FlexPod Datacenter with Oracle 21c RAC DNFS, on Cisco UCS X-Series, 100G Fabric, and NetApp AFF800

Design and Deployment Guide

Published: May 2023

In partnership with:

About the Cisco Validated Design Program

The Cisco Validated Design (CVD) program consists of systems and solutions designed, tested, and documented to facilitate faster, more reliable, and more predictable customer deployments. For more information, go to: http://www.cisco.com/go/designzone.

Executive Summary

Cisco Validated Designs include systems and solutions that are designed, tested, and documented to facilitate and improve customer deployments. The success of the FlexPod solution is driven through its ability to evolve and and incorporate both technology and product innovations in the areas of management, compute, storage, and networking. This document explains the design details of incorporating the Cisco X-Series modular platform with end-to-end 100Gbps networking into the FlexPod Datacenter and the ability to monitor and manage FlexPod components from the cloud using Cisco Intersight.

The FlexPod Datacenter with NetApp All Flash AFF system is a converged infrastructure platform that combines best-of breed technologies from Cisco and NetApp into a powerful converged platform for enterprise applications. Cisco and NetApp works closely with Oracle to support the most demanding transactional and response-time-sensitive databases required by today's businesses.

This Cisco Validated Design (CVD) describes the reference FlexPod Datacenter architecture using Cisco UCS X-Series and NetApp All Flash AFF Storage for deploying a highly available Oracle 21c RAC Databases environment. This document shows the hardware and software configuration of the components involved, results of various tests and offers implementation and best practices guidance using Cisco UCS X-Series Compute Servers, Cisco Fabric Interconnect Switches, Cisco Nexus Switches, NetApp AFF Storage and Oracle RAC Database.

Like all other FlexPod solution designs, FlexPod Datacenter with end-to-end 100Gbps Ethernet is configurable according to demand and usage. Customers can purchase exactly the infrastructure they need for their current application requirements and can then scale up by adding more resources to the FlexPod system or scale out by adding more FlexPod instances. By moving the management from the fabric interconnects into the cloud, the solution can respond to the speed and scale of customer deployments with a constant stream of new capabilities delivered from Cisco Intersight software-as-a-service model at cloud-scale. For customers that require management within the secure site, Cisco Intersight is also offered within an on-site appliance with both connected and not connected or air gap options.

Solution Overview

This chapter contains the following:

Introduction

The Cisco Unified Computing System X-Series (Cisco UCS-X) with Intersight Managed Mode (IMM) is a modular compute system, configured and managed from the cloud. It is designed to meet the needs of modern applications and to improve operational efficiency, agility, and scale through an adaptable, future-ready, modular design. The Cisco Intersight platform is a Software-as-a-Service (SaaS) infrastructure lifecycle management platform that delivers simplified configuration, deployment, maintenance, and support.

Powered by the Cisco Intersight cloud-operations platform, the Cisco UCS with X-Series enables the next-generation cloud-operated FlexPod infrastructure that not only simplifies data-center management but also allows the infra-structure to adapt to the unpredictable needs of modern applications as well as traditional workloads. With the Cisco Intersight platform, customers get all the benefits of SaaS delivery and the full lifecycle management of Cisco Intersight-connected distributed servers and integrated NetApp storage systems across data centers, remote sites, branch offices, and edge environments.

This CVD describes how the Cisco Unified Computing System (Cisco UCS) X-Series can be used in conjunction with NetApp AFF All Flash storage systems to implement a mission-critical application such as an Oracle 21c Real Application Clusters (RAC) databases solution using end to end 100G on Oracle dNFS. This CVD documents validation of the real-world performance, ease of management, and agility of the FlexPod Datacenter with Cisco UCS and All Flash AFF in high-performance Oracle RAC Databases environments.

Audience

The intended audience for this document includes, but is not limited to, sales engineers, field consultants, database administrators, IT architects, Oracle database architects, and customers who want to deploy Oracle RAC 21c database solution on FlexPod Converged Infrastructure with NetApp clustered Data ONTAP and the Cisco UCS X-Series platform using Intersight Managed Mode (IMM) to deliver IT efficiency and enable IT innovation. A working knowledge of Oracle RAC Database, Linux, Storage technology, and Network is assumed but is not a prerequisite to read this document.

Purpose of this Document

This document provides a step-by-step configuration and implementation guide for the FlexPod Datacenter with Cisco UCS X-Series Compute Servers, Cisco Fabric Interconnect Switches, Cisco Nexus Switches and NetApp

AFF Storage to deploy an Oracle RAC Database solution. This document provides reference for incorporating Cisco Intersight—managed Cisco UCS X-Series platform with end-to-end 100Gbps within FlexPod Datacenter infrastructure. The document introduces various design elements and explains various considerations and best practices for a successful deployment.

The document also highlights the design and product requirements for integrating compute, network, and storage systems to Cisco Intersight to deliver a true cloud-based integrated approach to infrastructure management. The goal of this document is to build, validate and evaluate the performance of this FlexPod reference architecture while running various types of Oracle OLTP and OLAP workloads using various benchmarking exercises and showcase Oracle database server read latency, peak sustained throughput and IOPS under various stress tests.

What's New in this Release?

The following design elements distinguish this version of FlexPod from previous models:

- Integration of Cisco UCS X-Series into FlexPod Datacenter
- Deploying and managing Cisco UCS X9508 chassis equipped with Cisco UCS X210c M6 compute nodes from the cloud using Cisco Intersight
- End-to-End 100Gbps Ethernet in FlexPod Datacenter
- Integration of the 5^{th} Generation Cisco UCS 6536 Fabric Interconnect into FlexPod Datacenter
- Integration of the 5^{th} Generation Cisco UCS 15000 Series VICs into FlexPod Datacenter
- Integration of the Cisco UCSX-I-9108-100G Intelligent Fabric Module into the Cisco X-Series 9508 Chassis
- Implementation of Oracle Direct NFS (dNFS) using end-to-end 100G network to optimize the I/O path between Oracle databases and the NFS Server
- Validation of Oracle 21c Grid Infrastructure and 21c Databases
- Support for the release of NetApp ONTAP 9.12.1

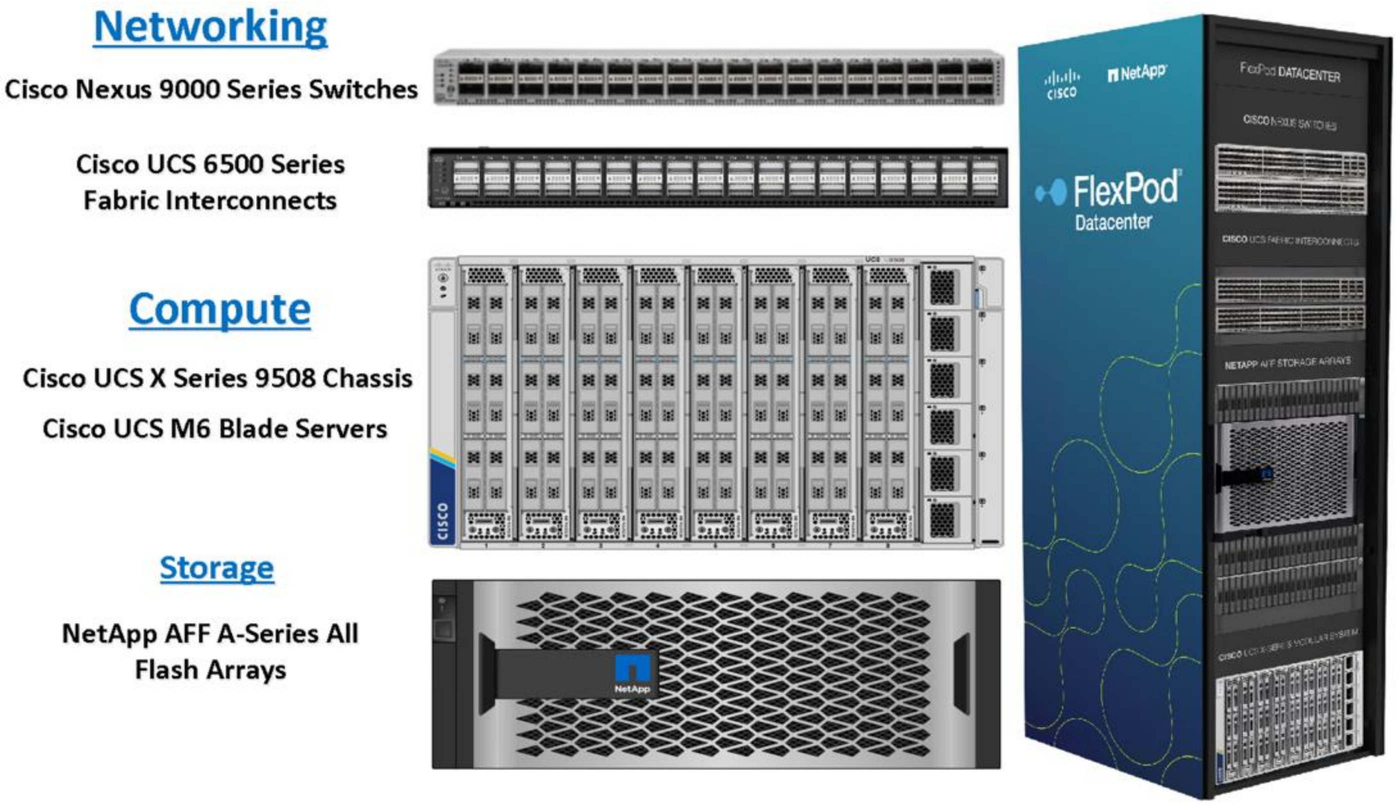
FlexPod System Overview

Built on groundbreaking technology from NetApp and Cisco, the FlexPod converged infrastructure platform meets and exceeds the challenges of simplifying deployments for best-in-class data center infrastructure. FlexPod is a defined set of hardware and software that serves as an integrated foundation for both virtualized and non-virtualized solutions. Composed of pre-validated storage, networking, and server technologies, FlexPod is designed to increase IT responsiveness to organizational needs and reduce the cost of computing with maximum uptime and minimal risk. Simplifying the delivery of data center platforms gives enterprises an advantage in delivering new services and applications.

FlexPod provides the following differentiators:

- Flexible design with a broad range of reference architectures and validated designs.
- Elimination of costly, disruptive downtime through Cisco UCS and NetApp ONTAP.
- Leverage a pre-validated platform to minimize business disruption and improve IT agility and reduce deployment time from months to weeks.

- Cisco Validated Designs (CVDs) and NetApp Validated Architectures (NVAs) covering a variety of use cases.



Cisco and NetApp have carefully validated and verified the FlexPod solution architecture and its many use cases while creating a portfolio of detailed documentation, information, and references to assist customers in transforming their data centers to this shared infrastructure model.

This reference FlexPod Datacenter architecture is built using the following infrastructure components for compute, network, and storage:

- Compute – Cisco UCS X-Series Chassis with Cisco UCS X210c M6 Blade Servers
- Network – Cisco UCS Fabric Interconnects and Cisco Nexus switches
- Storage – NetApp AFF All Flash Storage systems

All the FlexPod components have been integrated so that customers can deploy the solution quickly and economically while eliminating many of the risks associated with researching, designing, building, and deploying similar solutions from the foundation. One of the main benefits of FlexPod is its ability to maintain consistency at scale. Each of the component (Cisco UCS, Cisco FI, Cisco Nexus, and NetApp controllers) families shown in figure above offers platform and resource options to scale up or scale out the infrastructure while supporting the same features.

Solution Summary

This solution provides an end-to-end architecture with Cisco Unified Computing System (Cisco UCS) and NetApp technologies to demonstrates the benefits for running Oracle Multitenant RAC Databases 21c environment with excellent performance, scalability and high availability using NFS. The design is flexible enough that the networking, computing, and storage can fit in one data center rack or be deployed according to a customer's data center design. The reference architecture reinforces the "wire-once" strategy, because as additional storage is added to the architecture, no re-cabling is required from the hosts to the Cisco UCS fabric interconnect.

This FlexPod Datacenter solution for deploying Oracle RAC 21c Databases with end-to-end 100Gbps Ethernet is built using the following hardware components:

- Fifth-generation Cisco UCS 6536 Fabric Interconnects to support 10/25/40/100GbE and Cisco Intersight platform to deploy, maintain and support UCS and FlexPod components.
- Two Cisco UCS X9508 Chassis with each chassis having two Cisco UCSX-I-9108-100G Intelligent Fabric Modules to deploy end to end 100GE connectivity.
- Total eight Cisco UCS X210c M6 Compute Nodes (4 Nodes per Chassis) with each node having one Cisco Virtual Interface Cards (VICs) 15231.

Note: Cisco UCS X210c M7 compute nodes are available today and they offer the opportunity for even better performance if incorporated into this FlexPod design.

- High-speed Cisco NX-OS-based Cisco Nexus C9336C-FX2 switching design to support up to 100GE connectivity.
- NetApp AFF A800 end-to-end NVMe storage with 100GE connectivity.

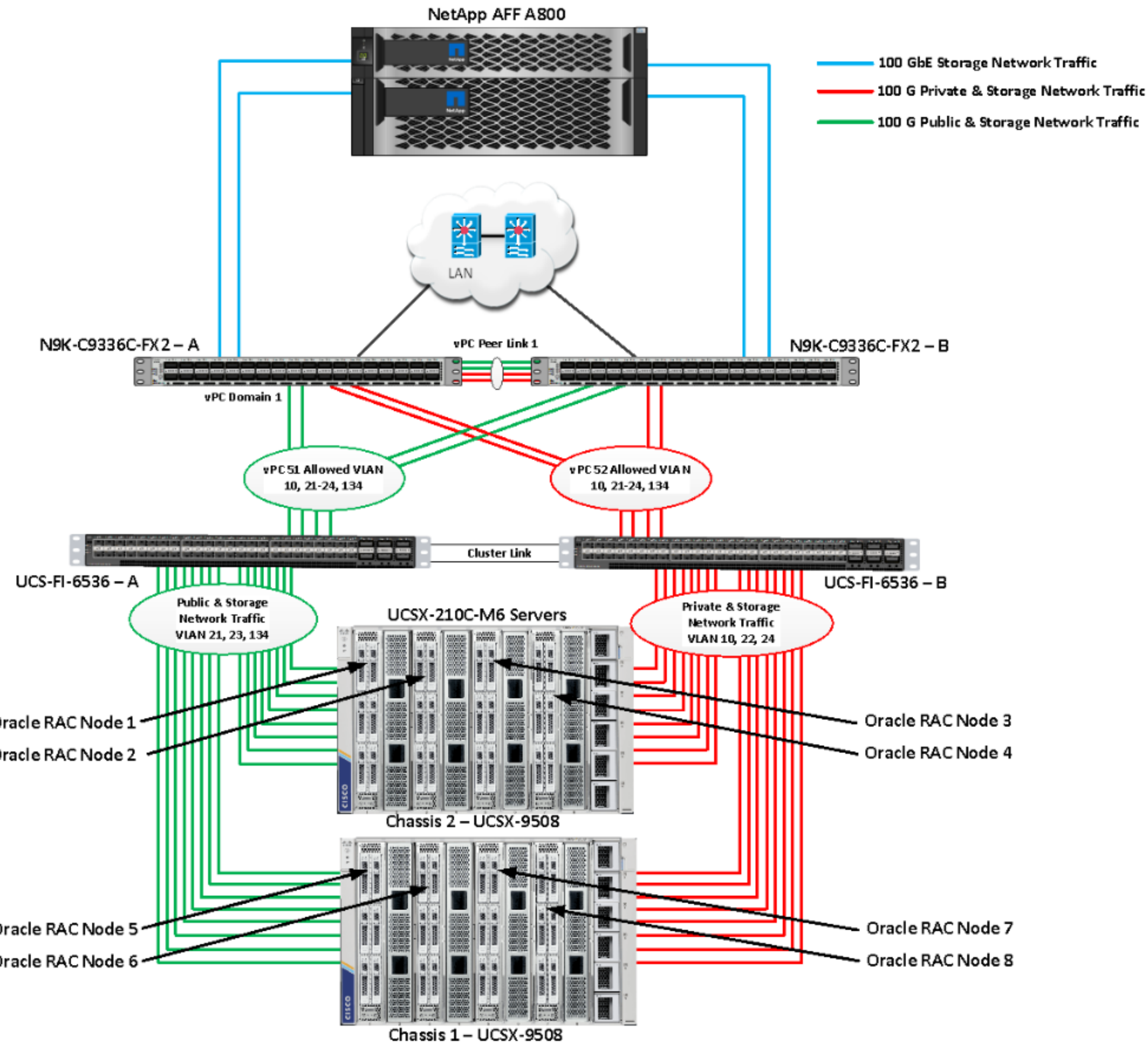
There are two modes to configure Cisco UCS, one is UCSM (UCS Managed) and the other is IMM (Intersight Managed Mode). This reference solution was deployed using Intersight Managed Mode (IMM). The best practices and setup recommendations are described later in this document.

Note: In this validated and deployed solution, the Cisco X-series is currently only supported in IMM mode.

Physical Topology

Figure 1 shows the architecture diagram of the FlexPod components to deploy an eight node Oracle RAC 21c Database solution on end-to-end 100GbE on IP based storage access NFS. This reference design is a typical network configuration that can be deployed in a customer's environments.

Figure 1. FlexPod components architecture



As shown in Figure 1, a pair of Cisco UCS 6536 Fabric Interconnects (FI) carries both storage and network traffic from the Cisco UCS X210c M6 server with the help of Cisco Nexus 9336C-FX2 switches. Both the Fabric Interconnects and the Cisco Nexus switches are clustered with the peer link between them to provide high availability.

As illustrated in above figure, 16 (8 x 100G link per chassis) links from the blade server chassis go to Fabric Interconnect – A. Similarly, 16 (8 x 100G link per chassis) links from the blade server chassis go to Fabric Interconnect – B. Fabric Interconnect – A links are used for Oracle Public Network Traffic (VLAN-134) and Storage Network Traffic (VLAN-21 and 23) shown as green lines while Fabric Interconnect – B links are used for Oracle Private Interconnect Traffic (VLAN 10) and Storage Network Traffic (VLAN-22 and 24) shown as red lines. Three virtual Port-Channels (vPCs) are configured to provide public network, private network, and storage network traffic paths for the server blades to northbound nexus switches and NFS storage system.

The Network File System (NFS) Storage access from both fabric interconnects to Cisco Nexus Switches and NetApp Storage Array are shown as blue lines.

Note: For the Oracle RAC configuration on Cisco Unified Computing System, we recommend keeping all private interconnects network traffic local on a single Fabric interconnect. In such a case, the private traffic will stay local to that fabric interconnect and will not be routed via northbound network switch. In that way, all the inter server blade (or RAC node private) communications will be resolved locally at the fabric interconnects and this significantly reduces latency for Oracle Cache Fusion traffic.

Additional 1Gb management connections will be needed for an out-of-band network switch that sits apart from this FlexPod infrastructure. Each Cisco UCS FI and Cisco Nexus switch is connected to the out-of-band network switch, and each NetApp AFF controller also has two connections to the out-of-band network switch.

Although this is the base design, each of the components can be scaled easily to support specific business requirements. For example, more servers or even blade chassis can be deployed to increase compute capacity, additional disk shelves can be deployed to improve I/O capability and throughput, and special hardware or software features can be added to introduce new features. This document guides you through the detailed steps for deploying the base architecture, as shown in Figure 1. These procedures cover everything from physical cabling to network, compute, and storage device configurations.

Design Topology

This section describes the hardware and software components used to deploy an eight node Oracle RAC 21c Databases Solution on this architecture.

The inventory of the components used in this solution architecture is listed in Table 1.

Table 1. Table for Hardware Inventory and Bill of Material

Name	Model/Product ID	Description	Quantity
Cisco UCS X Blade Server Chassis	UCSX-9508	Cisco UCS X Series Blade Server Chassis, 7RU which can house a combination of compute nodes and a pool of future I/O resources that may include GPU accelerators, disk storage, and nonvolatile memory.	2
Cisco UCS 9108 100G IFM (Intelligent Fabric Module)	UCSX-I-9108-100G	Cisco UCS 9108 100G IFM connects the I/O fabric between the Cisco UCS X9508 Chassis and 6536 Fabric Interconnects 800Gb/s (8x100Gb/s) Port IO Module for 8 compute nodes	4
Cisco UCS X210c M6 Compute Server	UCSX-210C-M6	Cisco UCS X210c M6 2 Socket Blade Server (2x 3rd Gen Intel Xeon Scalable Processors)	8
Cisco UCS VIC 15231	UCSX-ML-V5D200G	Cisco UCS VIC 15231 2x100/200G mLOM for X Compute Node	8
Cisco UCS 6536 Fabric Interconnect	UCS-FI-6536	Cisco UCS 6536 Fabric Interconnect providing both network connectivity and management capabilities for the system	2

Name	Model/Product ID	Description	Quantit y
Cisco Nexus Switch	N9K-9336C-FX2	Cisco Nexus 9336C-FX2 Switch	2
NetApp AFF Storage	AFF A800	NetApp AFF A-Series All Flash Arrays	1

In this solution design, we used 8 identical Cisco UCS X210c M6 Blade Servers to configure the Oracle Linux 8.6 Operating system and then deploy an 8 node Oracle RAC Databases. The Cisco UCS X210c M6 Server configuration is listed in Table 2.

Table 2. Cisco UCS X210c M6 Compute Server Configuration

Cisco UCS X210c M6 Server Configuration	
Processor	2 x Intel(R) Xeon(R) Gold 6348 CPU @ 2.60GHz (56 CPU Cores)
Memory	16 x Samsung 32GB DDR4-3200-MHz (512 GB)
VIC 15231	Cisco UCS VIC 15231 Blade Server MLOM (200G for compute node) (2x100G through each fabric)

Table 3. vNIC Configured on each Linux Host

vNIC Details	
vNIC 0 (eth0)	Management and Public Network Traffic Interface for Oracle RAC. MTU = 1500
vNIC 1 (eth1)	Private Server-to-Server Network (Cache Fusion) Traffic Interface for Oracle RAC. MTU = 9000
vNIC 2 (eth2)	Database IO Traffic to NetApp Storage Controller. VLAN 21. MTU=9000
vNIC 3 (eth3)	Database IO Traffic to NetApp Storage Controller. VLAN 22. MTU=9000
vNIC 4 (eth4)	Database IO Traffic to NetApp Storage Controller. VLAN 23. MTU=9000
vNIC 5 (eth5)	Database IO Traffic to NetApp Storage Controller. VLAN 24. MTU=9000

Six VLANs were configured to carry public, private, and storage VLAN traffic as listed in Table 4.

Table 4. VLAN Configuration

VLAN Configuration

VLAN Configuration		
VLAN		
Name	ID	Description
Default VLAN	1	Native VLAN
Public VLAN	134	VLAN for Public Network Traffic
Private VLAN	10	VLAN for Private Network Traffic
Storage VLAN 21	21	NFS VLAN for Storage Network Traffic Through FI-A Side
Storage VLAN 22	22	NFS VLAN for Storage Network Traffic Through FI-B Side
Storage VLAN 23	23	NFS VLAN for Storage Network Traffic Through FI-A Side
Storage VLAN 24	24	NFS VLAN for Storage Network Traffic Through FI-B Side

This FlexPod solution consist of NetApp All Flash AFF Series Storage as listed in Table 5.

Table 5. NetApp AFF A800 Storage Configuration

Storage Components	Description
AFF Flash Array	NetApp All Flash AFF A800 Storage Array (24 x 1.75 TB NVMe SSD Drives)
Capacity	41.82 TB
Connectivity	4x100 Gb/s (2x100 G per Controller) (Data Rate: 100 Gb/s Ethernet, PCI Express Gen3: SERDES @ 8.0GT/s, 16 lanes) (MCX516A-CCAT) 1 Gb/s redundant Ethernet (Management port)
Physical	4 Rack Units

Table 6. Software and Firmware Revisions

Software and Firmware	Version
Cisco UCS FI 6536	Bundle Version 4.2(3b) or NX-OS Version – 9.3(5)I42(3b) Image Name - intersight-ucs-infra-5gfi.4.2.3b.bin
Cisco UCS X210c M6 Server	5.0(4a) Image Name - intersight-ucs-server-210c-m6.5.0.4a.bin

Software and Firmware	Version
Cisco UCS Adapter VIC 1440	5.2(3c)
Cisco eNIC (Cisco VIC Ethernet NIC Driver) (modinfo enic)	4.3.0.1-918.18 (kmod-enic-4.3.0.1-918.18.oluek_5.4.17_2136.307.3.1.x86_64)
Oracle Linux Server	Oracle Linux Release 8 Update 6 for x86 (64 bit) (Kerel - 5.4.17-2136.307.3.1.el8uek.x86_64)
Oracle Database 21c Grid Infrastructure for Linux x86-64	21.3.0.0.0
Oracle Database 21c Enterprise Edition for Linux x86-64	21.3.0.0.0
Cisco Nexus 9336C-FX2 NXOS	9.2(3)
NetApp Storage AFF A800	ONTAP 9.12.1P1
FIO	fio-3.19-3.el8.x86_64
Oracle Swingbench	2.5.971
SLOB	2.5.4.0

Solution Configuration

This chapter contains the following:

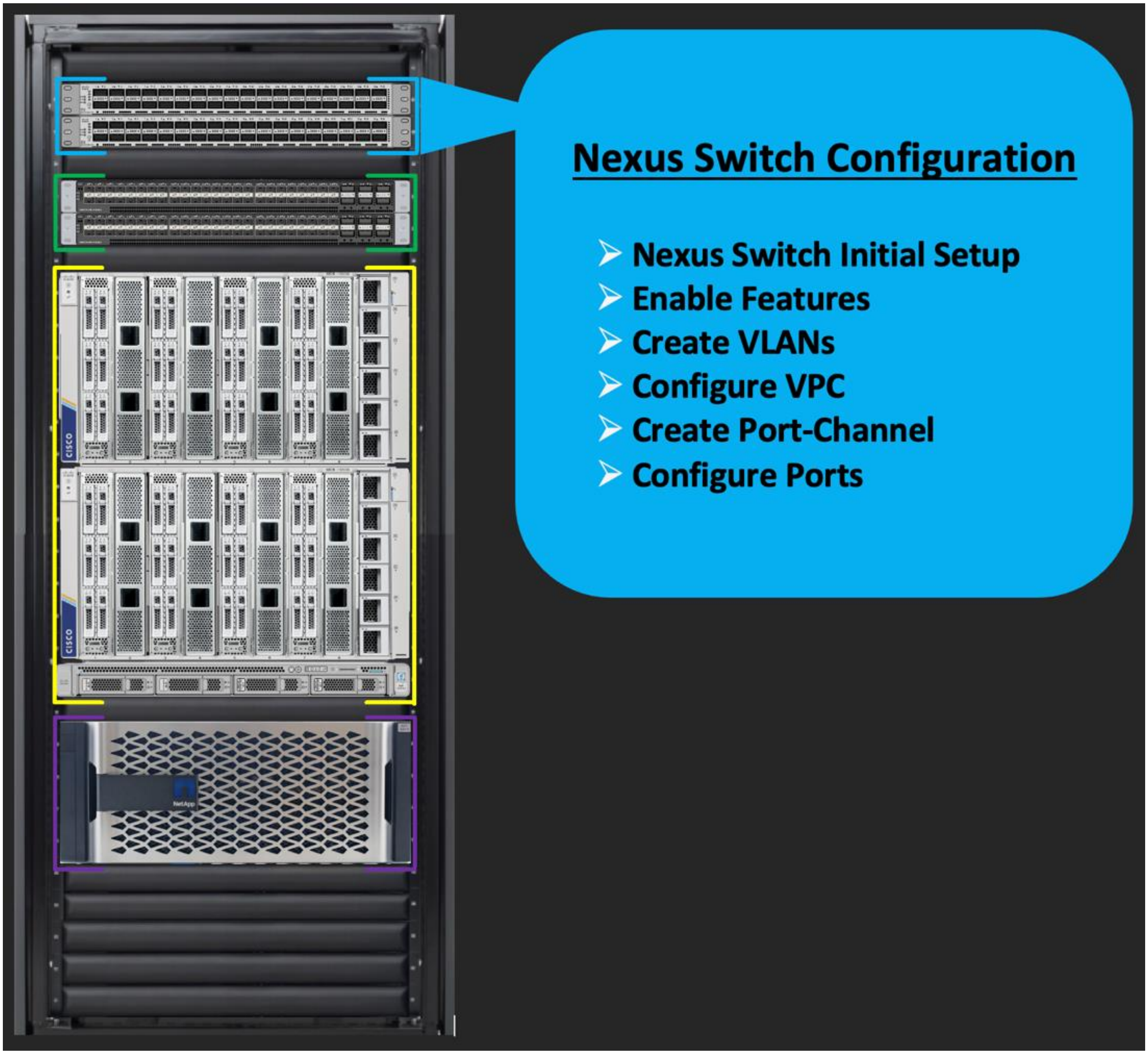
- Cisco Nexus Switch Configuration
- Cisco UCS X-Series Configuration – Intersight Managed Mode (IMM)
- NetApp AFF A800 Storage Configuration

Cisco Nexus Switch Configuration

This section details the high-level steps to configure Cisco Nexus Switches.

Figure 2 illustrates the high-level overview and steps for configuring various components to deploy and test the Oracle RAC Database 21c on this FlexPod reference architecture.

Figure 2. Cisco Nexus Switch configuration architecture



The following procedures describe how to configure the Cisco Nexus switches for use in a base FlexPod environment. This procedure assumes you're using Cisco Nexus 9336C-FX2 switches deployed with the 100Gb end-to-end topology.

Note: On initial boot and connection to the serial or console port of the switch, the NX-OS setup should automatically start and attempt to enter Power on Auto Provisioning.

Cisco Nexus A Switch

Procedure 1. Initial Setup for the Cisco Nexus A Switch

Step 1. To set up the initial configuration for the Cisco Nexus A switch on <nexus-A-hostname>, follow these steps:

```
Abort Power on Auto Provisioning and continue with normal setup? (yes/no) [n]: yes
Do you want to enforce secure password standard (yes/no) [y]: Enter
Enter the password for "admin": <password>
Confirm the password for "admin": <password>
Would you like to enter the basic configuration dialog (yes/no): yes
Create another login account (yes/no) [n]: Enter
Configure read-only SNMP community string (yes/no) [n]: Enter
Configure read-write SNMP community string (yes/no) [n]: Enter
Enter the switch name: <nexus-A-hostname>
Continue with Out-of-band (mgmt0) management configuration? (yes/no) [y]: Enter
Mgmt0 IPv4 address: <nexus-A-mgmt0-ip>
Mgmt0 IPv4 netmask: <nexus-A-mgmt0-netmask>
Configure the default gateway? (yes/no) [y]: Enter
IPv4 address of the default gateway: <nexus-A-mgmt0-gw>
Configure advanced IP options? (yes/no) [n]: Enter
Enable the telnet service? (yes/no) [n]: Enter
Enable the ssh service? (yes/no) [y]: Enter
Type of ssh key you would like to generate (dsa/rsa) [rsa]: Enter
Number of rsa key bits <1024-2048> [1024]: Enter
Configure the ntp server? (yes/no) [n]: y
NTP server IPv4 address: <global-ntp-server-ip>
Configure default interface layer (L3/L2) [L3]: L2
Configure default switchport interface state (shut/noshut) [noshut]: Enter
Configure CoPP system profile (strict/moderate/lenient/dense/skip) [strict]: Enter
Would you like to edit the configuration? (yes/no) [n]: Enter
```

Cisco Nexus B Switch

Similarly, follow the steps in the procedure Initial Setup for the Cisco Nexus A Switch to setup the initial configuration for the Cisco Nexus B Switch and change the relevant switch hostname and management IP address according to your environment.

Procedure 1. Configure Global Settings

Configure the global setting on both Cisco Nexus Switches.

Step 1. Login as admin user into the Cisco Nexus Switch A and run the following commands to set the global configurations on switch A:

```
configure terminal
feature interface-vlan
feature hsrp
feature lacp
feature vpc
feature lldp
spanning-tree port type network default
spanning-tree port type edge bpduguard default

port-channel load-balance src-dst l4port

policy-map type network-qos jumbo
  class type network-qos class-default
    mtu 9216

system qos
  service-policy type network-qos jumbo

vrf context management
  ip route 0.0.0.0/0 10.29.135.1
copy run start
```

Step 2. Login as admin user into the Nexus Switch B and run the same above commands to set global configurations on Nexus Switch B.

Note: Make sure to run copy run start to save the configuration on each switch after the configuration is completed.

Procedure 2. VLANs Configuration

Create the necessary virtual local area networks (VLANs) on both Cisco Nexus switches.

Step 1. Login as admin user into the Cisco Nexus Switch A.

Step 2. Create VLAN 134 for Public Network Traffic, VLAN 10 for Private Network Traffic, and VLAN 21,22,23,24 for Storage Network Traffic.

```
configure terminal

vlan 134
name Oracle_RAC_Public_Traffic
no shutdown

vlan 10
```

```
name Oracle_RAC_Private_Traffic
no shutdown
vlan 21
name Storage_Traffic_A1
no shutdown

vlan 22
name Storage_Traffic_B1
no shutdown

vlan 23
name Storage_Traffic_A2
no shutdown

vlan 24
name Storage_Traffic_B2
no shutdown

interface Ethernet1/29
  description To-Management-Uplink-Switch
  switchport access vlan 134
  speed 1000

copy run start
```

Step 3. Login as admin user into the Nexus Switch B and similar way, create all the VLANs (134,10,21,22,23 and 24) for Oracle RAC Public Network, Private Network and Storage Network Traffic.

Note: Make sure to run copy run start to save the configuration on each switch after the configuration is completed.

Virtual Port Channel (vPC) Summary for Network Traffic

A port channel bundles individual links into a channel group to create a single logical link that provides the aggregate bandwidth of up to eight physical links. If a member port within a port channel fails, traffic previously carried over the failed link switches to the remaining member ports within the port channel. Port channeling also load balances traffic across these physical interfaces. The port channel stays operational as long as at least one physical interface within the port channel is operational. Using port channels, Cisco NX-OS provides wider bandwidth, redundancy, and load balancing across the channels.

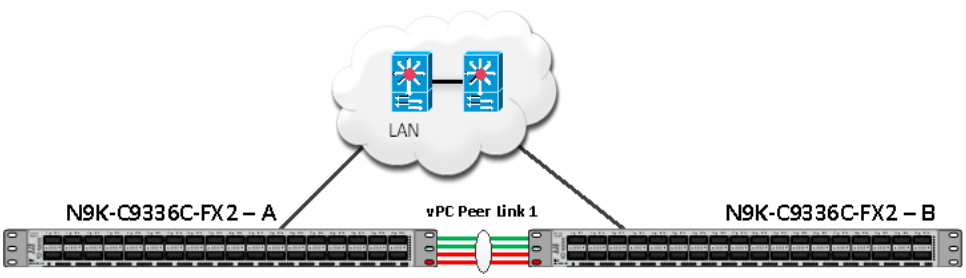
In the Cisco Nexus Switch topology, a single vPC feature is enabled to provide HA, faster convergence in the event of a failure, and greater throughput. The Cisco Nexus vPC configurations with the vPC domains and corresponding vPC names and IDs for Oracle Database Servers are listed in Table 7.

Table 7. vPC Summary

vPC Domain	vPC Name	vPC ID
1	Peer-Link	1
51	vPC FI-A	51
52	vPC FI-B	52
13	vPC Storage A	13
14	vPC Storage A	14

As listed in Table 7, a single vPC domain with Domain ID 1 is created across two Nexus switches to define vPC members to carry specific VLAN network traffic. In this topology, we defined a total number of 5 vPCs.

vPC ID 1 is defined as Peer link communication between the two Cisco Nexus switches. vPC IDs 51 and 52 are configured for both Cisco UCS fabric interconnects. vPC IDs 13 and 14 are configured between both Cisco Nexus Switches and NetApp Storage Controller.



Note: A port channel bundles up to eight individual interfaces into a group to provide increased bandwidth and redundancy.

Procedure 3. Create vPC Peer-Link

Note: For vPC 1 as Peer-link, we used interfaces 1 to 4 for Peer-Link. You may choose an appropriate number of ports based on your needs.

Create the necessary port channels between devices on both Cisco Nexus Switches.

Step 1. Login as admin user into the Cisco Nexus Switch A:

```
configure terminal

vpc domain 1
  peer-keepalive destination 10.29.134.44 source 10.29.134.43
  auto-recovery
```

```
interface port-channel 1
  description vPC peer-link
  switchport mode trunk
  switchport trunk allowed vlan 1,10,21-24,134
  spanning-tree port type network
  vpc peer-link
  no shut

interface Ethernet1/1
  description Peer link connected to ORA21C-N9K-B-Eth1/1
  switchport mode trunk
  switchport trunk allowed vlan 1,10,21-24,134
  channel-group 1 mode active
  no shut

interface Ethernet1/2
  description Peer link connected to ORA21C-N9K-B-Eth1/2
  switchport mode trunk
  switchport trunk allowed vlan 1,10,21-24,134
  channel-group 1 mode active
  no shut

interface Ethernet1/3
  description Peer link connected to ORA21C-N9K-B-Eth1/3
  switchport mode trunk
  switchport trunk allowed vlan 1,10,21-24,134
  channel-group 1 mode active
  no shut

interface Ethernet1/4
  description Peer link connected to ORA21C-N9K-B-Eth1/4
  switchport mode trunk
  switchport trunk allowed vlan 1,10,21-24,134
  channel-group 1 mode active
  no shut

exit
copy run start
```

Step 2. Login as admin user into the Cisco Nexus Switch B and repeat step 1 to configure the second Cisco Nexus Switch.

Note: Make sure to change the description of the interfaces and peer-keepalive destination and source IP addresses.

Step 3. Configure the vPC on the other Cisco Nexus switch. Login as admin for the Cisco Nexus Switch B:

```
configure terminal

vpc domain 1
  peer-keepalive destination 10.29.134.43 source 10.29.134.44
  auto-recovery

interface port-channel 1
  description vPC peer-link
  switchport mode trunk
  switchport trunk allowed vlan 1,10,21-24,134
  spanning-tree port type network
  vpc peer-link
  no shut

interface Ethernet1/1
  description Peer link connected to ORA21C-N9K-A-Eth1/1
  switchport mode trunk
  switchport trunk allowed vlan 1,10,21-24,134
  channel-group 1 mode active
  no shut

interface Ethernet1/2
  description Peer link connected to ORA21C-N9K-A-Eth1/2
  switchport mode trunk
  switchport trunk allowed vlan 1,10,21-24,134
  channel-group 1 mode active
  no shut

interface Ethernet1/3
  description Peer link connected to ORA21C-N9K-A-Eth1/3
  switchport mode trunk
  switchport trunk allowed vlan 1,10,21-24,134
  channel-group 1 mode active
  no shut

interface Ethernet1/4
```

```
    description Peer link connected to ORA21C-N9K-A-Eth1/4
    switchport mode trunk
    switchport trunk allowed vlan 1,10,21-24,134
    channel-group 1 mode active
    no shut

  exit
  copy run start
```

Create vPC Configuration between Cisco Nexus and Fabric Interconnect Switches

This section describes how to create and configure port channel 51 and 52 for network traffic between the Cisco Nexus and Fabric Interconnect Switches.

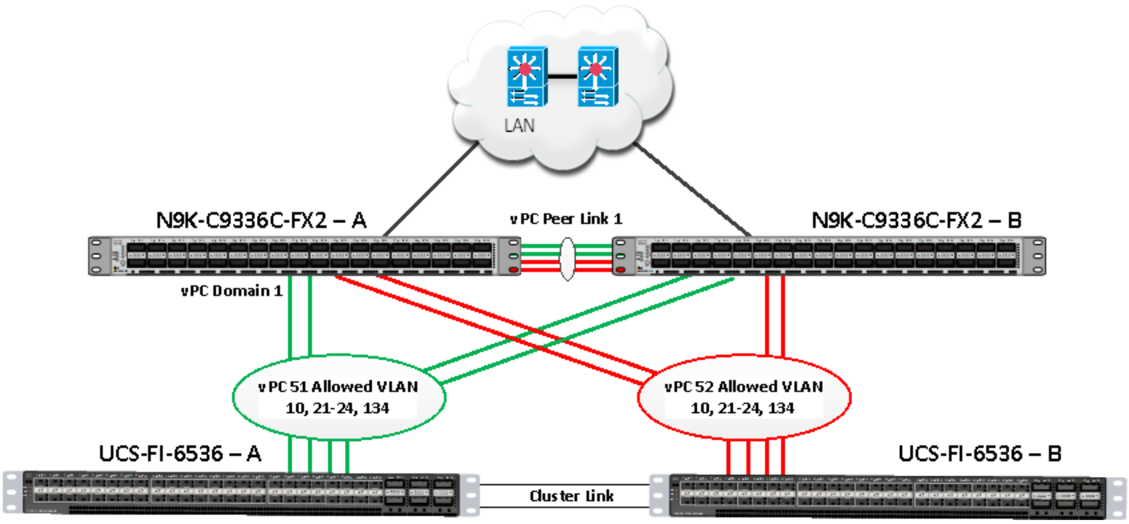


Table 8 lists the vPC IDs, allowed VLAN IDs, and ethernet uplink ports.

Table 8. vPC IDs and VLAN IDs

vPC Description	vPC ID	Fabric Interconnects Ports	Cisco Nexus Switch Ports	Allowed VLANs
Port Channel FI-A	51	FI-A Port 1/27	N9K-A Port 1/9	10,21,22,23,24,134 Note: VLAN 10,22,24 is needed for failover.
		FI-A Port 1/28	N9K-A Port 1/10	
		FI-A Port 1/29	N9K-B Port 1/9	
		FI-A Port 1/30	N9K-B Port 1/10	
Port Channel FI-B	52	FI-B Port 1/27	N9K-A Port	10,21,22,23,24,134

vPC Description	vPC ID	Fabric Interconnects Ports	Cisco Nexus Switch Ports	Allowed VLANs
			1/11	Note: VLAN 21,23,134 is needed for failover.
		FI-B Port 1/28	N9K-A Port 1/12	
		FI-B Port 1/29	N9K-B Port 1/11	
		FI-B Port 1/30	N9K-B Port 1/12	

Verify the port connectivity on both Cisco Nexus Switches

Cisco Nexus A Connectivity

```
ORA21C-N9K-A# show lldp neighbors
Capability codes:
  (R) Router, (B) Bridge, (T) Telephone, (C) DOCSIS Cable Device
  (W) WLAN Access Point, (P) Repeater, (S) Station, (O) Other
Device ID            Local Intf      Hold-time  Capability  Port ID
ORA21C-N9K-B         Eth1/1          120        BR          Ethernet1/1
ORA21C-N9K-B         Eth1/2          120        BR          Ethernet1/2
ORA21C-N9K-B         Eth1/3          120        BR          Ethernet1/3
ORA21C-N9K-B         Eth1/4          120        BR          Ethernet1/4
ORA21C-FI-A          Eth1/9          120        BR          Eth1/27
ORA21C-FI-A          Eth1/10         120        BR          Eth1/28
ORA21C-FI-B          Eth1/11         120        BR          Eth1/27
ORA21C-FI-B          Eth1/12         120        BR          Eth1/28
Total entries displayed: 8
```

Cisco Nexus B Connectivity

```
ORA21C-N9K-B# show lldp neighbors
Capability codes:
  (R) Router, (B) Bridge, (T) Telephone, (C) DOCSIS Cable Device
  (W) WLAN Access Point, (P) Repeater, (S) Station, (O) Other
Device ID            Local Intf      Hold-time  Capability  Port ID
ORA21C-N9K-A         Eth1/1          120        BR          Ethernet1/1
ORA21C-N9K-A         Eth1/2          120        BR          Ethernet1/2
ORA21C-N9K-A         Eth1/3          120        BR          Ethernet1/3
ORA21C-N9K-A         Eth1/4          120        BR          Ethernet1/4
ORA21C-FI-A          Eth1/9          120        BR          Eth1/29
ORA21C-FI-A          Eth1/10         120        BR          Eth1/30
ORA21C-FI-B          Eth1/11         120        BR          Eth1/29
ORA21C-FI-B          Eth1/12         120        BR          Eth1/30
Total entries displayed: 8
```

Procedure 1. Configure the port channels on the Cisco Nexus Switches

Step 1. Login as admin user into Cisco Nexus Switch A and run the following commands:

```
configure terminal
interface port-channel51
  description connect to ORA21C-FI-A
  switchport mode trunk
  switchport trunk allowed vlan 1,10,21-24,134
  spanning-tree port type edge trunk
  mtu 9216
  vpc 51
  no shutdown

interface port-channel52
  description connect to ORA21C-FI-B
  switchport mode trunk
  switchport trunk allowed vlan 1,10,21-24,134
  spanning-tree port type edge trunk
  mtu 9216
  vpc 52
  no shutdown

interface Ethernet1/9
  description Fabric-Interconnect-A-27
  switchport mode trunk
  switchport trunk allowed vlan 1,10,21-24,134
```

```
  spanning-tree port type edge trunk
  mtu 9216
  channel-group 51 mode active
  no shutdown

interface Ethernet1/10
  description Fabric-Interconnect-A-28
  switchport mode trunk
  switchport trunk allowed vlan 1,10,21-24,134
  spanning-tree port type edge trunk
  mtu 9216
  channel-group 51 mode active
  no shutdown

interface Ethernet1/11
  description Fabric-Interconnect-B-27
  switchport mode trunk
  switchport trunk allowed vlan 1,10,21-24,134
  spanning-tree port type edge trunk
  mtu 9216
  channel-group 52 mode active
  no shutdown

interface Ethernet1/12
  description Fabric-Interconnect-B-28
  switchport mode trunk
  switchport trunk allowed vlan 1,10,21-24,134
  spanning-tree port type edge trunk
  mtu 9216
  channel-group 52 mode active
  no shutdown

copy run start
```

Step 2. Login as admin user into Cisco Nexus Switch B and run the following commands to configure the second Cisco Nexus Switch:

```
configure terminal

interface port-channel51
  description connect to ORA21C-FI-A
```

```
  switchport mode trunk
  switchport trunk allowed vlan 1,10,21-24,134
  spanning-tree port type edge trunk
  mtu 9216
  vpc 51
  no shutdown

interface port-channel52
  description connect to ORA21C-FI-B
  switchport mode trunk
  switchport trunk allowed vlan 1,10,21-24,134
  spanning-tree port type edge trunk
  mtu 9216
  vpc 52
  no shutdown

interface Ethernet1/9
  description Fabric-Interconnect-A-29
  switchport mode trunk
  switchport trunk allowed vlan 1,10,21-24,134
  spanning-tree port type edge trunk
  mtu 9216
  channel-group 51 mode active
  no shutdown

interface Ethernet1/10
  description Fabric-Interconnect-A-30
  switchport mode trunk
  switchport trunk allowed vlan 1,10,21-24,134
  spanning-tree port type edge trunk
  mtu 9216
  channel-group 51 mode active
  no shutdown

interface Ethernet1/11
  description Fabric-Interconnect-B-29
  switchport mode trunk
  switchport trunk allowed vlan 1,10,21-24,134
  spanning-tree port type edge trunk
```

```
  mtu 9216
  channel-group 52 mode active
  no shutdown

interface Ethernet1/12
  description Fabric-Interconnect-B-30
  switchport mode trunk
  switchport trunk allowed vlan 1,10,21-24,134
  spanning-tree port type edge trunk
  mtu 9216
  channel-group 52 mode active
  no shutdown

copy run start
```

Create vPC Configuration between Cisco Nexus and NetApp Storage Array

This section describes how to create and configure port channel 13 and 14 for network traffic between the Cisco Nexus Switches and NetApp Storage Controllers.

Table 9 lists the vPC IDs, allowed VLAN IDs, and ethernet uplink ports.

Table 9. vPC IDs and VLAN IDs

vPC Description	vPC ID	Cisco Nexus Switch Ports	NetApp Storage Ports	Allowed VLANs
Storage Port Channel 13	13	N9K-A Port 1/17	FlexPod-A800-CT1:e5a	21,22,23,24
		N9K-B Port 1/17	FlexPod-A800-CT1:e5b	
Storage Port Channel 14	14	N9K-APort 1/18	FlexPod-A800-CT2:e5a	21,22,23,24
		N9K-B Port 1/18	FlexPod-A800-CT2:e5b	

Procedure 1. Configure the port channels on Cisco Nexus Switches

Step 1. Login as admin user into the Cisco Nexus Switch A and run the following commands:

```
configure terminal
interface port-channel13
  description PC-NetApp-A
  switchport mode trunk
  switchport trunk allowed vlan 21-24
  spanning-tree port type edge trunk
  mtu 9216
```

```
  vpc 13
  no shutdown

interface port-channel14
  description PC-NetApp-B
  switchport mode trunk
  switchport trunk allowed vlan 21-24
  spanning-tree port type edge trunk
  mtu 9216
  vpc 14
  no shutdown

interface Ethernet1/17
  description FlexPod-A800-CT1:e5a
  switchport mode trunk
  switchport trunk allowed vlan 21-24
  mtu 9216
  channel-group 13 mode active
  no shutdown

interface Ethernet1/18
  description FlexPod-A800-CT2:e5a
  switchport mode trunk
  switchport trunk allowed vlan 21-24
  mtu 9216
  channel-group 14 mode active
  no shutdown

copy run start
```

Step 2. Login as admin user into the Cisco Nexus Switch B and run the following commands to configure the second Cisco Nexus Switch:

```
configure terminal

interface port-channel13
  description PC-NetApp-A
  switchport mode trunk
  switchport trunk allowed vlan 21-24
  spanning-tree port type edge trunk
  mtu 9216
```

```
  vpc 13
  no shutdown

interface port-channel14
  description PC-NetApp-B
  switchport mode trunk
  switchport trunk allowed vlan 21-24
  spanning-tree port type edge trunk
  mtu 9216
  vpc 14
  no shutdown

interface Ethernet1/17
  description FlexPod-A800-CT1:e5b
  switchport mode trunk
  switchport trunk allowed vlan 21-24
  mtu 9216
  channel-group 13 mode active
  no shutdown

interface Ethernet1/18
  description FlexPod-A800-CT2:e5b
  switchport mode trunk
  switchport trunk allowed vlan 21-24
  mtu 9216
  channel-group 14 mode active
  no shutdown

copy run start
```

Verify All vPC Status

Procedure 1. Verify the status of all port-channels using Cisco Nexus Switches

Step 1. Cisco Nexus Switch A Port-Channel Summary:

```
ORA21C-N9K-A# show port-channel summary
Flags:  D - Down        P - Up in port-channel (members)
        I - Individual  H - Hot-standby (LACP only)
        s - Suspended   r - Module-removed
        b - BFD Session Wait
        S - Switched    R - Routed
        U - Up (port-channel)
        p - Up in delay-lacp mode (member)
        M - Not in use. Min-links not met
--------------------------------------------------------------------------------
Group Port-       Type     Protocol  Member Ports
      Channel
--------------------------------------------------------------------------------
1     Po1(SU)     Eth      LACP      Eth1/1(P)    Eth1/2(P)    Eth1/3(P)
                                     Eth1/4(P)
13    Po13(SU)    Eth      LACP      Eth1/17(P)
14    Po14(SU)    Eth      LACP      Eth1/18(P)
51    Po51(SU)    Eth      LACP      Eth1/9(P)    Eth1/10(P)
52    Po52(SU)    Eth      LACP      Eth1/11(P)   Eth1/12(P)
```

Step 2. Cisco Nexus Switch B Port-Channel Summary:

```
ORA21C-N9K-B# show port-channel summary
Flags:  D - Down        P - Up in port-channel (members)
        I - Individual  H - Hot-standby (LACP only)
        s - Suspended   r - Module-removed
        b - BFD Session Wait
        S - Switched    R - Routed
        U - Up (port-channel)
        p - Up in delay-lacp mode (member)
        M - Not in use. Min-links not met
--------------------------------------------------------------------------------
Group Port-       Type     Protocol  Member Ports
      Channel
--------------------------------------------------------------------------------
1     Po1(SU)     Eth      LACP      Eth1/1(P)    Eth1/2(P)    Eth1/3(P)
                                     Eth1/4(P)
13    Po13(SU)    Eth      LACP      Eth1/17(P)
14    Po14(SU)    Eth      LACP      Eth1/18(P)
51    Po51(SU)    Eth      LACP      Eth1/9(P)    Eth1/10(P)
52    Po52(SU)    Eth      LACP      Eth1/11(P)   Eth1/12(P)
```

Step 3. Cisco Nexus Switch A vPC Status:

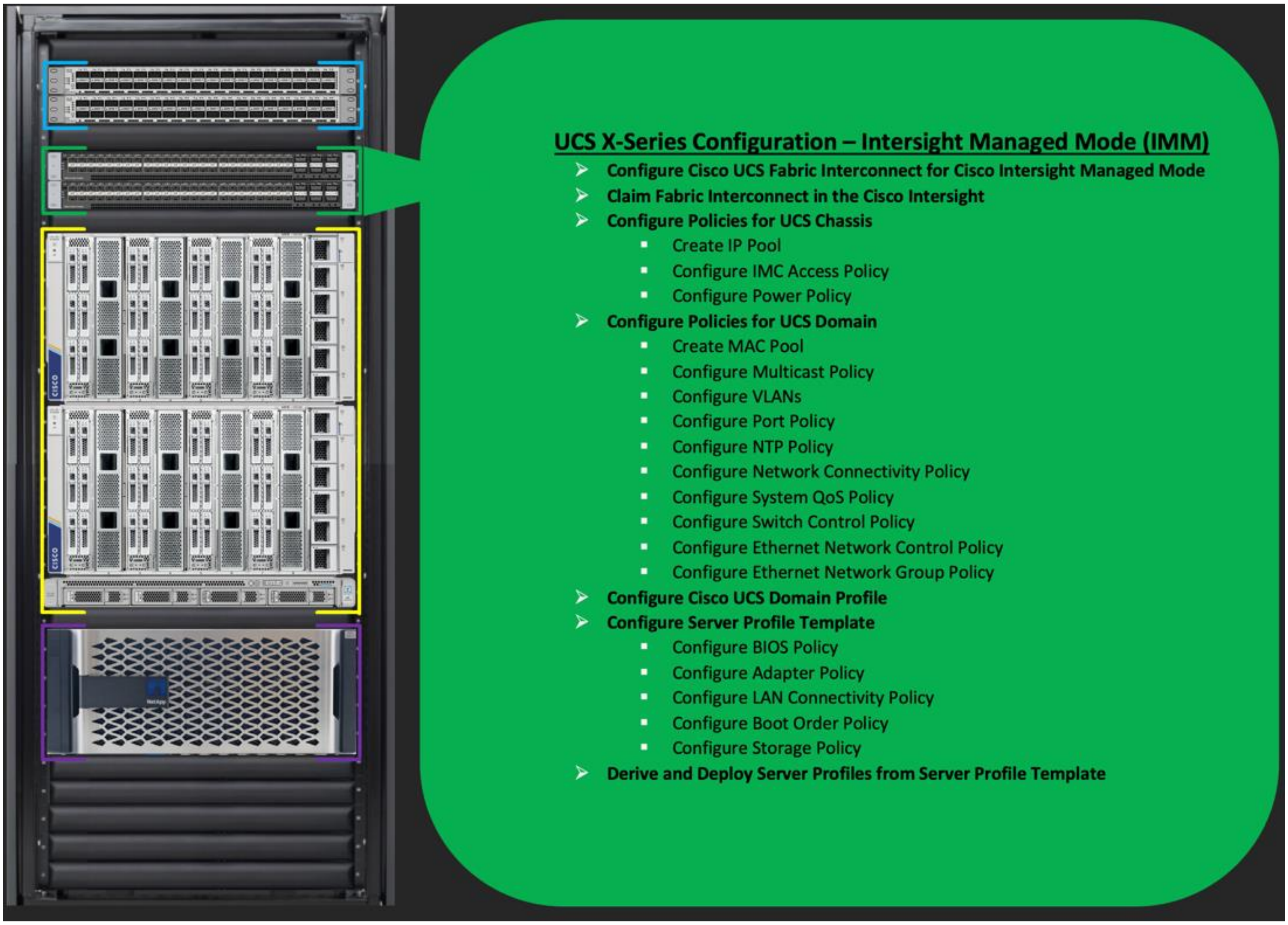
```
ORA21C-N9K-A# show vpc brief
Legend:
                (*) - local vPC is down, forwarding via vPC peer-link

vPC domain id                     : 1
Peer status                       : peer adjacency formed ok
vPC keep-alive status             : peer is alive
Configuration consistency status  : success
Per-vlan consistency status       : success
Type-2 consistency status         : success
vPC role                          : secondary
Number of vPCs configured         : 4
Peer Gateway                      : Disabled
Dual-active excluded VLANs        : -
Graceful Consistency Check        : Enabled
Auto-recovery status              : Disabled
Delay-restore status              : Timer is off.(timeout = 30s)
Delay-restore SVI status          : Timer is off.(timeout = 10s)
Operational Layer3 Peer-router    : Disabled
Virtual-peerlink mode             : Disabled

vPC Peer-link status
---------------------------------------------------------------------
id    Port   Status Active vlans
--    ----   ------ -------------------------------------------------
1     Po1    up     1,10,21-24,134

vPC status
----------------------------------------------------------------------------
Id    Port          Status Consistency Reason                Active vlans
--    ------------  ------ ----------- ------                ---------------
13    Po13          up     success     success               21-24

14    Po14          up     success     success               21-24

51    Po51          up     success     success               1,10,21-24,134

52    Po52          up     success     success               1,10,21-24,134


Please check "show vpc consistency-parameters vpc <vpc-num>" for the
consistency reason of down vpc and for type-2 consistency reasons for
any vpc.
```

Step 4. Cisco Nexus Switch B vPC Status:

```
ORA21C-N9K-B# show vpc brief
Legend:
                (*) - local vPC is down, forwarding via vPC peer-link

vPC domain id                     : 1
Peer status                       : peer adjacency formed ok
vPC keep-alive status             : peer is alive
Configuration consistency status  : success
Per-vlan consistency status       : success
Type-2 consistency status         : success
vPC role                          : primary
Number of vPCs configured         : 4
Peer Gateway                      : Disabled
Dual-active excluded VLANs        : -
Graceful Consistency Check        : Enabled
Auto-recovery status              : Disabled
Delay-restore status              : Timer is off.(timeout = 30s)
Delay-restore SVI status          : Timer is off.(timeout = 10s)
Operational Layer3 Peer-router    : Disabled
Virtual-peerlink mode             : Disabled

vPC Peer-link status
---------------------------------------------------------------------
id    Port   Status Active vlans
--    ----   ------ -------------------------------------------------
1     Po1    up     1,10,21-24,134

vPC status
----------------------------------------------------------------------------
Id    Port          Status Consistency Reason                Active vlans
--    ------------  ------ ----------- ------                ---------------
13    Po13          up     success     success               21-24

14    Po14          up     success     success               21-24

51    Po51          up     success     success               1,10,21-24,134

52    Po52          up     success     success               1,10,21-24,134


Please check "show vpc consistency-parameters vpc <vpc-num>" for the
consistency reason of down vpc and for type-2 consistency reasons for
any vpc.
```

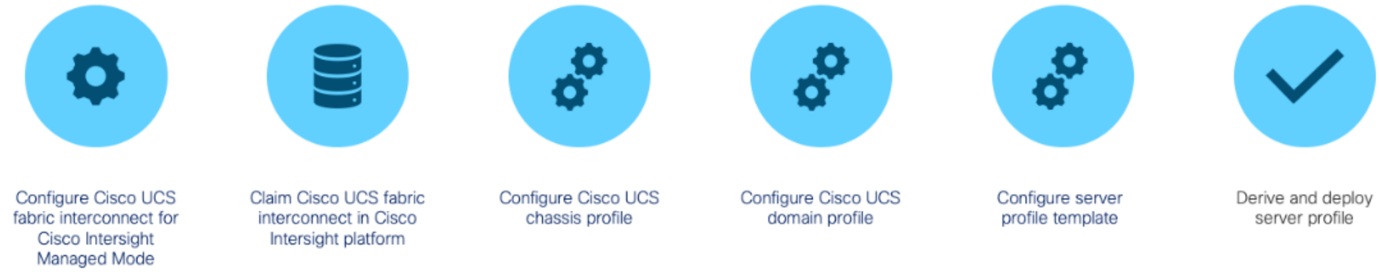
Cisco UCS X-Series Configuration – Intersight Managed Mode (IMM)

This section details the high-level steps for the Cisco UCS X-Series Configuration in Intersight Managed Mode.



