Introducing Cisco Unified Computing System

Learn Cisco UCS with Cisco UCSPE

Stuart Fordham

Apress®

Stuart Fordham

Introducing Cisco Unified Computing System

Learn Cisco UCS with Cisco UCSPE

Stuart Fordham
Bedfordshire, UK

ISBN 978-1-4842-8985-3 e-ISBN 978-1-4842-8986-0
https://doi.org/10.1007/978-1-4842-8986-0

with regard to jurisdictional claims in published maps and institutional affiliations.

This Apress imprint is published by the registered company APress Media, LLC, part of Springer Nature.
The registered company address is: 1 New York Plaza, New York, NY 10004, U.S.A.

To my family.

Introduction

This book is a guide on how to set up and configure a UCS (Unified Computing System). The beauty of it is that you don’t need to run out and buy one; you can use the UCS Platform Emulator running on a virtual machine on a laptop!

We start with setting up the virtual machine, and then look at how a real-life install would be completed. From there we will dive into the hardware that makes up a UCS system and then look at the networking requirements to make your UCS talk to the rest of the network. Once we have this in place, we move on to creating policies and profiles, which are needed by our servers. After this, we finish off with security and troubleshooting.

This book is intended for people looking to get an introduction to Cisco’s UCS.

Acknowledgments

Thanks as always to my family and my amazing team at work.

Table of Contents

About the Author

Stuart Fordham

, CCIE 49337, is the Network Manager and Infrastructure Team Leader for SmartCommunications SC Ltd, which is the only provider of a cloud-based, next-generation customer communications platform. Stuart has written a series of books on SD-WAN, BGP, MPLS, VPNs, and NAT, as well as a CCNA study guide and a Cisco ACI Cookbook. He lives in the UK with his wife and twin sons.

About the Technical Reviewer

Luca Berton

is an Ansible Automation Expert, who has been working with the Red Hat Ansible Engineer Team for three years. With more than 15 years of experience as a system administrator, he has strong expertise in infrastructure hardening and automation. An enthusiast of the open source, he supports the community by sharing his knowledge in different events of public access. Geek by nature, Linux by choice, Fedora of course.

S. Fordham, *Introducing Cisco Unified Computing System*
https://doi.org/10.1007/978-1-4842-8986-0_1

1. Setting Up UCSPE

Stuart Fordham[1] ✉
(1) Bedfordshire, UK

The Cisco UCS (Unified Computing System) is an extensive system in terms of size and cost. Because of this, it is difficult to get hands-on experience within the home. Thankfully, Cisco has released an emulator, UCSPE (UCS Platform Emulator), which runs happily on a laptop!

There are some limitations to UCSPE when compared to the whole system. We cannot install operating systems on the blades and rack mount computers, we cannot connect to other networking equipment, such as switches and SAN storage, and we cannot perform tasks such as setting up LDAP authentication.

These limitations will not stop us from having some fun though, as we can learn a lot about how to operate and maintain a UCS, without even having to leave the house!

So, let’s start by downloading and setting up UCSPE.

Setting Up UCSPE

UCSPE is a free download from Cisco.com. You will need a Cisco ID, so sign up if you do not have one already (https://id.cisco.com/signin/register). You can easily find UCSPE by searching for it on the main Cisco page (Figure 1-1).

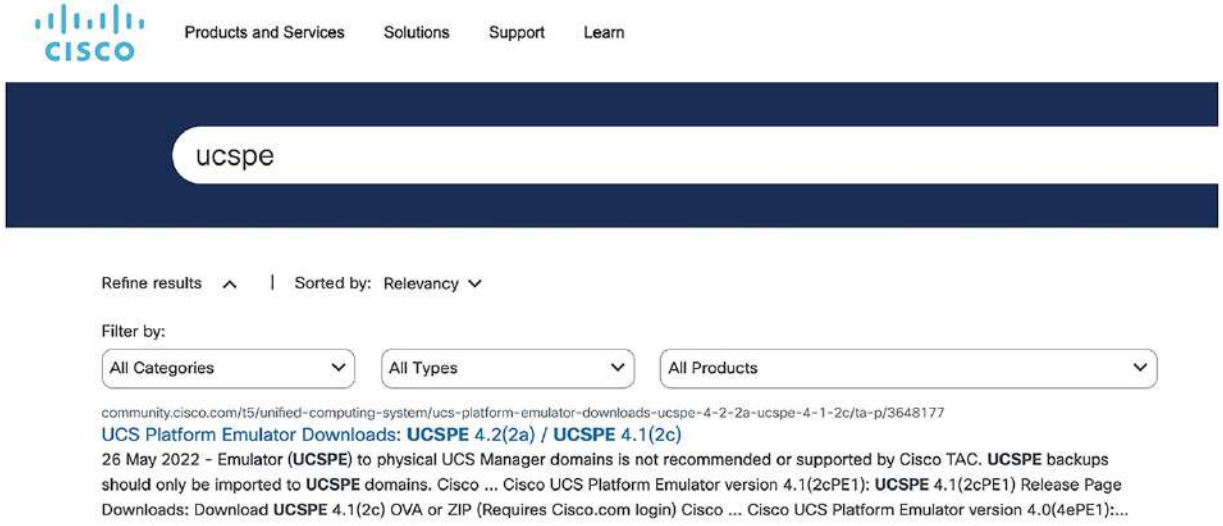

Figure 1-1 Searching for UCSPE

Click on the link (as shown in Figure 1-1) and you will be taken to the main page for UCSPE where you can click the link for the OVA and Zip file downloads (Figure 1-2).

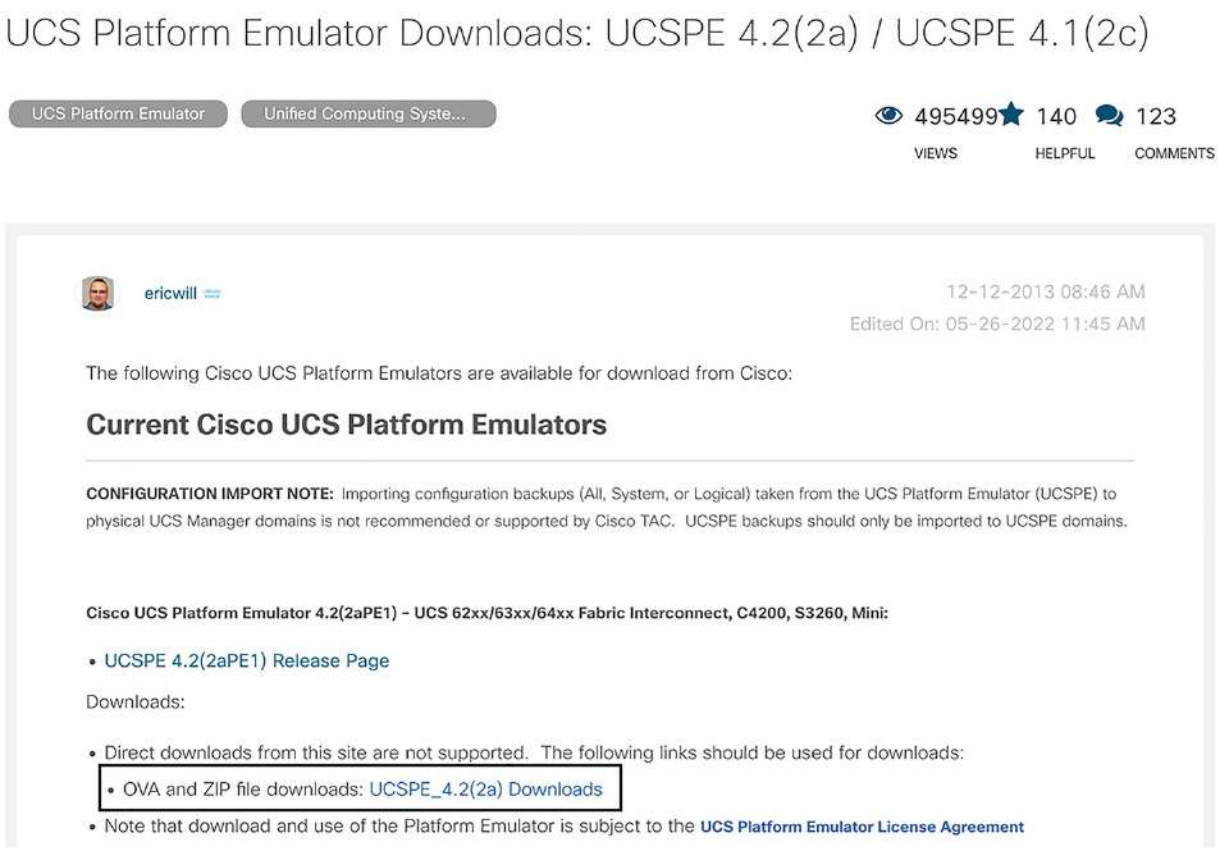

Figure 1-2 The UCSPE page

You will then be prompted to sign in using your Cisco ID. Once you have signed in, you can download the UCSPE software (Figure 1-3).

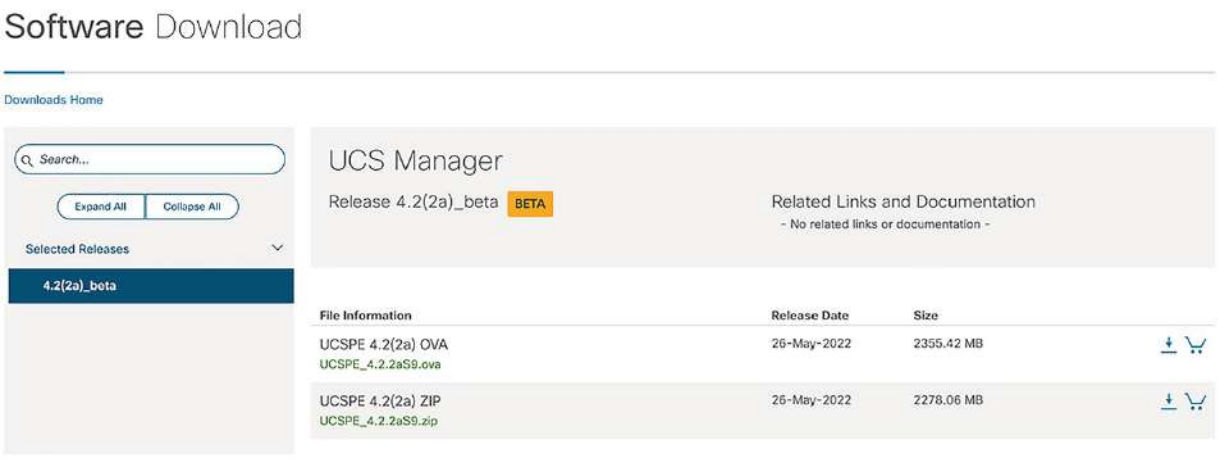

Figure 1-3 The UCSPE download options

The OVA file is the easiest option to use. You will need to accept the Cisco license agreement to download it. At the time of writing, the current version is 4.2(2a).

Importing UCSPE into VMWare

To run UCSPE, you will need the following available hardware:

- 1 CPU
- 2048MB memory (2GB)

This is hardly resource-intensive, so it should run happily on most modern computers. Our platform will primarily use VMWare (Fusion), but UCSPE will also run fine on VirtualBox or other hypervisors. Installing the VMWare software is not covered in the book.

Firstly, start VMWare, and select the option to import (Figure 1-4).

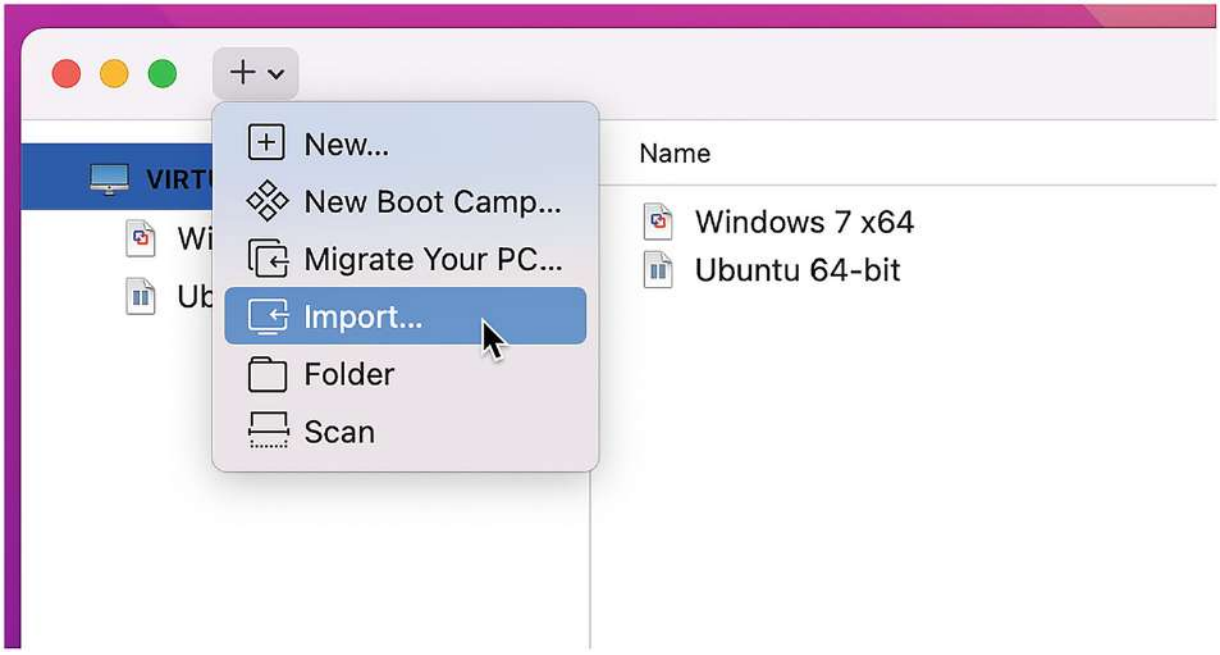

Figure 1-4 Importing into Fusion

In the window that pops up, click on “Choose File...” (Figure 1-5).

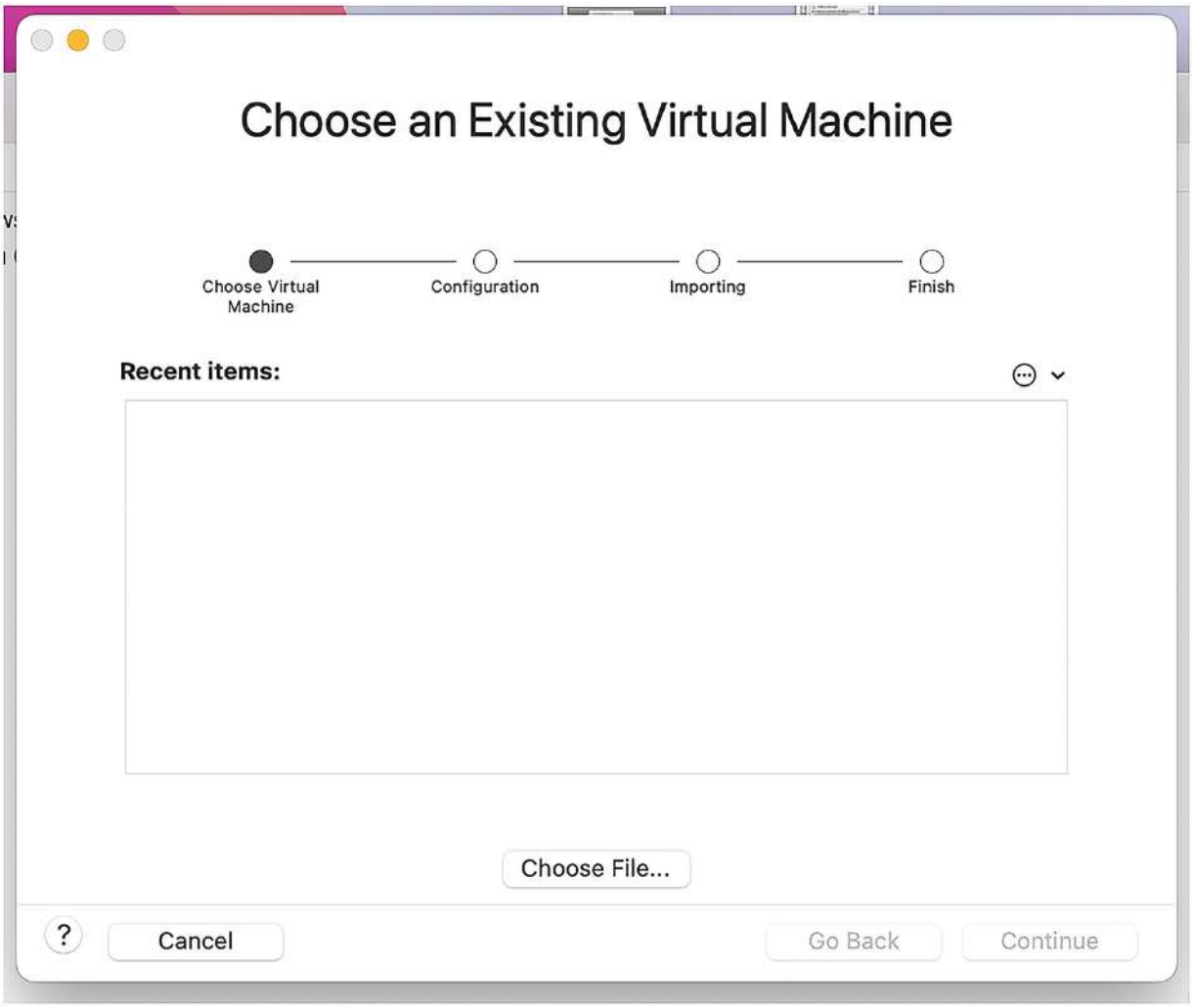

Figure 1-5 Select the file

Select the UCSPE file you downloaded earlier and click on “Open.” Click Continue again (Figure 1-6).

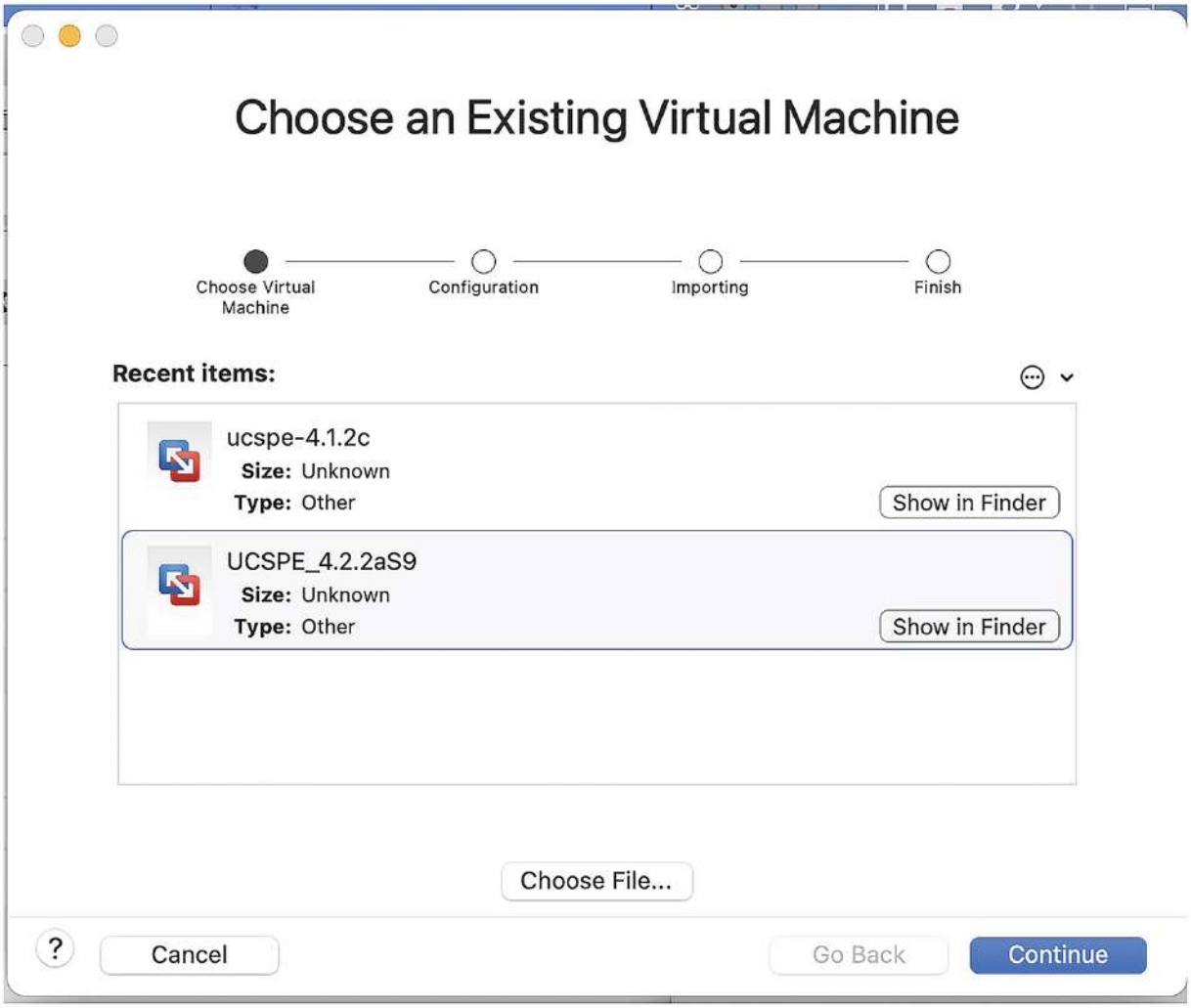

Figure 1-6 Click Continue

In the next window, you can rename the VM and select where to store it (Figure 1-7). Click “Save.”

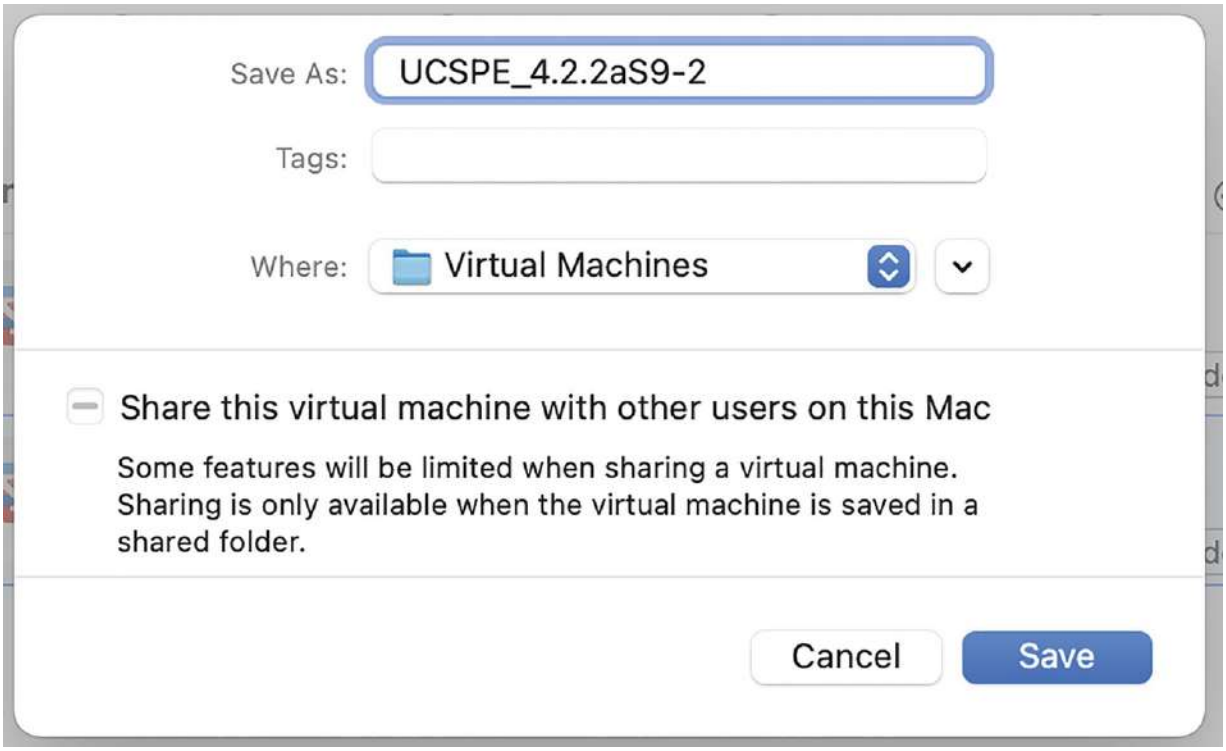

Figure 1-7 Save the virtual machine

The virtual machine will start to import into VMWare (Figure 1-8).

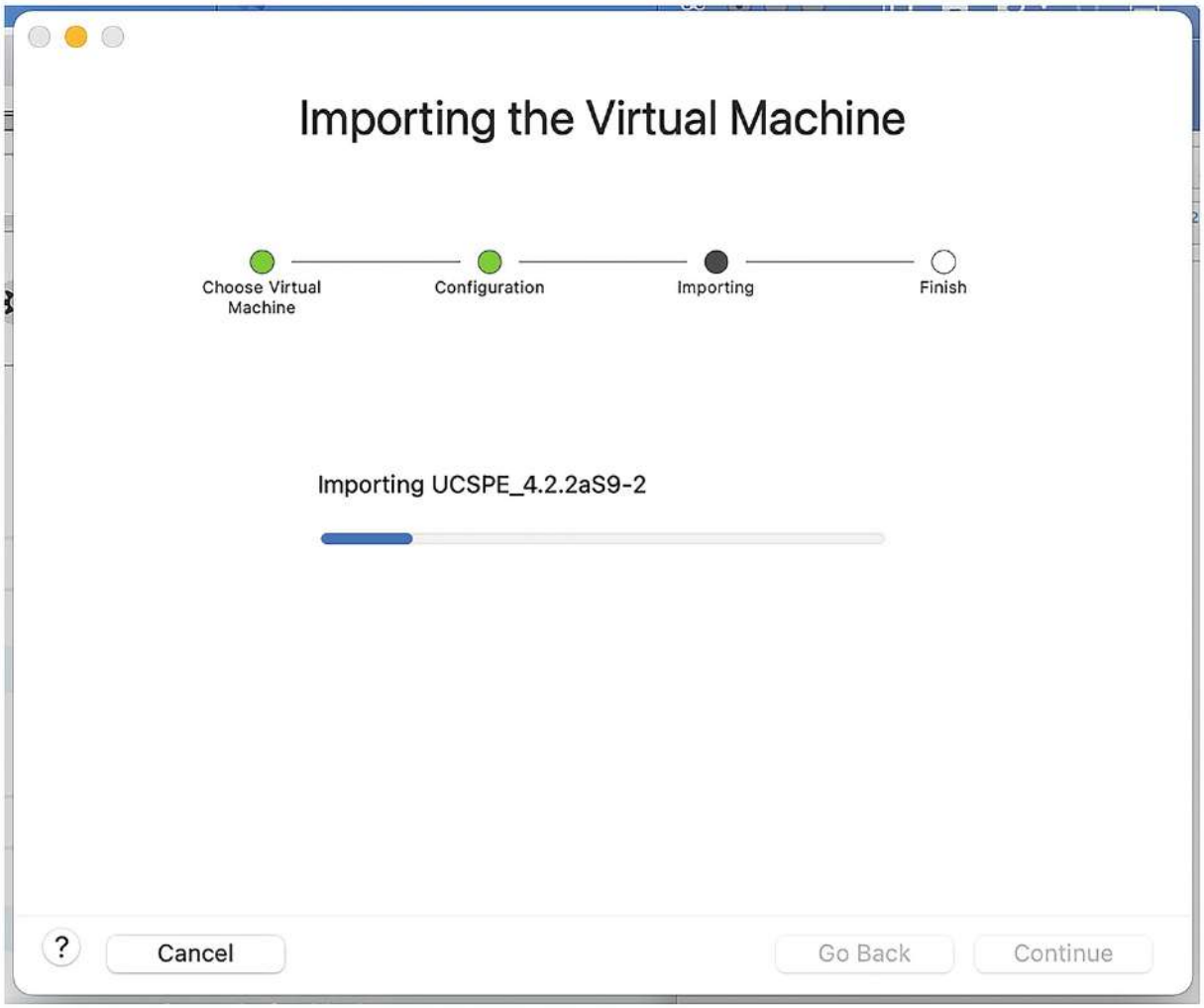

Figure 1-8 Complete the import

Once the import has been completed, you can customize the settings or click Finish (Figure 1-9).

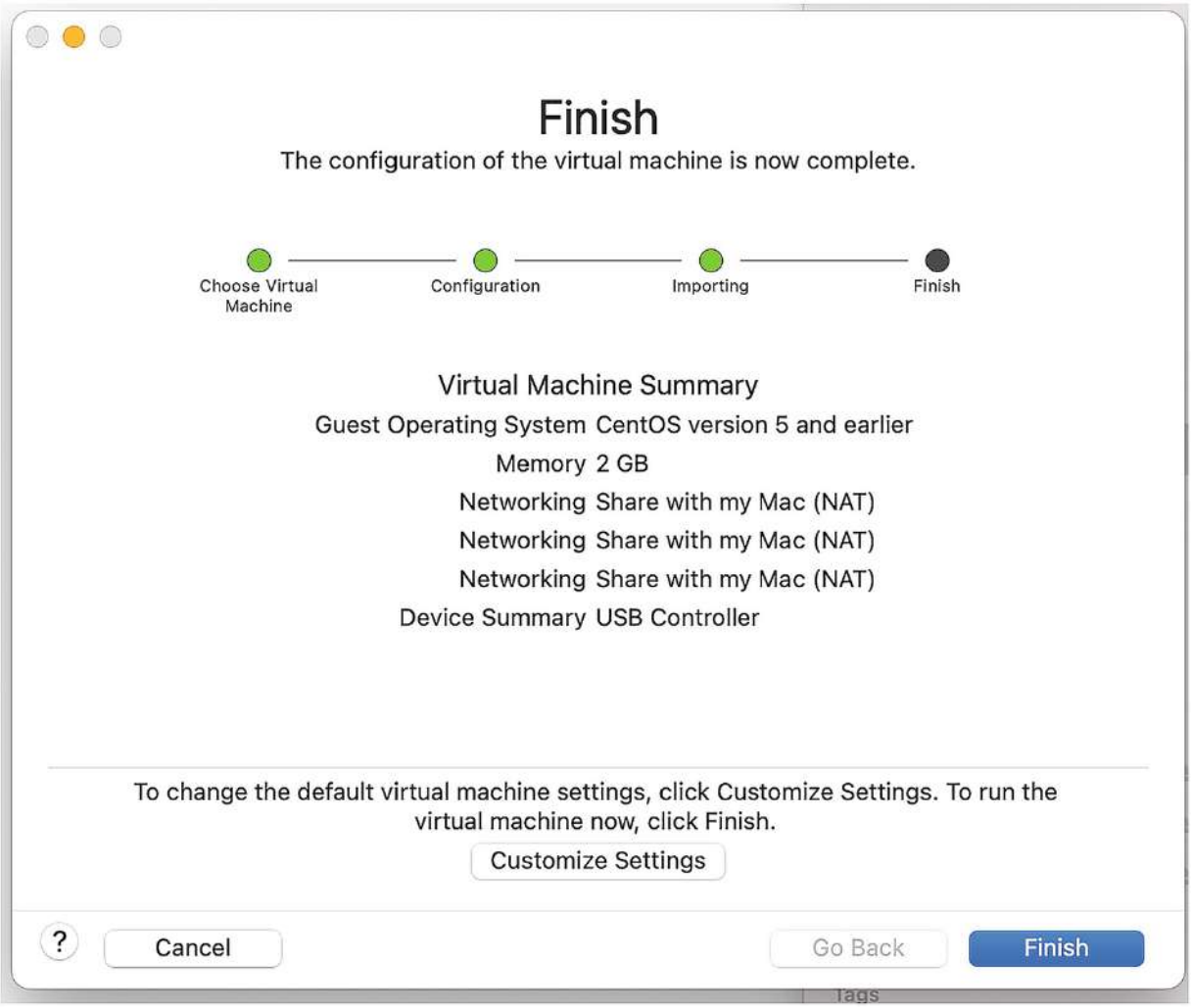

Figure 1-9 The final VM settings

Starting UCSPE

Start the UCSPE virtual machine by right-clicking on it and selecting Start.

When you start UCSPE for the first time, it runs through a quick installation and then starts the services, which can take a few minutes. Subsequent starts do not trigger an installation. The VM will then do some basic tasks, like using DHCP to gain an IP address (well, three actually; one for each of the Fabric Interconnects and one for the virtual IP, or VIP) (Figure 1-10).

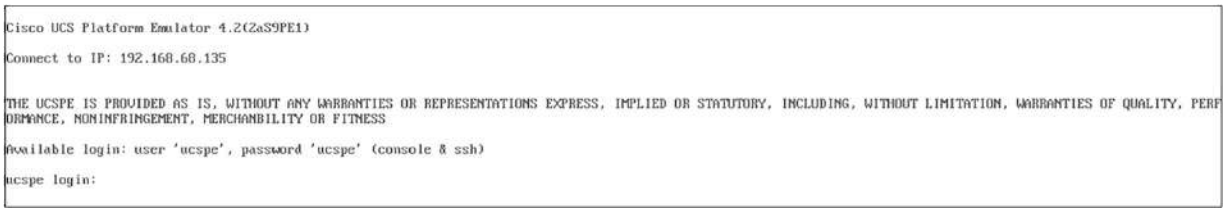

Figure 1-10 Starting UCSPE

In the next stage, UCSPE will complete the backend tasks, such as setting up SSH and GUI access, as well as generating the hardware catalog, which we will look at in the next chapter (Figure 1-11).

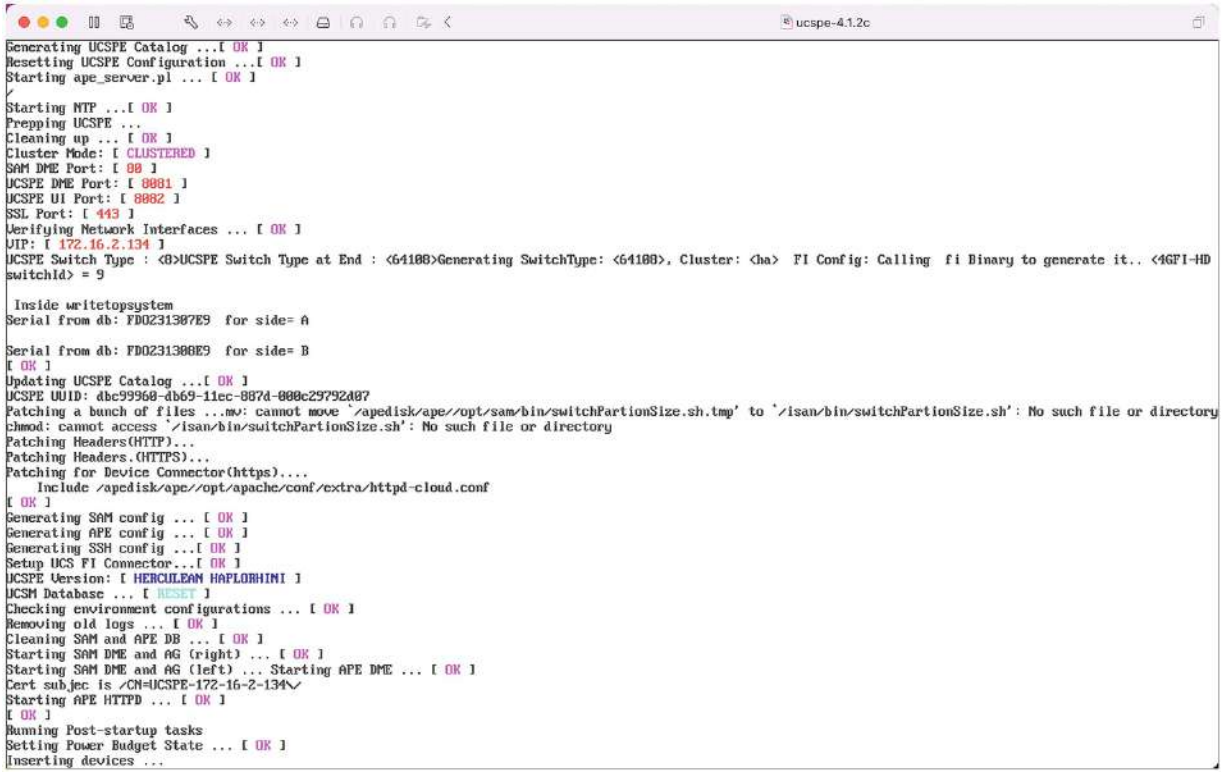

Figure 1-11 The VM is starting

Once everything has been completed, you will see the login details, which are the IP address of the VIP and the username and password (which are both "*ucspe*") (Figure 1-12).

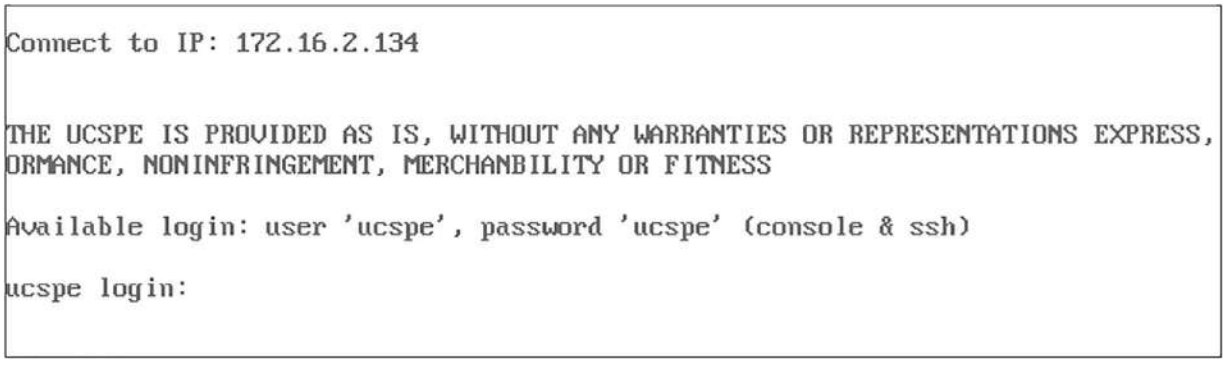

Figure 1-12 The login details

You can login and use the UCSPE VM console to show you more network details, the general status, and to

perform functions such as resetting the system, rebooting, or shutting down (Figure 1-13).

Figure 1-13 The UCSPE Console

You can now browse to the GUI using HTTPS (in this instance, it would be `https://172.16.2.134`) (Figure 1-14).

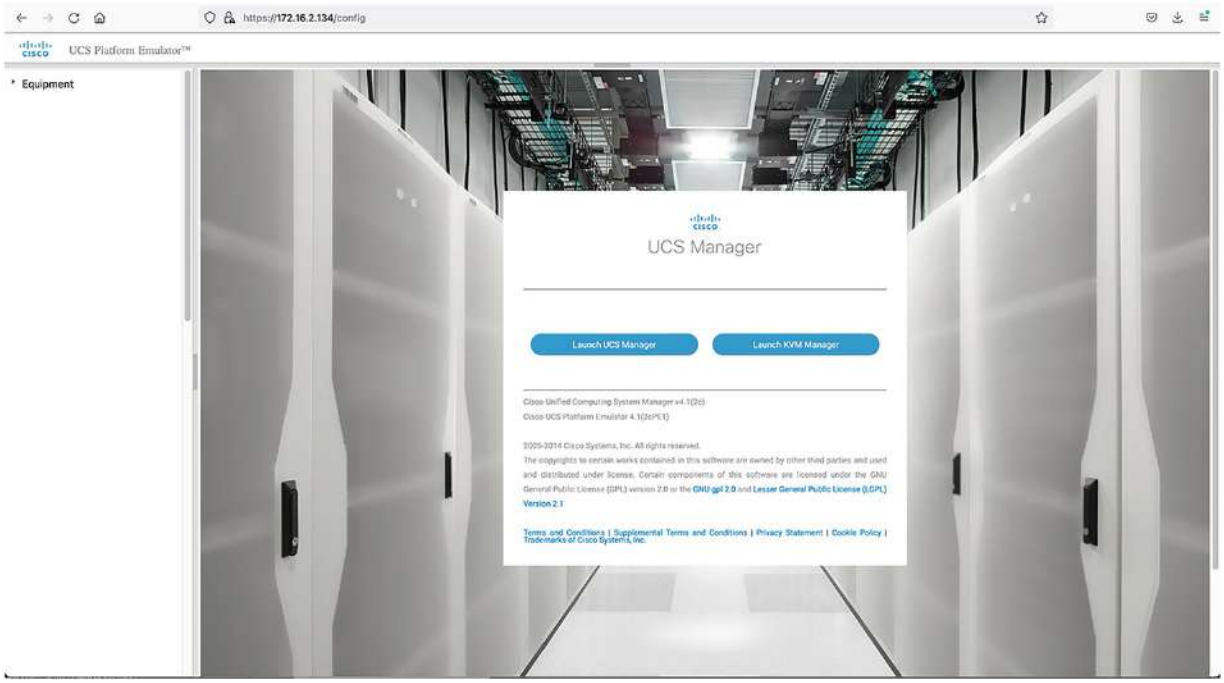

Figure 1-14 The UCSPE GUI

You can see that we have a list of our equipment on one side, and on the other we can log into UCS Manager (using the default username and password “ucspe”).

When UCSPE starts up, it will generate a fairly random environment, and that is where we will start in the next chapter after we look at how a UCS would be set up in the real world.

Real-World UCS Setup

If you are setting up a physical UCS, you will need to allocate three IP addresses; one for the management of each of the Fabric Interconnects (FICs), and one for the virtual IP (VIP) that will be used for the cluster management.

Once you have racked the FICs, we need to do the essential cabling for them, which will be management (LAN) interfaces that will connect to your upstream or management switches, and then we need to connect the cluster interfaces. These are the L1 and L2 interfaces, and we cable L1 on the first FIC to L1 on the other FIC and L2 to L2.

In this section, we will be configuring two FICs. FIC A will have the IP address 10.99.1.10, with a /24 subnet mask (255.255.255.0). FIC B will use the IP address 10.99.1.11. The VIP will be 10.99.1.200, and the default gateway will be 10.99.1.1. We will configure a DNS server IP address of 10.99.1.5. The cluster will be called “Mastering-UCS.”

Only power up the first FIC at first. We only power up the second one once the first FIC has been configured.

Connect your computer to the first FIC using a console cable, open your terminal software (like PuTTY or SecureCRT) and get on to the serial port using the following settings:

- 9600 baud
- 8 data bits
- No parity
- 1 stop bit

After connecting, you will see the following prompt:

```
—- Please read the following carefully —-
```

```
At the request of the publisher, I have had to
change
the output shown in the Cisco console, because
it is also shown on a website. Even though what
you would
Have seen would have come straight from the Cisco
device.
This was done to avoid any copyright issues.
So, I hope you understand that (for this chapter
only)
Things are going to get super weird.
Everything goes back to normal in the next chapter
though, so please bear with the incoming
strangeness, but they wouldn't listen to me.
```

We are going to use the console for our configuration, so type “console” and press enter.

```
Choose a config technique. (console/gui) ? console
```

Type in “*setup*” next, as we are not restoring from a backup.

```
Here you choose a setup mode; New or rebuild from
backup. (setup/restore) ? setup
```

This is a new Fabric Interconnect, so type “*Yes*” to start a fresh setup.

```
Thou hast elected to make a new Fabric
interconnect. Resume? (y/n): y
```

Type in a new password for the admin user when prompted. The complexity needs to be at least eight characters.

```
Choose a p@ssw0rd for "admin": Admin123
```

```
Type it in again "admin": Admin123
```

We will set up a cluster, so type in “yes” when prompted.

```
Might this interconnect device be half of a
cluster(select "ney" for lonesome-mode)? (yes/no)
[n]: yes
```

Because this is the primary switch that we are setting up, it will be the “A” of the pair. The secondary switch will be “B.”

```
Pop in the switch fabric (A/B) []: A
```

Next, we get to name our cluster. The name we chose will also be used on each Fabric Interconnect, with them getting either -A or -, depending on if they are the first FI, or the second.

```
Enter the system name: Mastering-UCS
```

Set the IP address, subnet, and gateway for the fabric interconnect.

```
Corporeal Switch Management0 IPv4 address :
10.99.1.10
Corporeal Switch Management0 IPv4 netmask :
255.255.255.0
IPv4 address of the gateway : 10.99.1.1
```

Now we type in the cluster IP address. It is this address that will be used by the primary Fabric Interconnect and is the IP you use when you fire up the UCS Manager.

```
Gathering IPv4 addy : 10.99.1.200
```

You can then configure your DNS servers. This is optional.

```
Wanna use a DNS Server? (yes/no) [n]: y
Pop in its numbers : 10.99.1.5
```

We are not going to configure a default domain name, so choose “no” for the next section.

```
Probably don't need a default domain name?
(yes/no) [n]:n
```

You will see a summary of the settings we have entered so far. Either type “*y*” to save the settings and restart the fabric interconnect, or select “*n*” to go back and make any modifications that you need to.

```
The following configuration will be pertained:

Switcharoo Role=A
UCS Designation=Mastering-UCS
Corporeal Switch Management0 IP Addy=10.99.1.10
Corporeal Switch Management0 IP
Netmask=255.255.255.0
Gateway=10.99.1.1
Nameserver=10.99.1.5
Gathering Enabled=yes
Gathering IP Addy=10.99.1.200

Save it and use this config (select "no" if you
want to re-enter)? (yes/no): yes

Sorting this out for you. Standby.
Configuration file – Ok
```

Our first fabric interconnect is now completed. Connect to the console on the second device and start it up.

From this point, it's just a matter of following the same steps.

```
—- Please read the following carefully —-
Still with us? Marvelous. Not much more to do now,
And we can get back to how things actually look
when
you configure a UCS. Again, I'd like to stress
that I did try
to keep everything as you'd see it on screen, but
it was a
case of either change it, or not get it published.
I hope you are well. Been to any good music gigs
recently? How's the family? I wish you a bright
and happy future.
Let's finish this off. Cheers.

Which do you prefer? (console/gui) ? console
```

The installation will detect the presence of Fabric Interconnect 1 (assuming you have connected the heartbeat ports together), and this switch will be added to the existing cluster. Type "*y*" to do this.

```
The setup is like a Jedi and has felt the aura of
another Fabric interconnect. Wanna add this switch
to the cluster? Carry on (y/n) ? y

Pop in the admin p@ssw0rd of the other Switcharoo
thing: Admin123
Attaching to the other switch… finito
Stealing info from other switch… finito
Other switch Management0 IP Addy: 10.99.1.10
Other switch Management0 subnet: 255.255.255.0
Gathering IP addy : 10.99.1.200
```

We now set the IP address for this fabric interconnect.

```
Corporeal Switch Management0 IPv4 thing :
10.99.1.11
```

Save the configuration.

```
Wanna keep the config (type "negative" if you
don't like it)? (yes/no): yes
Saving config. Give it a second or three.
Finally, we're done – Thanks
```

Navigate to `https://10.99.1.200` using a browser to log into the UCS.

Summary

We started this chapter by downloading the latest version of UCSPE from the Cisco website, importing it into VMWare, and starting it up. We then compared this to how a real UCS installation works in a clustered environment.

In the next chapter, we will configure UCSPE according to a topology (instead of the random assortment of hardware that it generates) and, as we do this, we will look in greater depth at the various components of a UCS environment.

S. Fordham, *Introducing Cisco Unified Computing System*
https://doi.org/10.1007/978-1-4842-8986-0_2

2. The UCS Components

Stuart Fordham[1]
(1) Bedfordshire, UK

When UCSPE starts up, it generates a new inventory. This comprises two chassis with five blades, one enclosure with two nodes, two fex, and ten rack servers. While the naming of the devices will vary between what you may see on your screen and the screenshots in this book, any differences should be minor.

We can see the equipment that has been created for us by clicking on the Equipment link on the left-hand side.

Managing UCSPE Hardware
Adding and Removing Devices

If you want to edit the auto-generated hardware that UCSPE has provided, then you can remove and add devices. You don’t have to remove (or add anything) here, this is more for reference if you want to create your own setup. However, it does give us an excellent, logical, way of introducing all the different components of the UCS and how they are connected. If you do go ahead and follow the following steps, then you might want to do a factory reset on the VM before the next chapter, which will set us up with a brand new UCS again before we move on to the next chapter.

Removing UCS Devices

Before we can remove a piece of hardware, we need to disconnect it first. The easiest way to do this is to click on the broken chain-link icon at the top right-hand corner (Figure 2-1). This will disconnect all the devices (apart from the Fabric Interconnects).

Figure 2-1 Disconnecting the devices

We can disconnect individual devices by clicking on the red circle next to the line item we want to remove if we

only want to remove one piece of equipment.

If you have chosen to remove all of them, once they change their green circle to a red one, click on the red circle at the top (Figure 2-2).

Figure 2-2 Removing the devices

This will remove the devices. If you find that (after waiting a few minutes) nothing has changed, then click on the trashcan icon on the device line item to delete them one by one. Refreshing the page is also useful here, as is clicking on the Equipment link, to update any changes that have been made.

The only devices that are left will be the fabric interconnects (Figure 2-3).

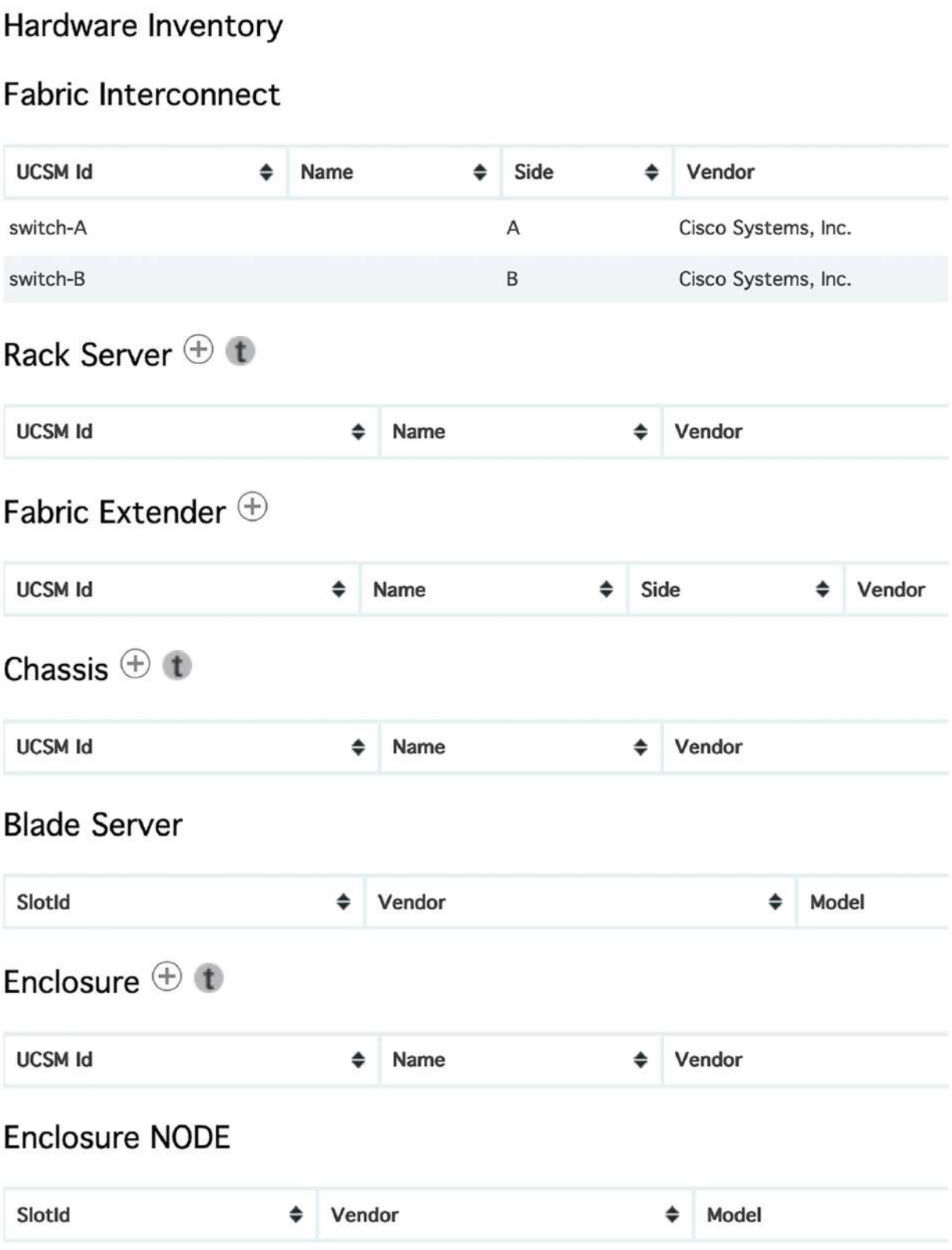

Figure 2-3 The hardware inventory

If the devices do not appear to change state, then click the Equipment link on the left-hand side and the screen should refresh. You may have to do this a lot with UCSPE as it can be slow to pick up changes, such as when we come to add hardware shortly.

