurations and logical backups from live UCS domains, thereby providing a "safe sandbox" to test configuration changes in a risk-free environment.
- "Visore" is an object browser for the managed objects and can be accessed by any supported browser at http://<insert-UCSM-IP-address>/visore.html. The Visore browser allows read-only access to the run-time schema, exposing the XML API native names for objects, classes, and attributes. Visore provides insight into both the native names and the relationship between objects in the object hierarchy.
- "goUCS" is similar to an SDK for access via the XML API access. goUCS[25] makes it easy to create XML API–based automation scripts for any activities (deployment, monitoring, etc.). goUCS provides a simple framework for posting XML to the Cisco UCS Manager and allows administrators to create customized, parameterized scripts and wrappers to easily extend their goUCS script libraries.

Conclusion

Cisco UCS Manager provides a powerful policy-driven framework for data center management and configuration. The use of policy can greatly assist in formalizing, standardizing and enforcing business rules as applied to computing, network, and storage access.

As capacity grows within a domain, Cisco UCS Manager's policy-driven framework facilitates ease of growth, automation, and simplified management. As capacity grows beyond a single UCS domain, the Cisco UCS Central[26] product can manage multiple globally distributed Cisco UCS domains with thousands of servers, while providing global ID pools, global service profiles, global policy configuration (and enforcement), global inventory, and global firmware management. Cisco UCS Central becomes the tool of choice for enabling management of multiple domains across data centers and geographies.

[22] http://software.cisco.com/portal/pub/download/portal/select.html?&mdfid=283850978&flowid=25021&softwareid=284574017
[23] http://developer.cisco.com/web/unifiedcomputing/sdk
[24] http://developer.cisco.com/web/unifiedcomputing/ucsemulatordownload
[25] http://developer.cisco.com/web/unifiedcomputing/goucs
[26] http://www.cisco.com/en/US/products/ps12502/index.html

Policy-based automation is the ultimate best practice, where use of the UCSM GUI diminishes over time. Data center designers, operators, and administrators are encouraged to explore all paths that lead to greater automation. Here are some general guidelines:

- Leverage the integration work already done by Cisco and its Partner Ecosystem[27] in the area of data center automation.
- Use “ConvertTo-UcsCmdlet” within the Cisco UCS PowerTool to capture configuration changes made through the UCSM GUI. Captured configuration changes are then displayed as corresponding PowerTool cmdlets.
- Focus on repetitive operational tasks. Use the XML API and SDKs to create parameterized scripts that can address and automate these common tasks.

Additional Resources

The following sites offer additional valuable resources around Cisco UCS:

- Cisco UCS Community Space: http://community.cisco.com/ucs
- Cisco UCS Support Forum Space: https://supportforums.cisco.com/community/netpro/data-center/unified-computing
- Cisco UCS White Papers and Technical Documents: http://www.cisco.com/en/US/prod/ps10265/ucs_white_paper.html
- Cisco UCS Advantage Video Library: http://www.cisco.com/en/US/prod/ps10265/ucs_advantage_video_library.html
- Cisco UCS and Data Center Validated Designs (CVD’s) : http://www.cisco.com/en/US/netsol/ns743/networking_solutions_program_home.html
- Cisco UCS Central Best Practice Guide: https://communities.cisco.com/docs/DOC-35264

About the Author

Jeff Silberman is a Data Center Architect and part of the original UCS Technical Marketing Team, who has been focused on Server and I/O virtualization for many years. Jeff has authored the original “UCS Best Practice/Quickstart Guide”, “UCS Central Best Practice Guide”, and the “UCS Deep Dive Methodology”. At Cisco, Jeff has been responsible for managing hundreds of customer proof of concepts, product reviews/demos, and technical "Deep Dives" with UCS, as well as presenting regularly at Cisco Live. Jeff came to Cisco through its acquisition of Topspin. Prior to Topspin, Jeff spent four years at NetApp in the Advanced Product Development Group, bringing some of the industry’s first Unified Fabric solutions to market for Oracle®/NetApp environments.

[27] http://www.cisco.com/en/US/prod/ps10265/ps10281/ucs_manager_ecosystem.html

Appendix: Cisco UCS Quick-Start Guide

To get a Cisco UCS system up and running as quickly as possible:

1. Cable the L1/L2 ports that connect the 2 FI's.
2. Allocate 3 IP addresses in the management/admin subnet for each FI and for the "Virtual IP."
3. Set hostname, IP addr, gateway, etc. from the Serial Console connection.
4. Set the Chassis Discovery Policy for the number of FEX->FI connections (1, 2 or 4), and the Link Grouping Preference.
5. Configure/Enable Server Ports; Configure/Enable Uplink Ports; Configure/Enable FC Ports.
6. Create Management IP Address Pool (typically same subnet as UCS Manager Admin IP).
7. Create "Host Firmware Policy" with packages from most recent UCS software bundle.
8. Create UUID Pool; Create MAC Pool; Create WWNN Pool; Create WWPN Pool (or populate the corresponding "default" pools). Embed domain ID's. Use Fabric-specific Pools for MAC and WWPN ("-A", "-B").
9. For SAN boot, create a unique "Boot Policy" for each storage array boot target.
10. Create VNIC templates ("eth0-A", "eth1-B"), that both draw from the above MAC Pool, and are associated with Fabric-A and Fabric-B respectively.
11. Create VHBA templates ("fc0-A", "fc1-B"), that both draw from the above WWPN Pool, and are associated with Fabric-A and Fabric-B respectively.
12. Create service-profile templates that draw from all earlier established pools, policies and templates, as appropriate.
13. Instantiate service-profile from template and associate service-profile to a given blade—**OR**—set service-profile template to associate with a particular Server Pool.
14. Configure PXE server, or map a bootable ISO image to the virtual-media CDROM drive to begin the OS installation.

Americas Headquarters
Cisco Systems, Inc.
San Jose, CA

Asia Pacific Headquarters
Cisco Systems (USA) Pte. Ltd.
Singapore

Europe Headquarters
Cisco Systems International BV Amsterdam,
The Netherlands

Cisco has more than 200 offices worldwide. Addresses, phone numbers, and fax numbers are listed on the Cisco Website at **www.cisco.com/go/offices.**

Printed in USA C11-697337-02 10/13