Cisco Intersight Managed Mode standardizes policy and operation management for Cisco UCS X-Series. The compute nodes in Cisco UCS X-Series are configured using server profiles defined in Cisco Intersight. These server profiles derive all the server characteristics from various policies and templates. At a high level, configuring Cisco UCS using Intersight Managed Mode consists of the steps shown in Figure 3.

Figure 3. Configuration Steps for Cisco Intersight Managed Mode



Procedure 1. Configure Cisco UCS Fabric Interconnect for Cisco Intersight Managed Mode

During the initial configuration, for the management mode, the configuration wizard enables you to choose whether to manage the fabric interconnect through Cisco UCS Manager or the Cisco Intersight platform. You can switch the management mode for the fabric interconnects between Cisco Intersight and Cisco UCS Manager at any time; however, Cisco UCS FIs must be set up in Intersight Managed Mode (IMM) for configuring the Cisco UCS X-Series system.

Step 1. Verify the following physical connections on the fabric interconnect:

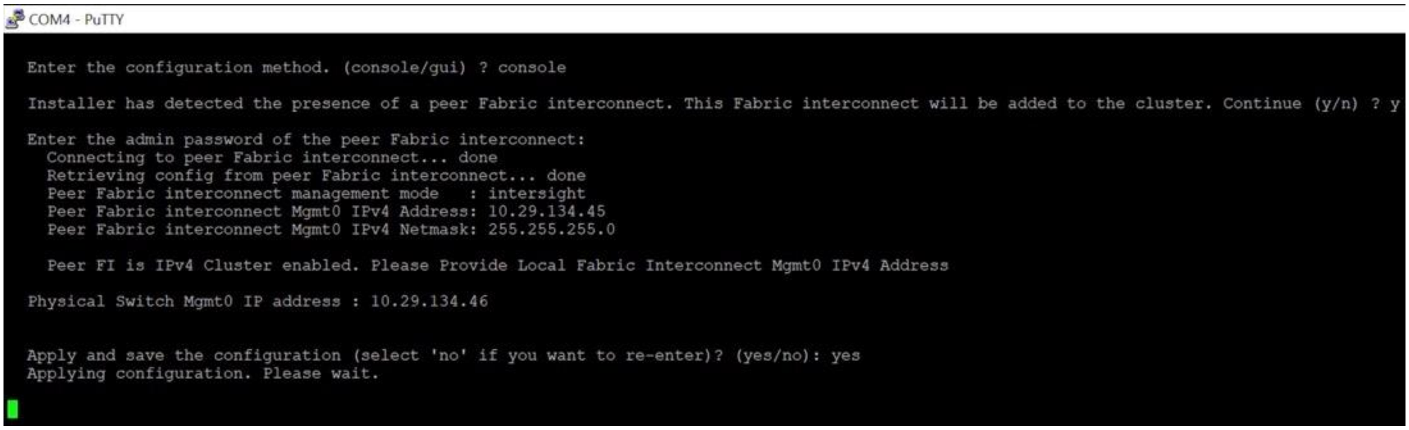
- The management Ethernet port (mgmt0) is connected to an external hub, switch, or router.
- The L1 ports on both fabric interconnects are directly connected to each other.
- The L2 ports on both fabric interconnects are directly connected to each other.

Step 2. Connect to the console port on the first fabric interconnect and configure the first FI as shown below:

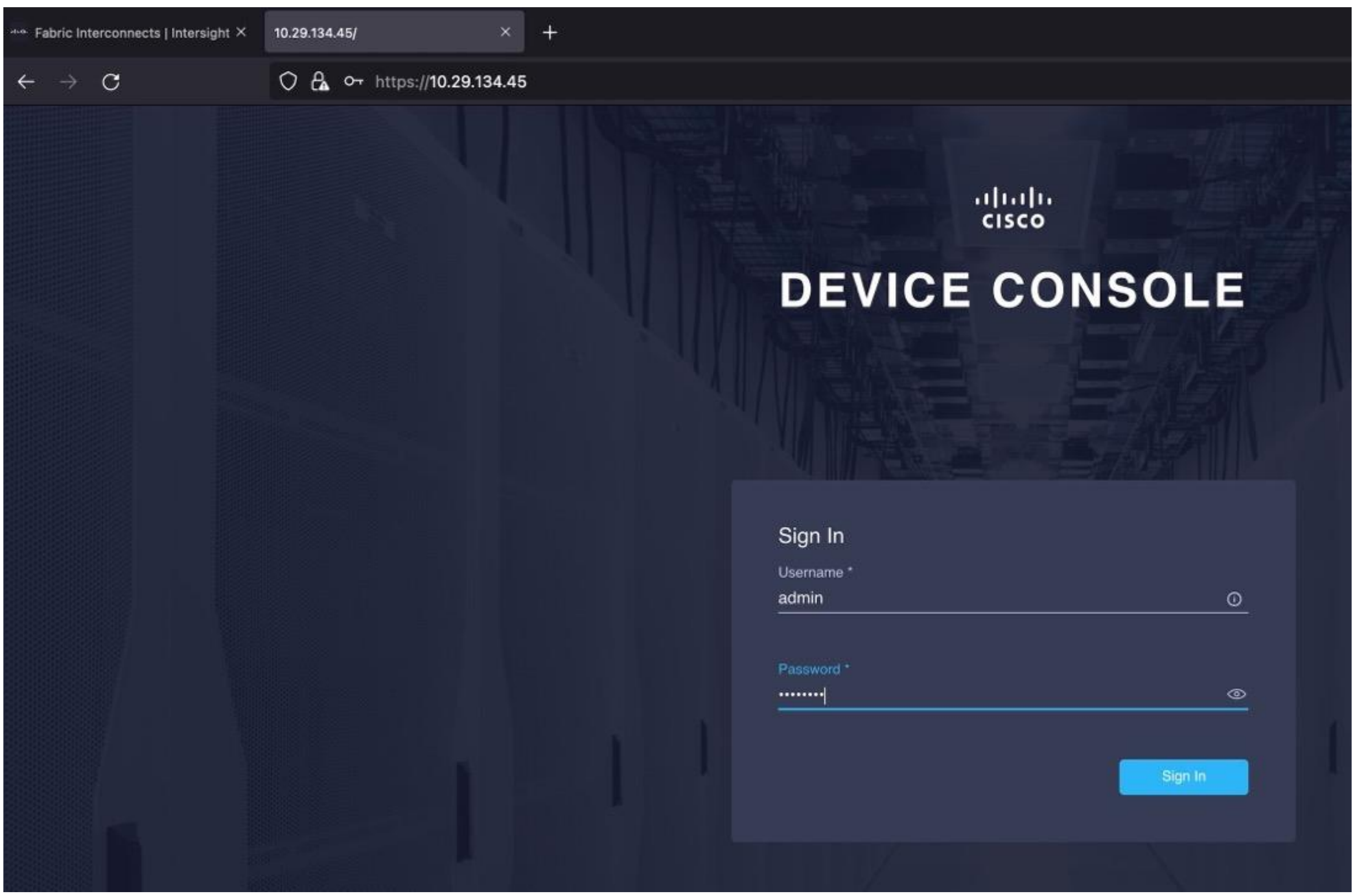
COM4 - PuTTY

```
  Enter the configuration method. (console/gui) ? console

  Enter the management mode. (ucsm/intersight)? intersight

  The Fabric interconnect will be configured in the intersight managed mode. Choose (y/n) to proceed: y

  Enforce strong password? (y/n) [y]: n

  Enter the password for "admin":
  Confirm the password for "admin":

  Enter the switch fabric (A/B) []: A

  Enter the system name:  ORA21C-FI

  Physical Switch Mgmt0 IP address : 10.29.134.45

  Physical Switch Mgmt0 IPv4 netmask : 255.255.255.0

  IPv4 address of the default gateway : 10.29.134.1

    DNS IP address : 171.70.168.183

  Configure the default domain name? (yes/no) [n]:

  Following configurations will be applied:

    Management Mode=intersight
    Switch Fabric=A
    System Name=ORA21C-FI
    Enforced Strong Password=no
    Physical Switch Mgmt0 IP Address=10.29.134.45
    Physical Switch Mgmt0 IP Netmask=255.255.255.0
    Default Gateway=10.29.134.1
    DNS Server=171.70.168.183

  Apply and save the configuration (select 'no' if you want to re-enter)? (yes/no): yes
  Applying configuration. Please wait.

  Configuration file - Ok
XML interface to system may become unavailable since ssh is disabled

 Completing basic configuration setup
```

Step 3. Connect the console port on the second fabric interconnect B and configure it as shown below:

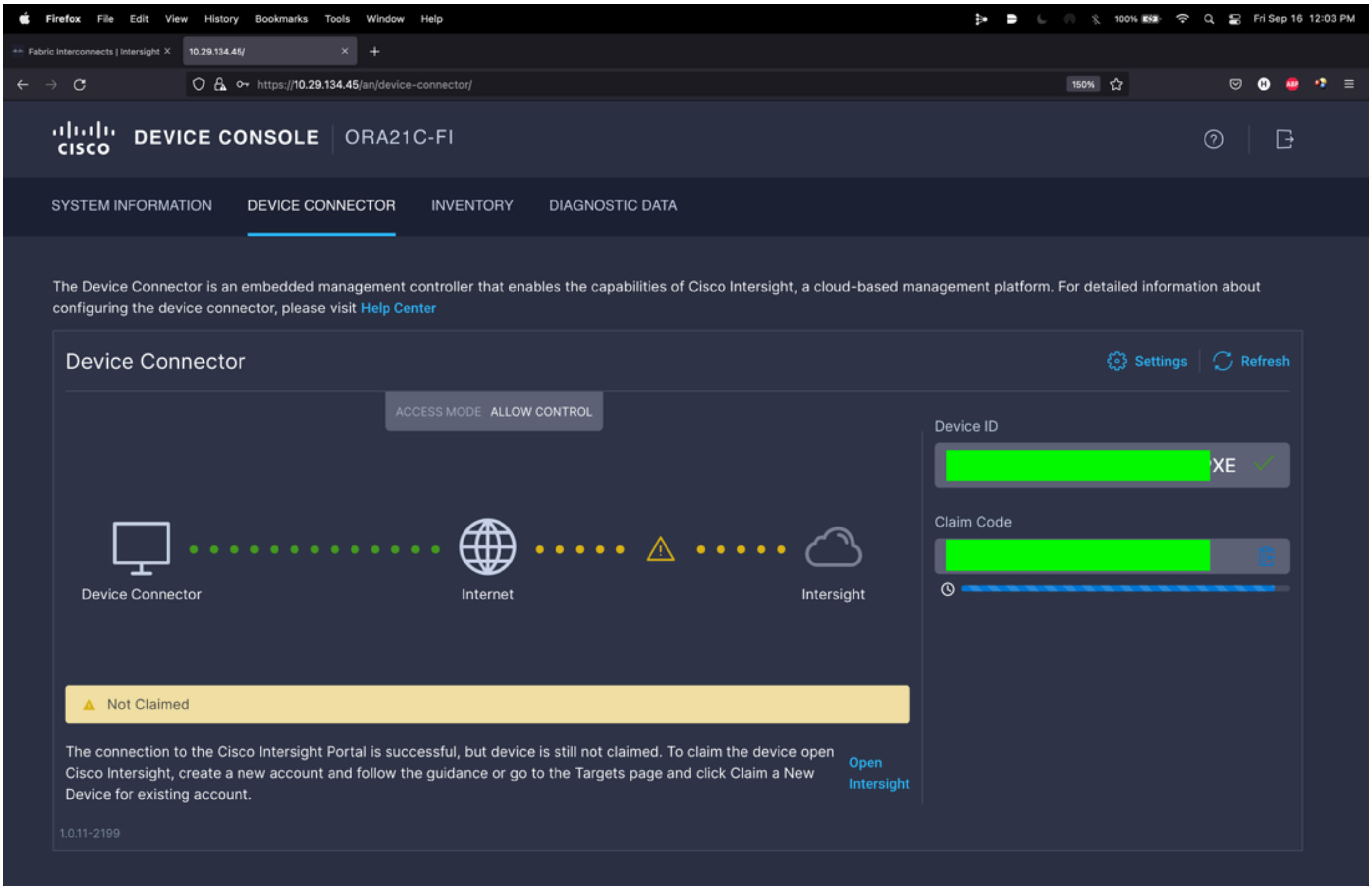


Step 4. After configuring both the FI management address, open a web browser and navigate to the Cisco UCS fabric interconnect management address as configured. If prompted to accept security certificates, accept, as necessary.



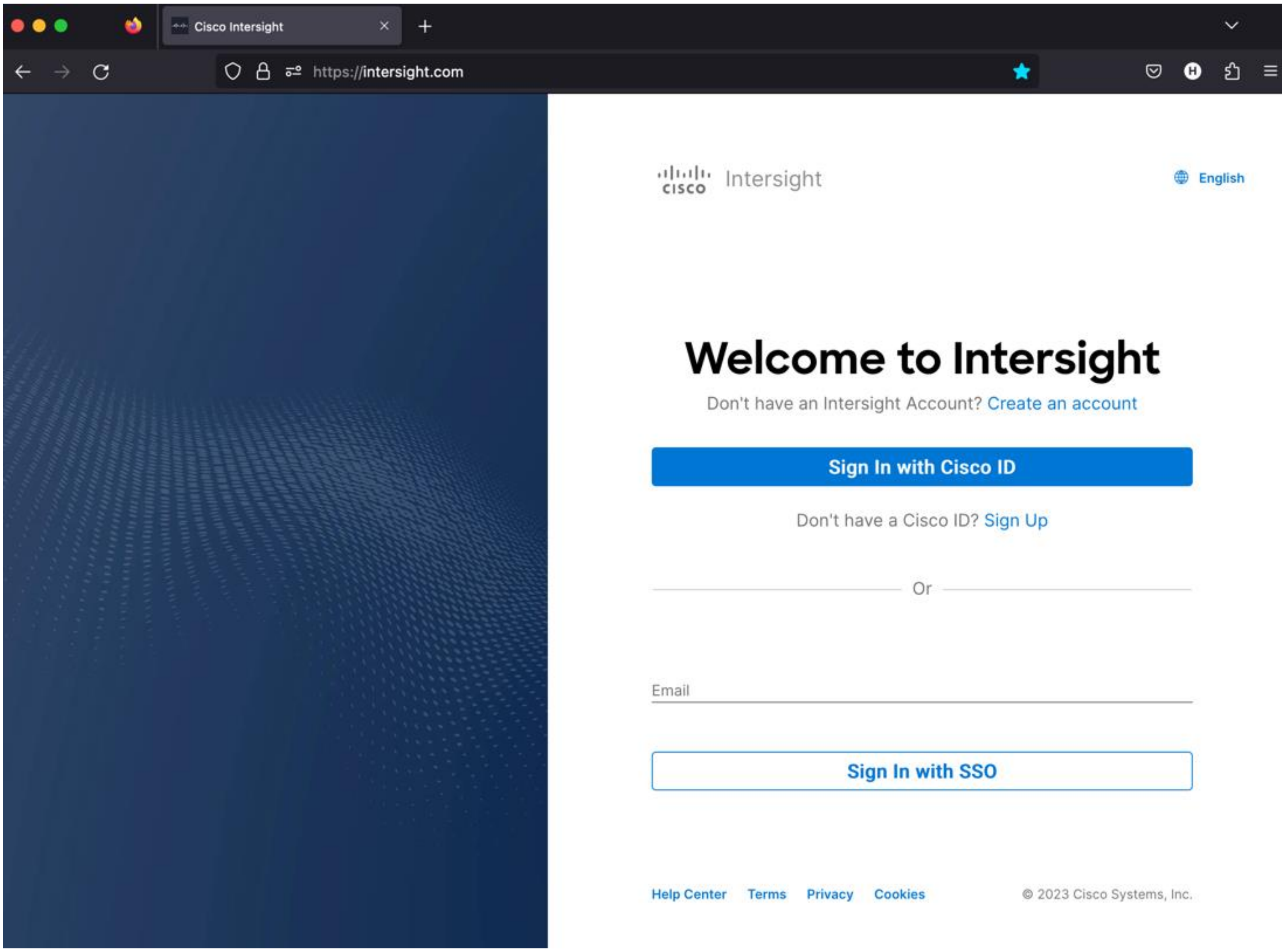
Step 5. Log into the device console for FI-A by entering your username and password.

Step 6. Go to the Device Connector tab and get the DEVICE ID and CLAIM Code as shown below:



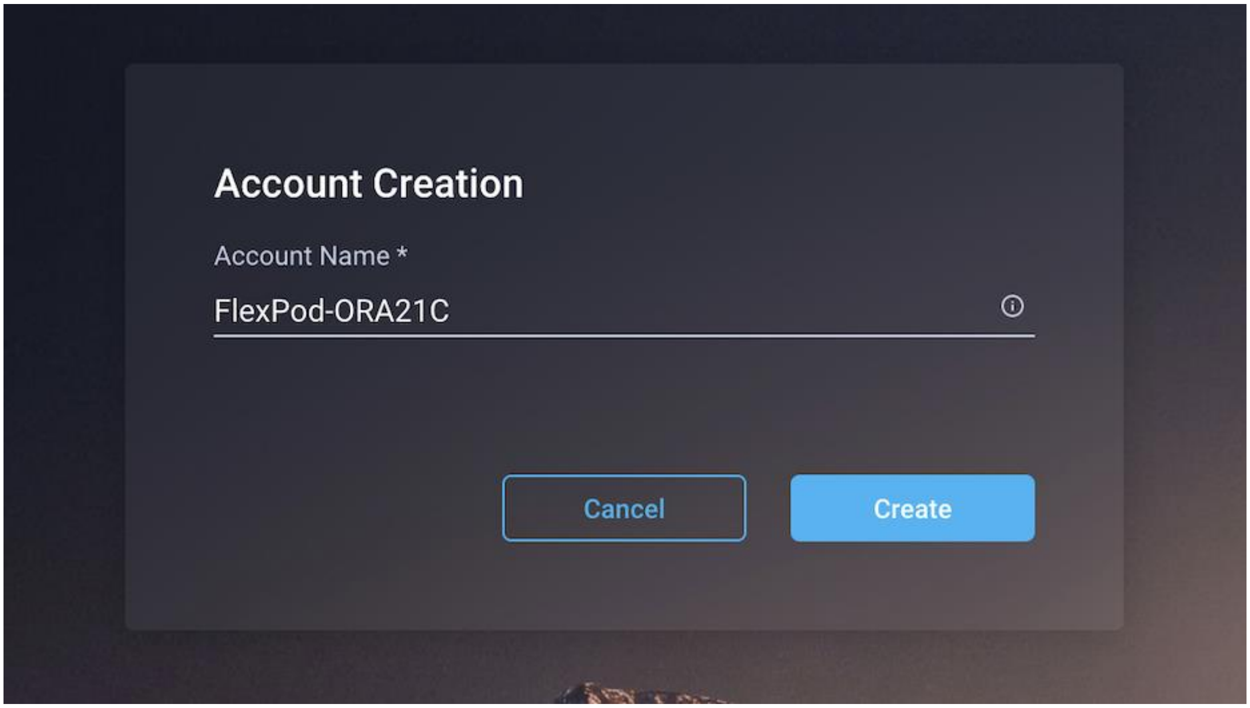
Procedure 2. Claim Fabric Interconnect in Cisco Intersight Platform

After setting up the Cisco UCS fabric interconnect for Cisco Intersight Managed Mode, FIs can be claimed to a new or an existing Cisco Intersight account. When a Cisco UCS fabric interconnect is successfully added to the Cisco Intersight platform, all future configuration steps are completed in the Cisco Intersight portal. After getting the device id and claim code of FI, go to [https://intersight.com/.](https://intersight.com/)

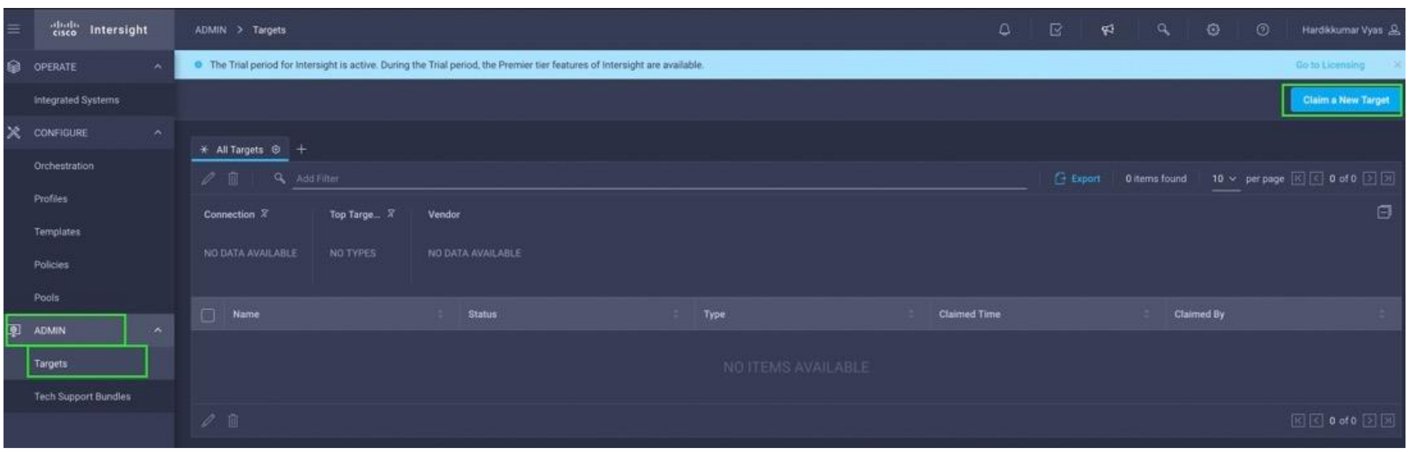


Step 1. Sign in with your Cisco ID or if you don't have one, click Sing Up and setup your account.

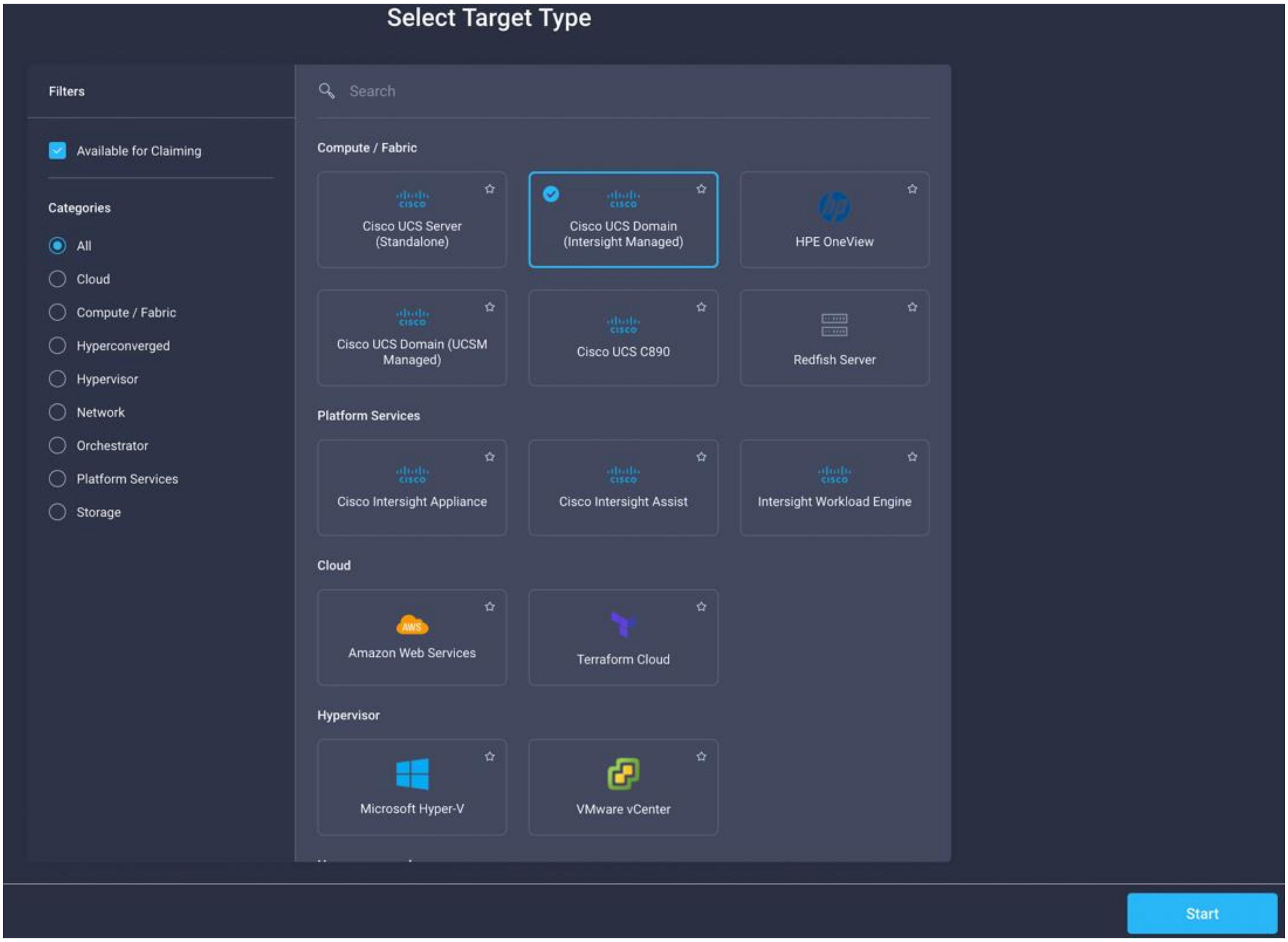
Note: We created the "FlexPod-ORA21C" account for this solution.



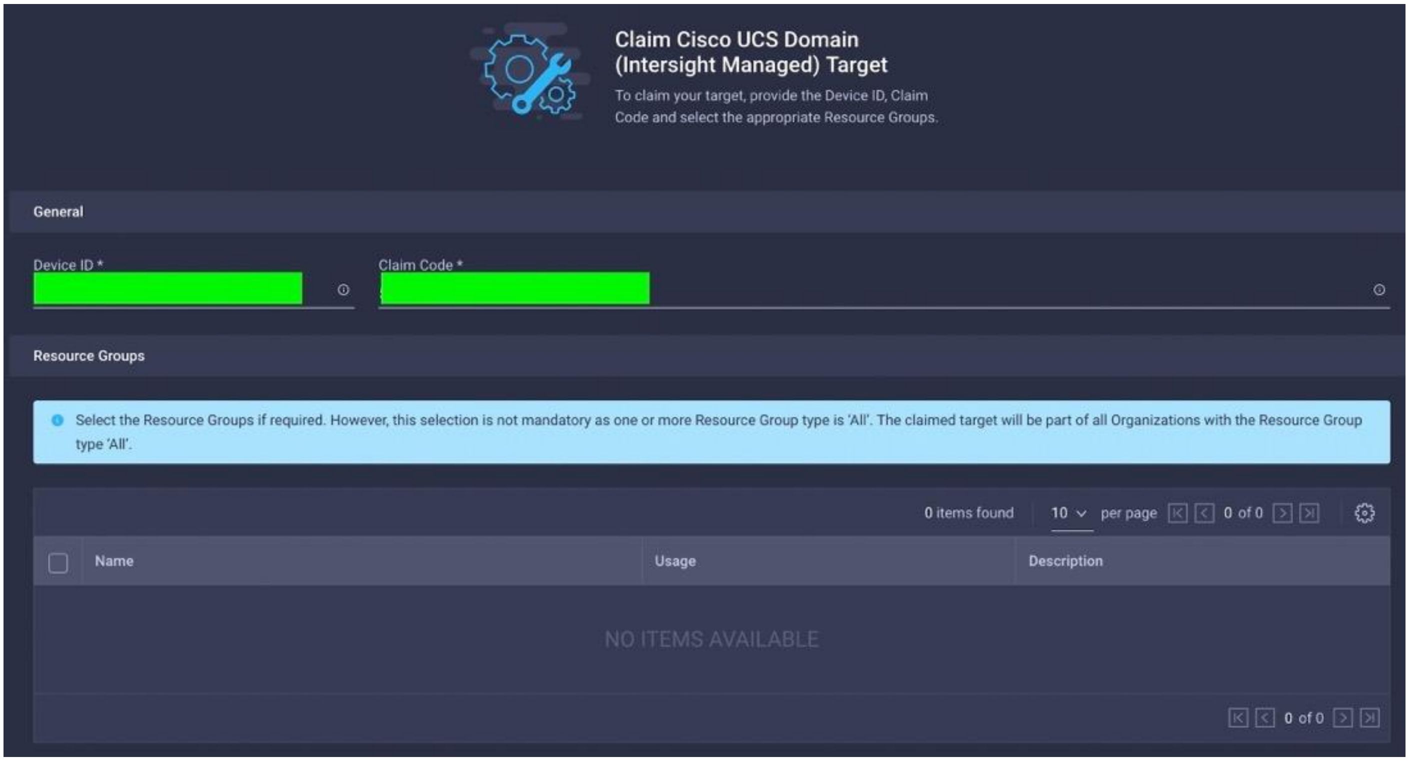
Step 2. After logging into your Cisco Intersight account, go to > ADMIN > Targets > Claim a New Target.



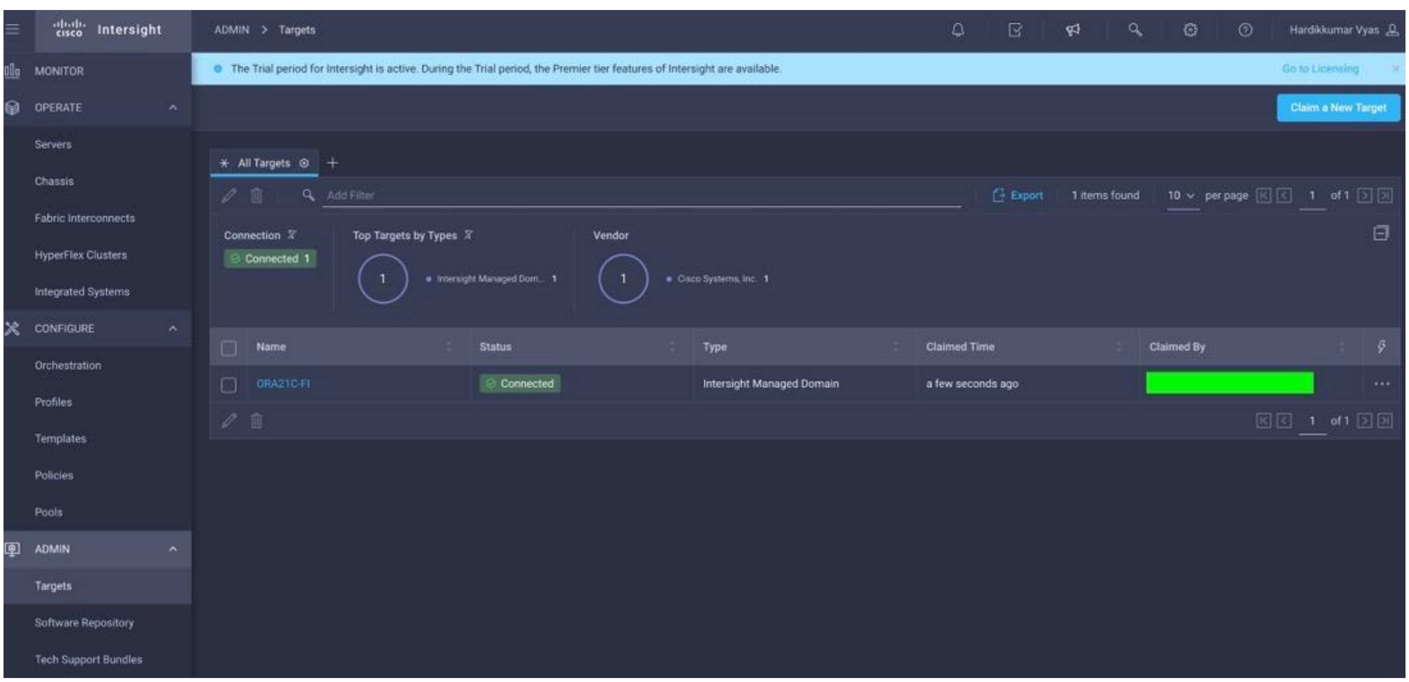
Step 3. For the Select Target Type, select "Cisco UCS Domain (Intersight Managed)" and click Start.

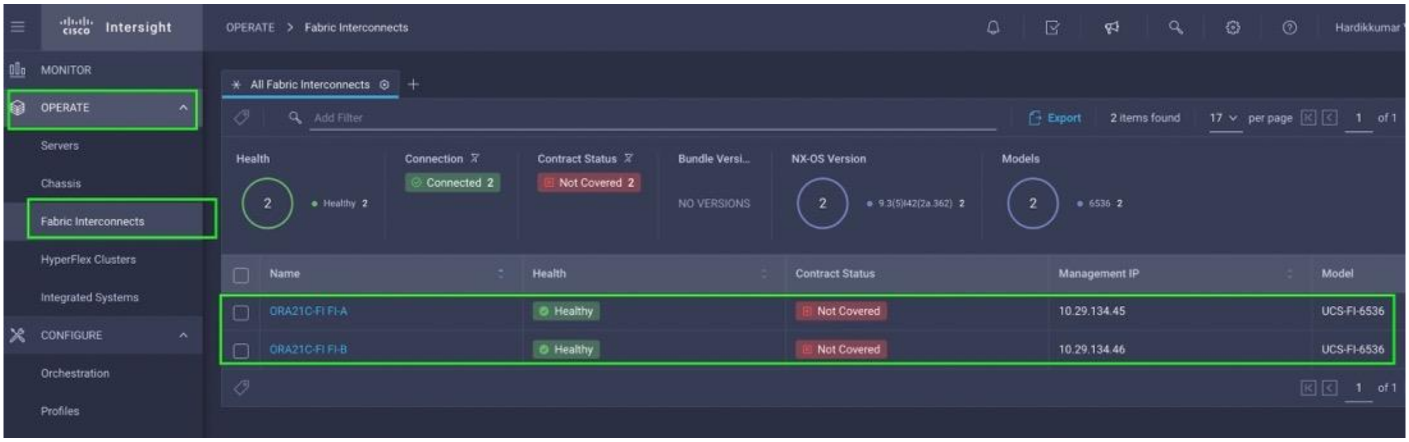


Step 4. Enter the Device ID and Claim Code which was previously captured. Click Claim to claim this domain in Cisco Intersight.



Step 5. When you claim this domain, you can see both FIs under this domain and verify it's under Intersight Managed Mode.





Procedure 3. Configure Policies for Cisco UCS Chassis

Note: For this solution, we configured Organization as "ORA21". We will configure all the profile, pools, and policies under this common organization to better consolidate resources.

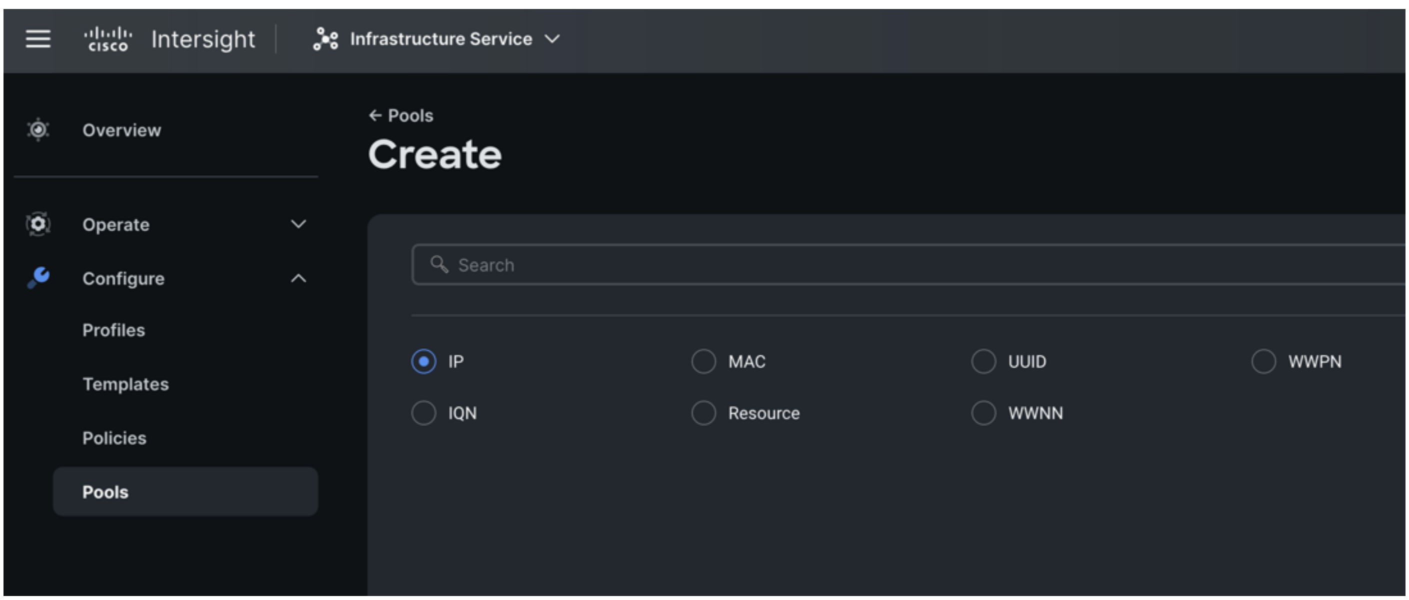
Step 1. To create Organization, go to Cisco Intersight > Settings > Organization and create depending upon your environment.

Note: We configured the IP Pool, IMC Access Policy, and Power Policy for the Cisco UCS Chassis profile as explained below.

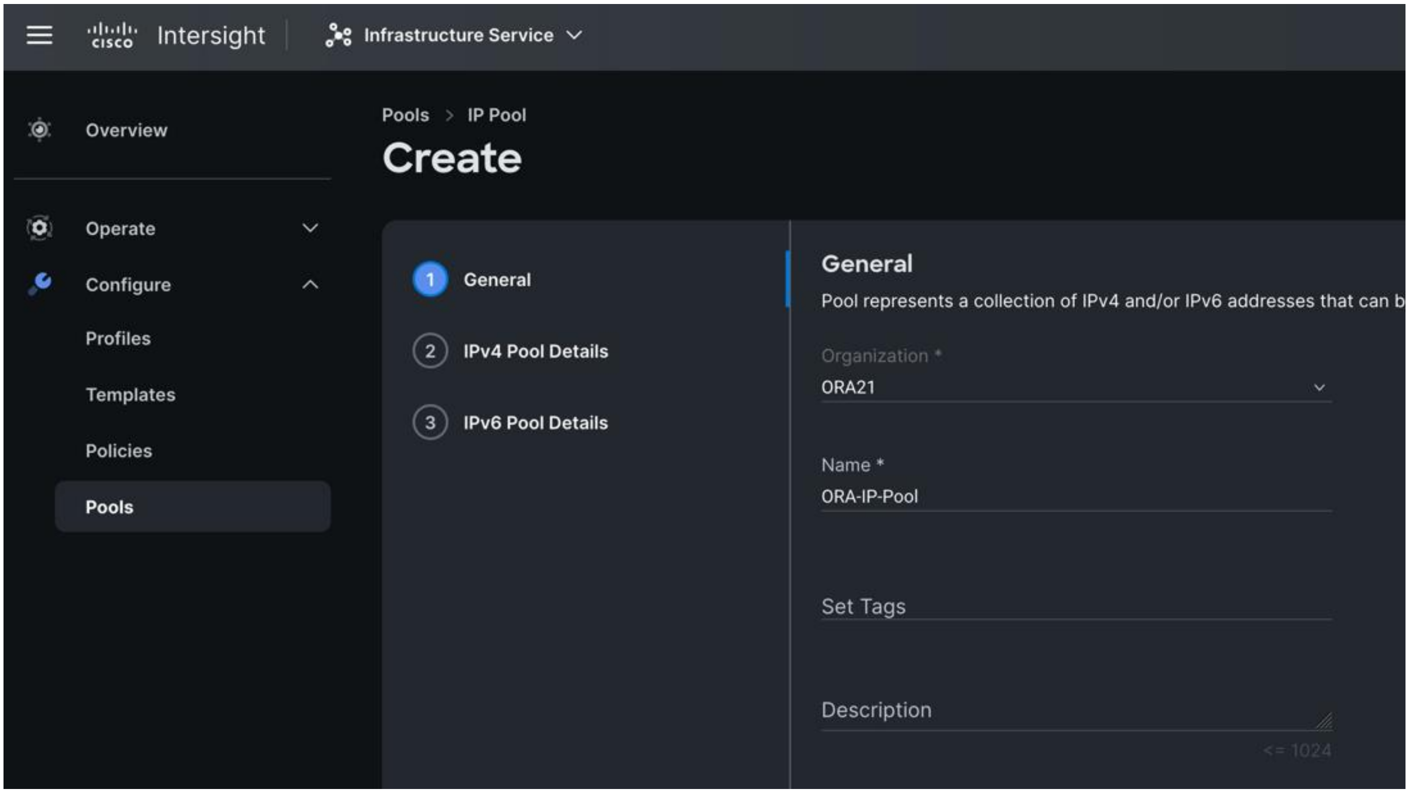
Procedure 4. Create IP Pool

Step 1. To configure the IP Pool for the Cisco UCS Chassis profile, go to > Infrastructure Service > Configure > Pools > and then select "Create Pool" on the top right corner.

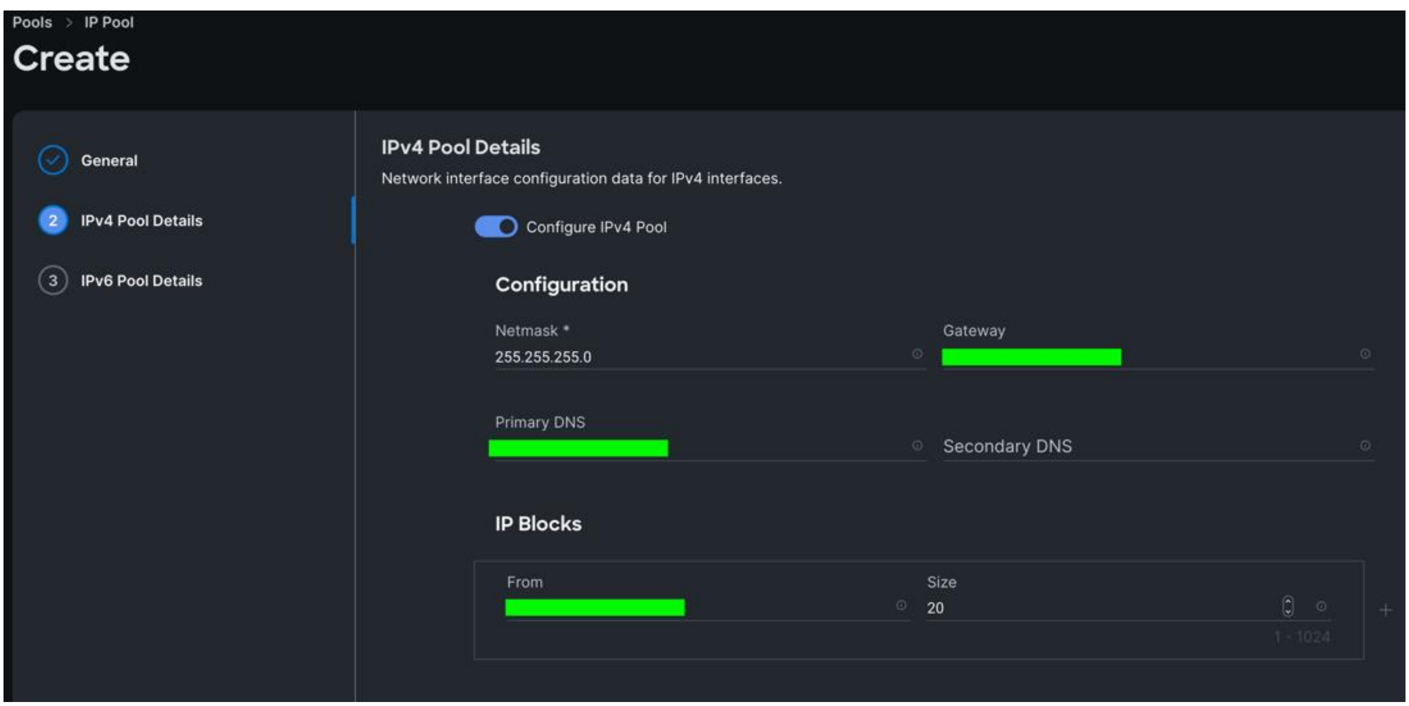
Step 2. Select option "IP" as shown below to create the IP Pool.



Step 3. In the IP Pool Create section, for Organization select "ORA21" and enter the Policy name "ORA-IP-Pool" and click Next.



Step 4. Enter Netmask, Gateway, Primary DNS, IP Blocks and Size according to your environment and click Next.

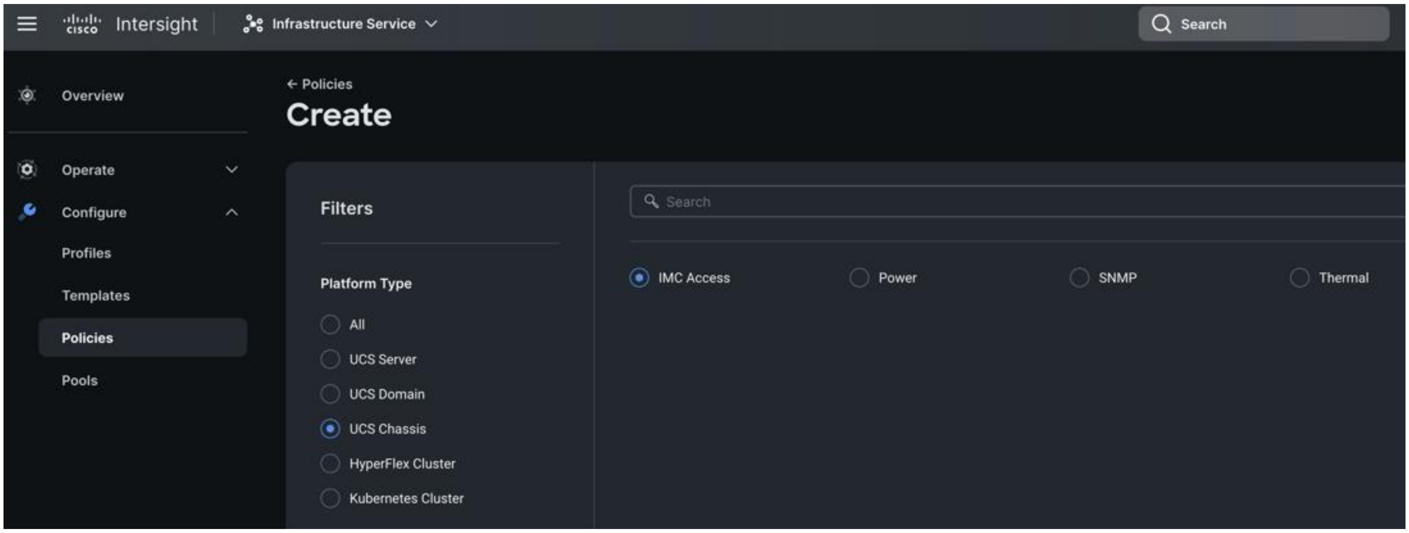


Note: For this solution, we did not configure the IPv6 Pool. Keep the Configure IPv6 Pool option off and click Create to create the IP Pool.

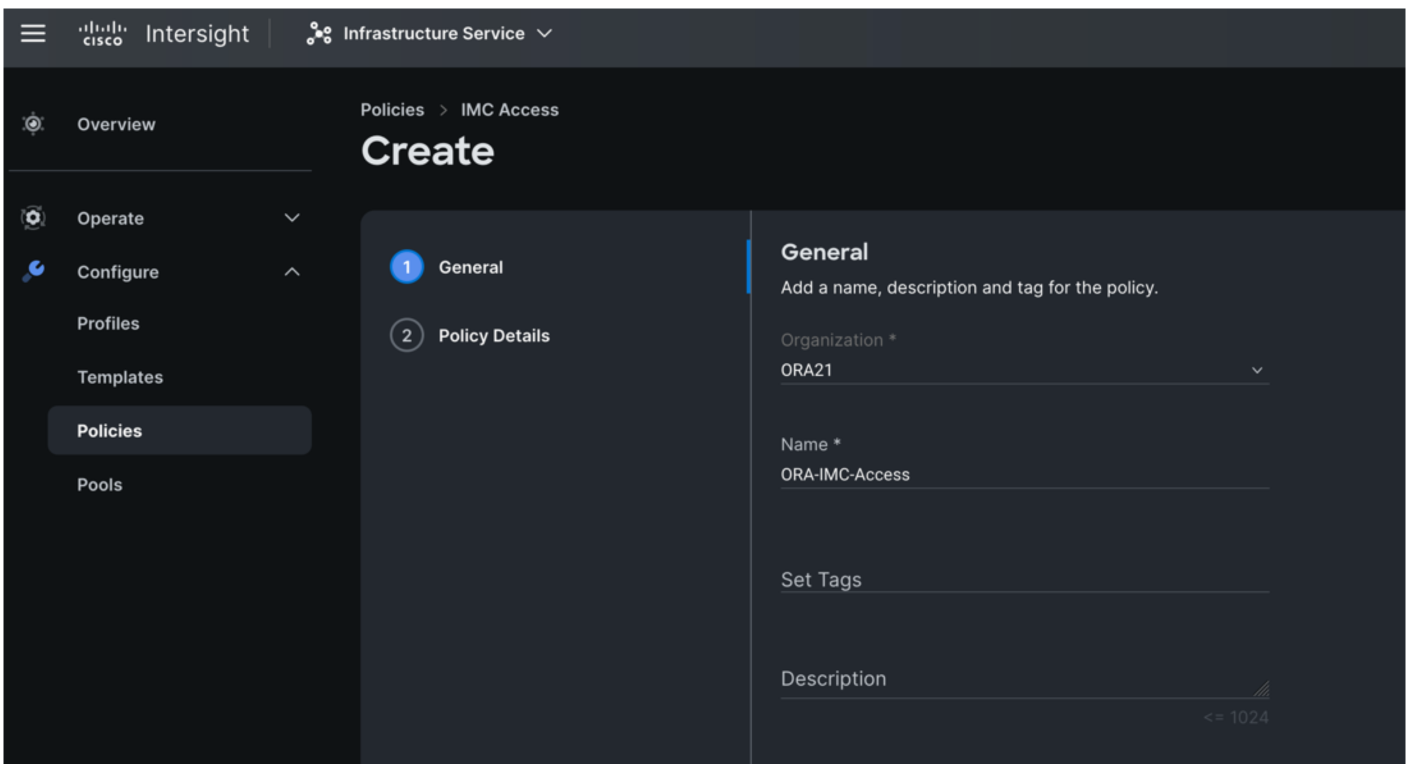
Procedure 5. Configure IMC Access Policy

Step 1. To configure the IMC Access Policy for the Cisco UCS Chassis profile, go to > Infrastructure Service > Configure > Polices > and click Create Policy.

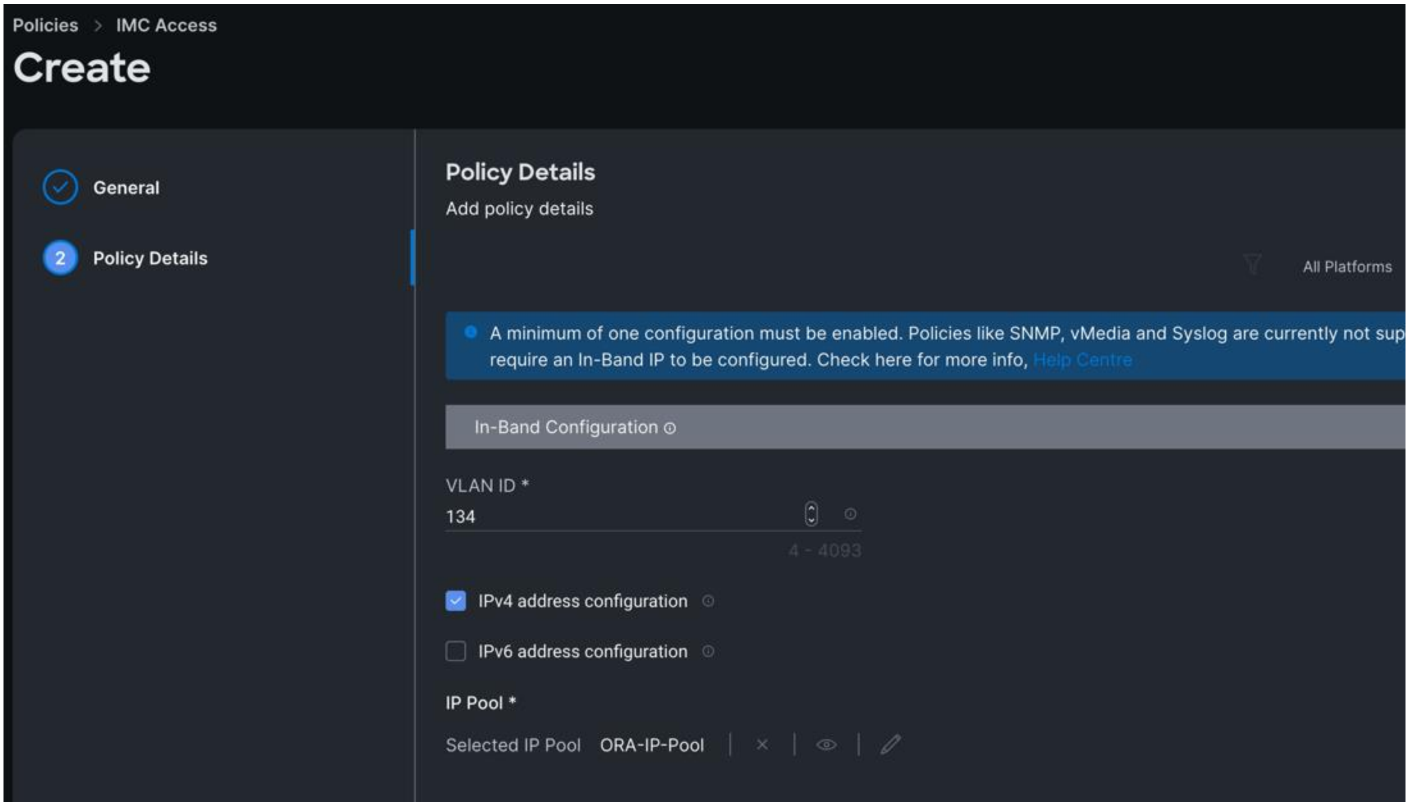
Step 2. Select the platform type "UCS Chassis" and select "IMC Access" policy.