Now that we have an empty canvas (so to speak) we can start with the Fabric Interconnects.

Fabric Interconnects

The Fabric Interconnects (also referred to as “FICs” or “FIs.” FI is better to use to avoid confusion, as FIC sounds much like FEX, which we will cover shortly) is where all the magic happens. This is where we manage the UCS estate as this is where the UCS software is held.

Generally, we would have two Fabric Interconnects, though you may also encounter a UCS-Mini. The UCS-Mini can handle between two and fifteen servers (a maximum of eight blade servers and seven rack servers), and places the FI (the UCS 6324 model) within the chassis, rather than them being separate hardware.

The FI runs a version of the Cisco Nexus software providing northbound connectivity to the rest of the network as well as connectivity to storage.

We can change the FI model if we want to by clicking on the cog icon at the top (Figure 2-4).

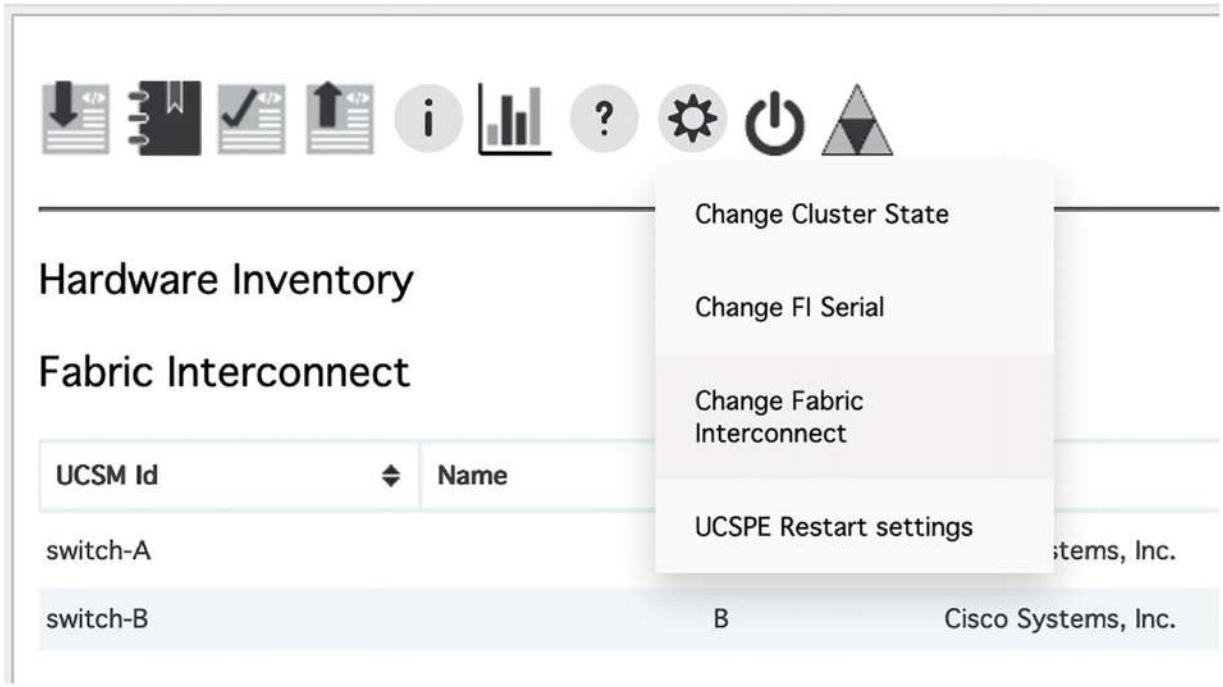

Figure 2-4 Changing the Fabric Interconnect

The available models are shown in Table 2-1.

Table 2-1 Fabric Interconnect models

Model	Size (RU)	100G ports	40/100G ports	40G ports	10/25G ports	10G ports	PSU	Fans
UCS-FI-M-6324	N/A	-	-	1	-	4	N/A	N/A
UCS-FI-6296UP	2	-	-	-	-	48+48[1]	2	2
UCS-FI-6332-16UP	1	-	-	24	-	16[2]	1+1	2+2
UCS-FI-6248UP	1	-	-	-	-	32+16[1]	2	1+1
UCS-FI-6454	1	54[3]	6	-	-	-	2	3+1
UCS-FI-64108	2	108[3]	12	-	96	-	2	2+1
UCS-FI-6332	1	-	-	32	-	-	1+1	2+2

If you do change the FI, then you will have to restart UCSPE (Figure 2-5).

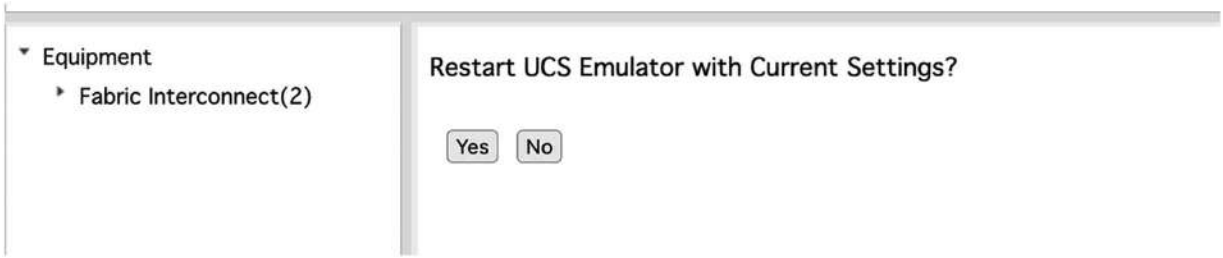

Figure 2-5 Restarting UCSPE

The interconnects come with two power supplies (PSUs) and four fans (Figure 2-6).

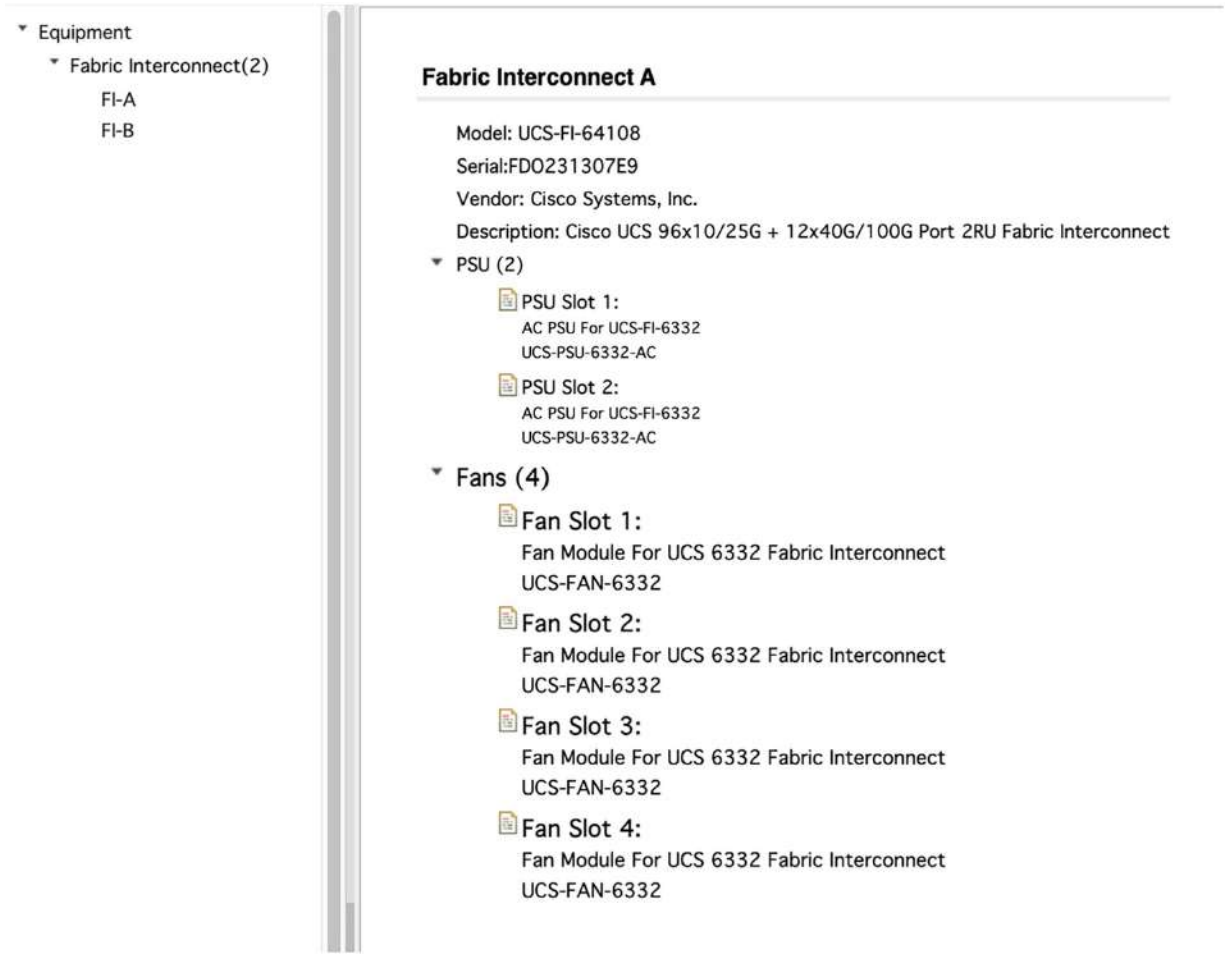

Figure 2-6 The Fabric Interconnects

Adding Devices

Next, we will come to our chassis.

Chassis

The chassis holds our blade servers. The chassis model options we have are

- UCSS-S3260 – a modular storage server with dual M5 server nodes.
- UCSC-C3X60 – similar to the S3260 but is now discontinued. Both are optimized for large datasets.
- UCSB-5108-DC.
- UCSB-5108-DC2.
- N20-C6508.
- UCSB-5108-AC2.

The 5108s are 8-slot 6RU chassis with two I/O bays. The N20-C6508 is the same as the previous, but is now discontinued.

You can add a chassis, such as the UCSB-5108-AC2, by clicking on the plus sign next to the word “Chassis” on the Equipment page. Enter the name for the chassis, select the model and click on “Add” (Figure 2-7).

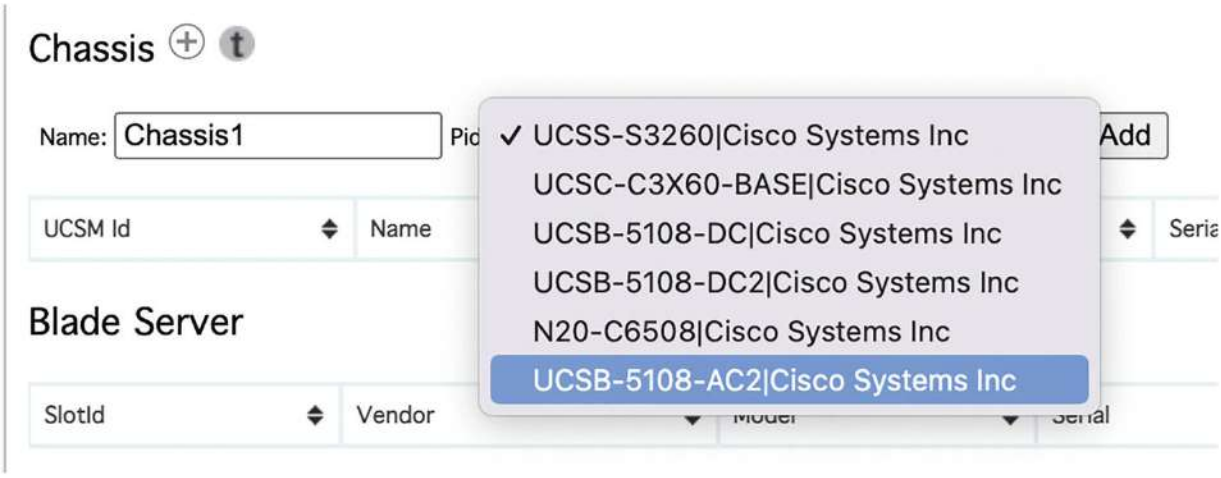

Figure 2-7 Adding a Chassis

The chassis will appear in our inventory on the left-hand side (Figure 2-8).

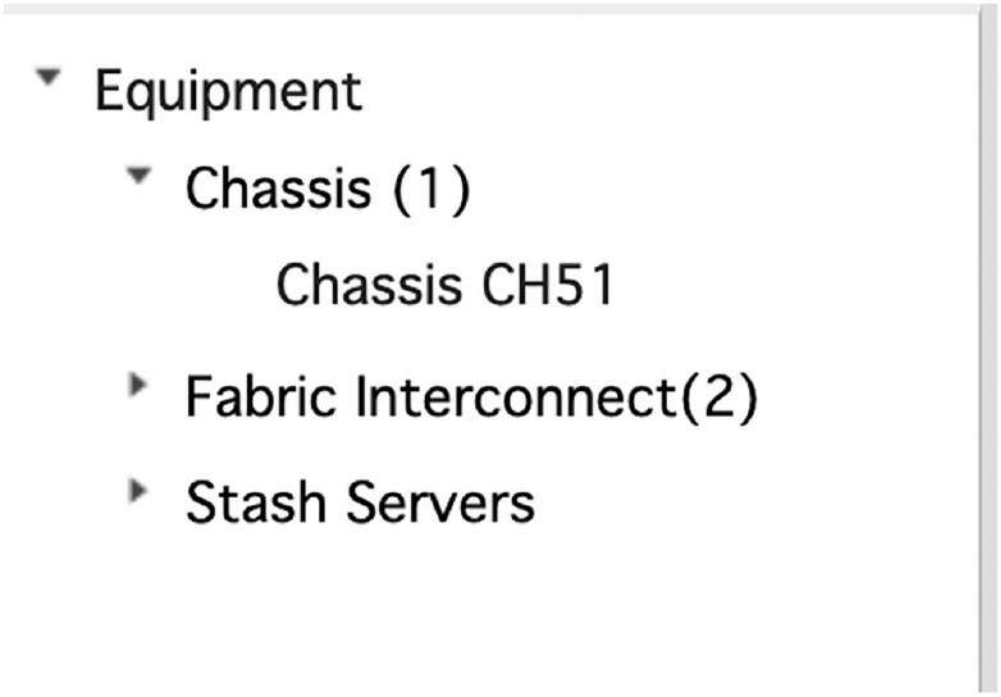

Figure 2-8 The new Chassis

Now that we have our first chassis, we need to fill it with the components that connect it to our FIs and make it hum gently[4] in the data center (power supplies and fans).

If we select the chassis and click on the edit button to the right on the item line, then we can see that we have many options of components we can add (Figure 2-9).

Figure 2-9 Chassis hardware

We are not going to add any blade servers at the moment, but we do need to add some power. We do this by clicking on “Psu” (not sure why Cisco didn’t capitalize all of

“PSU,” but there we are), selecting an appropriate model (such as the Platinum II AC power supply), and dragging it up to the chassis, above where the model is shown and underneath the plus and minus buttons (Figure 2-10).

Figure 2-10 Adding a PSU to a chassis

Once you let go, you can select how many to add. UCSPE will tell us how many available slots we have available to fill. We can decide which slot to add an item to by typing in the slot number (such as “1”) or a range, by typing in “1-4” and pressing enter. You should now have four power supplies (Figure 2-11):

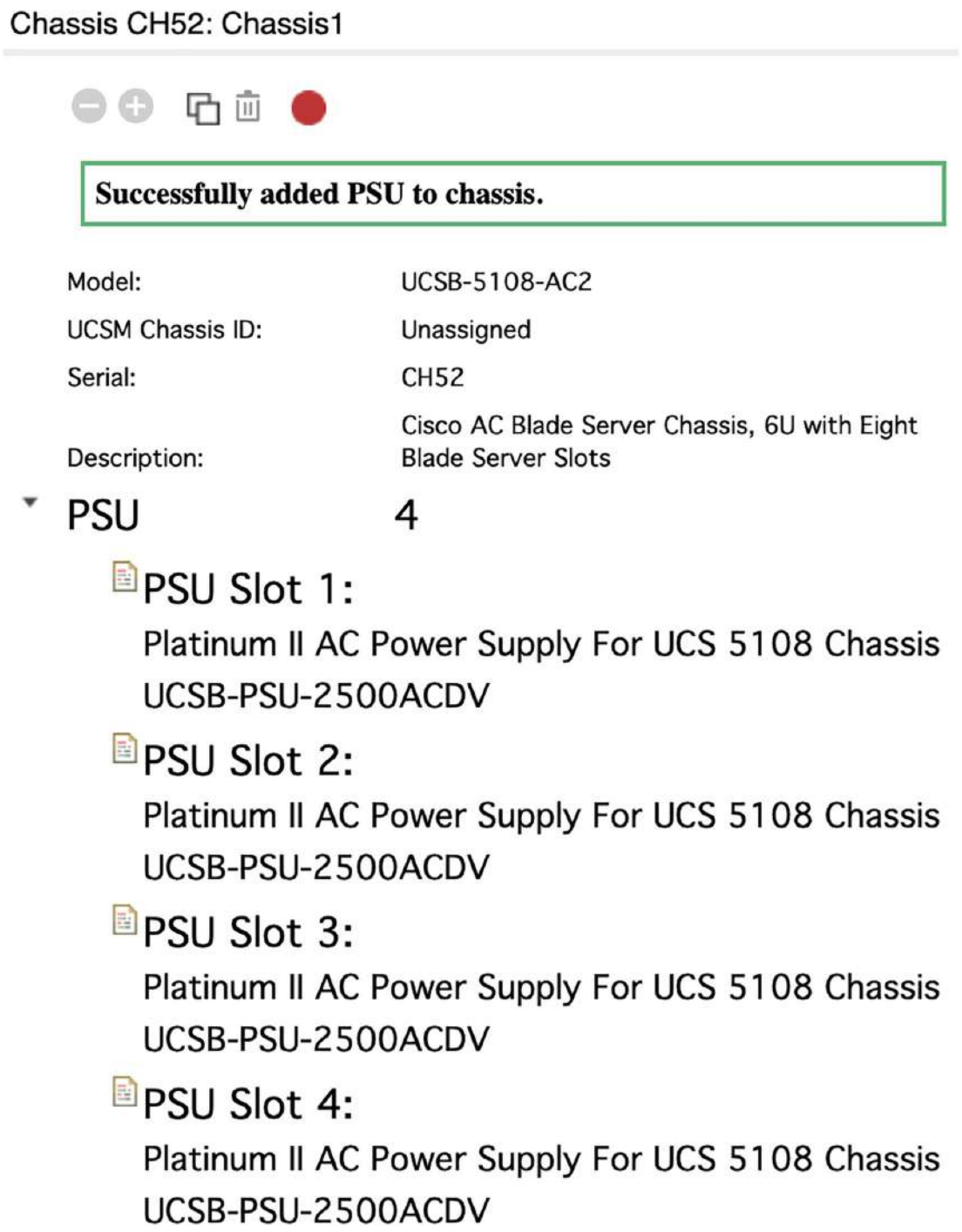

Figure 2-11 Our Chassis has power!

Chassis also require fans and we add these in the same manner, by clicking on the Fan link next and adding eight fans (Figure 2-12). In the box, you can type “1-8” to add all eight fans in one go.

Chassis CH52: Chassis1

Successfully added Fan to chassis.

Model:	UCSB-5108-AC2
UCSM Chassis ID:	Unassigned
Serial:	CH52
Description:	Cisco AC Blade Server Chassis, 6U with Eight Blade Server Slots

▼ Fans 8

- Fan Slot 1:
  Fan Module For UCS 5108 Blade Server Chassis
  N20-FAN5
- Fan Slot 2:
  Fan Module For UCS 5108 Blade Server Chassis
  N20-FAN5
- Fan Slot 3:
  Fan Module For UCS 5108 Blade Server Chassis
  N20-FAN5 ✕
- Fan Slot 4:
  Fan Module For UCS 5108 Blade Server Chassis
  N20-FAN5
- Fan Slot 5:
  Fan Module For UCS 5108 Blade Server Chassis
  N20-FAN5
- Fan Slot 6:
  Fan Module For UCS 5108 Blade Server Chassis
  N20-FAN5
- Fan Slot 7:
  Fan Module For UCS 5108 Blade Server Chassis
  N20-FAN5
- Fan Slot 8:
  Fan Module For UCS 5108 Blade Server Chassis
  N20-FAN5

▶ PSU 4

Figure 2-12 Adding chassis fans

Next, we can add the IOMs. The IOMs are “In/Out Modules.” These are also known as FEXs. They are the line cards that connect our chassis to our fabric interconnects. They also provide the interface connections to the blade servers, and CMC (Chassis Management Controller), which is used for monitoring our components, such as fans, power supplies, and temperatures and this is also the component that is responsible for monitoring blade insertion and removal. Lastly, they also provide Chassis Management Switch (CMS), which gives us the KVM (Keyboard, Video, Mouse), Serial over LAN (SoL), and Intelligent Platform Management Interface (IPMI) abilities to our blades.

The IOM options we have are

Model	Fabric ports	Server ports	Throughput
2304	4x40GE	8x40Gbps	320Gbps[5]
2208XP	8x10Gbps	32x10Gbps	80Gbps
2204XP	4x10Gbps	16x10Gbps	40Gbps
2408	8x25GE	32-10Gbps	400Gbps[5]

If we add two 2408 IOMs and click the word “Equipment” in the left-hand pane, then we should see the chassis change, listing the (currently disconnected) IOM ports (Figure 2-13).

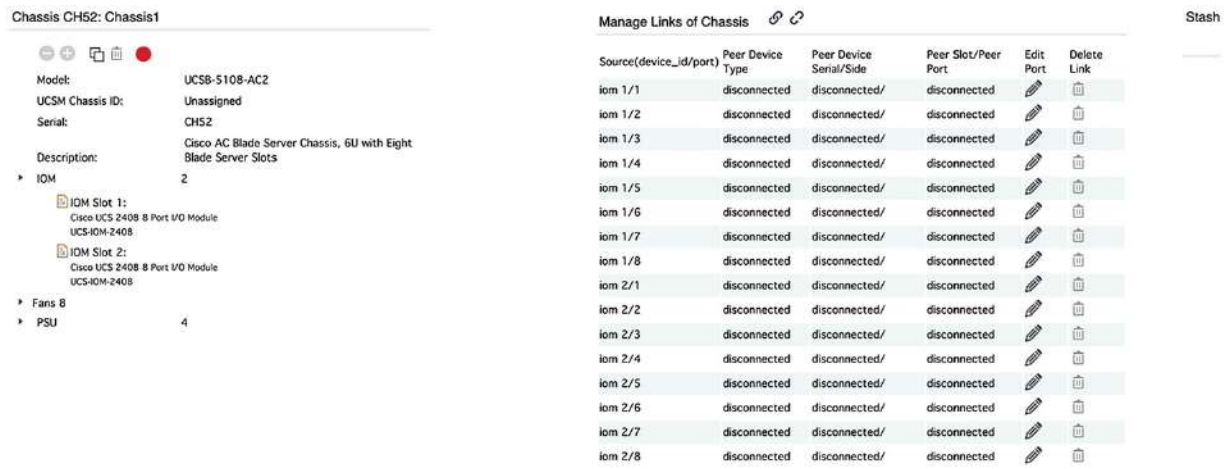

Figure 2-13 IOMs in our Chassis

We are not going to configure the IOMs just yet, instead, we are going to see how we can quickly create two more. If we click back onto the main Equipment list, then we can click on the duplicate icon next to our chassis, and again to create two more chassis.

We should have three chassis now (Figure 2-14).

Figure 2-14 Three Chassis

This ability to duplicate can save a lot of time if we need to add multiples of the same hardware component and is especially useful when adding servers.

We can now connect our chassis to our Fabric Interconnects. We can do this by editing the first chassis and clicking the pencil icon under “Edit Port” on port 1/1 (IOM 1, port 1). The Peer Device Type needs to be set to “fi.” Select FI A and select a free port (such as 1/20). Repeat the process for iom 1/2, selecting the next available port on the same FI (1/21).

Next, edit iom 2/1 (IOM 2, port 1) selecting FI B, and the same port number as used on iom 1/1 (port 1/20), and repeat for 2/2, selecting the next port (1/21) (Figure 2-15).

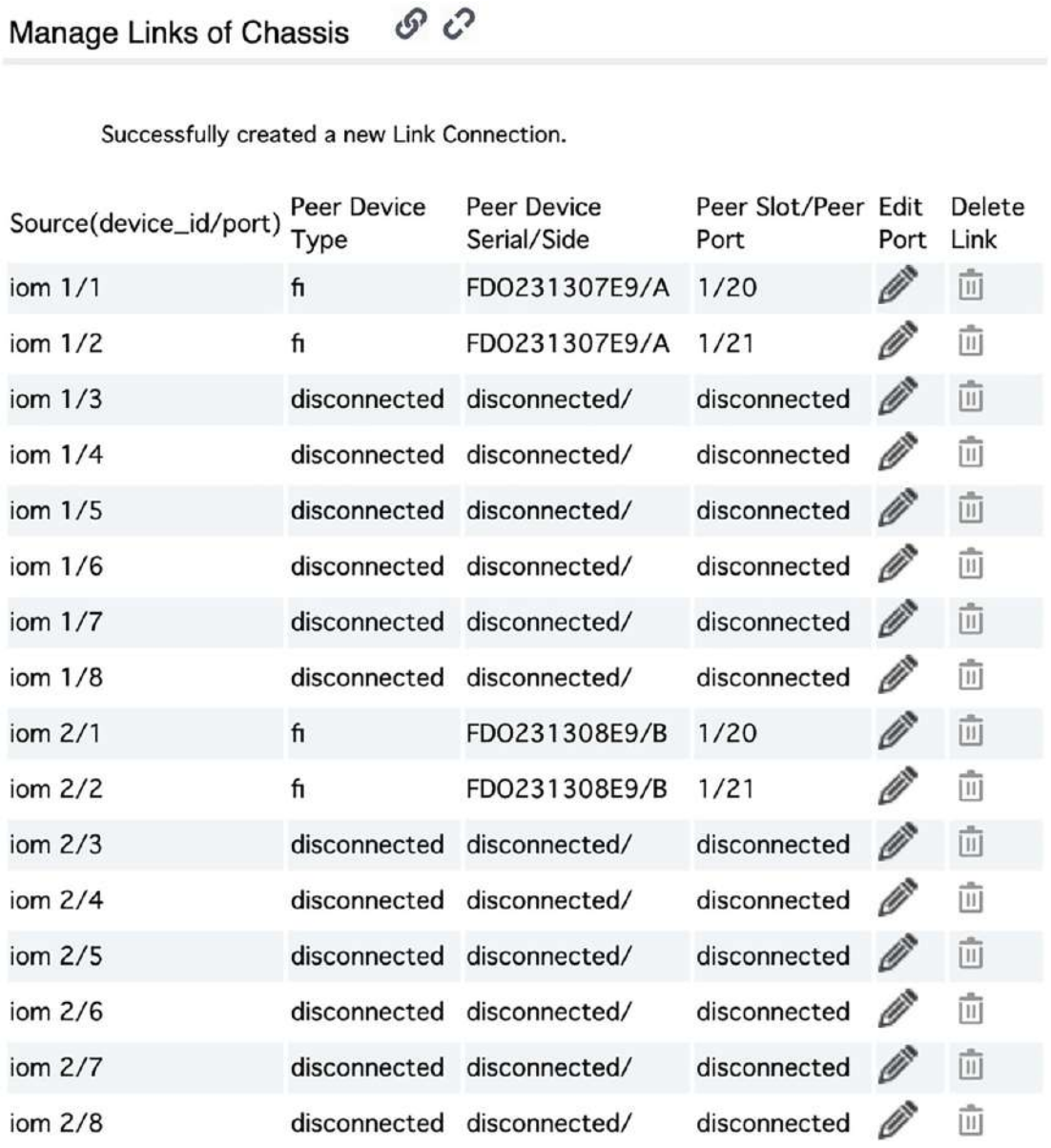

Figure 2-15 Chassis 1 IOM connectivity.

Repeat the process on the other chassis, following Table 2-2.

Table 2-2 IOM connectivity

Chassis	IOM	IOM Port	Peer Device	Peer Port
2	1	1/1	FI A	1/24
2	1	1/2	FI A	1/25
2	2	2/1	FI B	1/24
2	2	2/2	FI B	1/25
3	1	1/1	FI A	1/28
3	1	1/2	FI A	1/29
3	2	2/1	FI B	1/28
3	2	2/2	FI B	1/29

The reason we leave gaps between the ports of one IOM and the ports of another (as they go into the FI) is that if we want to increase bandwidth later on, we can keep things nice and neat and ordered. Also, bear in mind that the cabling is one IOM to one FI. The IOM, essentially, becomes part of the FI, so we never cross the streams. This isn't Ghostbusters. Bad things really will happen. Maybe not end-of-the-world type stuff, but certainly a call to Cisco TAC (Technical Assistance Center)!

If you were to start UCS now, this is what your systems would look like (Figure 2-16).

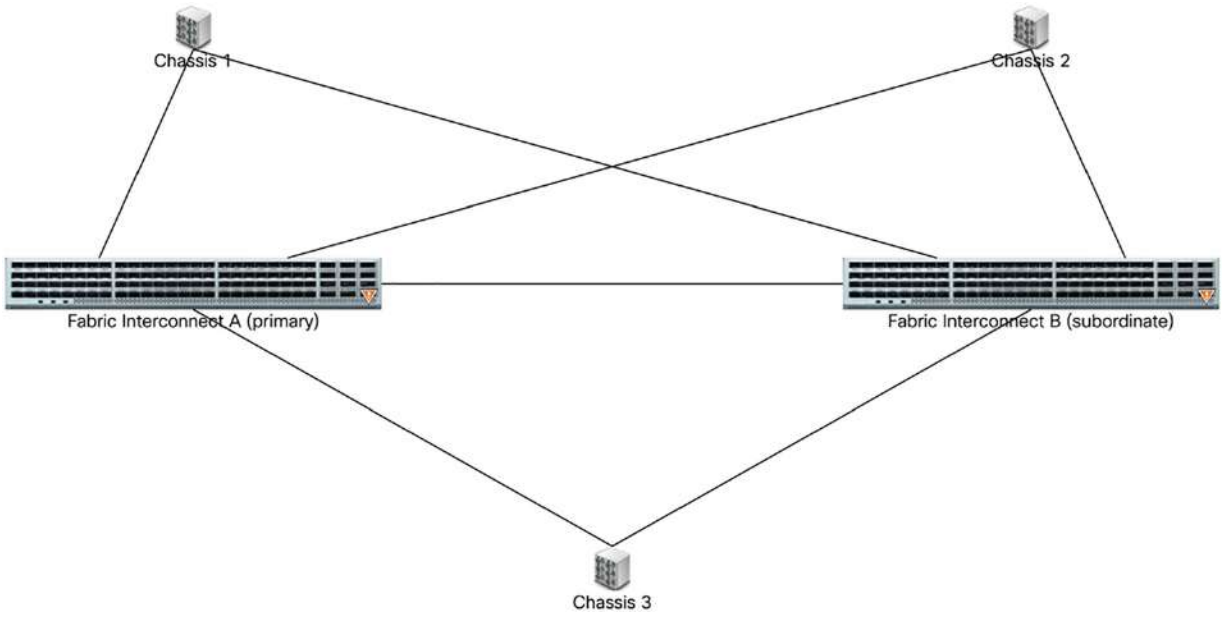

Figure 2-16 Our topology so far

Blade Servers

Our chassis are fairly useless if we have no (B-series) servers to run in them. So, add some servers by dragging the server from the menu at the bottom into the chassis. Each server will need CPU, memory, and storage, so add these as well. When adding servers, plan them out carefully. For example, if you have three chassis and servers that will perform different functions (such as ESXi hypervisors, database servers, application servers, and so on), share these out across all three chassis so that if one chassis has an issue, the servers in the other chassis can continue to server your data an environment as required.

Once you have added your servers, click on the red button next to each of the chassis (to remove them) and then click the green button to insert them again. The red button should turn green, as well as the green button next to each of the servers. You may need to wait a few minutes before you can insert it again.

Our chassis will look something like this (Figure 2-17):

Figure 2-17 Our chassis

Now that we have some blade servers, we should add some rack servers. Before we do this, however, we are going to add some FEXs.

FEX

FEX stands for “Fabric Extender.” These allow us to increase the number of ports we have at our disposal. The options we have in UCSPE are

Model	Server ports	Uplinks
N2K-C2232TM-E-10GE	32x 1/10GBASE-T	8x10GE
N2K-C2232TM-10GE	32x 1/10GBASE-T	8x10GE
N2K-C2148T-1GE	48x 1G Base-T	4x SFP+
N2K-C2232PP-10GE	32x 1/10GE SFP/SFP+	8x10GE
N2K-C2348UPQ-10GE	48x 1/10 Gigabit Ethernet (SFP/SFP+)	6x 40GB
N9K-C93180YC-FX3	48x 1/10/25-GBps fiber	6x 40/100GB

Rack Servers

In UCSPE, the rack servers (C-series) connect to the FEXs and there is a wide variety of servers to choose from. Way too many to list here with all their differences, but similarly to the blade servers, you will need to add CPU, memory, disks, I/O adapters, storage controllers, and PSUs.

When we connect rack servers, we need to consider how we connect them; we have options of “Direct Attach Server,” “Single Wire Management,” or “Dual Wire Management.”

Direct Attach Mode

In Direct Attach mode (available with UCS version 3.1 or later), the servers attach directly to the fabric interconnects, bypassing the need to have FEXs.

Single Wire Management

As the name suggests, single wire management uses a single cable into the FEX for management and data traffic.

Dual Wire Management

In dual wire mode, separate cables are used for data and management.

Enclosures

UCS enclosures, such as the UCSC-C4200-SFF can host up to four “nodes,” such as the C125. These are designed for dense compute form factor with high core densities, where the ability to scale out with compute-intensive machines is critical.

We do need to add fans and power to the enclosures, but not IOMs. We can add the nodes by dragging them onto the chassis as we do with the other hardware. Once we have added the nodes (as well as the CPU, memory, I/O adapters, and disks), we can connect the node to the FEX by clicking the chain icon where it says “Manage Links of Node.”

We can see how this all looks by clicking the Equipment link at the top left-hand corner and then clicking the UCS icon at the end of the row of icons. We can log in using the username and password of “ucspe.”

The default UCSPE-generated layout will look a little like this (Figure 2-18):

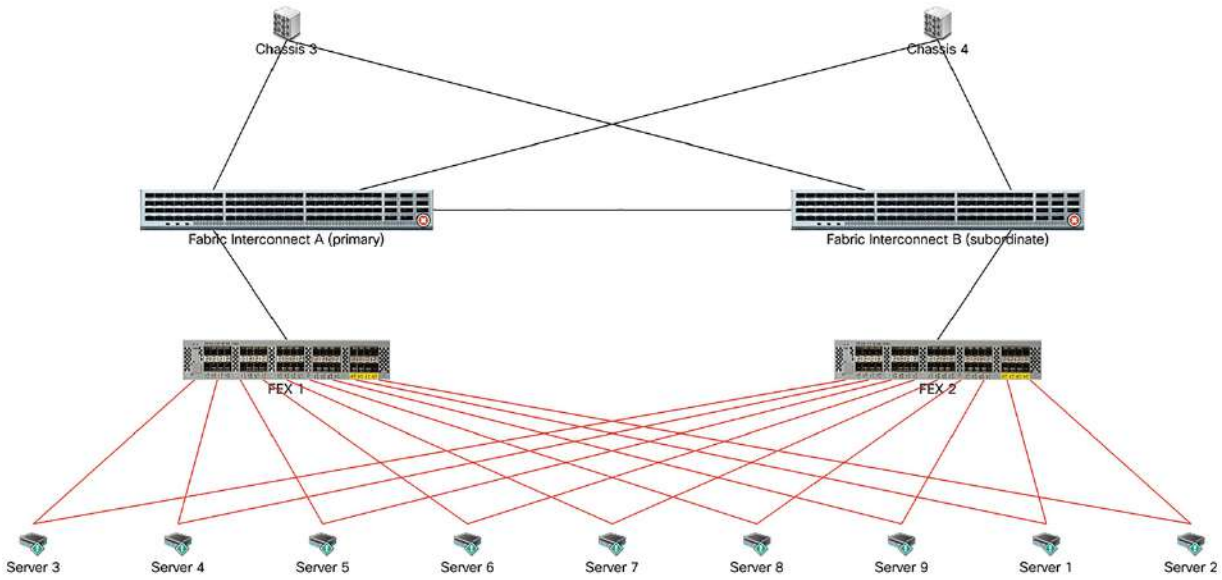

Figure 2-18 Our UCS environment

Summary

In this chapter, we looked at how to add and remove the components available to us in UCSPE, as well as how to connect them all together.

Footnotes

1 Using an expansion module

2 It is technically sixteen 1 and 10Gbps and FCoE, or 4-, 8- and 16-Gbps Fibre Channel unified ports

3 These are 10/25/40/100Gbps and FCoE ports

4 When I say “hum gently” I am joking. Without all four PSUs plugged in these things sound like a plane taking off!

5 Across two IOMs

S. Fordham, *Introducing Cisco Unified Computing System*
https://doi.org/10.1007/978-1-4842-8986-0_3

3. Northbound Networking and SAN

Stuart Fordham[1]
(1) Bedfordshire, UK

In the first two chapters, we set up UCSPE and looked at how to connect it physically, both to its own components and to the rest of the network. The physical cabling we looked at in the first chapter was, however, purely for management. This would allow us to control our UCS, but the blades and rack servers would have no connectivity to the rest of the network. In this chapter, we will be focusing on how to add the networking components that will let our UCS talk to the rest of the world.

UCS networking

At the moment, our UCS servers will be disconnected from the rest of the network, so we need to add means for them to pass packets to the rest of the network. We have a couple of ways to achieve this. We can use “Uplink” ports, or port-channels. We will start by looking at uplink ports.

Uplink ports

We start to configure uplink ports by going to the Equipment tab in UCS manager, scrolling down to the fabric interconnect, and then into the ports. Uplink ports can either be configured on the fixed module (fixed ports that are part of the FI) or on the expansion module (a module purchased separately and installed in the FI). We select the port we want to configure and then, from the “Reconfigure” menu, set it as an Uplink port (Figure 3-1).

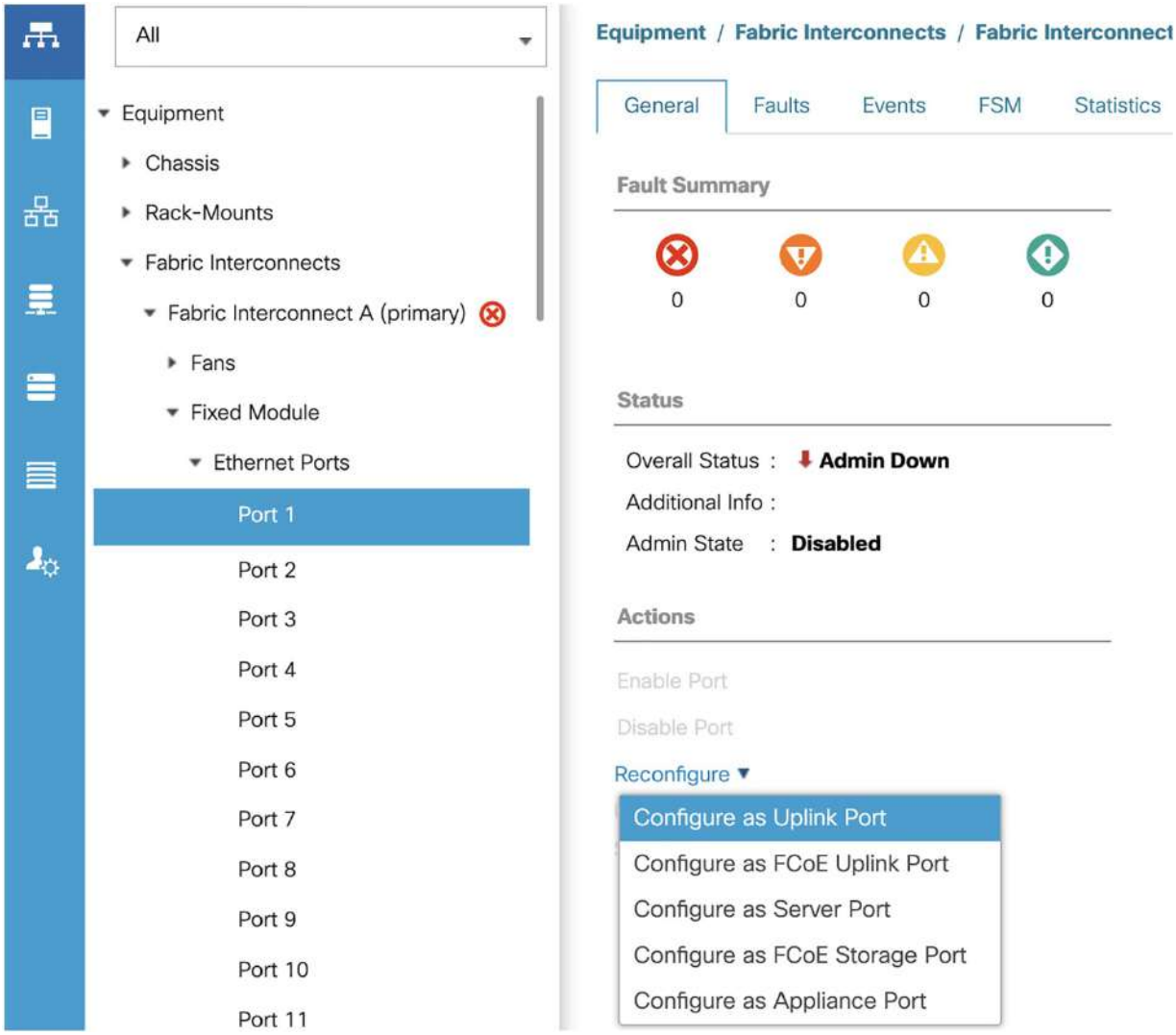

Figure 3-1 Reconfiguring a port as an Uplink port

We will then need to confirm that we do wish to reconfigure the port, and then acknowledge the action once completed.

The next step in creating our uplinks is to head into the network tab in UCS Manager, and select the LAN option from the left-hand side. We will be able to see our uplink interface listed under the relevant fabric ("Fabric A" for FI-A, "Fabric B" for FI-B) (Figure 3-2).

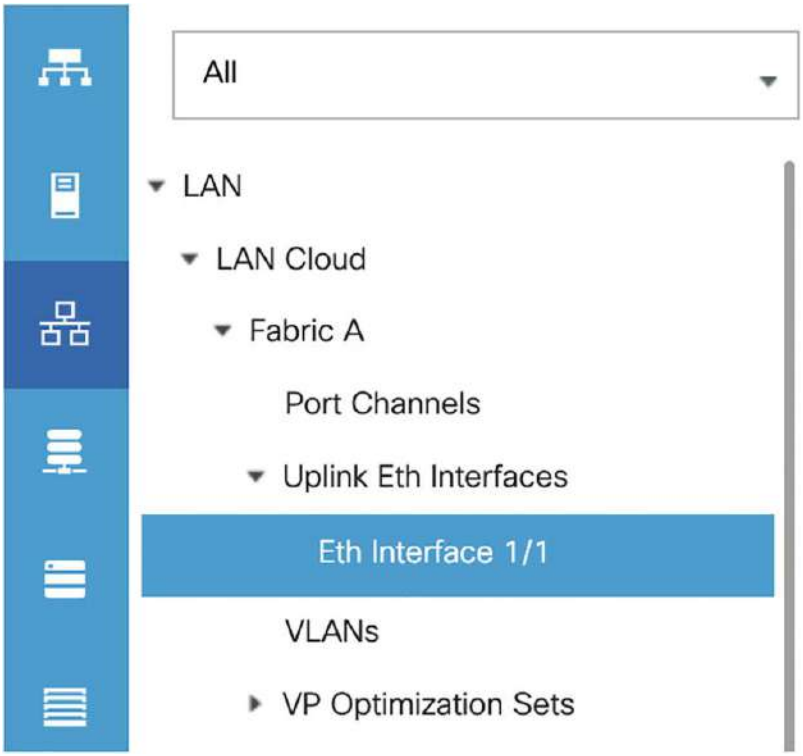

Figure 3-2 Our Uplink interface

On the other side of the page, we have options we can set for the interface (Figure 3-3). Firstly, we can give it a label, which can be helpful to quickly identify which network device and port we are connected to.

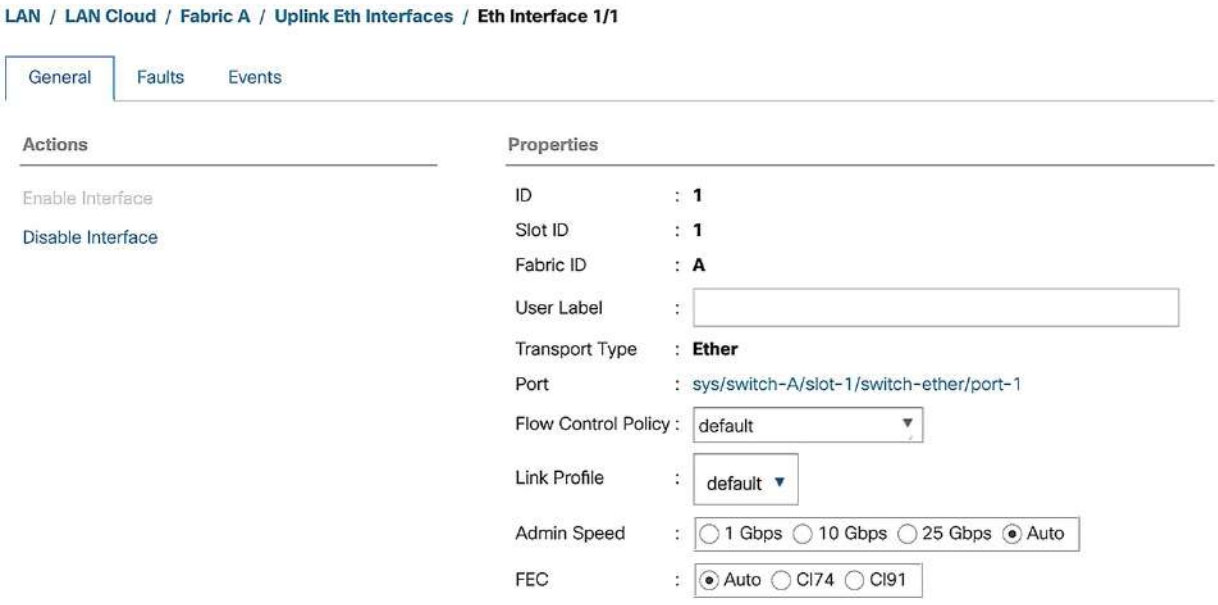

Figure 3-3 Uplink interface settings

Next, we can set a flow control policy, which controls how the port acts (in the sending and receiving of pause frames) when the receive buffer is full. We create a flow control policy by going to LAN ➤ Policies ➤ root ➤ Flow Control Policies and clicking “Add.” We give it a name, set the priority and Receive and Send to “on” (otherwise the flow-control packets won’t be sent or received) (Figure 3-4).

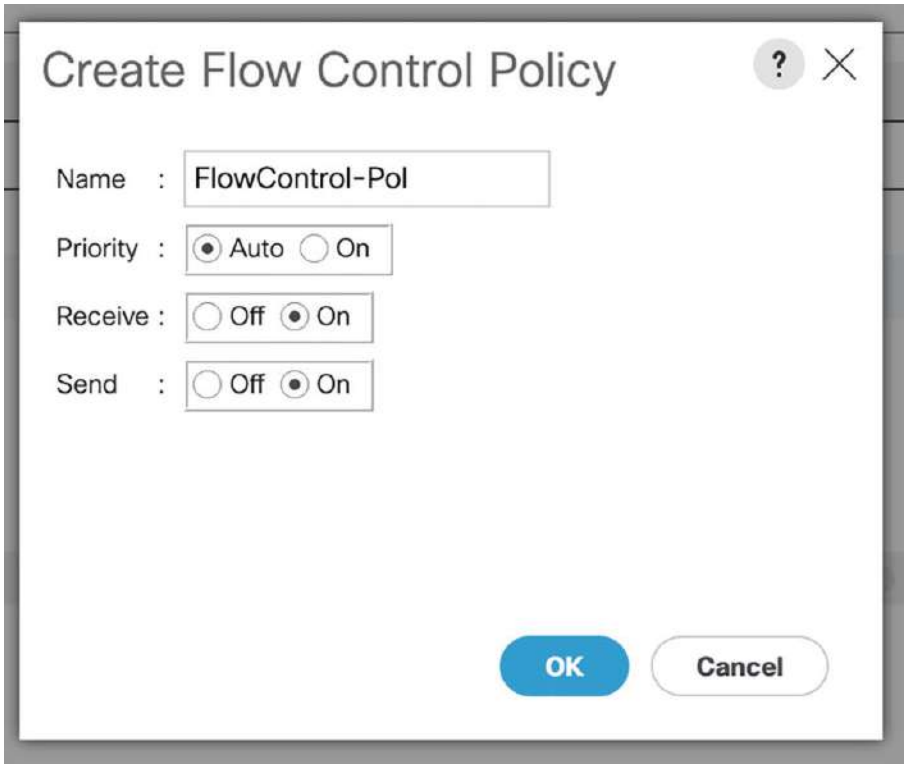

Figure 3-4 Flow Control Policy

We then assign this to the interface (Figure 3-5).

Figure 3-5 Interface Flow control

The Link Profile controls how the interface works with UDLD (UniDirectional Link Detection). We can create a new Link Profile by going to LAN ➤ Policies ➤ LAN Cloud ➤ Link Profile ➤ default and clicking “Create UDLD Link Policy.” Here we name the policy and set it to enabled and set the mode, either normal or aggressive (Figure 3-6). Normal mode will detect mis-cabling issues, whereas aggressive mode will detect when a link has become unidirectional. Normal mode is not used very much, as Aggressive mode will give us the good stuff that we need, such as “bad” ports being disabled so that failover can happen.