Step 3. In the IMC Access Create section, for Organization select "ORA21" and enter the Policy name "ORA-IMC-Access" and click Next.



Step 4. In the Policy Details section, enter the VLAN ID as 134 and select the IP Pool "ORA-IP-Pool."

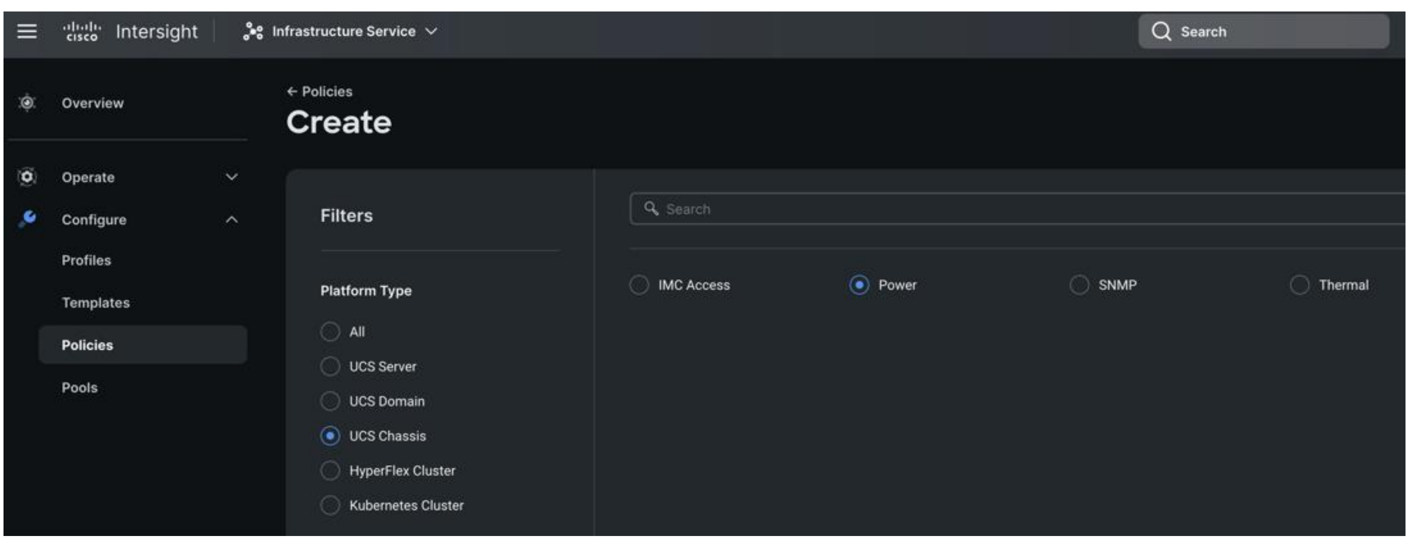


Step 5. Click Create to create this policy.

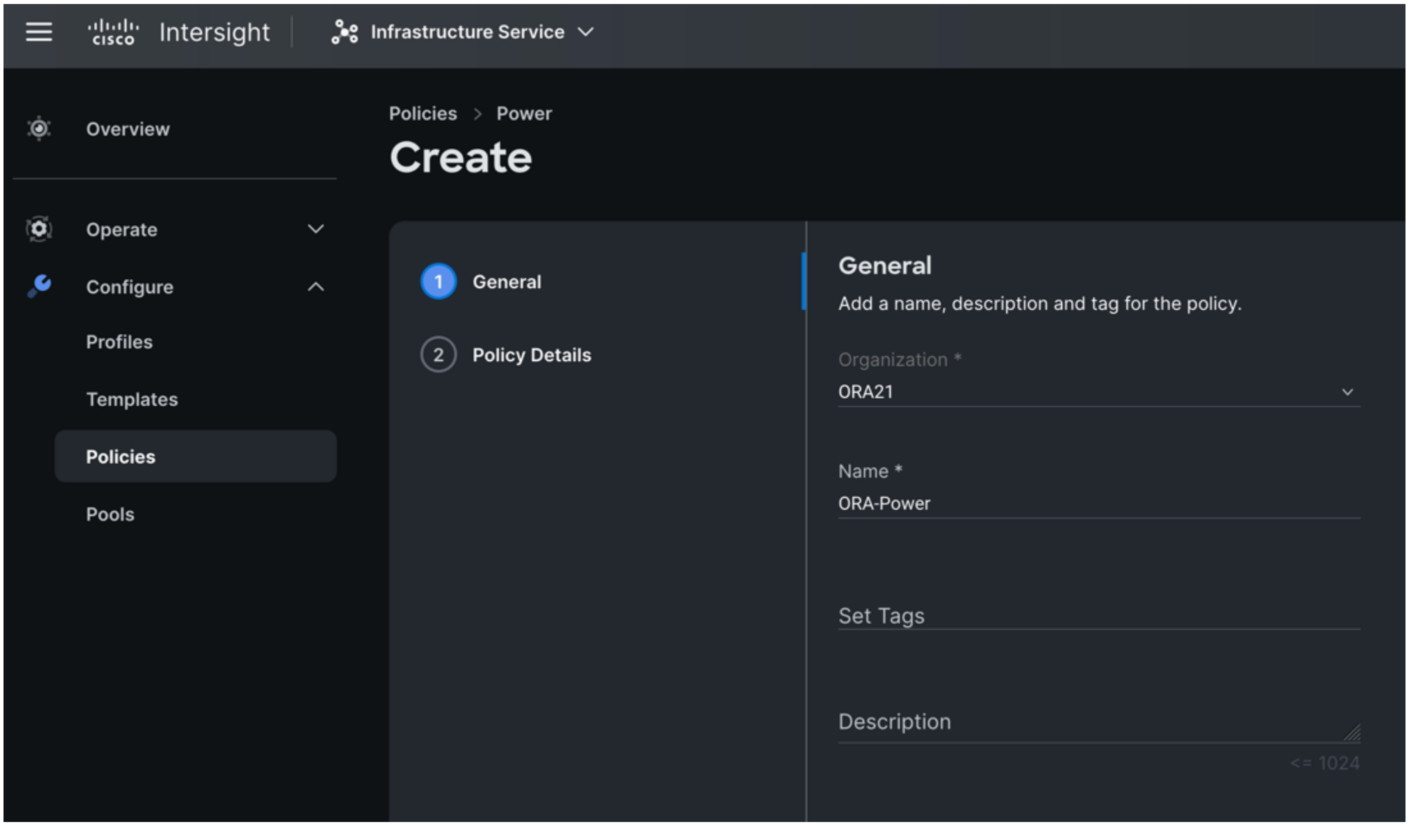
Procedure 6. Configure Power Policy

Step 1. To configure the Power Policy for the Cisco UCS Chassis profile, go to > Infrastructure Service > Configure > Polices > and click Create Policy.

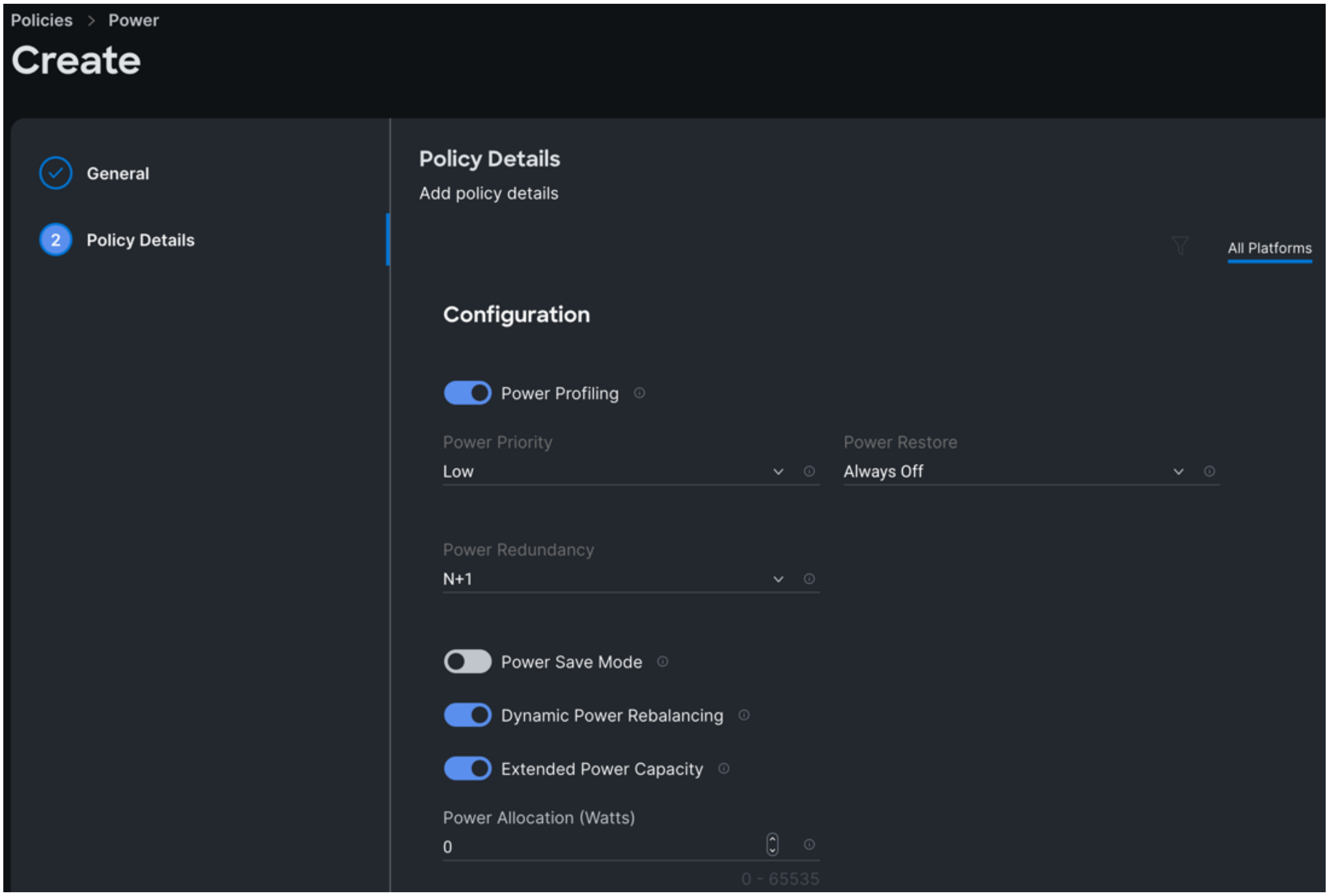
Step 2. Select the platform type "UCS Chassis" and select "Power."



Step 3. In the Power Policy Create section, for Organization select "ORA21" and enter the Policy name "ORA-Power" and click Next.



Step 4. In the Policy Details section, for Power Redundancy select N+1 and turn off Power Save Mode.



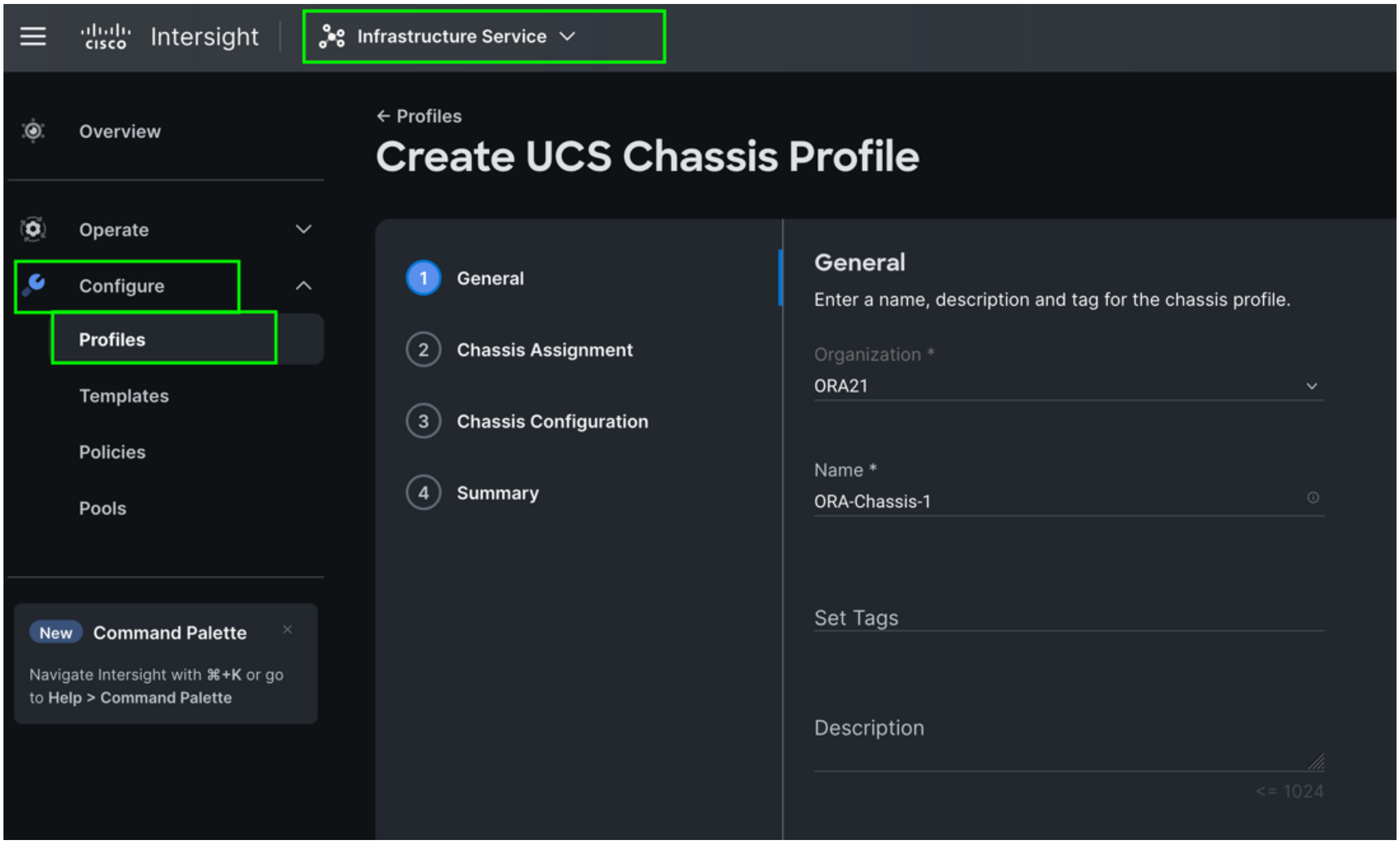
Step 5. Click Create to create this policy.

Procedure 7. Create Cisco UCS Chassis Profile

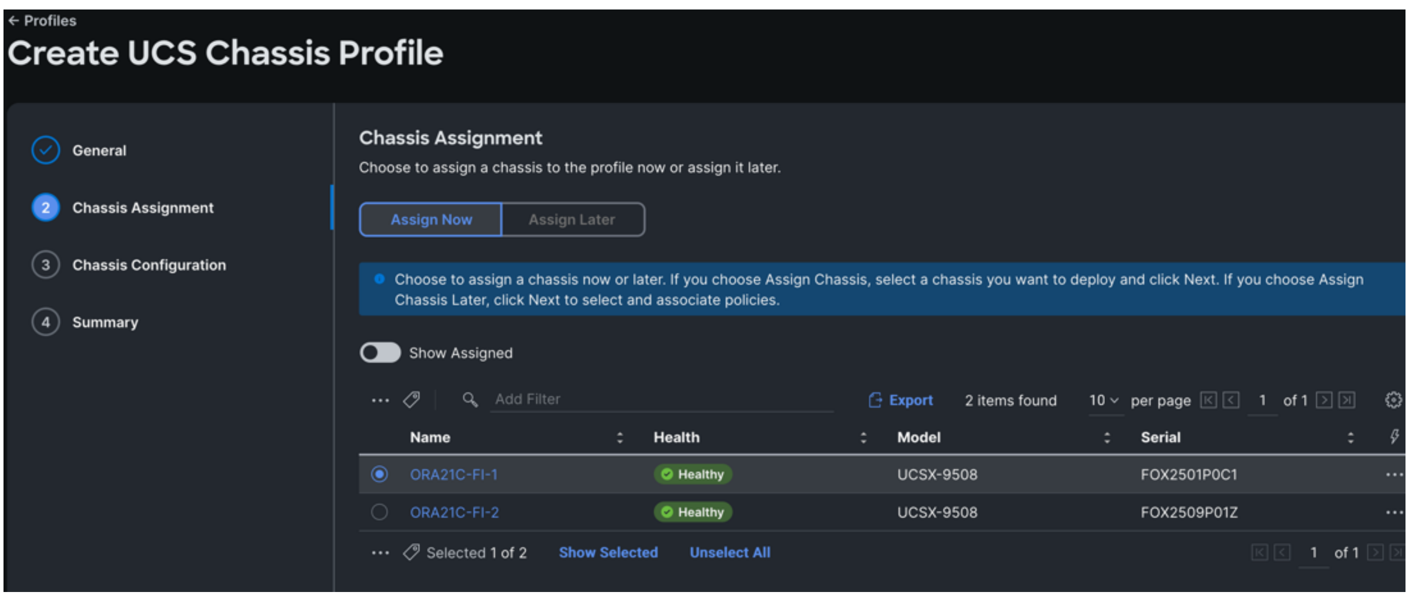
A Cisco UCS Chassis profile enables you to create and associate chassis policies to an Intersight Managed Mode (IMM) claimed chassis. When a chassis profile is associated with a chassis, Cisco Intersight automatically configures the chassis to match the configurations specified in the policies of the chassis profile. The chassis-related policies can be attached to the profile either at the time of creation or later. Please refer to this link for more details: https://intersight.com/help/saas/features/chassis/configure#chassis_profiles.

The chassis profile in a FlexPod is used to set the power policy for the chassis. By default, UCSX power supplies are configured in GRID mode, but the power policy can be utilized to set the power supplies in non-redundant or N+1/N+2 redundant modes

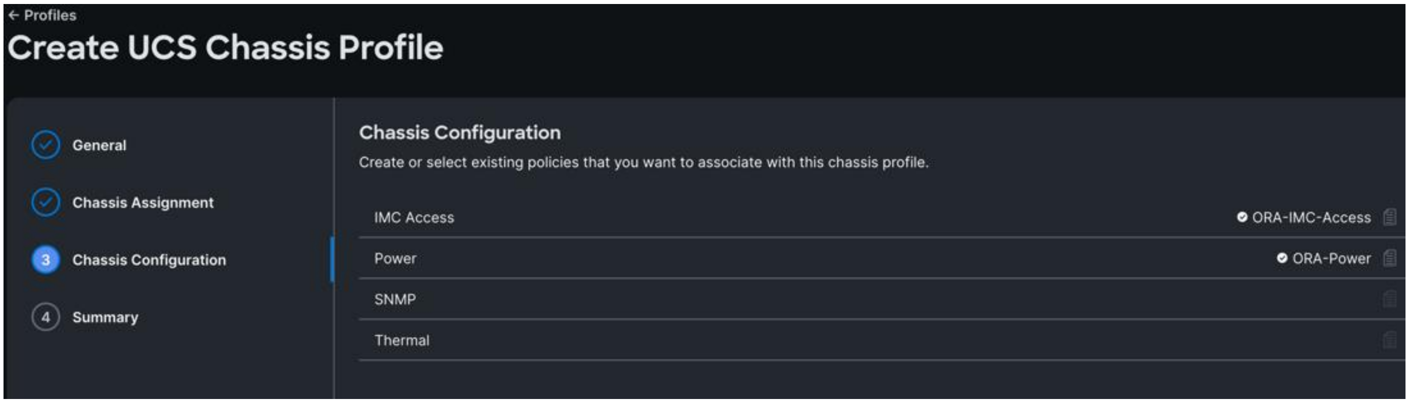
Step 1. To create a Cisco UCS Chassis Profile, go to Infrastructure Service > Configure > Profiles > UCS Chassis Domain Profiles tab > and click Create UCS Chassis Profile.



Step 2. In the Chassis Assignment menu, for the first chassis, click "ORA21C-FI-1" and click Next.

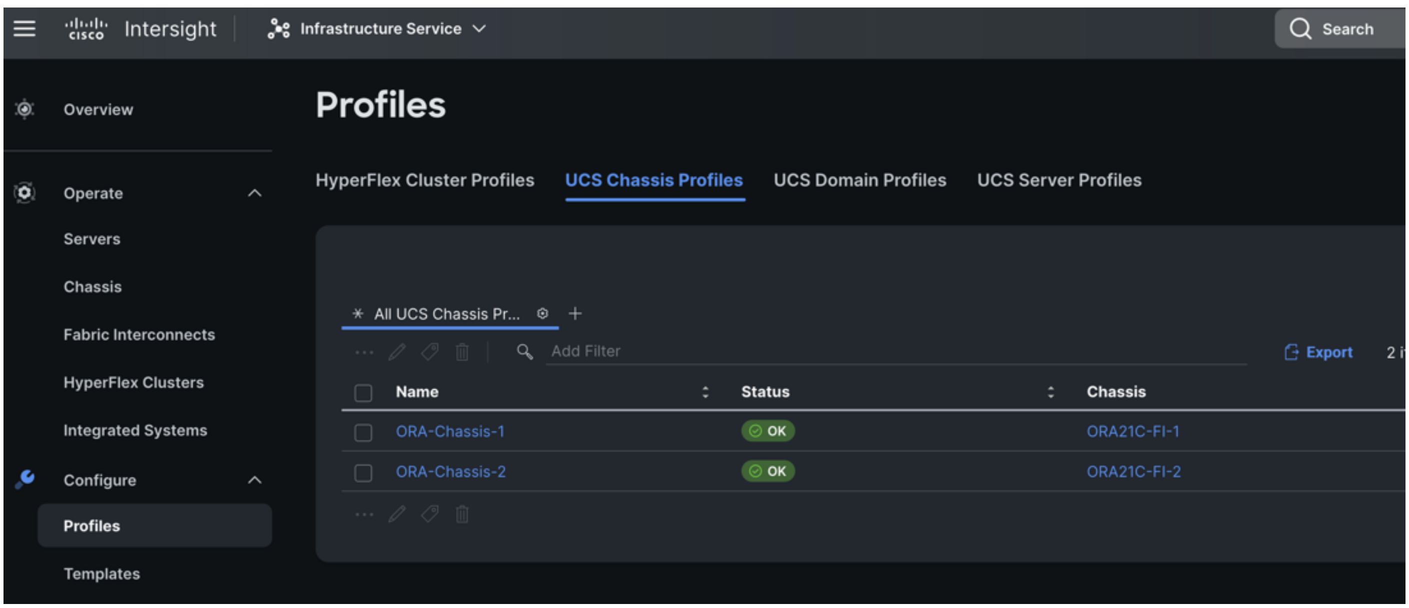


Step 3. In the Chassis configuration section, for the policy for IMC Access select "ORA-IMC-Access" and for the Power policy select "ORA-Power."



Step 4. Review the configuration settings summary for the Chassis Profile and click Deploy to create the Cisco UCS Chassis Profile for the first chassis.

Note: For this solution, we created two Chassis Profile (ORA-Chassis-1 and ORA-Chassis-2) and assigned to both the chassis as shown below:

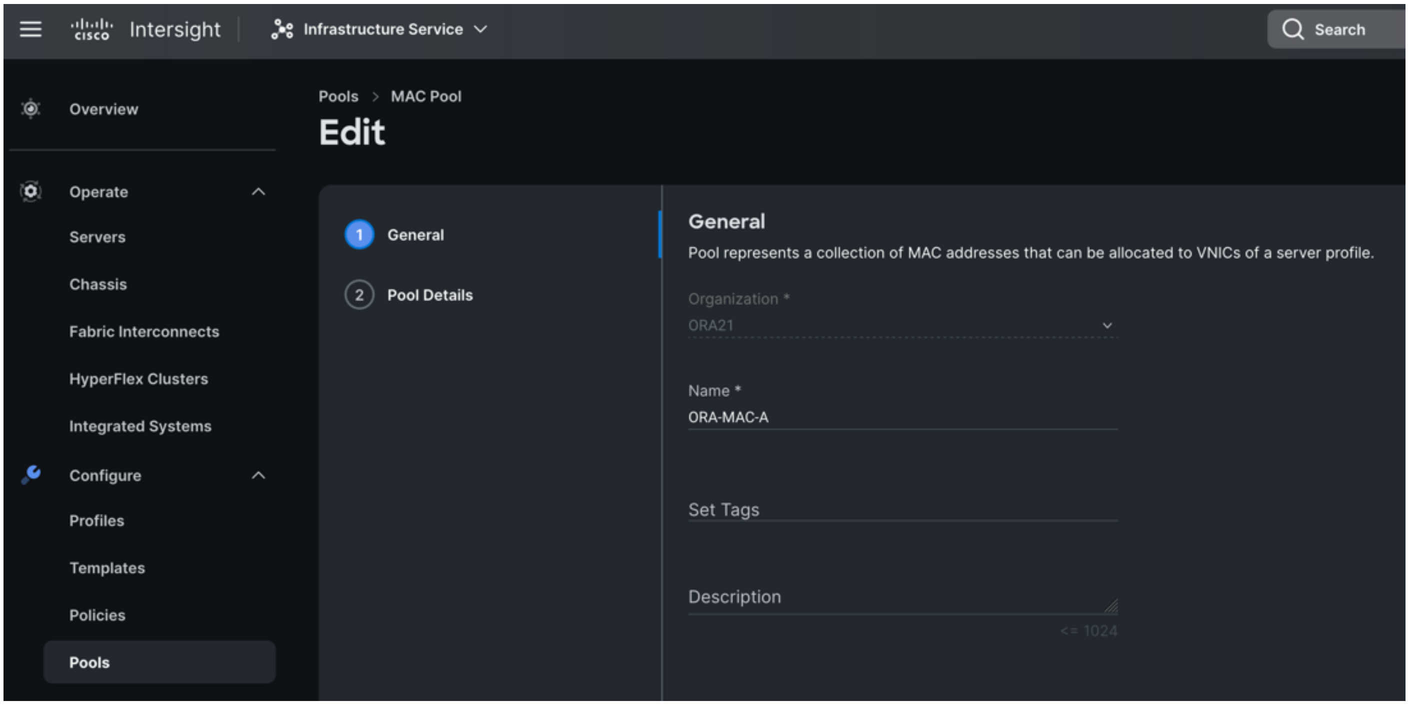


Configure Policies for Cisco UCS Domain

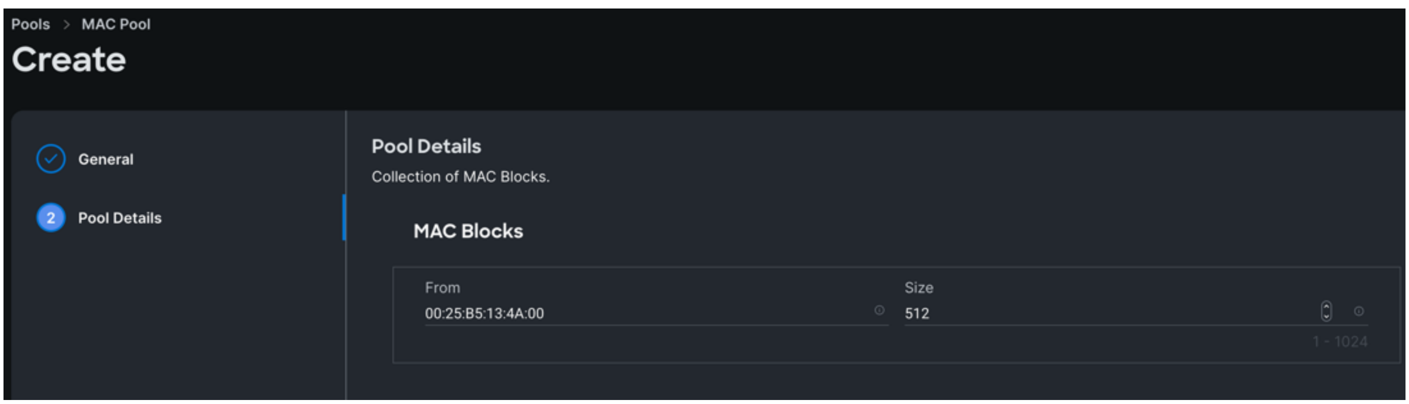
Procedure 1. Create MAC Pool

Step 1. To configure a MAC Pool for a Cisco UCS Domain profile, go to > Infrastructure Service > Configure > Pools > and click Create Pool. Select option MAC to create MAC Pool.

Step 2. In the MAC Pool Create section, for the Organization, select “ORA21” and for the Policy name “ORA-MAC-A.” Click Next.



Step 3. Enter the MAC Blocks from and Size of the pool according to your environment and click Create.

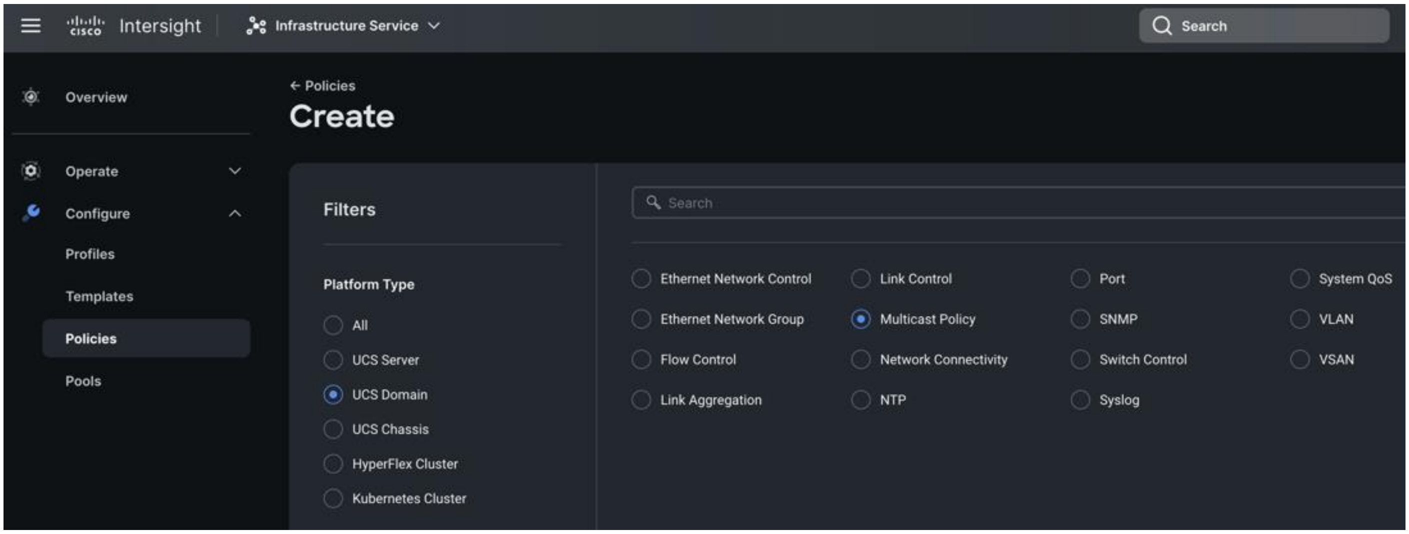


Note: For this solution, we configured four MAC Pools. ORA-MAC-A for vNICs MAC Address VLAN 134 (public network traffic) on all the servers through FI-A Side. ORA-MAC-B for vNICs MAC Address of VLAN 10 (private network traffic) on all servers through FI-B Side. ORA-MAC-Storage-A for vNICs MAC Address of VLAN 21 and VLAN 23 (storage network traffic) on all servers through FI-A Side. ORA-MAC-Storage-B for vNICs MAC Address of VLAN 22 and VLAN 24 (storage network traffic) on all servers through FI-B Side.

Step 4. Create three additional MAC Pool to provide MAC addresses to all vNICs running on different VLAs.

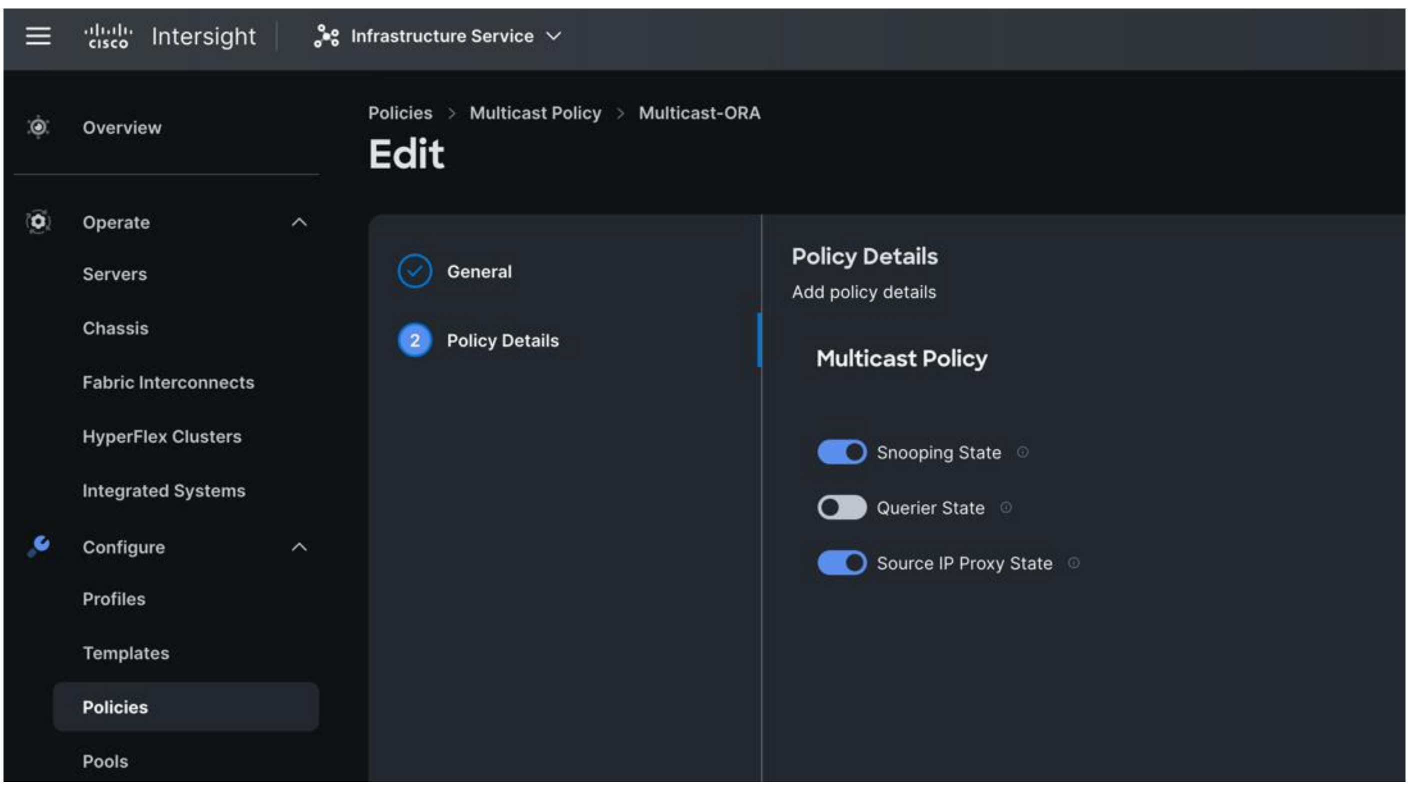
Procedure 2. Configure Multicast Policy

Step 1. To configure Multicast Policy for a Cisco UCS Domain profile, go to > Infrastructure Service > Configure > Polices > and click Create Policy. For the platform type select "UCS Domain" and for Policy, select "Multicast Policy."



Step 2. In the Multicast Policy Create section, for the Organization select "ORA21" and for the Policy name "Multicast-ORA." Click Next.

Step 3. In the Policy Details section, select Snooping State and Source IP Proxy State.

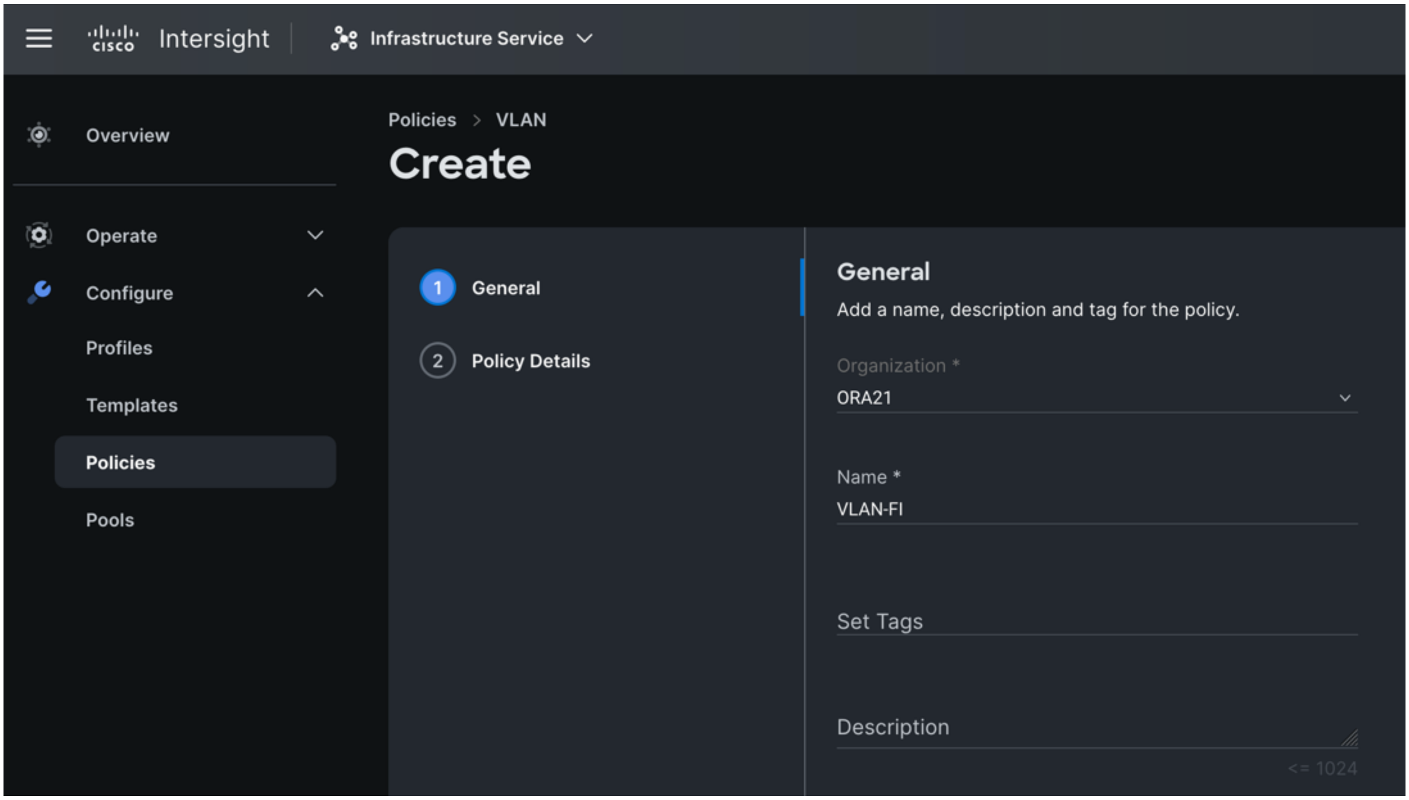


Step 4. Click Create to create this policy.

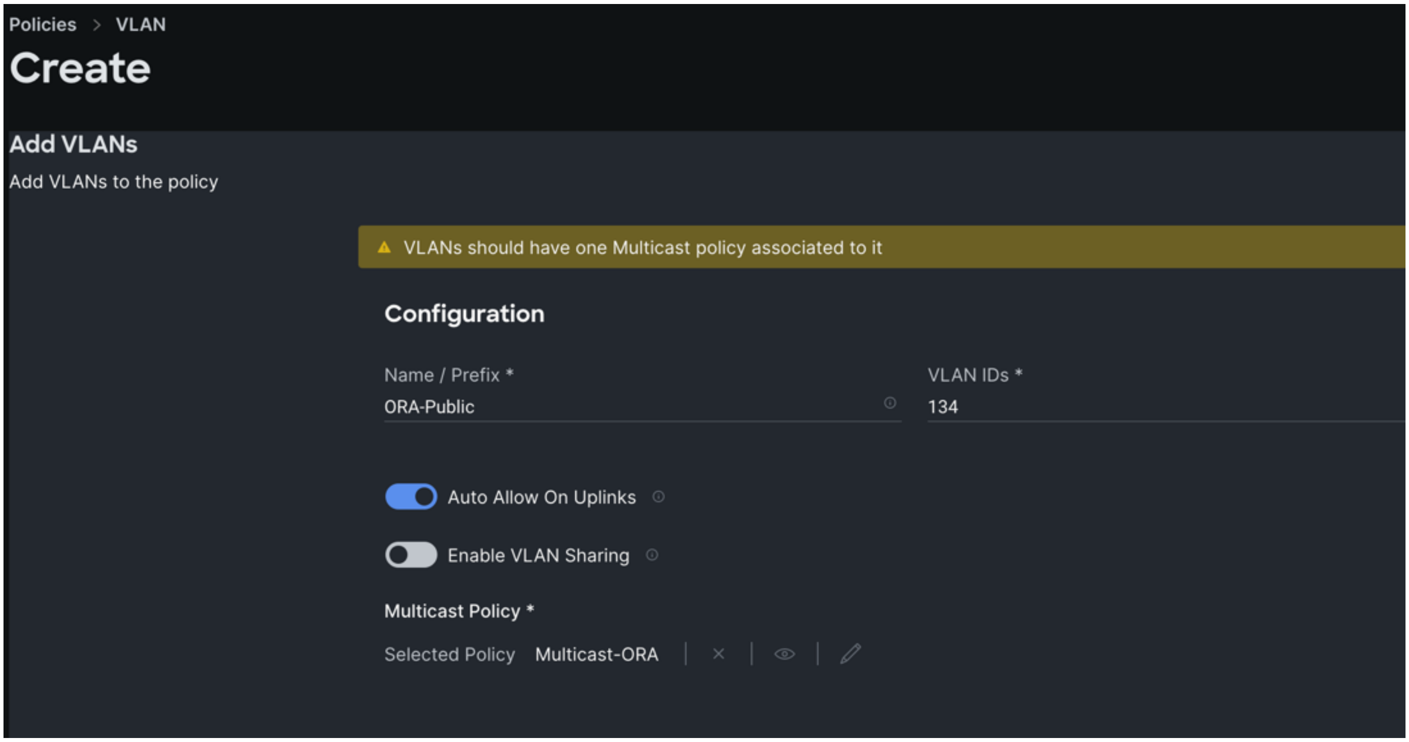
Procedure 3. Configure VLANs

Step 1. To configure the VLAN Policy for the Cisco UCS Domain profile, go to > Infrastructure Service > Configure > Polices > and click Create Policy. For the platform type select "UCS Domain" and for the Policy select "VLAN."

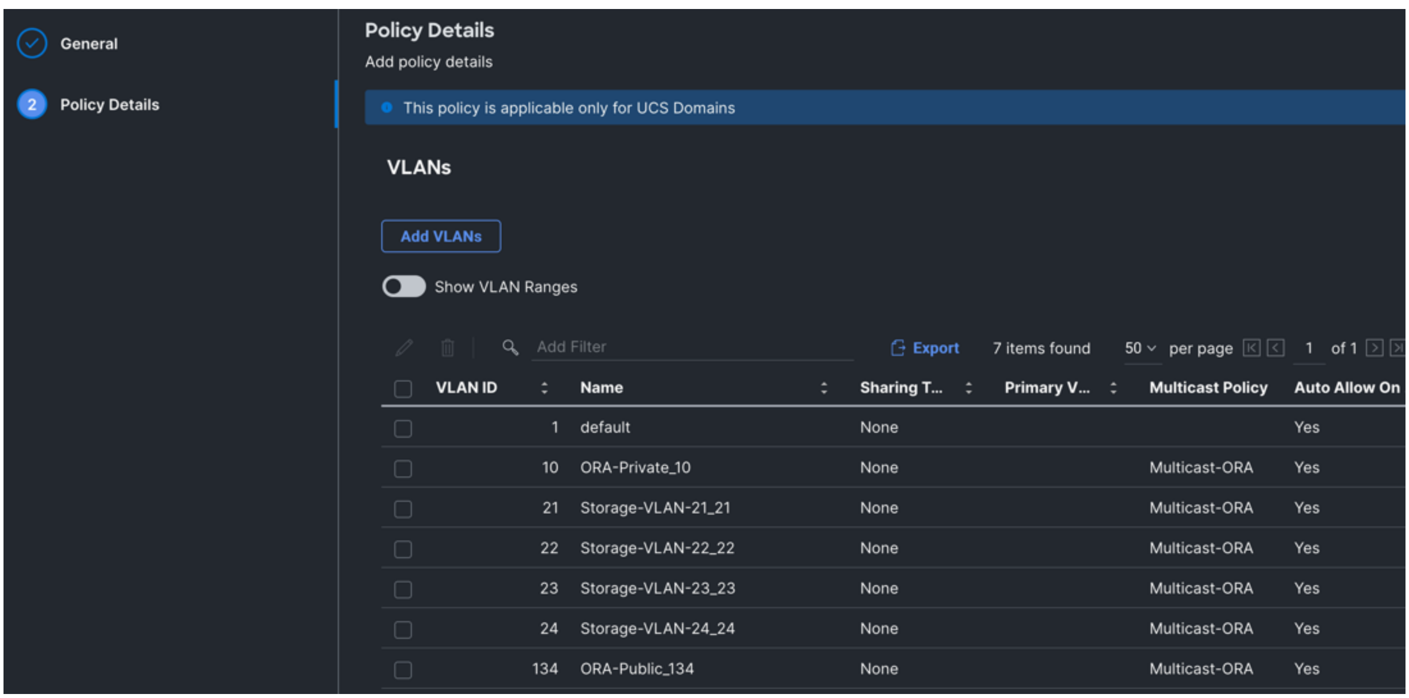
Step 2. In the VLAN Policy Create section, for the Organization select "ORA21" and for the Policy name select "VLAN-FI." Click Next.



Step 3. In the Policy Details section, to configure the individual VLANs, select " Add VLANs." Provide a name, VLAN ID for the VLAN and select the Multicast Policy as shown below.



Step 4. Click Add to add this VLAN to the policy. Add VLAN 10, 21, 22, 23 and 24 and provide the names to various network traffic of this solution.

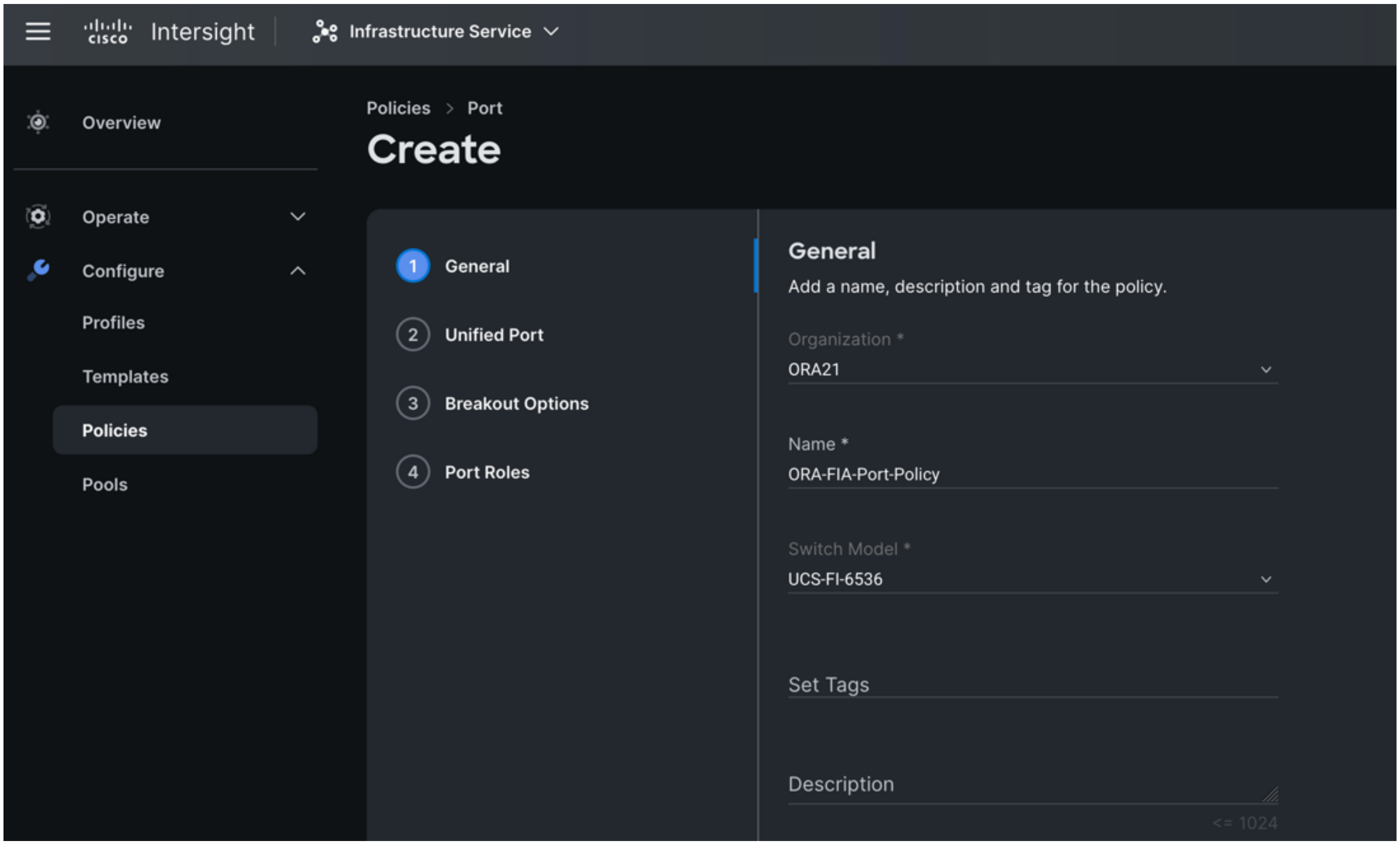


Step 5. Click Create to create this policy.

Procedure 4. Configure Port Policy

Step 1. To configure the Port Policy for the Cisco UCS Domain profile, go to > Infrastructure Service > Configure > Polices > and click Create Policy. For the platform type select "UCS Domain" and for the policy, select "Port."

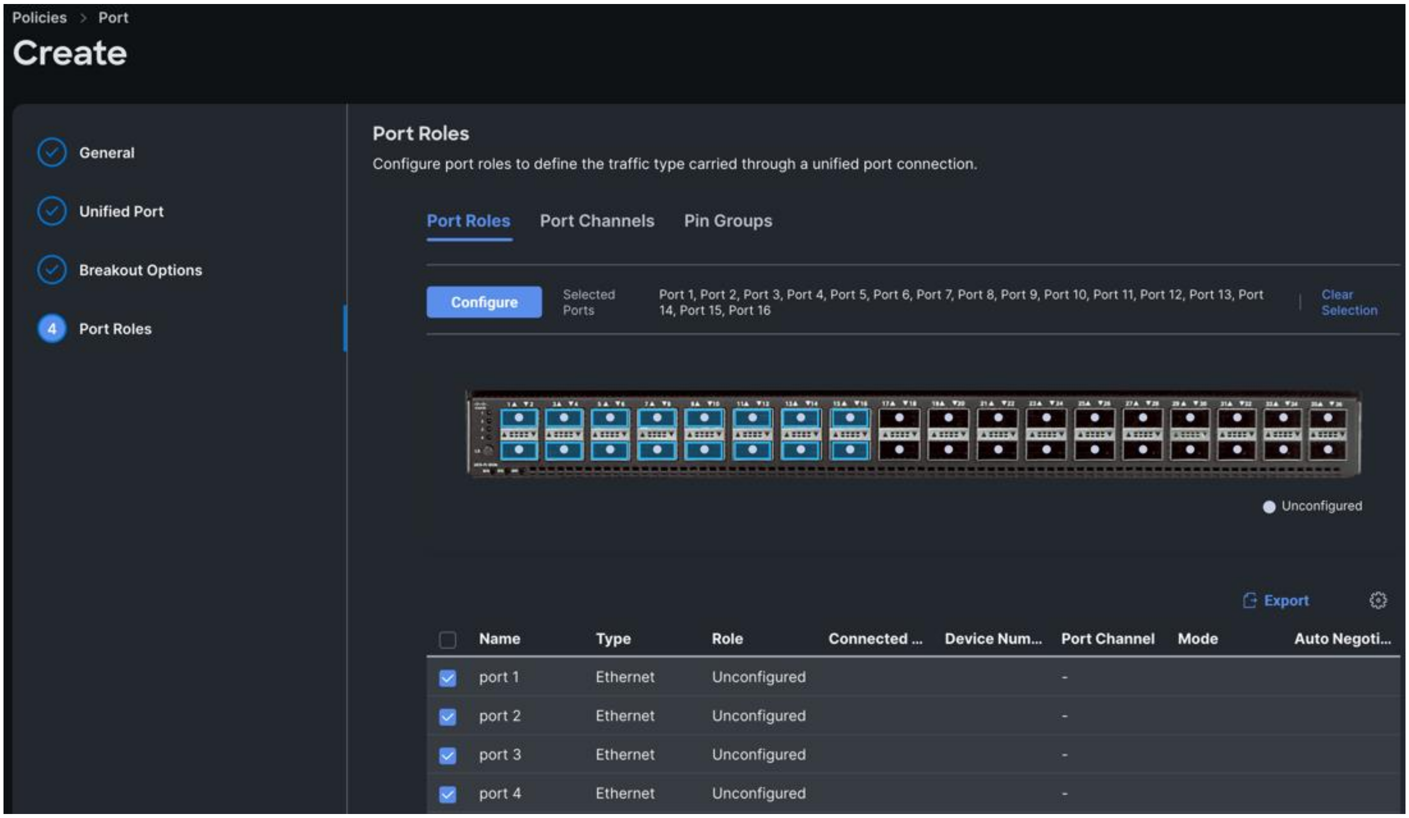
Step 2. In the Port Policy Create section, for the Organization, select "ORA21", for the policy name select "ORA-FIA-Port-Policy" and for the Switch Model select " UCS-FI-6536." Click Next.



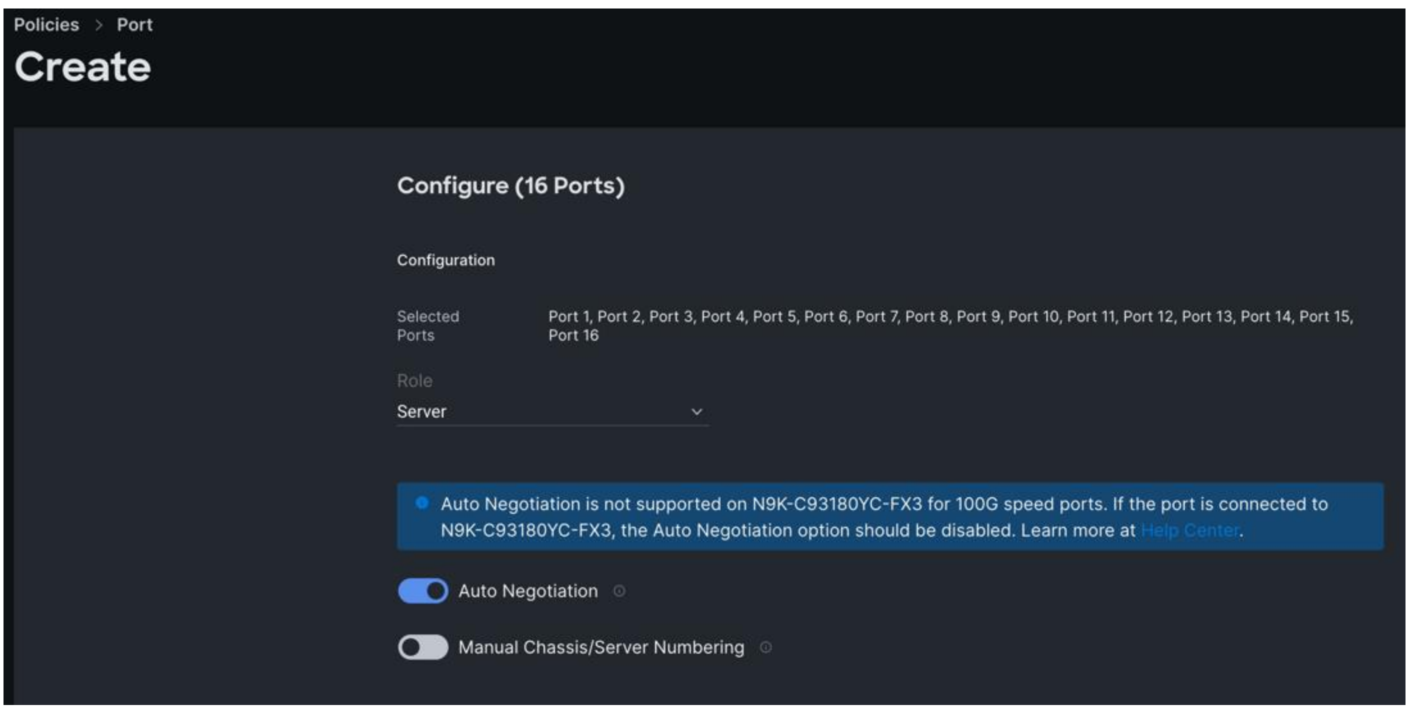
Note: We did not configure the Fibre Channel Ports for this solution. In the Unified Port section, leave it as default and click Next.

Note: We did not configure the Breakout options for this solution. Leave it as default and click Next.

Step 3. In the Port Role section, select port 1 to 16 and click Configure.

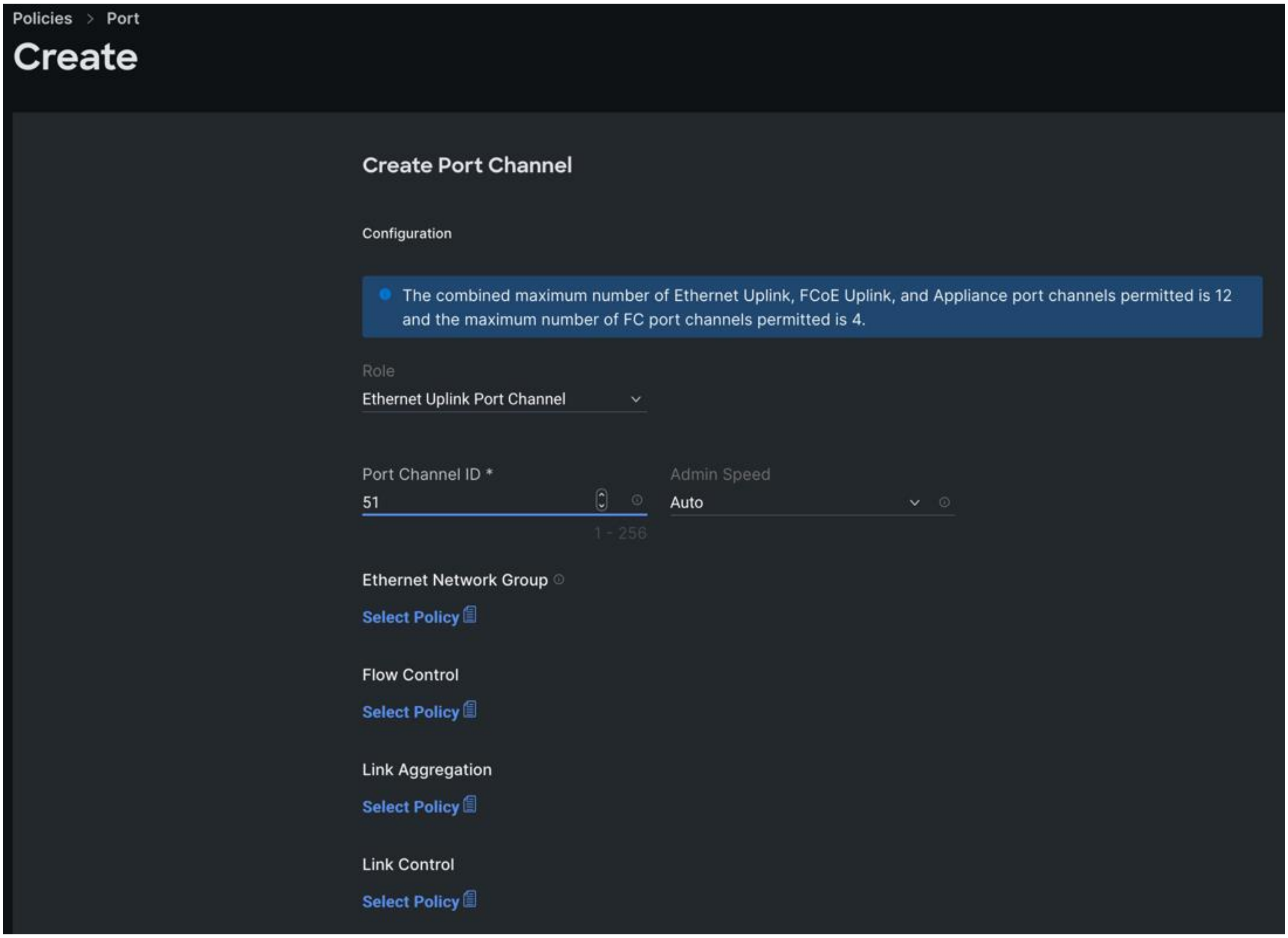


Step 4. In the Configure section, for Role select Server and keep the Auto Negotiation ON.

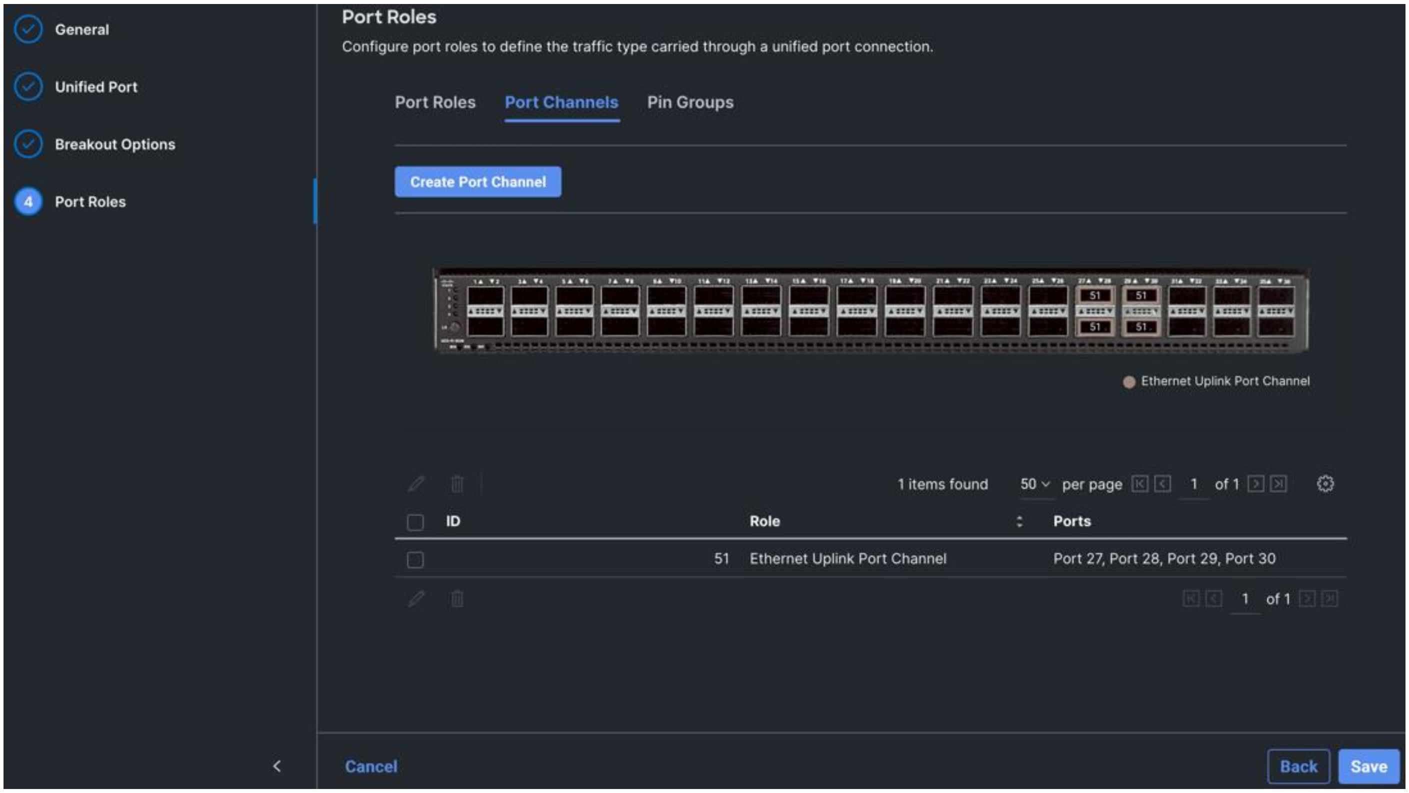


Step 5. Click SAVE to add this configuration for port roles.

Step 6. Go to the Port Channels tab and select Port 27 to 30 and click Create Port Channel between FI-A and both Cisco Nexus Switches. In the Create Port Channel section, for Role select Ethernet Uplinks Port Channel, and for the Port Channel ID select 51 and select Auto for the Admin Speed.

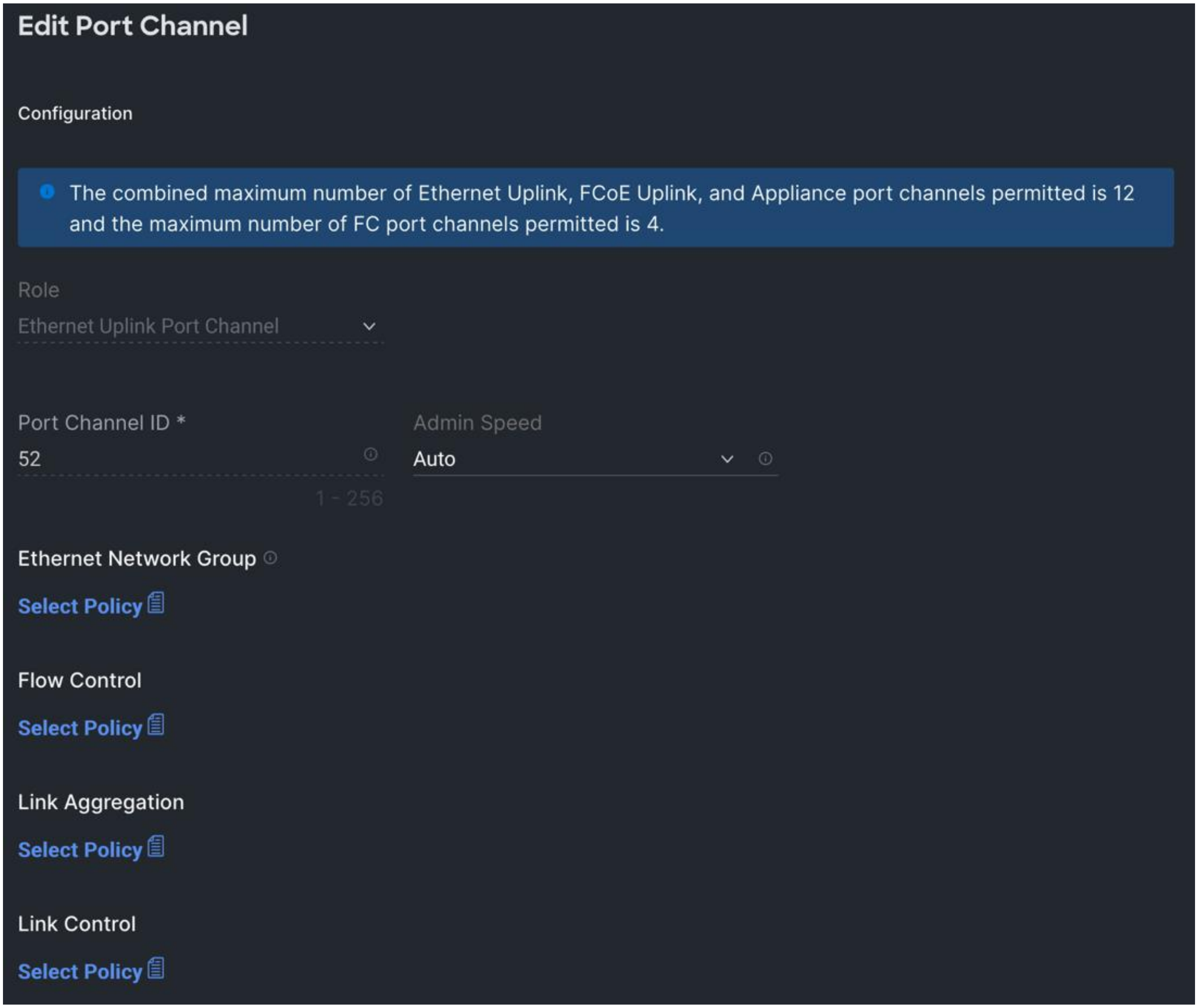


Step 7. Click SAVE to add this configuration for uplink port roles.



Step 8. Click SAVE again to complete this configuration for all the server ports and uplink port roles.

Note: We configured the FI-B ports and created a Port Policy for FI-B, “ORA-FIB-Port-Policy.” In the FI-B port policy, we configured port 1 to 16 for server ports and port 27 to 30 as the ethernet uplink ports. For FI-B, we configured Port-Channel ID as 52, as shown below, to create another Port Channel between FI-B to both Cisco Nexus switches.



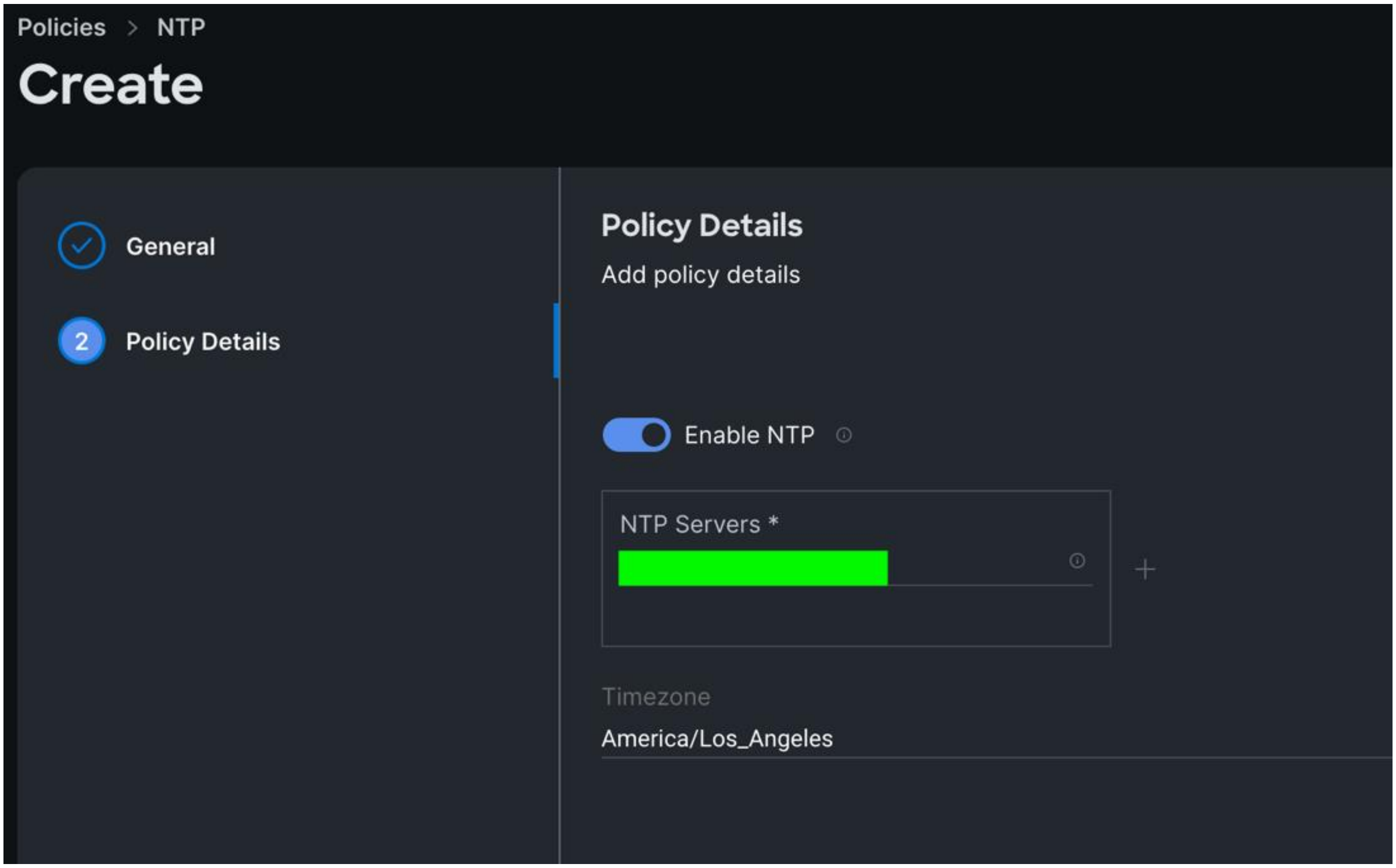
This completes the Port Policy for Cisco UCS Domain profile.

Procedure 5. Configure NTP Policy

Step 1. To configure the NTP Policy for the Cisco UCS Domain profile, go to > Infrastructure Service > Configure > Polices > and click Create Policy. For the platform type select “UCS Domain” and for the policy select “NTP.”

Step 2. In the NTP Policy Create section, for the Organization select “ORA21” and for the policy name select “NTP-Policy.” Click Next.

Step 3. In the Policy Details section, select the option to enable the NTP Server and enter your NTP Server details as shown below.



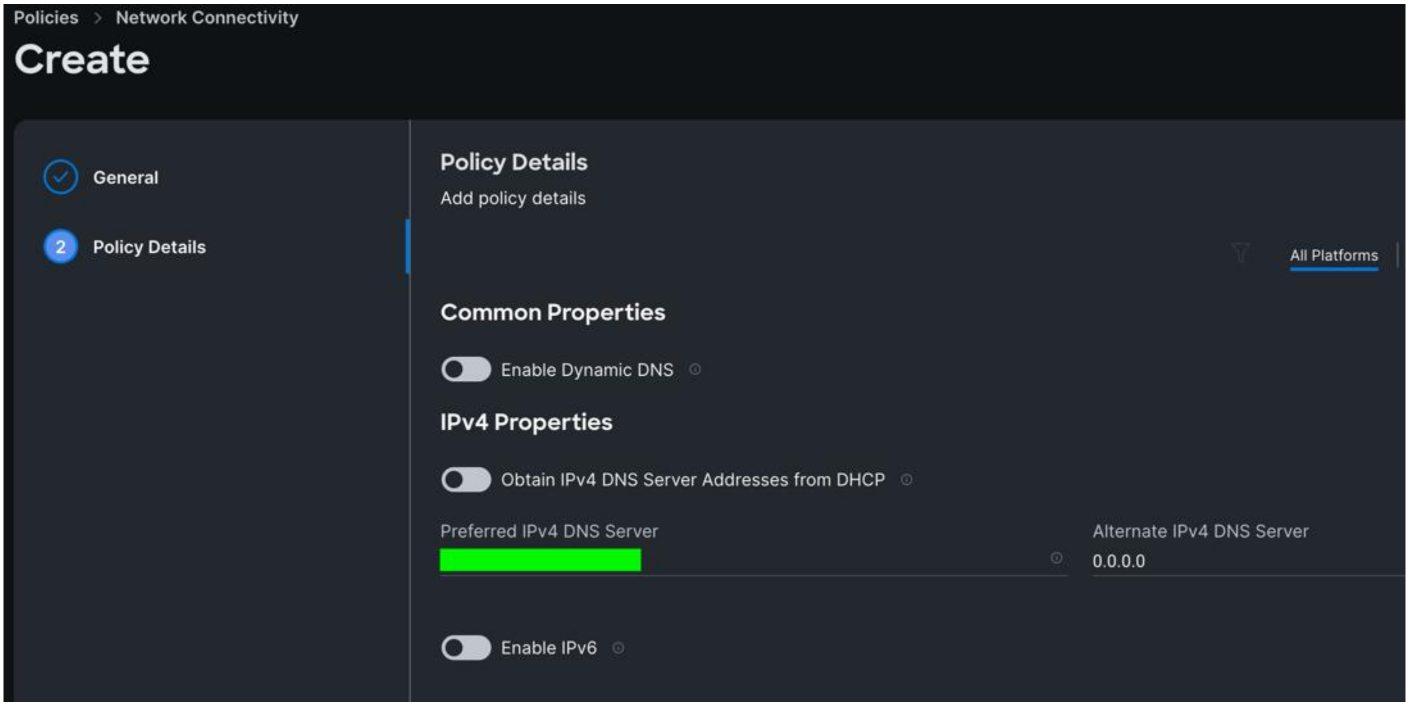
Step 4. Click Create.

Procedure 6. Configure Network Connectivity Policy

Step 1. To configure to Network Connectivity Policy for the Cisco UCS Domain profile, go to > Infrastructure Service > Configure > Polices > and click Create Policy. For the platform type select "UCS Domain" and for the policy select "Network Connectivity."

Step 2. In the Network Connectivity Policy Create section, for the Organization select "ORA21" and for the policy name select "Network-Connectivity-Policy." Click Next.

Step 3. In the Policy Details section, enter the IPv4 DNS Server information according to your environment details as shown below.



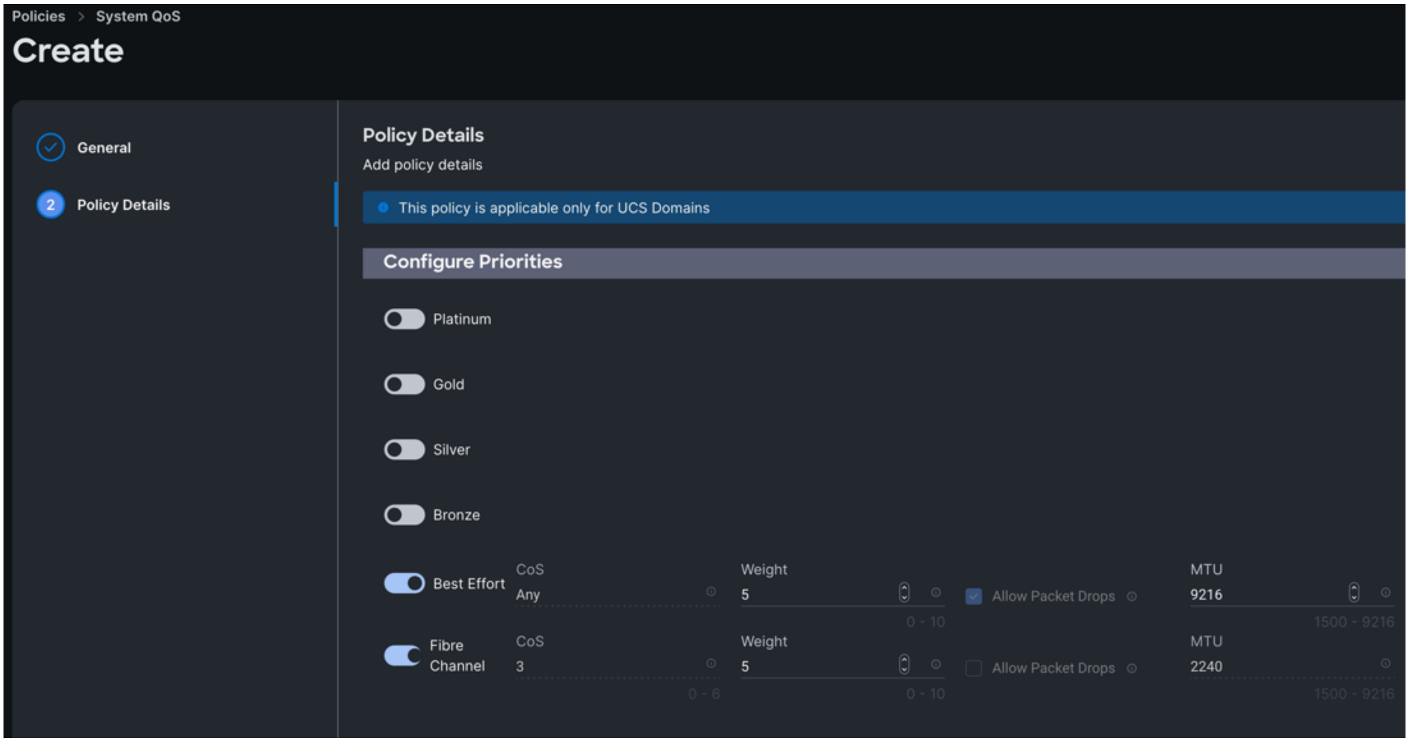
Step 4. Click Create.

Procedure 7. Configure System QoS Policy

Step 1. To configure the System QoS Policy for the Cisco UCS Domain profile, go to > Infrastructure Service > Configure > Polices > and click Create Policy. For the platform type select “UCS Domain” and for the policy select “System QoS.”

Step 2. In the System QoS Policy Create section, for the Organization select “ORA21” and for the policy name select “ORA-QoS.” Click Next.

Step 3. In the Policy Details section under Configure Priorities, select Best Effort and set the MTU size to 9216.



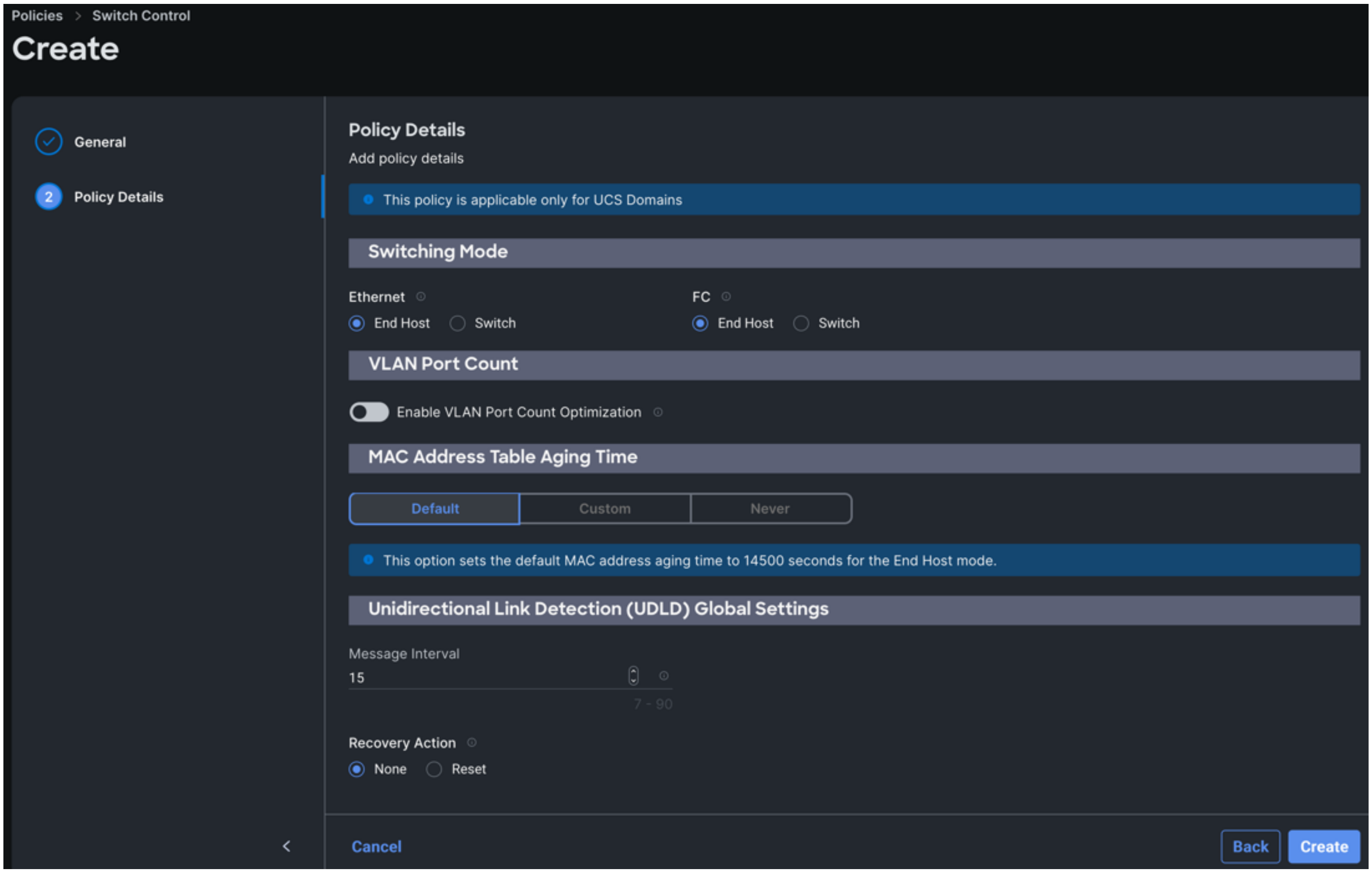
Step 4. Click Create.

Procedure 8. Configure Switch Control Policy

Step 1. To configure the Switch Control Policy for the UCS Domain profile, go to > Infrastructure Service > Configure > Polices > and click Create Policy. For the platform type select "UCS Domain" and for the policy select "Switch Control."

Step 2. In the Switch Control Policy Create section, for the Organization select "ORA21" and for the policy name select "ORA-Switch-Control." Click Next.

Step 3. In the Policy Details section, for the Switching Mode for Ethernet, keep " End Host" Mode.



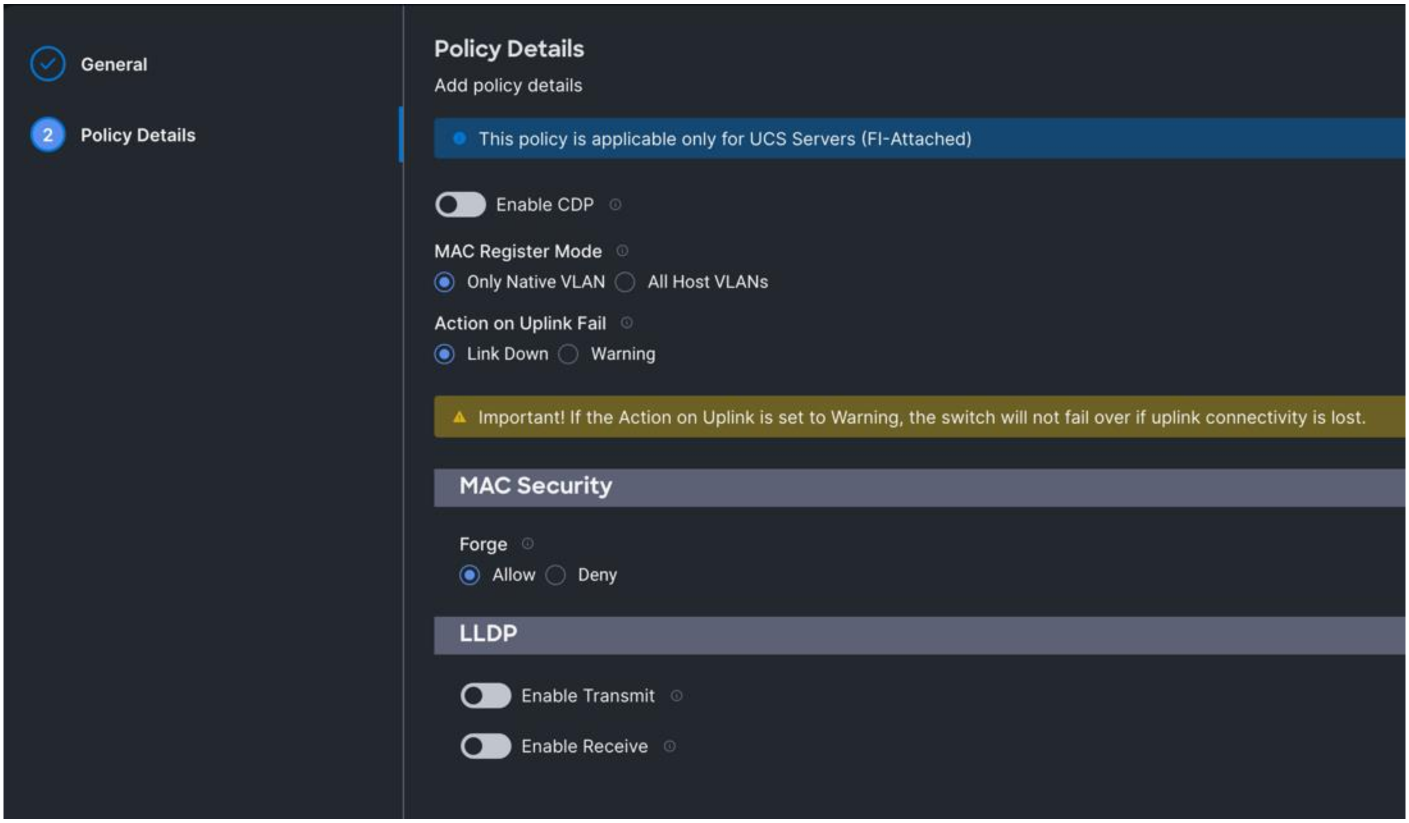
Step 4. Click Create to create this policy.

Procedure 9. Configure Ethernet Network Control Policy

Step 1. To configure the Ethernet Network Control Policy for the UCS Domain profile, go to > Infrastructure Service > Configure > Polices > and click Create Policy. For the platform type select “UCS Domain” and for the policy select “Ethernet Network Control.”

Step 2. In the Switch Control Policy Create section, for the Organization select “ORA21” and for the policy name enter “ORA-Eth-Network-Control.” Click Next.

Step 3. In the Policy Details section, keep the parameter as shown below.



Step 4. Click Create to create this policy.

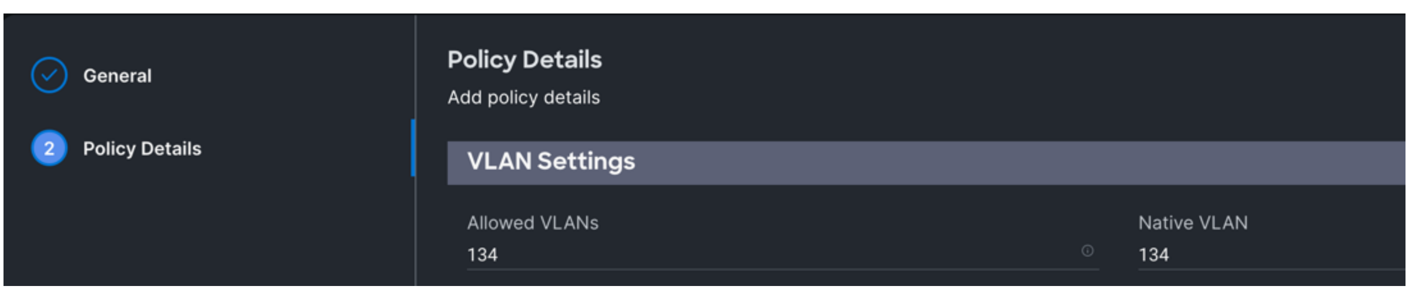
Procedure 10. Configure Ethernet Network Group Policy

Note: We configured six Ethernet Network Groups to allow six different VLAN traffic for this solution.

Step 1. To configure the Ethernet Network Group Policy for the UCS Domain profile, go to > Infrastructure Service > Configure > Polices > and click Create Policy. For the platform type select "UCS Domain" and for the policy select "Ethernet Network Group."

Step 2. In the Switch Control Policy Create section, for the Organization select "ORA21" and for the policy name enter "Eth-Network-134." Click Next.

Step 3. In the Policy Details section, for the Allowed VLANs and Native VLAN enter 134 as shown below.



Step 4. Click Create to create this policy for VLAN 134.

Note: For this solution, we did the following:

Created "Eth-Network-10" and added VLAN 10 for the Allowed VLANs and Native VLAN.
For VLAN 21, created "Eth-Network-21" and added VLAN 21 for the Allowed VLANs and Native VLAN.
For VLAN 22, created "Eth-Network-22" and added VLAN 22 for the Allowed VLANs and Native VLAN.
For VLAN 23, created "Eth-Network-23" and added VLAN 23 for the Allowed VLANs and Native VLAN.
For VLAN 24, created "Eth-Network-24" and added VLAN 24 for the Allowed VLANs and Native VLAN.

Note: We used these Ethernet Network Group policies and applied them on different vNICs to carry individual VLAN traffic for this solution.

Configure Cisco UCS Domain Profile

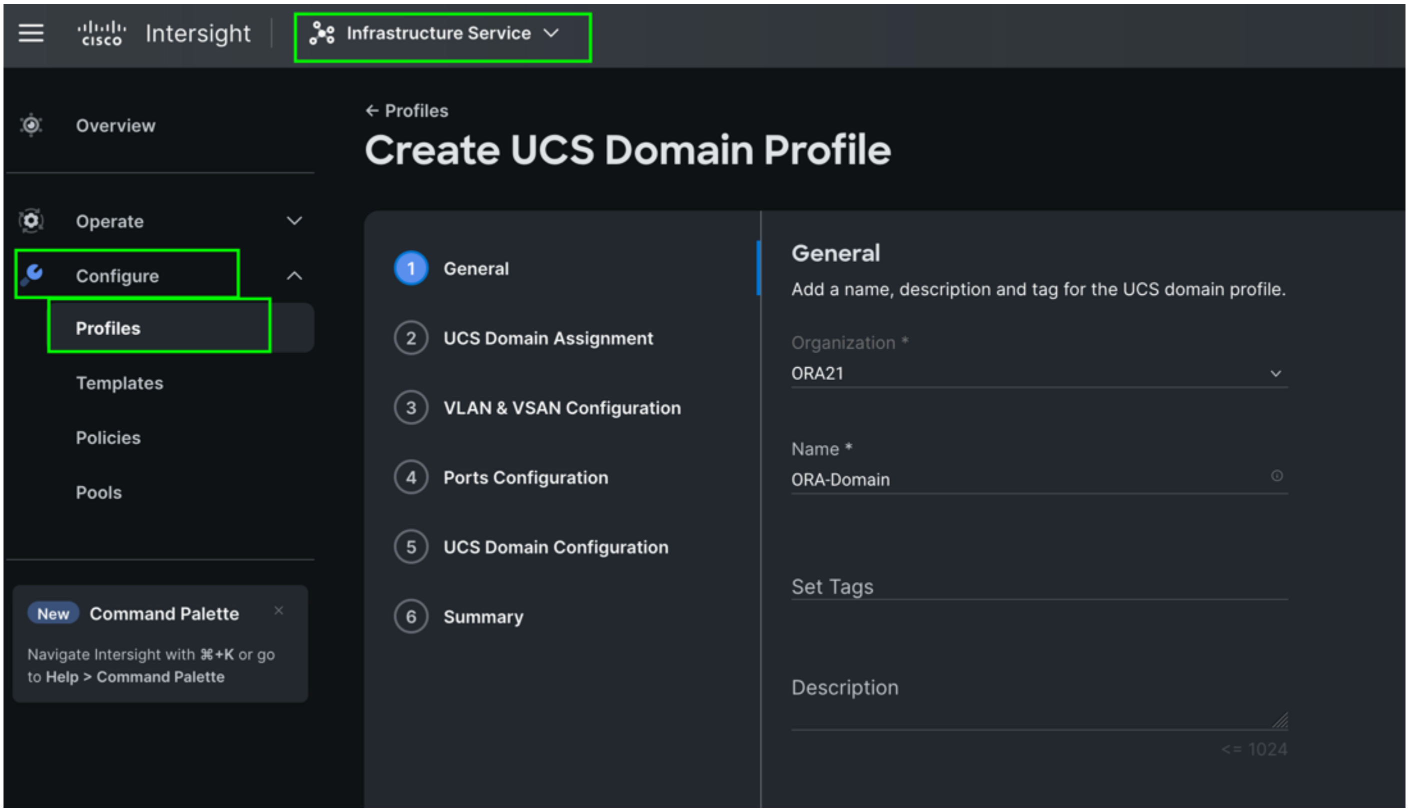
In Cisco Intersight, a domain profile configures a fabric interconnect pair through reusable policies, allows for configuration of the ports and port channels, and configures the VLANs and VSANs in the network. It defines the characteristics of and configures ports on fabric interconnects. You can create a domain profile and associate it with a fabric interconnect domain. The domain-related policies can be attached to the profile either at the time of creation or later. One UCS Domain profile can be assigned to one fabric interconnect domain. Refer to this link for more information: https://intersight.com/help/saas/features/fabric_interconnects/configure#domain_profile

Some of the characteristics of the Cisco UCS domain profile in the FlexPod environment are:

- A single domain profile (ORA-Domain) is created for the pair of Cisco UCS fabric interconnects.
- Unique port policies are defined for the two fabric interconnects.
- The VLAN configuration policy is common to the fabric interconnect pair because both fabric interconnects are configured for the same set of VLANs.
- The Network Time Protocol (NTP), network connectivity, and system Quality-of-Service (QoS) policies are common to the fabric interconnect pair.

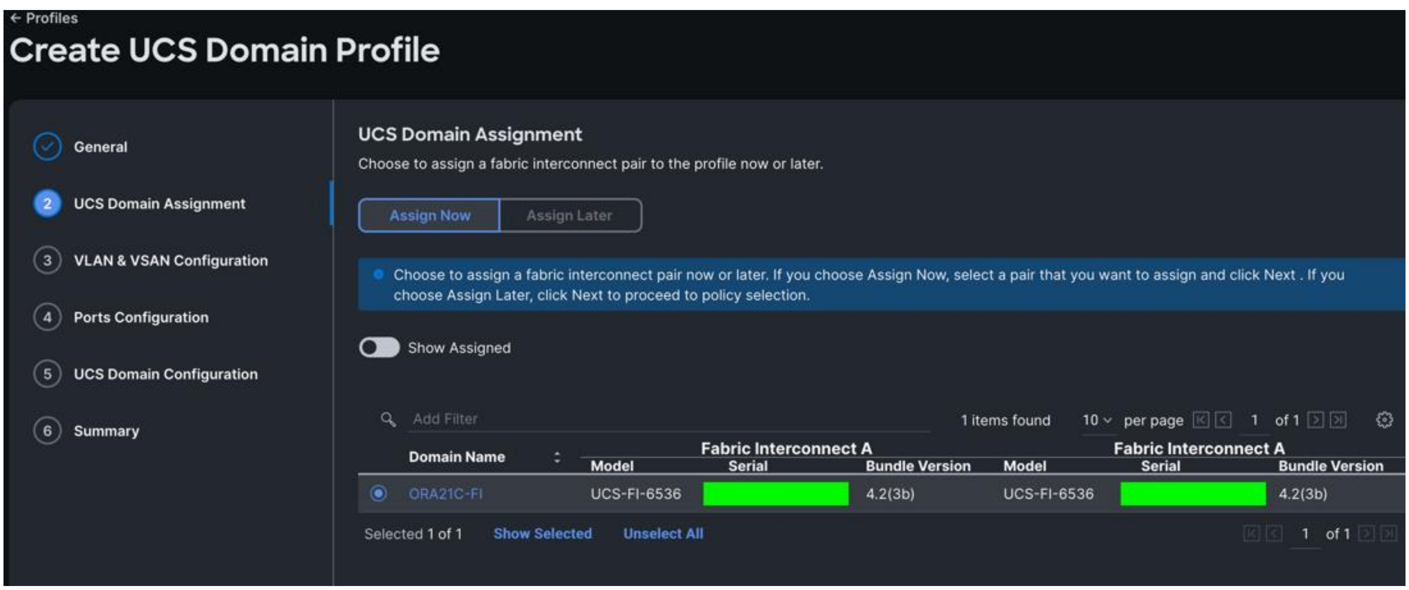
Procedure 1. Create a domain profile

Step 1. To create a domain profile, go to Infrastructure Service > Configure > Profiles > then go to the UCS Domain Profiles tab and click Create UCS Domain Profile.

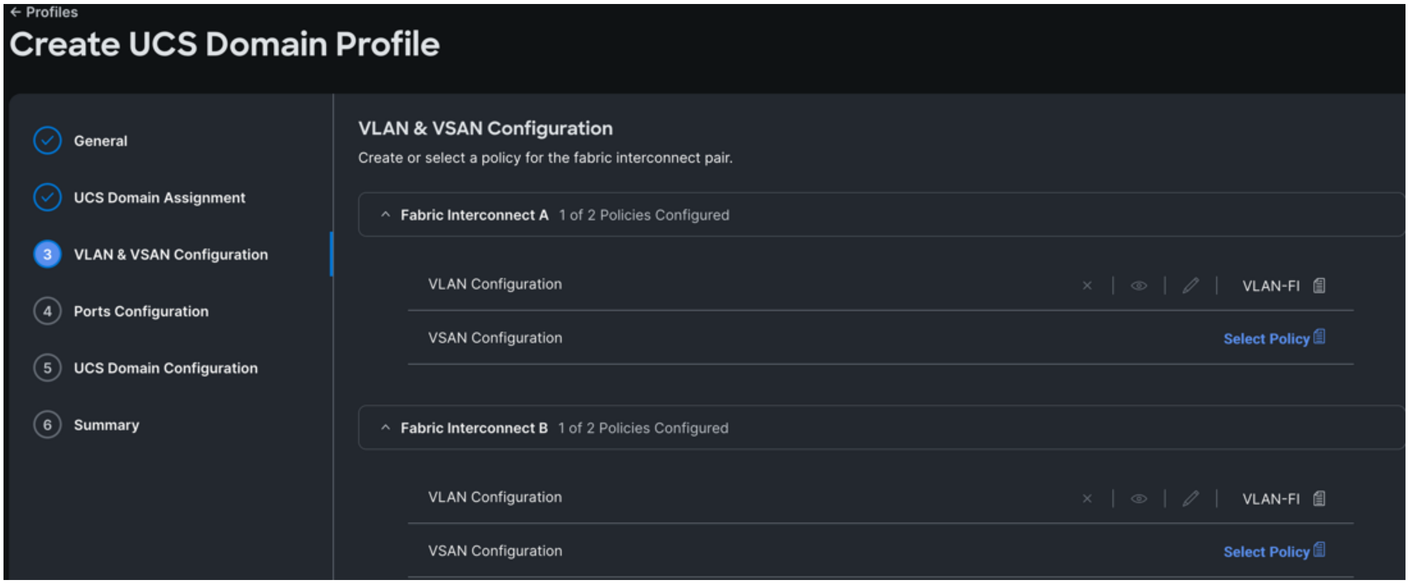


Step 2. For the domain profile name, enter "ORA-Domain" and for the Organization select what was previously configured. Click Next.

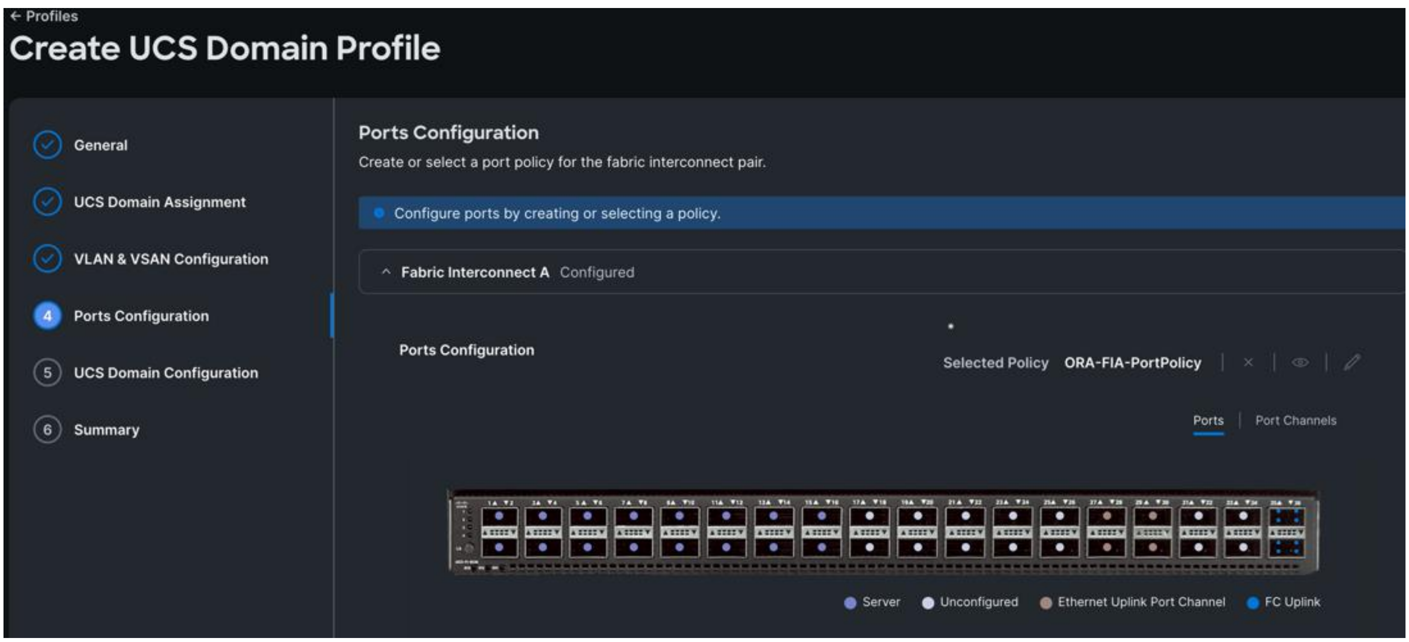
Step 3. In the UCS Domain Assignment menu, for the Domain Name select "ORA21C-FI" which was added previously into this domain and click Next.



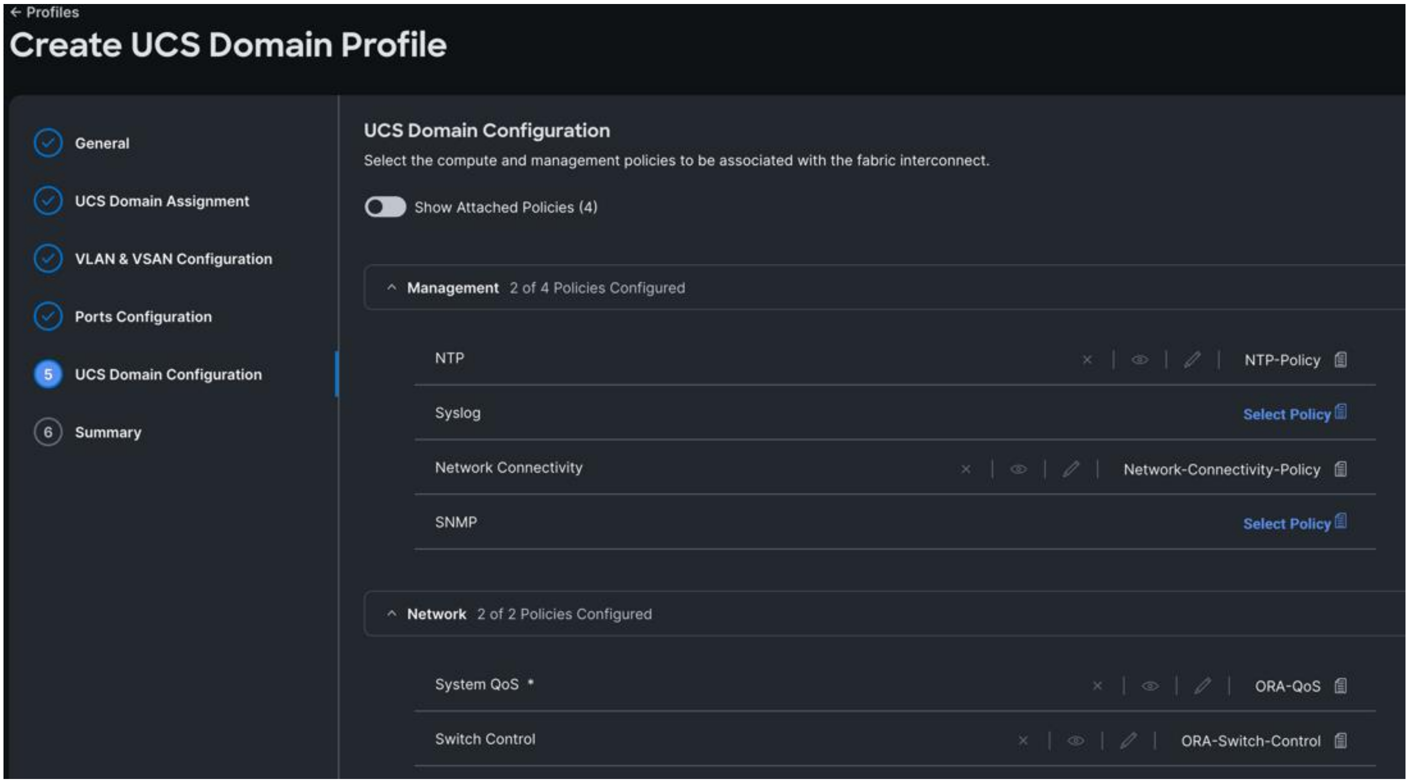
Step 4. In the VLAN & VSAN Configuration screen, for the VLAN Configuration select "VLAN-FI" and then click Next.



Step 5. In the Port Configuration section, for the Port Configuration Policy for FI-A select "ORA-FIA-PortPolicy" and "ORA-FIB-PortPolicy" the click Next.



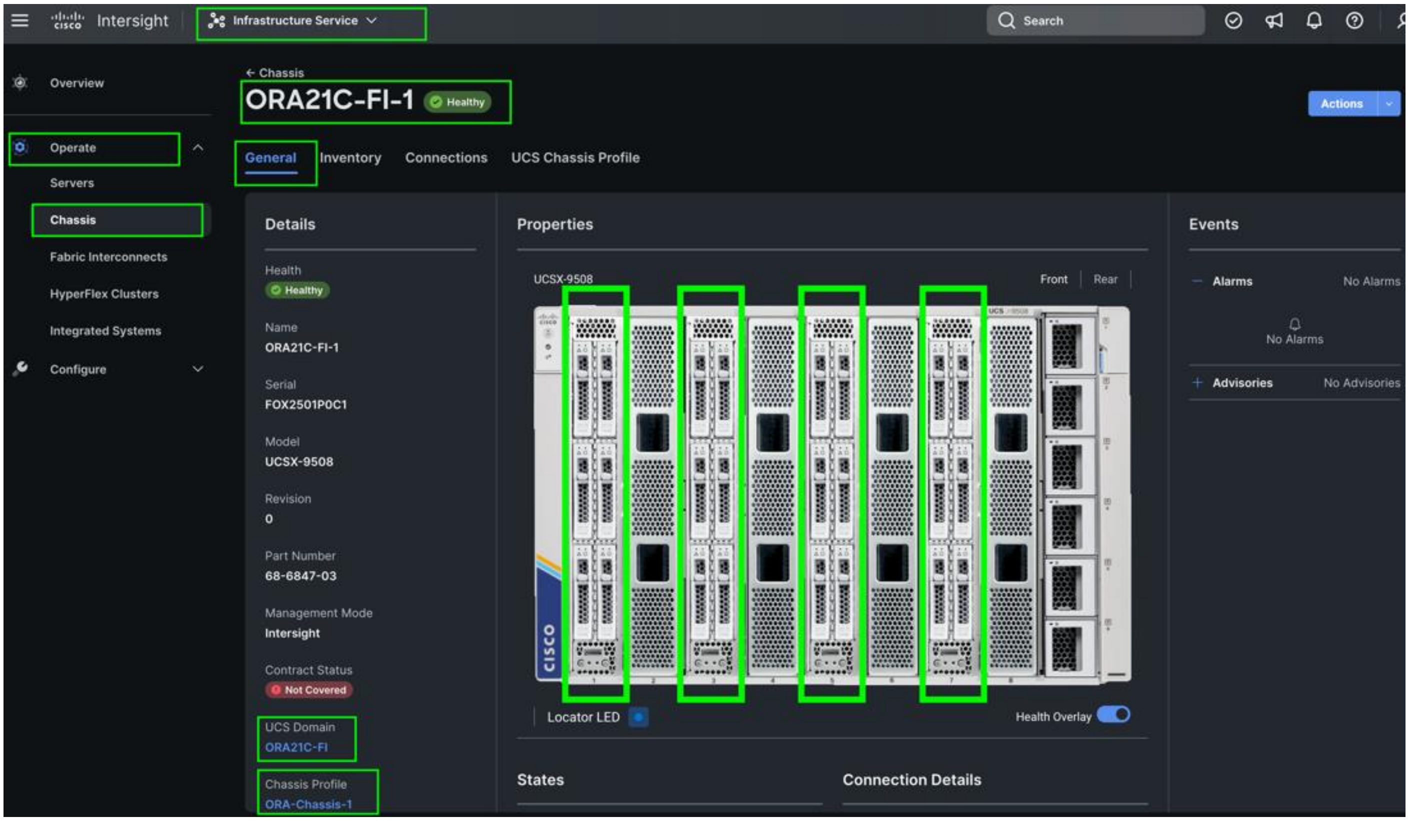
Step 6. In the UCS Domain Configuration section, select the policy for NTP, Network Connectivity, System QoS and Switch Control as shown below.