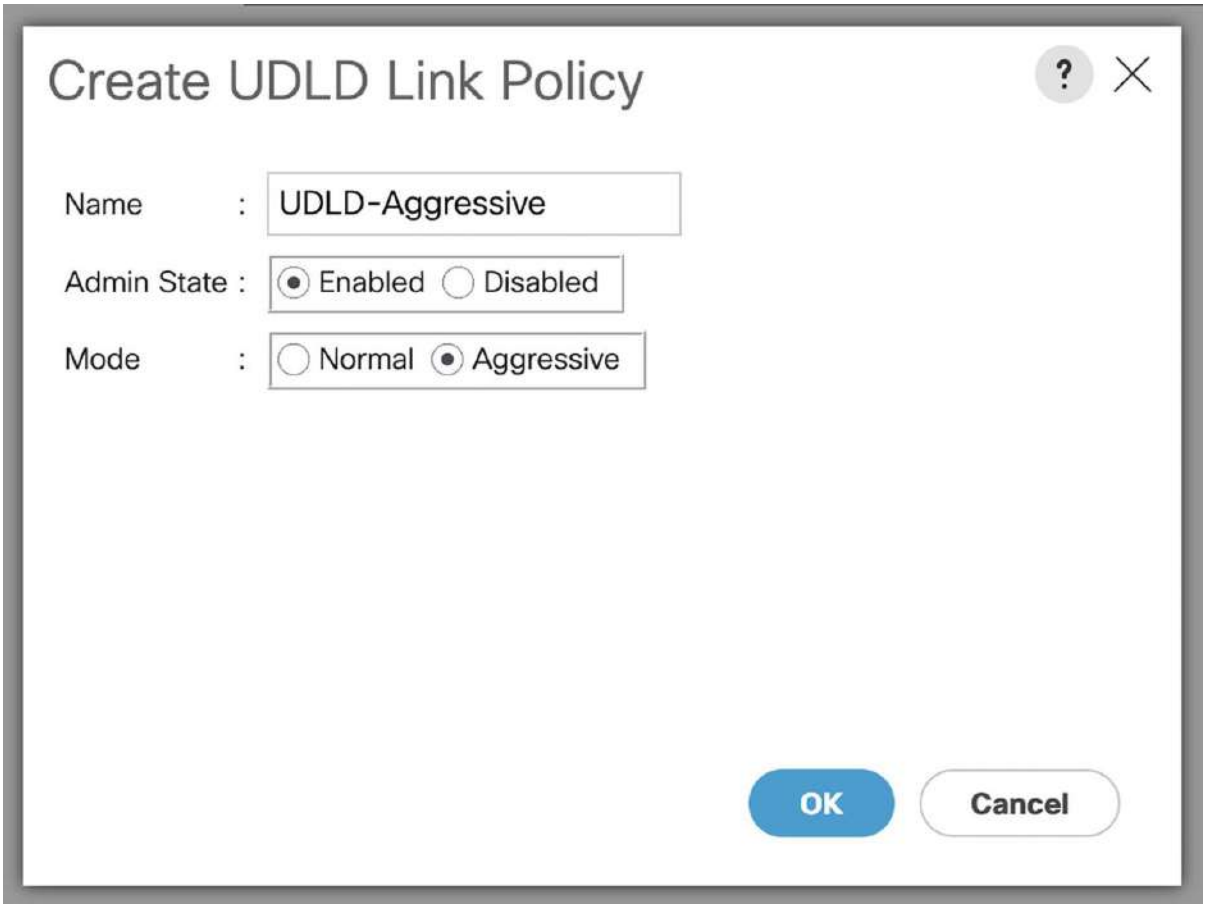

Figure 3-6 UDLD modes

Now that we have a UDLD policy, we can create a link policy by going up one level (LAN ➤ Policies ➤ LAN Cloud ➤ Link Profile) and clicking “Add.” We can name the policy and assign the UDLD-Aggressive link policy to it (Figure 3-7).

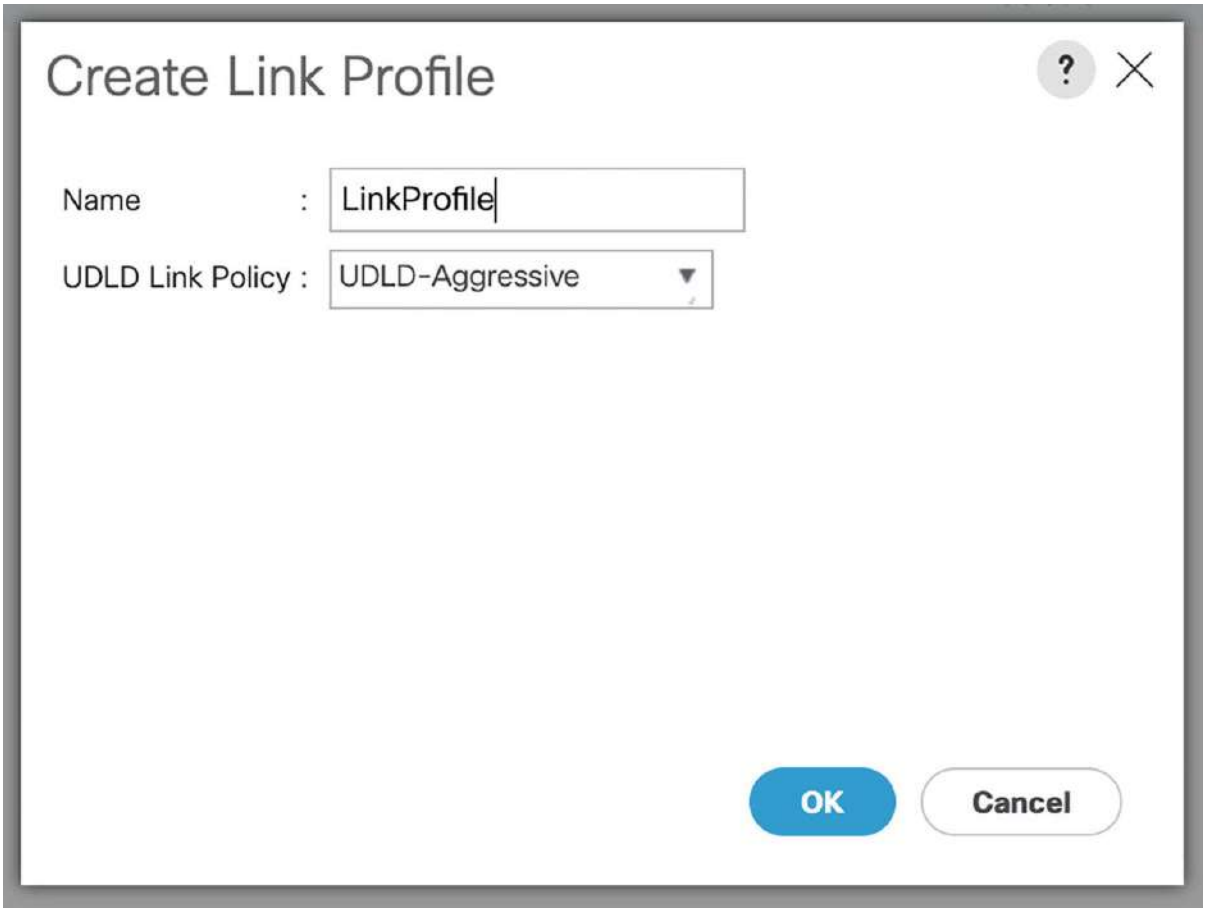

Figure 3-7 Link Profile

The next step is to assign this profile to our interface (Figure 3-8), making sure that we click “Save Changes” at the bottom of the screen.

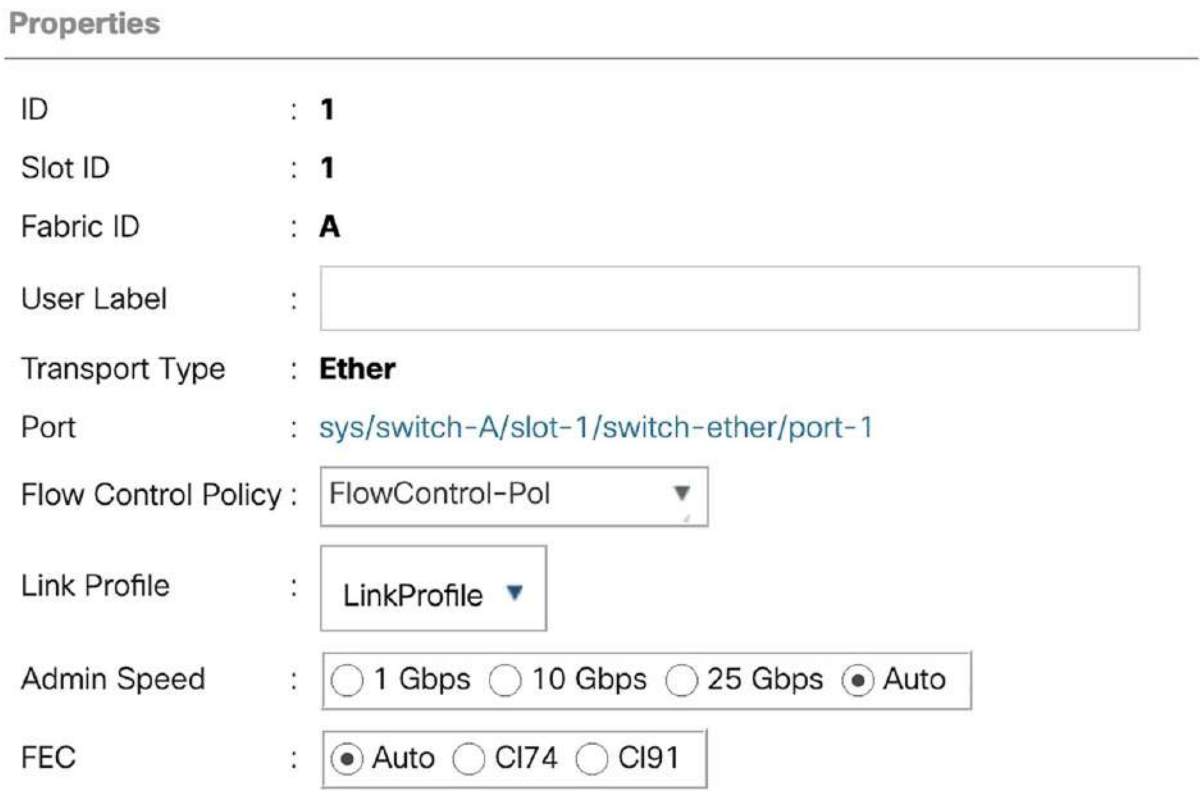

Figure 3-8 Assigning a link profile to an interface

The following settings control the port speed and the Forwarding Equivalence Class (FEC), which is a form of quality of service.

Clearly, one uplink interface alone will not be enough; we should, at a very minimum, add an uplink on the second FI. Ideally, we would have second interfaces on each FI, going to the other upstream switch to provide a level of redundancy (Figure 3-9).

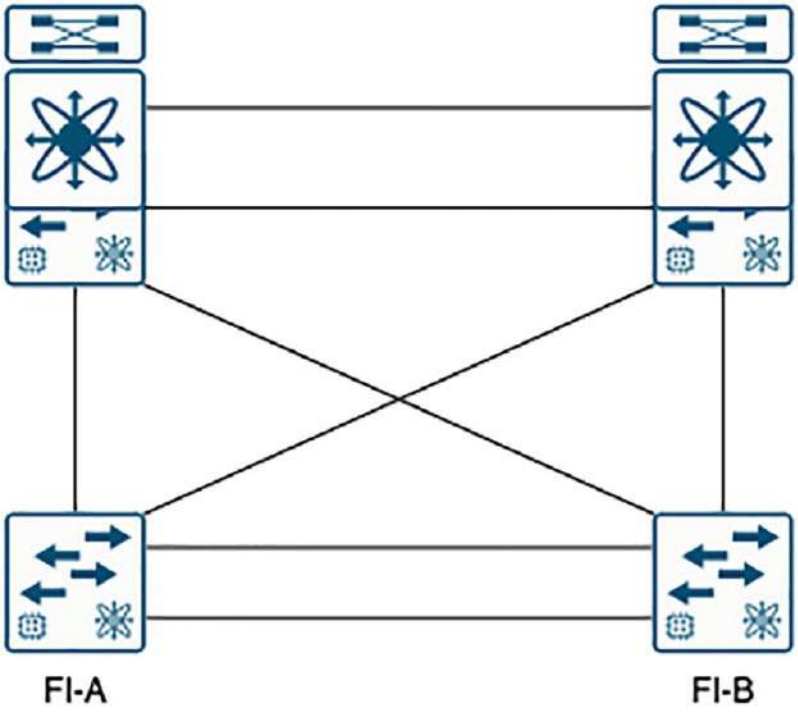

Figure 3-9 FI redundant uplinks

While two interfaces are good, we are not making the best of our capabilities. With uplink ports, traffic is pinned to one of these links. One isn't much fun when we could use all four cables at the same time (turning single 40GBps links into a combined 80GBps link).

To do this, we need to create another uplink on FI-A (1/2). We can do this without going to the Equipment tab, we just need to go to LAN ➤ LAN Cloud ➤ Uplink Eth Interfaces and click on "Add," and select Port 2, by double clicking on it, under the fixed module (Figure 3-10):

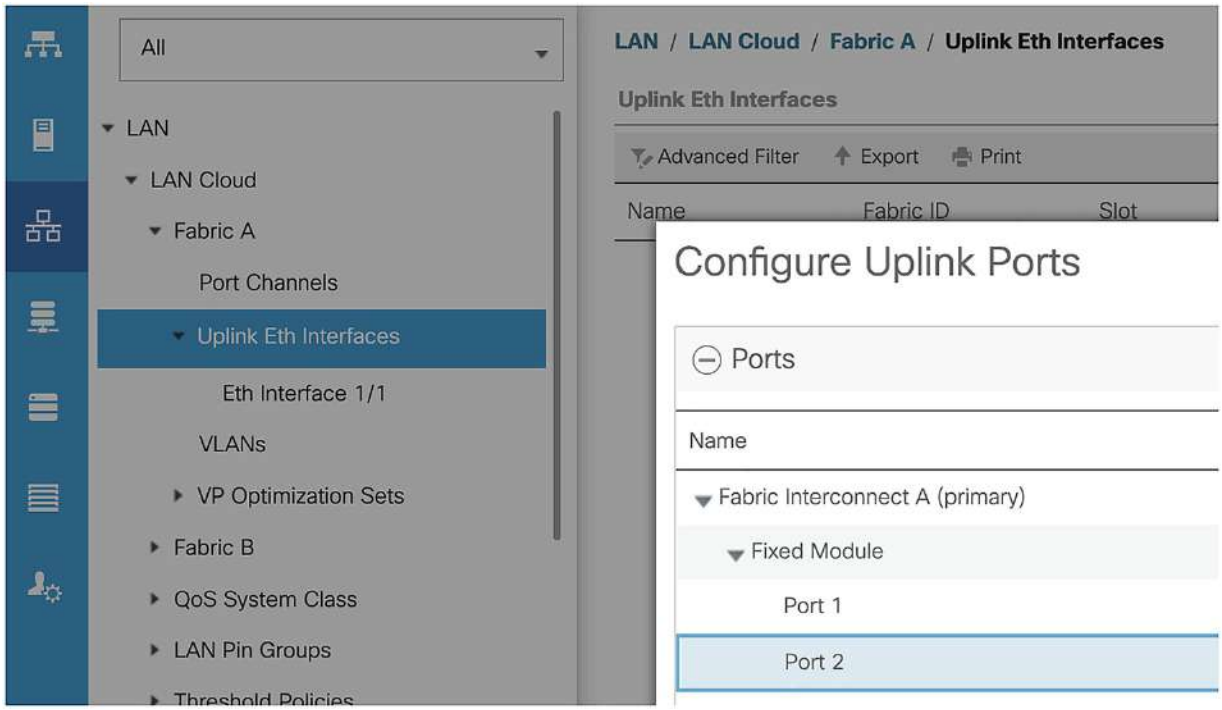

Figure 3-10 Adding another uplink

We also need to add two uplinks (eth1/1 and eth 1/2) to FI-B, using the same method.

Now we have an even number of links, we can create our port channels.

We create the port channels by going top LAN ➤ LAN Cloud ➤ Fabric A ➤ Port Channels. Click the “Add.” We set the port channel number, and give it a name (Figure 3-11). Click “Next”

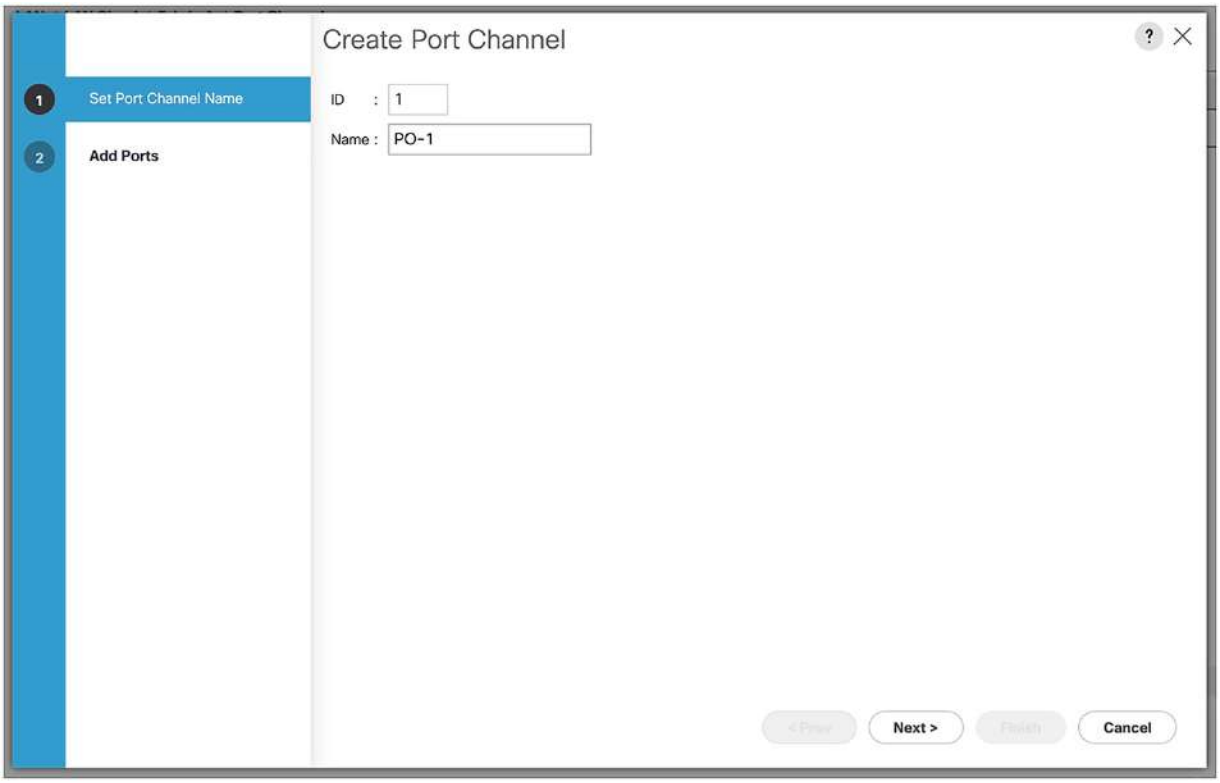

Figure 3-11 Creating the port-channel

In the next window, select the interfaces to add to the port channel (Figure 3-12).

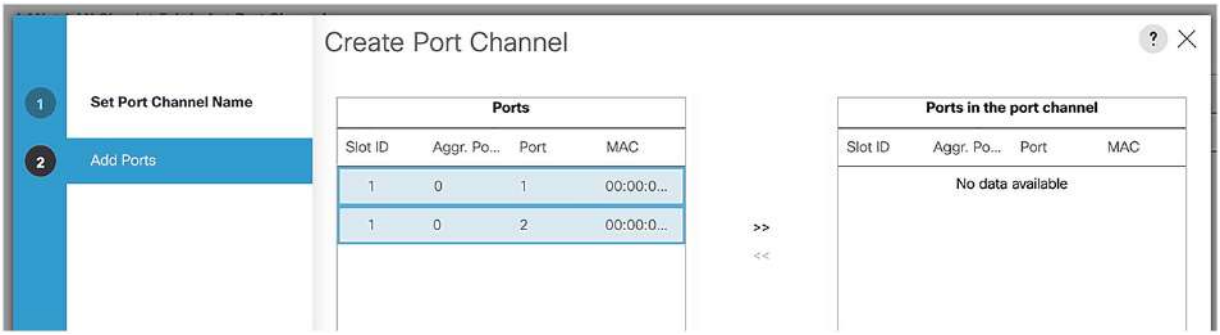

Figure 3-12 Adding interfaces to the port-channel

Click the double arrow button to move them into the port channel (Figure 3-13).

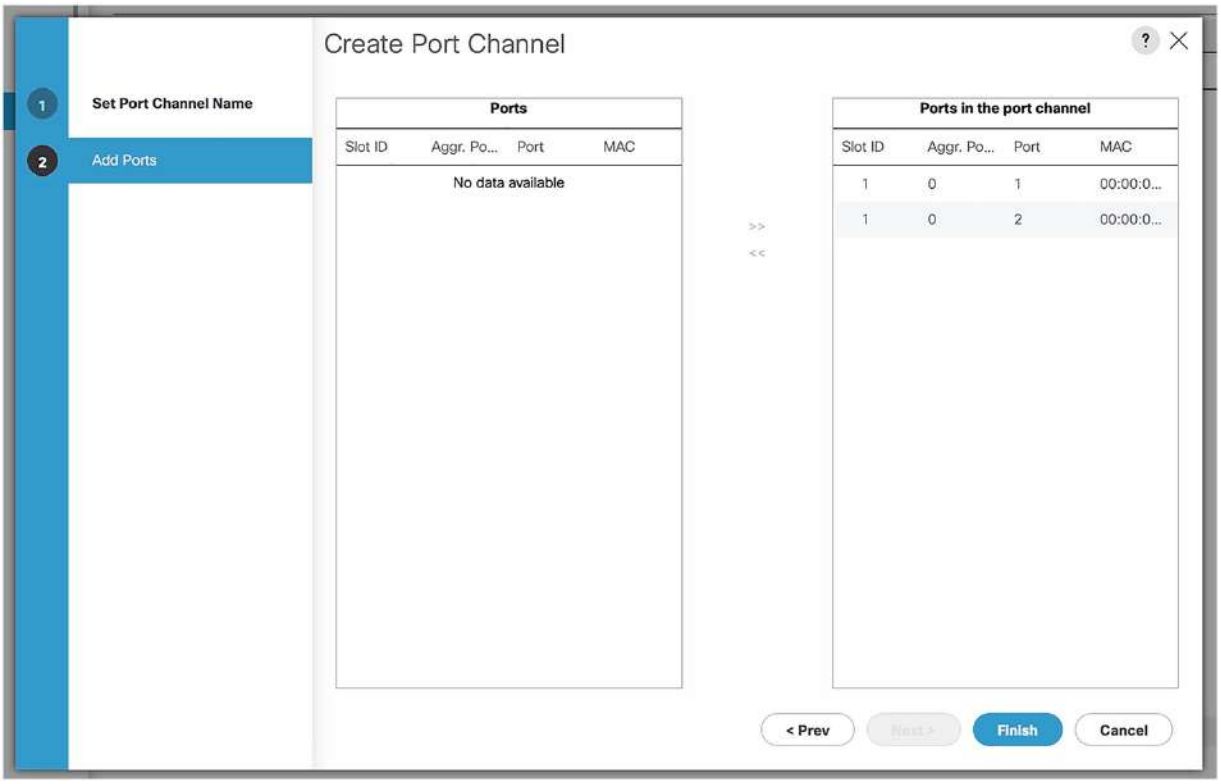

Figure 3-13 Added interfaces

Once you have added the interfaces, click “Finish.” You will receive a message to say that the port channel has been created (Figure 3-14).

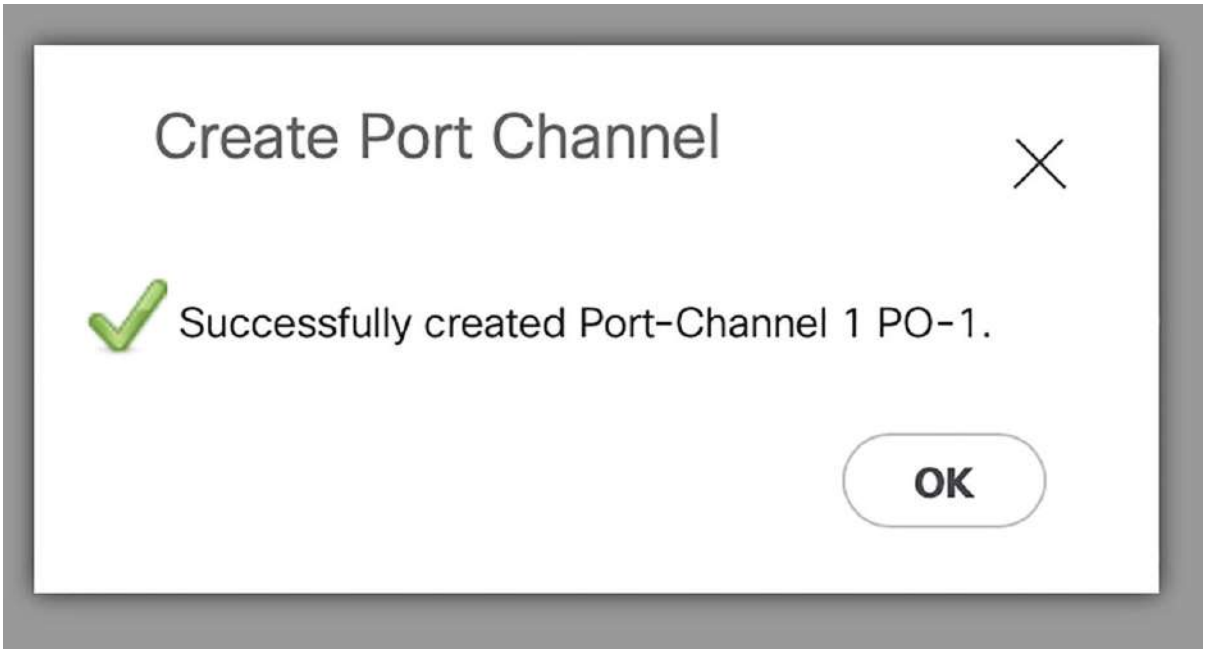

Figure 3-14 The port channel has been created.

Repeat the process, adding port channel 2 to FI-B.

Our UCS port channel setup will look like this (Figure 3-15):

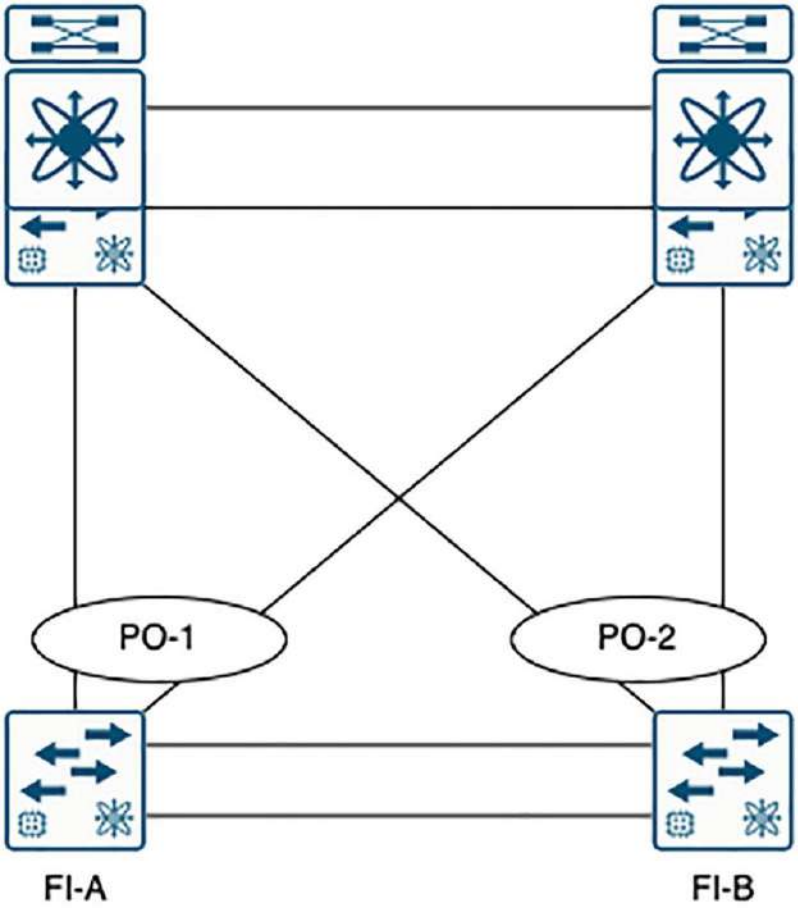

Figure 3-15 Port-Channel topology

From the point of view of the Nexus switches above our FI's, the configuration would look like this (Figure 3-16):

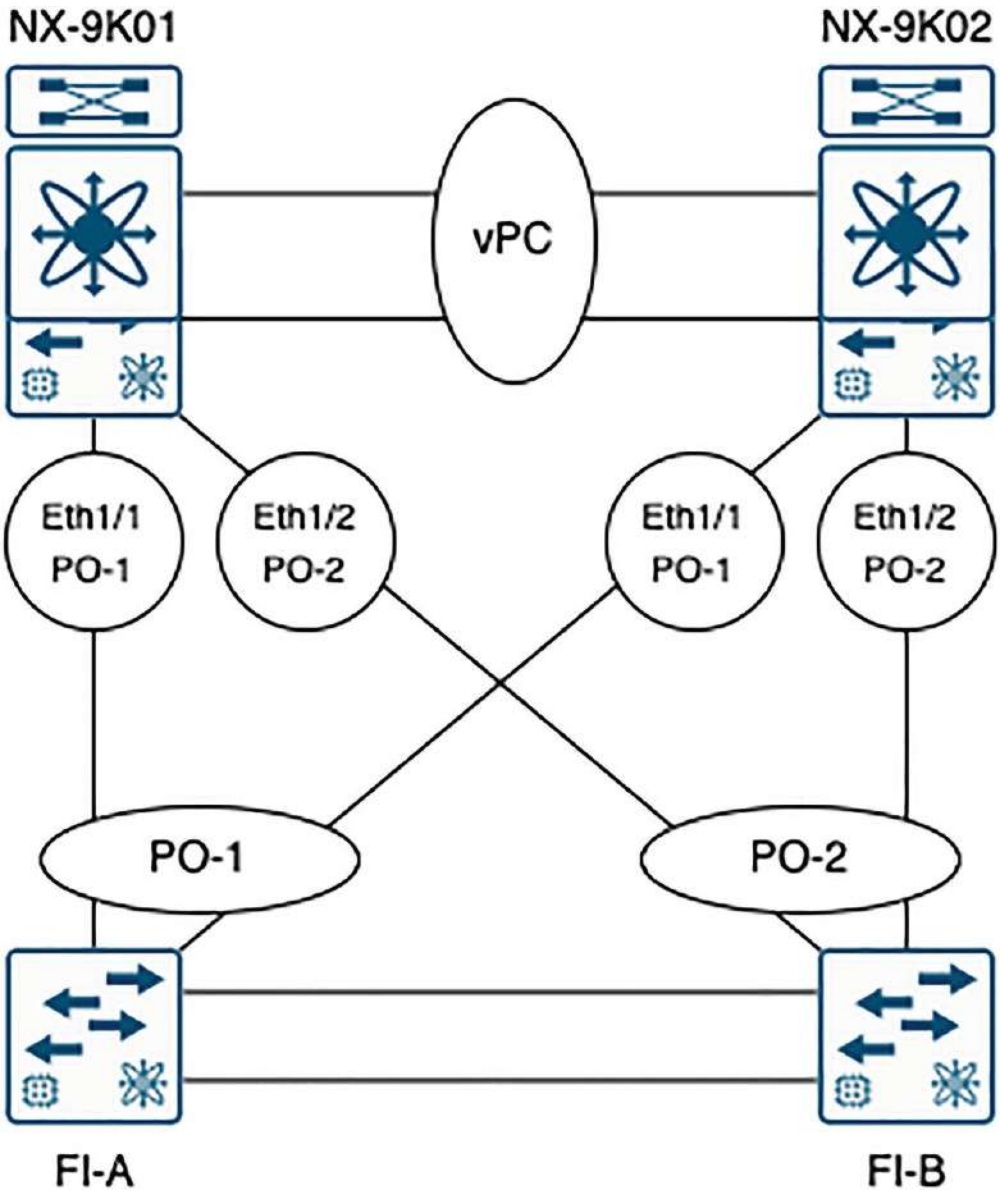

Figure 3-16 Completed Port-Channels

The Nexus interfaces and port channels would be configured as follows:

```
NX-9K01# sh run int eth1/1

interface Ethernet1/1
  description ### FI-A PORT 1 UPLINK ###
  switchport
  switchport mode trunk
```

```
  switchport trunk allowed vlan all
  spanning-tree port type edge
  spanning-tree bpduguard enable
  speed 40000
  no negotiate auto
  channel-group 1 mode active
  no shutdown

NX-9K01# sh run int eth1/2

interface Ethernet1/2
  description ### FI-B PORT 1 UPLINK ###
  switchport
  switchport mode trunk
  switchport trunk allowed vlan all
  spanning-tree port type edge
  spanning-tree bpduguard enable
  speed 40000
  no negotiate auto
  channel-group 2 mode active
  no shutdown

NX-9K01# sh run int po 1

interface port-channel1
  description ### VPC to FI-A ###
  switchport
  switchport mode trunk
  switchport trunk allowed vlan all
  speed 40000
  no negotiate auto
  no lacp suspend-individual
  vpc 1

NX-9K01# sh run int po 2

interface port-channel2
```

```
  description ### VPC to FI-B ###
  switchport
  switchport mode trunk
  switchport trunk allowed vlan all
  speed 40000
  no negotiate auto
  no lacp suspend-individual
  vpc 2
```

And the other switch configuration:

```
NX-9K02# sh run int eth 1/1

interface Ethernet1/1
  description ### FI-A PORT 2 UPLINK ###
  switchport
  switchport mode trunk
  switchport trunk allowed vlan all
  spanning-tree port type edge
  spanning-tree bpduguard enable
  speed 40000
  no negotiate auto
  channel-group 1 mode active
  no shutdown

NX-9K02# sh run int eth 1/2

interface Ethernet1/2
  description ### FI-B PORT 2 UPLINK ###
  switchport
  switchport mode trunk
  switchport trunk allowed vlan all
  spanning-tree port type edge
  spanning-tree bpduguard enable
  speed 40000
  no negotiate auto
```

```
  channel-group 2 mode active
  no shutdown

NX-9K02# sh run int po 1

interface port-channel1
  description ### VPC to FI-A ###
  switchport
  switchport mode trunk
  switchport trunk allowed vlan all
  speed 40000
  no negotiate auto
  no lacp suspend-individual
  vpc 1

NX-9K02# sh run int po 2

interface port-channel2
  description ### VPC to FI-B ###
  switchport
  switchport mode trunk
  switchport trunk allowed vlan all
  speed 40000
  no negotiate auto
  no lacp suspend-individual
  vpc 2
```

In this configuration, we can set the interfaces to run together (instead of singularly) and also benefit from a considerable speed increase. While this is not something we can achieve within the sandboxed environment that is UCSPE (as our port channel status will show as “Indeterminate,” we can see this in a real-life example (Figure 3-17):

Figure 3-17 80 Gbps port channel

Summary

In this chapter, we configured uplink ports to connect our UCS to the rest of the network. In the next chapter, we will start configuring the policies we need for our servers.

S. Fordham, *Introducing Cisco Unified Computing System*
https://doi.org/10.1007/978-1-4842-8986-0_4

4. Policies

Stuart Fordham[1]
(1) Bedfordshire, UK

At this stage, you are probably hungry to do some actual UCS server configuration, I am, and this is where it all starts; with policies.

Policies are used to create service profile templates, and from these templates we can assign service profiles to our servers. Before we start though, we should create our UCS organization.

Creating the UCS Organization

We create UCS organizations to simplify our management. They offer us a hierarchical way of organizing our policies (as well as our pools and service profiles). We create the organization by going to the Servers tab, expanding any one of the options, such as *Servers* ➤ *Policies* ➤ *root* ➤ *Sub-Organizations,* and selecting "*Add*." Give the organization a name, and click "*OK*" (Figure 4-1).

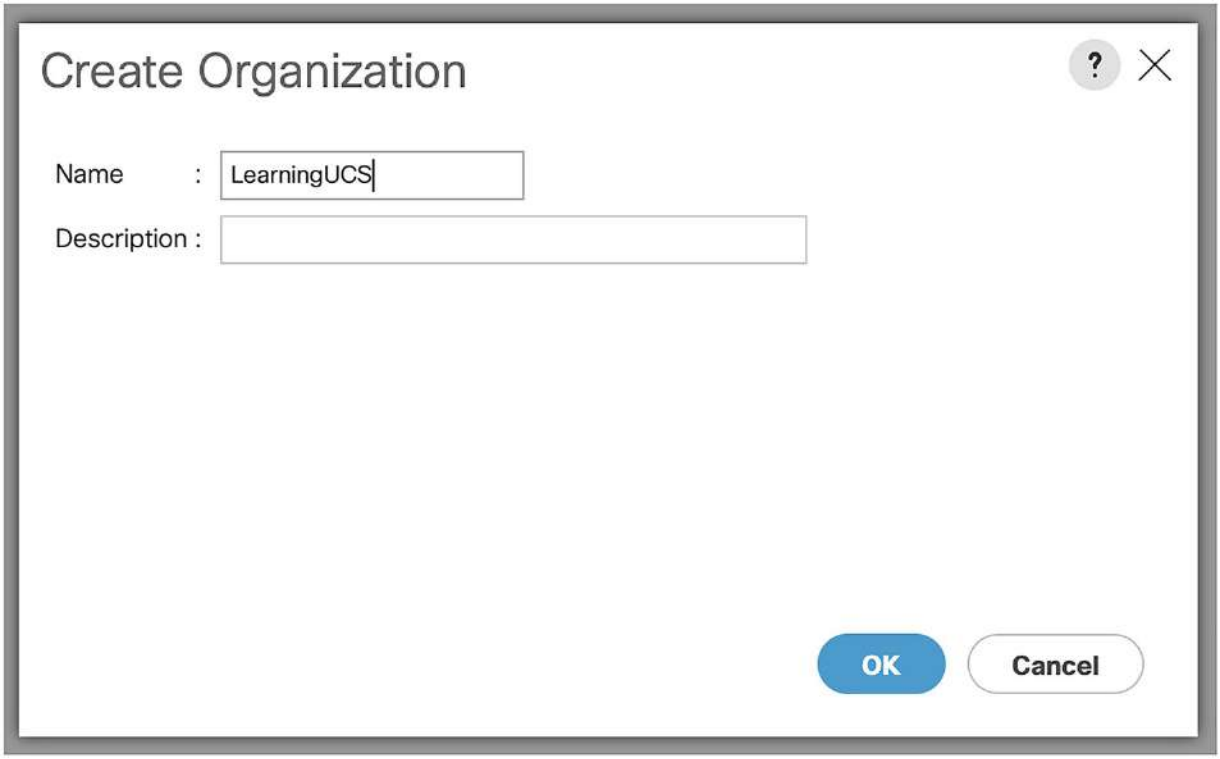

Figure 4-1 Creating the Organization

You will receive an acknowledgment that the organization has been created (Figure 4-2).

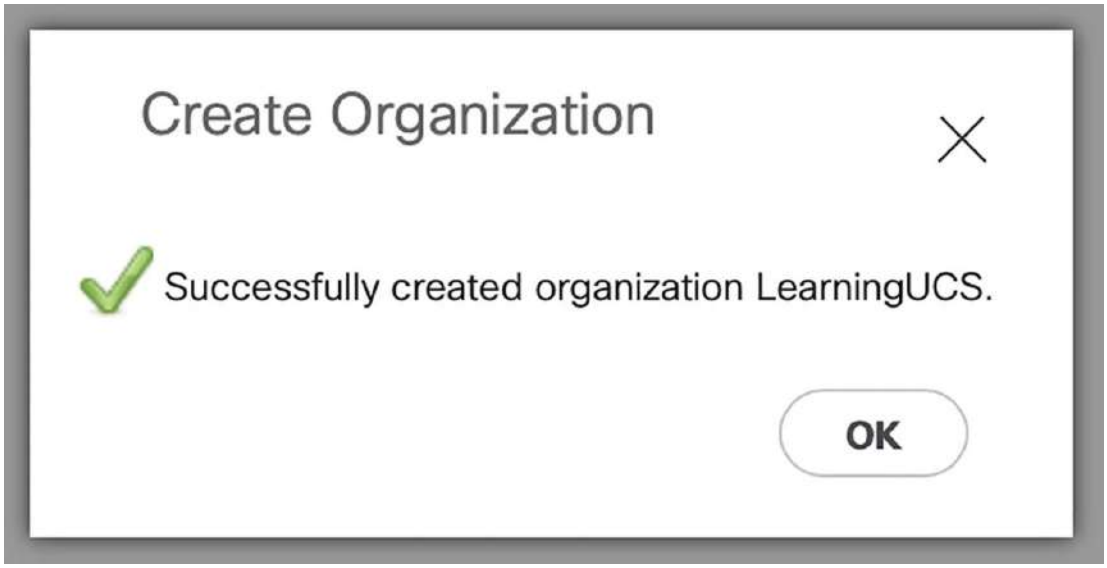

Figure 4-2 The organization has been completed

You will also notice that the same organization has been created under Service Profiles and also under Service Profile Templates, as well as Pools. You will also find the new organization in the LAN tab, the SAN tab, the Storage tab, and the Chassis tab.

Now, we can start to create our policies. We will not cover every single policy option, as there are a lot of them. Instead, we will focus on the ones required to create a service profile template, which will then be applied to our servers.

Storage Policies

Our storage policy is going to be quite simple; we will just create a mirrored RAID volume from our two disks. To do this, go to "*Servers ➤ Policies ➤ root*," right-click "*Local Disk Config Policies*" and select the pop-up option to create one. Call it "*LocalDiskPol*" and set the mode to "*RAID 1 Mirrored*" (Figure 4-3).

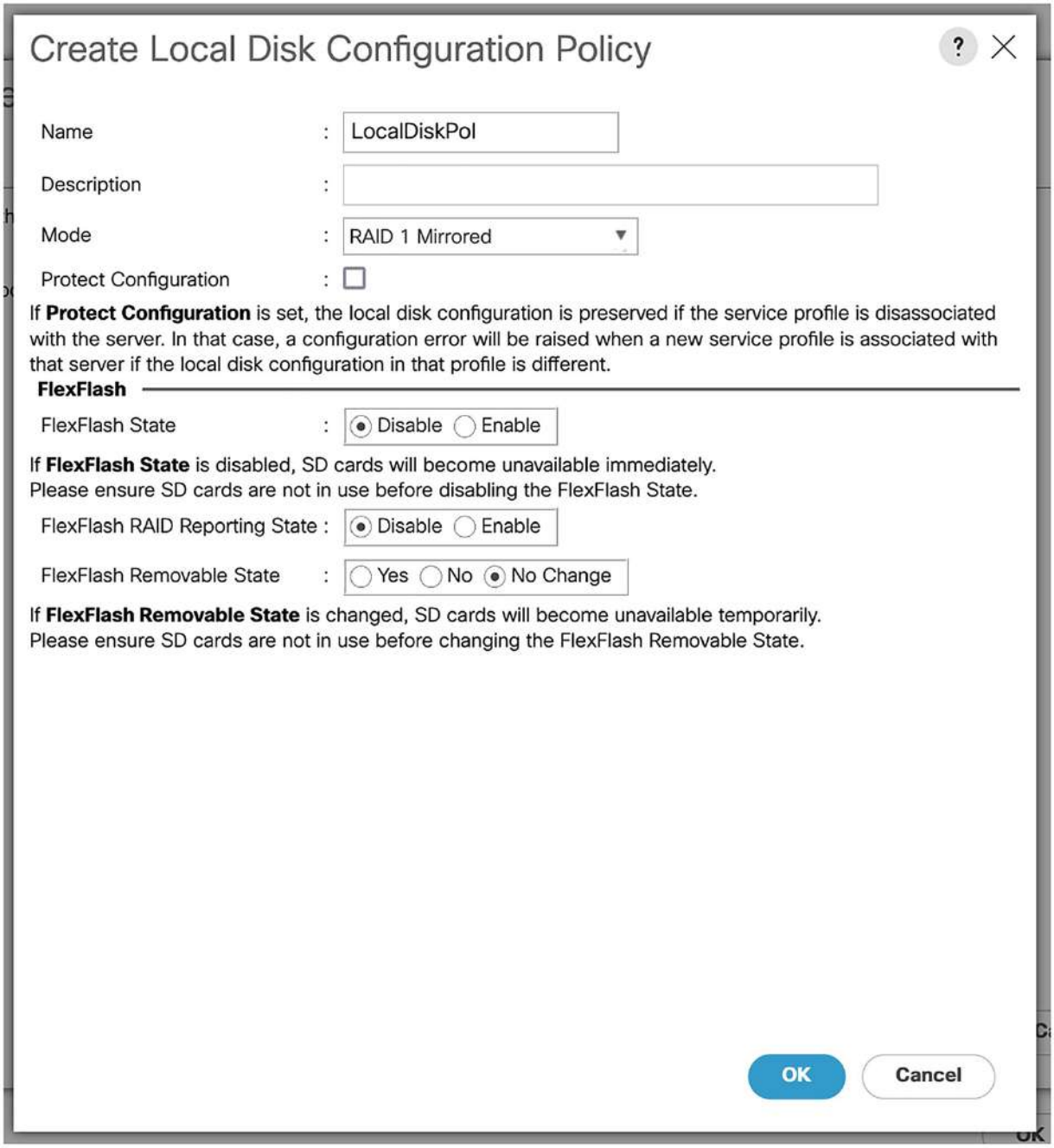

Figure 4-3 Local Disk policy

Dynamic vNIC Connection Policies

Dynamic vNICs are not applicable to us (in our sandboxed environment), as these are used for determining connectivity between virtual machines and dynamic vNICs running on servers with VIC adapters. However, if we were running through the service profile wizard (which we will do in the next chapter), this is where we would set up this connectivity. In the same wizard though, is the VLAN creation, which is where we are going to sidestep to.

Creating VLANs

UCSPE has already created some VLANs for us, but we will create some more, by going to *LAN ➤ LAN Cloud ➤ VLANs*. We can create VLANs on a per-Fabric basis, or on both fabrics at the same time. Click "*All*" and then click "*Add*.. Our first VLAN will be called "DB," and will have a VLAN ID of 10 (Figure 4-4).

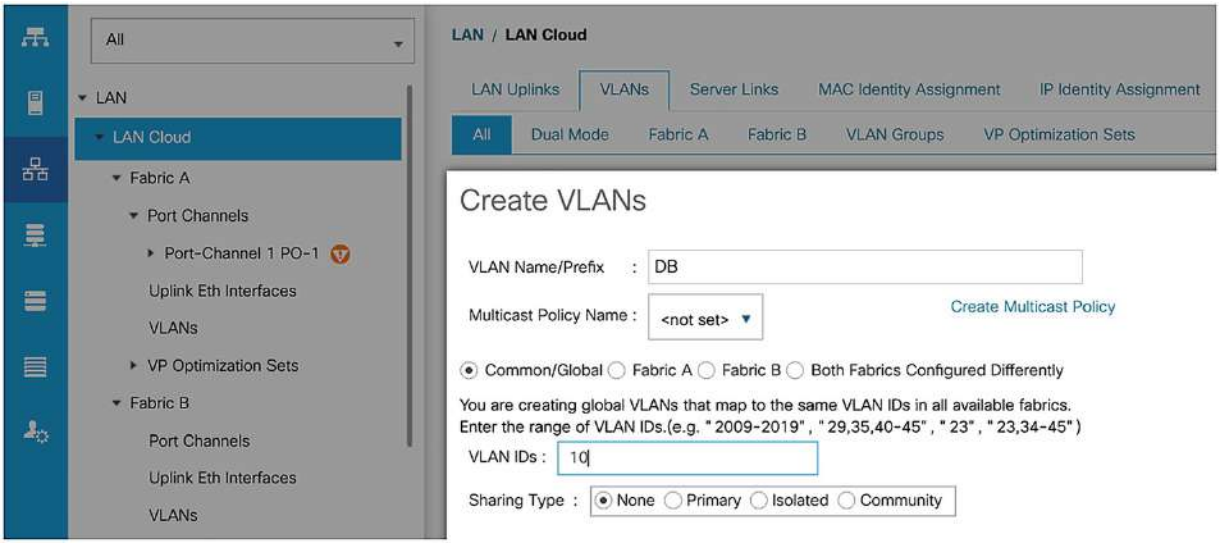

Figure 4-4 Creating the DB VLAN

As the GUI shows, we can use this to create ranges of VLANs on both fabrics (Common/Global), individual

fabrics, or we can configure the fabrics differently. The latter option allows us to specify different VLAN IDs for each fabric (though the name we give this particular VLAN will be the same across both fabrics, just the VLAN ID will be different).

The sharing type is for setting up private VLANs (PVLAN) and allows us, if we should so desire, to isolate ports. We create a primary VLAN and one (or more) secondary VLANs, which can either be an isolated or a community VLAN. Isolated ports can only communicate with the associated port in the primary VLAN, not even with each other. Community ports – communicate with each other and with promiscuous ports. For both Isolated and Community VLANs, we must create a primary VLAN first.

Repeat the process, creating a VLAN 13, which is our DMZ. Our VLANs should look like Figure 4-5.

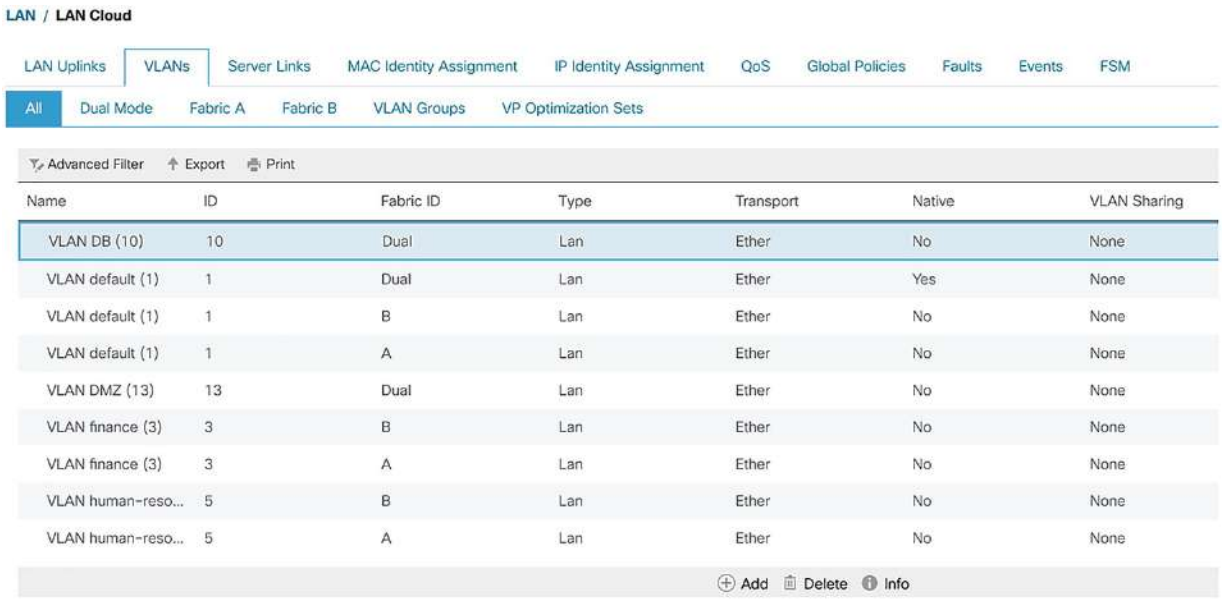

Figure 4-5 Our VLANs

vNIC/vHBA Placement

UCS blades have a component called a “Mezzanine” card. Mezzanine cards can give us storage acceleration, port expansion, GPUs (Graphics Processing Units) and VICs (Virtual Interface Cards). We also have mLOMs (modular LAN on Motherboard) cards, which offer VIC expansion.

The UCS, as we spoke about back in Chapter 2, has an IOM and each IOM has a defined internal bandwidth (the bandwidth that goes to the blades). The 2104 has 2x 10GB, the 2204 has 4x 10GB, and the 2208 has 8x 10GB. This means that a blade can get 80Gb-KR bandwidth across a pair of IOMs.