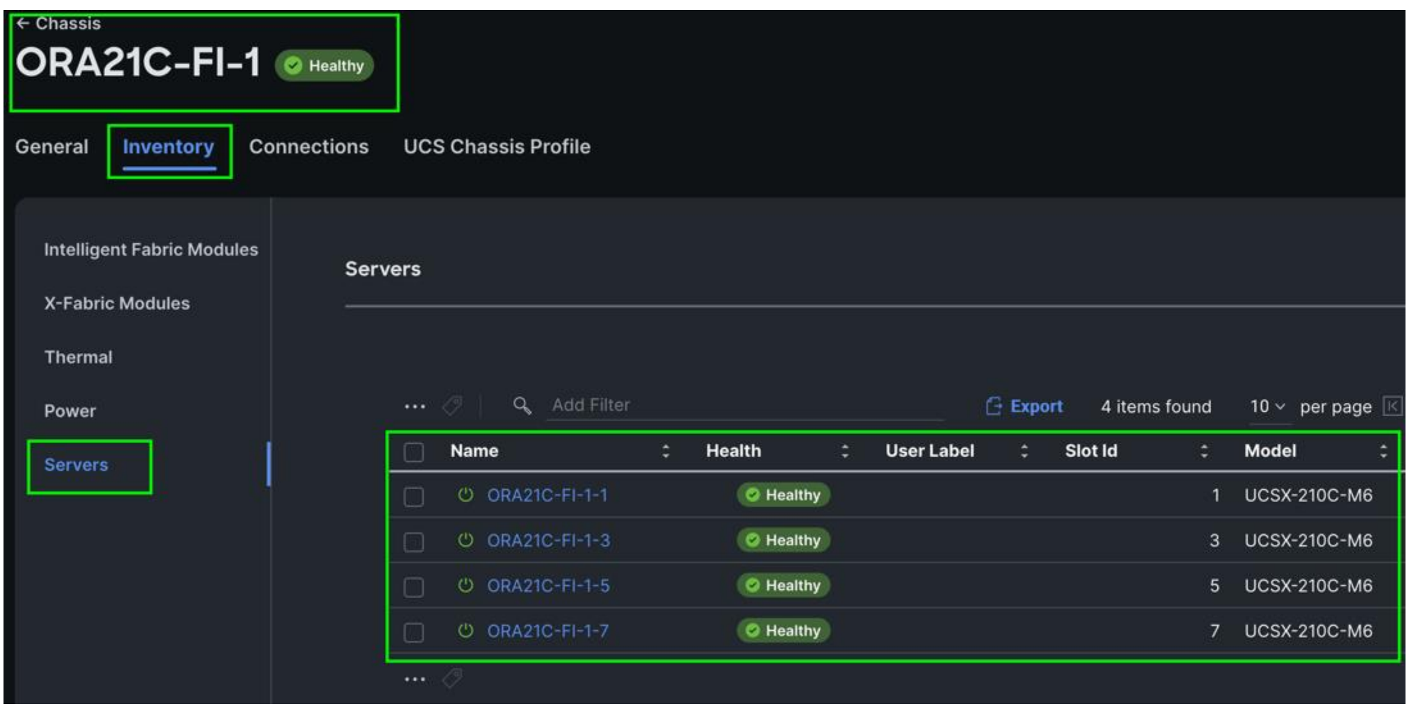
Step 7. In the Summary window, review the policies and click Deploy to create Domain Profile.

After the Cisco UCS domain profile has been successfully created and deployed, the policies including the port policies are pushed to the Cisco UCS fabric interconnects. The Cisco UCS domain profile can easily be cloned to install additional Cisco UCS systems. When cloning the Cisco UCS domain profile, the new Cisco UCS domains utilize the existing policies for the consistent deployment of additional Cisco UCS systems at scale.

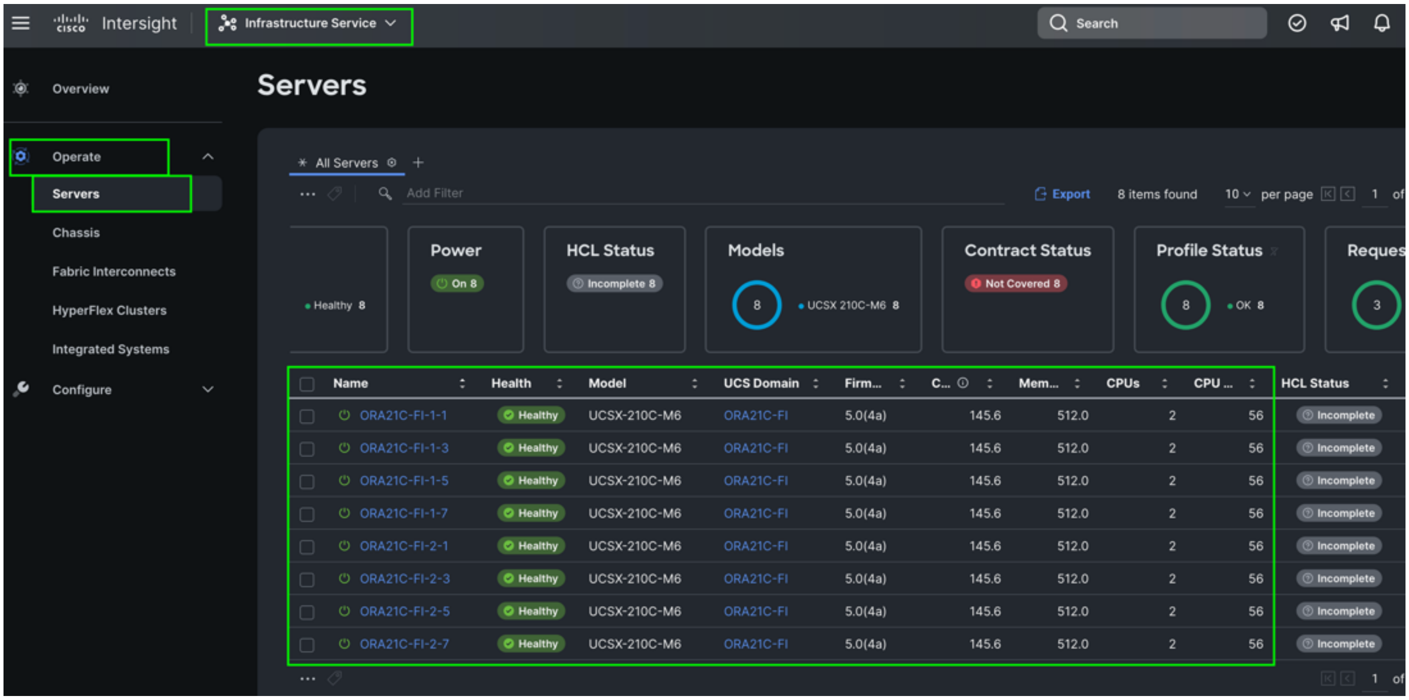
The Cisco UCS X9508 Chassis and Cisco UCS X210c M6 Compute Nodes are automatically discovered when the ports are successfully configured using the domain profile as shown below.



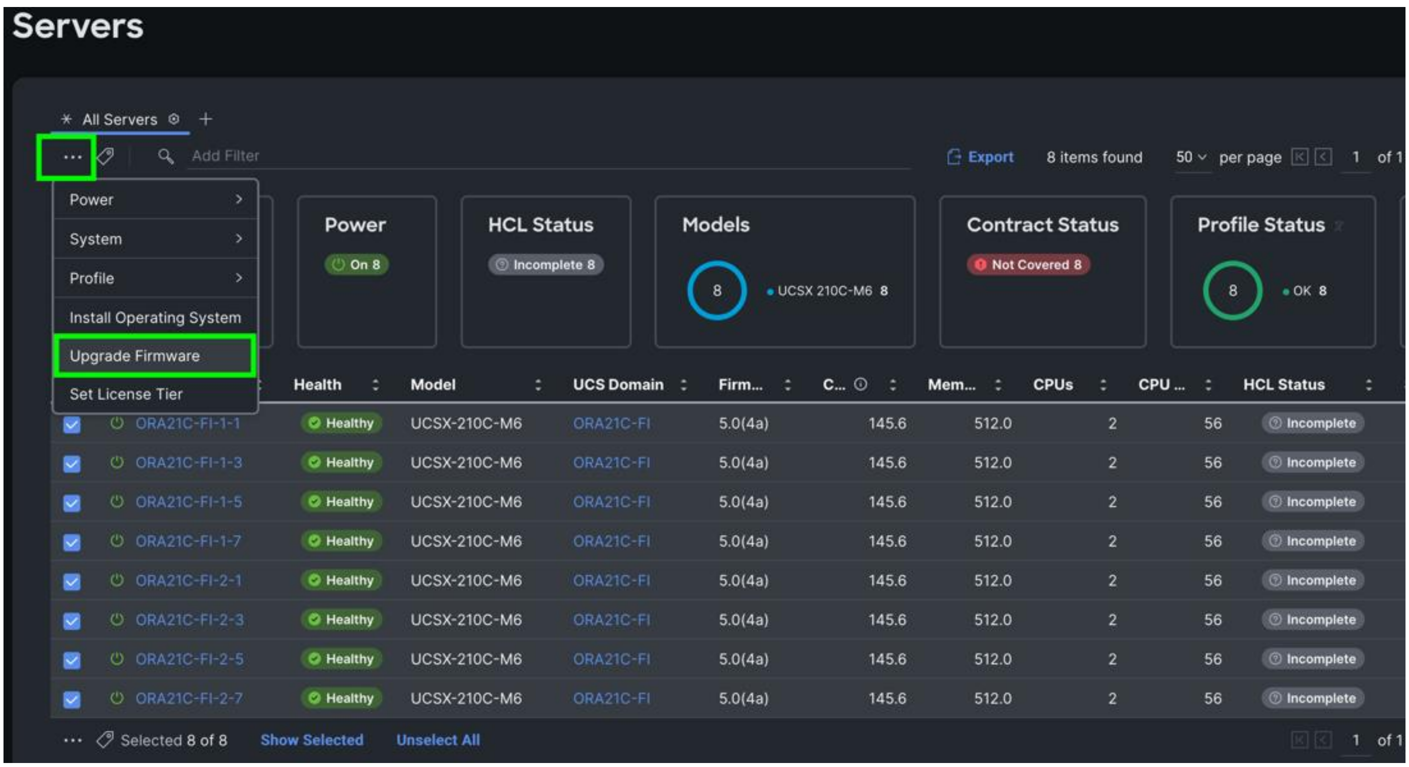
Intersight
Infrastructure Service
Search
Overview
Operate
Servers
Chassis
Fabric Interconnects
HyperFlex Clusters
Integrated Systems
Configure
← Chassis
ORA21C-FI-1
Healthy
Actions
General
Inventory
Connections
UCS Chassis Profile
Details
Health
Healthy
Name
ORA21C-FI-1
Serial
FOX2501P0C1
Model
UCSX-9508
Revision
0
Part Number
68-6847-03
Management Mode
Intersight
Contract Status
Not Covered
UCS Domain
ORA21C-FI
Chassis Profile
ORA-Chassis-1
Properties
UCSX-9508
Front
Rear
Locator LED
Health Overlay
States
Connection Details
Events
Alarms
No Alarms
No Alarms
Advisories
No Advisories



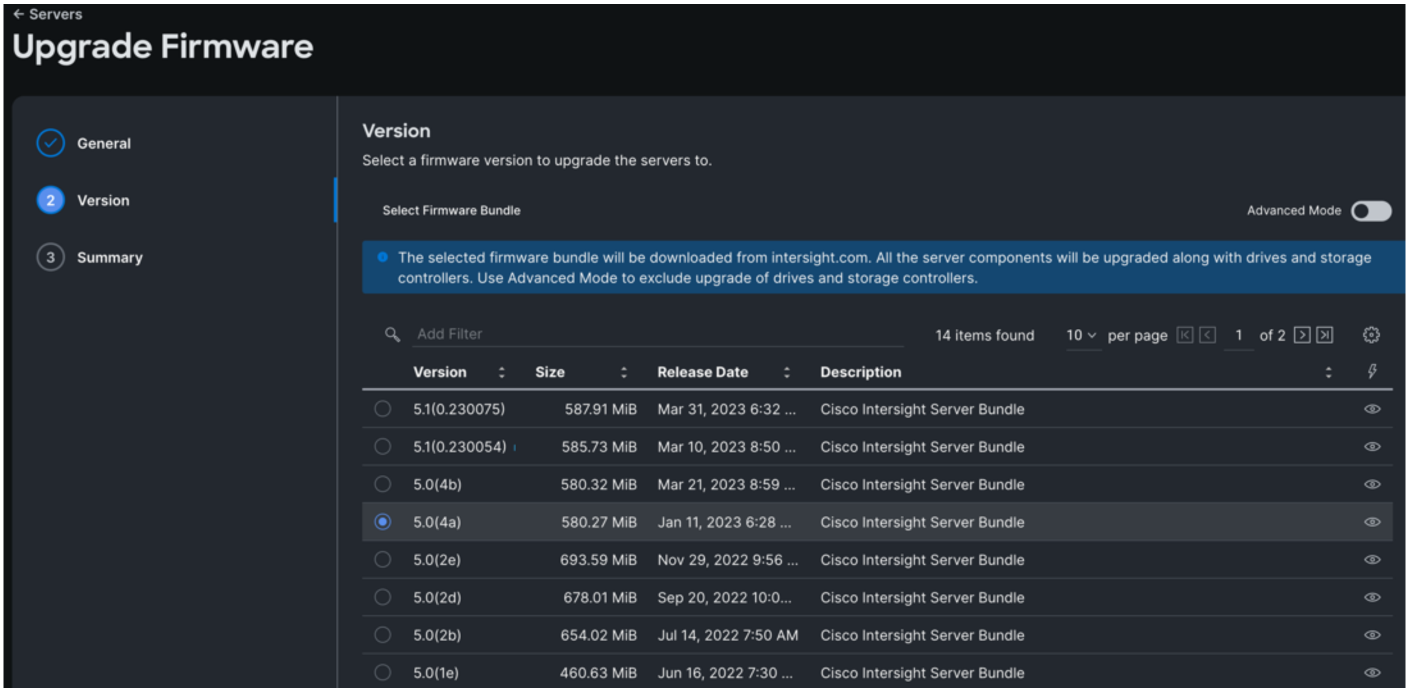
← Chassis
ORA21C-FI-1
Healthy
General
Inventory
Connections
UCS Chassis Profile
Intelligent Fabric Modules
X-Fabric Modules
Thermal
Power
Servers
Servers
Add Filter
Export
4 items found
10
per page
Name
Health
User Label
Slot Id
Model
ORA21C-FI-1-1
Healthy
1
UCSX-210C-M6
ORA21C-FI-1-3
Healthy
3
UCSX-210C-M6
ORA21C-FI-1-5
Healthy
5
UCSX-210C-M6
ORA21C-FI-1-7
Healthy
7
UCSX-210C-M6



Step 8. After discovering the servers successfully, upgrade all server firmware through IMM to the supported release. To do this, check the box for All Servers and then click the ellipses and from the drop-down list, select Upgrade Firmware.



Step 9. In the Upgrade Firmware section, select all servers and click Next. In the Version section, for the supported firmware version release select "5.0(4a)" and click Next, then click Upgrade to upgrade the firmware on all servers simultaneously.



After the successful firmware upgrade, you can create a server profile template and a server profile for IMM configuration.

Configure Server Profile Template

A server profile template enables resource management by simplifying policy alignment and server configuration. A server profile template is created using the server profile template wizard. The server profile template wizard groups the server policies into the following categories to provide a quick summary view of the policies that are attached to a profile:

- Compute Configuration: BIOS, Boot Order, and Virtual Media policies.
- Management Configuration: Certificate Management, IMC Access, IPMI (Intelligent Platform Management Interface) Over LAN, Local User, Serial Over LAN, SNMP (Simple Network Management Protocol), Syslog and Virtual KVM (Keyboard, Video, and Mouse).
- Storage Configuration: SD Card, Storage.
- Network Configuration: LAN connectivity and SAN connectivity policies.

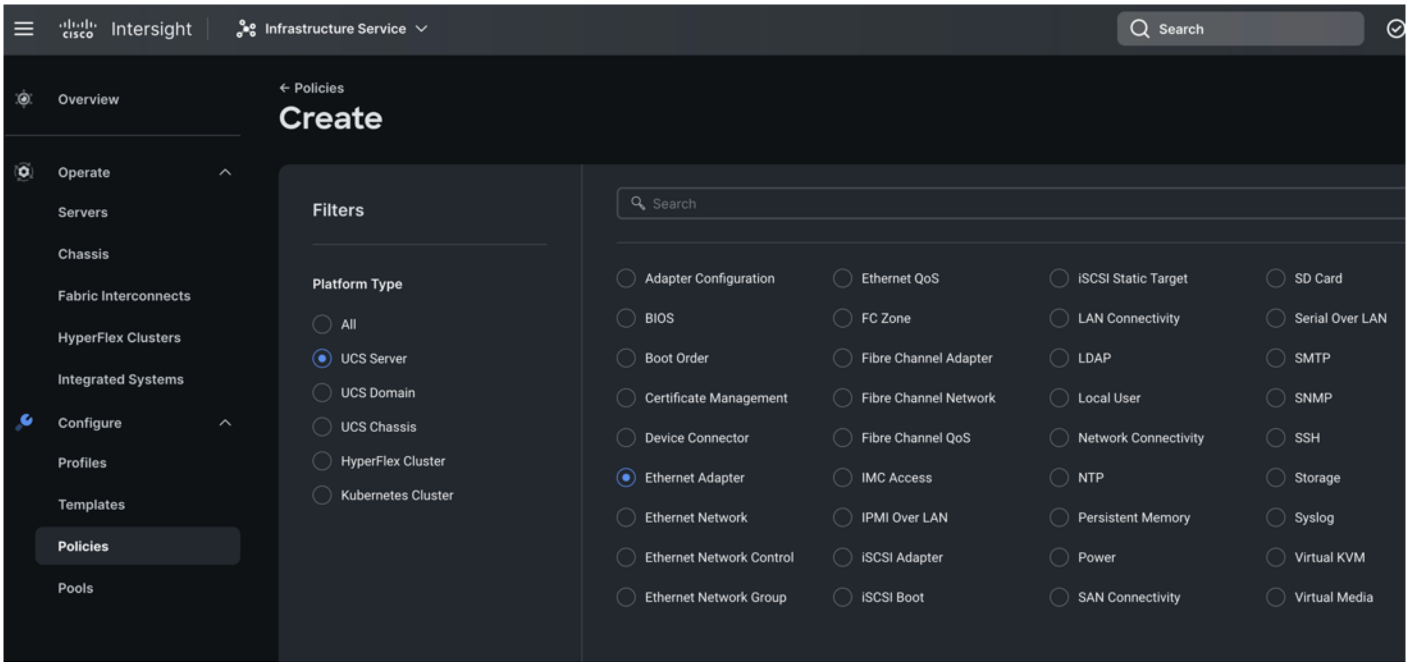
Some of the characteristics of the server profile template for FlexPod are as follows:

- BIOS policy is created to specify various server parameters in accordance with FlexPod best practices.
- Boot order policy defines virtual media (KVM mapper DVD) and local boot through virtual driver.
- IMC access policy defines the management IP address pool for KVM access.

- LAN connectivity policy is used to create six virtual network interface cards (vNICs) – One vNIC for Server Node Management and Public Network Traffic, second vNIC for Private Server-to-Server Network (Cache Fusion) Traffic Interface for Oracle RAC, four vNICs for Database IO Traffic to NetApp Storage Controller. Various policies and pools are also created for the vNIC configuration.

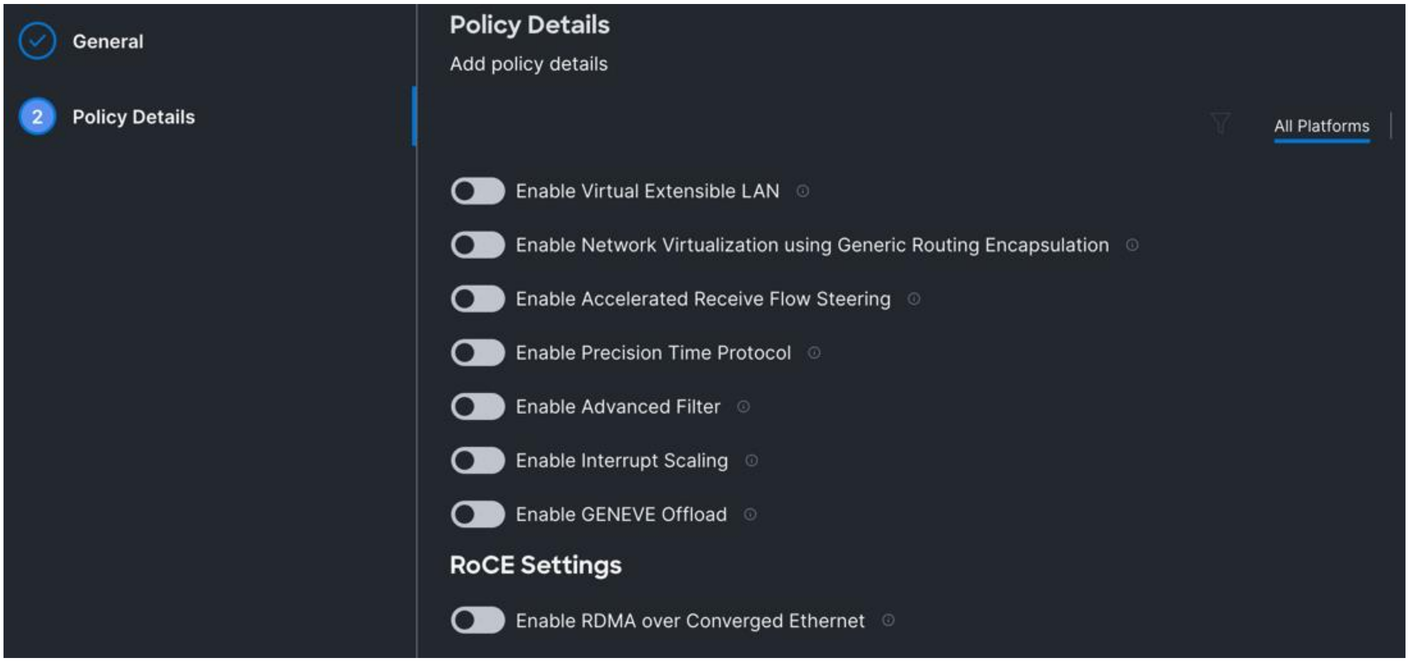
Procedure 1. Configure Adapter Policy

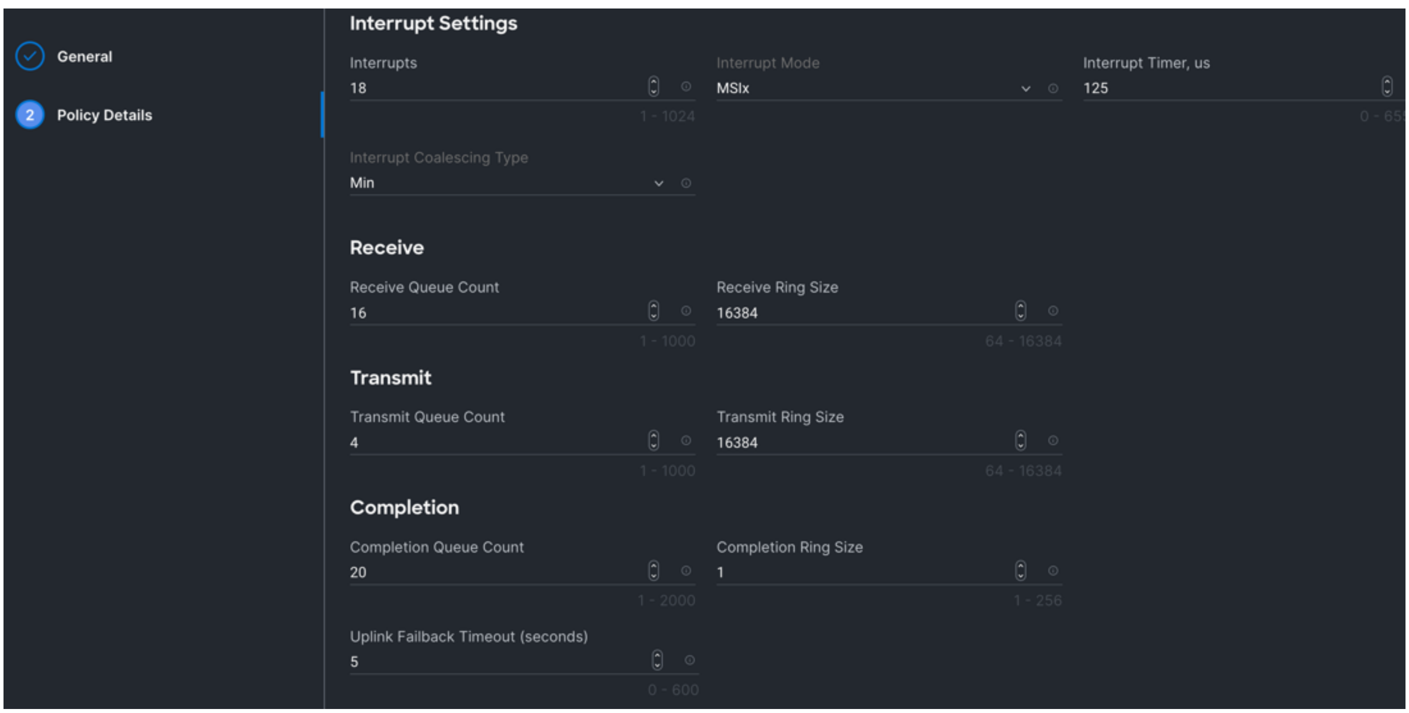
Step 1. To configure the Adapter Policy for the UCS Server profile, go to > Infrastructure Service > Configure > Polices > and click Create Policy. For the platform type select “UCS Server” and for the policy select “Ethernet Adapter.”

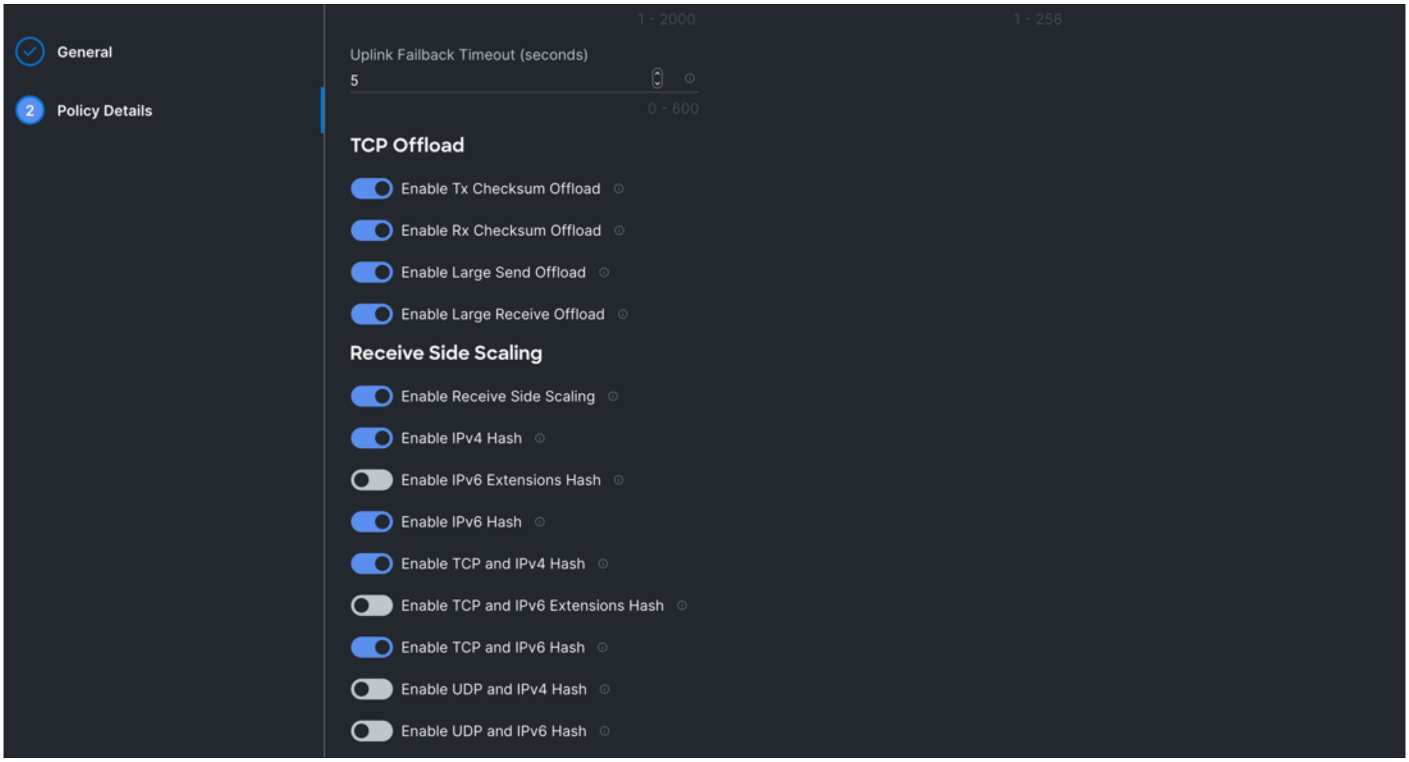


Step 2. In the Ethernet Adapter Configuration section, for the Organization select “ORA21” and for the policy name enter “ORA-Linux-Adapter.” click Next.

Step 3. In the Policy Details section, for the recommended performance on the ethernet adapter, keep the “Interrupt Settings” parameter.


General
2
Policy Details
Policy Details
Add policy details
All Platforms
Enable Virtual Extensible LAN
Enable Network Virtualization using Generic Routing Encapsulation
Enable Accelerated Receive Flow Steering
Enable Precision Time Protocol
Enable Advanced Filter
Enable Interrupt Scaling
Enable GENEVE Offload
RoCE Settings
Enable RDMA over Converged Ethernet


General
2
Policy Details
Interrupt Settings
Interrupts
18
1 - 1024
Interrupt Mode
MSIx
Interrupt Timer, us
125
Interrupt Coalescing Type
Min
Receive
Receive Queue Count
16
1 - 1000
Receive Ring Size
16384
64 - 16384
Transmit
Transmit Queue Count
4
1 - 1000
Transmit Ring Size
16384
64 - 16384
Completion
Completion Queue Count
20
1 - 2000
Completion Ring Size
1
1 - 256
Uplink Failback Timeout (seconds)
5
0 - 600



Step 4. Click Create to create this policy.

Procedure 2. Configure LAN Connectivity Policy

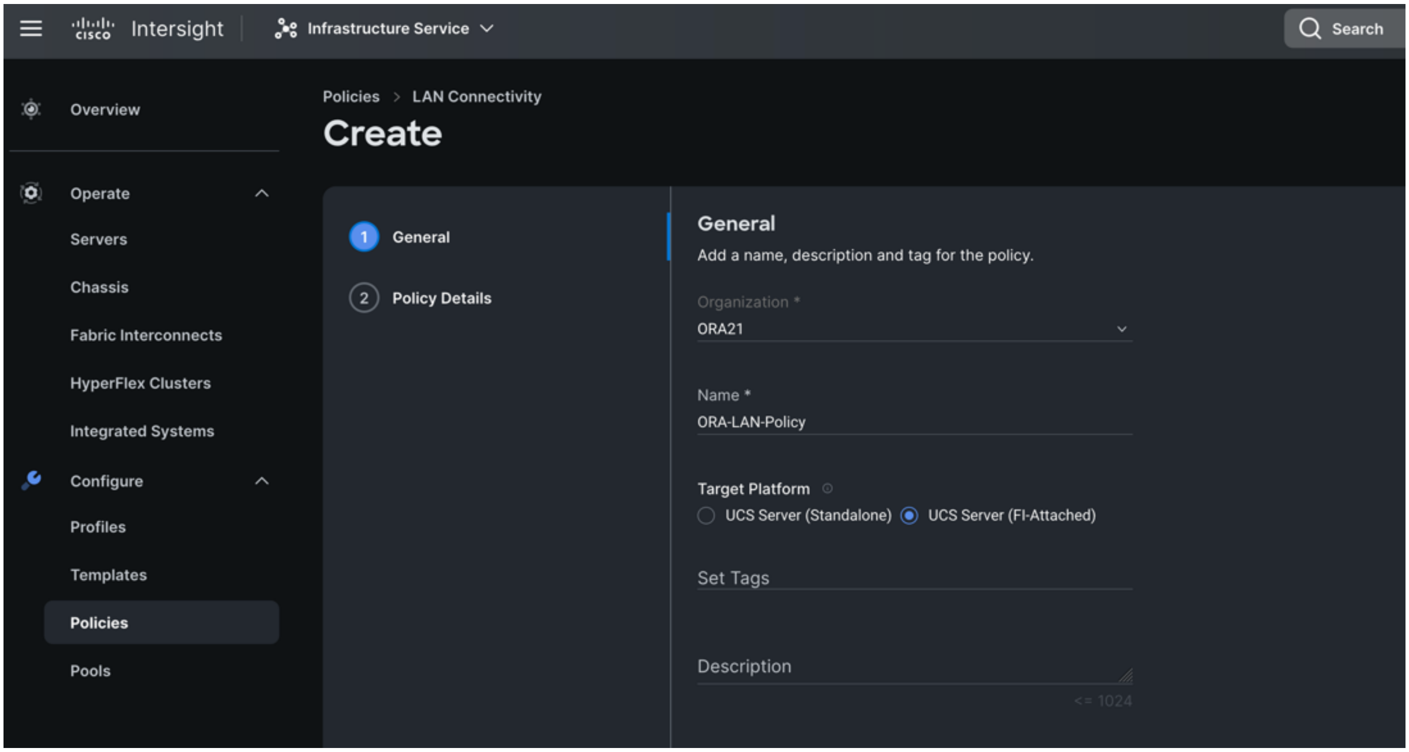
Six vNICs were configured per server as shown in Table 10.

Table 10. Configured VNICs

Name	Switch ID	PCI-Order	MAC Pool	Fail-Over
vNIC0	FI - A	0	ORA-MAC-A	Enabled
vNIC1	FI - B	1	ORA-MAC-B	Enabled
vNIC2	FI - A	2	ORA-MAC-Storage-A	Enabled
vNIC3	FI - B	3	ORA-MAC-Storage-B	Enabled
vNIC4	FI - A	4	ORA-MAC-Storage-A	Enabled
vNIC5	FI - B	5	ORA-MAC-Storage-B	Enabled

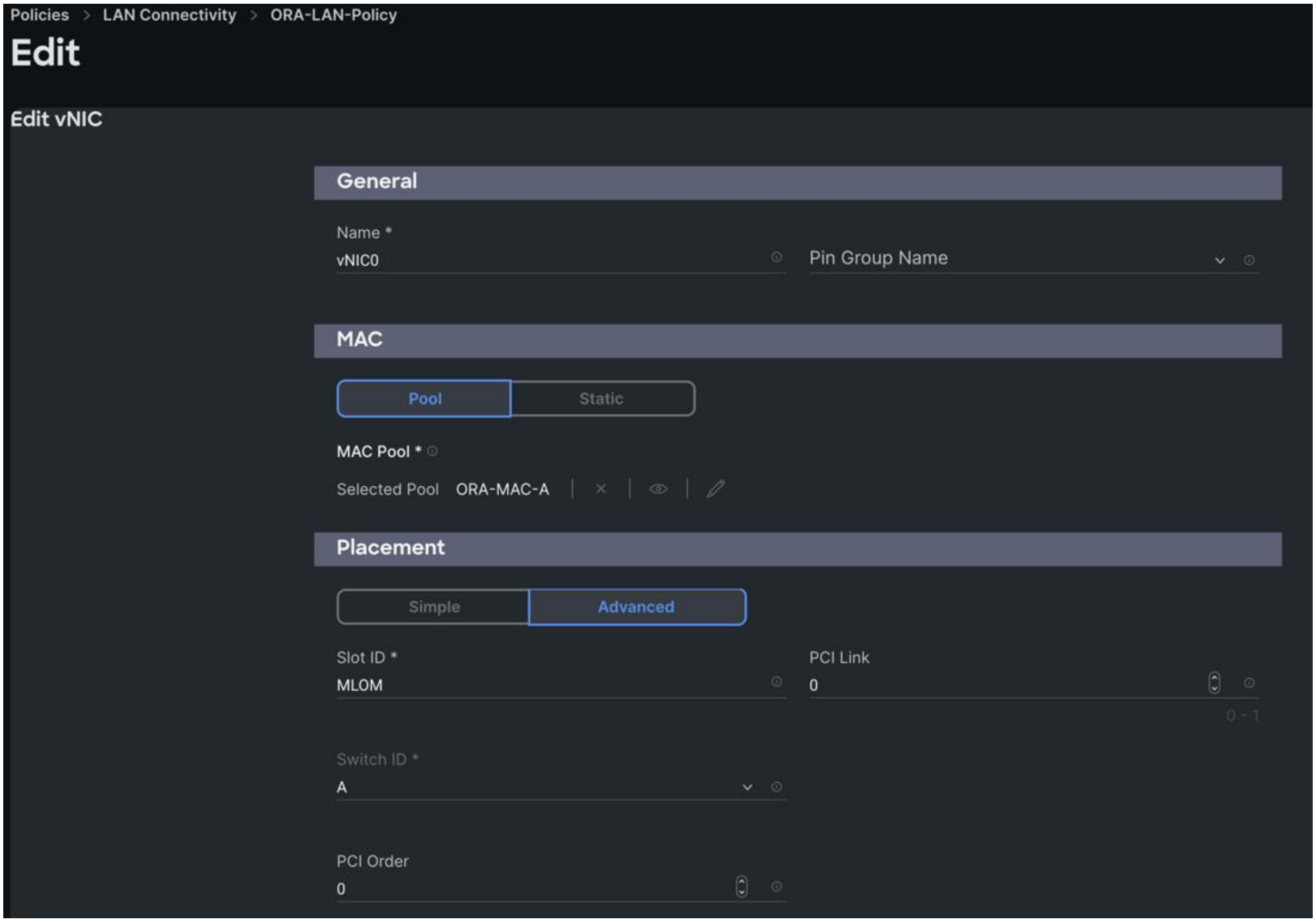
Step 1. To configure the LAN Connectivity Policy for the UCS Server profile, go to > Infrastructure Service > Configure > Polices > and click Create Policy. For the platform type select “UCS Server” and for the policy select “LAN Connectivity.”

Step 2. In the LAN Connectivity Policy Create section, for the Organization select “ORA21”,for the policy name enter “ORA-LAN-Policy” and for the Target Platform select UCS Server (FI-Attached). Click Next.

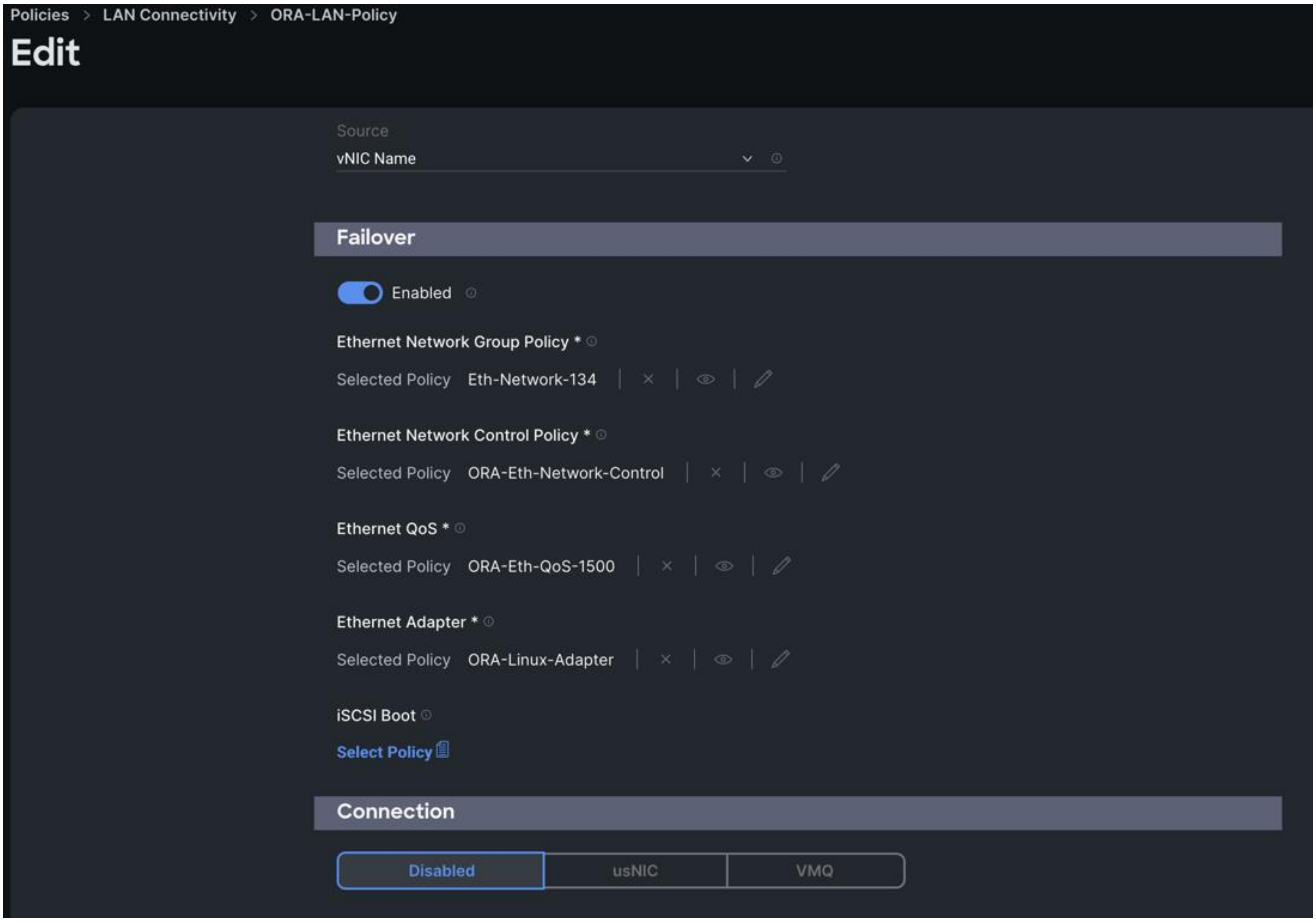


Step 3. In the Policy Details section, click Add vNIC. In the Add vNIC section, for the first vNIC enter vNIC0. In the Edit vNIC section, for the vNIC name enter " vNIC0" and for the MAC Pool select " ORA-MAC-A."

Step 4. In the Placement option, click Advanced and for the Slot ID enter " MLOM" , for the Switch ID select " A" and for the PCI Order select " 0" .



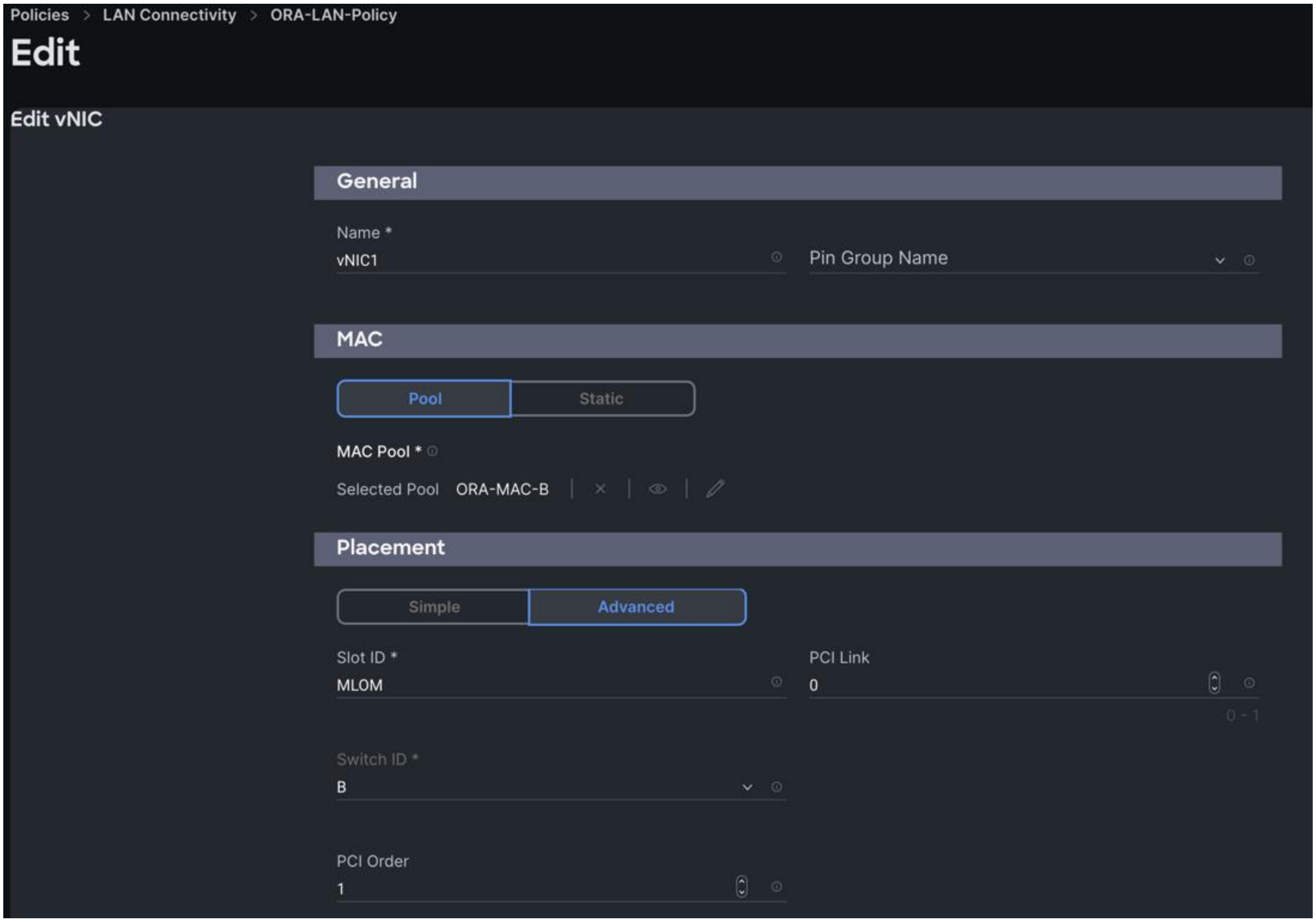
Step 5. For Failover select Enable for this vNIC configuration. This enables the vNIC to failover to another FI.



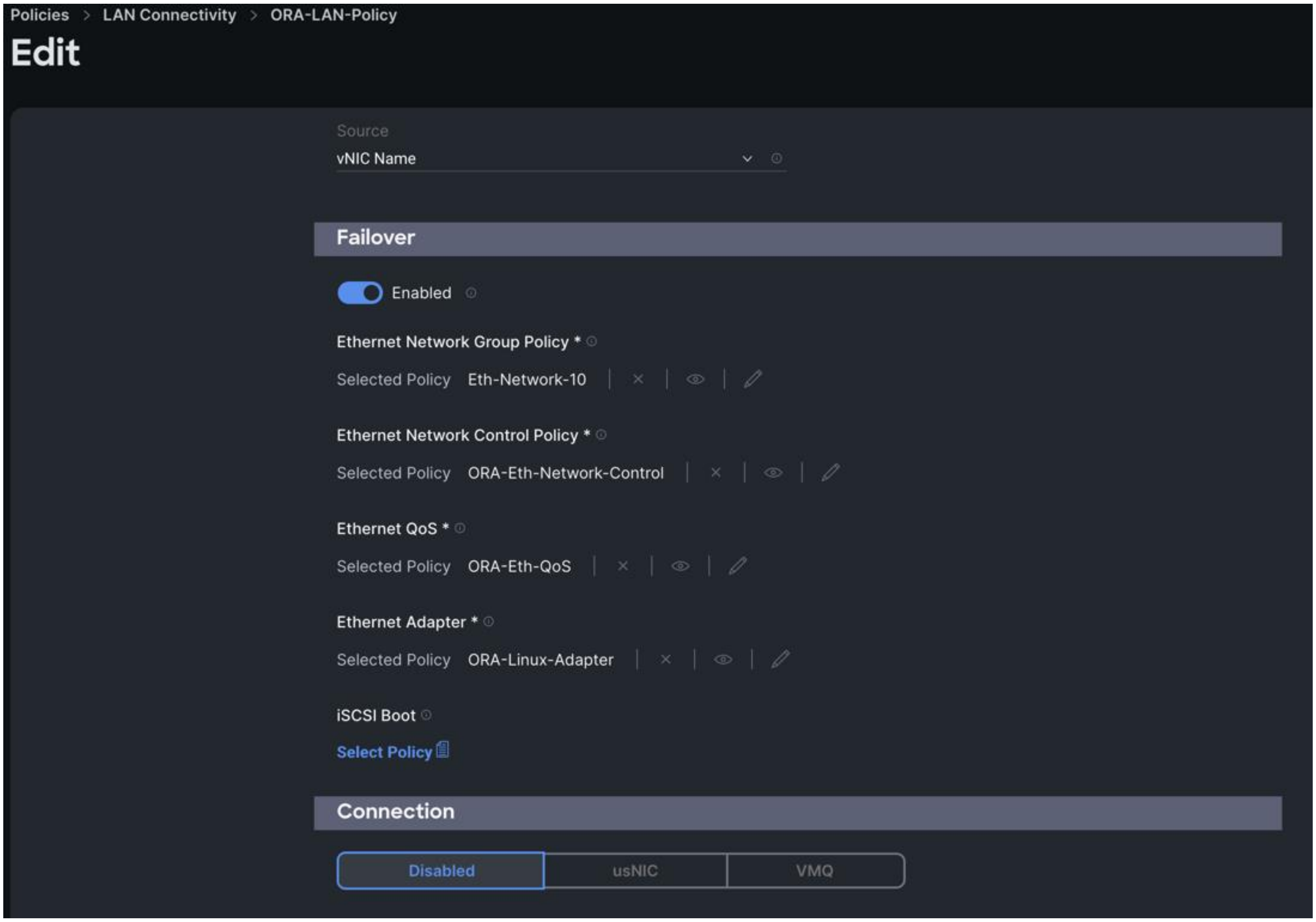
Step 6. Select the Ethernet Network Group Policy (Eth-Network-134), Ethernet Network Control Policy, Ethernet QoS, and Ethernet Adapter. Click Add to add vNIC0 into this policy.

Step 7. Add a second vNIC. For the name enter " vNIC1" and for the MAC Pool select " ORA-MAC-B."

Step 8. In the Placement option, click Advanced and for the Slot ID enter " MLOM" , for the Switch ID select " B" and for the PCI Order select " 1."



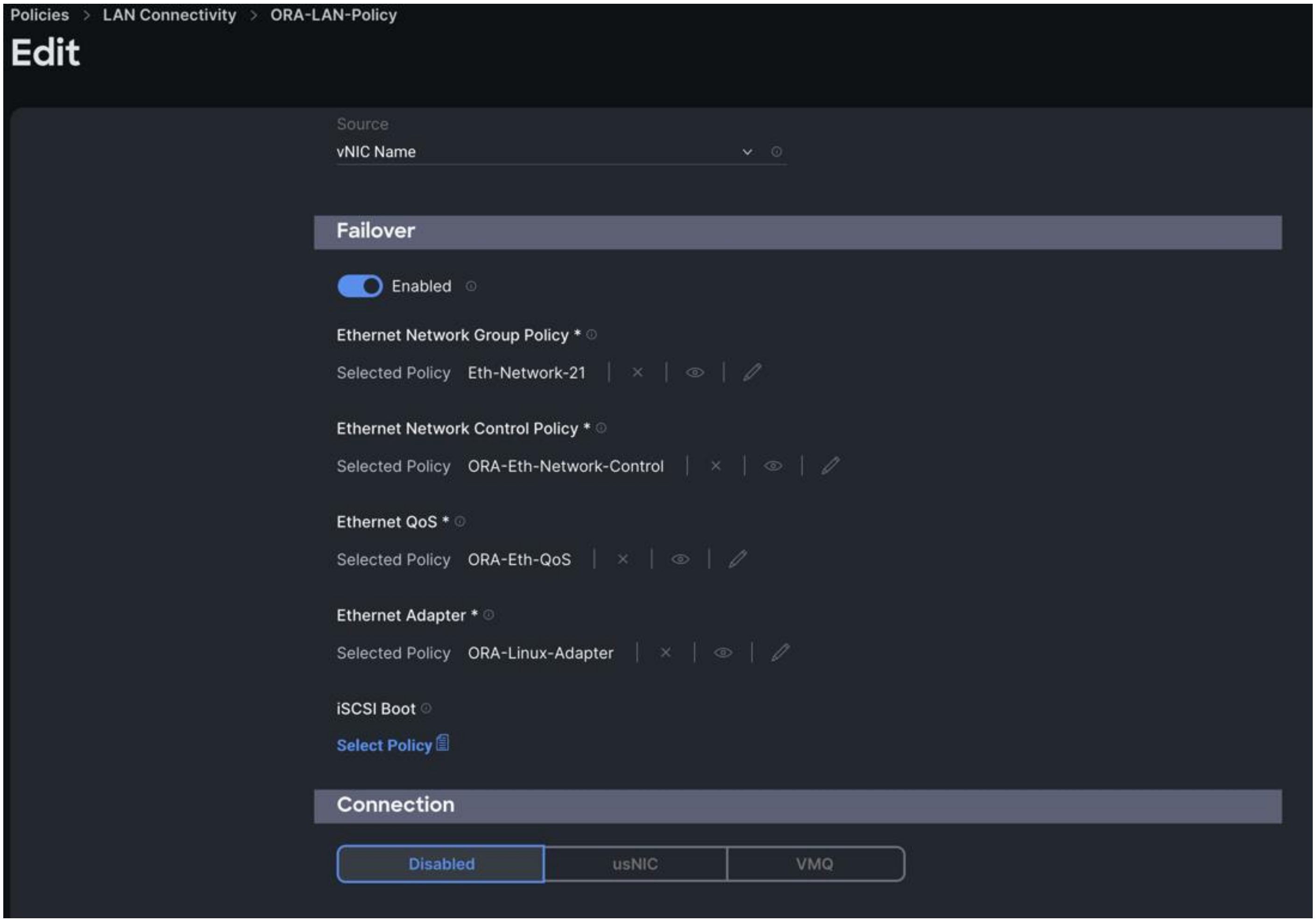
Step 9. For Failover select Enable for this vNIC configuration. This enables the vNIC to failover to another FI. Select Ethernet Network Group Policy (Eth-Network-10), Ethernet Network Control Policy, Ethernet QoS, and Ethernet Adapter.



Step 10. Click Add to add vNIC1 into this policy.

Step 11. Add a third vNIC. For the name enter " vNIC2" and for the MAC Pool select " ORA-MAC-Storage-A" . In the Placement option, click Advanced and for the Slot ID select " MLOM" , for the Switch ID select " A" and for the PCI Order select " 2" .