The “KR” in this equation is a data rate specification across a backplane medium (K), using a 64B/66B (R) coding scheme (which is all to do with the electrical encoding at the physical layer) in a single lane configuration. For a deeper dive into this, check out this very good blog post: www.tbijlsma.com/2012/03/how-ucs-achieves-80gbe-of-bandwidth-per-blade/

We can control how each of our vNICs are assigned to these lanes through a “Placement Policy,” allowing us to utilize the hardware capacity to its fullest. Such as all having all vNICs on one card and all vHBAs (virtual Host Bus Adaptors) on another card; this could be due to compatibility reasons, or card speed.

To create a placement policy we would go to *Servers ➤ Policies ➤ root ➤ Sub-Organizations ➤ LearningUCS ➤ vNIC/vHBA Placement Policies*. Although we don’t need to create one ourselves, we would do so by clicking on the “Add” button (Figure 4-6).

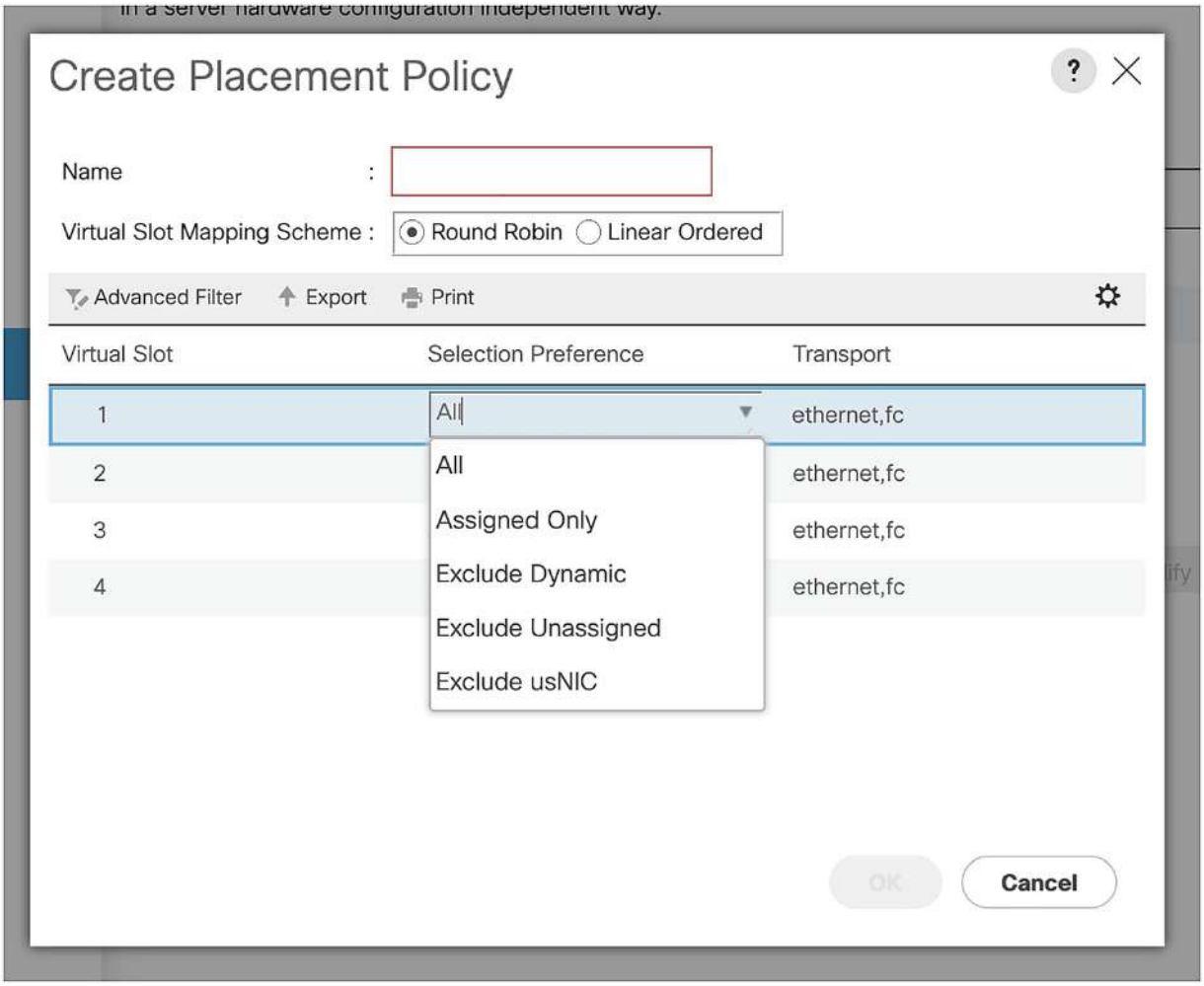

Figure 4-6 Placement Policies

The options we have are

All: the vCON (virtual network interface connection) is used for all vNICS and vHBAs that are assigned to it, not assigned to it, or are dynamic.
Assigned only: Only vNICs and vHBAs are assigned to the vCON.
Exclude-Dynamic: The vCON cannot be used for dynamic vNICS or vHBAs.
Exclude-Unassigned: the vCON can only be used for vNICs or vHBAs assigned to it, or dynamic vNICs and vHBAs.

Exclude usNIC: The vCON cannot be used by user-space NICs.

User-space NICs bypass the kernel when sending packets, improving the performance of the software. For a great blog post on actual use-cases, have a read of *https://jeremywaldrop.wordpress.com/2010/08/26/cisco-ucs-vnicvhba-placement-policies/*

vMedia Policies

vMedia policies allow us to boot our servers from ISO images stored on a share. We create these by going to *Servers ➤ Policies ➤ root ➤ Sub-Organizations ➤ LearningUCS ➤ vMedia Policies*. To create one, click *"Add"* and enter the details, such as those in Figure 4-7.

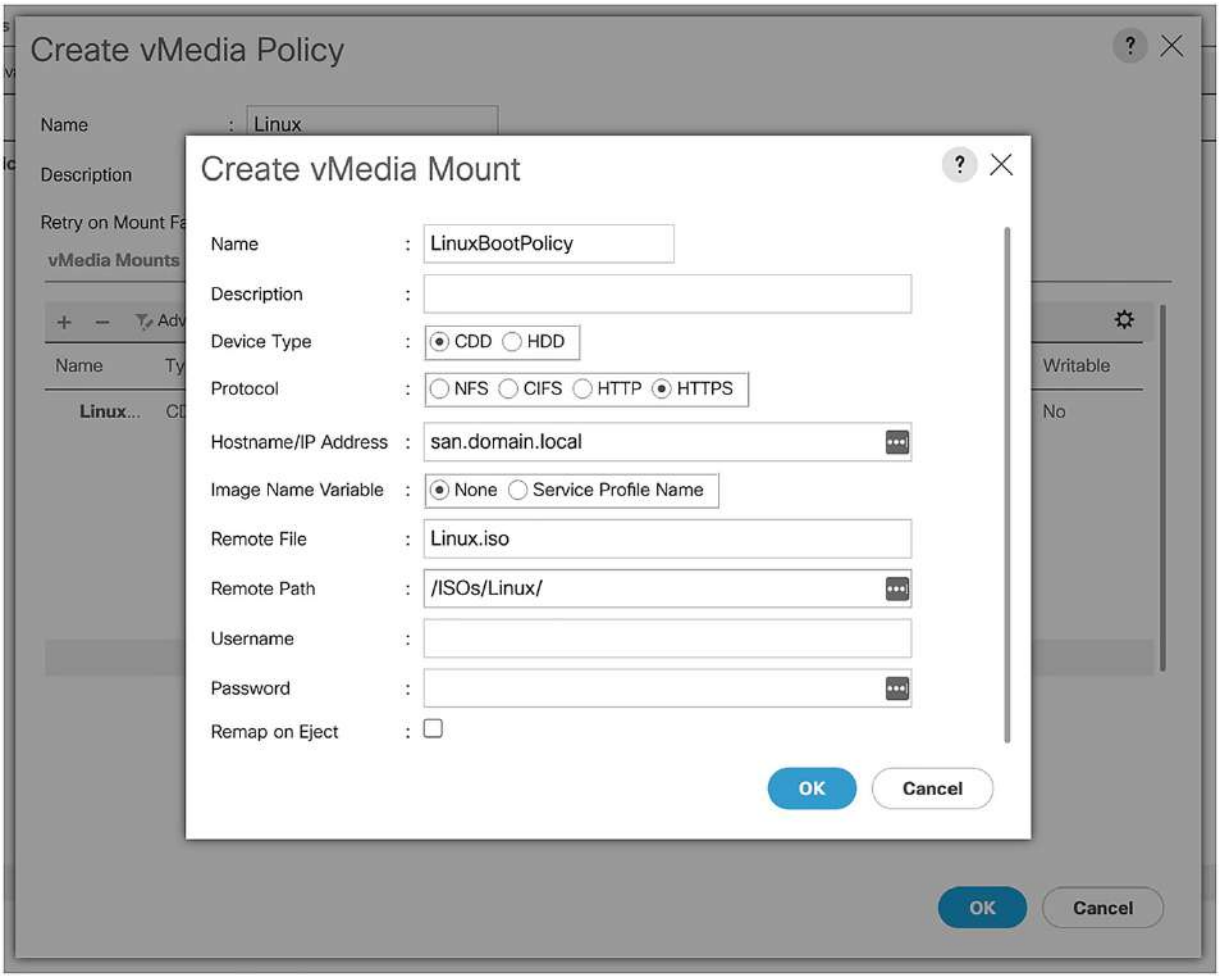

Figure 4-7 A vMedia policy

In the preceding policy, we would be loading a CD ISO image called Linux.iso from `https://san.domain.local/ISOs/Linux`. Well, depending on our server boot policy, that is.

Server Boot Policies

Server boot policies control how we boot our servers and in what order we try these options. We configure a boot policy by going to *Servers* ➤ *Policies* ➤ *root* ➤ *Sub-Organizations* ➤ *LearningUCS* ➤ *Boot Policies*. Click on *"Add"* to create a new policy. In Figure 4-8, we are creating a policy to first boot from a CD (or DVD) mounted via the CIMC. It will then try to boot from a local LUN if no CD or DVD is found.

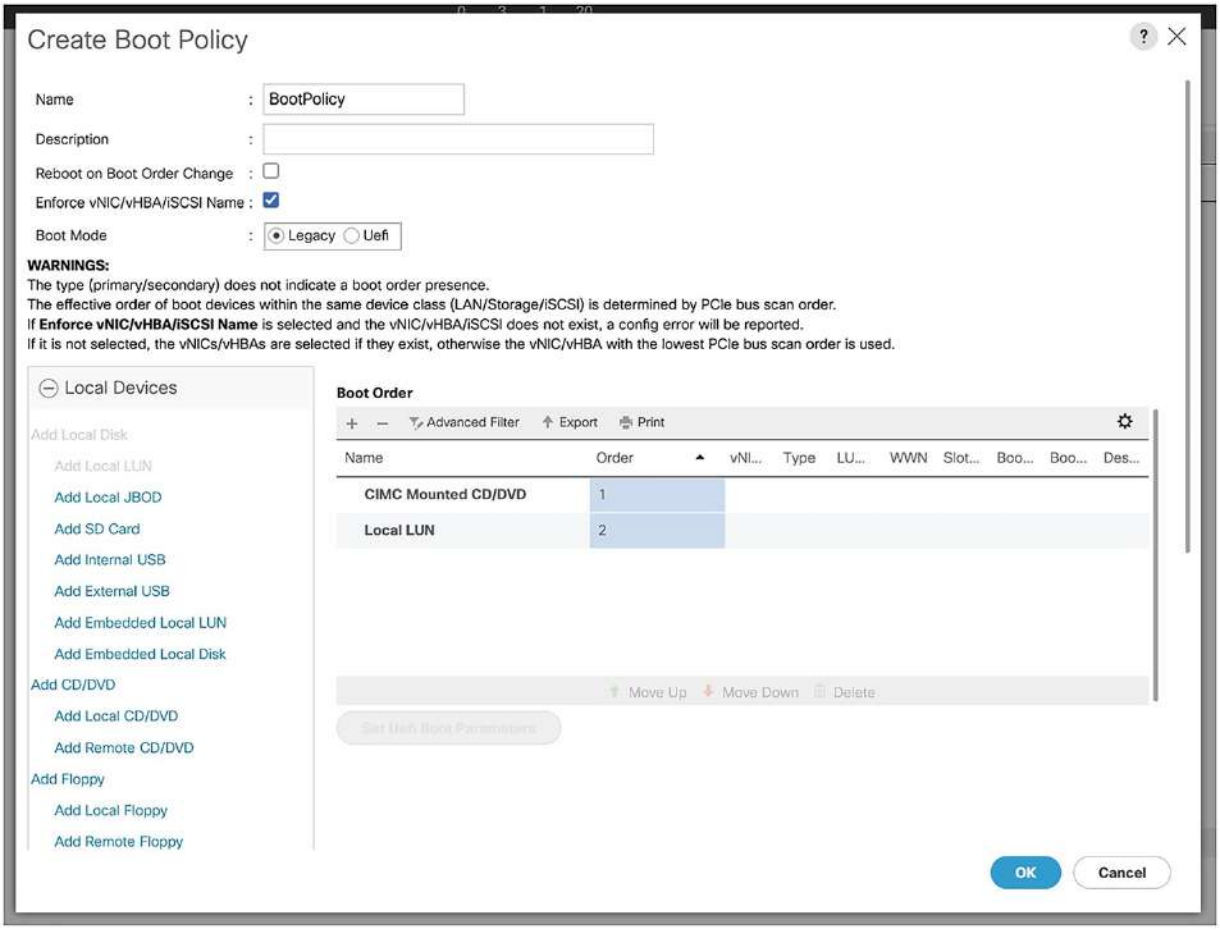

Figure 4-8 Server Boot Policy

Maintenance Policies

We will, from time to time, have to perform maintenance on our UCS, usually in the way of upgrading the firmware. You may upgrade at a particular time, taking the inevitable reboots of the fabric and IOMs as you go. However, you may not want the blades to reboot at the same time, so, unless you want to cause an outage, it's a good idea to implement a maintenance policy. Head to *Servers ➤ Policies ➤ root ➤ Sub-Organizations ➤ LearningUCS ➤ Maintenance Policies* and click "*Add*." Create a maintenance policy that will (at a very minimum) require a user acknowledgment before rebooting the servers (Figure 4-9).

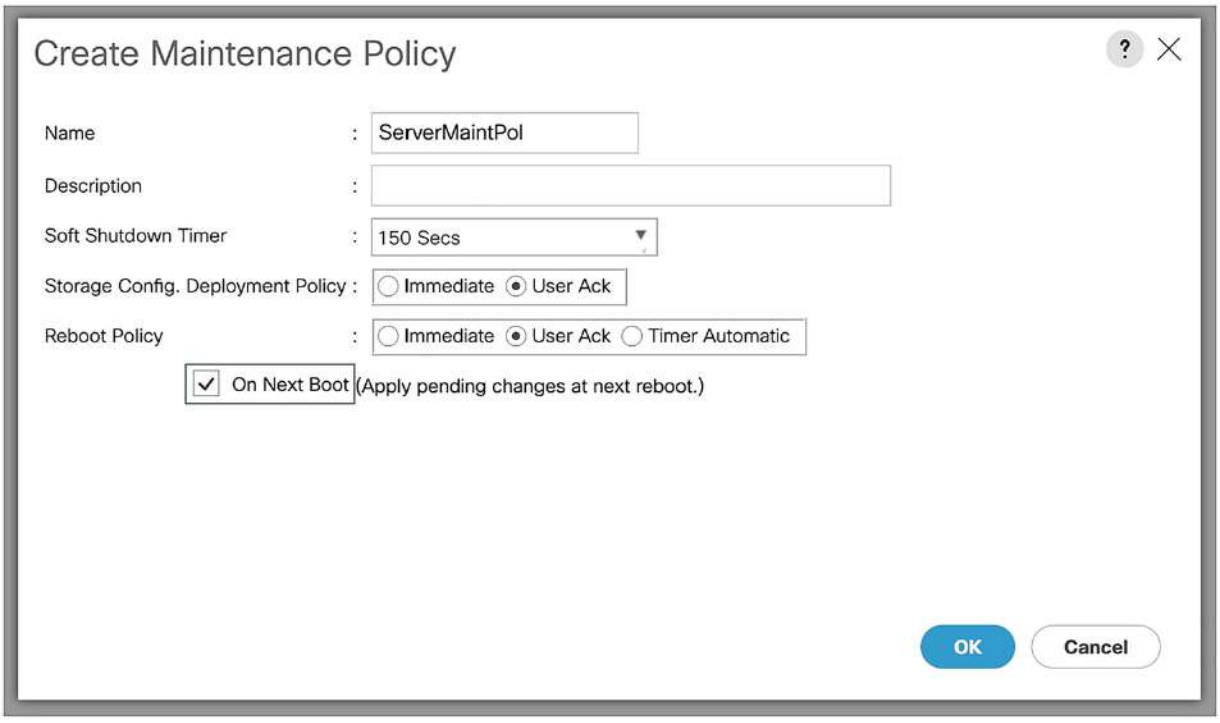

Figure 4-9 Maintenance Policy

Server Pool Policies

Server Pool are used for servers that share characteristics, such as type, amount of memory, drive configuration, or the type of CPU. We create a pool first (*Servers ➤ Pools ➤ root ➤ Server Pools*). We start by naming our server pool as shown in Figure 4-10.

Figure 4-10 Creating a server pool

Next, we add our servers, selecting them in the first window (Figure 4-11).

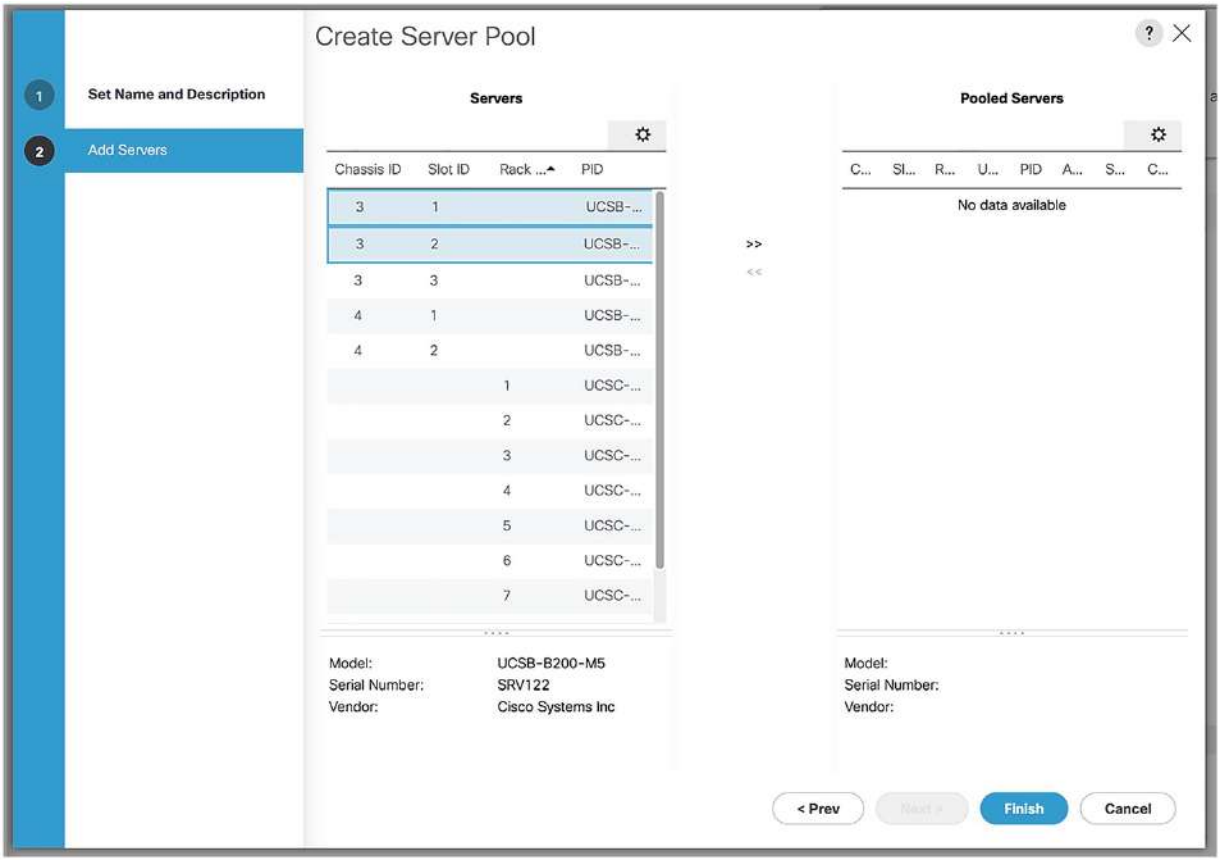

Figure 4-11 Selecting the servers for the pool

Once we have added the servers (Figure 4-12), click "*Finish*."

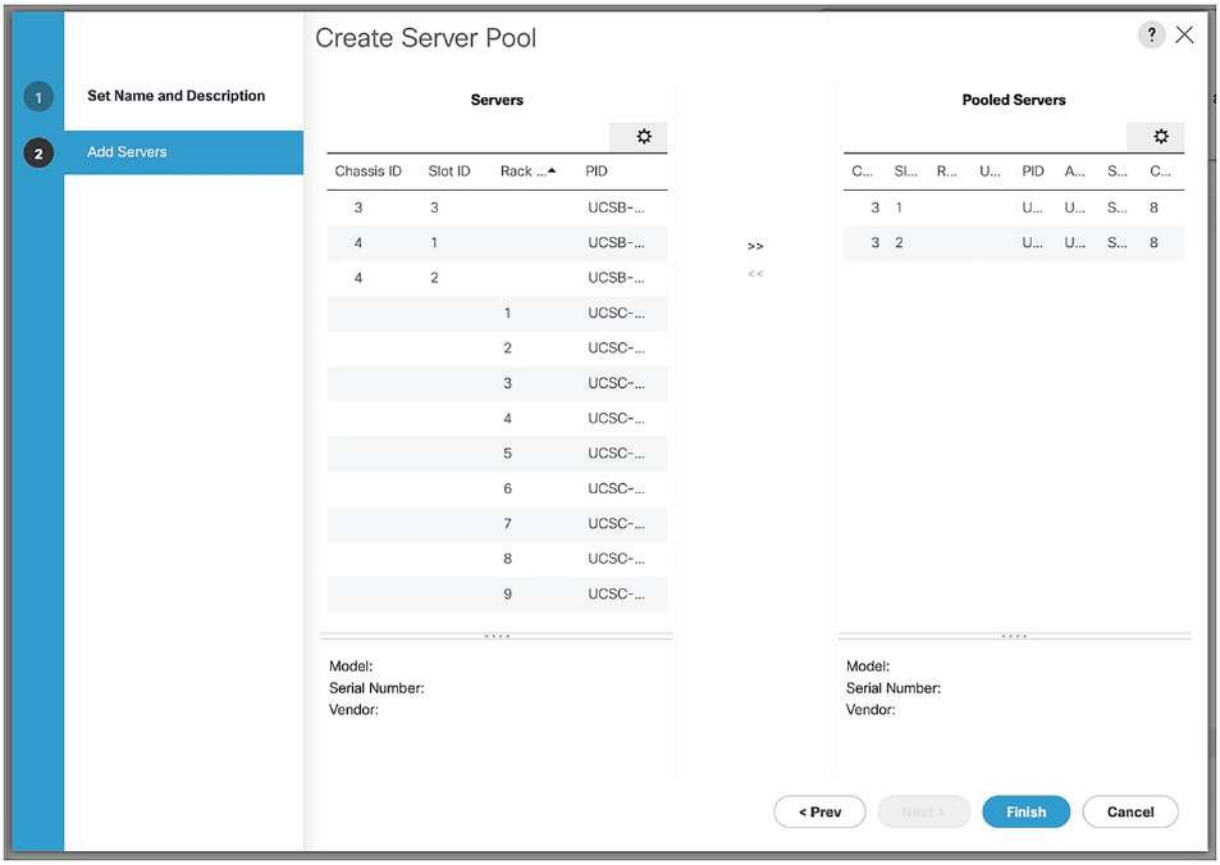

Figure 4-12 Our server pool

We can also create a pool qualification, which will, as we just mentioned, pool servers based on characteristics. We do this from *Servers ➤ Policies ➤ root ➤ Sub-Organizations ➤ LearningUCS ➤ Server Pool Policy Qualifications (Figure 4-13)*.

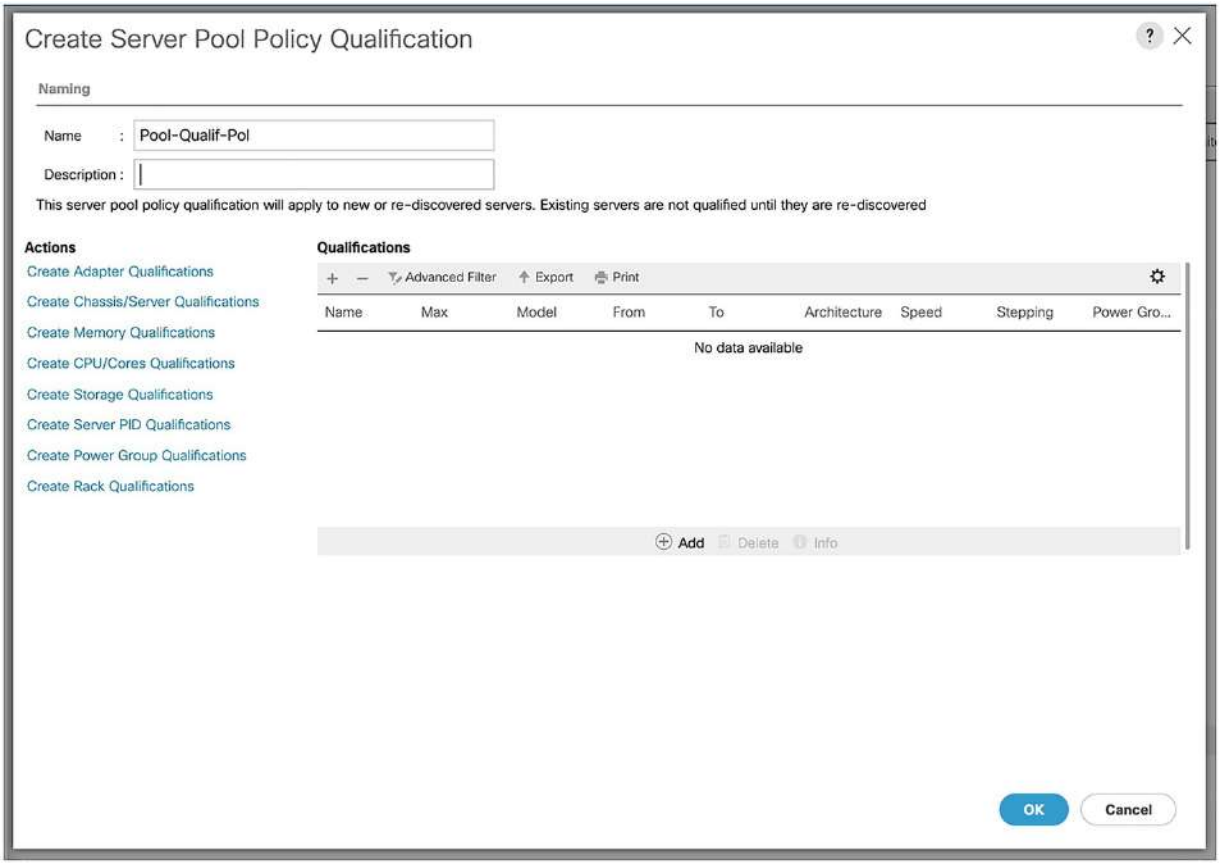

Figure 4-13 Server pool policy qualifications

In our qualification, we are going to keep it simple and just match against the server product ID (PID), as shown in Figure 4-14.

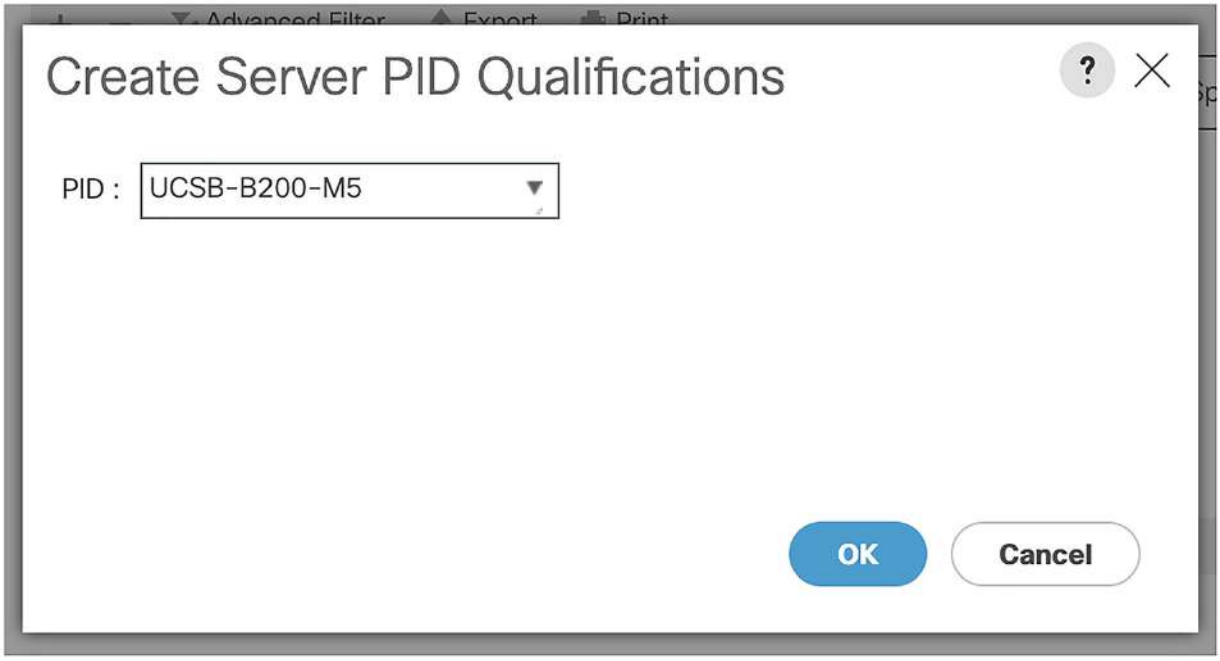

Figure 4-14 A server PID qualification

Once we have added this (Figure 4-15), we can click on *"Finish"* to create the qualification.

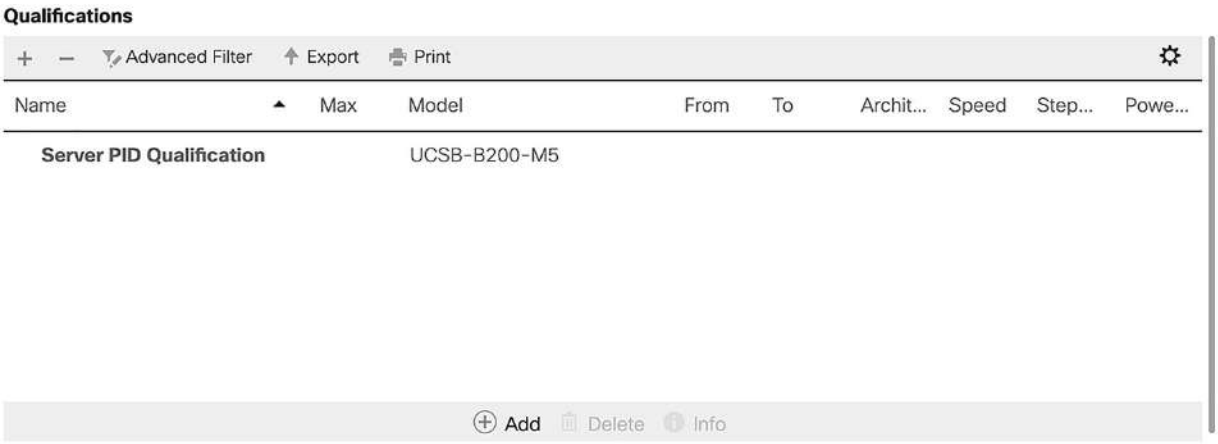

Figure 4-15 Our completed qualification

The last step is to create a policy to tie these all together. We do this by going to *Servers ➤ Policies ➤ root ➤ Sub-Organizations ➤ LearningUCS ➤ Server Pool Policies*. We name our policy and either assign the policy to a pool or we can select the qualification, but not both

(Figure 4-16). While we can set both when we create the policy, once we go back into it, we will find the pool empty. Pool assignments are fairly static, whereas qualifications are more dynamic in nature.

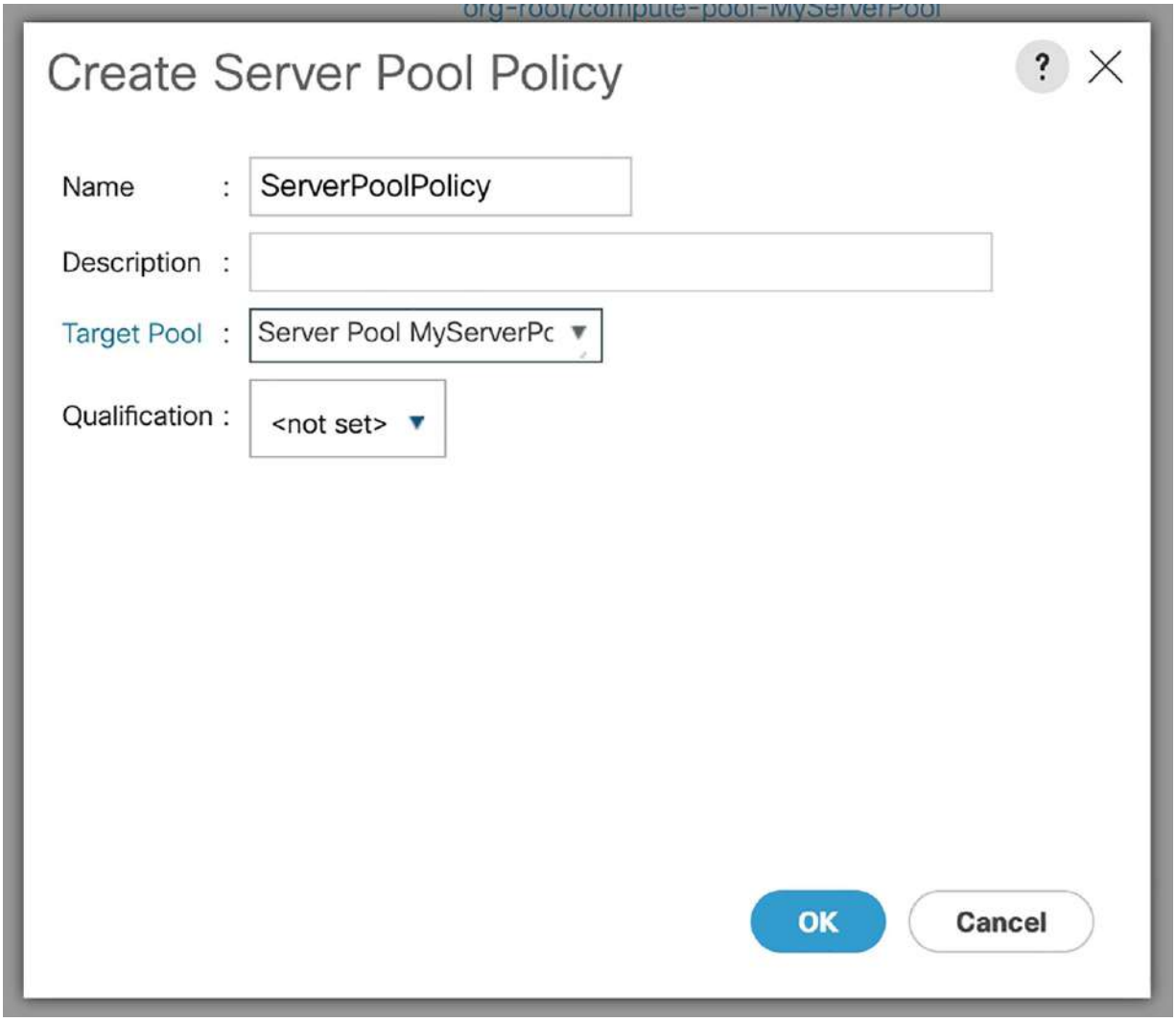

Figure 4-16 Our Server Pool Policy

The last policies we are going to cover are some small but very important ones!

Operational Policies

Operational policies cover aspects of the servers like BIOS, IPMI, management IP addresses, power control, scrub policies, KVM management and graphics card policies. There are three that we should cover, starting with management IP addresses.

Management IP Addresses

The management IP addresses come from a defined pool of IP addresses and it is to one of these addresses we connect to when we launch the KVM from the UCS GUI. We create the pool by going to *LAN ➤ Pools ➤ root ➤ Sub-Organizations ➤ LearningUCS ➤ IP Pools*. We can create them under *LAN ➤ Pools ➤ root ➤ IP Pools* as well, if you so desire. Create a new IP pool called *"KVM-IP-Pool"* (Figure 4-17).

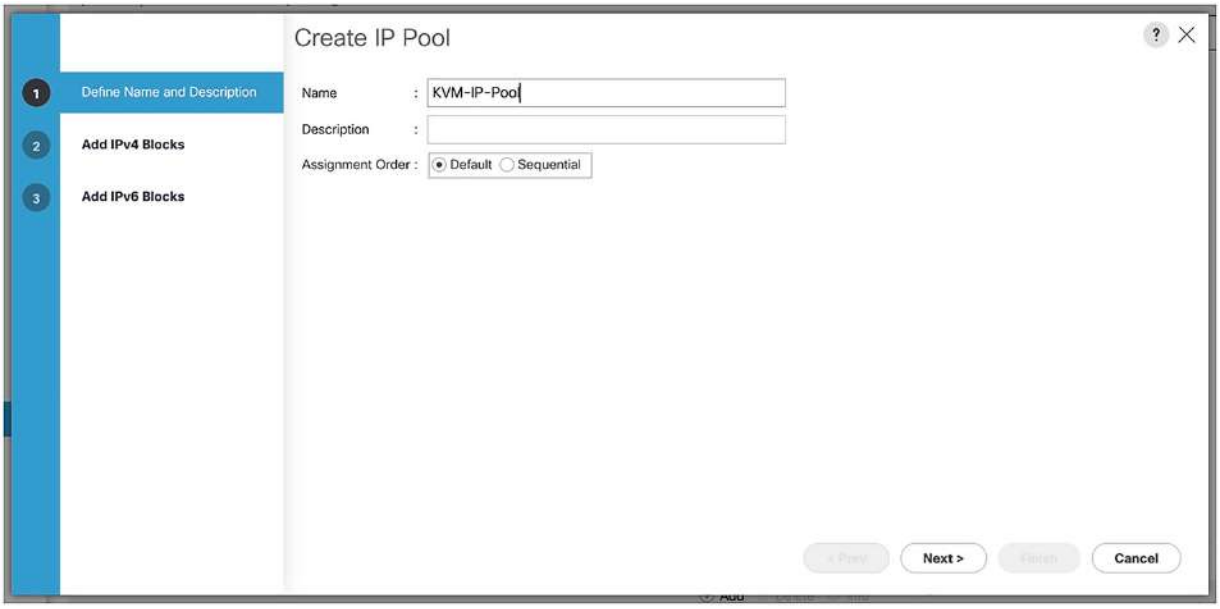

Figure 4-17 The KVM pool

Click “*Next,*” and assign a block of IP addresses (Figure 4-18). This needs to be large enough to cover all the servers you have (and any future ones).

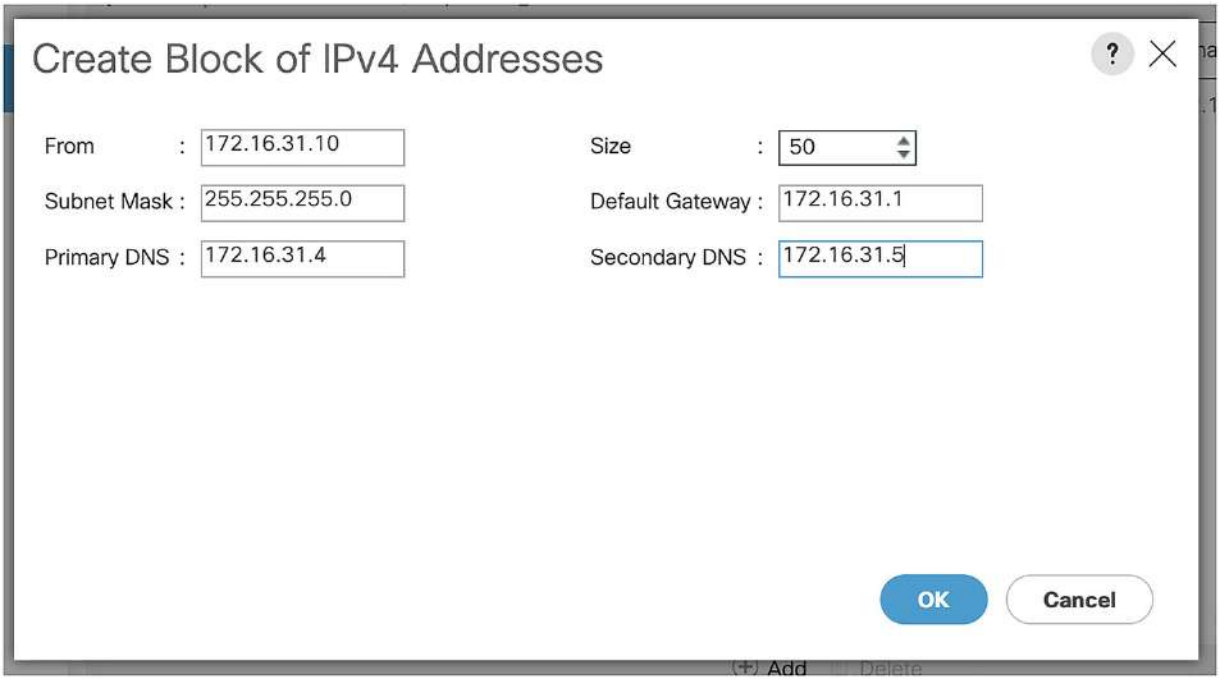

Figure 4-18 The KVM Pool IP range

Pick a range that doesn’t overlap with anything (such as your DHCP scope) otherwise this could cause issues in your environment. The pool will appear in the GUI (Figure 4-19).

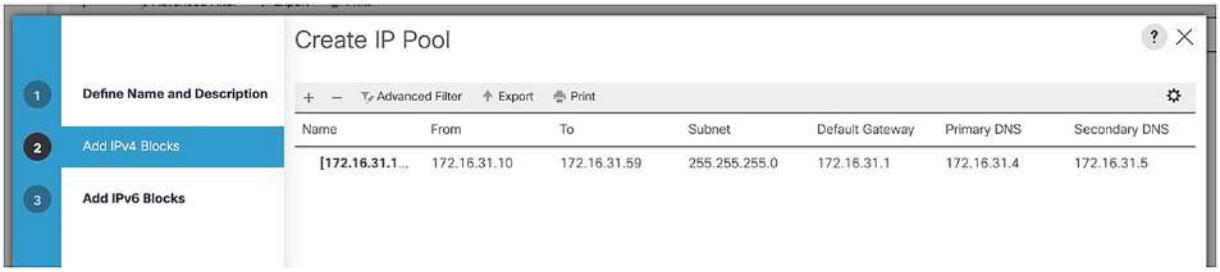

Figure 4-19 The finished IP pool

As we are not adding an IPv6 pool, click “*Next,*” and then click “*Finish.*” Now that we have our port range, we need to specify which port we will be using.

KVM Management Policy

The default KVM port is 2068, but we can change that by going to *Servers ➤ Policies ➤ root ➤ Sub-Organizations ➤ LearningUCS ➤ KVM Management Policy*. Create a new policy called KVM-Port-Policy, setting the port to 3099 (Figure 4-20).

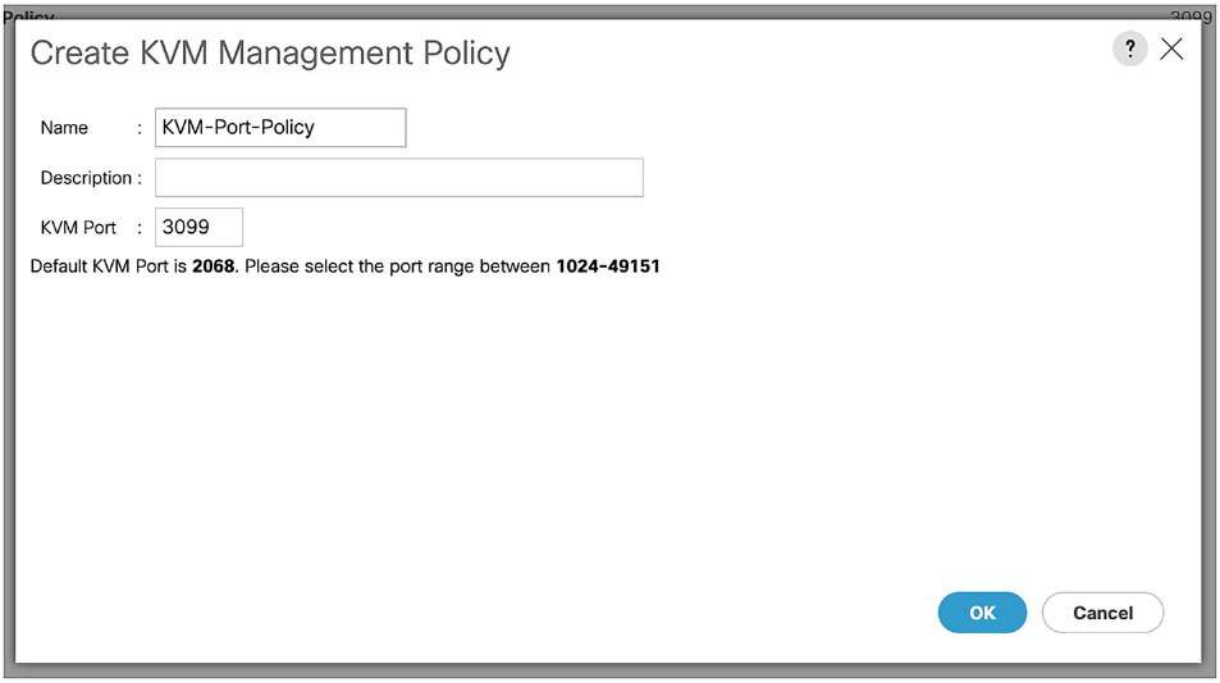

Figure 4-20 The KVM management port policy

Onto our final policy.

Scrub Policies

The last set of policies we are going to implement are scrub policies. These control how the disks on a server will be treated in scenarios such as moving blades. For example, you are balancing the blades in your UCS chassis, evening out three application blades across three chassis. You have arranged the downtime, attached the service profile to the empty destination slot, and removed the blade. When you put it in the new chassis slot, the blade is picked up and

once it's booted up, you find that (due to the configured Scrub policy) the disks have been wiped.

This is where scrub policies will save you. Head to *Servers ➤ Policies ➤ root ➤ Sub-Organizations ➤ LearningUCS ➤ Scrub Policies*. Set all the options to "*No*" (Figure 4-21).

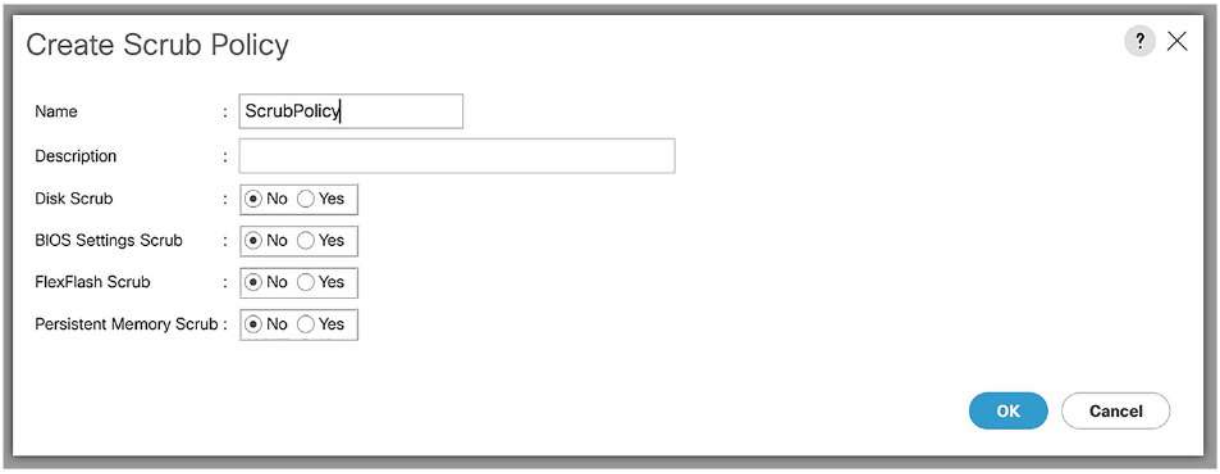

Figure 4-21 Our scrub policy

Now, we can keep our data safe if we move a blade! Before we move onto the next chapter, however, we need to create a few more items, namely our pools and a VSAN

UUID Pool

We need to be able to identify our servers in UCS, well, more specifically, the UCS systems need to identify our servers. While we can name them (by giving them labels) in the UCS GUI, the backend systems have a different way of referencing the servers, and this is through a UUID. The UUID (Unique Identifier) is a 128-bit reference. We can create a pool of UUIDs, saving us from manually assigning them to each of our servers. To create the pool, go to *Servers ➤ Pools ➤ root ➤ Sub-Organizations ➤ LearningUCS ➤ UUID Suffix Pools*. Click "Add" and create a block of 30 UUIDs, as in Figure 4-22.

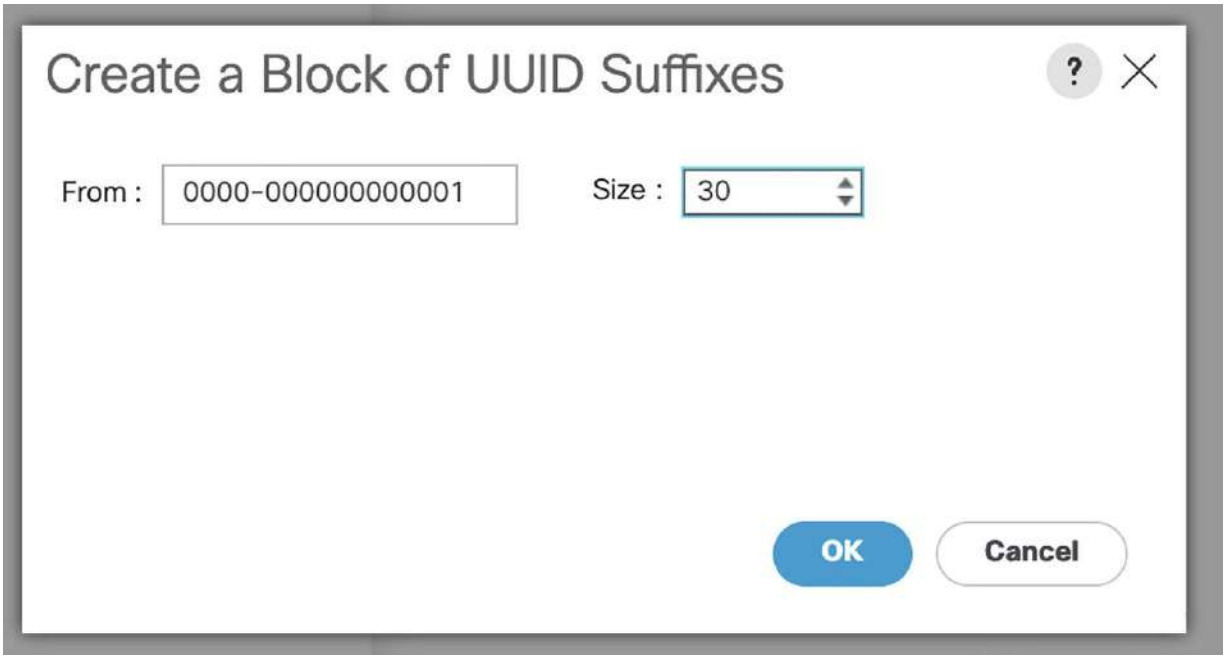

Figure 4-22 Creating a UUID Suffix pool

Once we have created our pool, we can see the sequential suffixes (Figure 4-23).

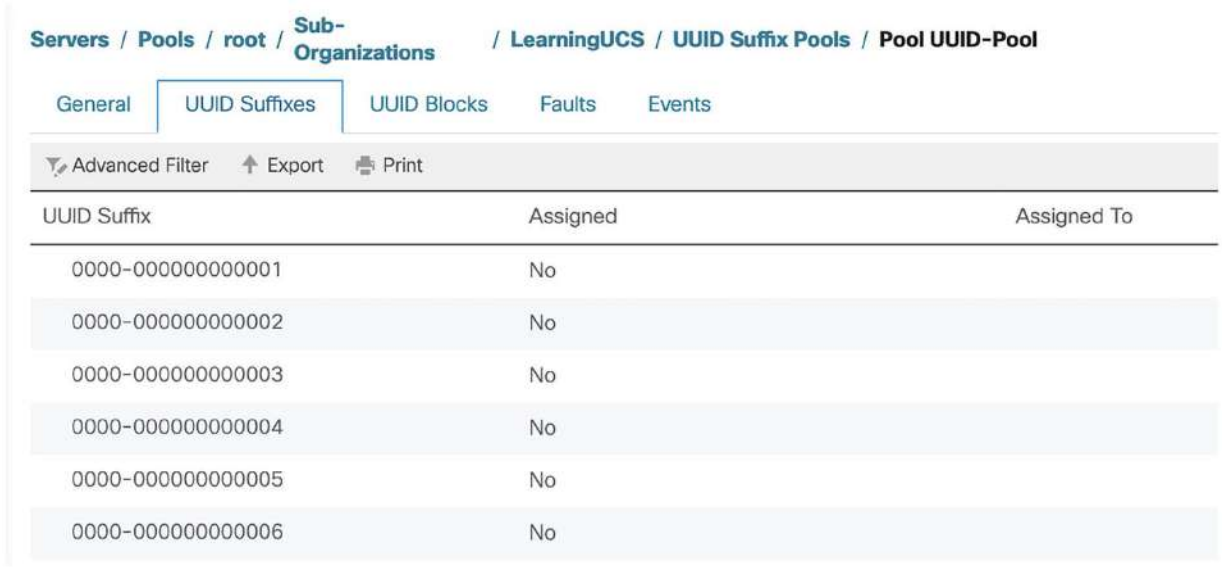

Figure 4-23 Our UUID pool

MAC Pools

In the same way that our servers need a unique identifier, so do our network interfaces. We do this through MAC pools. Navigate to *LAN ➤ Pools ➤ root*. Click “Add” and name the MAC pool (such as “MyMacPool,” as in Figure 4-24).

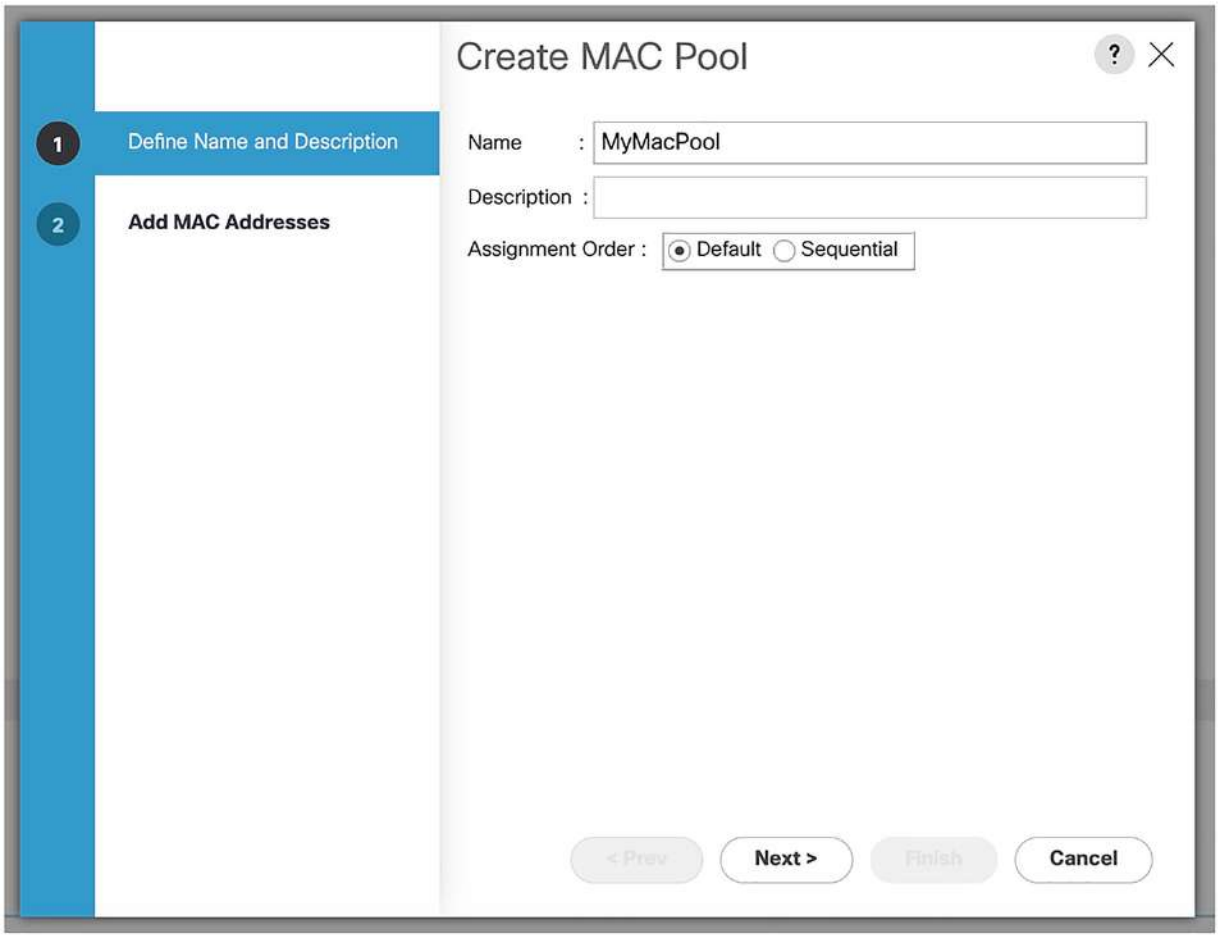

Figure 4-24 MyMacPool

Click “Next” to add the MAC addresses (Figure 4-25). Cisco suggests that the block uses 00:25:B5:xx:xx:xx for compatibility reasons.

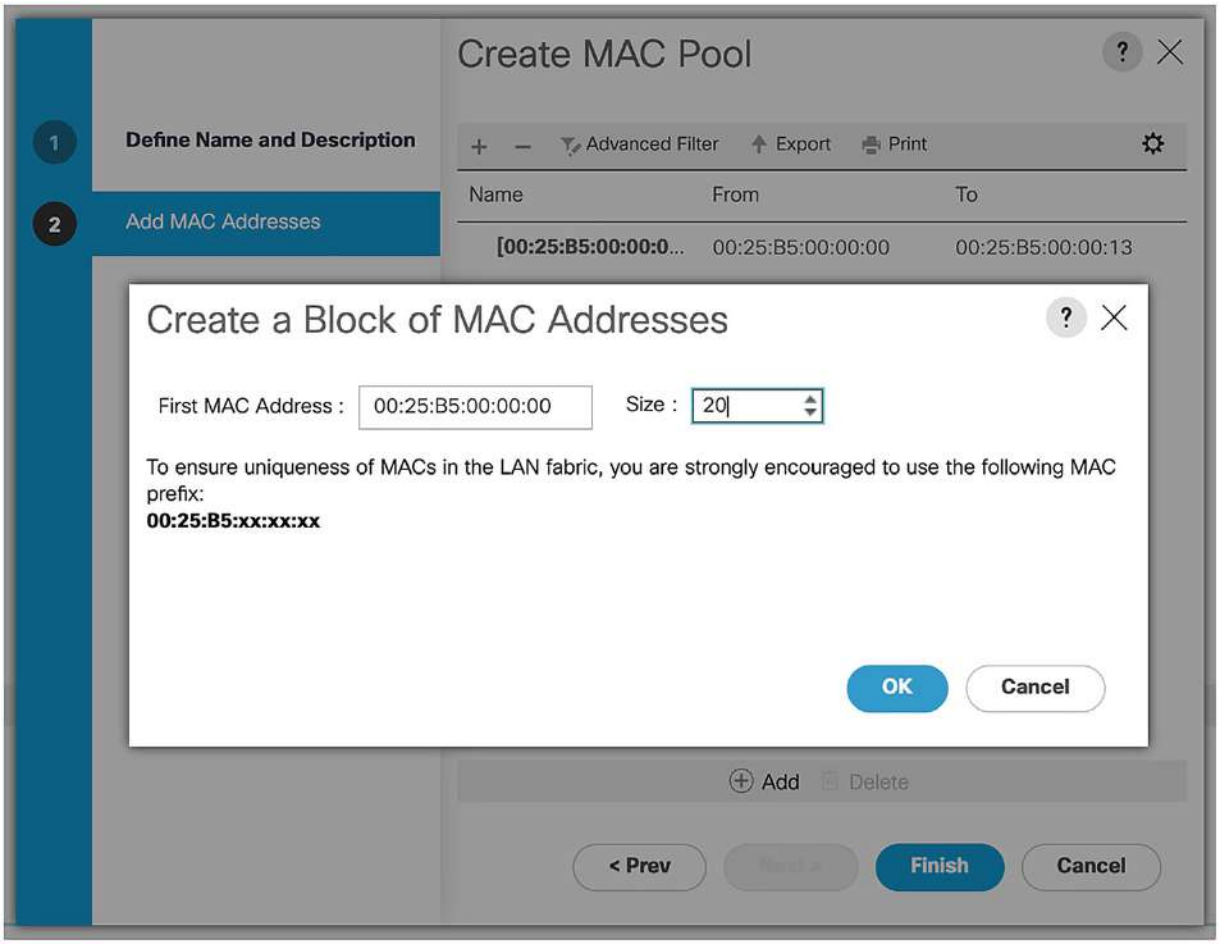

Figure 4-25 Our MAC address block

WWNN

In the same way that we created blocks of IDs for our servers and MAC addresses for our network cards, our SAN fabric will also need some uniqueness. We do this through the WWNN (World Wide Node Names) pool, which has a number of WWNs (World Wide Names). Navigate to "*SAN ➤ Pools ➤ root ➤ Sub-Organizations ➤ LearningUCS*" and right-click WWNN Pools, choosing the option to create a new one. Name the pool "wwnn-pool" (Figure 4-26) and click "*Next.*"

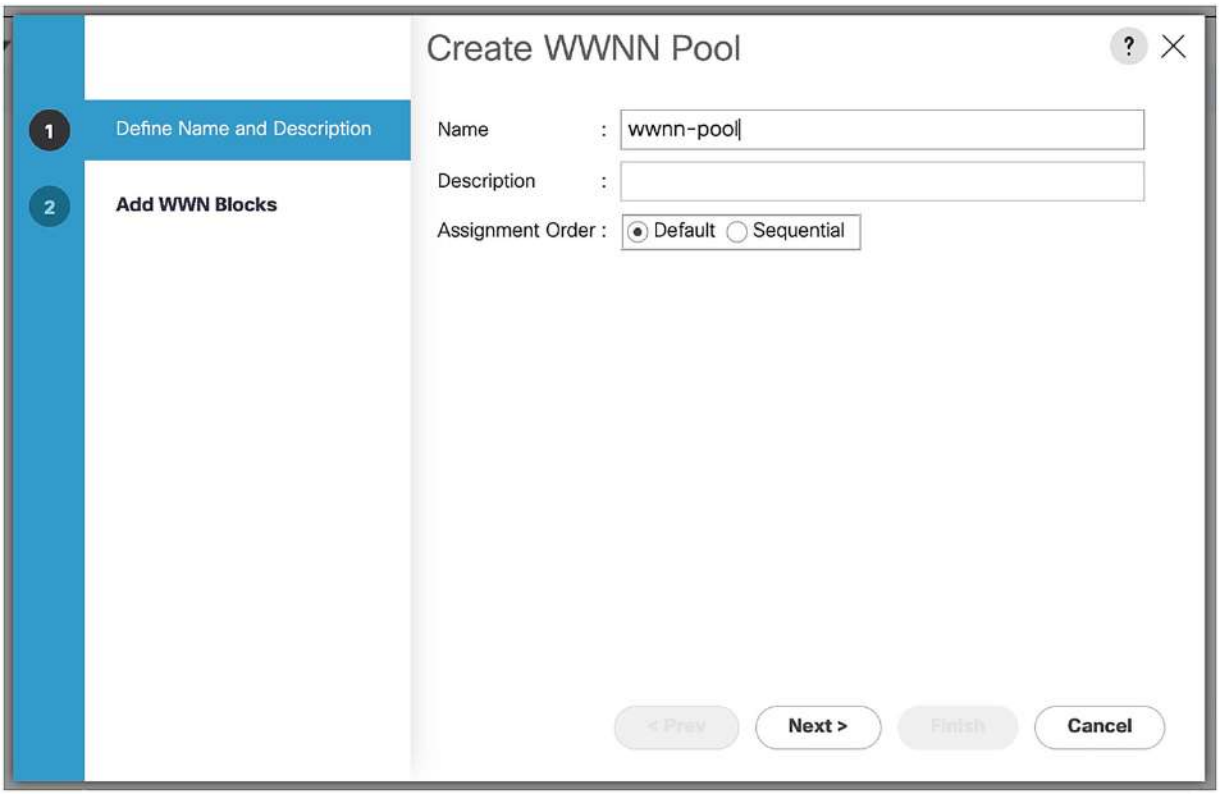

Figure 4-26 Our WWNN pool

Create a block of sixty WWNs, following the naming advice of Cisco (20:00:00:25:b5:xx:xx:xx), as shown in Figure 4-27.

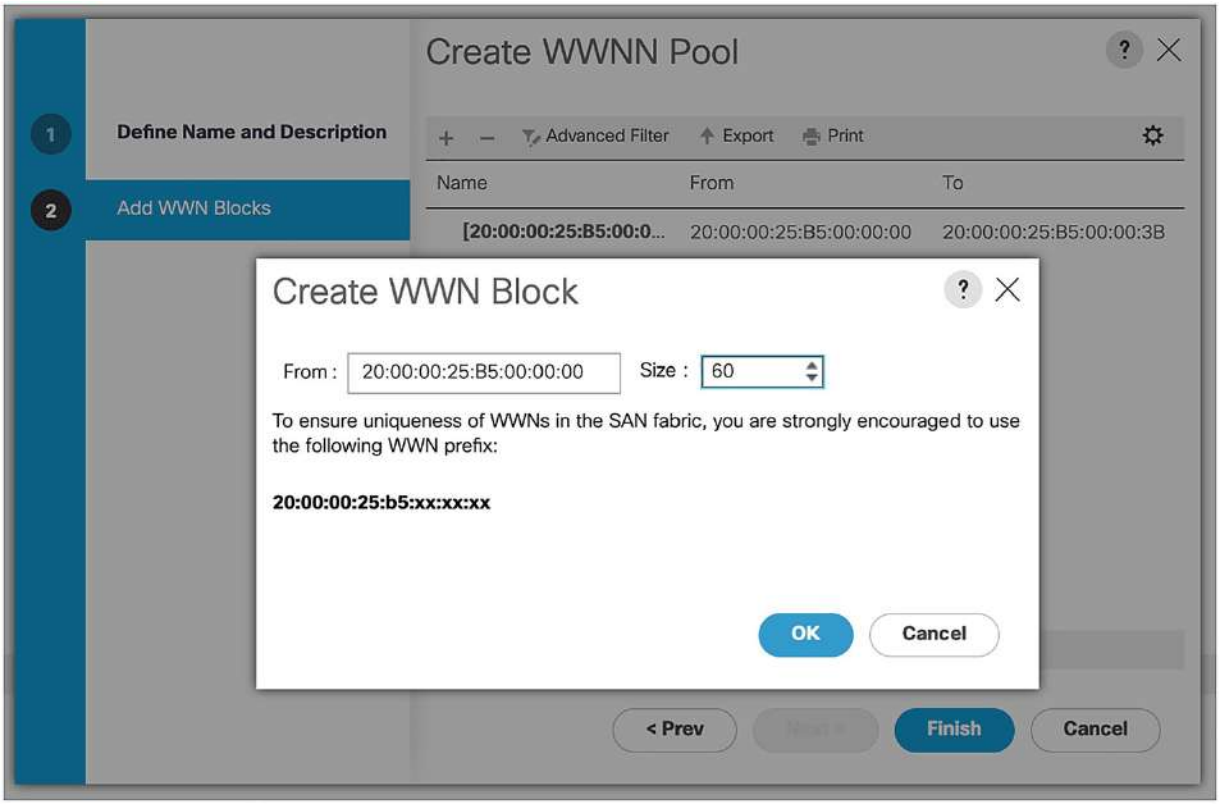

Figure 4-27 Our WWN block

VSAN

The last component we are going to create is our VSAN. This will enable us to separate our storage traffic. We will be using the 2000 and 2001 as our VSAN and FCoE (Fibre Channel over Ethernet) IDs (as these are the ones Cisco suggests), as shown in Figure 4-28.

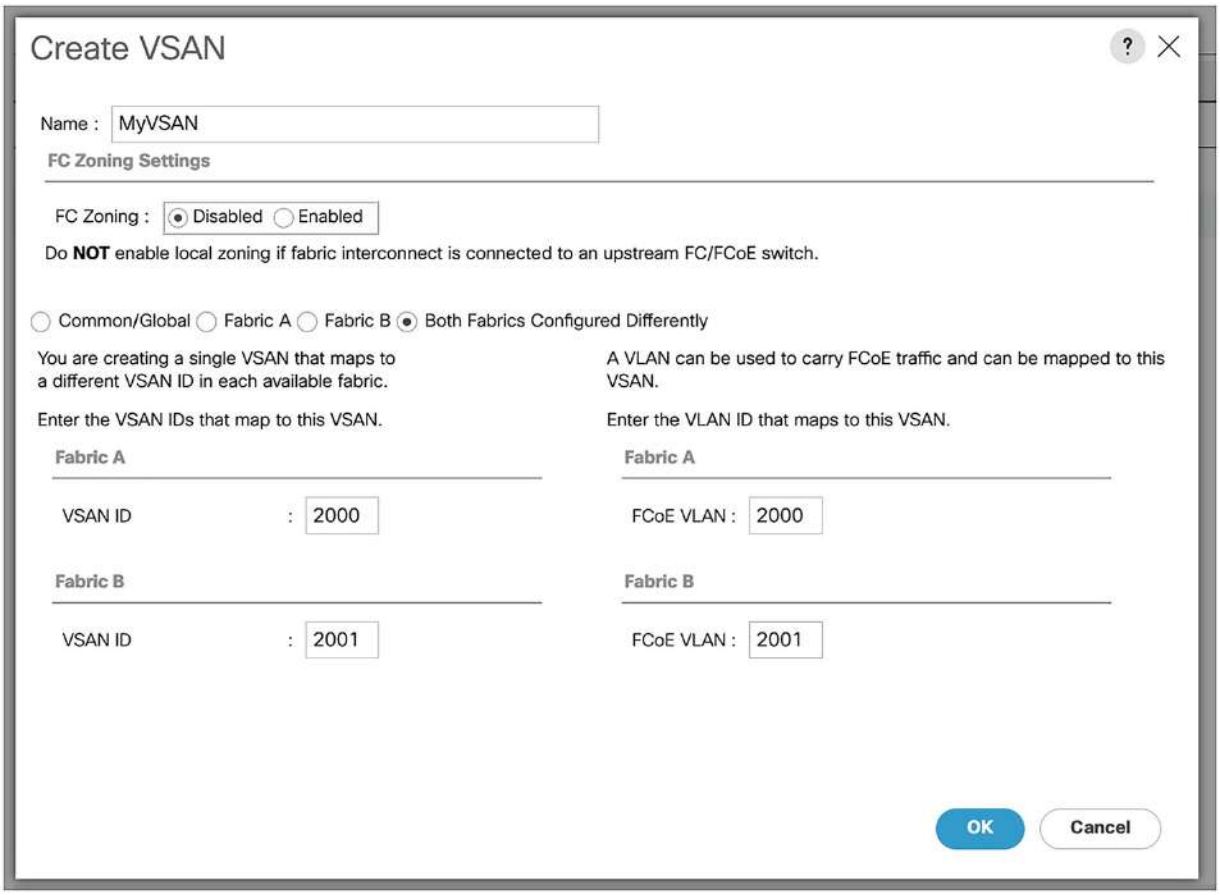

Figure 4-28 Our VSAN

Summary

In this chapter, we have created the policies and pools to control our servers. In the next chapter, we will start assigning these to our servers.

S. Fordham, *Introducing Cisco Unified Computing System*
https://doi.org/10.1007/978-1-4842-8986-0_5

5. Service Profiles and Templates

Stuart Fordham[1] ✉
(1) Bedfordshire, UK

In this chapter, we will be creating a Service Profile template from the pools and policies we created in Chapter 4.

Creating Service Profile Templates

Navigate to “*Servers ➤ Service Profile Templates ➤ root ➤ Sub-Organizations ➤ LearningUCS ➤ Service Profiles ➤ Service Profiles*” and click “*Add*” (Figure 5-1).

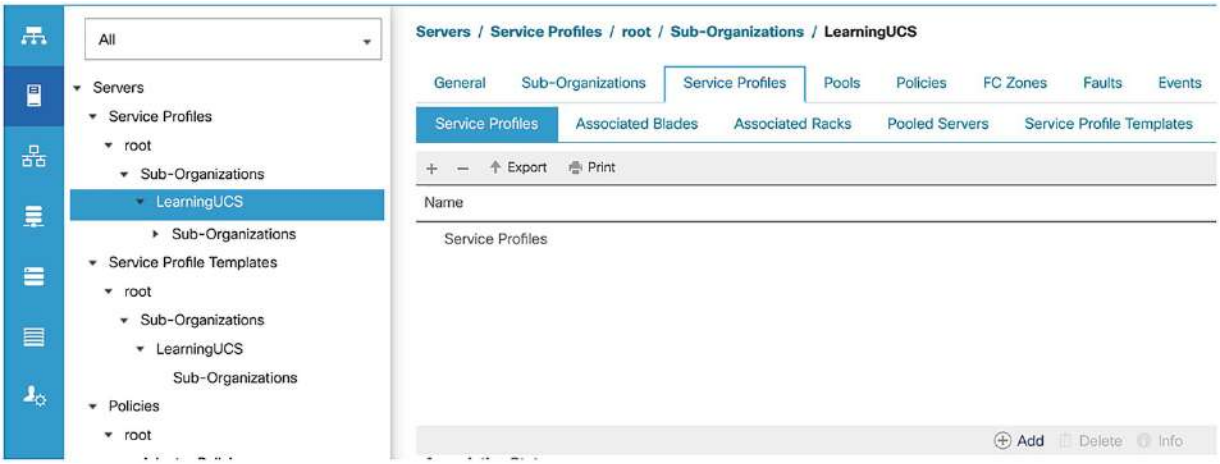

Figure 5-1 The Service Profile page

Choose the expert option from the pop-up menu (Figure 5-2).

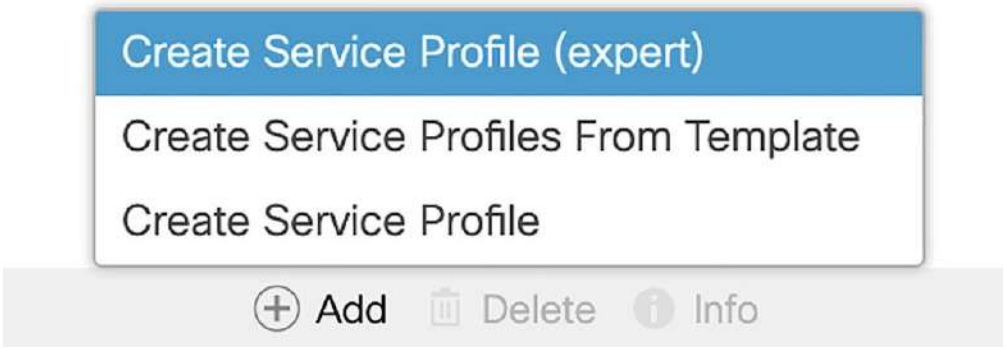

Figure 5-2 Choose the expert option

Name the service profile “*B200-template*” and from the UUID Assignment drop-down, select the UUID-Pool (Figure 5-3).

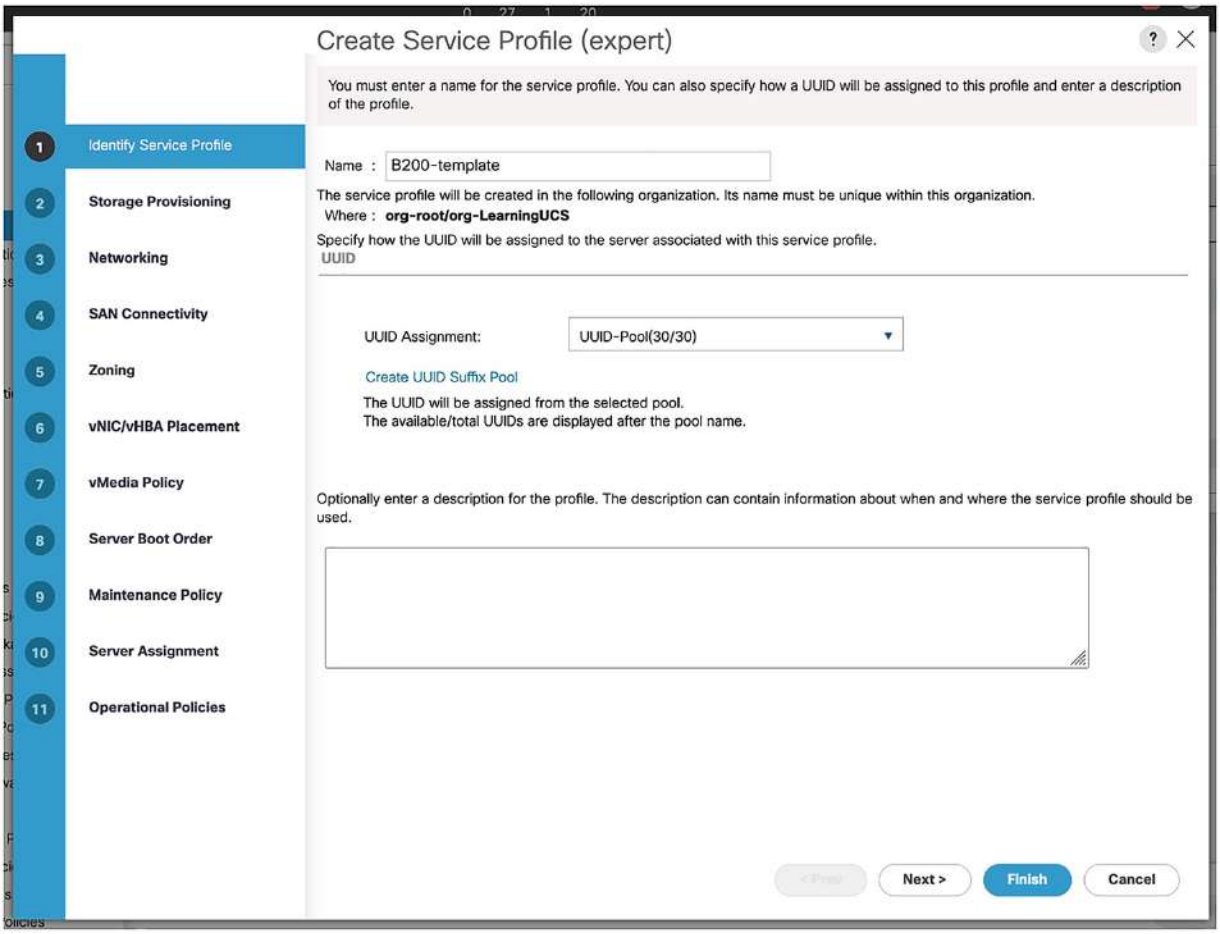

Figure 5-3 Naming the template

Click “*Next*” to move on to the next page. On this page, we will set our storage options, we can select a storage profile, or a policy, or a local disk configuration policy, which is the option we will use (Figure 5-4).

Select “*LocalDiskPol*” from the dropdown box.

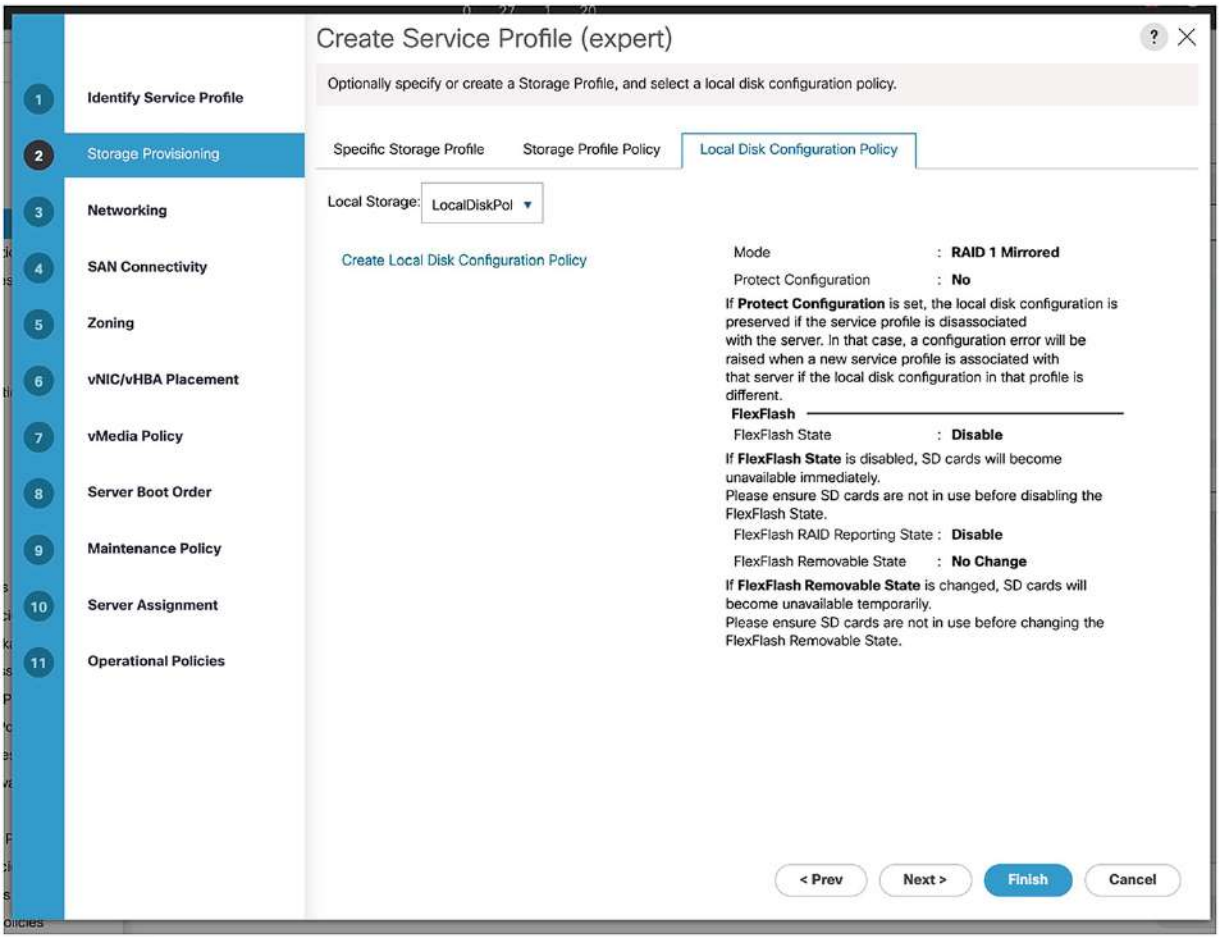

Figure 5-4 Selecting the LocalDiskPol

Click "*Next*" to move on to the Networking page.

We did not create a dynamic vNIC policy in the last chapter, but we did create some new VLANs. Assign eth0 to the DMZ VLAN, and eth1 to the DB VLAN (Figure 5-5).

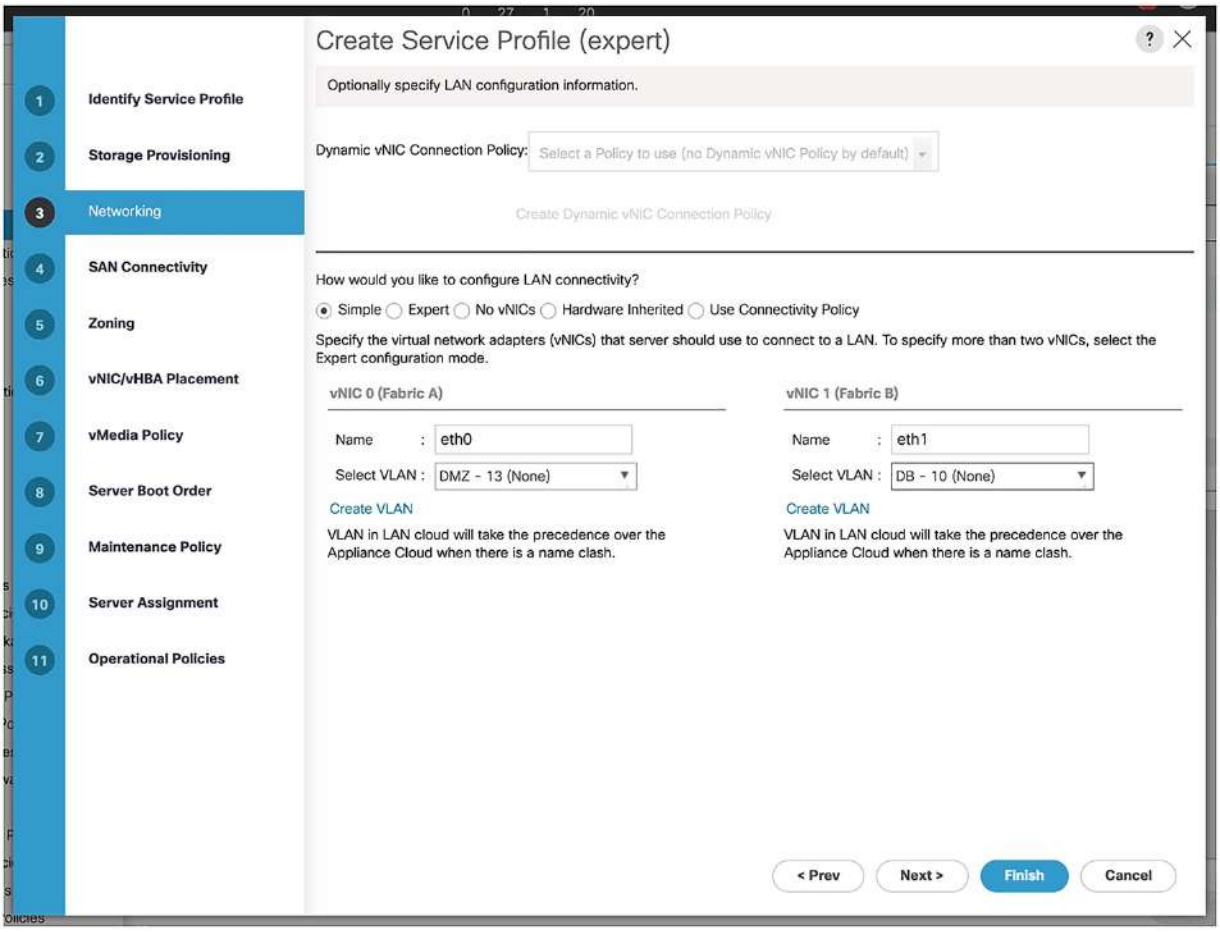

Figure 5-5 The networking options

Clicking Next will take us to the SAN connectivity page. Here, we will assign the wwnn-pool we previously created (Figure 5-6).

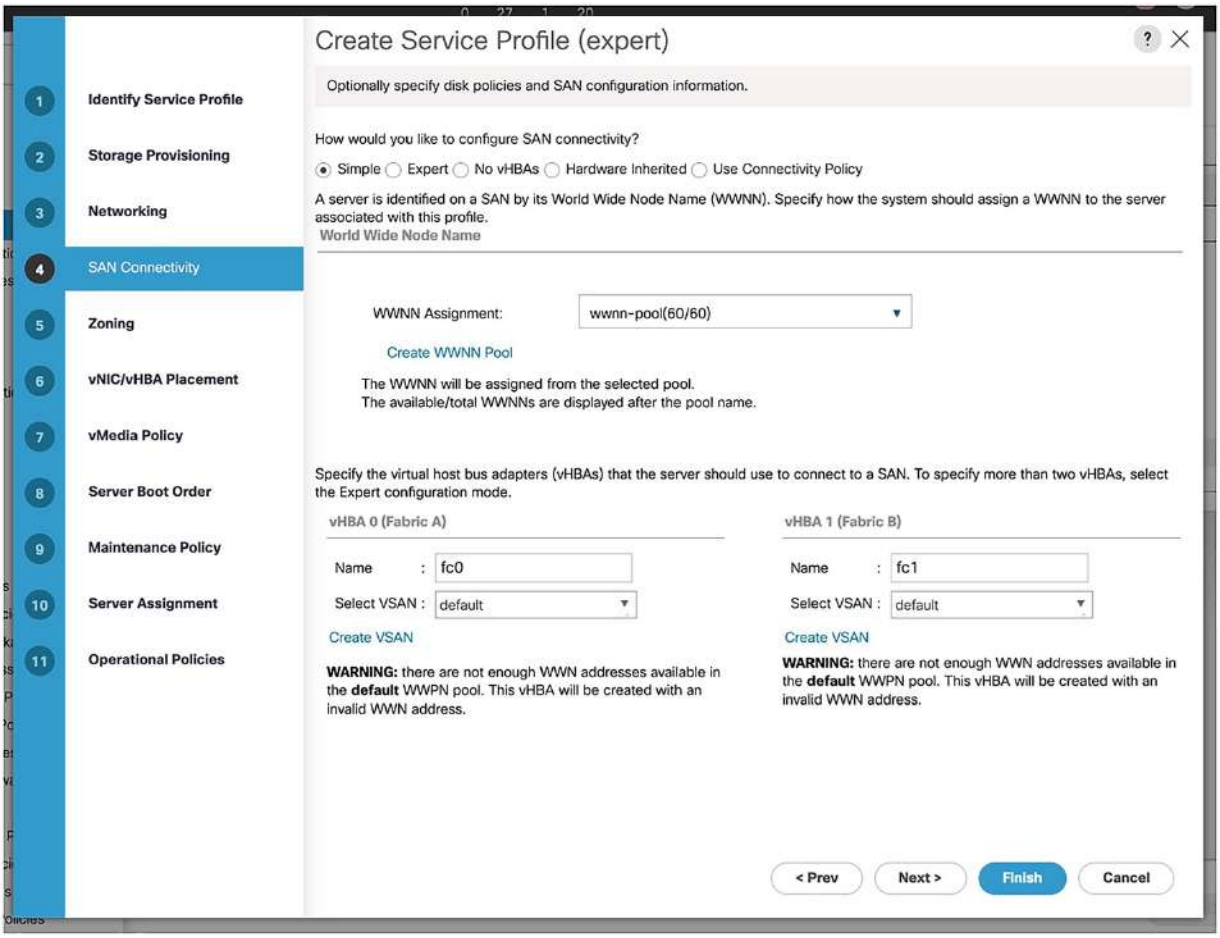

Figure 5-6 The SAN connectivity page

Click on “Next” to move forward into the zoning tab (Figure 5-7). We are not going to set up any HBA zoning here, so click “Next” again.

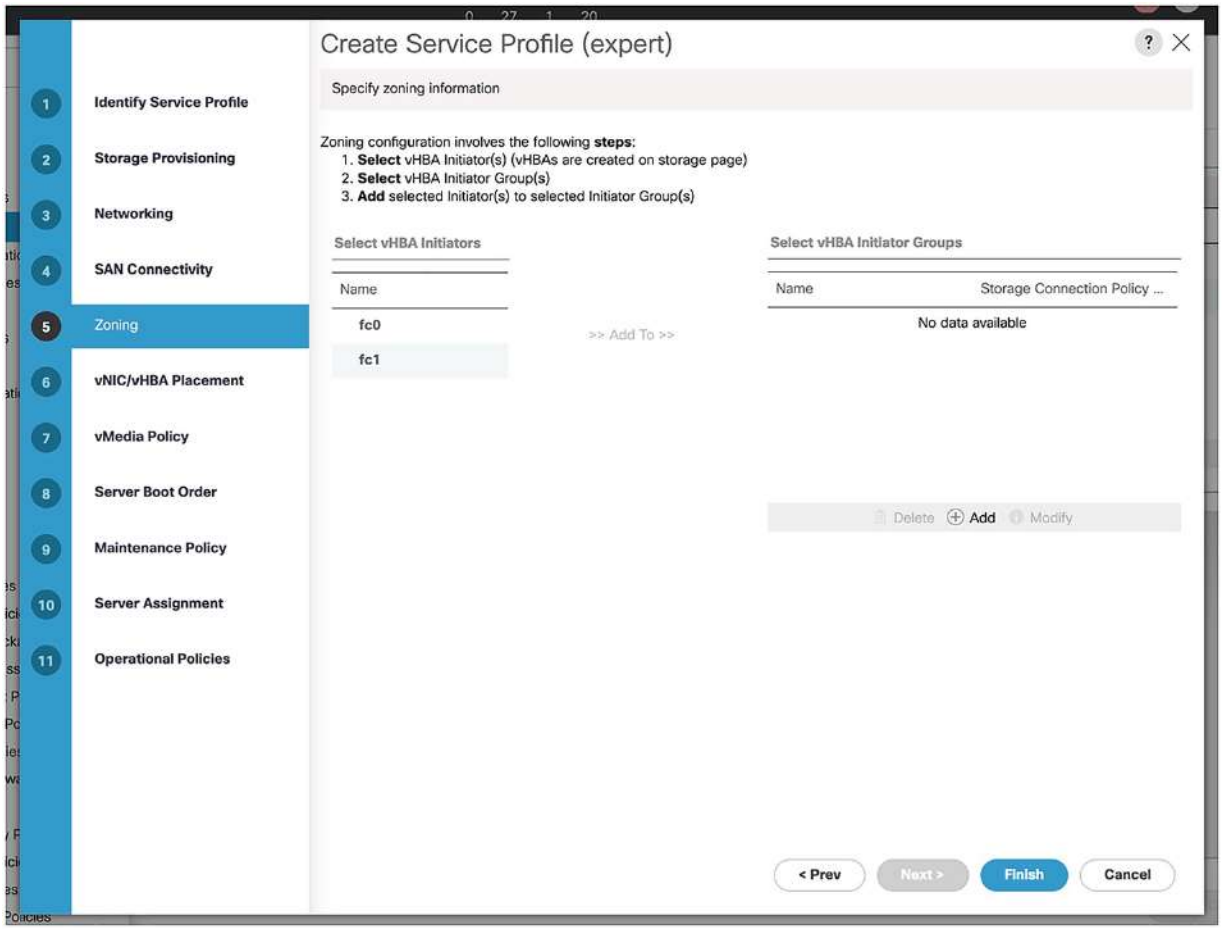

Figure 5-7 The zoning options

On the vNIC/vHBA Placement tab, we can choose how our network interfaces get their placement (Figure 5-8).

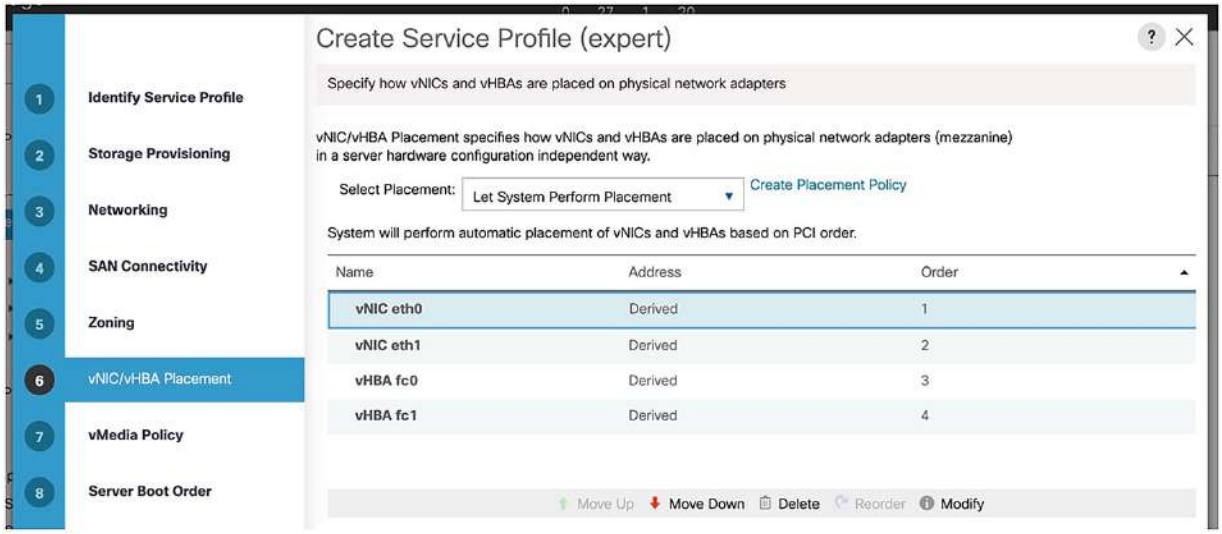

Figure 5-8 Network interface placement

Select “vNIC eth0” and click “Modify” (Figure 5-9). Select “*MyMACPool*” for the MAC address assignment, assign eth0 to the DMZ VLAN, then click “OK.”

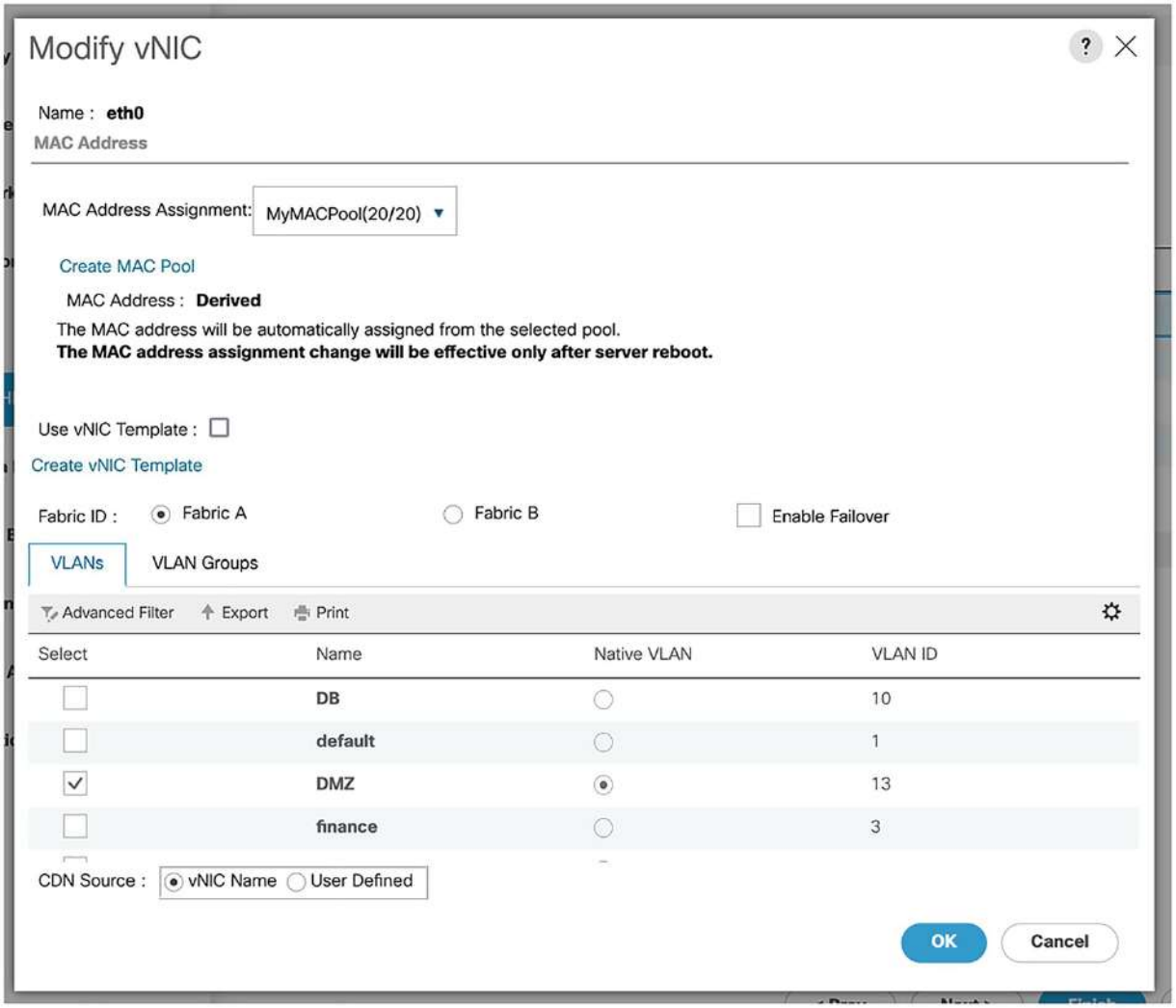

Figure 5-9 Configuring eth0

Do the same for eth1, assigning it to the DB VLAN (Figure 5-10).