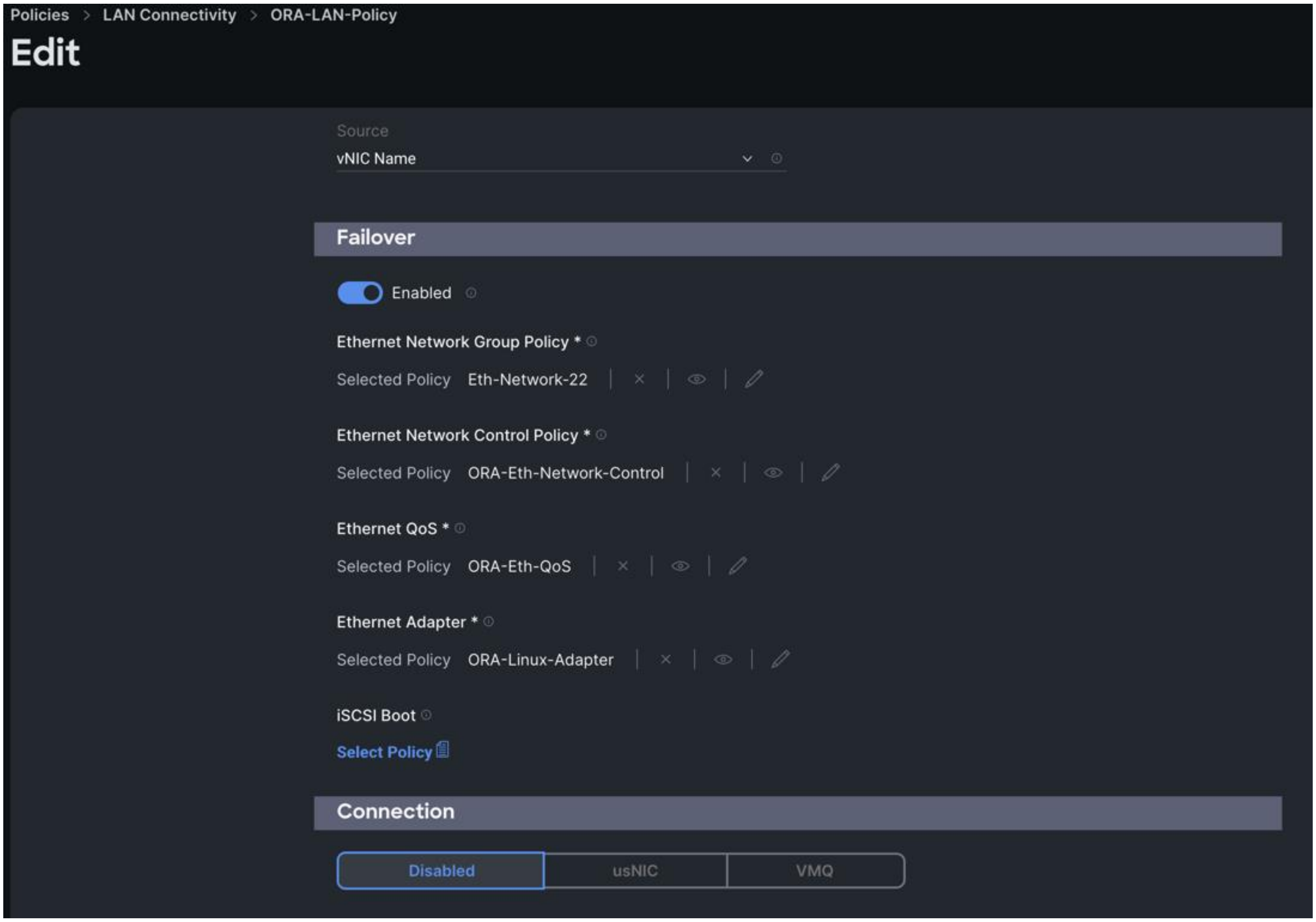
Step 12. Enable Failover for this vNIC configuration. Select Ethernet Network Group Policy (Eth-Network-21), Ethernet Network Control Policy, Ethernet QoS, and Ethernet Adapter.



Step 13. Click Add to add vNIC2 into this policy.

Step 14. Add a fourth. For the name enter " vNIC3" and for the MAC Pool select " ORA-MAC-Storage-B" . In the Placement option, click Advanced, and for the Slot ID select " MLOM" , for the Switch ID select " B" and for the PCI Order select " 3" .

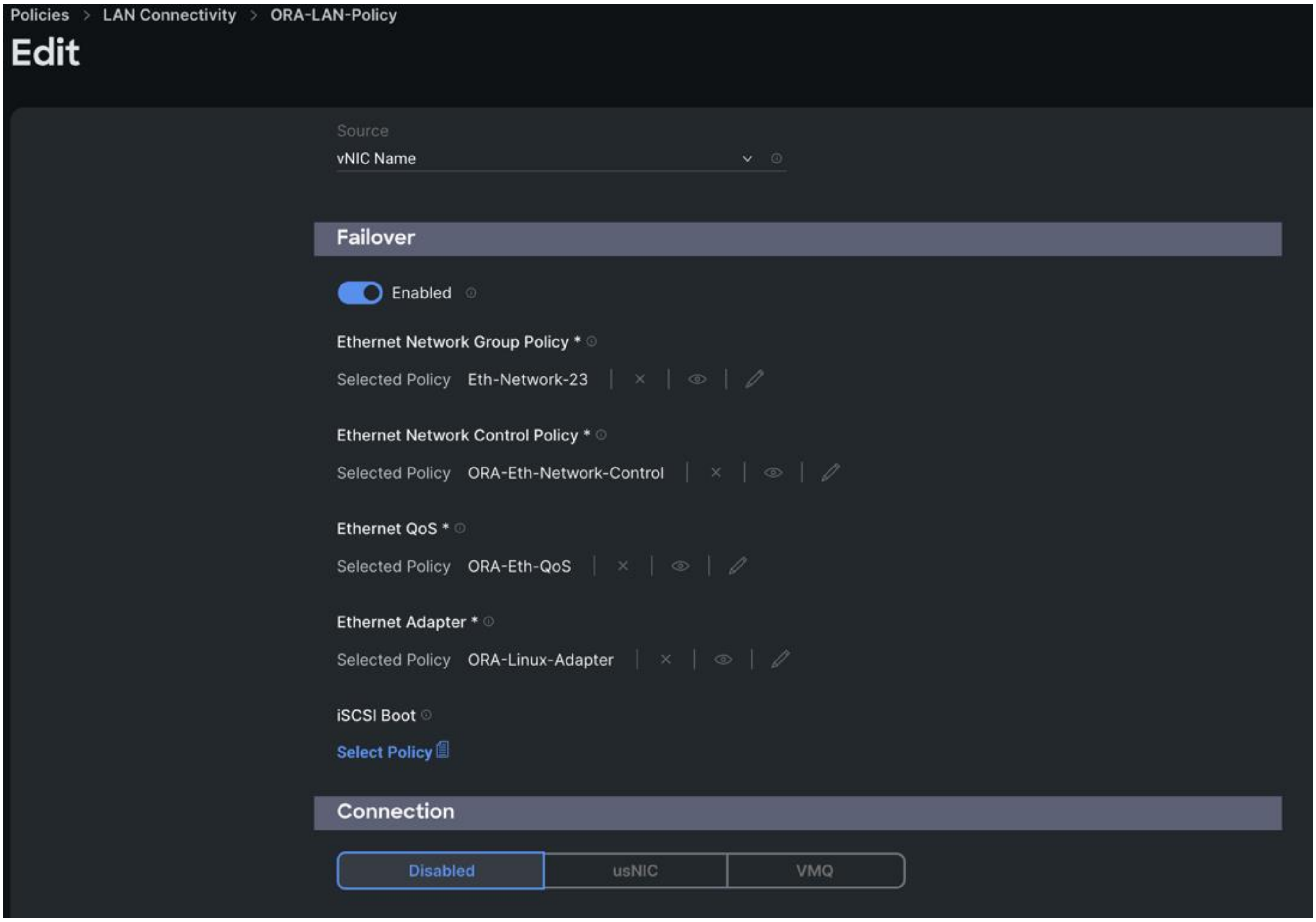
Step 15. Enable Failover for this vNIC configuration. Select Ethernet Network Group Policy (Eth-Network-22), Ethernet Network Control Policy, Ethernet QoS, and Ethernet Adapter.



Step 16. Click Add to add vNIC3 into this policy.

Step 17. Add a fifth vNIC. For the name enter "vNIC4" and for the MAC Pool select "ORA-MAC-Storage-A". In the Placement option, click Advanced and for the Slot ID select "MLOM", for the Switch ID select "A" and fort the PCI Order select "4".

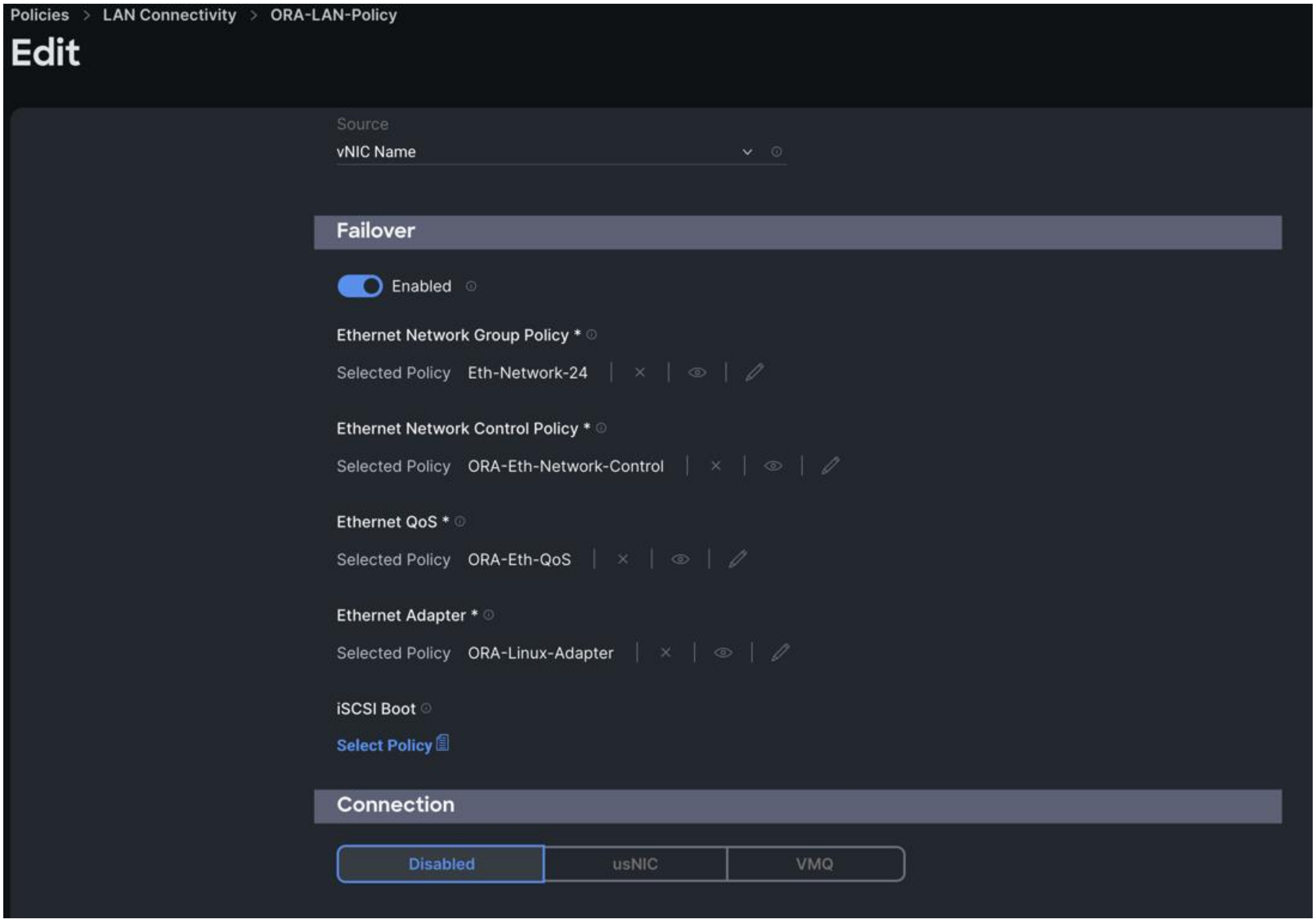
Step 18. Enable Failover for this vNIC configuration. Select Ethernet Network Group Policy (Eth-Network-23), Ethernet Network Control Policy, Ethernet QoS, and Ethernet Adapter.



Step 19. Click Add to add vNIC4 into this policy.

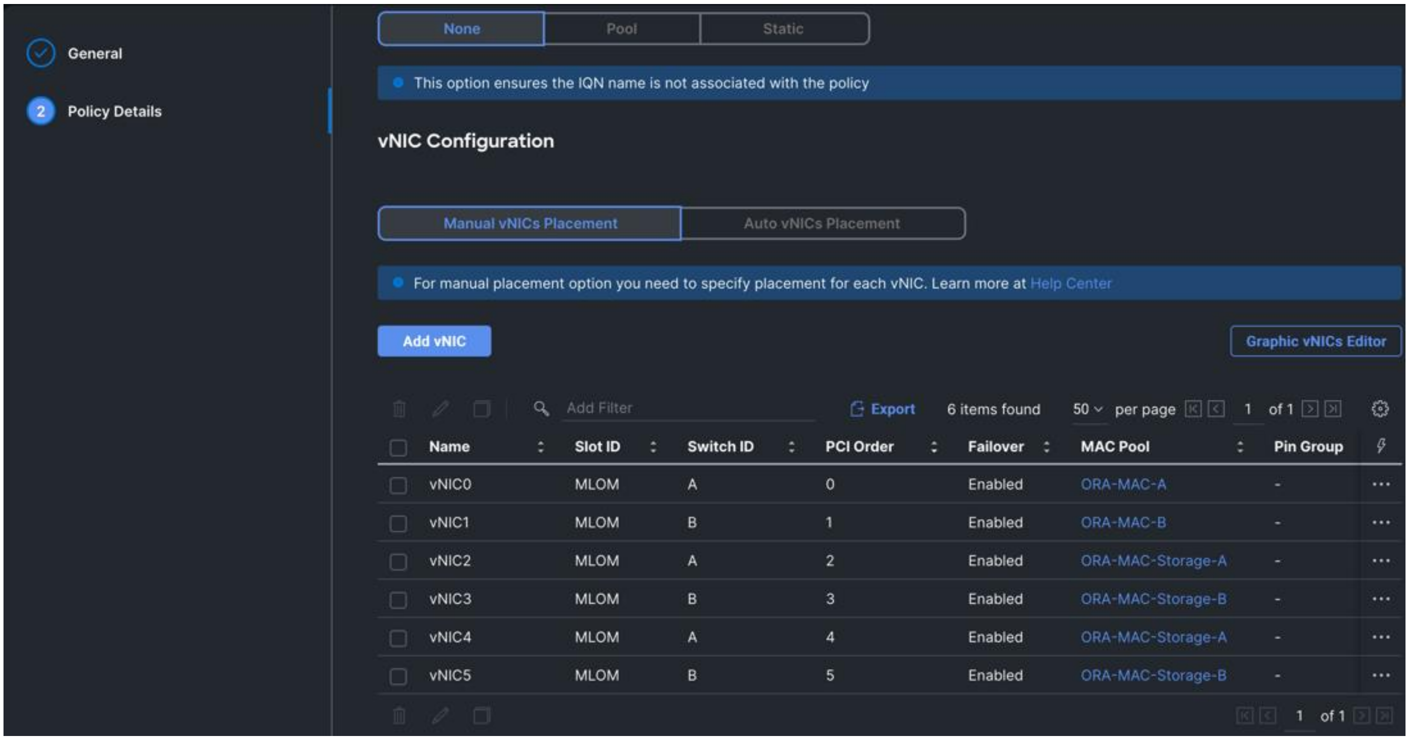
Step 20. Add a sixth vNIC. For the name enter "vNIC5" and for the MAC Pool select "ORA-MAC-Storage-B". In the Placement option, click Advanced and for the Slot ID select "MLOM", for the Switch ID select "B" and for the PCI Order select "5".

Step 21. Enable Failover for this vNIC configuration. Select Ethernet Network Group Policy (Eth-Network-24), Ethernet Network Control Policy, Ethernet QoS, and Ethernet Adapter.



Step 22. Click Add to add vNIC5 into this policy.

Step 23. After adding these vNICs, review and make sure the Switch ID, PCI Order, Failover Enabled and MAC Pool are as shown below.



Step 24. Click Create to create this policy.

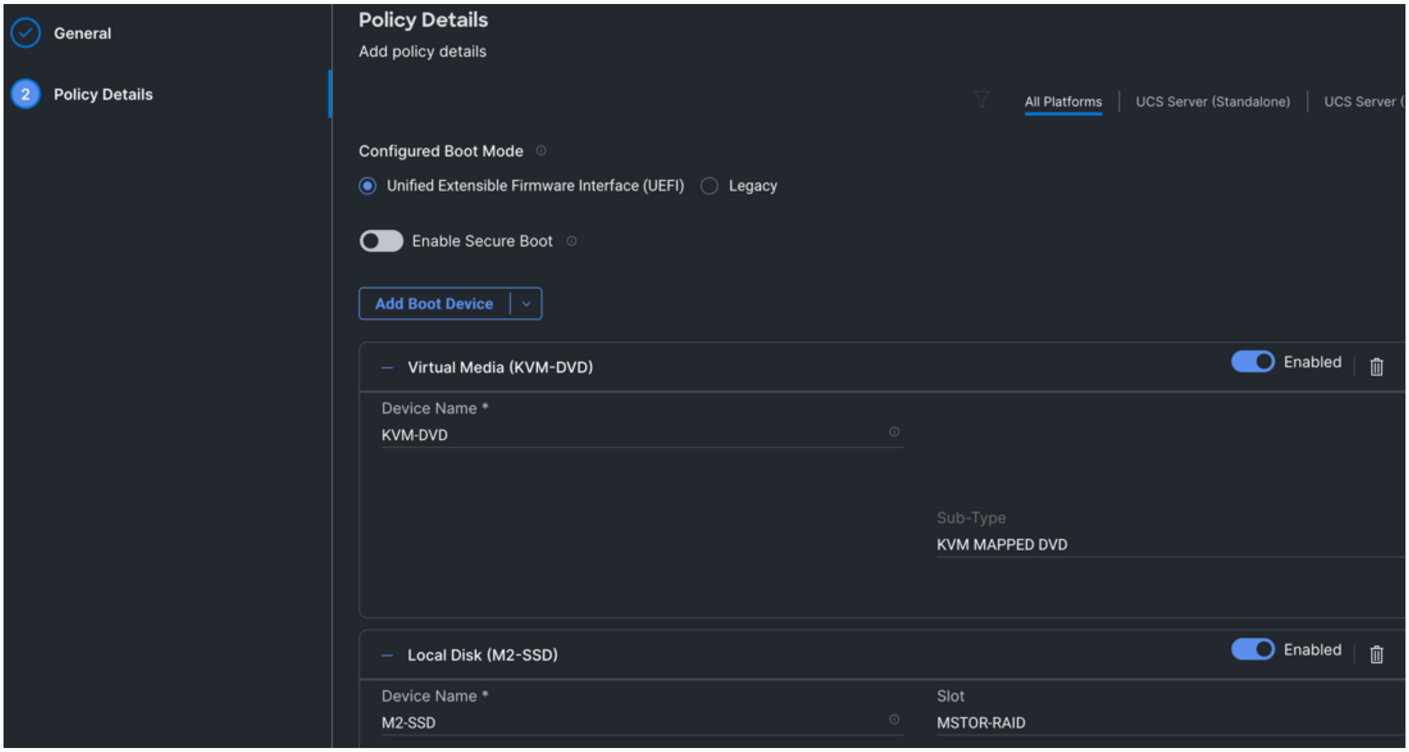
Procedure 3. Configure Boot Order Policy

For this solution, two local server nodes M.2 SSD were used, and the virtual drive was configured to install the OS locally on each node.

Step 1. To configure Boot Order Policy for UCS Server profile, go to > Infrastructure Service > Configure > Polices > and click Create Policy. For the platform type select "UCS Server" and for the policy select "Boot Order."

Step 2. In the Boot Order Policy Create section, for the Organization select "ORA21" and for the name of the Policy select "Local-Boot." Click Next.

Step 3. In the Policy Details section, click Add Boot Device and for the boot order add "Virtual Media" (KVM-DVD) and "Local Disk" (M2-SSD) as shown below.

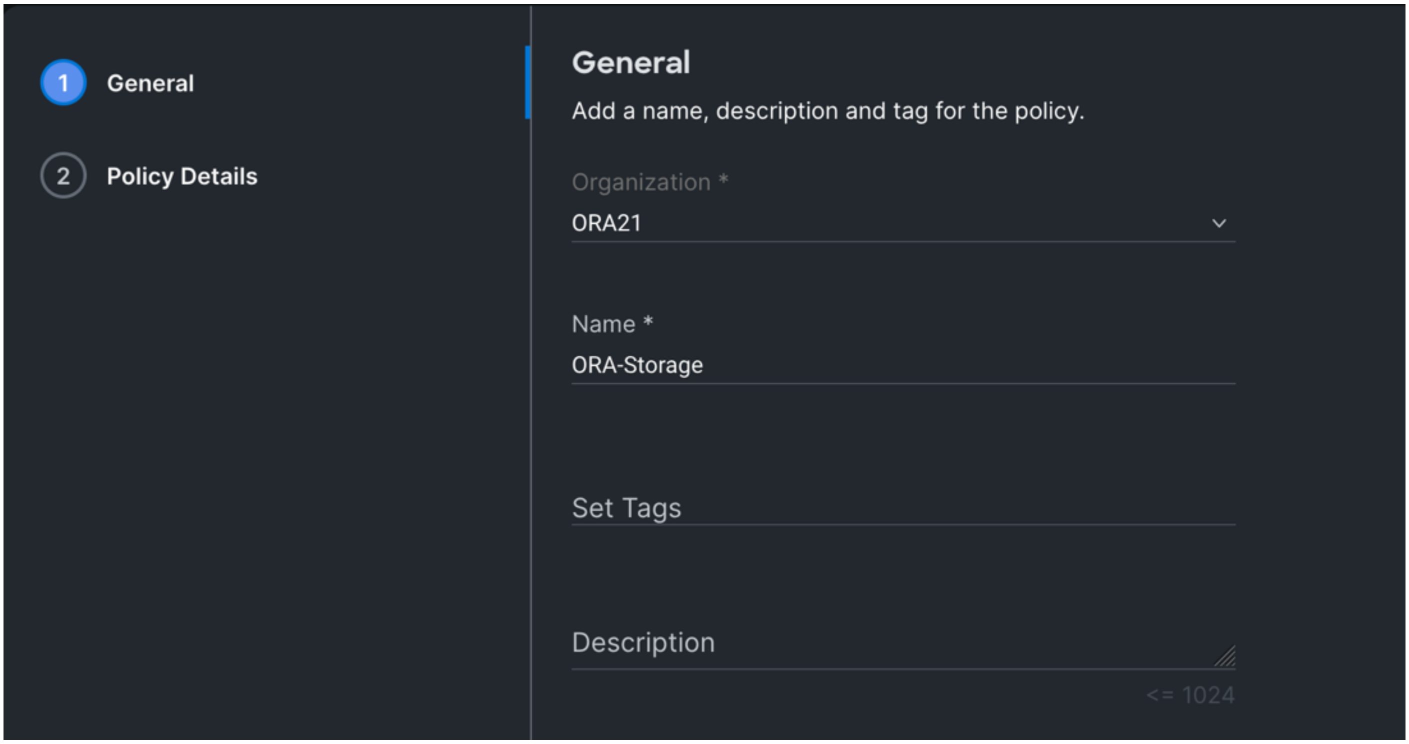


Step 4. Click Create to create this policy.

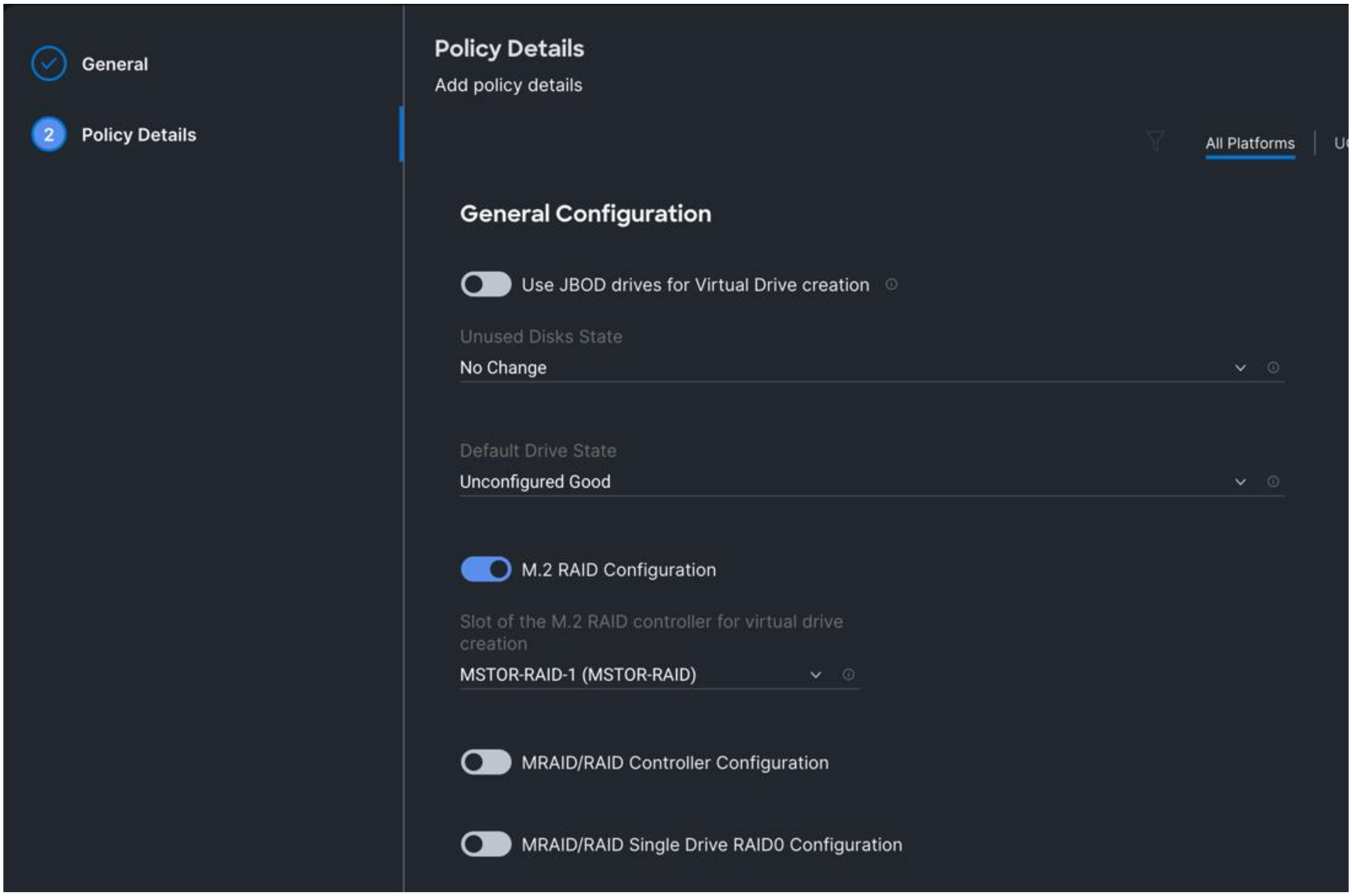
Procedure 4. Configure Storage Policy

Step 1. To configure the Storage Policy for the UCS Server profile, go to > Infrastructure Service > Configure > Polices > and click Create Policy. For the platform type select "UCS Server" and for the policy select "Storage."

Step 2. In the Storage Policy Create section, for the Organization select "ORA21" and for the policy name select "ORA-Storage." Click Next.



Step 3. In the Policy Details section, enable “M.2 RAID” and select the slot for the M.2 RAID controller for virtual driver creation.

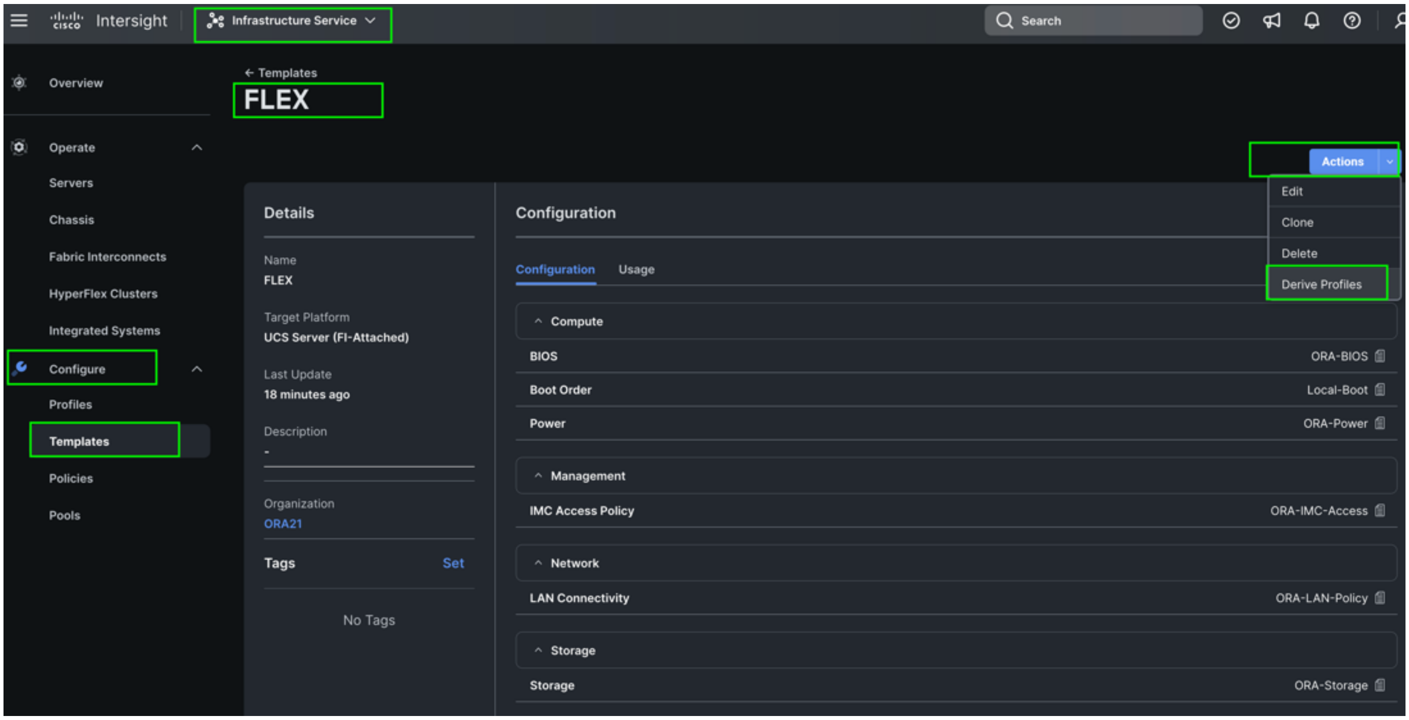


Step 4. Click Create to create this policy. You will use these policies while configuring the server profile template and the server profile as explained in the next section.

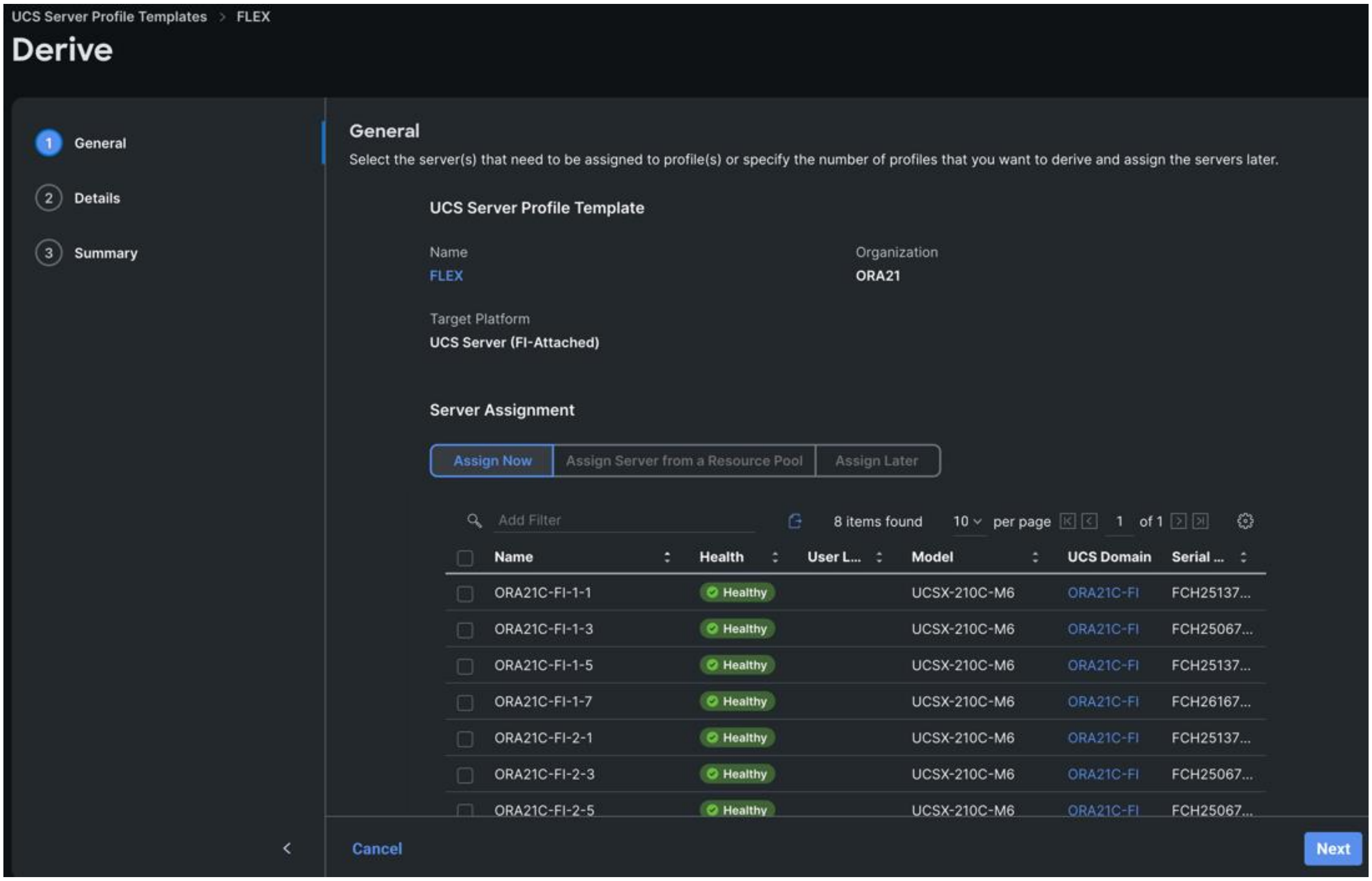
Derive and Deploy Server Profile from Server Profile Template

During the initial configuration for the management mode, the configuration wizard enables you to choose whether to manage the fabric interconnect.

The Cisco Intersight server profile allows server configurations to be deployed directly on the compute nodes based on polices defined in the server profile template. After a server profile template has been successfully created, server profiles can be derived from the template and associated with the Cisco UCS X210c M6 Compute Nodes, as shown below:

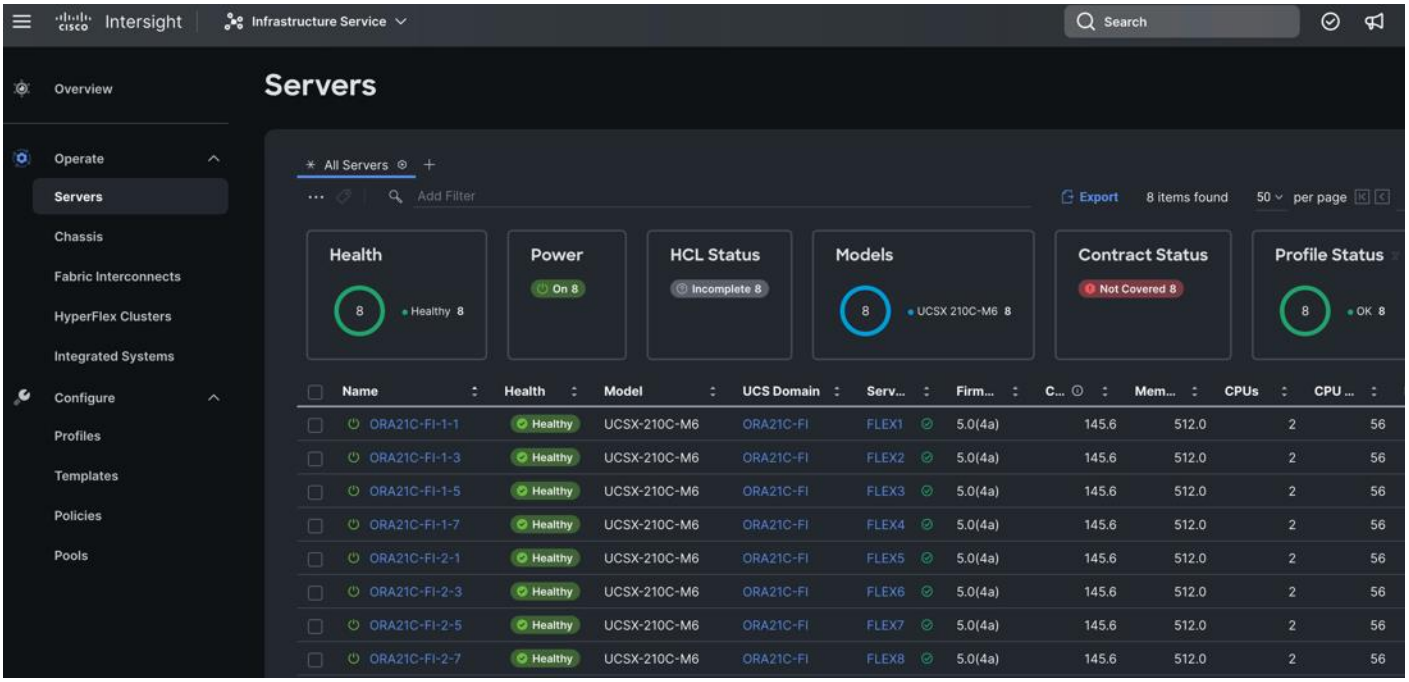


Select all eight servers from the chassis by clicking the checkbox and name the server profile “FLEX1” to “FLEX8” for all eight server nodes.



Note: For this solution, we configured eight server profile as FLEX1 to FLEX8. We assigned the server profile FLEX1 to Chassis 1 Server 1, server profile FLEX2 to Chassis 1 Server 3, server profile FLEX3 to Chassis 1 Server 5 and server profile FLEX4 to Chassis 1 Server 7. We also assigned server profile FLEX5 to Chassis 2 Server 1, server profile FLEX6 to Chassis 2 Server 3, server profile FLEX7 to Chassis 2 Server 5 and server profile FLEX8 to Chassis 2 Server 7.

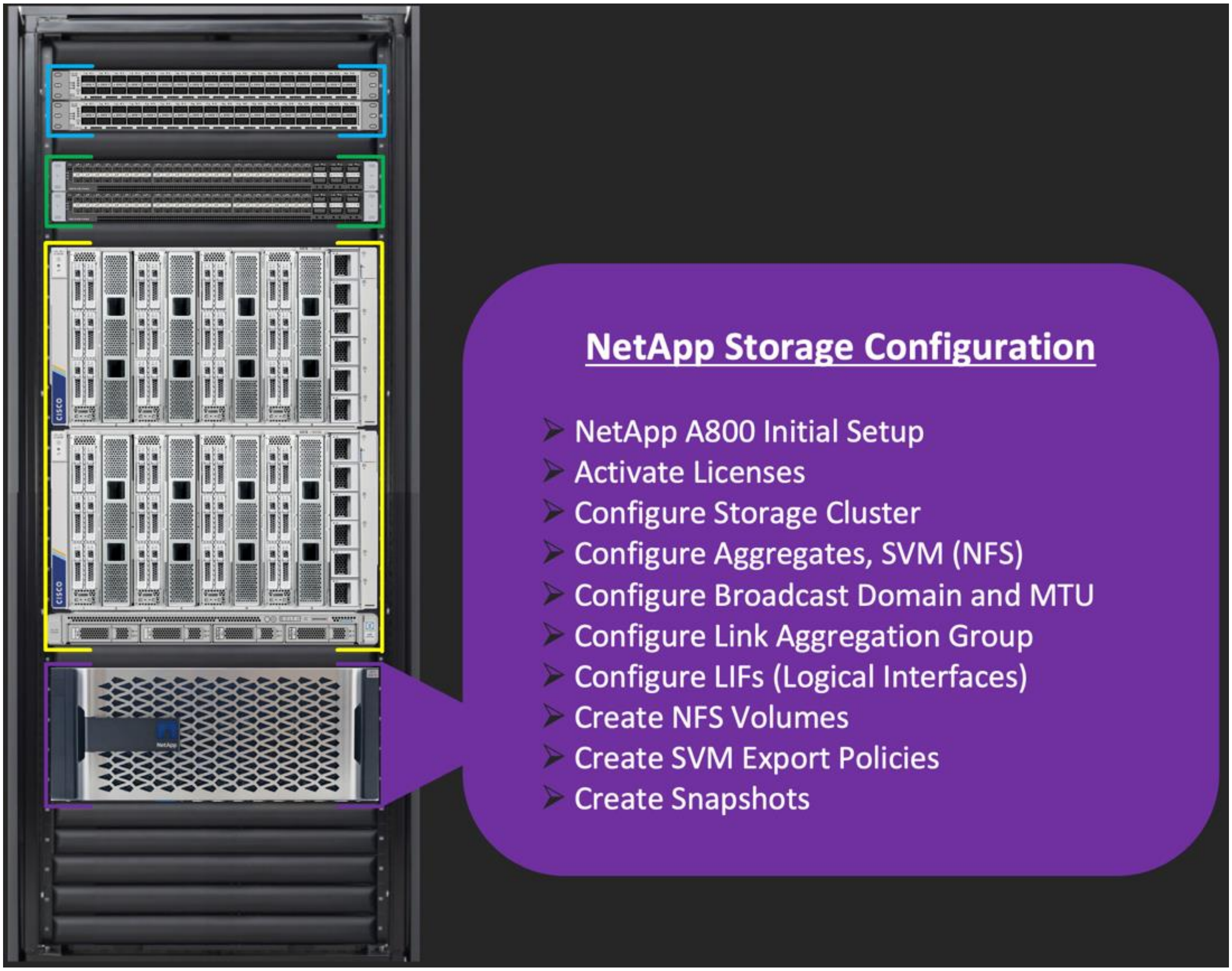
The following screenshot shows the server profile with the Cisco UCS domain and assigned servers from both chassis:



After the successful deployment of the server profile, the Cisco UCS X210c M6 Compute Nodes are configured with the parameters defined in the server profile. This completed Cisco UCS X-Series and Intersight Managed Mode (IMM) configuration can boot each server node from local virtual drive.

NetApp AFF A800 Storage Configuration

This section details the high-level steps to configure the NetApp Storage for this solution.



NetApp Storage Connectivity

Note: It is beyond the scope of this document to explain the detailed information about the NetApp storage connectivity and infrastructure configuration. For installation and setup instruction for the NetApp AFF A800 System, go to: https://docs.netapp.com/us-en/ontap-systems/a800/index.html

For more information, go to the Cisco site: https://www.cisco.com/c/en/us/solutions/design-zone/data-center-design-guides/flexpod-design-guides.html

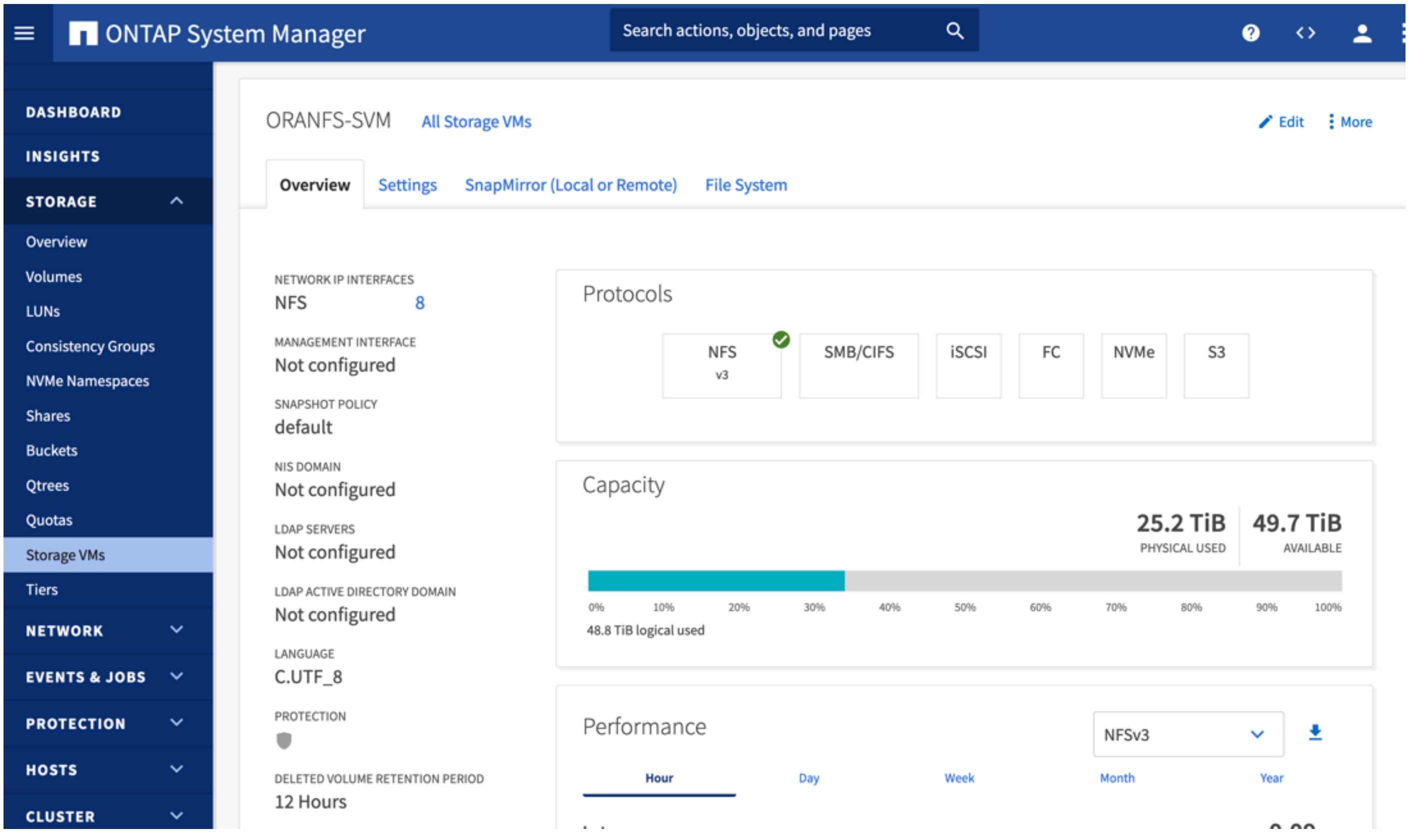
This section describes the storage layout and design considerations for the storage and database deployment. For all the database deployment, two aggregates (one aggregate on each storage node) were configured, and each aggregate contains 12 SSD (1.75 TB each) drives that were subdivided into RAID DP groups as shown below.

```
FlexPod-A800::> agg show
  (aggr)


Aggregate     Size Available Used% State   #Vols  Nodes            RAID Status
--------- -------- --------- ----- ------- ------ ---------------- ------------
aggr0_flexpod_ora_ntap_1_0 159.9GB 7.66GB 95% online 1
                                                  FlexPod-A800-CT1 raid_dp,
                                                                   normal
aggr0_flexpod_ora_ntap_2_0 159.9GB 7.66GB 95% online 1
                                                  FlexPod-A800-CT2 raid_dp,
                                                                   normal
aggr1_node1 16.29TB   5.81TB   64% online      44 FlexPod-A800-CT1 raid_dp,
                                                                   normal
aggr1_node2 16.29TB   7.04TB   57% online      39 FlexPod-A800-CT2 raid_dp,
                                                                   normal
4 entries were displayed.
```

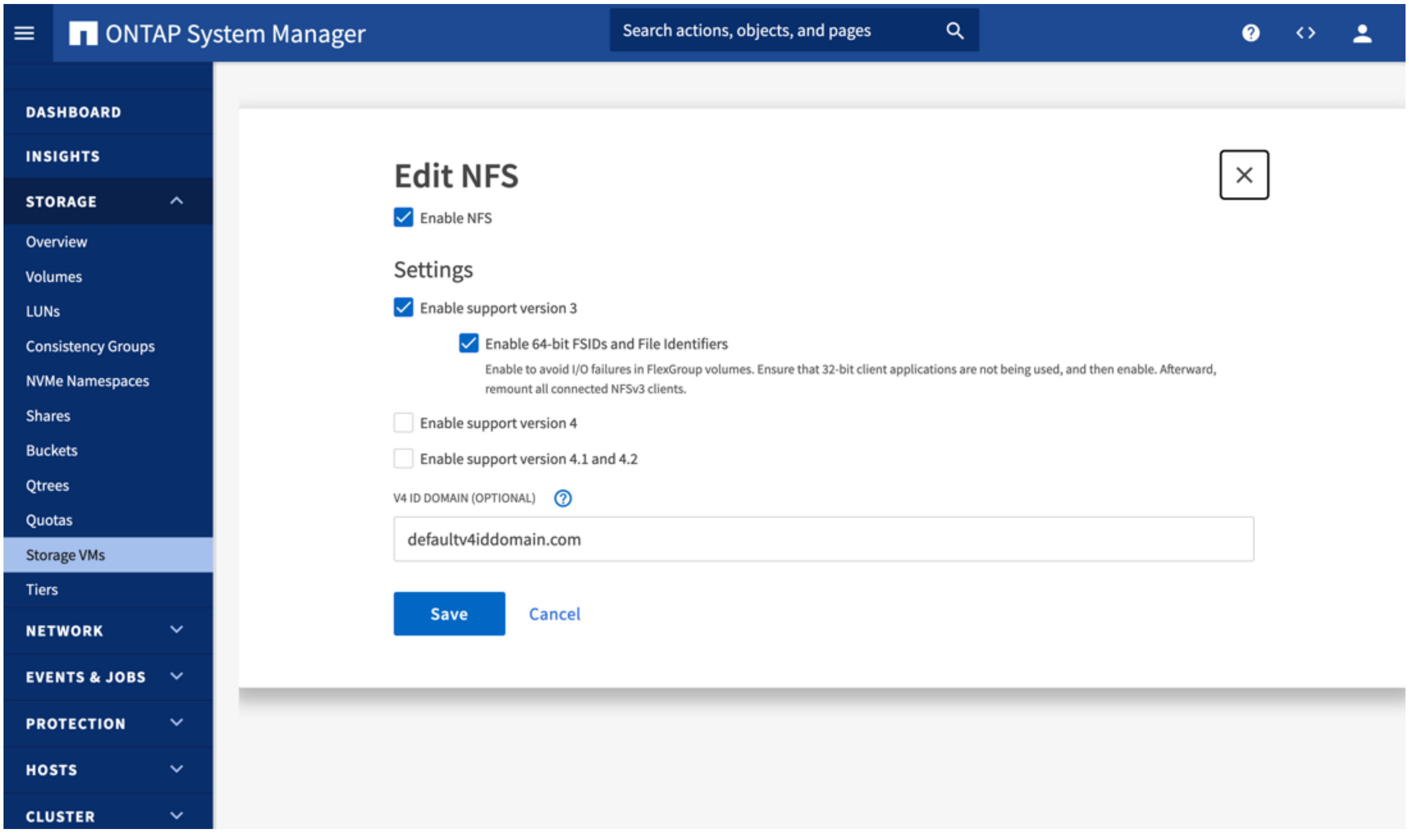
The screenshot below shows the Storage VMs (formally known as Vserver) configured as "ORANFS-SVM" for this solution.

```
FlexPod-A800::> vserver show
                               Admin      Operational Root
Vserver     Type    Subtype    State      State       Volume     Aggregate
----------- ------- ---------- ---------- ----------- ---------- ----------
Eng-NVME    data    default    running    running     Eng_NVME_  aggr1_
                                                      root       node2
FlexPod-A800 admin  -          -          -           -          -
FlexPod-A800-CT1 node -        -          -           -          -
FlexPod-A800-CT2 node -        -          -           -          -
Infra-SVM   data    default    running    running     svm_root   aggr1_
                                                                 node1
ORANFS-SVM  data    default    running    running     ORANFSSVM_ aggr1_
                                                      root       node1
6 entries were displayed.
```

The SVM named "ORANFS-SVM" was configured to carry all NFS traffic for this Oracle RAC Databases solution.