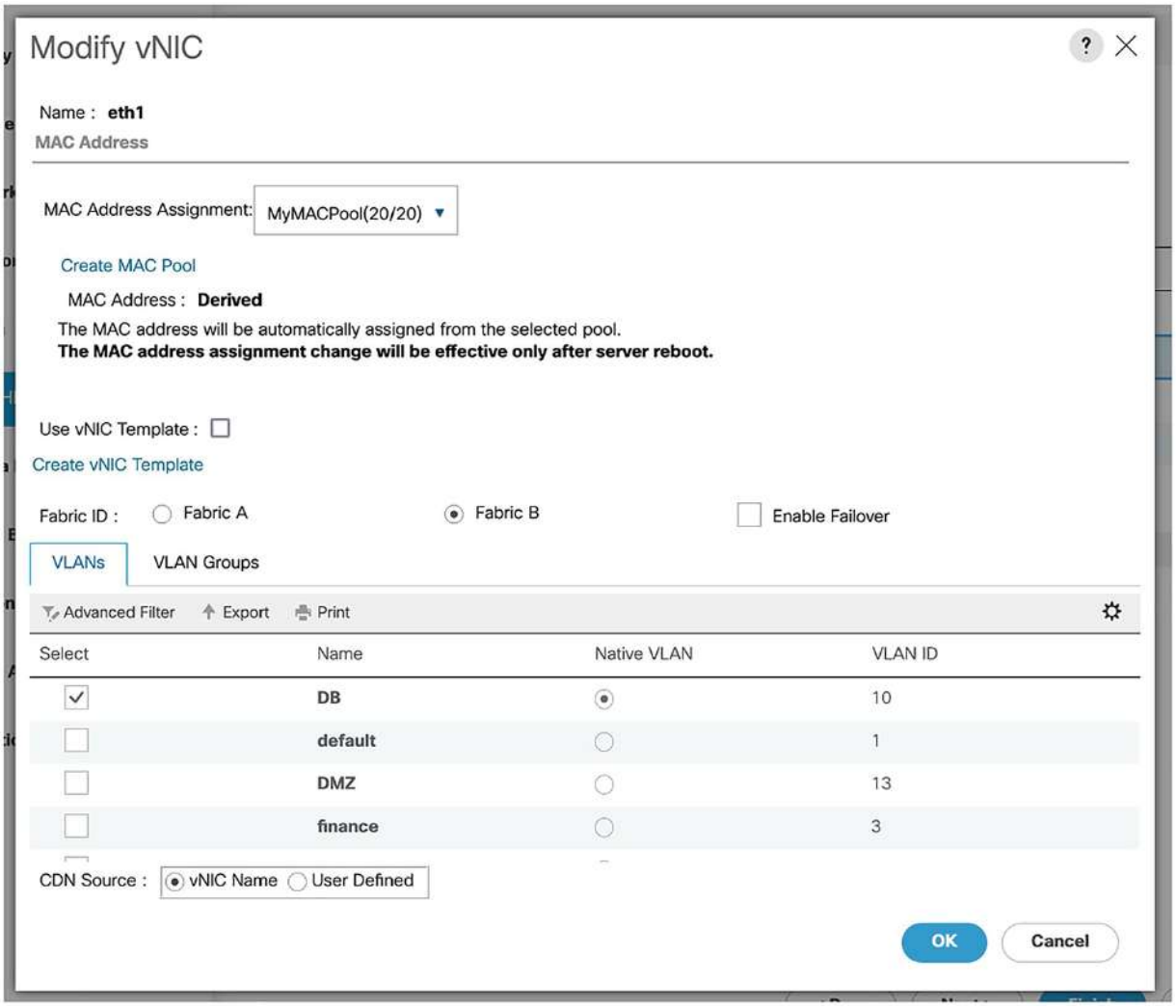

Figure 5-10 Configuring eth1

For the fc0 and fc1 interfaces, assign them to the WWPN pool and MyVSAN (Figures 5-11 and 5-12).

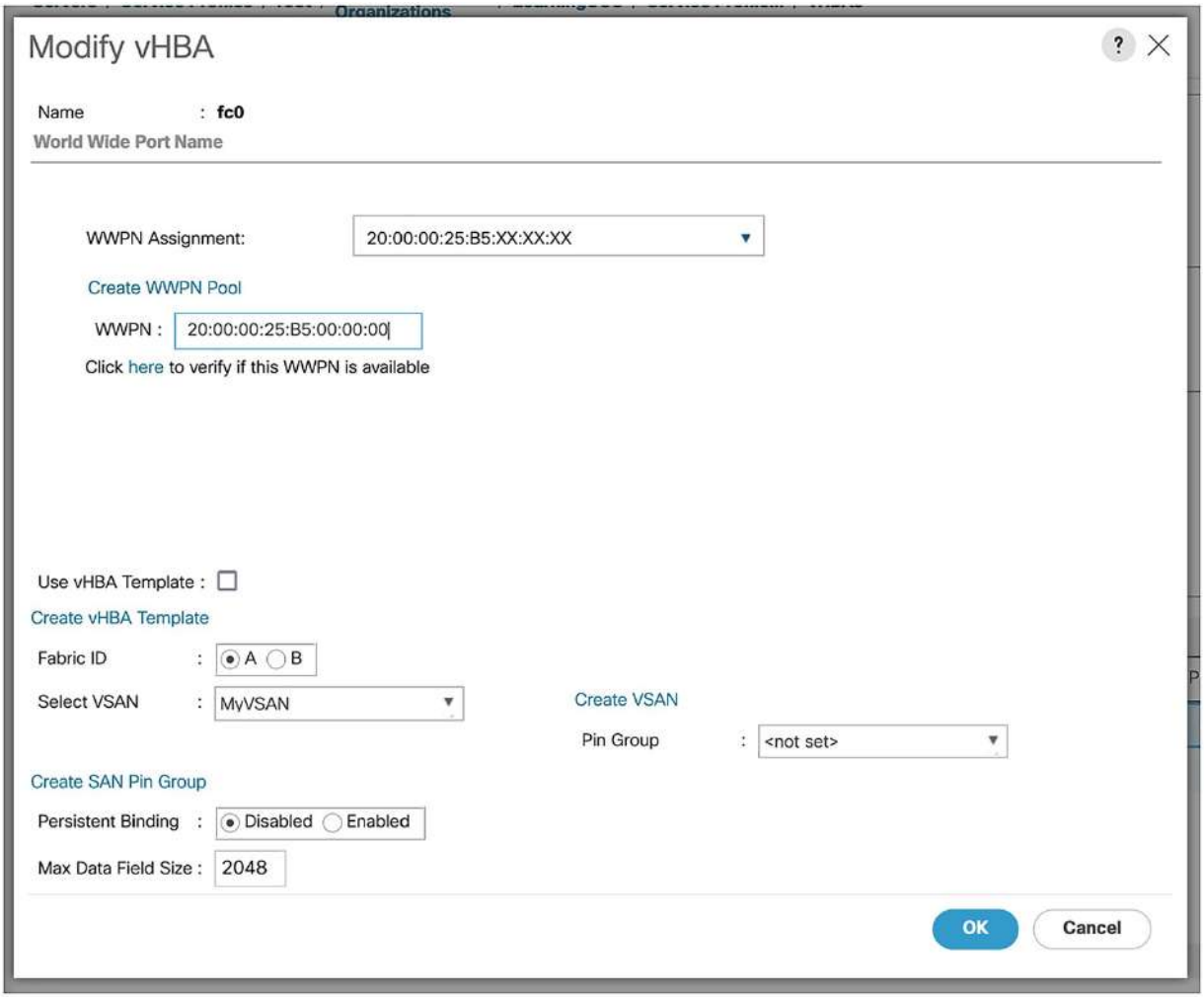

Figure 5-11 Configuring fc0

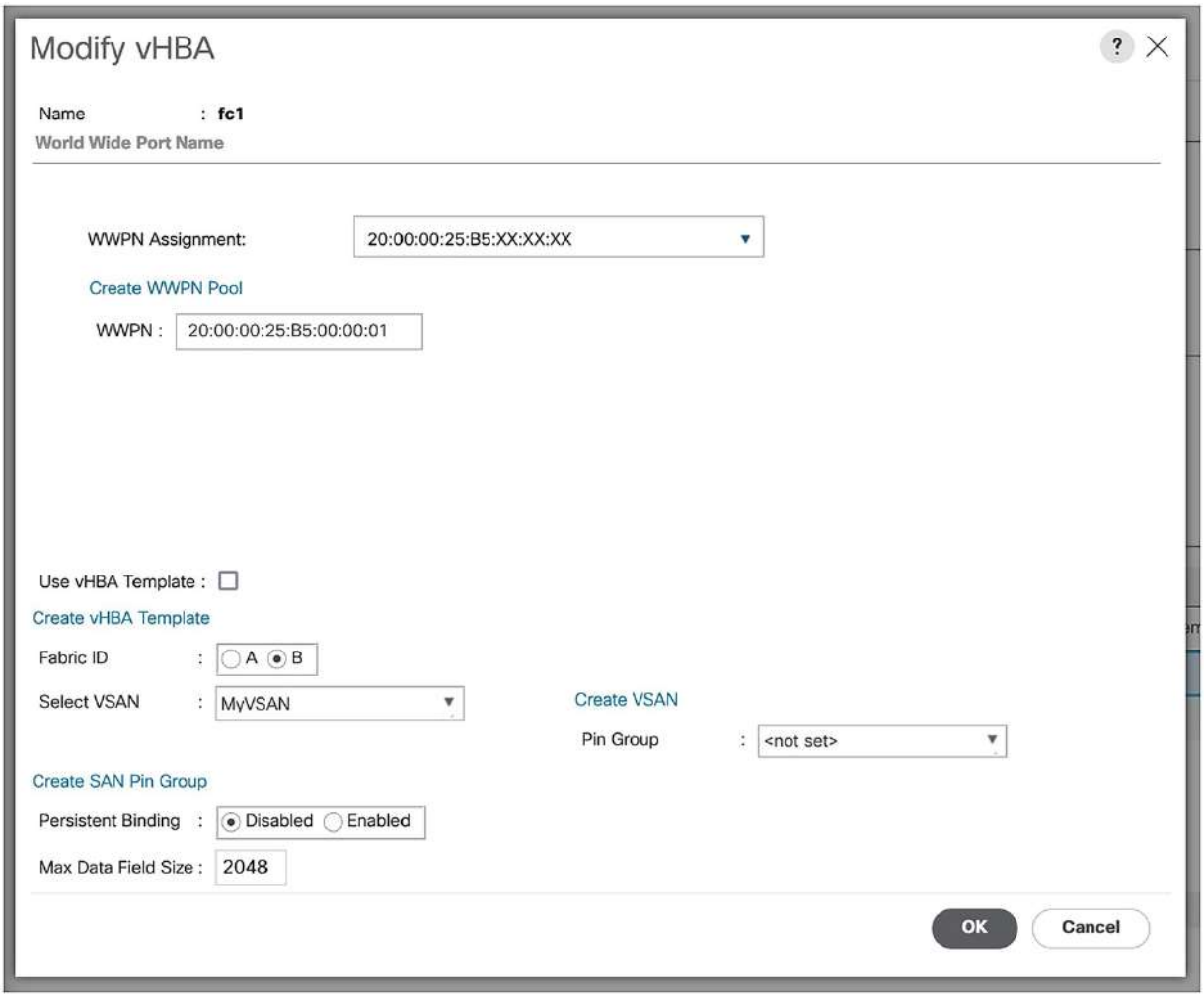

Figure 5-12 Configuring fc1

Click "*Next*" once all four interfaces have been configured.

The next screen is the vMedia tab. Here we will assign the Linux vMedia policy we created (Figure 5-13), which will set up the Linux.iso image to be used when we boot our server (at least it would work in a real environment, but we'll just have to pretend in UCSPE).

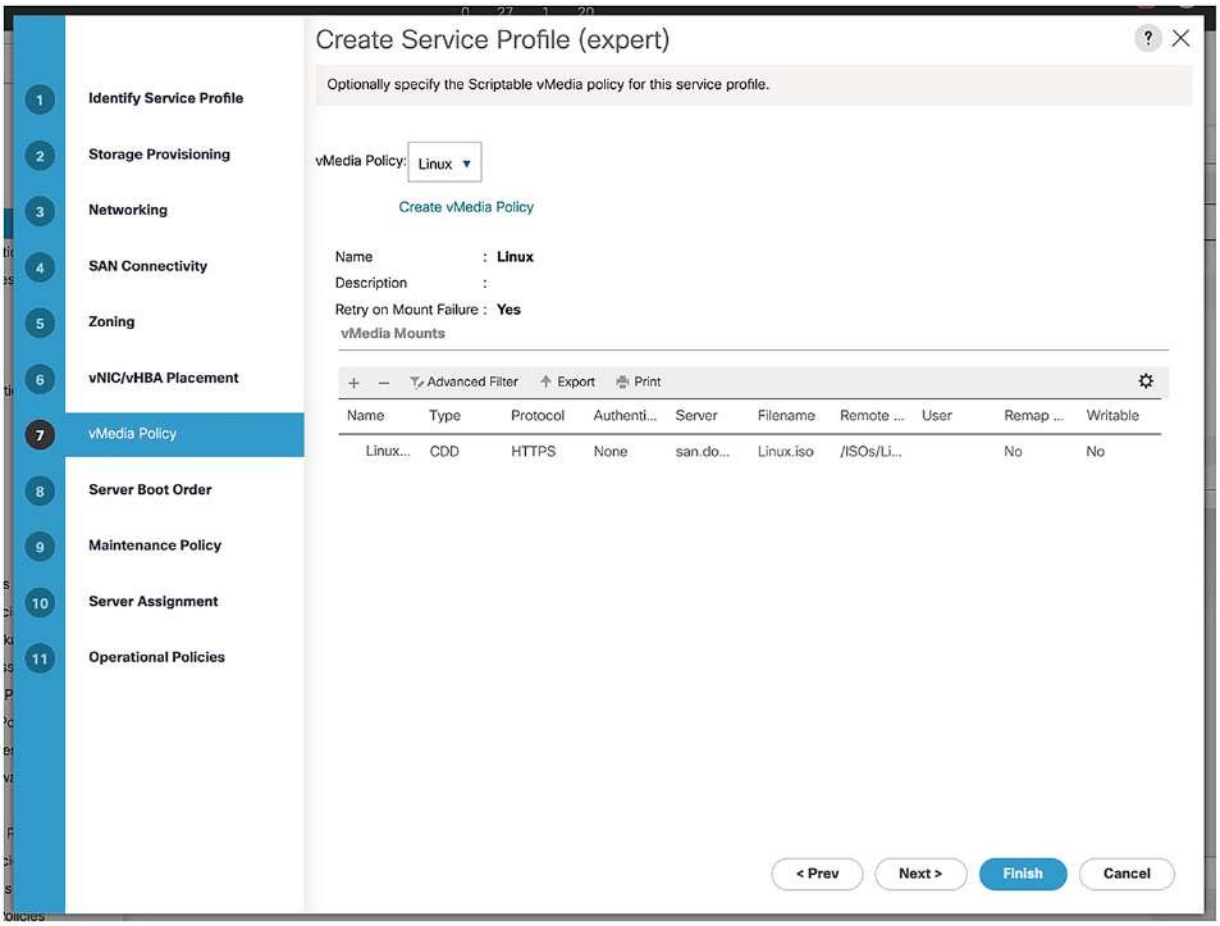

Figure 5-13 Setting the vMedia policy

The next tab is where we set our server boot order (Figure 5-14), first trying the CIMC mounted CD or DVD, and then the local hard disk. Select "*BootPolicy*" from the drop-down menu.

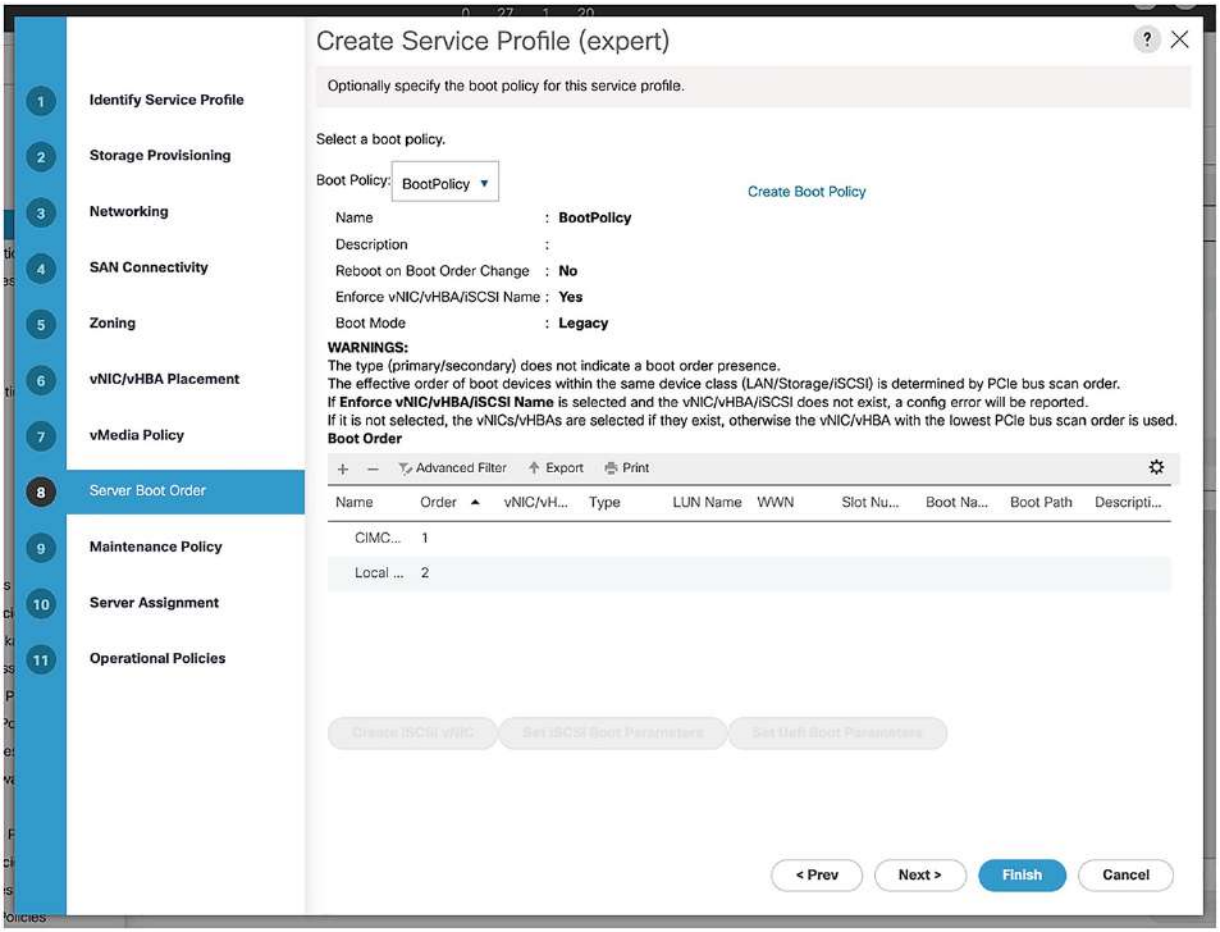

Figure 5-14 The server boot order

Clicking "*Next*" takes us to the Maintenance Policy window. Here, we are going to select "*ServerMaintPol*" from the drop-down, as shown in Figure 5-15.

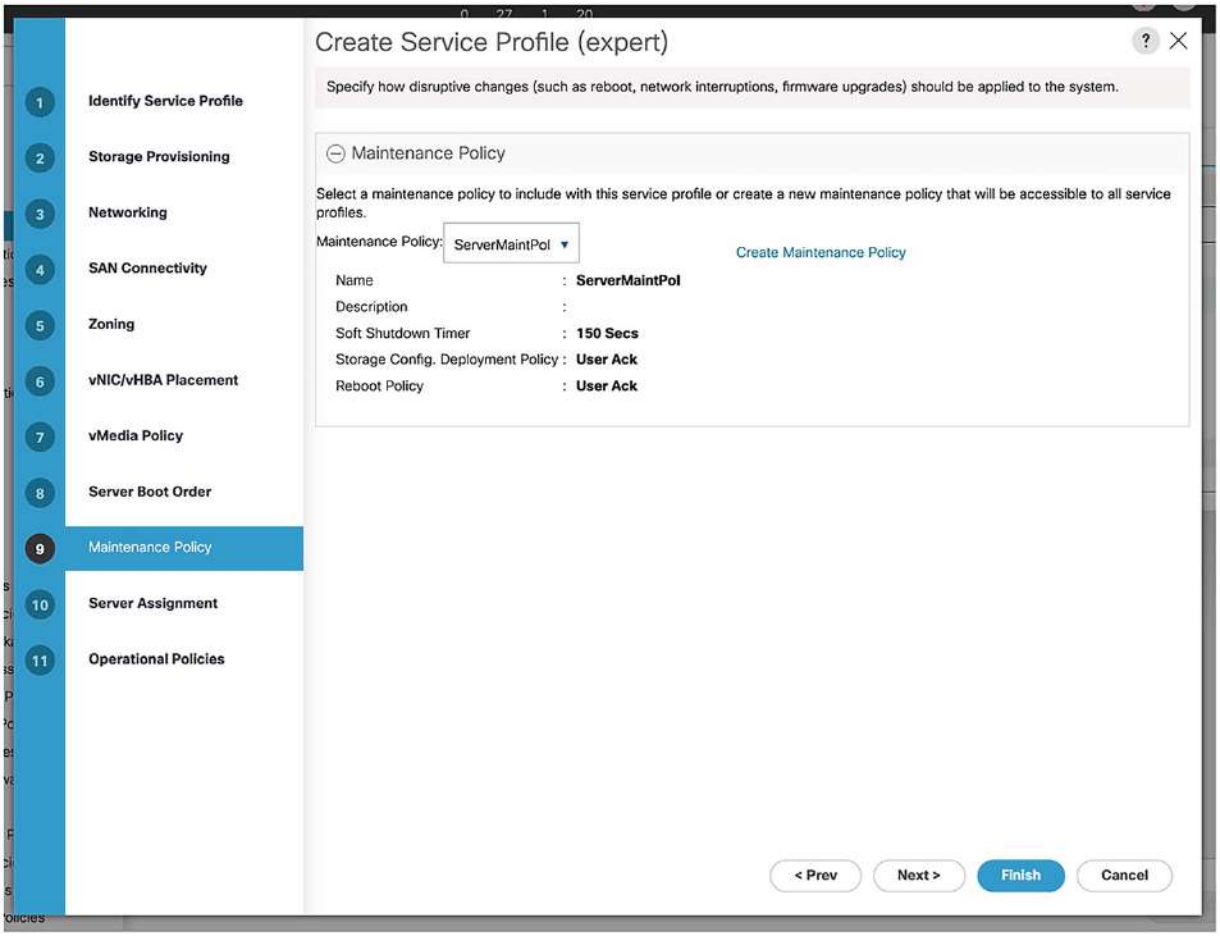

Figure 5-15 The server maintenance policy

Our penultimate window is for server assignment. We can choose to assign our template to a server, as well as set our firmware policies to control our BIOS, disk controllers, and adaptors (Figure 5-16).

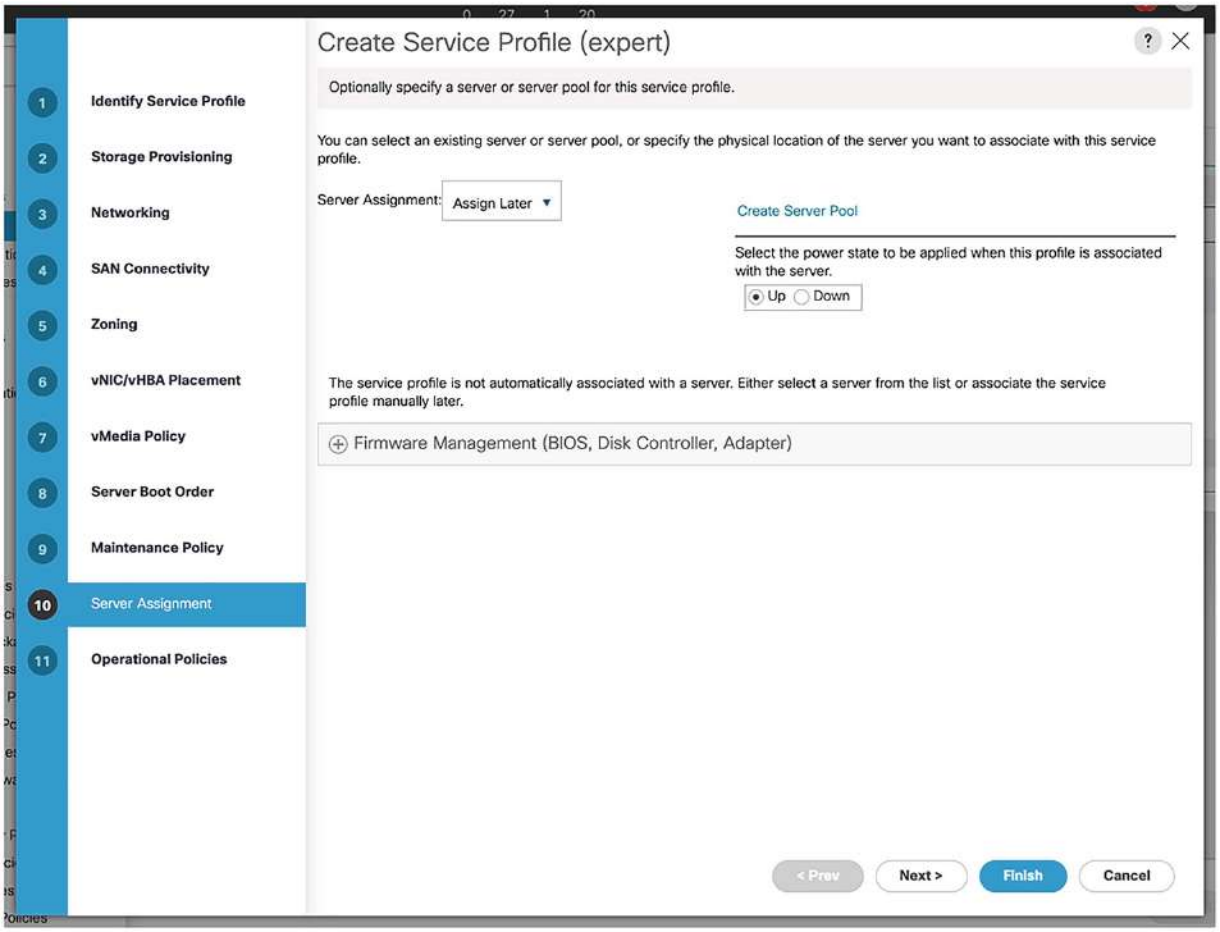

Figure 5-16 Firmware policies

The final window is where we set our operational policies, such as the KVM-IP-Pool for our management IP addresses (Figure 5-17).

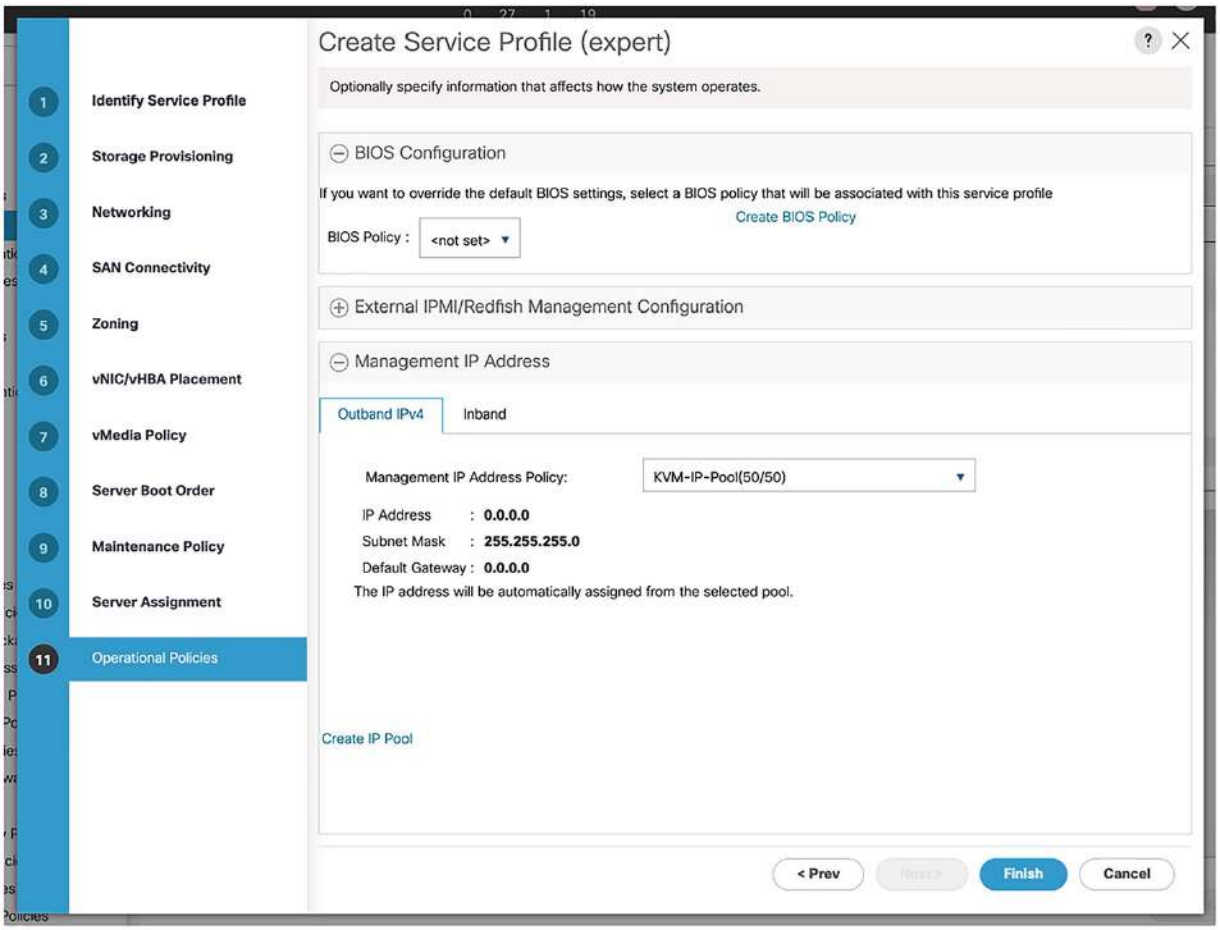

Figure 5-17 Management IP address policy

This is where we also set our scrub and management port policies, which are "*ScrubPolicy*" and "*KVM-Port-Policy,*" respectively (Figure 5-18).

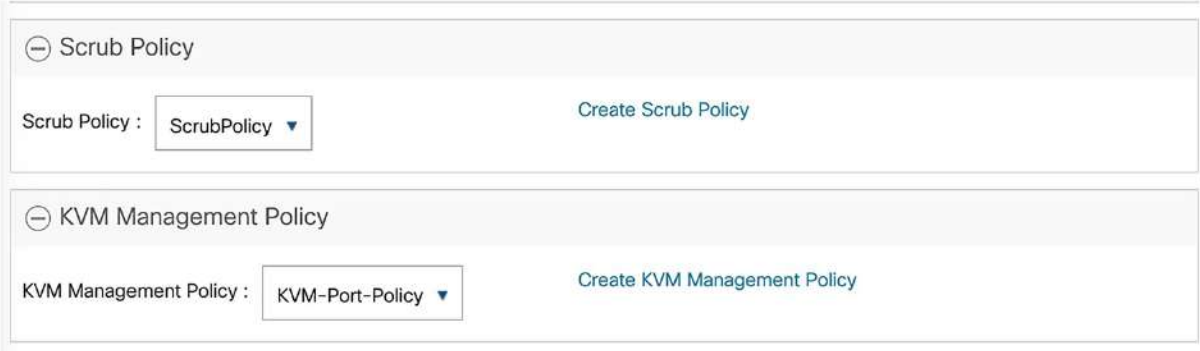

Figure 5-18 Scrub and KVM port policies

Once we have finished setting these, click "*Finish*." We should receive a notice that our service profile has been successfully created (Figure 5-19).

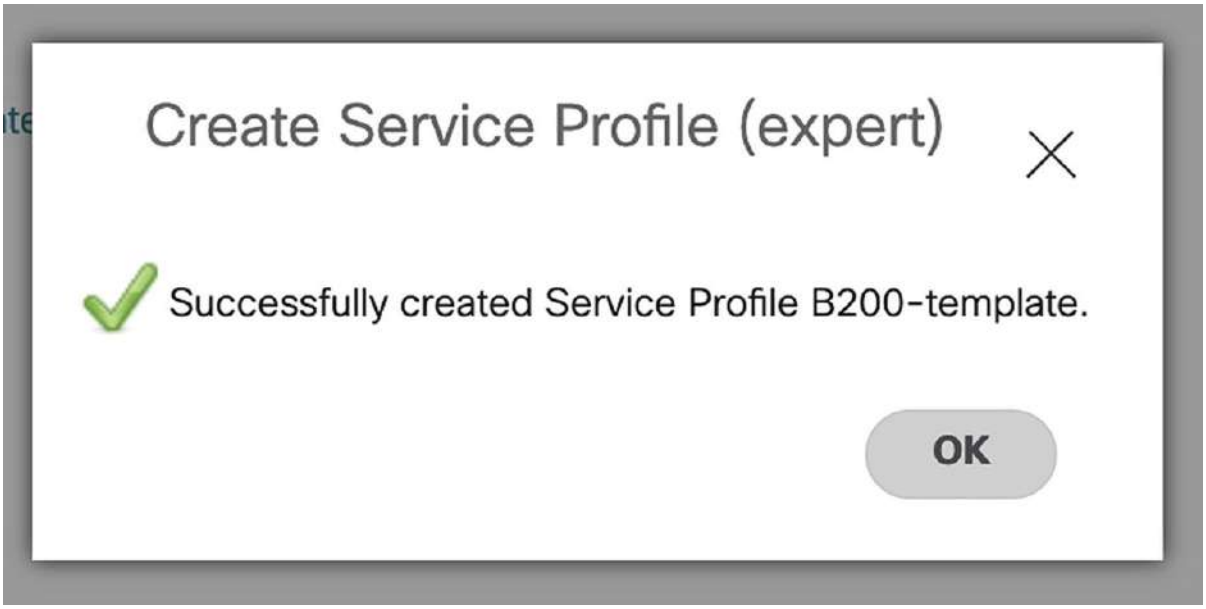

Figure 5-19 Our template is complete

Now that we have a template, the next logical step is to associate this with a server. Select the first server in the first chassis, which, in this instance, is Server 1 in Chassis 3 (*Equipment ➤ Chassis ➤ Chassis 3 ➤ Server 1*). Select the server and in the actions pane, we can see an option for "*Associate Service Profile*" (Figure 5-20).

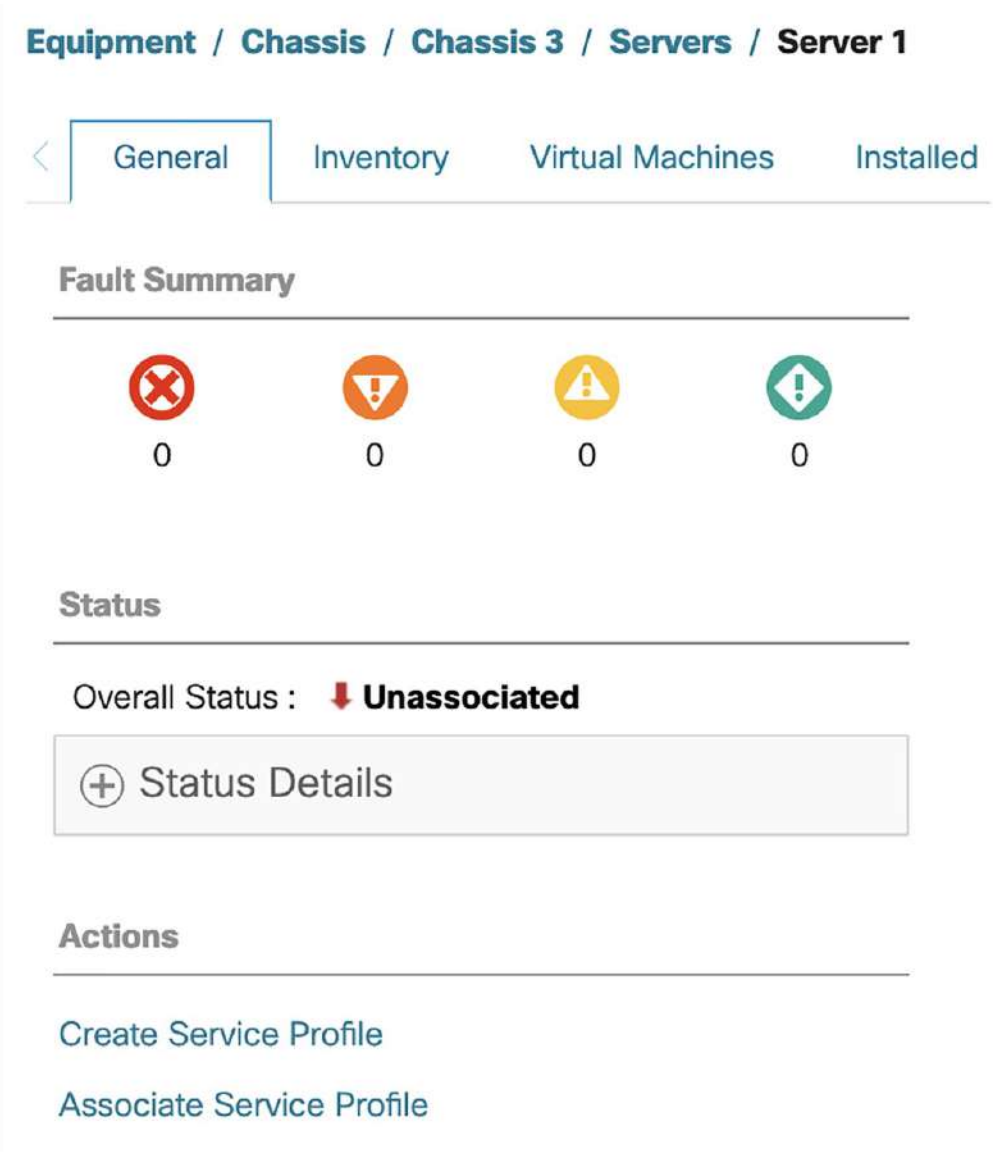

Figure 5-20 Associating a template to a server

Clicking on this link will bring up a new window in which we can select from the service profiles that we have created. Select our template, and click “OK” (Figure 5-21).

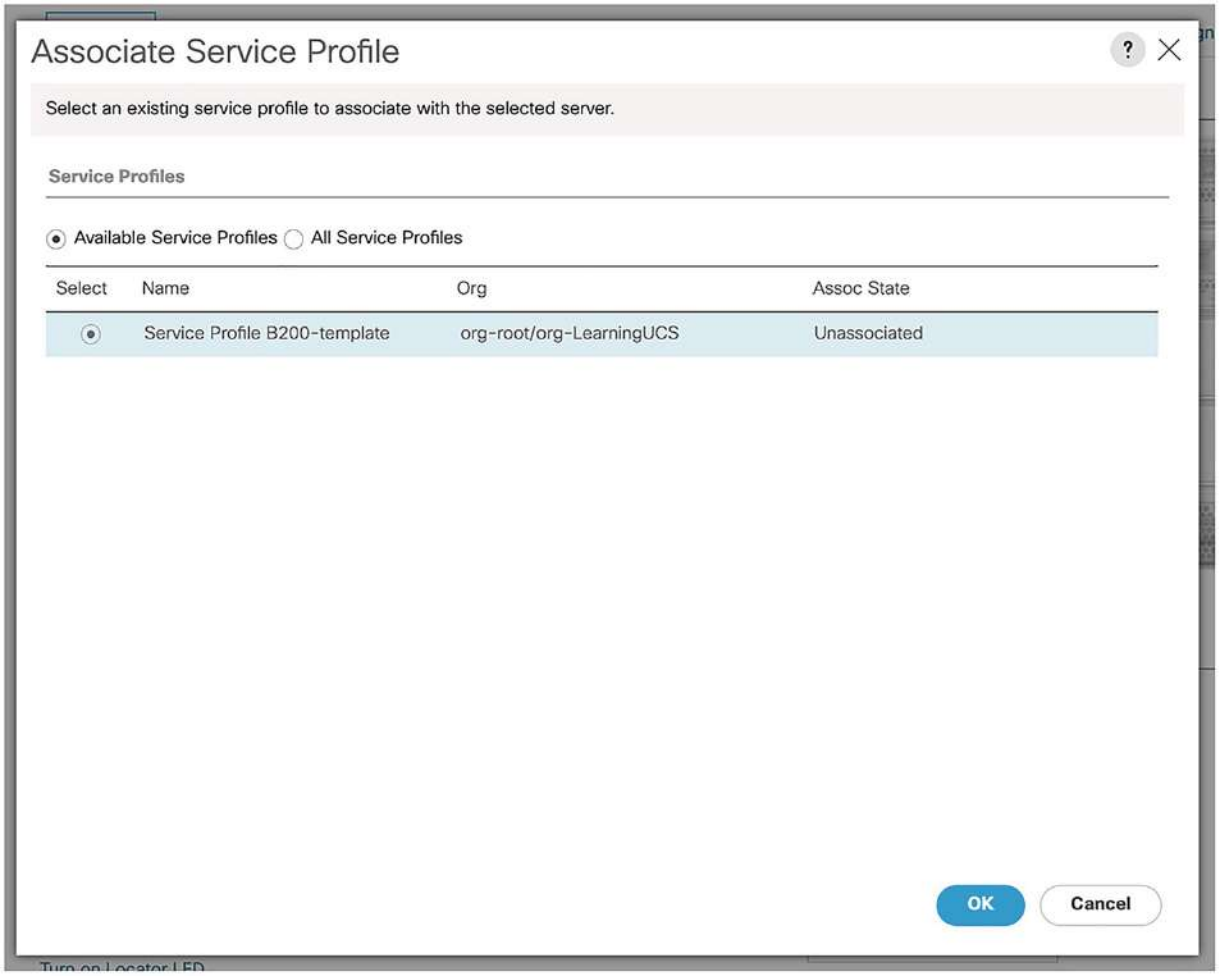

Figure 5-21 Selecting the service profile

Once we click OK, we will see a warning box pop up. This will list any issues with our template. More specifically, it will list any issues that may occur by assigning the template to the particular piece of hardware that we have chosen. The template may be perfectly fine, but specific policies may have different effects on different hardware variations – such as disk configurations or the number of network interfaces we have, for example.

Hopefully, there should not be any issues, and we should just get a warning that the blade will reboot (Figure 5-22).

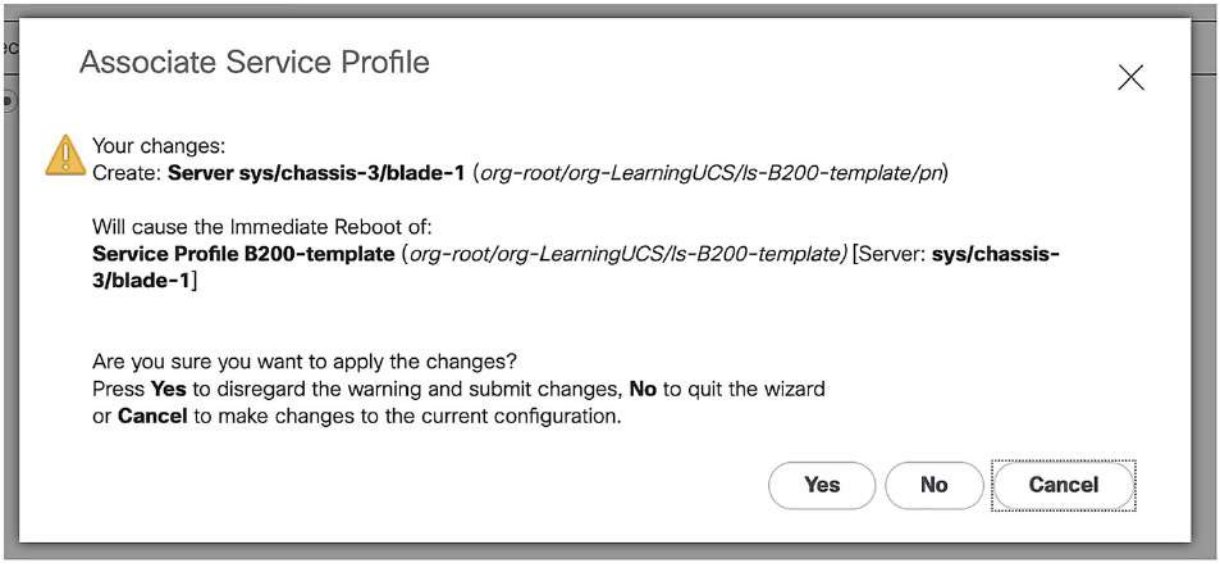

Figure 5-22 Associating the service profile

Once you click "*Yes*" to confirm, you will receive a final dialog to say that the operation has started (Figure 5-23).

Figure 5-23 The service profile is now associated

We can watch the operation from the blade's General page. After a few moments, the Overall Status will change from "*Unassociated*" (Figure 5-20) to "*Config*" (Figure 5-24).

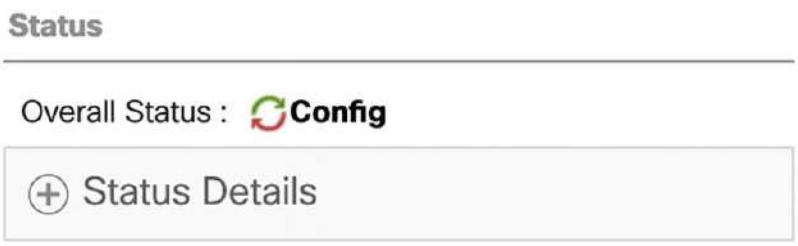

Figure 5-24 The Config status

Expanding the Status Details box will show us what task is currently running (Figure 5-25).

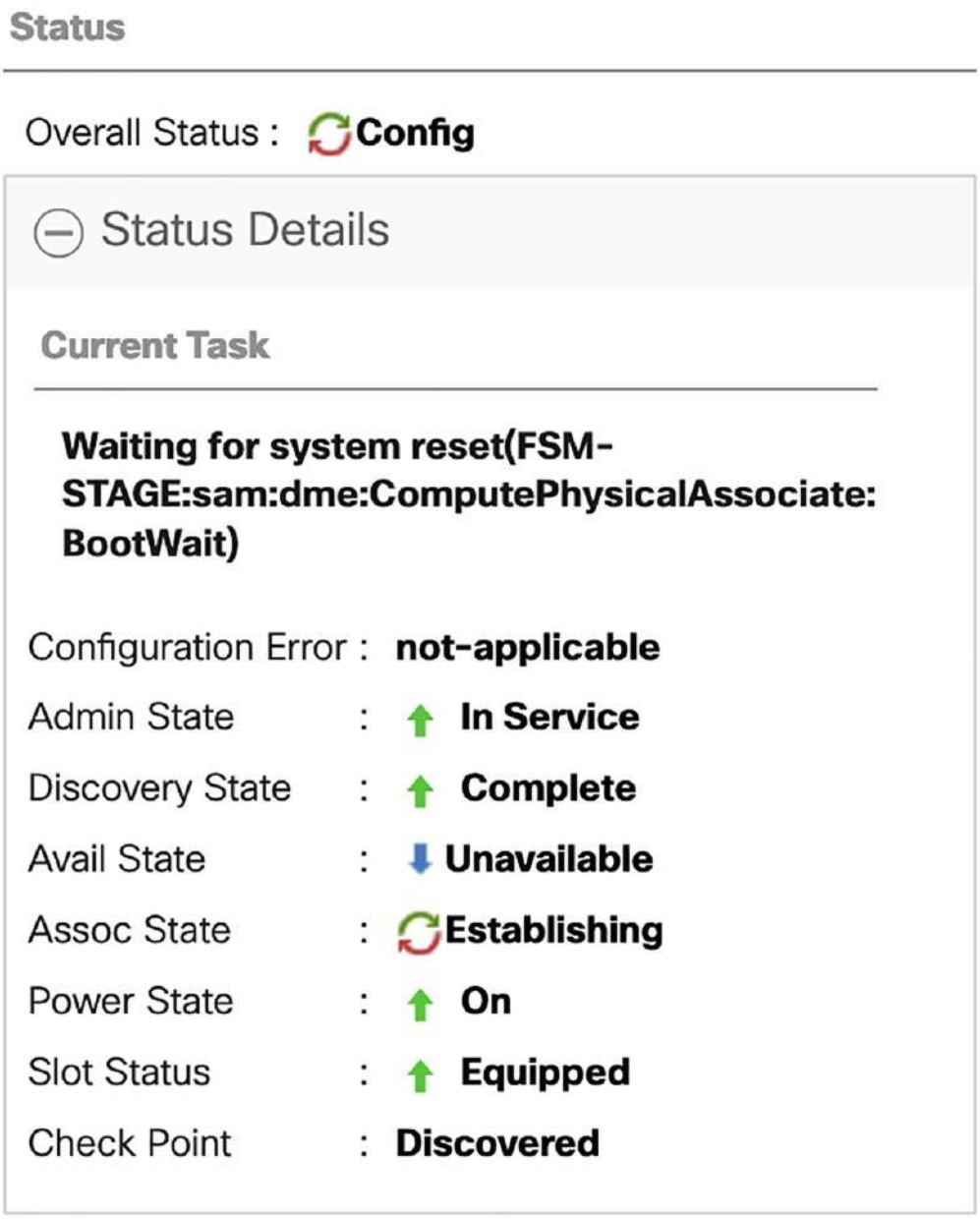

Figure 5-25 The Current Task

There is more information to be found, though, and this is in the FSM tab. This shows in much greater detail all the steps that occur when assigning a service profile to a server (such as Figure 5-26). The FSM tab is useful in many other scenarios, which we will look at in the troubleshooting chapter (Chapter 7).

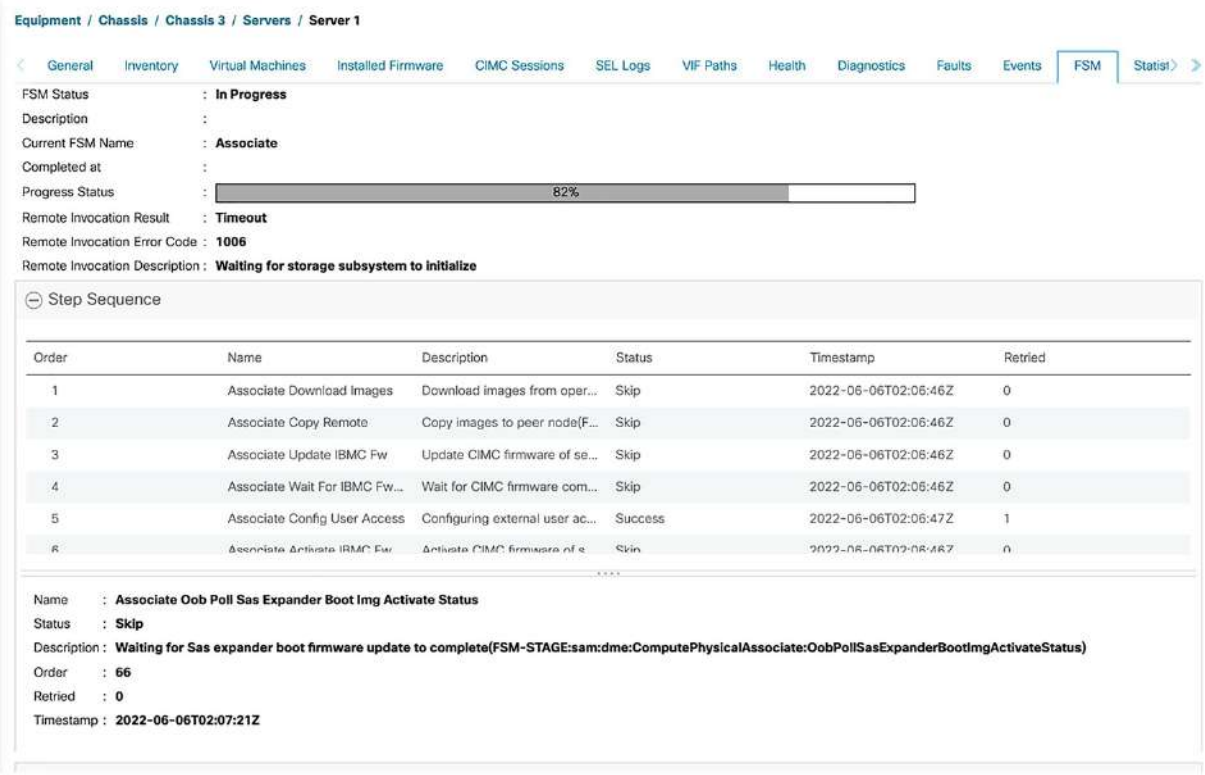

Figure 5-26 The FSM tab

Once all the tasks have been completed, we will see the status change to "*Success*," and the server will be powered on (Figure 5-27).

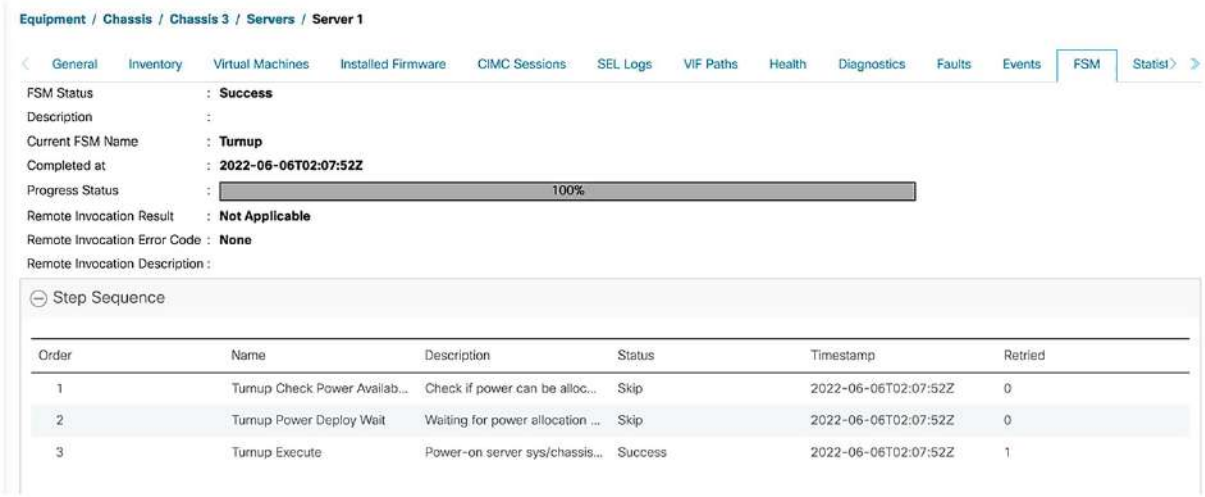

Figure 5-27 The association is a success

Returning to the general tab, we will see that the server now has an overall status of “OK” and that the server has been powered on (Figure 5-28).

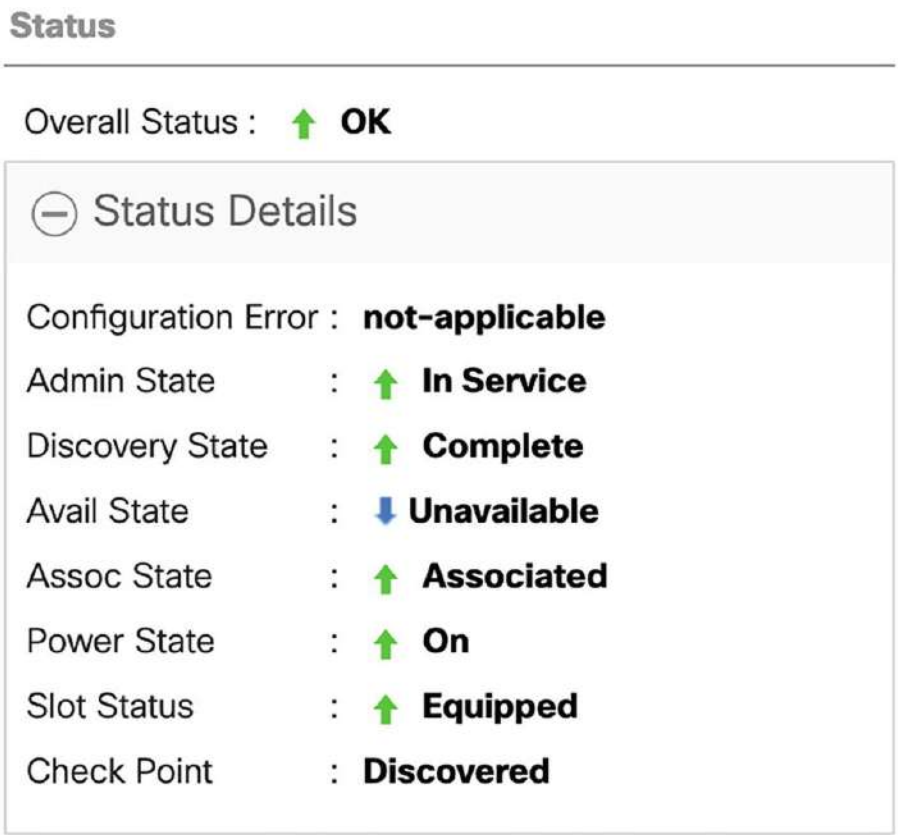

Figure 5-28 The server is now associated with a service profile

From the servers General tab, we can also see which template has been applied (Figure 5-29). This will be shown in its full path format (*org-root/org-LearningUCS/ts-B200-template*).

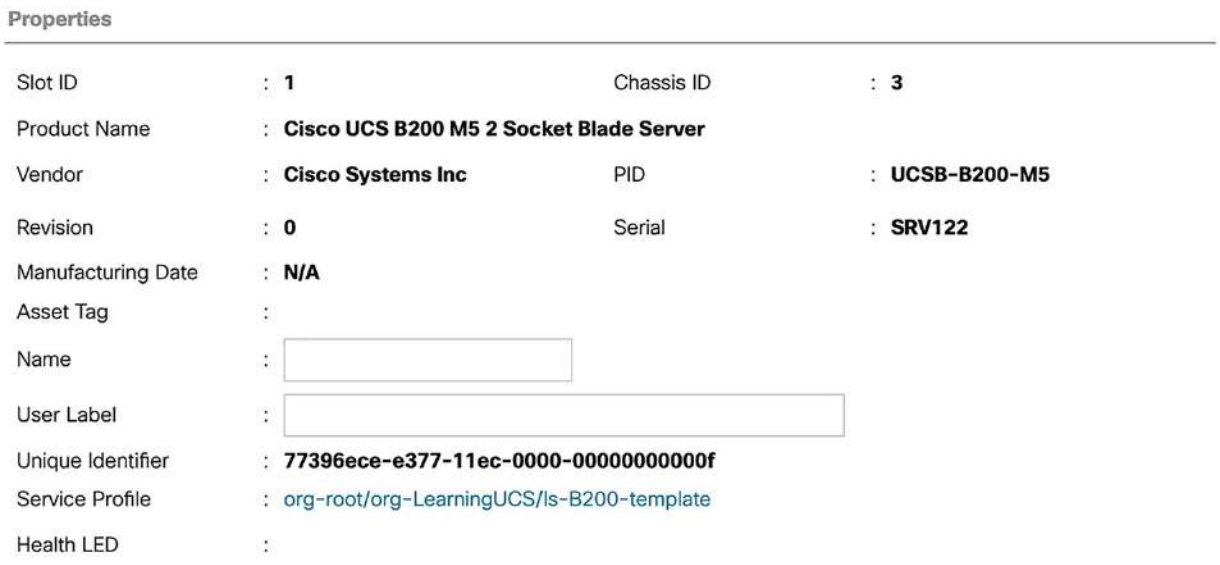

Figure 5-29 The service profile is shown on the general tab

We can also check the success of applying our template by checking the various settings applied to our server, such as the boot order (Figure 5-30).

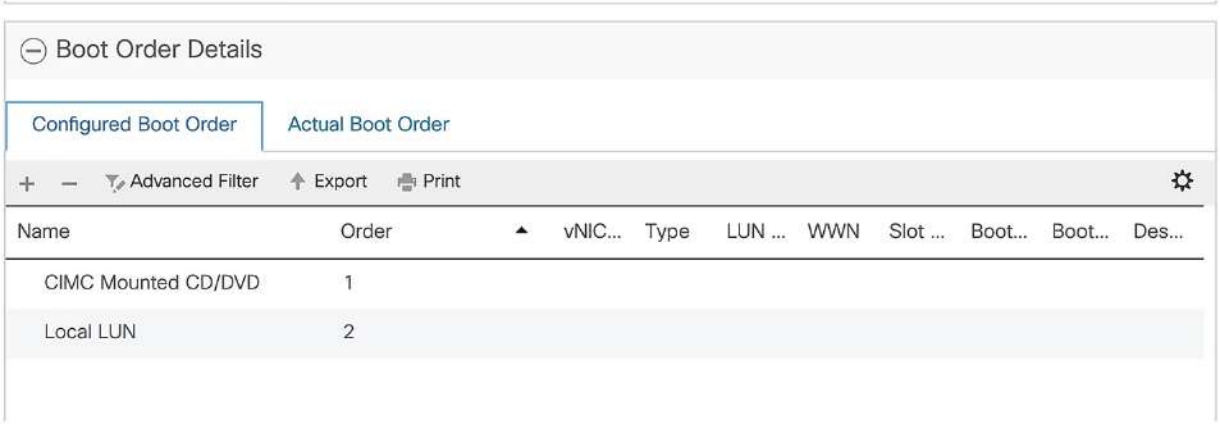

Figure 5-30 The boot order settings

The RAID settings can be found in the Inventory, under the Storage tab (Figure 5-31).

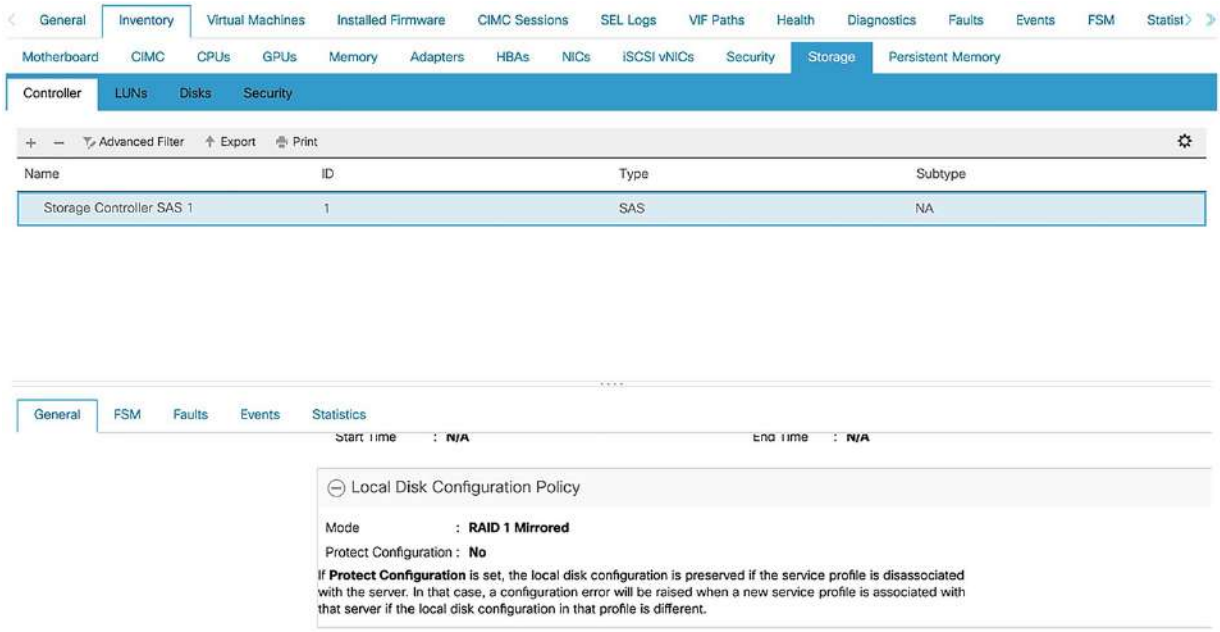

Figure 5-31 The RAID settings

The MAC address assignment can also be found in the Inventory, this time under the NICs tab (Figure 5-32).

Figure 5-32 The MAC address assignment

We can check the management IP assigned to the server from the General tab as well (Figure 5-33), which shows us the pool name as well (KVM-IP-Pool).

Figure 5-33 The management IP

We can also look in other places to check whether the template assignment has been successful, such as the UUID pools (*Servers ➤ Pools ➤ root ➤ Sub-Organizations ➤ LearningUCS ➤ UUID Suffix Pools ➤ UUID-Pool*) as shown in Figure 5-34.

Figure 5-34 The UUID pool usage

Summary

In this chapter, we created a service template using the policies and pools we created in Chapter 4. We then assigned this to a server and checked that the correct template settings had been applied.

In the next chapter, we will look at securing the UCS.

S. Fordham, *Introducing Cisco Unified Computing System*
https://doi.org/10.1007/978-1-4842-8986-0_6

6. UCS Security

Stuart Fordham[1]
(1) Bedfordshire, UK

In this chapter, we will be securing our UCS system through AAA for logins and hardening of the system.

AAA

Presently, we use the inbuild admin account, “USCPE.” However, on a live system, you would naturally need to use multiple accounts, which should be individual and trackable. We do this through AAA (Authentication, Authorization, and Accounting), whereby we connect to and use the accounts from a directory service. The options we have for this are LDAP, RADIUS, and TACACS+. This chapter will look at how this would be achieved through LDAP (specifically, Microsoft Active Directory). While UCSPE is a self-contained environment, we can run through the setup to see how it looks in real life.

When using AAA, we map groups of users to one of the in-built UCS roles. These roles are

Role name	Description
aaa	The AAA administrator role gives read-and-write access to the users and roles as well as the AAA configurations. It has read access to the rest of the UCS system.
admin	As the name suggests, the admin role has full access to the entire UCS system.
facility-manager	The facility manager role has read/write access to the power management portion of the system. It has read access to the rest of the UCS system.
network	This role has read/write access to the fabric interconnects and also to network security functions. It has read access to the rest of the UCS system.
operations	Read/write access to logs and faults. It has read access to the rest of the UCS system.
read-only	Read-only access to the system.
server-compute	Read/write access to the majority of the service profile settings. Cannot create, modify, or delete vNICs or vHBAs.
server-equipment	Read/write access to physical server-related operations. It has read access to the rest of the UCS system.

Role name	Description
server-profile	Read/write access to logical server-related operations. It has read access to the rest of the UCS system.
server-security	Read/write access to server security-related operations. It has read access to the rest of the UCS system.
storage	Read/write access to storage operations. It has read access to the rest of the UCS system.

We start by creating an LDAP provider, by going to "*Admin ➤ User Management ➤ LDAP ➤ LDAP Providers*" and clicking "*Add*." Enter the LDAP details specific to your environment, such as those shown in Figure 6-1.

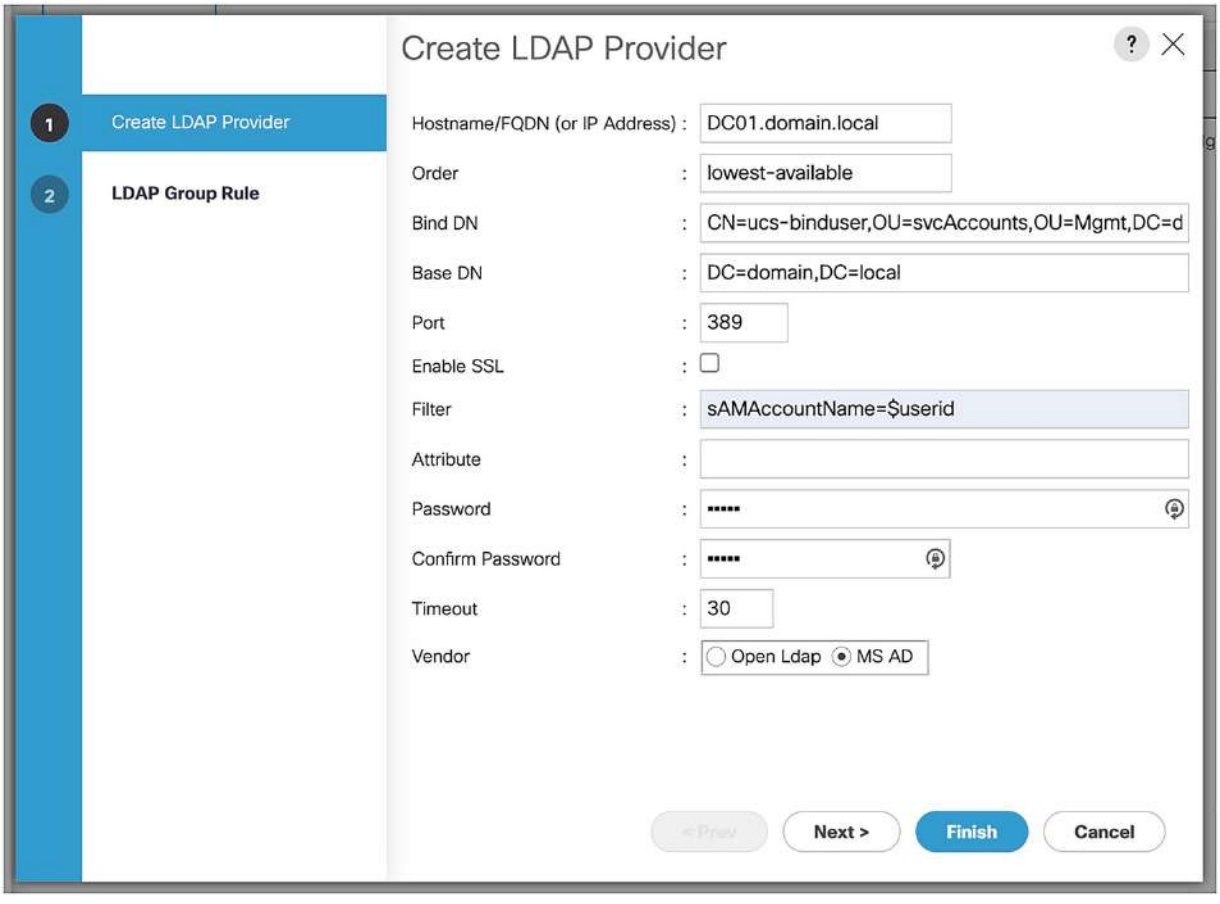

Figure 6-1 Creating the LDAP provider

In the preceding, we are using the following settings (details such as the Bind DN would be in the Active Directory):

Hostname/FQDN (or IP address): DC01.domain.local
Order: lowest-available
Bind DN: CN=ucs-binduser,OU=svcAccounts,OU=Mgmt,DC=domain,DC=local
Base DN: DC=domain,DC=local
Port: 389
Enable SSL: unticked
Filter: sAMAccountName=$userid
Attribute: empty
Password: $tr0ngP4ssw0rd!
Confirm Password: $tr0ngP4ssw0rd!
Vendor: MS AD

Click next and set group authorization to "*Enable*" and group recursion to recursive (Figure 6-2). This means that the UCS will use the target attribute (memberOf) to check if the user authenticating is a member of a group. Group recursion allows the UCS to look through the user directory level by level until it finds the user.

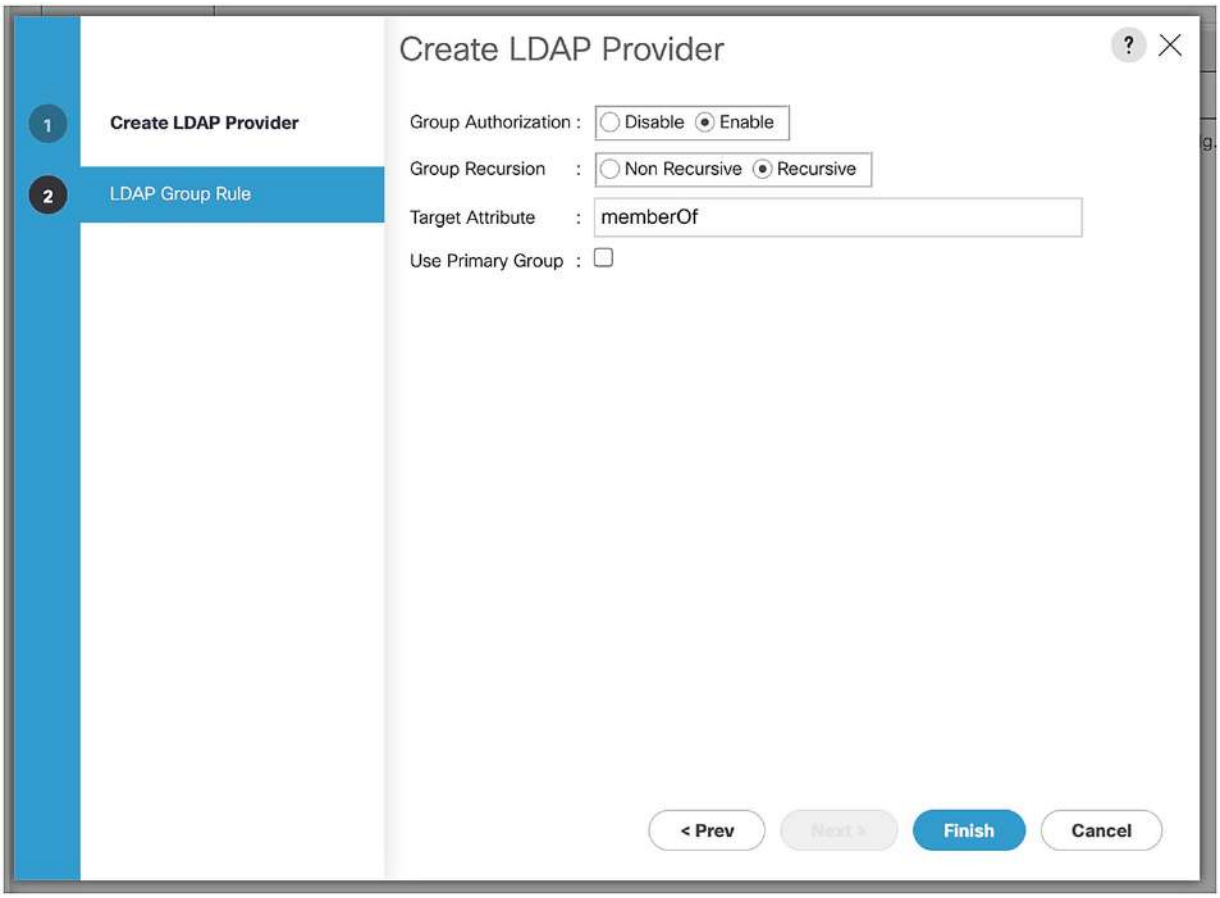

Figure 6-2 The LDAP provider group settings

Leave the Target Attribute as "*memberOf,*" and "*Use Primary Group*" unticked. Once you are done, click "*Finish*." You will get a notification that the LDAP provider has been created (Figure 6-3).

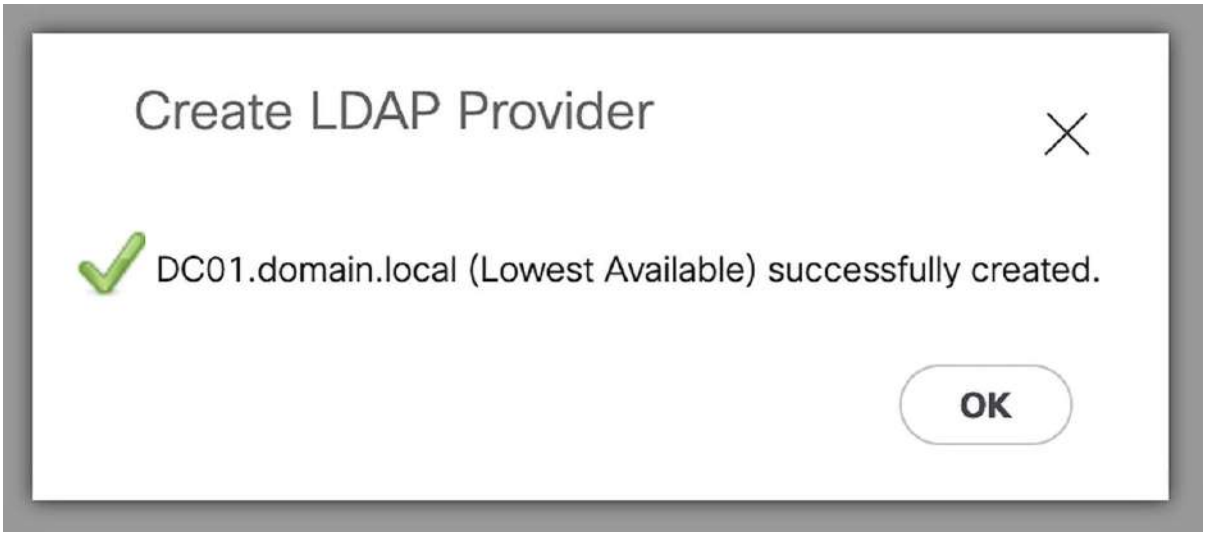

Figure 6-3 The LDAP provider has been created

Usually, you would have more than one LDAP provider to offer a level of resiliency.

We now need to create an LDAP Provider Group. Go to "*Admin ➤ User Management ➤ LDAP ➤ LDAP Provider Groups*" and click "*Add*." Name the provider group (Figure 6-4).

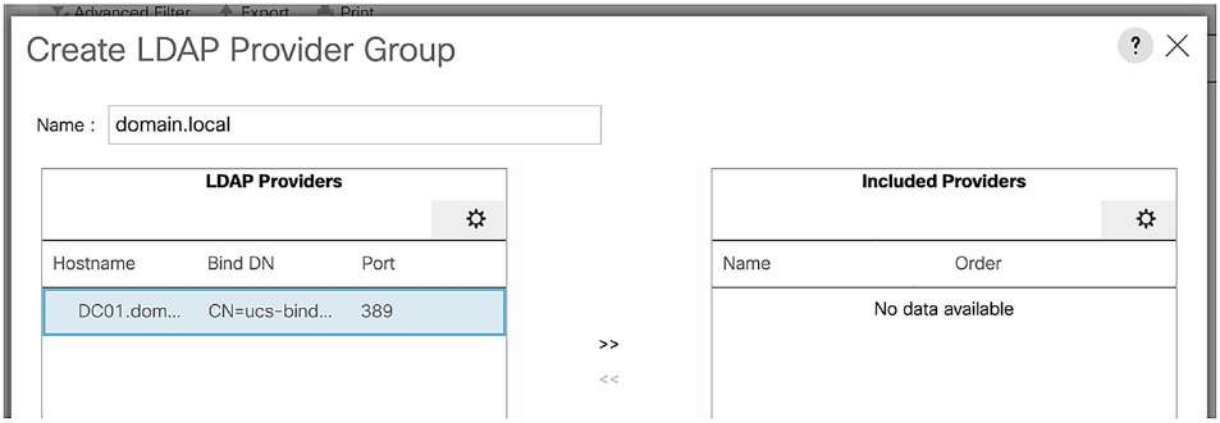

Figure 6-4 Creating an LDAP provider group

Select the LDAP providers you created in the first step, and click the arrows to move it from LDAP Providers to Included Providers (Figure 6-5).

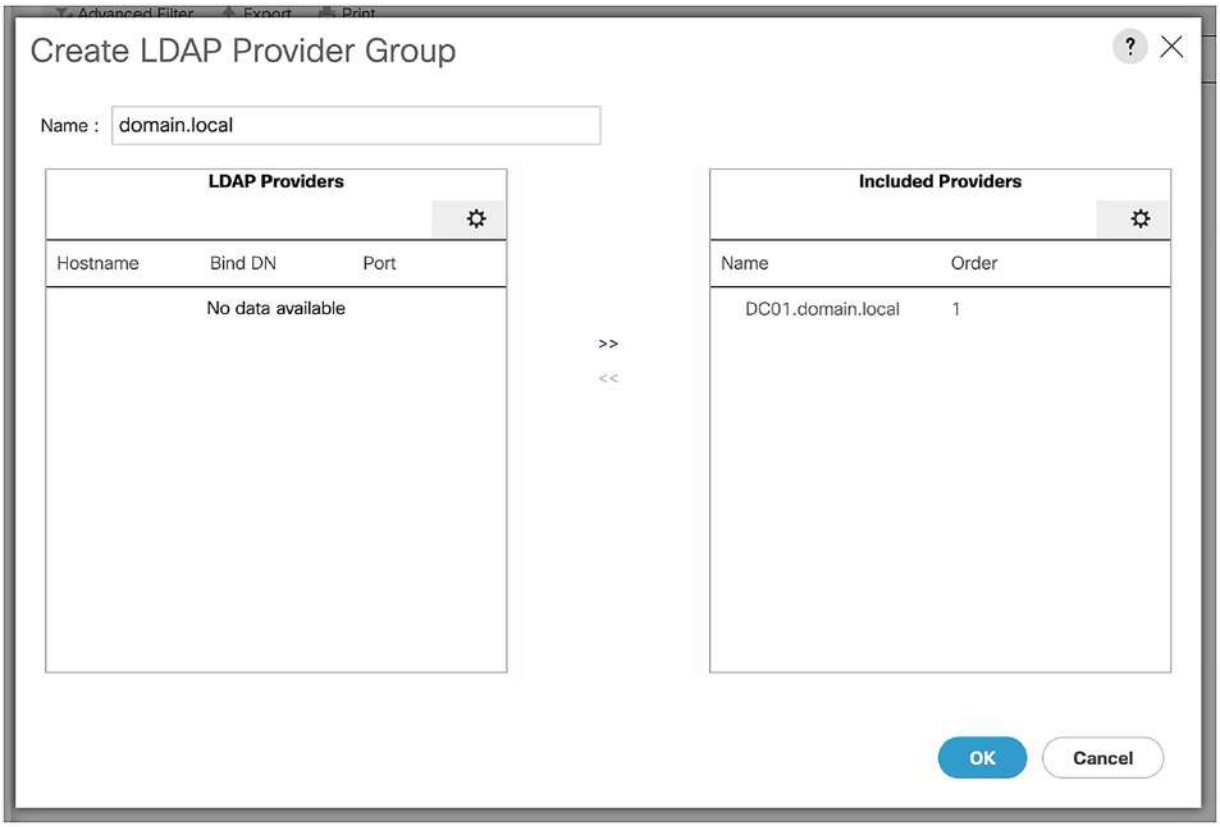

Figure 6-5 The LDAP provider group

Click "*OK*" to save the group and you will see the notice of completion (Figure 6-6).

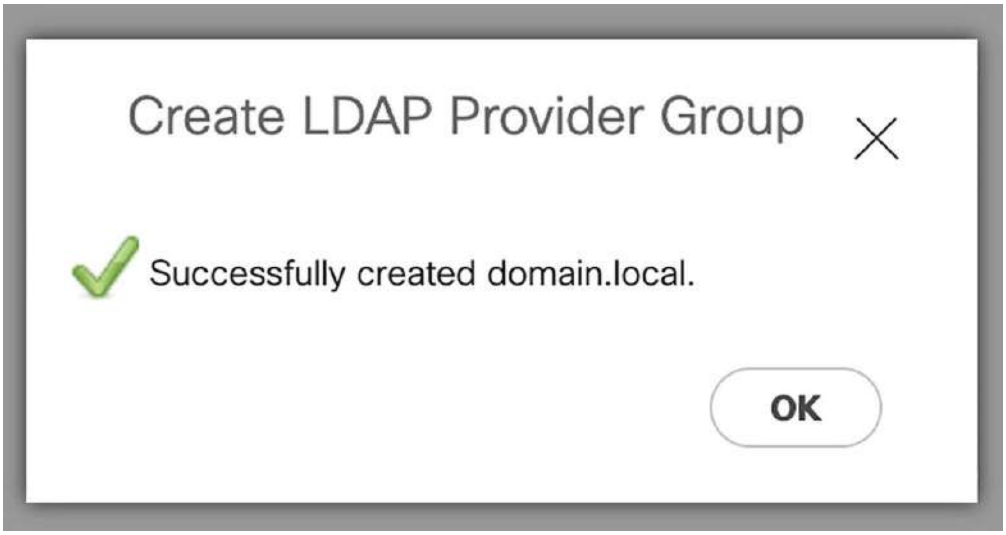

Figure 6-6 The LDAP provider group has been created

We will next create some LDAP group maps, to link our AD groups to the UCS roles. We do this from "*Admin ➤ User Management ➤ LDAP ➤ LDAP Group Maps*." Click "*Add*" and add the LDAP group DN, and click on the desired role, similar to Figure 6-7.

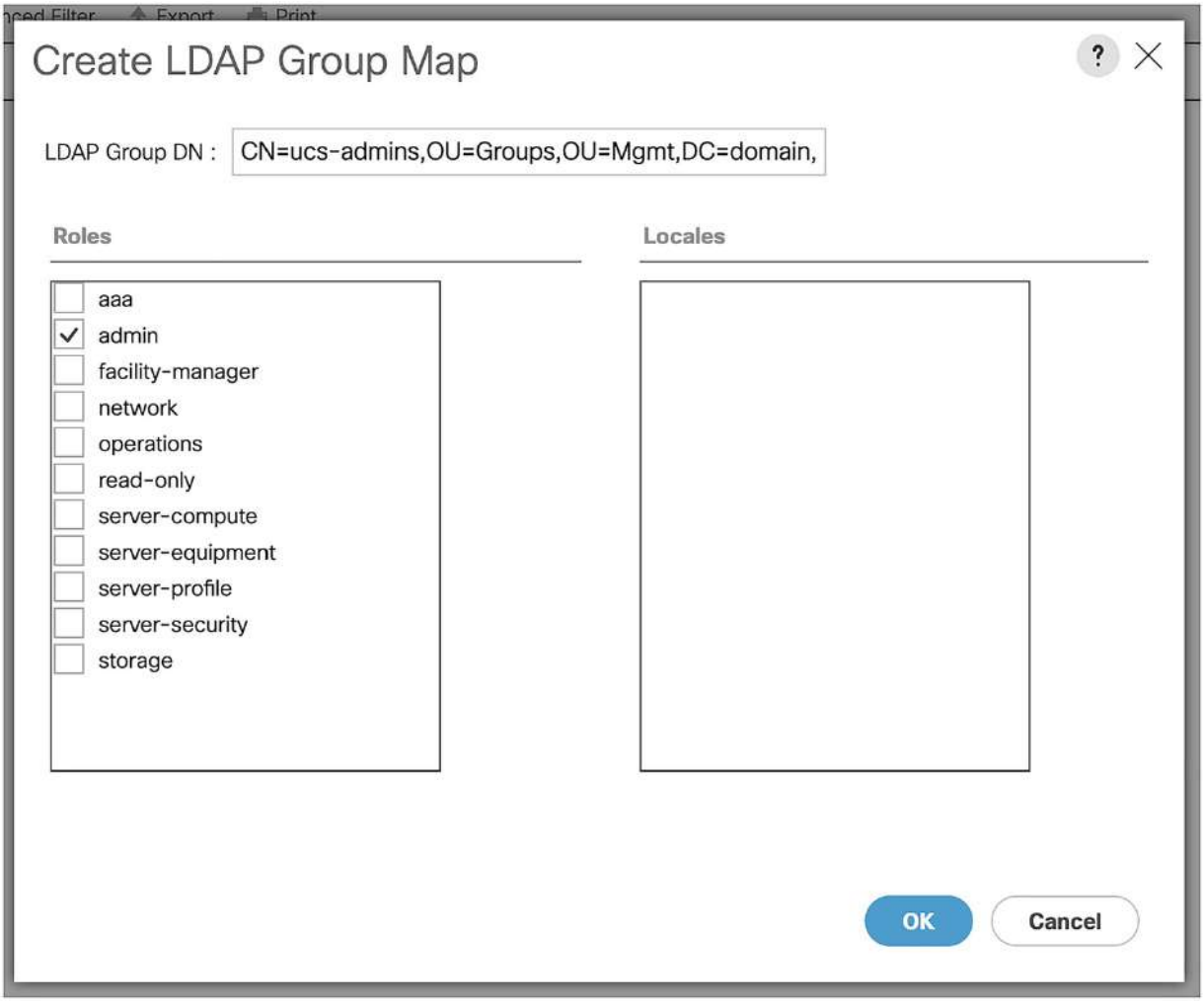

Figure 6-7 Creating an Admin LDAP Group Map

Create another for the read-only role (such as Figure 6-8).

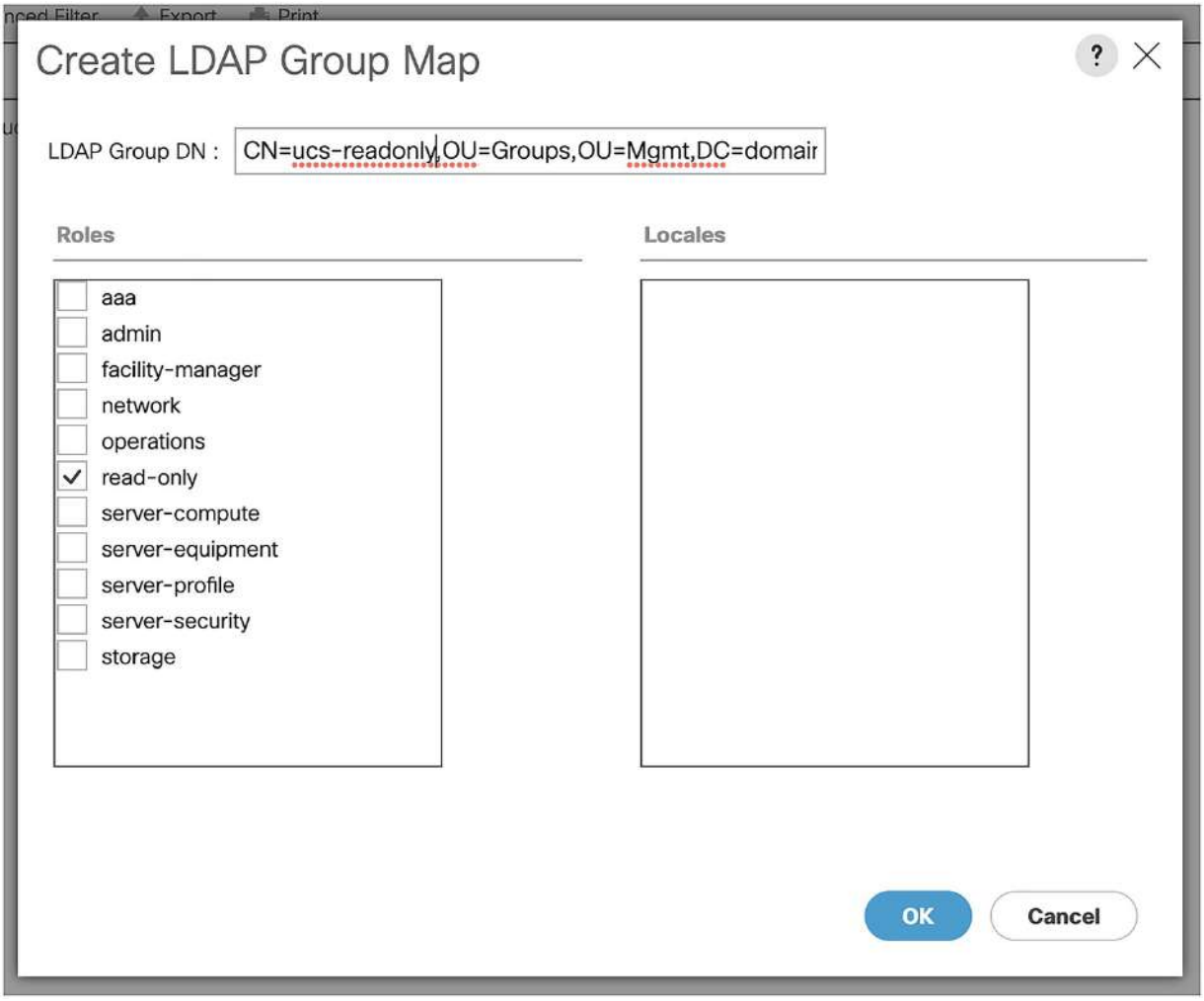

Figure 6-8 Creating a read-only LDAP Group Map

Our two group mappings will be shown in the GUI (Figure 6-9).

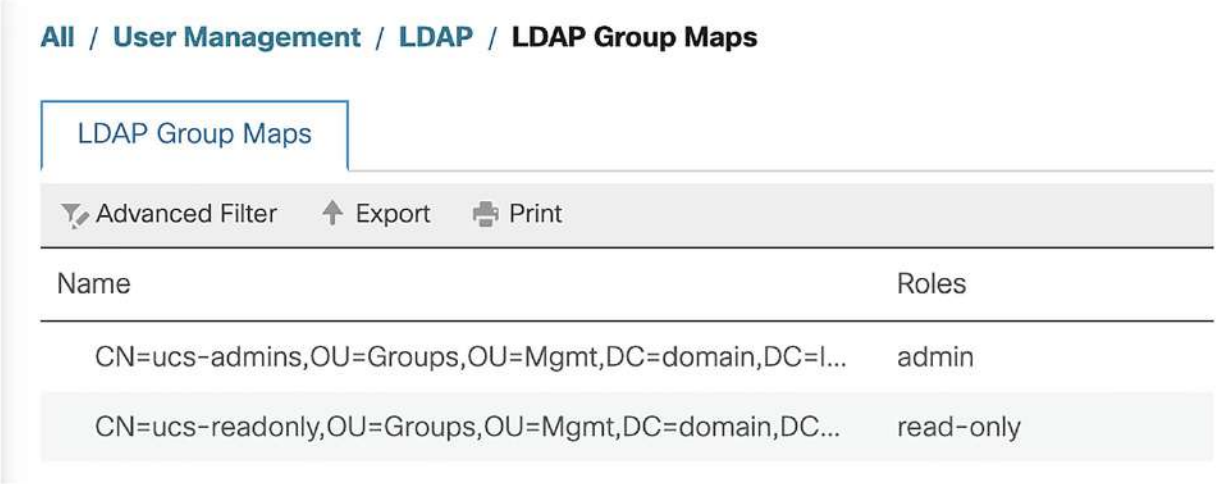

Figure 6-9 The completed LDAP group mappings

Now that we have our LDAP provider set up, we need to tell the UCS how to use it. Go to "*Admin ➤ User Management ➤ Authentication ➤ Authentication Domains*" and click "*Add*." In the Create a Domain window, name the domain, and set the realm to LDAP, selecting the provider group we created earlier (Figure 6-10).

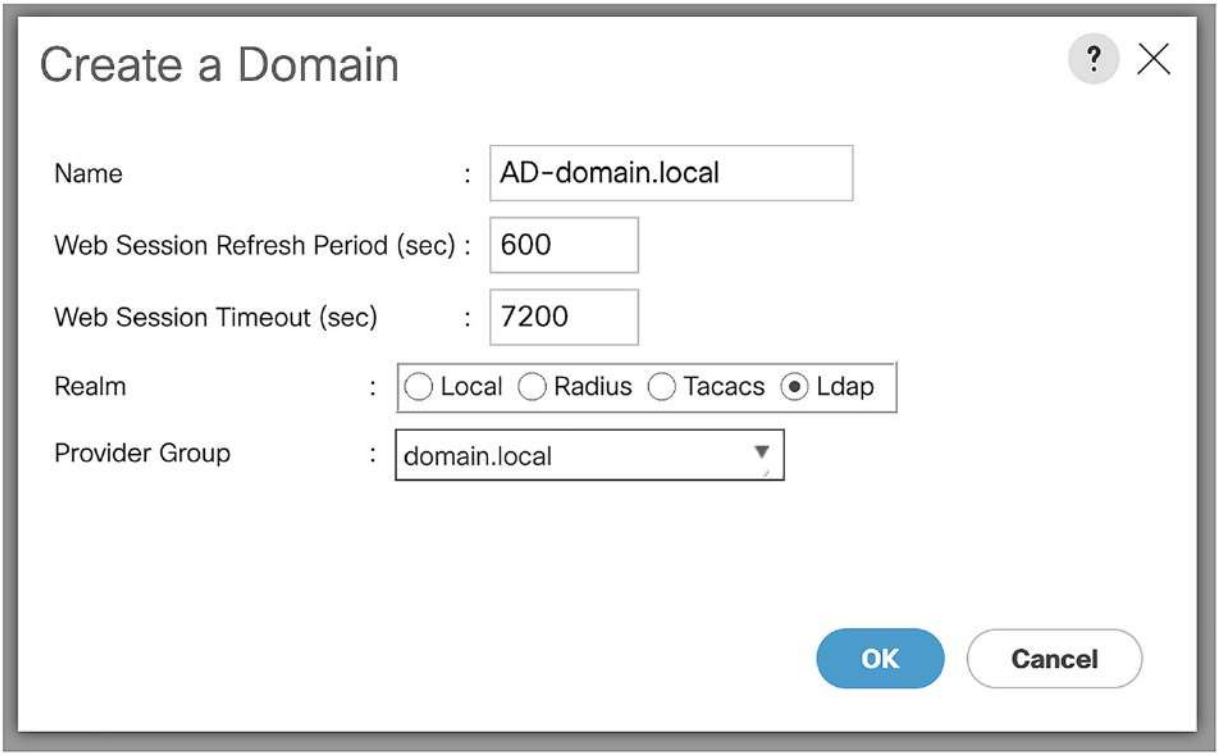

Figure 6-10 Creating the authentication domain

The next time we log in, the UCS will first search the domain.local Active Directory for the user, before falling back to the local user account database on the UCS. If the user is not found in either Active Directory or the local database, then the login will be denied.

Hardening the Web Interface

By default, UCS will use a self-signed certificate (in the case of UCSPE, there is no certificate for the web interface). We can't assign a certificate in UCSPE (as this is a closed-off playground), but we can run through the steps.

Navigate to "*Admin ➤ Key Management*" and click "Trusted Points." Trustpoints are the certification authorities we are using. Click "*Add*." Add the certificate from your Certification Authority and click OK to create the trust point (Figure 6-11).

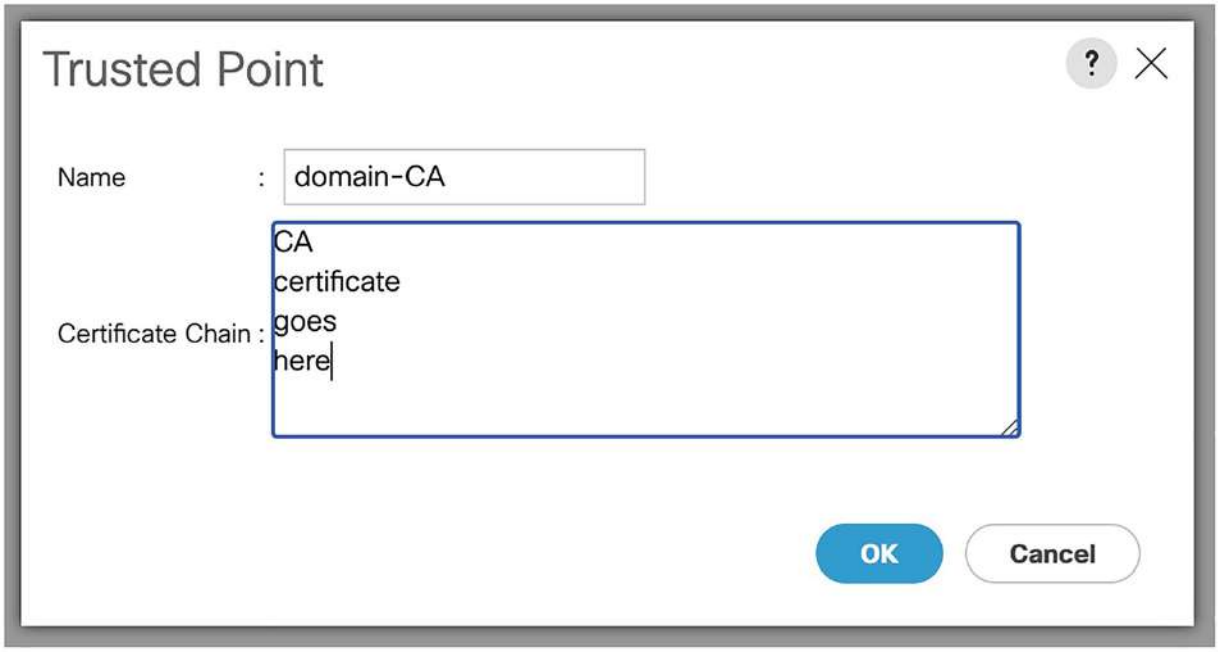

Figure 6-11 Creating the trust point

Click Key Management again, and then "*Key Rings*." Click "*Add*" and create a new key ring (Figure 6-12). Click "*OK*."

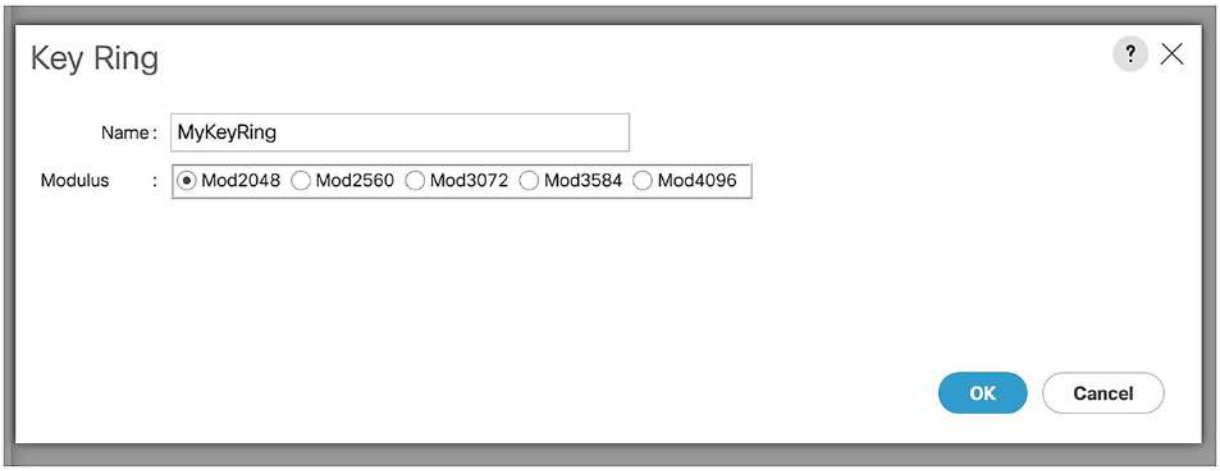

Figure 6-12 Creating a key ring

Select the new key ring and then click "*Create Certificate Request*." Enter the details (as shown in Figure 6-13).

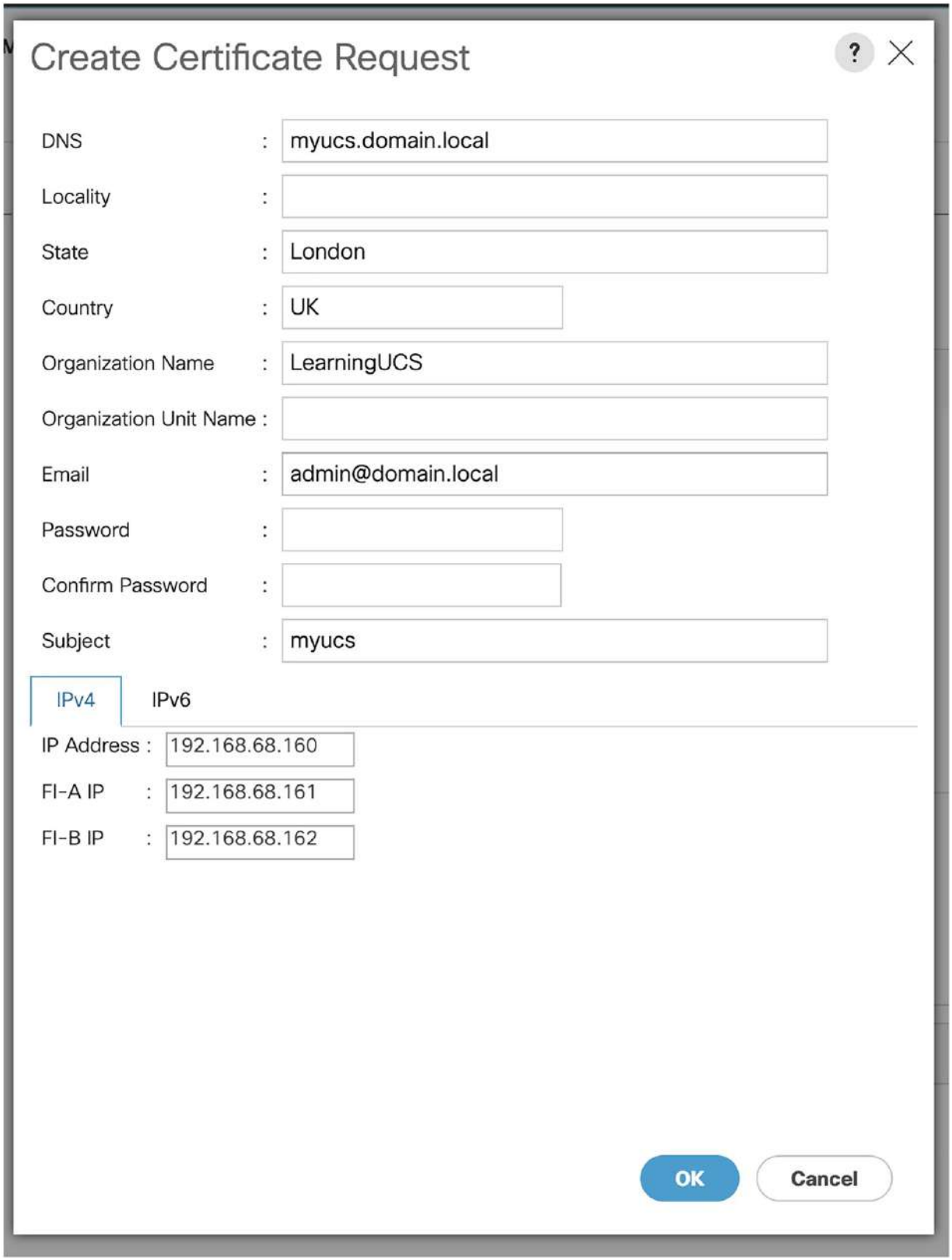

Figure 6-13 Creating the certificate request

We will see the final details in the GUI (Figure 6-14).