Only the NFS V3 protocol was allowed for “ORANFS-SVM” as shown below:



The detailed configuration for ORANFS-SVM is shown below:

```
FlexPod-A800::> vserver show -vserver ORANFS-SVM

                                    Vserver: ORANFS-SVM
                               Vserver Type: data
                            Vserver Subtype: default
                               Vserver UUID: 9c45eff6-b63d-11ec-9509-00a098b92509
                                Root Volume: ORANFSSVM_root
                                  Aggregate: aggr1_node1
                                 NIS Domain: -
                 Root Volume Security Style: unix
                                LDAP Client: -
               Default Volume Language Code: C.UTF-8
                            Snapshot Policy: default
                              Data Services: data-cifs, data-flexcache,
                                             data-iscsi, data-nfs,
                                             data-nvme-tcp
                                    Comment:
                               Quota Policy: default
                List of Aggregates Assigned: aggr1_node1, aggr1_node2
 Limit on Maximum Number of Volumes allowed: unlimited
                        Vserver Admin State: running
                  Vserver Operational State: running
   Vserver Operational State Stopped Reason: -
                          Allowed Protocols: nfs, s3
                       Disallowed Protocols: cifs, fcp, iscsi, ndmp, nvme
           Is Vserver with Infinite Volume: false
                           QoS Policy Group: -
                        Caching Policy Name: -
                                Config Lock: false
                               IPspace Name: Default
                         Foreground Process: -
                    Logical Space Reporting: false
                  Logical Space Enforcement: false
Default Anti_ransomware State of the Vserver's Volumes: disabled
            Enable Analytics on New Volumes: false
    Enable Activity Tracking on New Volumes: false
```

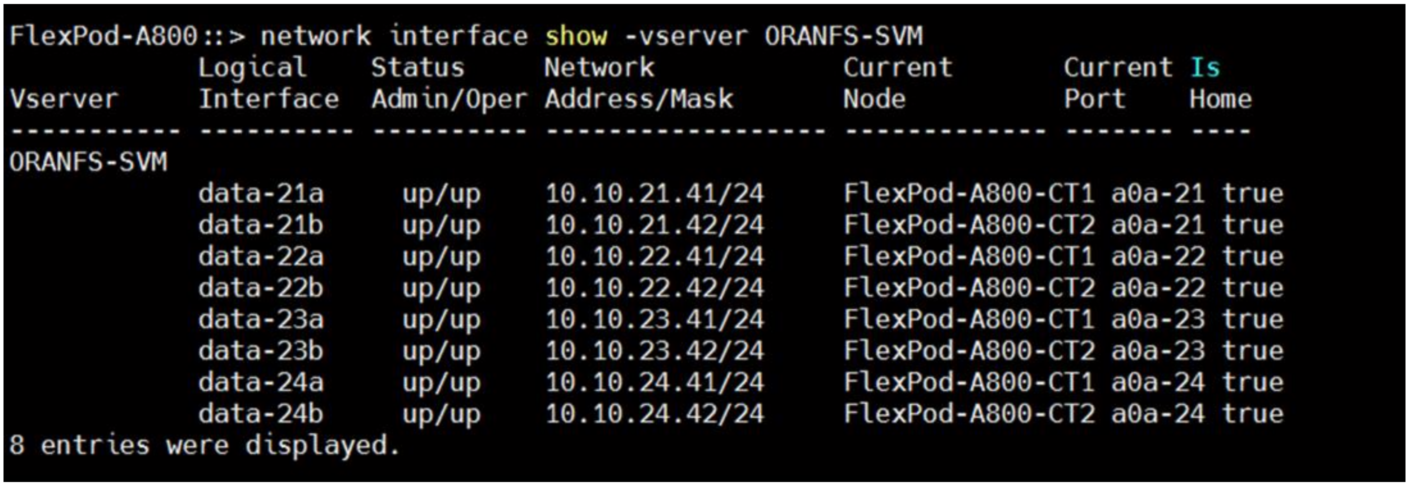
For this solution, the broadcast-domain was configured as “NFS-data” with 9000 MTU and assigned to the default IPspace as shown below:

```
FlexPod-A800::> network port broadcast-domain show
IPspace Broadcast                                         Update
Name    Domain Name    MTU  Port List                     Status Details
------- ----------- ------  ----------------------------- --------------
Cluster Cluster       9000
                            FlexPod-A800-CT2:e0a          complete
                            FlexPod-A800-CT2:e1a          complete
                            FlexPod-A800-CT1:e0a          complete
                            FlexPod-A800-CT1:e1a          complete
Default Default       9000
                            FlexPod-A800-CT2:e0M          complete
                            FlexPod-A800-CT2:e4a          complete
                            FlexPod-A800-CT2:e4b          complete
                            FlexPod-A800-CT1:e0M          complete
                            FlexPod-A800-CT1:e4a          complete
                            FlexPod-A800-CT1:e4b          complete
        NFS-data      9000
                            FlexPod-A800-CT2:a0a-21       complete
                            FlexPod-A800-CT2:a0a-22       complete
                            FlexPod-A800-CT2:a0a-23       complete
                            FlexPod-A800-CT2:a0a-24       complete
                            FlexPod-A800-CT1:a0a-21       complete
                            FlexPod-A800-CT1:a0a-22       complete
                            FlexPod-A800-CT1:a0a-23       complete
                            FlexPod-A800-CT1:a0a-24       complete
3 entries were displayed.
```

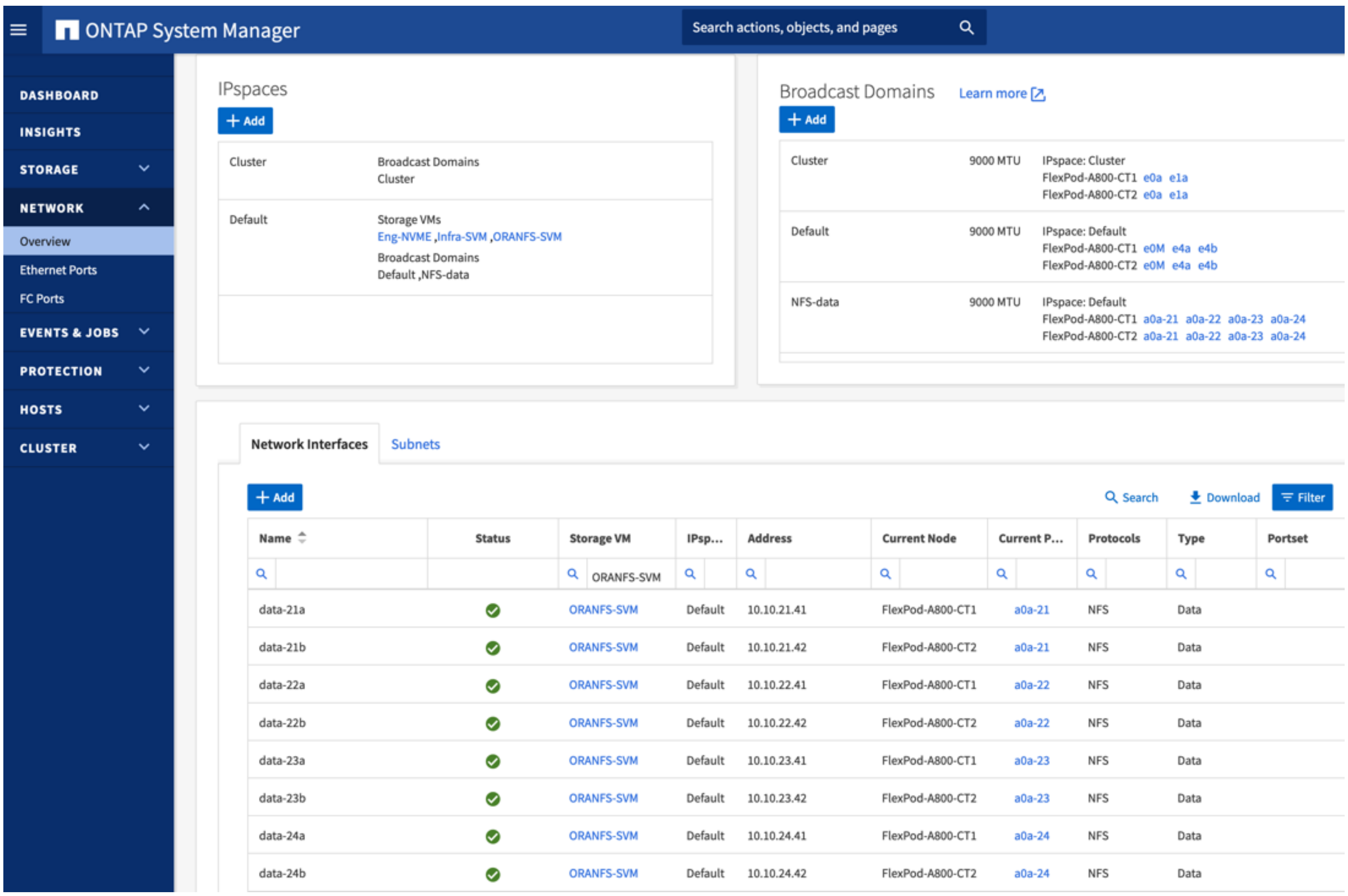
One "Link Aggregation Group" as "a0a" was configured across both NetApp controller nodes (FlexPod-A800-CT1 and FlexPod-A800-CT2) across all four 100G ports as show below, to enable the storage network traffic across all ports and provide high availability.

```
FlexPod-A800::> ifgrp show
         Port       Distribution                   Active
Node     IfGrp      Function     MAC Address       Ports   Ports
-------- ---------- ------------ ----------------- ------- -------------------
FlexPod-A800-CT1
         a0a        port         02:a0:98:b9:54:e3 full    e5a, e5b
FlexPod-A800-CT2
         a0a        port         02:a0:98:b9:25:09 full    e5a, e5b
2 entries were displayed.
```

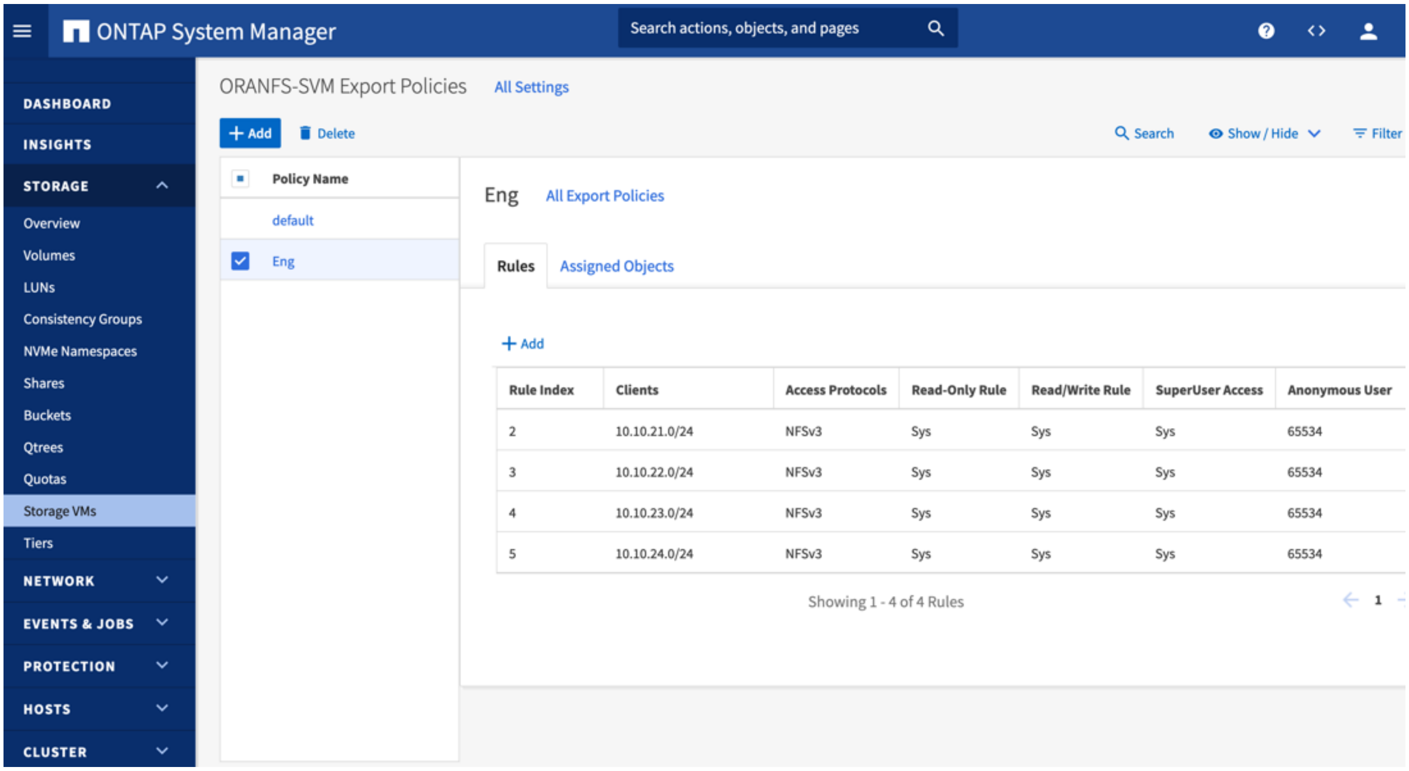
With "ORANFS-SVM," a total eight Logical Interfaces (LIFs) were configured across both storage controller nodes. "Link Aggregation Group" was configured as "a0a" and the configured data interface for all four VLANs as "data-21a", "data-22a", "data-23a" and "data-24a" on each controller so that all four VLAN networks (21 to 24) go across both controllers as shown below:



The following screenshot shows the overview of the network configuration used in this solution:



The export policy “Eng” was configured and added rules with clients subnets for UNIX systems to allow the NFSv3 protocol as shown below:



To test and validate various benchmarking and database deployments, multiple volumes were created. An equal number of volumes were distributed on each of the storage controllers by placing them into the aggregate equally.

Operating System and Database Deployment

This chapter contains the following:

The design goal of this reference architecture was to represent a real-world environment as closely as possible. As explained in previously, a server profile was created within Cisco Intersight to rapidly deploy all stateless servers on an eight node Oracle RAC. For this solution, the local virtual drive (local raid volume) was configured on each blade server into a Cisco UCS IMM configuration for local boot. The Oracle Linux Server 8.6 with UEK Kernel (5.4.17-2136.307.3.1.el8uek.x86_64) was used and configured network interfaces to create NFS clients to mount database volumes on each of the server node. After configuring the operating system and network connectivity, all prerequisites packages were configured to install the Oracle Database 21c Grid Infrastructure and Oracle Database 21c software to create an eight node Oracle Multitenant RAC 21c database solution for this solution.

This chapter describes the high-level steps to configure the Oracle Linux Hosts and deploy the Oracle RAC Database solution.

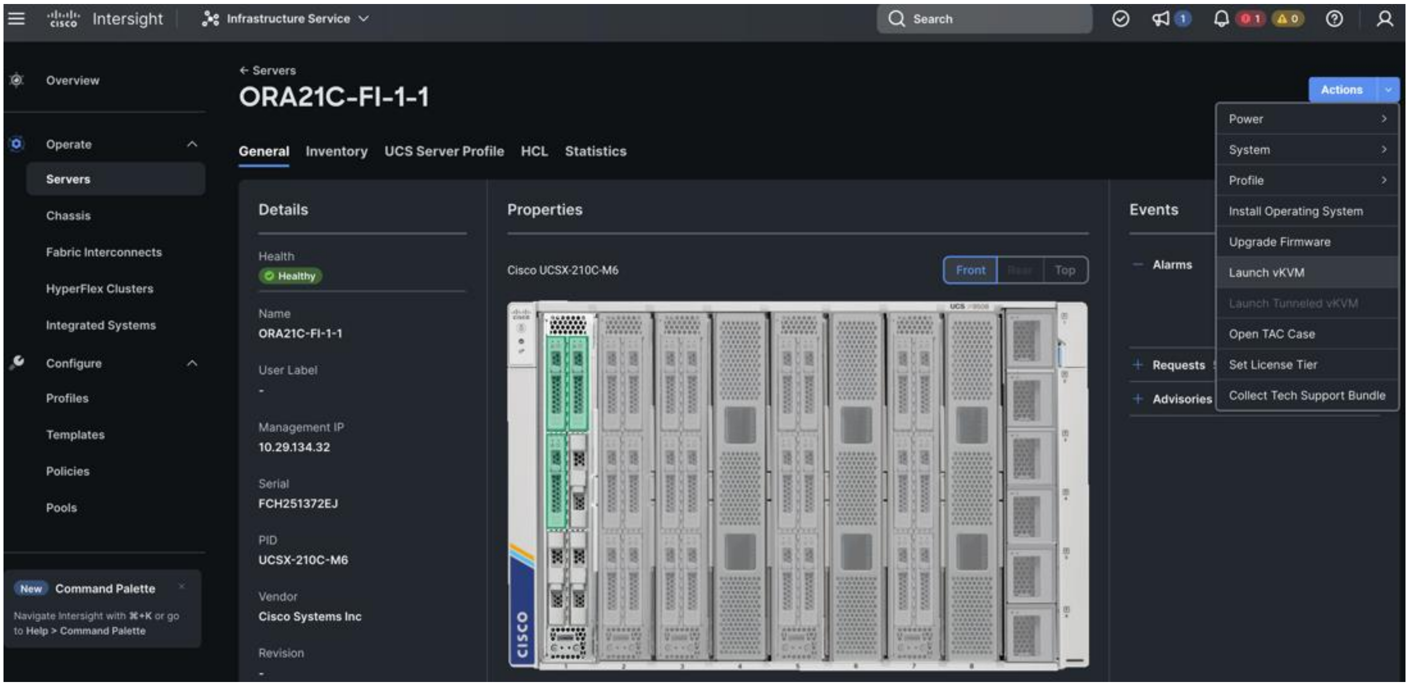
Configure the Operating System

Note: The detailed installation process is not explained in this document, but the following procedure describes the key steps for the OS installation.

Procedure 1. Configure OS

Step 1. Download the Oracle Linux 8.6 OS image from https://edelivery.oracle.com/linux.

Step 2. Launch the vKVM console on your server by going to Cisco Intersight > Infrastructure Service > Operate > Servers > click Chassis 1 Server 1 > from the Actions drop-down list select Launch vKVM.



Step 3. Click Accept security and open KVM. Click Virtual Media > vKVM-Mapped vDVD. Click Browse and map the Oracle Linux ISO image, click Open and then click Map Drive. After mapping the iso file, click Power > Power Cycle System to reboot the server.

When the Server boots, it will detect the boot order and start booting from the Virtual mapped DVD as previously configured.

Step 4. During the server boot order, it detects the virtual media connected as Oracle Linux ISO DVD media and it will launch the Oracle Linux OS installer. Select language and for the Installation destination assign the local virtual drive. Apply the hostname and click Configure Network to configure any or all the network interfaces. Alternatively, you can configure only the "Public Network" in this step. You can configure additional interfaces as part of post OS install steps.

Note: For an additional RPM package, we recommend selecting the "Customize Now" option and the relevant packages according to your environment.

Step 5. After the OS installation finishes, reboot the server, and complete the appropriate registration steps.

Step 6. Repeat steps 1 – 4 on all server nodes and install Oracle Linux 8.6 to create an eight node linux system.

Step 7. Optionally, you can choose to synchronize the time with ntp server. Alternatively, you can choose to use the Oracle RAC cluster synchronization daemon (OCSSD). Both NTP and OCSSD are mutually exclusive and OCSSD will be setup during GRID install if NTP is not configured.

Set Default Kernel to UEK

For the x86_64 platform, Oracle Linux 8.6 ships with the following default kernel packages:

- kernel-4.18.0-372.9.1.el8 (Red Hat Compatible Kernel (RHCK))

- kernel-uek-5.4.17-2136.307.3 (Unbreakable Enterprise Kernel Release 6 (UEK R6))

For new installations, the UEK kernel is automatically enabled and installed. It also becomes the default kernel on first boot. For this solution design, the Oracle UEK Kernel was used.

Procedure 1. Configure the default kernel to UEK

After installing Oracle Linux 8.6 on all the server nodes (flex1, flex2, flex3, flex4, fle5, flex6, flex7 and flex8), you can configure the default kernel to UEK.

Step 1. Check the list of installed kernels:

```
[root@flex1 ~]# ls -al /boot/vmlinuz-*
-rwxr-xr-x. 1 root root 10377840 Jan 13 12:28
/boot/vmlinuz-0-rescue-c254888825f74248aa010088ef06066e
-rwxr-xr-x. 1 root root 10467936 May 11  2022 /boot/vmlinuz-4.18.0-372.9.1.el8.x86_64
-rwxr-xr-x. 1 root root 10377840 May  9  2022 /boot/vmlinuz-5.4.17-2136.307.3.1.el8uek.x86_64
```

Step 2. Set the default kernel and reboot the node:

```
[root@flex1 ~]# grubby --set-default=/boot/vmlinuz-5.4.17-2136.307.3.1.el8uek.x86_64
[root@flex1 ~]# systemctl reboot
```

Step 3. After the node reboots, verify the default kernel boot:

```
[root@flex1 ~]# grubby --default-kernel
/boot/vmlinuz-5.4.17-2136.307.3.1.el8uek.x86_64
```

Step 4. Repeat steps 1 - 3 and configure the UEK as the default kernel boot on all nodes.

Install the ENIC Driver for Linux OS

For this solution, the Linux ENIC drivers were configured as follows:

```
[root@flex1 ~]# modinfo enic
filename:       /lib/modules/5.4.17-2136.307.3.1.el8uek.x86_64/extra/enic/enic.ko
version:        4.3.0.1-918.18
retpoline:      Y
license:        GPL v2
author:         Scott Feldman <scofeldm@cisco.com>
description:    Cisco VIC Ethernet NIC Driver
srcversion:     F80A23088A7B93D0F83CC78
alias:          pci:v00001137d00000071sv*sd*bc*sc*i*
alias:          pci:v00001137d00000044sv*sd*bc*sc*i*
alias:          pci:v00001137d00000043sv*sd*bc*sc*i*
depends:
retpoline:      Y
name:           enic
vermagic:       5.4.17-2136.307.3.1.el8uek.x86_64 SMP mod_unload modversions
sig_id:         PKCS#7
signer:         Cisco UCS Driver Signing REL Cert
sig_key:        D0:54:9A:88:88:DD:0E:7A
sig_hashalgo:   sha256
signature:      A0:49:5D:92:B3:5C:3B:B5:F6:F5:01:8C:A2:E7:37:B6:41:AE:39:14:
                A6:B0:97:CE:63:25:C4:25:B5:5C:BE:AF:06:08:78:8F:78:D7:13:4F:
                CF:FF:77:7E:F5:34:04:8D:5F:E4:3B:21:AF:5C:D5:B4:9A:49:82:BA:
                9F:9F:FC:6B:5C:F4:E3:AF:9D:AE:AD:69:3B:AA:1C:35:6F:64:F4:D7:
                8F:7D:89:E1:5D:21:AF:E9:2C:B5:1E:7B:D4:68:3D:23:72:8B:B8:54:
                9D:B9:DA:FC:49:F7:70:0A:E5:D4:B2:93:05:E4:F9:84:31:65:48:CA:
                1B:F7:2F:27:57:07:E3:CE:39:08:42:97:B9:E9:88:DD:7C:E3:03:C8:
                21:3D:F8:FA:76:17:0F:3D:8C:55:AD:D6:B5:76:E8:88:A2:1E:9F:30:
                8D:CC:DB:86:03:E7:15:9E:8B:04:A5:52:C3:2C:7C:F2:62:26:F3:02:
                E3:DA:9E:F9:97:2A:39:D3:45:D5:6B:7B:F7:AC:DC:F1:72:A1:14:B4:
                05:C6:18:23:B1:2B:22:BE:29:81:9B:45:A4:6E:17:56:88:A3:9F:39:
                24:B4:4B:92:A6:50:FE:EA:0A:6F:70:03:1A:FF:17:1D:B7:36:CE:DB:
                8F:FE:58:3C:FE:E1:D6:A9:08:78:89:83:0A:46:B3:F1
parm:           rxcopybreak:Maximum size of packet that is copied to a new buffer on receive (uint)
[root@flex1 ~]# cat /sys/module/enic/version
4.3.0.1-918.18
[root@flex1 ~]# rpm -qa kmod-enic
kmod-enic-4.3.0.1-918.18.oluek_5.4.17_2136.307.3.1.x86_64
```

Procedure 1. Install ENIC Drivers for Linux OS

Step 1. Download the supported Cisco UCS Linux Drivers for the Cisco UCS X-Series Blade Server Software for Linux from: [https://software.cisco.com/download/home/286329080/type/283853158/release/5.1(0a).](https://software.cisco.com/download/home/286329080/type/283853158/release/5.1(0a))

Step 2. Check the current driver version by running the following commands:

```
[root@flex1 ~]# modinfo enic
[root@flex1 ~]# cat /sys/module/enic/version
```

Step 3. Mount the driver ISO file to the virtual drive. Go to the Network folder to get the Cisco VIC ENIC driver for Oracle Linux 8.6. SCP that connect the ENIC driver to the Linux Host and SSH into the host to install the driver.

Step 4. Install the supported Linux ENIC drivers, by running the following commands:

```
[root@flex1 software]# rpm -ivh kmod-enic-4.3.0.1-918.18.oluek_5.4.17_2136.307.3.1.x86_64.rpm
Verifying...                          ################################# [100%]
Preparing...                          ################################# [100%]
Updating / installing...
```

```
    1: kmod-enic-4.3.0.1-918.18.oluek_5.4.17_2136.307.3.1.x86_64
################################# [100%]
```

Step 5. Reboot the server and verify that the new driver is running:

```
[root@flex1 ~]# modinfo enic | grep version
version:        4.3.0.1-918.18
srcversion:     F80A23088A7B93D0F83CC78
vermagic:       5.4.17-2136.307.3.1.el8uek.x86_64 SMP mod_unload modversions

[root@flex1 ~]# cat /sys/module/enic/version
4.3.0.1-918.18
```

Step 6. Repeat steps 1 - 5 and configure the ENIC drivers on all eight Linux nodes.

Note: You should use a matching ENIC and FNIC pair. Check the Cisco UCS supported driver release for more information about the supported kernel version: https://www.cisco.com/c/en/us/support/docs/servers-unified-computing/ucs-manager/116349-technote-product-00.html.

Configure Public, Private, and Storage Network Interfaces

If you have not configured network settings during OS installation, then configure it now. Each node must have at least six network interface cards (NICs), or network adapters. One adapter is for the public network interface, one adapter is for the private network interface (RAC interconnect) and four adapters are for the storage network interfaces.

Procedure 1. Configure Management Public and Private Network Interfaces

Step 1. Login as a root user into each Linux node and go to "/etc/sysconfig/network-scripts/"

Step 2. Configure the Public network, Private network, and Storage network IP addresses according to your environments.

Note: Configure the Private, Public and Storage network with the appropriate IP addresses on all eight Linux Oracle RAC nodes.

Configure OS Prerequisites for Oracle Software

To successfully install the Oracle RAC Database 21c software, configure the operating system prerequisites on all eight Linux nodes.

Note: Follow the steps according to your environment and requirements. For more information, see the Install and Upgrade Guide for Linux for Oracle Database 21c:

Note: https://docs.oracle.com/en/database/oracle/oracle-database/21/cwlin/index.html

Note: https://docs.oracle.com/en/database/oracle/oracle-database/21/ladbi/index.html

Procedure 1. Configure the OS prerequisites

Step 1. To configure the operating system prerequisites using RPM for Oracle 21c software on Linux node, install the "oracle-database-preinstall-21c (oracle-database-preinstall-21c-1.0-1.el8.x86_64.rpm)" rpm package on all eight nodes. You can also download the required packages from: https://public-yum.oracle.com/oracle-linux-8.html

Step 2. If you plan to use the "oracle-database-preinstall-21c" rpm package to perform all your prerequisites setup automatically, then login as root user and issue the following command on all each of the RAC nodes:

```
[root@flex1 ~]# yum install oracle-database-preinstall-21c-1.0-1.el8.x86_64.rpm
```

Note: If you have not used the " oracle-database-preinstall-21c " package, then you will have to manually perform the prerequisites tasks on all the nodes.

Configure Additional OS Prerequisites

After configuring the automatic or manual prerequisites steps, you have a few additional steps to complete the prerequisites to install the Oracle database software on all eight Linux nodes.

Procedure 1. Disable SELinux

Since most organizations might already be running hardware-based firewalls to protect their corporate networks, you need to disabled Security Enhanced Linux (SELinux) and the firewalls at the server level for this reference architecture.

Step 1. Set the secure Linux to permissive by editing the `"/etc/selinux/config"` file, making sure the SELINUX flag is set as follows:

```
SELINUX=permissive
```

Procedure 2. Disable Firewall

Step 1. Check the status of the firewall by running following commands. (The status displays as active (running) or inactive (dead)). If the firewall is active / running, run this command to stop it:

```
systemctl status firewalld.service
systemctl stop firewalld.service
```

Step 2. To completely disable the firewalld service so it does not reload when you restart the host machine, run the following command:

```
systemctl disable firewalld.service
```

Procedure 3. Create Grid User

Step 1. Run this command to create a grid user:

```
useradd -u 54322 -g oinstall -G dba grid
```

Procedure 4. Set the User Passwords

Step 1. Run these commands to change the password for Oracle and Grid Users:

```
passwd oracle
passwd grid
```

Procedure 5. Configure "/etc/hosts"

Step 1. Login as a root user into the Linux

Step 2. node and edit the “/etc/hosts” file.

Step 3. Provide the details for Public IP Address, Private IP Address, SCAN IP Address, and Virtual IP Address for all the nodes. Configure these settings in each Oracle RAC Nodes as shown below:

```
[root@flex1 ~]# cat /etc/hosts
127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4
##::1         localhost localhost.localdomain localhost6 localhost6.localdomain6
###     Public IP       ###
10.29.134.101    flex1   flex1.ciscoucs.com
10.29.134.102    flex2   flex2.ciscoucs.com
10.29.134.103    flex3   flex3.ciscoucs.com
10.29.134.104    flex4   flex4.ciscoucs.com
10.29.134.105    flex5   flex5.ciscoucs.com
10.29.134.106    flex6   flex6.ciscoucs.com
10.29.134.107    flex7   flex7.ciscoucs.com
10.29.134.108    flex8   flex8.ciscoucs.com
###       Virtual IP          ###
10.29.134.109    flex1-vip       flex1-vip.ciscoucs.com
10.29.134.110    flex2-vip       flex2-vip.ciscoucs.com
10.29.134.111    flex3-vip       flex3-vip.ciscoucs.com
10.29.134.112    flex4-vip       flex4-vip.ciscoucs.com
10.29.134.113    flex5-vip       flex5-vip.ciscoucs.com
10.29.134.114    flex6-vip       flex6-vip.ciscoucs.com
10.29.134.115    flex7-vip       flex7-vip.ciscoucs.com
10.29.134.116    flex8-vip       flex8-vip.ciscoucs.com
###       Private IP          ###
192.168.10.101     flex1-priv      flex1-priv.ciscoucs.com
192.168.10.102     flex2-priv      flex2-priv.ciscoucs.com
192.168.10.103     flex3-priv      flex3-priv.ciscoucs.com
192.168.10.104     flex4-priv      flex4-priv.ciscoucs.com
192.168.10.105     flex5-priv      flex5-priv.ciscoucs.com
192.168.10.106     flex6-priv      flex6-priv.ciscoucs.com
192.168.10.107     flex7-priv      flex7-priv.ciscoucs.com
192.168.10.108     flex8-priv      flex8-priv.ciscoucs.com
###       SCAN IP             ###
10.29.134.117    flex-scan       flex-scan.ciscoucs.com
10.29.134.118    flex-scan       flex-scan.ciscoucs.com
10.29.134.119    flex-scan       flex-scan.ciscoucs.com
```

Step 4. You must configure the following addresses manually in your corporate setup:

- A Public and Private IP Address for each Linux node
- A Virtual IP address for each Linux node
- Three single client access name (SCAN) address for the oracle database cluster

Note: These steps were performed on all of the eight linux nodes. These steps complete the prerequisites for the Oracle Database 21c installation at OS level on the Oracle RAC Nodes.

Procedure 6. Configure "/etc/sysctl.conf" Parameter

You need to configure additional parameters for "/etc/sysctl.conf" specifically for the Oracle Database environments deploying on NFS protocol. Refer to the Oracle support notes **762374.1** for more detail: https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=486725239930951&id=762374.1&displayIndex=1&_afrWindowMode=0&_adf.ctrl-state=z0q3mn8eh_211

Note: These settings may change as new architectures evolve.

Configure NFS on NetApp Storage

You will use the "OCRVOTE" file system on the storage array to store the OCR (Oracle Cluster Registry) files, Voting Disk files, and other clusterware files.

Note: Multiple file systems were created to store data files, control files, and log files for the database.

Procedure 1. Create NFS Mount Point in "/etc/fstab"

The following local directories were created on each Oracle RAC node to mount the NFS file system:

/ocrvote → OCR, Voting disk, Clusterware Files

/<database-name>data → Data files for database

/<database-name>log → Log files for database

/fio → File systems to run FIO Workloads

Step 1. Edit "/etc/fstab" file in each Oracle RAC node and enter the following to configure the mount option for all file systems:

```
10.10.21.41:/ocrvote    /ocrvote        nfs
rw,bg,hard,rsize=32768,wsize=32768,nfsvers=3,actimeo=0,nointr,timeo=600,tcp

10.10.21.41:/findata01   /findata01       nfs
rw,bg,hard,rsize=524288,wsize=524288,nfsvers=3,actimeo=0,nointr,timeo=600,tcp

10.10.22.42:/findata02   /findata02       nfs
rw,bg,hard,rsize=524288,wsize=524288,nfsvers=3,actimeo=0,nointr,timeo=600,tcp

10.10.23.41:/findata03   /findata03       nfs
rw,bg,hard,rsize=524288,wsize=524288,nfsvers=3,actimeo=0,nointr,timeo=600,tcp
```

```
10.10.24.42:/findata04   /findata04       nfs
rw,bg,hard,rsize=524288,wsize=524288,nfsvers=3,actimeo=0,nointr,timeo=600,tcp

10.10.21.41:/findata05   /findata05       nfs
rw,bg,hard,rsize=524288,wsize=524288,nfsvers=3,actimeo=0,nointr,timeo=600,tcp

10.10.22.42:/findata06   /findata06       nfs
rw,bg,hard,rsize=524288,wsize=524288,nfsvers=3,actimeo=0,nointr,timeo=600,tcp

10.10.23.41:/findata07   /findata07       nfs
rw,bg,hard,rsize=524288,wsize=524288,nfsvers=3,actimeo=0,nointr,timeo=600,tcp

10.10.24.42:/findata08   /findata08       nfs
rw,bg,hard,rsize=524288,wsize=524288,nfsvers=3,actimeo=0,nointr,timeo=600,tcp

10.10.21.41:/findata09   /findata09       nfs
rw,bg,hard,rsize=524288,wsize=524288,nfsvers=3,actimeo=0,nointr,timeo=600,tcp

10.10.22.42:/findata10   /findata10       nfs
rw,bg,hard,rsize=524288,wsize=524288,nfsvers=3,actimeo=0,nointr,timeo=600,tcp

10.10.23.41:/findata11   /findata11       nfs
rw,bg,hard,rsize=524288,wsize=524288,nfsvers=3,actimeo=0,nointr,timeo=600,tcp

10.10.24.42:/findata12   /findata12       nfs
rw,bg,hard,rsize=524288,wsize=524288,nfsvers=3,actimeo=0,nointr,timeo=600,tcp

10.10.21.41:/findata13   /findata13       nfs
rw,bg,hard,rsize=524288,wsize=524288,nfsvers=3,actimeo=0,nointr,timeo=600,tcp

10.10.22.42:/findata14   /findata14       nfs
rw,bg,hard,rsize=524288,wsize=524288,nfsvers=3,actimeo=0,nointr,timeo=600,tcp

10.10.23.41:/findata15   /findata15       nfs
rw,bg,hard,rsize=524288,wsize=524288,nfsvers=3,actimeo=0,nointr,timeo=600,tcp

10.10.24.42:/findata16   /findata16       nfs
rw,bg,hard,rsize=524288,wsize=524288,nfsvers=3,actimeo=0,nointr,timeo=600,tcp

10.10.21.41:/finlog01   /finlog01       nfs
rw,bg,hard,rsize=524288,wsize=524288,nfsvers=3,actimeo=0,nointr,timeo=600,tcp

10.10.22.42:/finlog02   /finlog02       nfs
rw,bg,hard,rsize=524288,wsize=524288,nfsvers=3,actimeo=0,nointr,timeo=600,tcp

10.10.23.41:/finlog03   /finlog03       nfs
rw,bg,hard,rsize=524288,wsize=524288,nfsvers=3,actimeo=0,nointr,timeo=600,tcp

10.10.24.42:/finlog04   /finlog04       nfs
rw,bg,hard,rsize=524288,wsize=524288,nfsvers=3,actimeo=0,nointr,timeo=600,tcp

10.10.21.41:/soedata01   /soedata01       nfs
rw,bg,hard,rsize=524288,wsize=524288,nfsvers=3,actimeo=0,nointr,timeo=600,tcp

10.10.22.42:/soedata02   /soedata02       nfs
rw,bg,hard,rsize=524288,wsize=524288,nfsvers=3,actimeo=0,nointr,timeo=600,tcp
```

```
10.10.23.41:/soedata03   /soedata03       nfs
rw,bg,hard,rsize=524288,wsize=524288,nfsvers=3,actimeo=0,nointr,timeo=600,tcp

10.10.24.42:/soedata04   /soedata04       nfs
rw,bg,hard,rsize=524288,wsize=524288,nfsvers=3,actimeo=0,nointr,timeo=600,tcp

10.10.21.41:/soedata05   /soedata05       nfs
rw,bg,hard,rsize=524288,wsize=524288,nfsvers=3,actimeo=0,nointr,timeo=600,tcp

10.10.22.42:/soedata06   /soedata06       nfs
rw,bg,hard,rsize=524288,wsize=524288,nfsvers=3,actimeo=0,nointr,timeo=600,tcp

10.10.23.41:/soedata07   /soedata07       nfs
rw,bg,hard,rsize=524288,wsize=524288,nfsvers=3,actimeo=0,nointr,timeo=600,tcp

10.10.24.42:/soedata08   /soedata08       nfs
rw,bg,hard,rsize=524288,wsize=524288,nfsvers=3,actimeo=0,nointr,timeo=600,tcp

10.10.21.41:/soedata09   /soedata09       nfs
rw,bg,hard,rsize=524288,wsize=524288,nfsvers=3,actimeo=0,nointr,timeo=600,tcp

10.10.22.42:/soedata10   /soedata10       nfs
rw,bg,hard,rsize=524288,wsize=524288,nfsvers=3,actimeo=0,nointr,timeo=600,tcp

10.10.23.41:/soedata11   /soedata11       nfs
rw,bg,hard,rsize=524288,wsize=524288,nfsvers=3,actimeo=0,nointr,timeo=600,tcp

10.10.24.42:/soedata12   /soedata12       nfs
rw,bg,hard,rsize=524288,wsize=524288,nfsvers=3,actimeo=0,nointr,timeo=600,tcp

10.10.21.41:/soedata13   /soedata13       nfs
rw,bg,hard,rsize=524288,wsize=524288,nfsvers=3,actimeo=0,nointr,timeo=600,tcp

10.10.22.42:/soedata14   /soedata14       nfs
rw,bg,hard,rsize=524288,wsize=524288,nfsvers=3,actimeo=0,nointr,timeo=600,tcp

10.10.23.41:/soedata15   /soedata15       nfs
rw,bg,hard,rsize=524288,wsize=524288,nfsvers=3,actimeo=0,nointr,timeo=600,tcp

10.10.24.42:/soedata16   /soedata16       nfs
rw,bg,hard,rsize=524288,wsize=524288,nfsvers=3,actimeo=0,nointr,timeo=600,tcp

10.10.21.41:/soelog01   /soelog01       nfs
rw,bg,hard,rsize=524288,wsize=524288,nfsvers=3,actimeo=0,nointr,timeo=600,tcp

10.10.22.42:/soelog02   /soelog02       nfs
rw,bg,hard,rsize=524288,wsize=524288,nfsvers=3,actimeo=0,nointr,timeo=600,tcp

10.10.23.41:/soelog03   /soelog03       nfs
rw,bg,hard,rsize=524288,wsize=524288,nfsvers=3,actimeo=0,nointr,timeo=600,tcp

10.10.24.42:/soelog04   /soelog04       nfs
rw,bg,hard,rsize=524288,wsize=524288,nfsvers=3,actimeo=0,nointr,timeo=600,tcp

10.10.21.41:/shdata01   /shdata01       nfs
rw,bg,hard,rsize=524288,wsize=524288,nfsvers=3,actimeo=0,nointr,timeo=600,tcp
```

```
10.10.22.42:/shdata02   /shdata02       nfs
rw,bg,hard,rsize=524288,wsize=524288,nfsvers=3,actimeo=0,nointr,timeo=600,tcp

10.10.23.41:/shdata03   /shdata03       nfs
rw,bg,hard,rsize=524288,wsize=524288,nfsvers=3,actimeo=0,nointr,timeo=600,tcp

10.10.24.42:/shdata04   /shdata04       nfs
rw,bg,hard,rsize=524288,wsize=524288,nfsvers=3,actimeo=0,nointr,timeo=600,tcp

10.10.21.41:/shdata05   /shdata05       nfs
rw,bg,hard,rsize=524288,wsize=524288,nfsvers=3,actimeo=0,nointr,timeo=600,tcp

10.10.22.42:/shdata06   /shdata06       nfs
rw,bg,hard,rsize=524288,wsize=524288,nfsvers=3,actimeo=0,nointr,timeo=600,tcp

10.10.23.41:/shdata07   /shdata07       nfs
rw,bg,hard,rsize=524288,wsize=524288,nfsvers=3,actimeo=0,nointr,timeo=600,tcp

10.10.24.42:/shdata08   /shdata08       nfs
rw,bg,hard,rsize=524288,wsize=524288,nfsvers=3,actimeo=0,nointr,timeo=600,tcp

10.10.21.41:/shdata09   /shdata09       nfs
rw,bg,hard,rsize=524288,wsize=524288,nfsvers=3,actimeo=0,nointr,timeo=600,tcp

10.10.22.42:/shdata10   /shdata10       nfs
rw,bg,hard,rsize=524288,wsize=524288,nfsvers=3,actimeo=0,nointr,timeo=600,tcp

10.10.23.41:/shdata11   /shdata11       nfs
rw,bg,hard,rsize=524288,wsize=524288,nfsvers=3,actimeo=0,nointr,timeo=600,tcp

10.10.24.42:/shdata12   /shdata12       nfs
rw,bg,hard,rsize=524288,wsize=524288,nfsvers=3,actimeo=0,nointr,timeo=600,tcp

10.10.21.41:/shdata13   /shdata13       nfs
rw,bg,hard,rsize=524288,wsize=524288,nfsvers=3,actimeo=0,nointr,timeo=600,tcp

10.10.22.42:/shdata14   /shdata14       nfs
rw,bg,hard,rsize=524288,wsize=524288,nfsvers=3,actimeo=0,nointr,timeo=600,tcp

10.10.23.41:/shdata15   /shdata15       nfs
rw,bg,hard,rsize=524288,wsize=524288,nfsvers=3,actimeo=0,nointr,timeo=600,tcp

10.10.24.42:/shdata16   /shdata16       nfs
rw,bg,hard,rsize=524288,wsize=524288,nfsvers=3,actimeo=0,nointr,timeo=600,tcp

10.10.21.41:/shlog01   /shlog01       nfs
rw,bg,hard,rsize=524288,wsize=524288,nfsvers=3,actimeo=0,nointr,timeo=600,tcp

10.10.22.42:/shlog02   /shlog02       nfs
rw,bg,hard,rsize=524288,wsize=524288,nfsvers=3,actimeo=0,nointr,timeo=600,tcp

10.10.23.41:/shlog03   /shlog03       nfs
rw,bg,hard,rsize=524288,wsize=524288,nfsvers=3,actimeo=0,nointr,timeo=600,tcp

10.10.24.42:/shlog04   /shlog04       nfs
rw,bg,hard,rsize=524288,wsize=524288,nfsvers=3,actimeo=0,nointr,timeo=600,tcp
```

Step 2. Mount the file system using the “mount -a” command.

Note: The Oracle Direct NFS (dNFS) configuration is completed at a later stage.

Step 3. Change the permission of the mount points to Oracle user as follows:

```
[root@flex1 ~]# chown -R grid:oinstall /ocrvote

[root@flex1 ~]# chown -R oracle:oinstall /<database-name>data

[root@flex1 ~]# chown -R oracle:oinstall /<database-name>log
```

Step 4. These NFS file systems were mounted on all eight nodes with similar mount names on the storage VLANs (21 – 24). Verify that all the file system volumes are mounted as follows:

```
[root@flex1 ~]# df -h /ocrvote/

Filesystem             Size  Used Avail Use% Mounted on

10.10.21.41:/ocrvote  190G  252M  190G   1% /ocrvote



[root@flex1 ~]# df -h /fin*/

Filesystem               Size  Used Avail Use% Mounted on

10.10.21.41:/findata01  380G  379G  1.9G 100% /findata01

10.10.22.42:/findata02  380G  199G  182G  53% /findata02

10.10.23.41:/findata03  380G  172G  209G  46% /findata03

10.10.24.42:/findata04  380G  136G  245G  36% /findata04

10.10.21.41:/findata05  380G  127G  254G  34% /findata05

10.10.22.42:/findata06  380G   94G  287G  25% /findata06

10.10.23.41:/findata07  380G   71G  310G  19% /findata07

10.10.24.42:/findata08  380G   56G  325G  15% /findata08

10.10.21.41:/findata09  380G   54G  327G  15% /findata09

10.10.22.42:/findata10  380G   54G  327G  15% /findata10

10.10.23.41:/findata11  380G   55G  326G  15% /findata11

10.10.24.42:/findata12  380G   56G  325G  15% /findata12

10.10.21.41:/findata13  380G   69G  312G  18% /findata13

10.10.22.42:/findata14  380G  119G  262G  32% /findata14

10.10.23.41:/findata15  380G  136G  245G  36% /findata15

10.10.24.42:/findata16  380G  151G  230G  40% /findata16
```

```
10.10.21.41:/finlog01    48G  7.7G   40G  17% /finlog01
10.10.22.42:/finlog02    48G  7.8G   40G  17% /finlog02
10.10.23.41:/finlog03    48G  512K   48G   1% /finlog03
10.10.24.42:/finlog04    48G  512K   48G   1% /finlog04

[root@flex1 ~]# df -h /soe*/
Filesystem              Size  Used Avail Use% Mounted on
10.10.21.41:/soedata01  1.9T  896G 1002G  48% /soedata01
10.10.22.42:/soedata02  1.9T  706G  1.2T  38% /soedata02
10.10.23.41:/soedata03  1.9T  816G  1.1T  43% /soedata03
10.10.24.42:/soedata04  1.9T  535G  1.4T  29% /soedata04
10.10.21.41:/soedata05  1.9T  579G  1.3T  31% /soedata05
10.10.22.42:/soedata06  1.9T  579G  1.3T  31% /soedata06
10.10.23.41:/soedata07  1.9T  585G  1.3T  31% /soedata07
10.10.24.42:/soedata08  1.9T  616G  1.3T  33% /soedata08
10.10.21.41:/soedata09  1.9T  654G  1.3T  35% /soedata09
10.10.22.42:/soedata10  1.9T  680G  1.2T  36% /soedata10
10.10.23.41:/soedata11  1.9T  660G  1.3T  35% /soedata11
10.10.24.42:/soedata12  1.9T  556G  1.4T  30% /soedata12
10.10.21.41:/soedata13  1.9T  519G  1.4T  28% /soedata13
10.10.22.42:/soedata14  1.9T  497G  1.4T  27% /soedata14
10.10.23.41:/soedata15  1.9T  534G  1.4T  29% /soedata15
10.10.24.42:/soedata16  1.9T  494G  1.4T  27% /soedata16
10.10.21.41:/soelog01    95G   27G   69G  28% /soelog01
10.10.22.42:/soelog02    95G   28G   68G  29% /soelog02
10.10.23.41:/soelog03    95G   27G   69G  28% /soelog03
10.10.24.42:/soelog04    95G   26G   70G  28% /soelog04

[root@flex1 ~]# df -h /sh*/
```

```
Filesystem              Size  Used Avail Use% Mounted on
10.10.21.41:/shdata01   973G  330G  644G  34% /shdata01
10.10.22.42:/shdata02   973G  688G  285G  71% /shdata02
10.10.23.41:/shdata03   973G  691G  283G  71% /shdata03
10.10.24.42:/shdata04   973G  306G  668G  32% /shdata04
10.10.21.41:/shdata05   973G  310G  663G  32% /shdata05
10.10.22.42:/shdata06   973G  316G  658G  33% /shdata06
10.10.23.41:/shdata07   973G  319G  654G  33% /shdata07
10.10.24.42:/shdata08   973G  323G  650G  34% /shdata08
10.10.21.41:/shdata09   973G  234G  739G  25% /shdata09
10.10.22.42:/shdata10   973G  220G  754G  23% /shdata10
10.10.23.41:/shdata11   973G  229G  745G  24% /shdata11
10.10.24.42:/shdata12   973G  237G  737G  25% /shdata12
10.10.21.41:/shdata13   973G  240G  733G  25% /shdata13
10.10.22.42:/shdata14   973G  243G  730G  25% /shdata14
10.10.23.41:/shdata15   973G  247G  727G  26% /shdata15
10.10.24.42:/shdata16   973G  252G  722G  26% /shdata16
10.10.21.41:/shlog01     48G   13G   35G  28% /shlog01
10.10.22.42:/shlog02     48G  9.7G   38G  21% /shlog02
10.10.23.41:/shlog03     48G  512K   48G   1% /shlog03
10.10.24.42:/shlog04     48G  512K   48G   1% /shlog04
```

By doing this, you can read/write data from/to the file system on all Oracle RAC nodes.

Step 5. When the OS level prerequisites and file systems are configured, you are ready to install the Oracle Grid Infrastructure as grid user. Download the Oracle Database 21c (21.3.0.0.0) for Linux x86-64 and the Oracle Database 21c Grid Infrastructure (21.3.0.0.0) for Linux x86-64 software from Oracle Software site. Copy these software binaries to Oracle RAC Node 1 and unzip all files into appropriate directories.

Note: These steps complete the prerequisites for the Oracle Database 21c Installation at OS level on the Oracle RAC Nodes.

Oracle Database 21c GRID Infrastructure Setup

This section describes the high-level steps for the Oracle Database 21c RAC installation. This document provides a partial summary of details that might be relevant.

Note: It is not within the scope of this document to include the specifics of an Oracle RAC installation; you should refer to the Oracle installation documentation for specific installation instructions for your environment. For more information, use this link for Oracle Database 21c install and upgrade guide: https://docs.oracle.com/en/database/oracle/oracle-database/21/cwlin/index.html

For this solution, one shared file system of 200 GB in size was created and shared across all eight Linux nodes for storing OCR and Voting Disk files for all RAC databases. Oracle 19c Release 19.3 Grid Infrastructure (GI) was installed on the first node as a grid user. The installation also configured and added the remaining seven nodes as a part of the GI setup. The Oracle Automatic Storage Management (ASM) was not configured for this deployment.

Complete the following procedures to install the Oracle Grid Infrastructure software for the Oracle Standalone Cluster.

Procedure 1. Create Directory Structure

Step 1. Download and copy the Oracle Grid Infrastructure image files to the first local node only. During installation, the software is copied and installed on all other nodes in the cluster.

Step 2. Create the directory structure according to your environment and run the following commands:

For example:

```
mkdir -p /u01/app/grid
mkdir -p /u01/app/21.3.0/grid
mkdir -p /u01/app/oraInventory
mkdir -p /u01/app/oracle/product/21.3.0/dbhome_1

chown -R grid:oinstall /u01/app/grid
chown -R grid:oinstall /u01/app/21.3.0/grid
chown -R grid:oinstall /u01/app/oraInventory
chown -R oracle:oinstall /u01/app/oracle
```

Step 3. As the grid user, download the Oracle Grid Infrastructure image files and extract the files into the Grid home:

```
cd /u01/app/21.3.0/grid
unzip -q <download_location>/LINUX.X64_213000_grid_home.zip
```

Procedure 2. Configure HugePages

HugePages is a method to have a larger page size that is useful for working with a very large memory. For Oracle Databases, using HugePages reduces the operating system maintenance of page states, and increases Translation Lookaside Buffer (TLB) hit ratio.

Advantage of HugePages:

- HugePages are not swappable so there is no page-in/page-out mechanism overhead.
- HugePages uses fewer pages to cover the physical address space, so the size of " bookkeeping" (mapping from the virtual to the physical address) decreases, so it requires fewer entries in the TLB and so TLB hit ratio improves.

- HugePages reduces page table overhead. Also, HugePages eliminates page table lookup overhead: Since the pages are not subject to replacement, page table lookups are not required.
- Faster overall memory performance: On virtual memory systems, each memory operation is two abstract memory operations. Since there are fewer pages to work on, the possible bottleneck on page table access is avoided.

Note: For this configuration, HugePages were used for all the OLTP and DSS workloads. Refer to the Oracle guidelines to configure HugePages: https://docs.oracle.com/en/database/oracle/oracle-database/21/ladbi/disabling-transparent-hugepages.html

Procedure 3. Run Cluster Verification Utility

This procedure verifies that all the prerequisites are met to install the Oracle Grid Infrastructure software. Oracle Grid Infrastructure ships with the Cluster Verification Utility (CVU) that can run to validate the pre and post installation configurations.

Step 1. Login as Grid User in Oracle RAC Node 1 and go to the directory where the Oracle Grid software binaries are located. Run the script named "runcluvfy.sh" as follows:

```
./runcluvfy.sh stage -pre crsinst -n flex1,flex2,flex3,flex4,flex5,flex6,flex7,flex8 –verbose
```

After the configuration, you are ready to install the Oracle Grid Infrastructure and Oracle Database 21c software.

Note: For this solution, Oracle home binaries were installed on the local virtual disk of the nodes. The OCR, Data, and Redo Log files reside in the shared NFS configured on NetApp Storage array.

Install and Configure Oracle Database Grid Infrastructure Software

Note: It is not within the scope of this document to include the specifics of an Oracle RAC installation. However, a partial summary of details is provided that might be relevant. Please refer to the Oracle installation documentation for specific installation instructions for your environment.

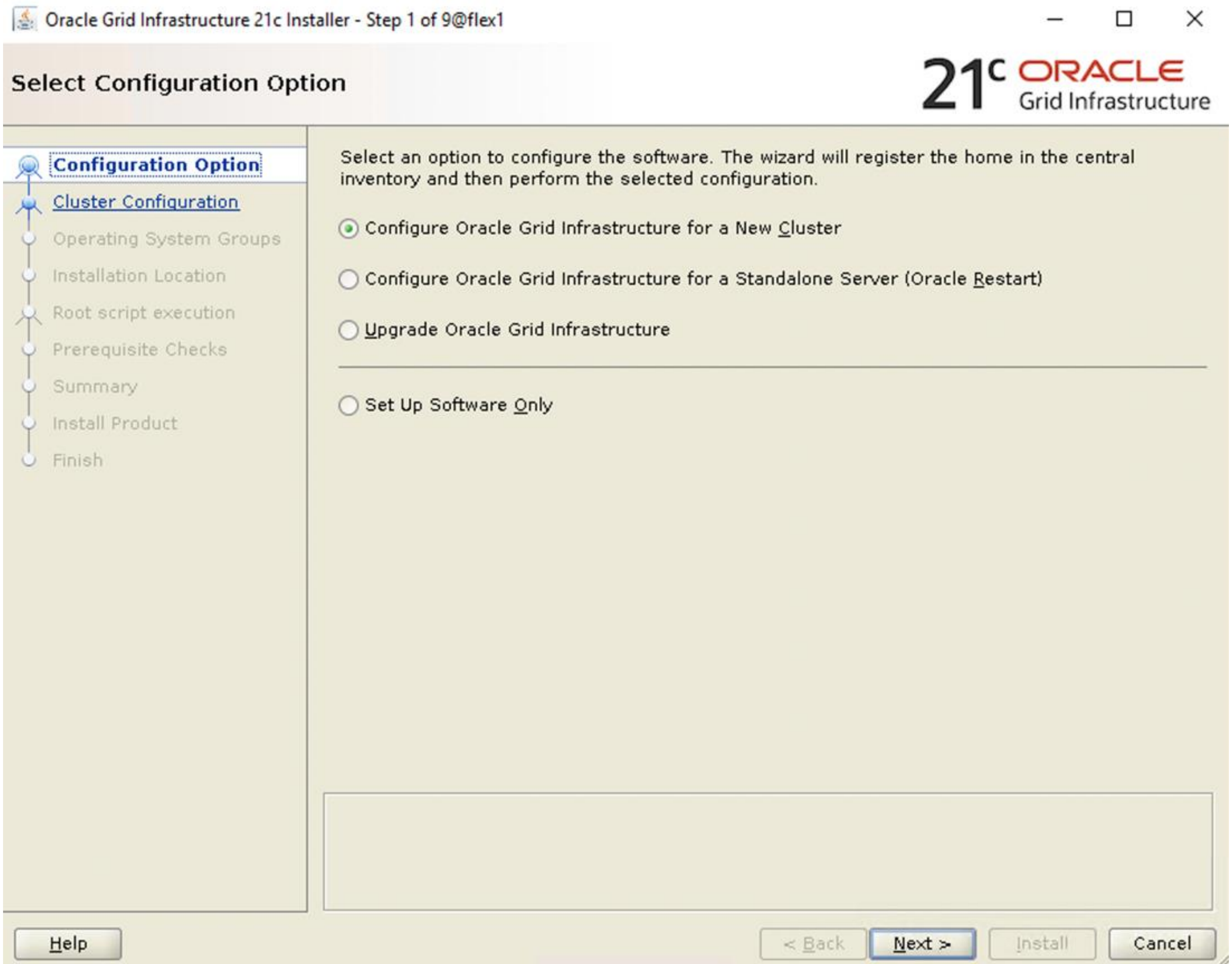
Procedure 1. Install and configure the Oracle Database Grid Infrastructure software

Step 1. Go to the Grid home where the Oracle 21c Grid Infrastructure software binaries are located and launch the installer as the "grid" user.

Step 2. Start the Oracle Grid Infrastructure installer by running the following command:

```
./gridSetup.sh
```

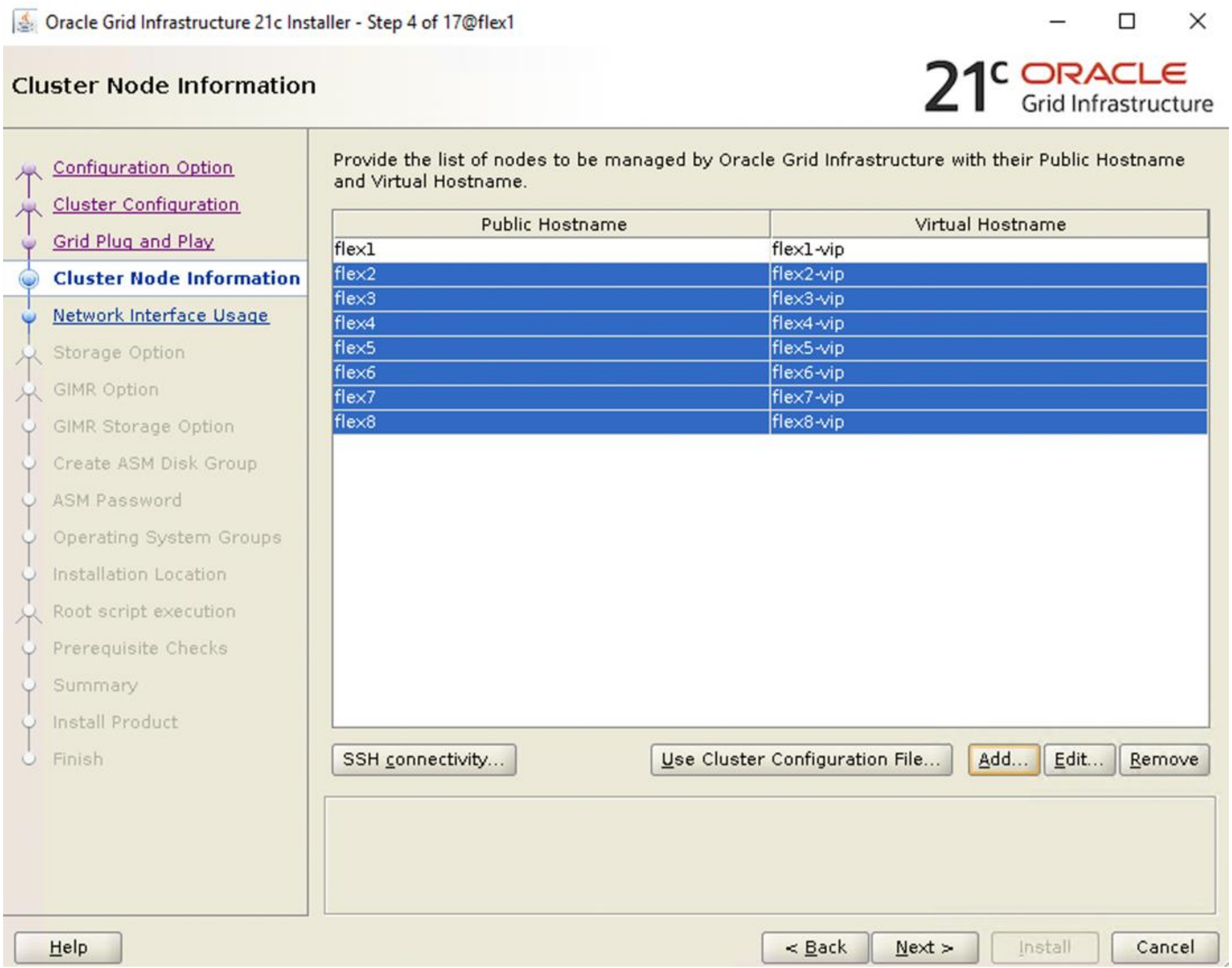
Step 3. Select the option "Configure Oracle Grid Infrastructure for a New Cluster," then click Next.



Step 4. For the Cluster Configuration select “Configure an Oracle Standalone Cluster,” then click Next.