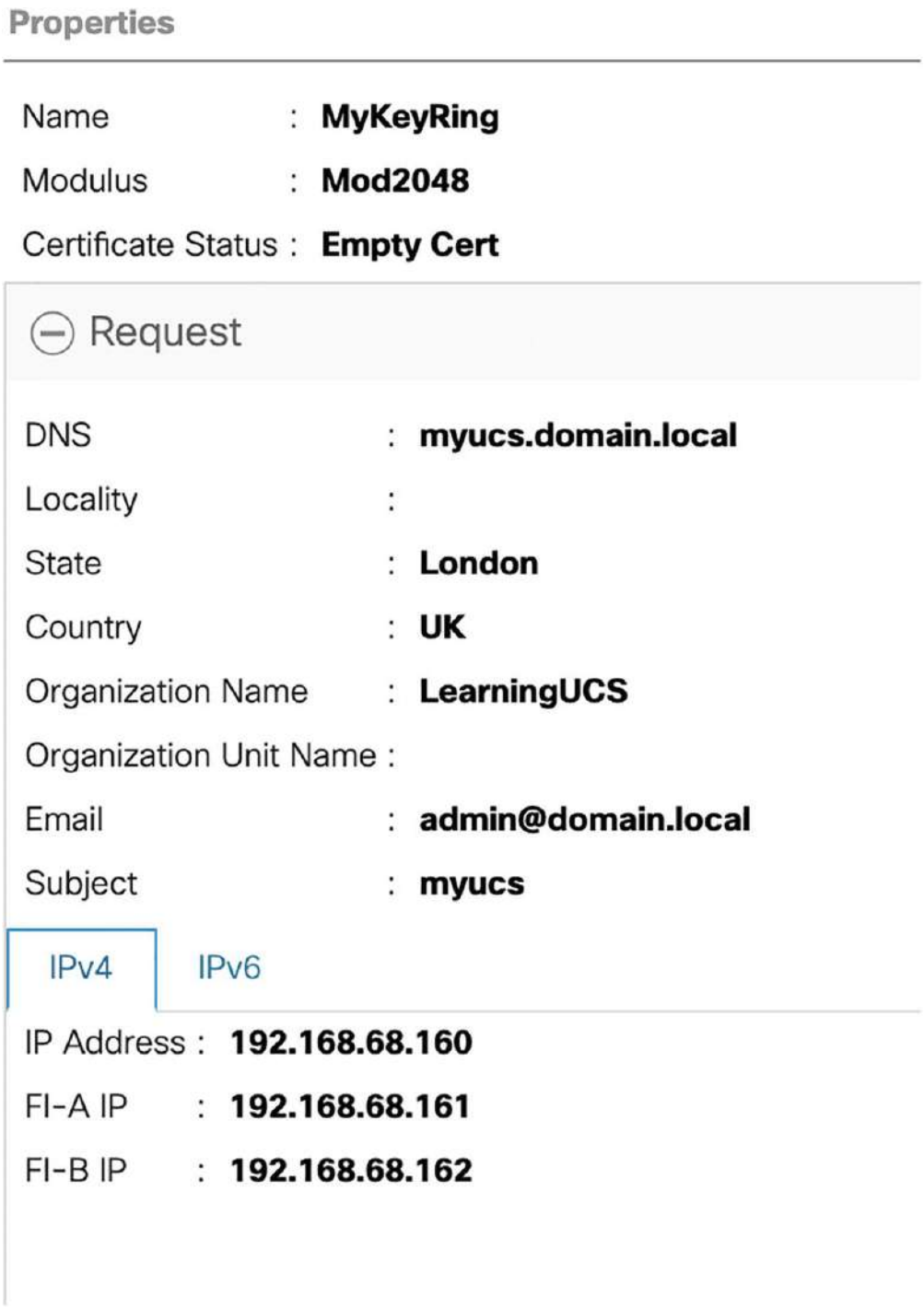

Figure 6-14 The final certificate settings

We would (on a real-life system) then copy the certificate from the same page, and have this certificate

signed by our certificate authority.

Once we have the signed certificate, download it and choose the Base64 encoded version, open it in a text editor, and select and copy the entire text, then click "*Certificate*," select the trust point, and paste in the signed certificate. The final step in the process is to select the key ring in the Communication Services (*Admin ➤ Communication Management ➤ Communication Services*). Under the HTTPS settings, select the newly created, and newly certified key ring (Figure 6-15).

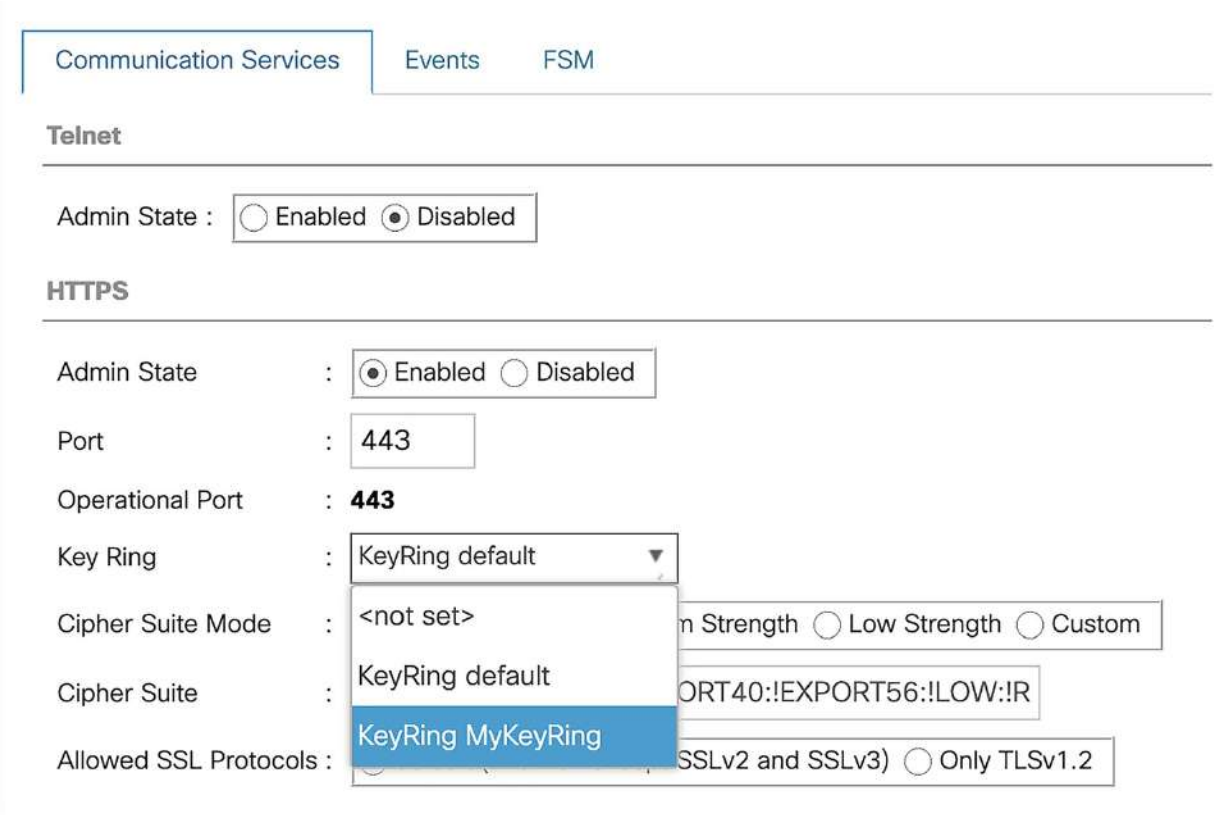

Figure 6-15 Setting the key ring to be used for HTTPS GUI access

There are a few other steps we can take to harden our UCS. Firstly, we should switch to HTTPS instead of permitting HTTP, which we do under the HTTP settings where we can redirect HTTP traffic to our UCS to the more secure HTTPS (Figure 6-16).

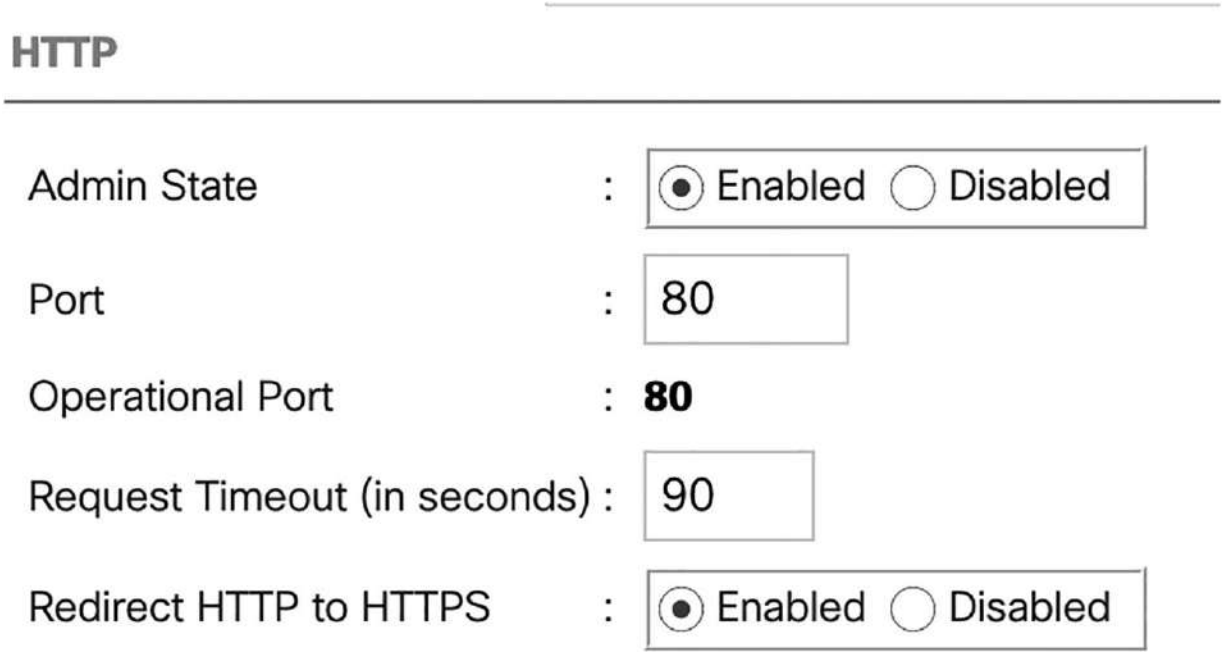

Figure 6-16 Redirecting HTTP to HTTPS

Next, we should disable Telnet, as anyone sniffing your traffic will be able to capture your unencrypted password (Figure 6-17).

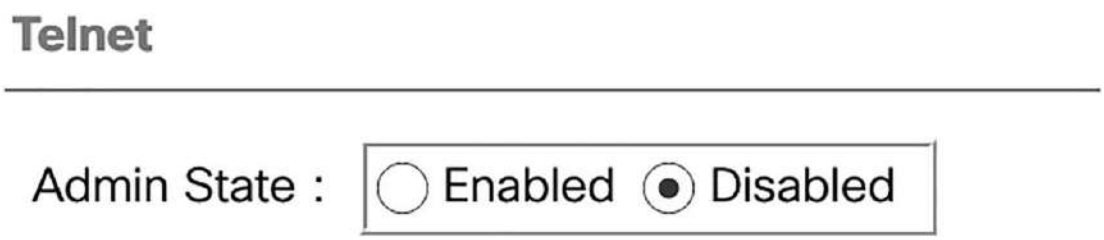

Figure 6-17 Disabling Telnet

Finally, we can get rid of the less secure ciphers by setting the cipher suite mode to "*Custom*" and setting the cipher suite to use "*ALL:!aNULL:!eNULL:!LOW:!3DES:!MD5:!EXP:!DSS:!MEDIUM*" (which excludes ciphers such as low, triple-DES, MD5, for example) and then set the "*Allowed SSL Protocols*" to "*Only TLSv1.2*" (Figure 6-18).

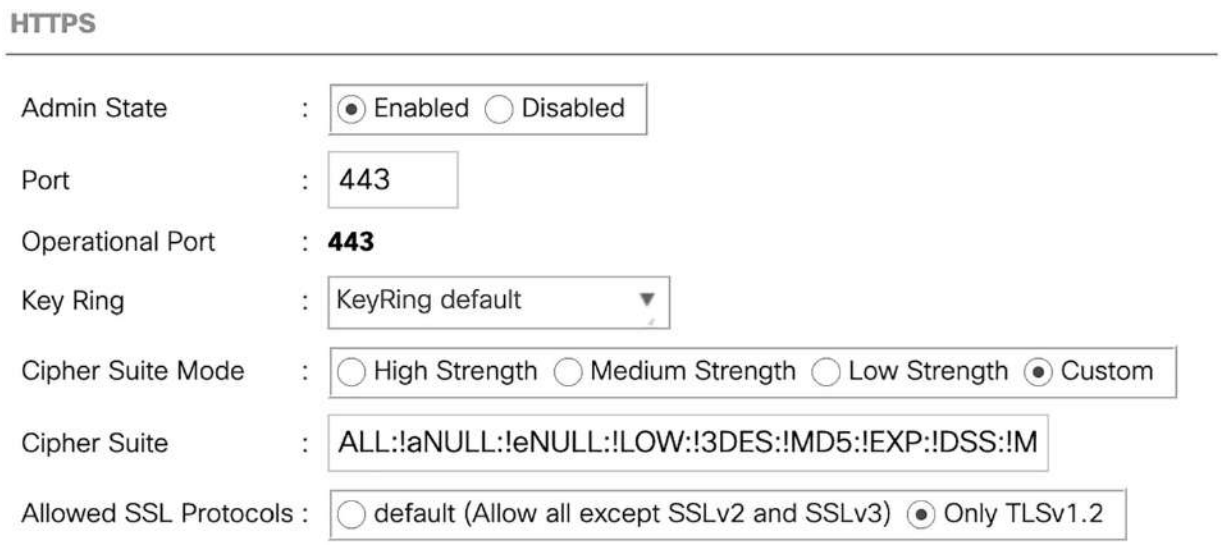

Figure 6-18 Hardening HTTPS

These settings should satisfy most PCI QSAs and security scans!

Summary

In this chapter, we secured access to our UCS system. In the next chapter, we will look at UCS troubleshooting.

S. Fordham, *Introducing Cisco Unified Computing System*
https://doi.org/10.1007/978-1-4842-8986-0_7

7. UCS Troubleshooting

Stuart Fordham[1] ✉
(1) Bedfordshire, UK

In this, our final chapter, we will look at ways we can troubleshoot the UCS and enhance our monitoring of the platform.

Call Home

Call home enables the UCS to send diagnostic data to chosen recipients either periodically or in the event of an issue, so in the event of a problem, Cisco can be notified automatically and have relevant data sent to them.

We start by going to "*All ➤ Communication Management ➤ Call Home*" and enabling it (Figure 7-1).

Figure 7-1 By default Call Home is disabled

Once we enable it, we need to enter our contact information and contract details (Figure 7-2).

Figure 7-2 The Call Home details

After you have entered the email details, scroll down and enter the SMTP server details, and click “Save

Changes” (Figure 7-3).

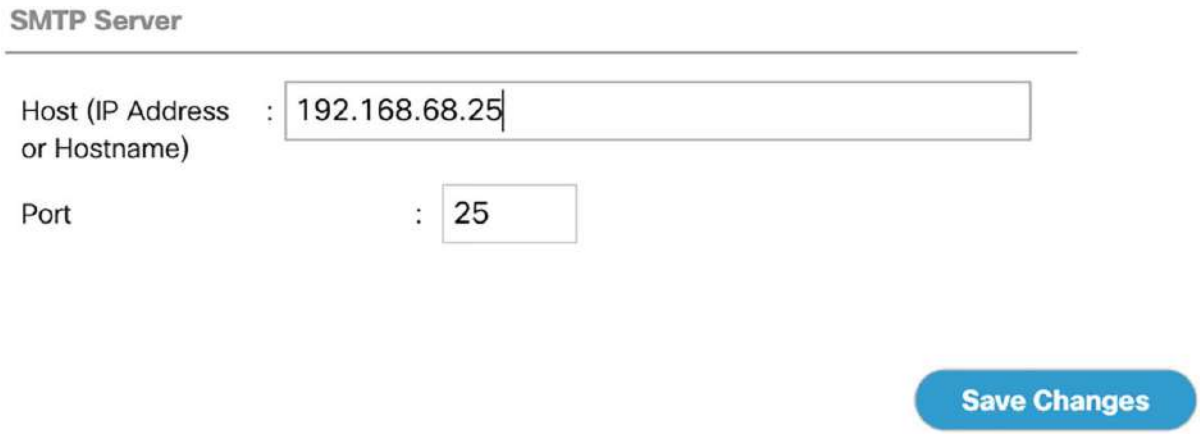

Figure 7-3 The SMTP server used for Call Home

Call Home uses profiles and these control who gets what information. The inbuilt ones are shown in Figure 7-4.

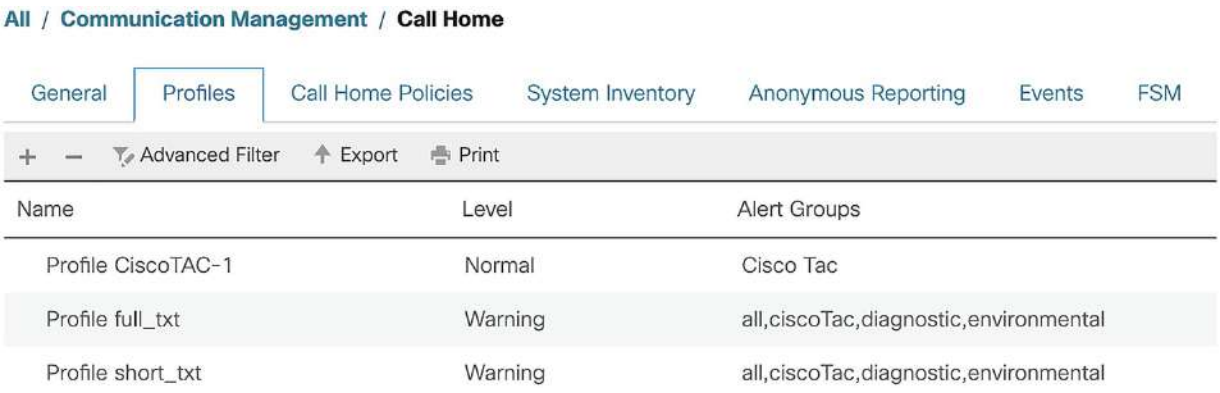

Figure 7-4 The Call Home profiles

We can create our own specific profiles by clicking the “Add” button and entering the details (as shown in Figure 7-5). The predefined alert groups cover environmental (power, fan, temperature alarms, and other similar issues), diagnostic (such as a server’s POST details), and all the other critical alerts (Cisco TAC).

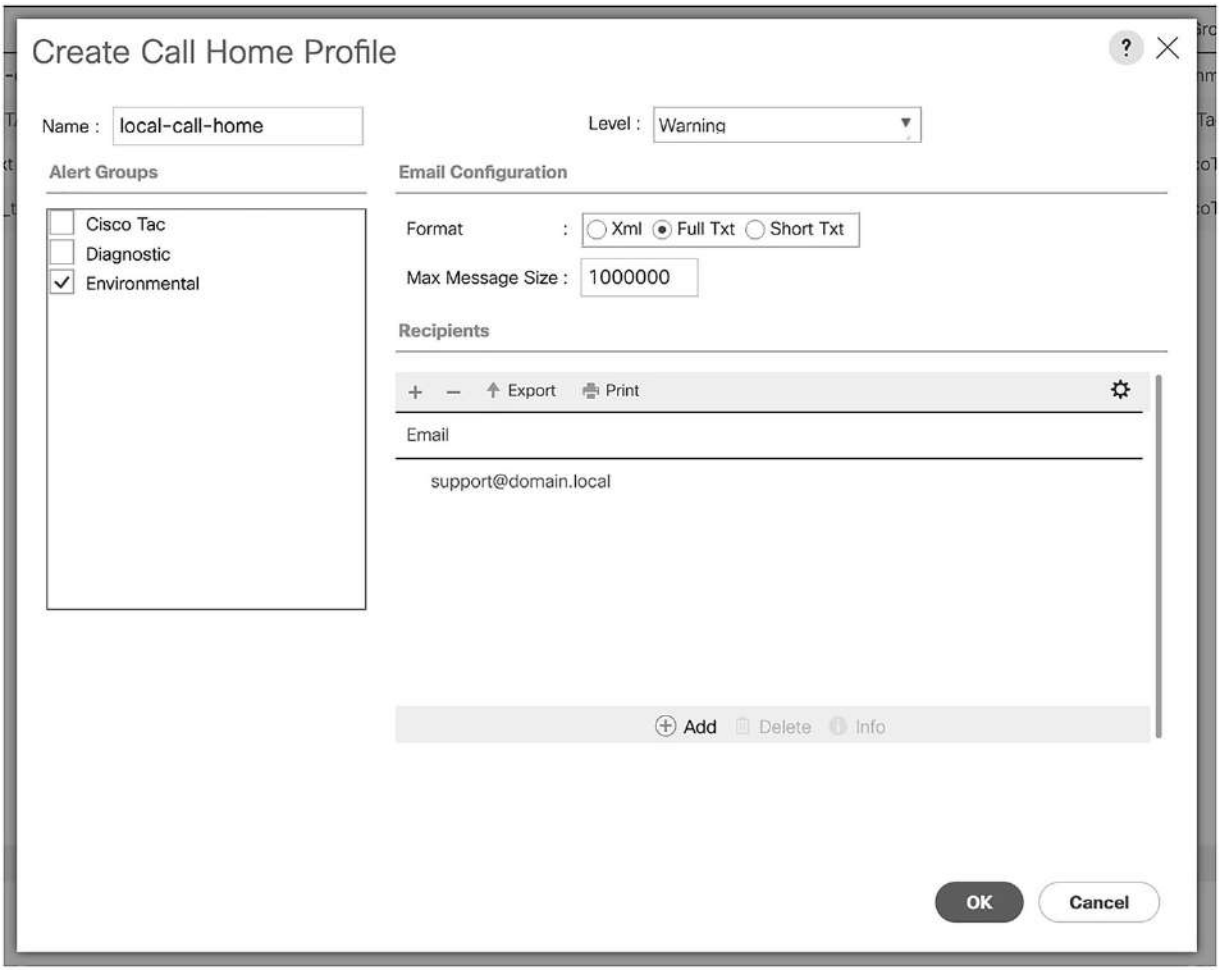

Figure 7-5 A custom Call Home profile

We can also (if we want to) enable or disable call home for specific causes (Figure 7-6).

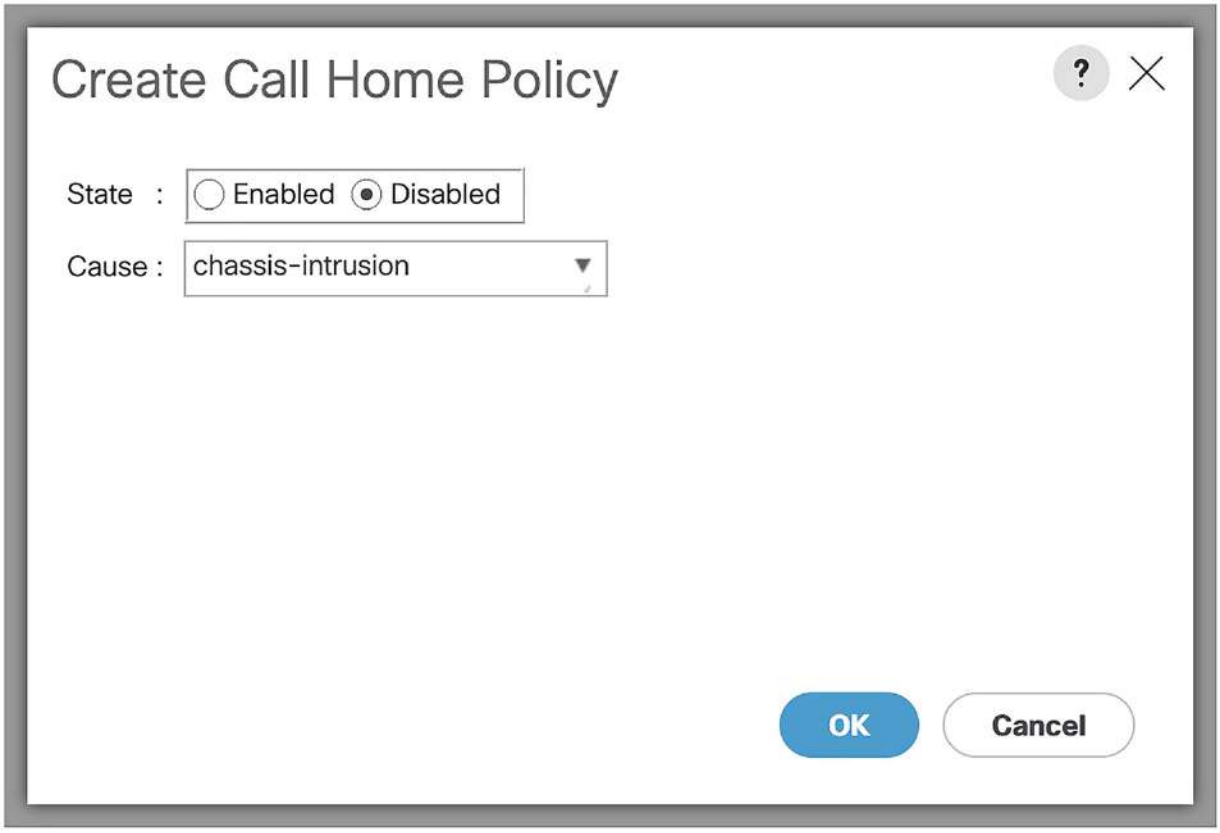

Figure 7-6 A Call Home policy

Call home can also send the system inventory (either periodically, or on-demand), as shown in Figure 7-7.

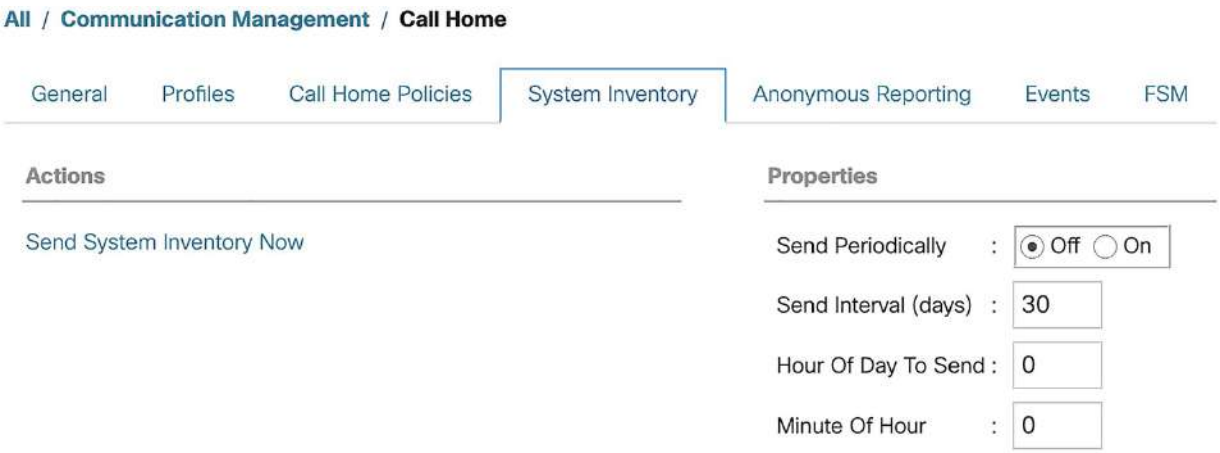

Figure 7-7 Sending system inventory

Anonymous reporting is independent of how call home is configured, in that call home may be disabled and

anonymous reporting can be enabled (and vice versa); this sends a minimal error and health information about the UCS (Figure 7-8).

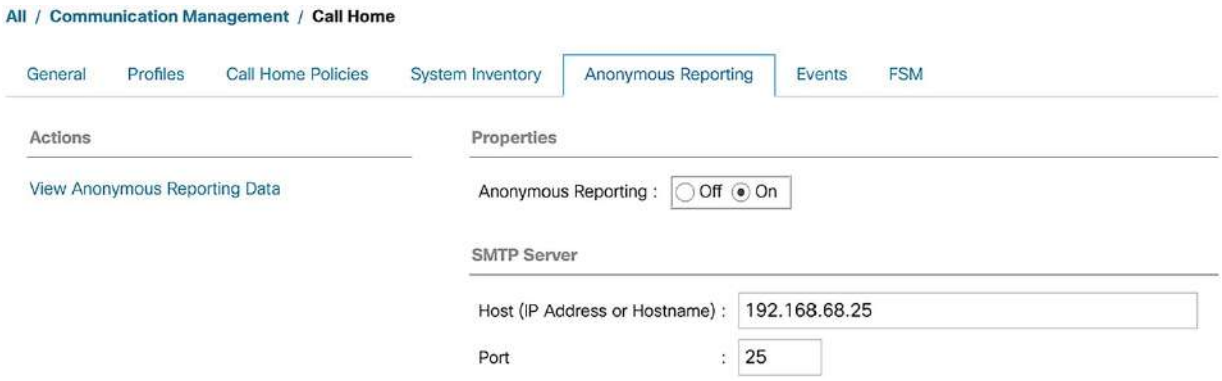

Figure 7-8 Enabling Anonymous Reporting

As we configure call home, we will see the actions we have taken in the events tab, and this is where we can look to see if call home has been triggered in the event of an issue, or if we have periodic reporting turned on (Figure 7-9).

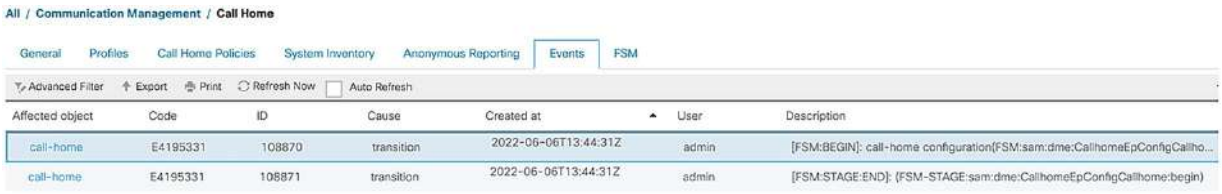

Figure 7-9 The Call Home events

Finally, the call home settings will be replicated from the primary FI to the secondary, and we can check the progress of this in the FSM tab (Figure 7-10).

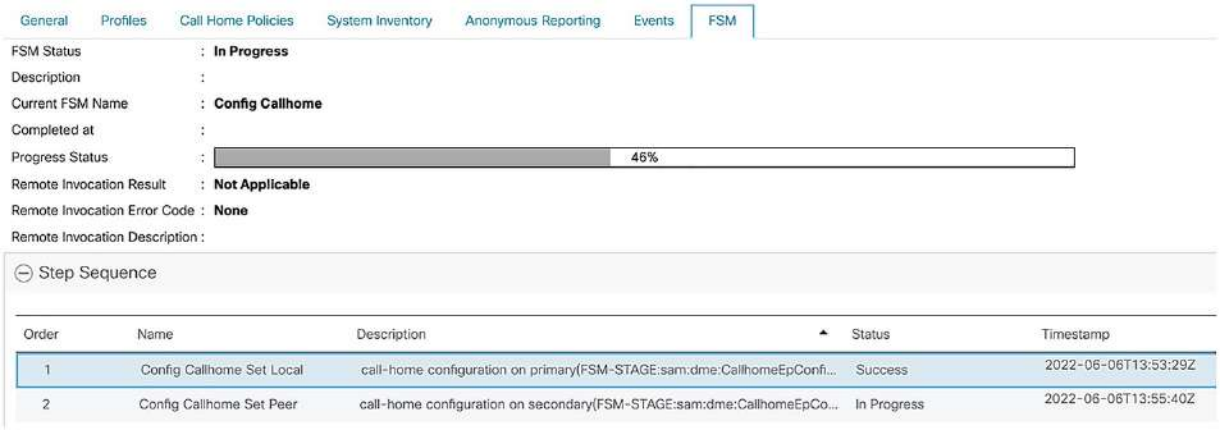

Figure 7-10 The Call Home FSM

SNMP

SNMP (Simple Network Monitoring Protocol) also allows us to be proactive in our management of the UCS. We can use a pull method, whereby the UCS can be polled by a network monitoring system (NMS) such as Zabbix, or a push method utilizing SNMP traps to capture a specific event and send them to an NMS.

To configure SNMP, go to "*Admin ➤ Communication Management ➤ Communication Services ➤ SNMP*" and click "Enabled" next to "Admin State." We then need to set the community or username, the protocol (TCP, UDP, or both), and the contact and location.

The UCS also supports SNMPv3 for greater security.

Logging and Events

While the use of Call Home and SNMP is a somewhat proactive approach, from time to time, we will need to go digging a bit deeper to find what issues we have, and this is certainly something we will need to do before we try and do an upgrade as upgrades can and will fail if we have critical issues with our UCS.

We can see how many issues we have from anywhere in the GUI as we have a group of icons at the top of the screen, as shown in Figure 7-11.

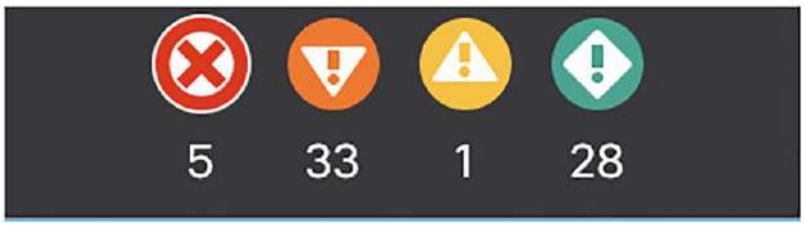

Figure 7-11 The GUI shows the active alerts

In the preceding, we have five critical issues, thirty-three major, one minor, and 28 warnings. We can see what these actually are by going to "Admin ➤ Faults, Events and Audit Logs," which shows the preceding categories as well as "Info," "Condition," "Cleared," "Soaking," and "Suppressed." There is another state of "flapping." The difference between the states is shown in Table 7-1.

Table 7-1 The different fault levels

Level	Description
Critical	A service-affecting condition that requires immediate resolution
Major	A service-affecting condition that requires urgent resolution
Minor	A non-service-affecting condition that requires fixing to prevent a more severe fault from occurring in the future
Warning	A potential or impending service-impacting fault but which currently has no significant effect on the system

Level	Description
Info	A basic notification or informational message
Condition	An informational message about a condition
Cleared	The fault has been cleared
Flapping	A fault has been raised, cleared and raised again within a short period of time.
Soaking	A fault has been raised and cleared within a short time interval (the flap interval), but it could be flapping, but if it doesn't reoccur then it moves into the cleared state
Suppressed	SNMP traps and call home notifications have been stopped

By default we will see all the levels apart from Soaking and Suppressed (Figure 7-12).

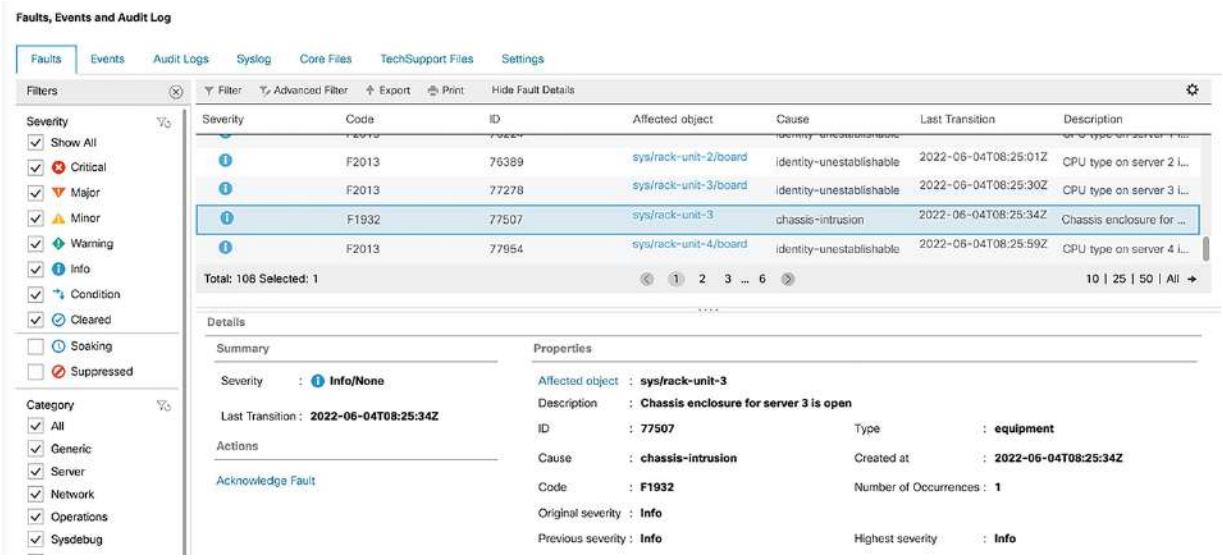

Figure 7-12 The UCS faults

We can acknowledge the faults that have been cleared and they will be removed from the list (Figure 7-13).

Figure 7-13 Fault acknowledgment

Faults are only removed once they have been resolved, so our UCSPE will remain in this state (unless we remove most of the configurations that we have put in that will not work in this sandboxed environment.

We can tackle faults as we see fit, such as starting with the most important ones (Figure 7-14), and going through them level by level.

Figure 7-14 Filtering faults

We can also look at events, which show us the affected object and the FSM description (Figure 7-15).

Figure 7-15 The event log

We also have the audit logs, allowing us to see who did what and when (Figure 7-16).

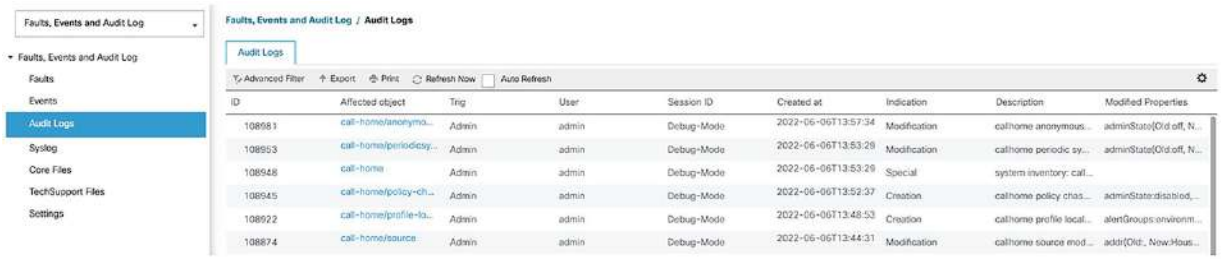

Figure 7-16 The audit log

SYSLOG

Having faults, events, and audit logs in the GUI is fine, but really all of this data should be sent to a more centralized logging system, and we can do that by using a SYSLOG server (Figure 7-17). We go to "*Admin ➤ All ➤ Faults, Events and Audit Log ➤ Syslog*," enabling the service and entering the hostname or IP address of the SYSLOG server.

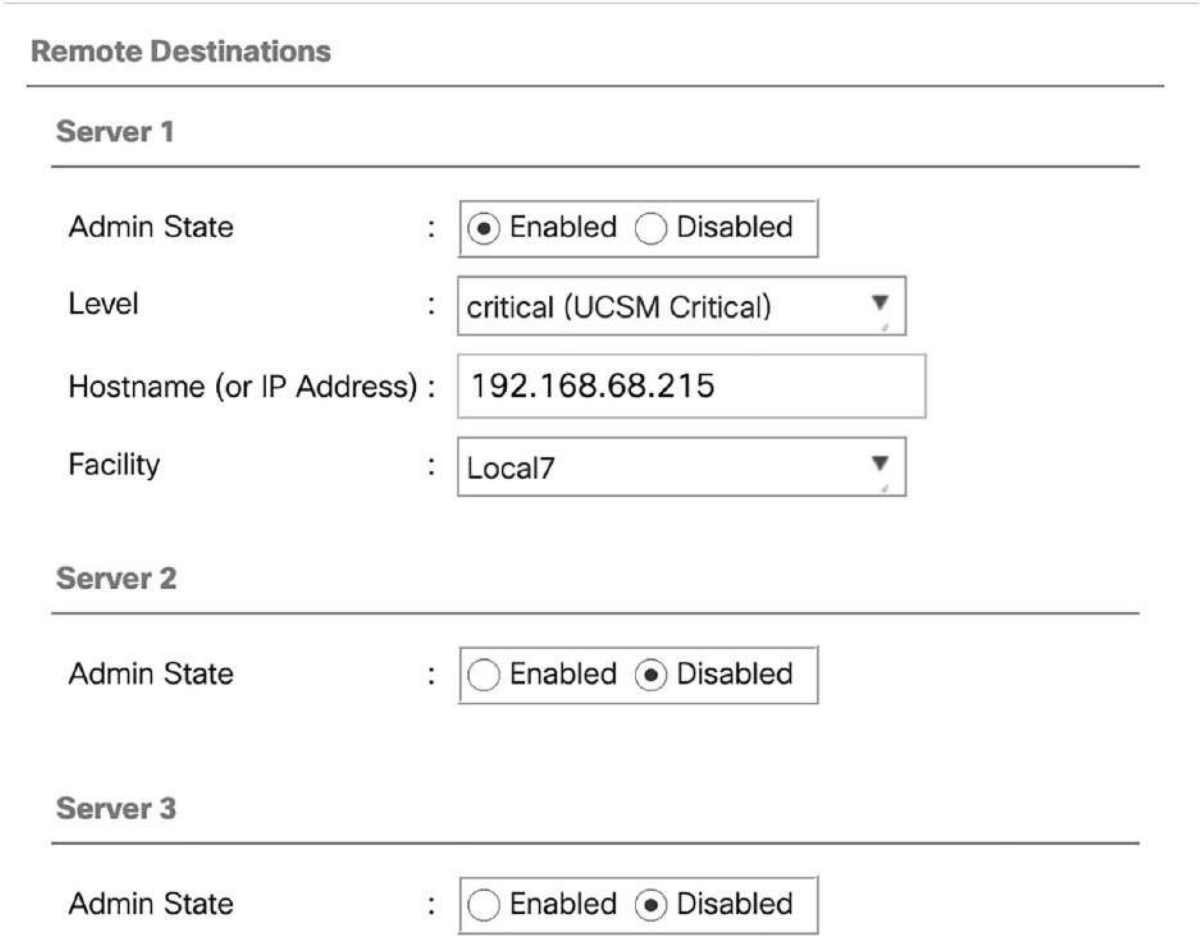

Figure 7-17 Setting up remote syslog

Techsupport Files

No matter how proactive we are, there will be times that we need to reach out to Cisco TAC and you will need to send them technical support files. We generate these on the UCS and download them to send over to Cisco. Navigate to "*Admin ➤ All ➤ Faults, Events and Audit Log ➤ TechSupport Files*." Click on "Add." You will see that the options are to "Create and Download a Tech Support File," or to just "Create a Tech Support File" (which you can download at another time), as shown in Figure 7-18.

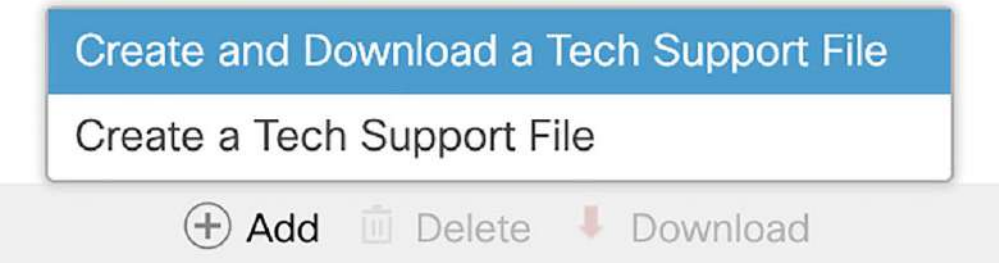

Figure 7-18 Generating a techsupport file

Whichever option we choose, we get the same options for what we want to generate. For the most part, UCSM will be sufficient, but what tech support file we generate does depend on the situation at hand. The options we have are shown in Figure 7-19 and explained in Table 7-2.

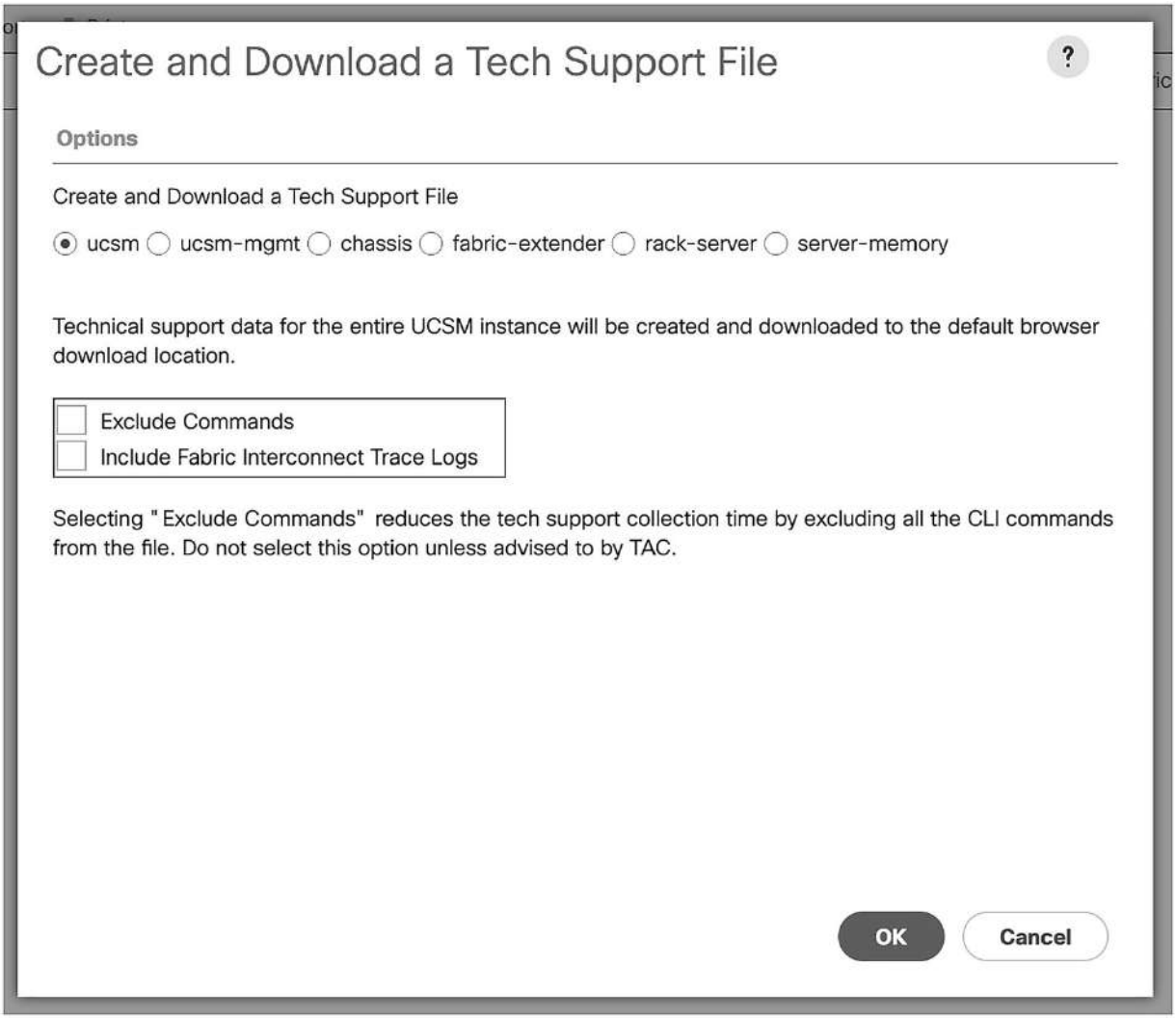

Figure 7-19 The techsupport file options

Table 7-2.

Option	Description
ucsm	Covers the entire UCS domain, but does not include chassis, fabric-extender, rack-server, or server memory
ucsm-mgmt	The UCS management services (but not the fabric interconnects)
Chassis	Either the CIMCs or I/O modules in a specific chassis
fabric-extender	As the name suggests, this is for the FIs
rack-server	The rack servers

Option	Description
server-memory	Includes B-series and C-series memory support data

In this way, we can be more targeted and send the tech support file for a particular rack server, for instance (Figure 7-20).

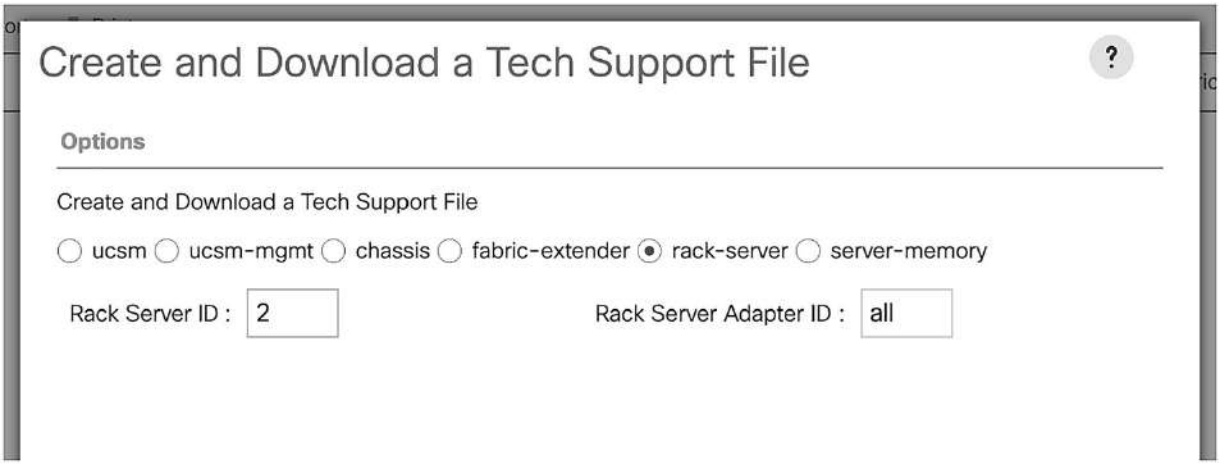

Figure 7-20 A rack server techsupport file

The resulting files can then be uploaded to the Cisco TAC case.

Summary

In this chapter, we looked at ways to help us troubleshoot issues with the UCS.

Thanks for reading; I hope the book has been useful to you.

Index

A

B

C

D, E

F

G

H

I, J

K

L

M

N

O

P, Q

R

S

T

U

V

W, X, Y

Z