Step 5. In next window, enter the Cluster Name and SCAN Name fields. Enter the names for your cluster and cluster scan that are unique throughout your entire enterprise network. You can also select to Configure GNS if you have configured your domain name server (DNS) to send to the GNS virtual IP address name resolution requests.

Step 6. In the Cluster node information window, click Add to add all eight nodes, Public Hostname and Virtual Host-name as shown below:

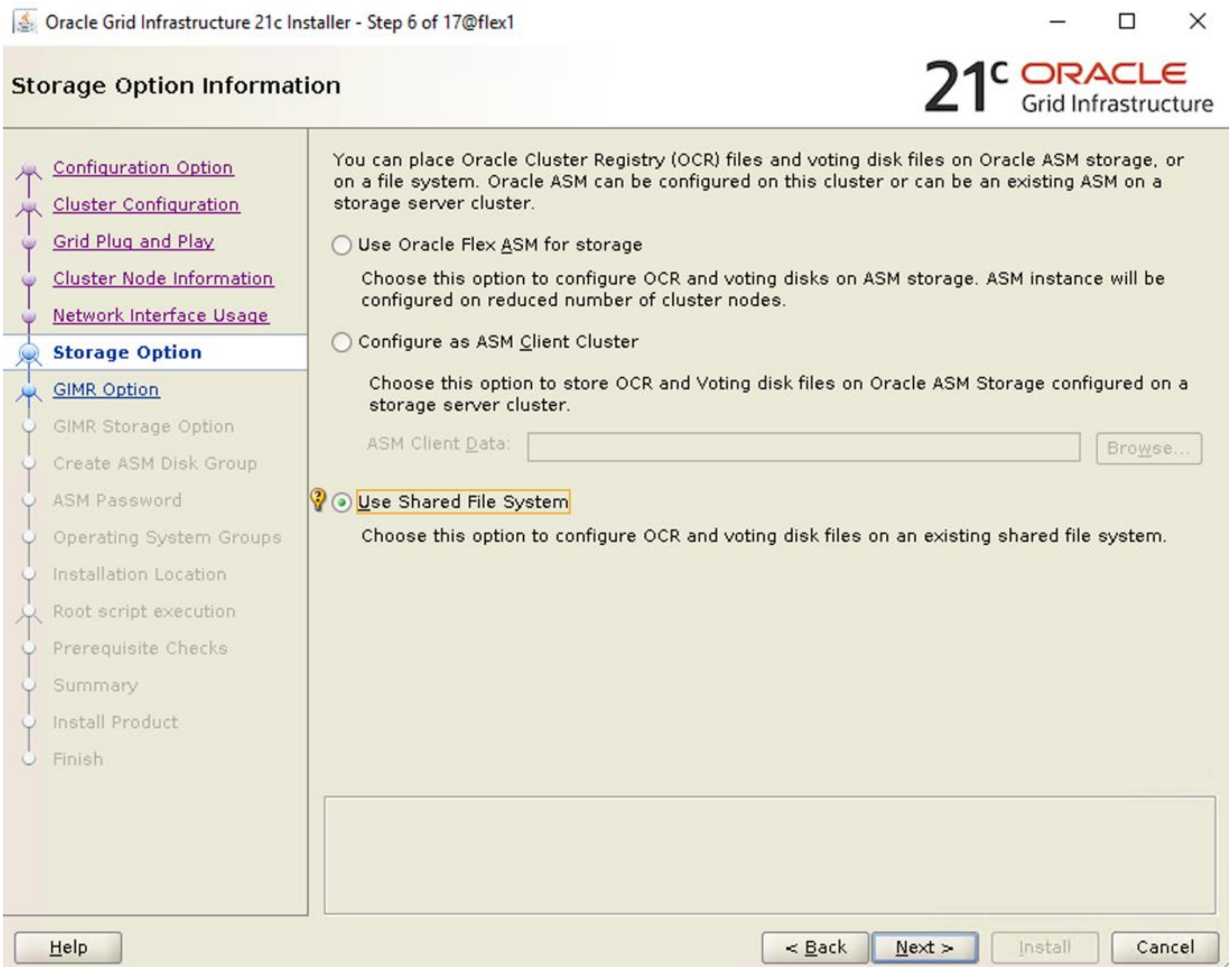


Step 7. As shown above, you will see all nodes listed in the table of cluster nodes. Click the SSH Connectivity. Enter the operating system username and password for the Oracle software owner (grid). Click Setup.

Step 8. A message window appears, indicating that it might take several minutes to configure SSH connectivity between the nodes. After some time, another message window appears indicating that password-less SSH connectivity has been established between the cluster nodes. Click OK to continue.

Step 9. In the Network Interface Usage screen, select the usage type for each network interface for Public and Private Network Traffic and click Next.

Step 10. In the storage option, select the option "Use Shared File System" then click Next.



Note: For this solution, the Grid Infrastructure software was deployed on a shared file system without ASM.

Step 11. In the Create GIMR Option, select the appropriate GIMR option depending upon your environments.

Step 12. In the Shared File System Storage Option window, select the shared OCR File Location and Voting Disk File Location on shared NFS volume, previously configured, for storing OCR and Voting disk files.

Step 13. Select “Do not use Intelligent Platform Management Interface (IPMI).” Click Next.

Step 14. You can configure to have this instance of the Oracle Grid Infrastructure and Oracle Automatic Storage Management to be managed by Enterprise Manager Cloud Control. For this solution, this option was not selected. You can choose to set it up according to your requirements.

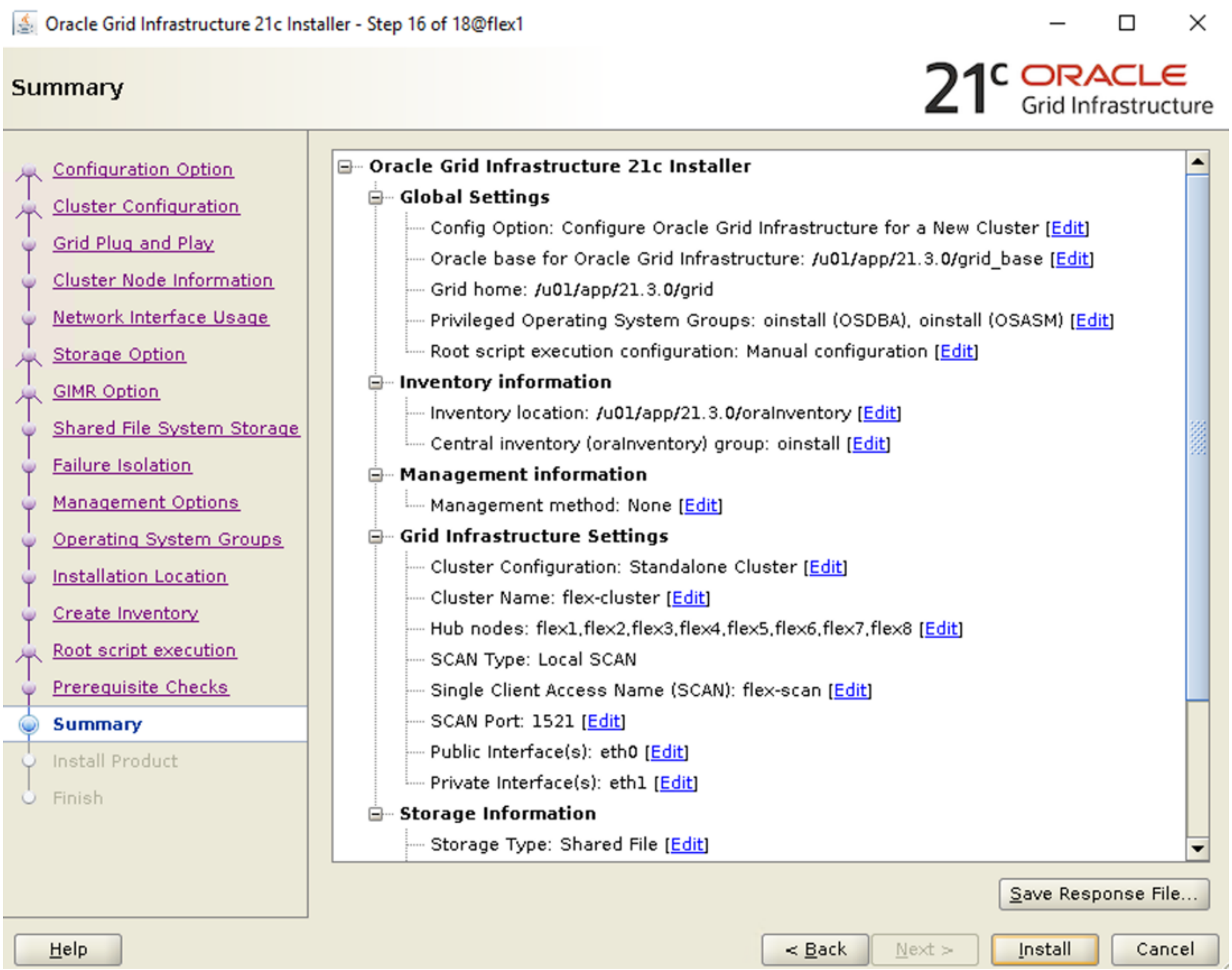
Step 15. Select the appropriate operating system group names for Oracle ASM according to your environments.

Step 16. Specify the Oracle base and inventory directory to use for the Oracle Grid Infrastructure installation and then click Next. The Oracle base directory must be different from the Oracle home directory. Click Next and select the Inventory Directory according to your setup.

Step 17. Click Automatically run configuration scripts to run scripts automatically and enter the relevant root user credentials. Click Next

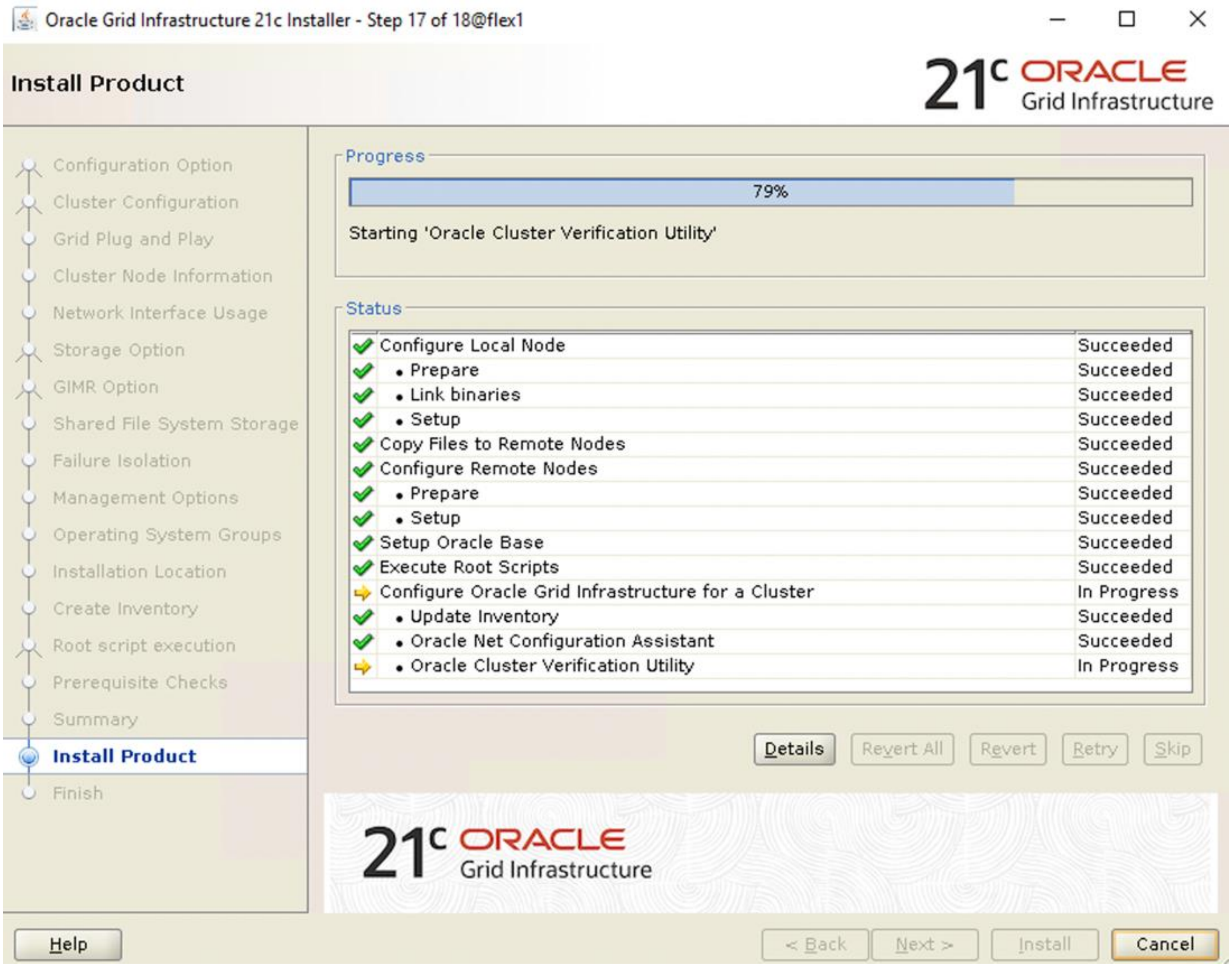
Step 18. Wait while the prerequisite checks complete. If you have any issues, click the " Fix & Check Again" . If any of the checks have a status of Failed and are not fixable, then you must manually correct these issues. After you have fixed the issue, you can click Check Again to have the installer check the requirement and update the status. Repeat as needed until all the checks have a status of Succeeded. Click Next

Step 19. Review the contents of the Summary window and then click Install. The installer displays a progress indicator enabling you to monitor the installation process.



Step 20. Select the password for the Oracle ASM SYS and ASMSNMP account, then click Next.

Step 21. Wait for the grid installer configuration assistants to complete.



Step 22. When the configuration completes successfully, click Close to finish, and exit the grid installer.

Step 23. When the GRID installation is successful, login to each of the nodes and perform the minimum health checks to make sure that the Cluster state is healthy. After your Oracle Grid Infrastructure installation is complete, you can install Oracle Database on a cluster.

Oracle Database Installation

After successfully installing the Oracle GRID, it's recommended to only install the Oracle Database 19c software. You can create databases using DBCA or database creation scripts at later stage.

It is not within the scope of this document to include the specifics of an Oracle RAC database installation. However, a partial summary of details is provided that might be relevant. Please refer to the Oracle database installation documentation for specific installation instructions for your environment here: https://docs.oracle.com/en/database/oracle/oracle-database/21/ladbi/index.html

Procedure 1. Install Oracle database software

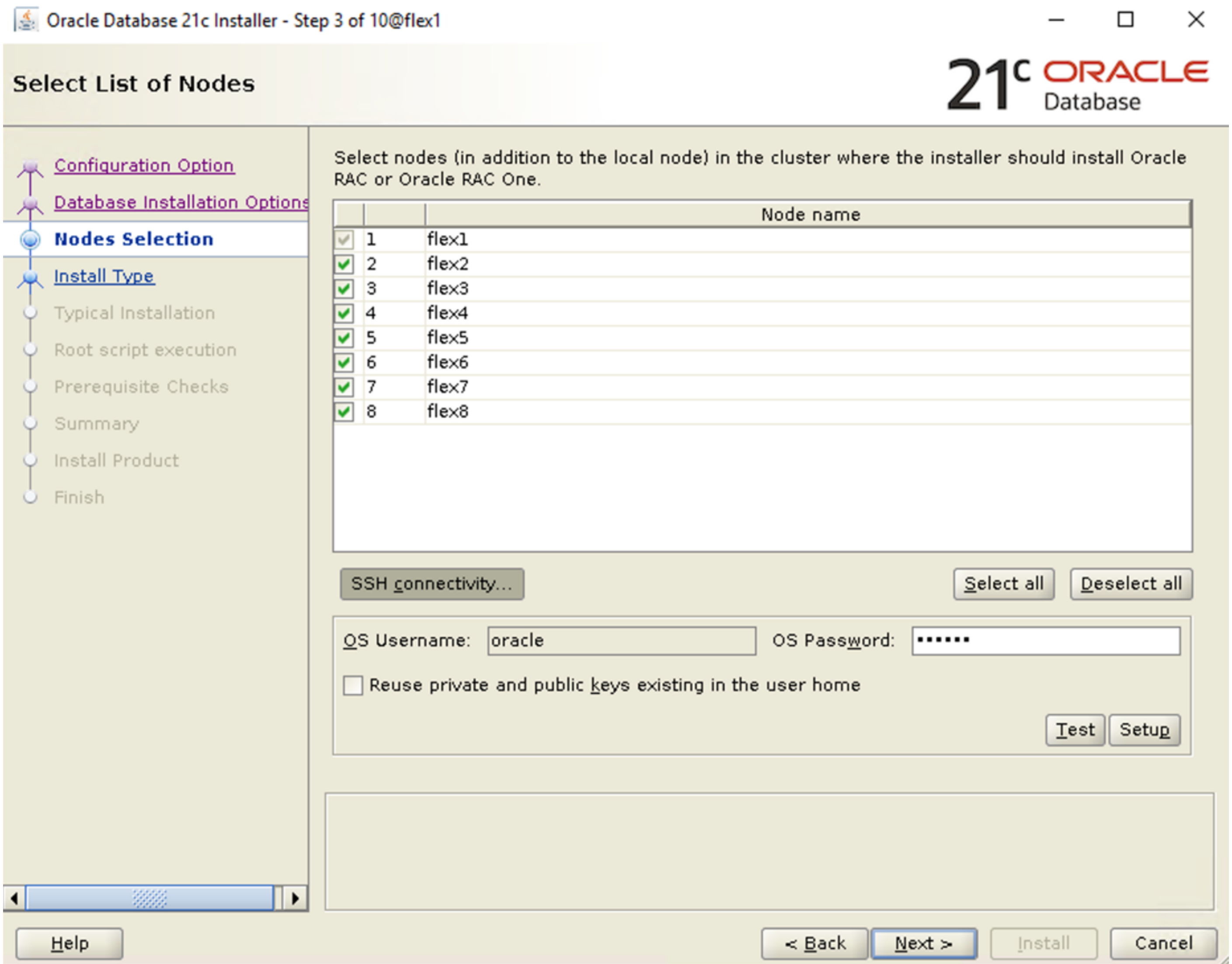
Complete the following steps as an “oracle” user:

Step 1. Start the “./runInstaller” command from the Oracle Database 21c installation media where the Oracle database software is located.

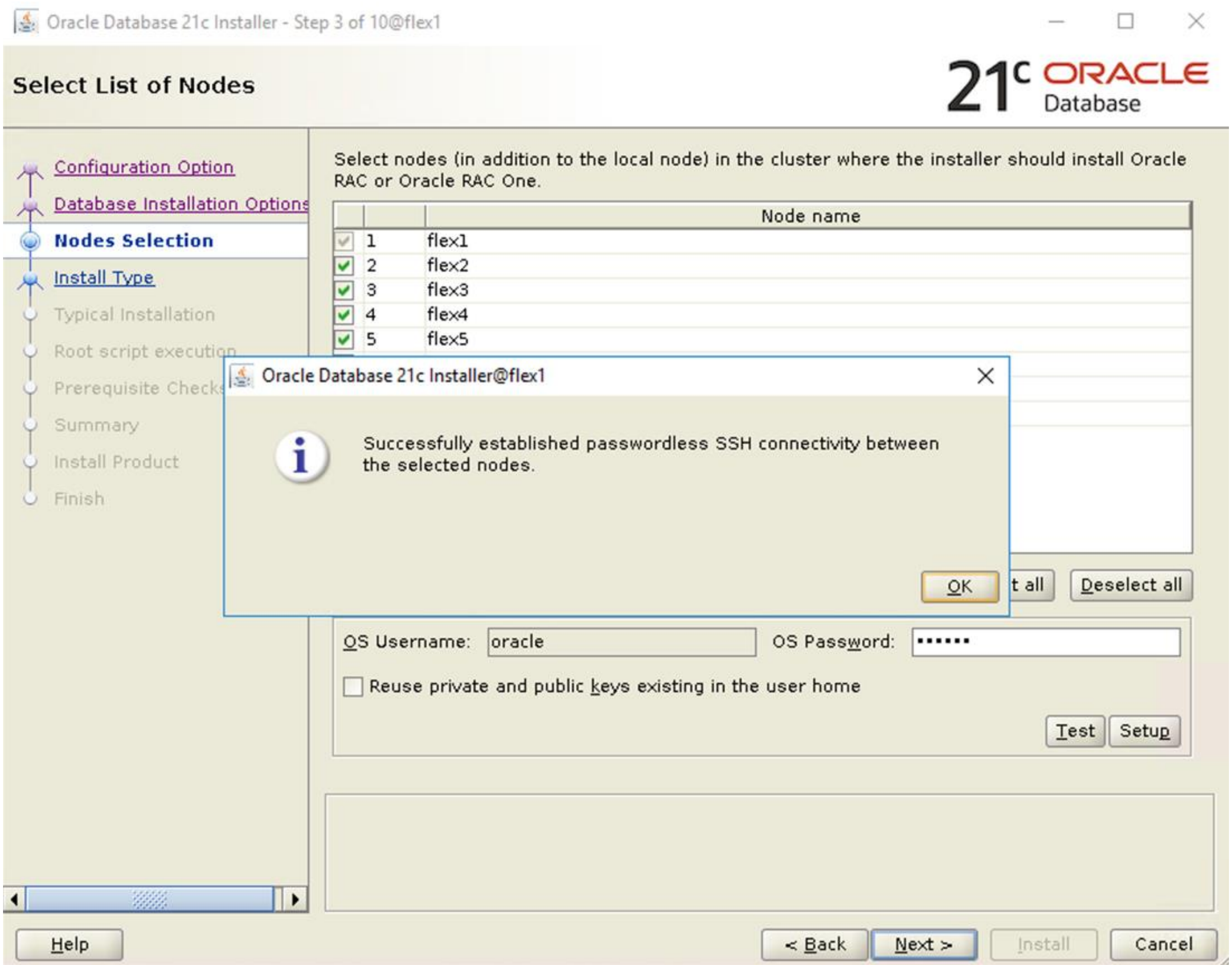
Step 2. Select the option “Set Up Software Only” into configuration Option.

Step 3. Select the option " Oracle Real Application Clusters database installation" and click Next.

Step 4. Select the nodes in the cluster where the installer should install Oracle RAC. For this setup, install the software on all eight nodes as shown below:



Step 5. Click " SSH Connectivity..." and enter the password for the " oracle" user. Click Setup to configure passwordless SSH connectivity and click Test to test it when it is complete. When the test is complete, click Next.



Step 6. Select the Database Edition Options according to your environments and then click Next.

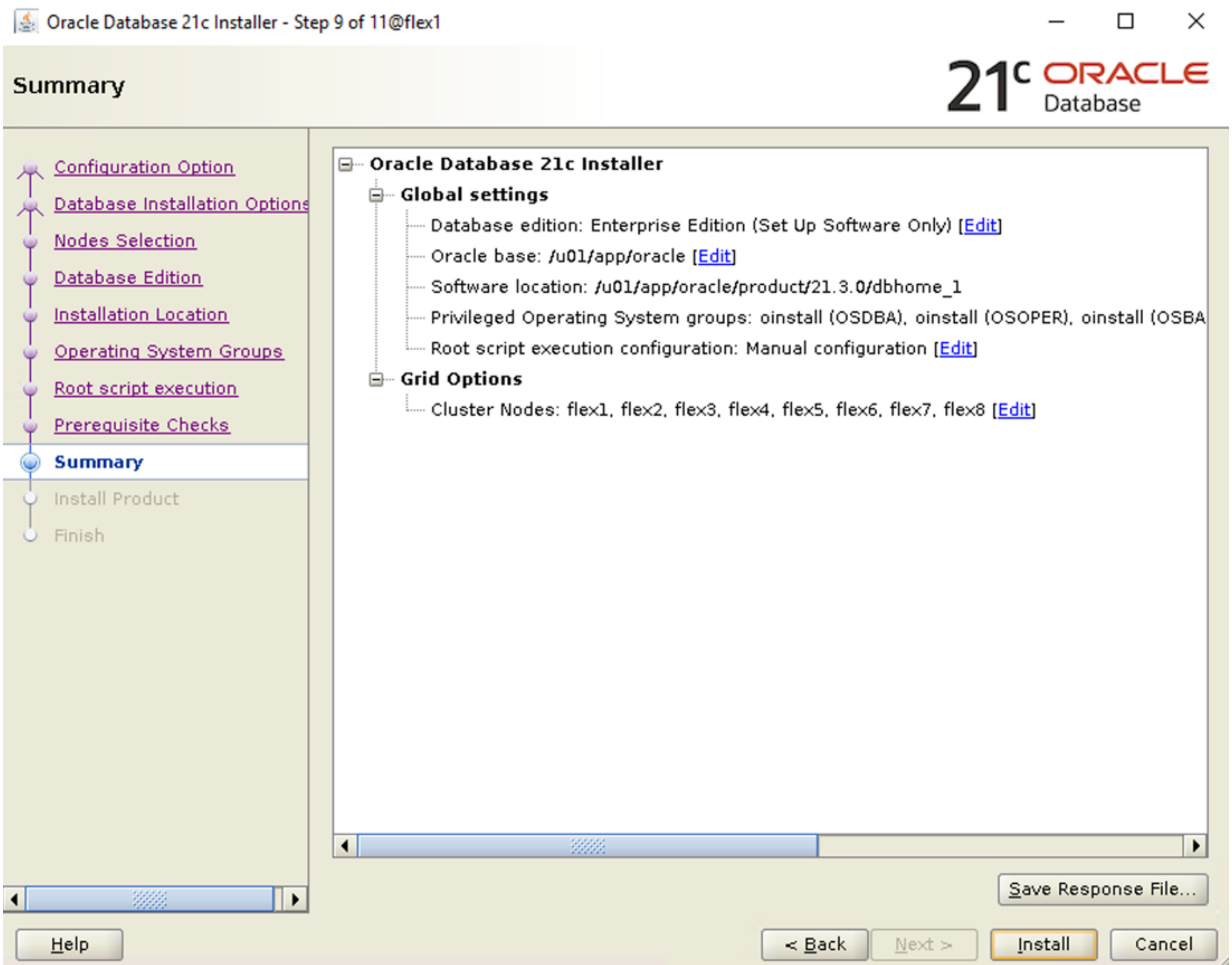
Step 7. Enter the appropriate Oracle Base, then click Next.

Step 8. Select the desired operating system groups and then click Next.

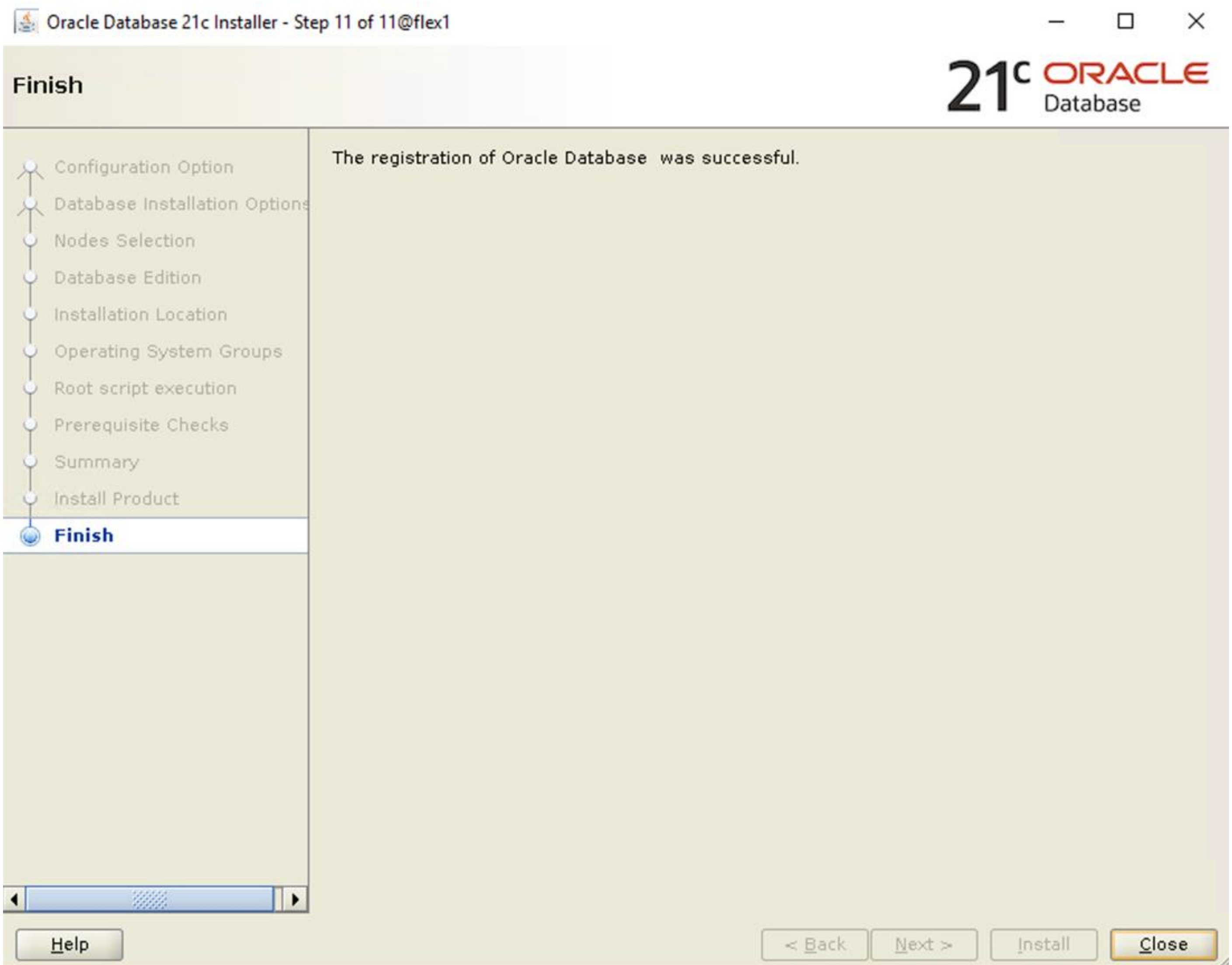
Step 9. Select the option Automatically run configuration script from the option Root script execution menu and click Next.

Step 10. Wait for the prerequisite check to complete. If there are any problems, click " Fix & Check Again" or try to fix those by checking and manually installing required packages. Click Next.

Step 11. Verify the Oracle Database summary information and then click Install.



Step 12. Wait for the installation of Oracle Database finish successfully, then click Close to exit of the installer.

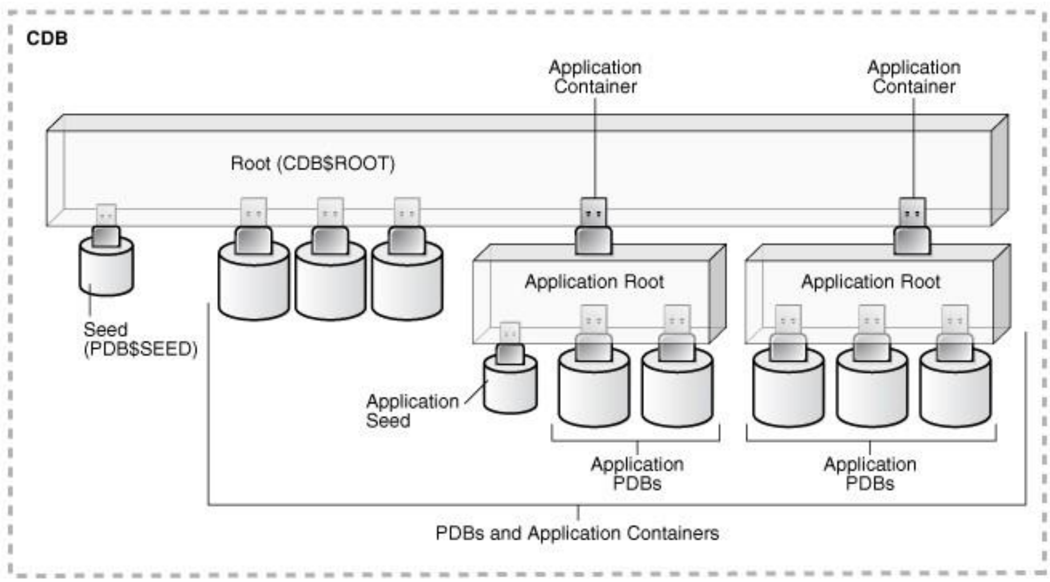


These steps complete the installation of the Oracle 21c Grid Infrastructure and Oracle 21c Database software.

Oracle Database Multitenant Architecture

The multitenant architecture enables an Oracle database to function as a multitenant container database (CDB). A CDB includes zero, one, or many customer-created pluggable databases (PDBs). A PDB is a portable collection of schemas, schema objects, and non-schema objects that appears to an Oracle Net client as a non-CDB. All Oracle databases before Oracle Database 12c were non-CDBs.

A container is logical collection of data or metadata within the multitenant architecture. The following figure represents possible containers in a CDB:



The multitenant architecture solves several problems posed by the traditional non-CDB architecture. Large enterprises may use hundreds or thousands of databases. Often these databases run on different platforms on multiple physical servers. Because of improvements in hardware technology, especially the increase in the number of CPUs, servers can handle heavier workloads than before. A database may use only a fraction of the server hardware capacity. This approach wastes both hardware and human resources. Database consolidation is the process of consolidating data from multiple databases into one database on one computer. The Oracle Multitenant option enables you to consolidate data and code without altering existing schemas or applications.

For more information on Oracle Database Multitenant Architecture, go to: https://docs.oracle.com/en/database/oracle/oracle-database/21/cncpt/CDBs-and-PDBs.html#GUID-5C339A60-2163-4ECE-B7A9-4D67D3D894FB

Now you are ready to run synthetic IO tests against this infrastructure setup. “fio” was used as primary tools for IO tests.

Note: You will configure Direct NFS as you get into the actual database testing with SLOB and Swingbench later in this document.

Scalability Test and Results

This chapter contains the following:

Before configuring a database for workload tests, it is extremely important to validate that this is indeed a balanced configuration that can deliver expected performance. In this solution, node and user scalability will be tested and validated on all 8 node Oracle RAC Databases with various database benchmarking tools.

Hardware Calibration Test using FIO

FIO is short for Flexible IO, a versatile IO workload generator. FIO is a tool that will spawn a number of threads or processes doing a particular type of I/O action as specified by the user. For this solution, FIO is used to measure the performance of a NetApp storage device over a given period. For the FIO Tests, 8 volumes of each 2TB in size were created and all volumes were distributed across both aggregates and thus both storage controllers. These 8 volumes were mounted on each Linux nodes and the FIO tests were run together on each node to perform IO operation as recorded below.

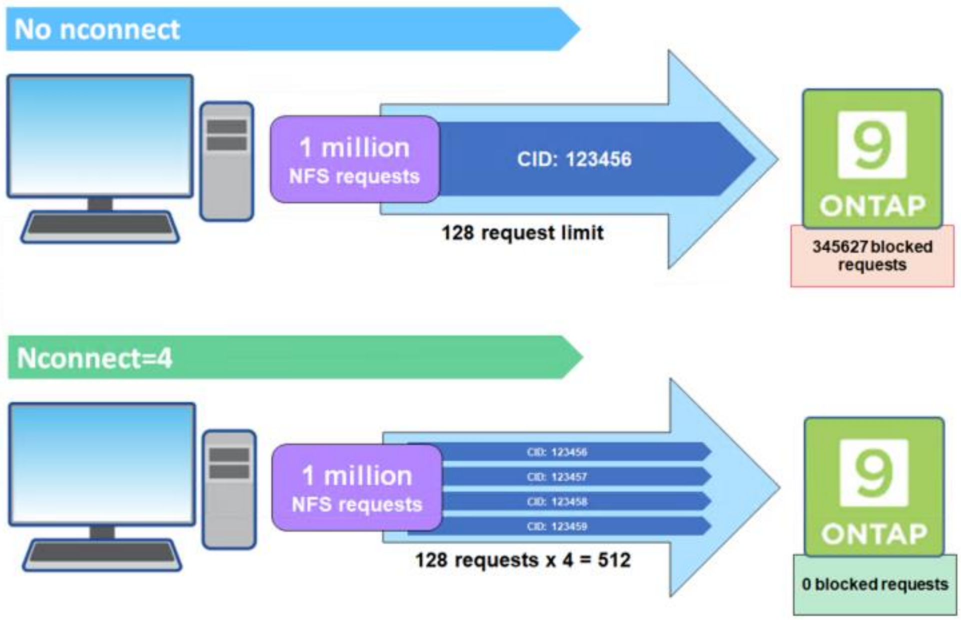
Various FIO tests for measuring IOPS, Latency and Throughput performance of this solution were run by changing block size parameter into the FIO test. For each FIO test, the read/write ratio as 0/100 % read/write, 50/50 % read/write, 70/30 % read/write, 90/10 % read/write and 100/0 % read/write were changed to scale the performance of the system. The tests were run for at least 4 hours to help ensure that this configuration can sustain this type of load for longer period.

The following is the sample “/etc/fstab” file shows FIO file systems mounted in Linux host on all nodes:

```
[root@flex1 4-VLAN-Test]# cat /etc/fstab

# /etc/fstab

# Created by anaconda on Fri Jan 13 19:58:12 2023
```

```
/dev/mapper/ol-root     /                       xfs     defaults        0 0

UUID=2300cce7-826b-48d8-9540-c9d4fc6c733e /boot                   xfs     defaults        0
0

UUID=7D1B-6D3C          /boot/efi               vfat    umask=0077,shortname=winnt 0 2

/dev/mapper/ol-swap     none                    swap    defaults        0 0

10.10.21.41:/fiodata1   /fiodata1       nfs
rw,bg,hard,rsize=524288,wsize=524288,nfsvers=3,actimeo=0,nointr,timeo=600,tcp,nconnect=16

10.10.22.41:/fiodata3   /fiodata3       nfs
rw,bg,hard,rsize=524288,wsize=524288,nfsvers=3,actimeo=0,nointr,timeo=600,tcp,nconnect=16

10.10.21.41:/fiodata5   /fiodata5       nfs
rw,bg,hard,rsize=524288,wsize=524288,nfsvers=3,actimeo=0,nointr,timeo=600,tcp,nconnect=16

10.10.22.41:/fiodata7   /fiodata7       nfs
rw,bg,hard,rsize=524288,wsize=524288,nfsvers=3,actimeo=0,nointr,timeo=600,tcp,nconnect=16

10.10.23.41:/fiodata2   /fiodata2       nfs
rw,bg,hard,rsize=524288,wsize=524288,nfsvers=3,actimeo=0,nointr,timeo=600,tcp,nconnect=16

10.10.24.41:/fiodata4   /fiodata4       nfs
rw,bg,hard,rsize=524288,wsize=524288,nfsvers=3,actimeo=0,nointr,timeo=600,tcp,nconnect=16

10.10.23.41:/fiodata6   /fiodata6       nfs
rw,bg,hard,rsize=524288,wsize=524288,nfsvers=3,actimeo=0,nointr,timeo=600,tcp,nconnect=16

10.10.24.41:/fiodata8   /fiodata8       nfs
rw,bg,hard,rsize=524288,wsize=524288,nfsvers=3,actimeo=0,nointr,timeo=600,tcp,nconnect=16
```

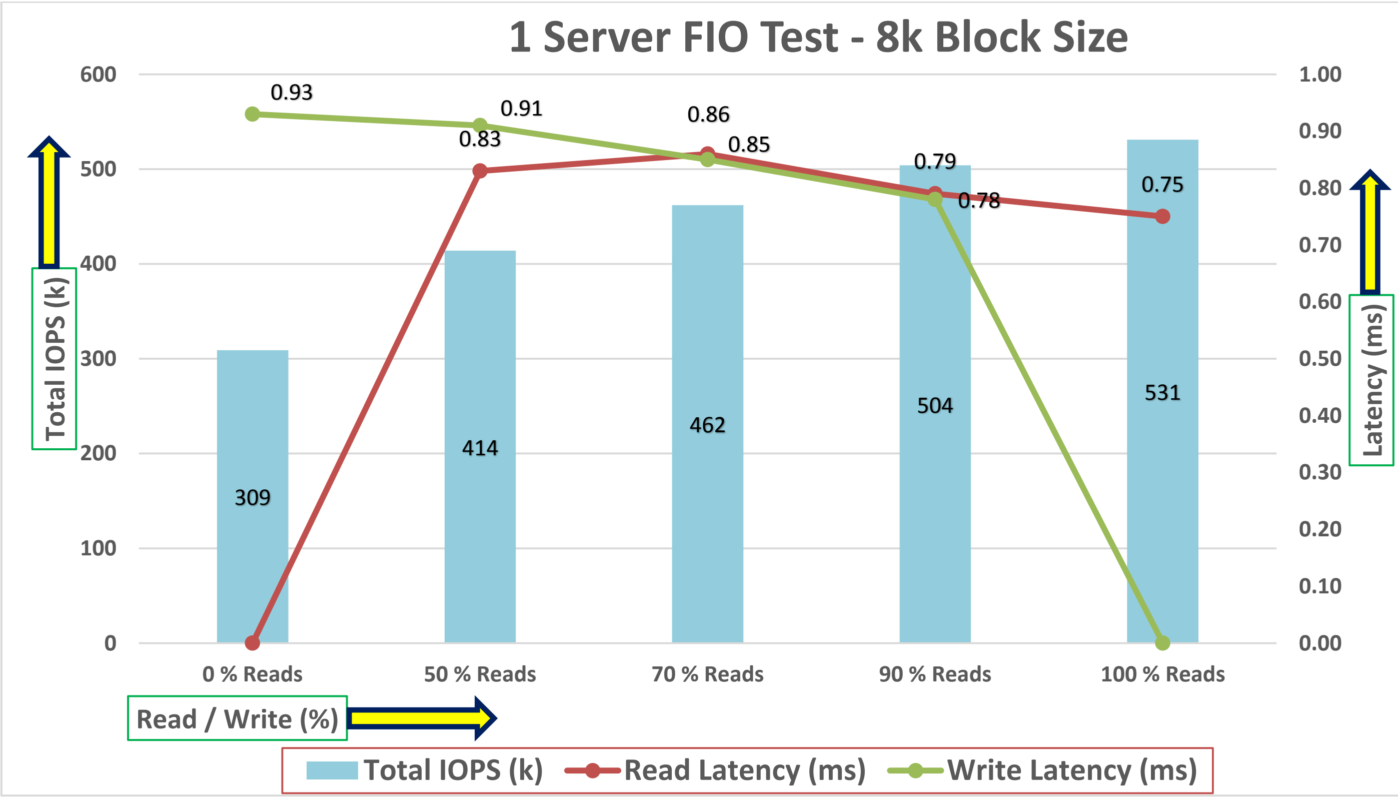
Note: We used “nconnect” parameter here to provide multiple transport connections per TCP connection or mount point. nconnect is designed to allocate more sessions across a single TCP connection as shown below. We only used this parameter for validating various FIO benchmark exercises.



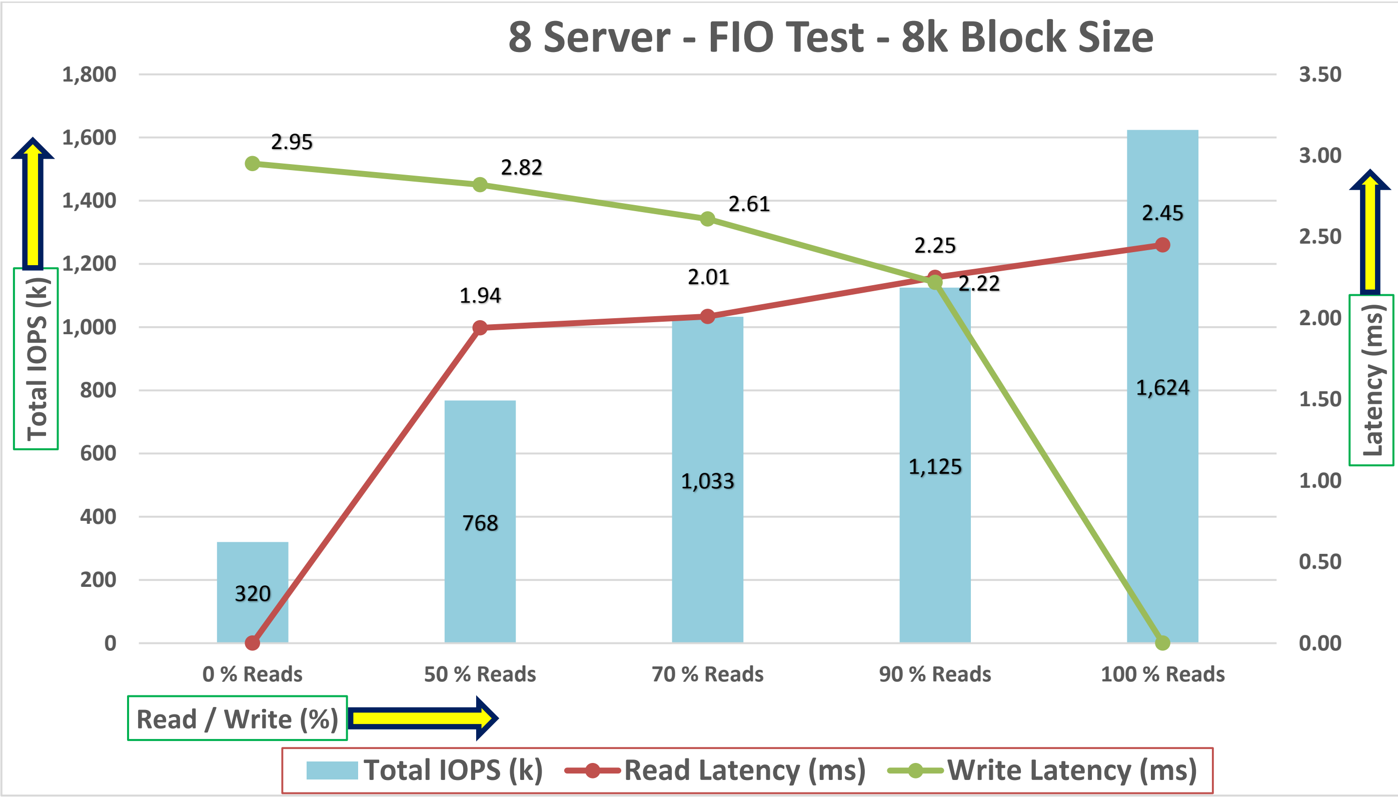
This “nconnect” parameter helps to better distribute NFS workloads and add some parallelism to the connection, which helps the NFS server handle. Refer to the NetApp NFS Best Practices documentation for more information: https://www.netapp.com/media/10720-tr-4067.pdf

IOPS Tests

Random read/write FIO test for the 8k block size representing OLTP type of workloads were run on a single node server as shown in the chart below.



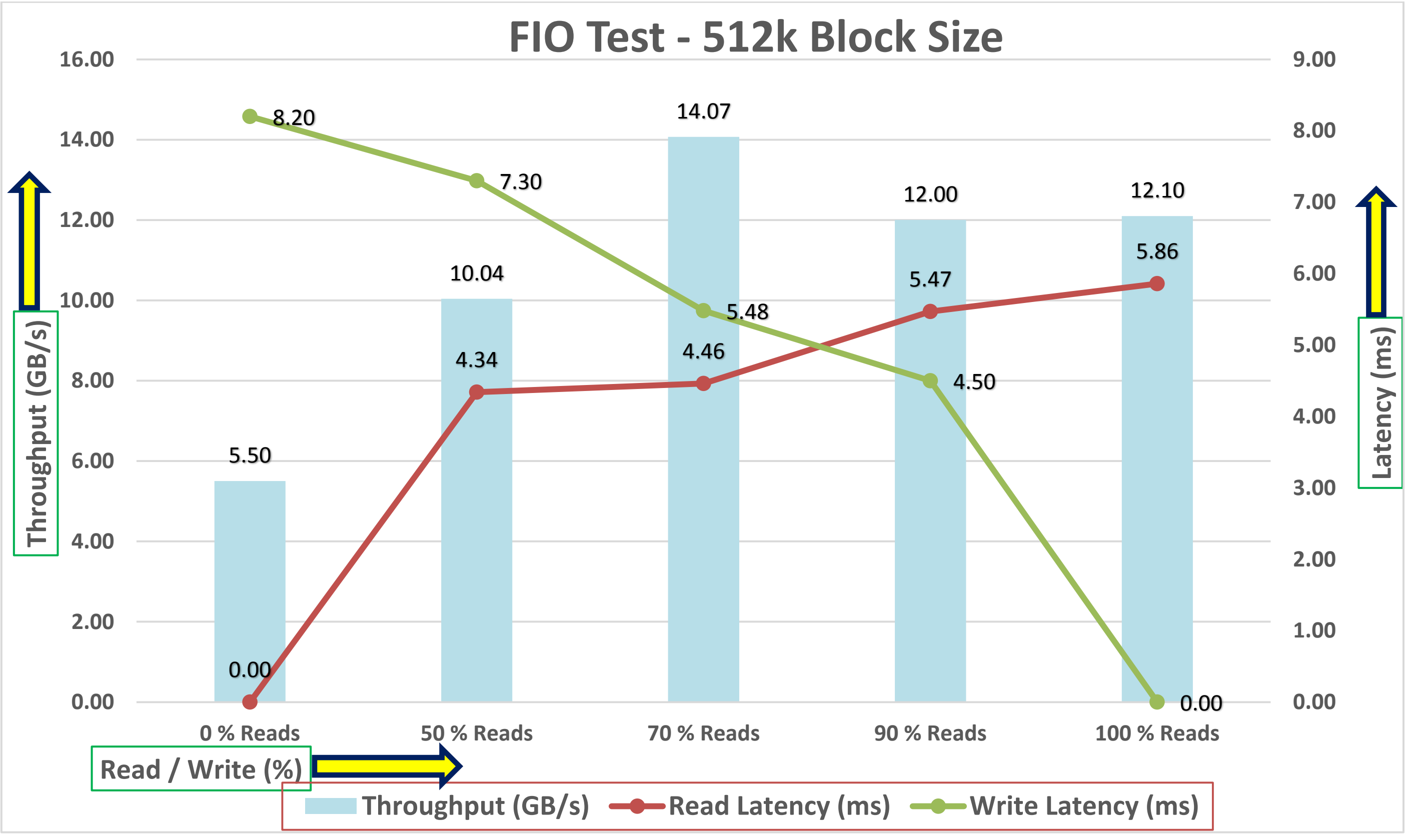
The chart below shows results for the random read/write FIO tests for the 8k block size representing OLTP type of workloads across all eight server nodes.



For the 100/0 % read/write test, we achieved around 1624k IOPS with the read latency around 2.45 millisecond. Similarly, for the 90/10 % read/write test, we achieved around 1125 IOPS with the read latency around 2.2 millisecond and the write latency around 2.2 millisecond. For the 70/30 % read/write test, we achieved around 1033k IOPS with the read latency around 2 millisecond and the write latency around 2.6 millisecond. For the 50/50 % read/write test, we achieved around 768k IOPS with the read latency around 1.9 millisecond and the write latency around 2.8 millisecond. For the 0/100 % read/write test, we achieved around 320k IOPS with the write latency around 2.9 millisecond. Reads and writes consume system resources differently.

Bandwidth Tests

The bandwidth tests are carried out with 512k IO Size and represents the DSS database type workloads. The chart below shows results for the sequential read/write FIO test for the 512k block size.



For the 100/0 % read/write test, we achieved around 12.1 GB/s throughput with the read latency around 5.8 millisecond. Similarly, for the 90/10 % read/write test, we achieved around 12 GB/s throughput with the read latency around 4.5 millisecond and the write latency around 5.4 millisecond. For the 70/30 % read/write test, we achieved around 14 GB/s throughput with the read latency around 4.3 millisecond and the write latency around 5.4 millisecond. For the 50/50 % read/write test, we achieved around 10 GB/s throughput with the read latency around 4.3 millisecond and the write latency around 7.3 millisecond. For the 0/100 % read/write test, we achieved around 5.5 GB/s throughput with the write latency around 8.2 millisecond.

The system under test benefited from slightly better resource distribution in the 70/30 R/W test, resulting in slightly improved peak IOPS in this test compared with the 90/10 and 100/0 R/W test. We did not see any performance dips or degradation over the period of run time. It is also important to note that this is not a benchmarking exercise, and these are practical and out of box test numbers that can be easily reproduced by anyone. At this time, we are ready to create OLTP database(s) and continue with database tests.

Database Creation with DBCA

We used Oracle Database Configuration Assistant (DBCA) to create multiple OLTP and DSS databases for SLOB and SwingBench test calibration. For SLOB Tests, we configured one container database as “FINCDB” and under this container, we create one pluggable database as “FINPDB.” For SwingBench SOE (OLTP type) workload tests, we configured one container database as “SOECDB” and under this container, we create two pluggable databases as “SOEPDB” and “ENGPDB” to demonstrate the system scalability running one OLTP and multiple OLTP databases for various SOE workloads. For SwingBench SH (DSS type) workload tests, we configured one container database as “SHCDB” and under this container, we created one pluggable database as “SHPDB.” Alternatively, you can use Database creation scripts to create the databases as well.

For all the database deployment, we have configured two aggregates (one aggregate on each storage node) into a single SVM(ORANFS-SVM), and each aggregate contains 11 SSD (1.75 TB Each) drives that were subdivided into RAID DP groups, plus one spare drive as explained earlier in the storage configuration section.

For each RAC database, we have created total number of 20 file system volumes and these volumes were shared and mounted across all 8 RAC nodes. For each RAC databases, we used 16 file system volumes to store the “data” and 4 file system volumes to store the “log” files for the databases. We distributed equal number of volumes on the storage nodes by placing those volumes equally into both the aggregates. All the database files were also spread evenly across the two nodes of the storage system so that each storage node served data for the databases.

The following storage commands lists all the volumes and storage configuration used in this solution:

```
FlexPod-A800::> volume show -vserver ORANFS-SVM

Vserver   Volume       Aggregate    State      Type       Size  Available Used%

--------- ------------ ------------ ---------- ---- ---------- ---------- -----

ORANFS-SVM ORANFSSVM_root aggr1_node1 online   RW          1GB    972.2MB    0%

ORANFS-SVM ocrvote      aggr1_node1  online     RW        200GB    189.8GB    0%

ORANFS-SVM fiodata1     aggr1_node1  online     RW          2TB     1.11TB   41%

ORANFS-SVM fiodata2     aggr1_node2  online     RW          2TB     1.11TB   41%

ORANFS-SVM fiodata3     aggr1_node1  online     RW          2TB     1.11TB   41%

ORANFS-SVM fiodata4     aggr1_node2  online     RW          2TB     1.11TB   41%

ORANFS-SVM fiodata5     aggr1_node1  online     RW          2TB     1.11TB   41%

ORANFS-SVM fiodata6     aggr1_node2  online     RW          2TB     1.11TB   41%

ORANFS-SVM fiodata7     aggr1_node1  online     RW          2TB     1.11TB   41%

ORANFS-SVM fiodata8     aggr1_node2  online     RW          2TB     1.11TB   41%

ORANFS-SVM findata01    aggr1_node1  online     RW        400GB     1.87GB   99%

ORANFS-SVM findata02    aggr1_node2  online     RW        400GB    181.9GB   52%

ORANFS-SVM findata03    aggr1_node1  online     RW        400GB    208.6GB   45%

ORANFS-SVM findata04    aggr1_node2  online     RW        400GB    244.4GB   35%

ORANFS-SVM findata05    aggr1_node1  online     RW        400GB    253.6GB   33%

ORANFS-SVM findata06    aggr1_node2  online     RW        400GB    286.2GB   24%

ORANFS-SVM findata07    aggr1_node1  online     RW        400GB    309.1GB   18%

ORANFS-SVM findata08    aggr1_node2  online     RW        400GB    324.9GB   14%
```

ORANFS-SVM	findata09	aggr1_node1	online	RW	400GB	326.6GB	14%
ORANFS-SVM	findata10	aggr1_node2	online	RW	400GB	326.3GB	14%
ORANFS-SVM	findata11	aggr1_node1	online	RW	400GB	325.9GB	14%
ORANFS-SVM	findata12	aggr1_node2	online	RW	400GB	324.5GB	14%
ORANFS-SVM	findata13	aggr1_node1	online	RW	400GB	311.6GB	17%
ORANFS-SVM	findata14	aggr1_node2	online	RW	400GB	261.8GB	31%
ORANFS-SVM	findata15	aggr1_node1	online	RW	400GB	244.9GB	35%
ORANFS-SVM	findata16	aggr1_node2	online	RW	400GB	229.3GB	39%
ORANFS-SVM	finlog01	aggr1_node1	online	RW	50GB	39.81GB	16%
ORANFS-SVM	finlog02	aggr1_node2	online	RW	50GB	39.70GB	16%
ORANFS-SVM	finlog03	aggr1_node1	online	RW	50GB	47.50GB	0%
ORANFS-SVM	finlog04	aggr1_node2	online	RW	50GB	47.50GB	0%
ORANFS-SVM	soedata01	aggr1_node1	online	RW	1.95TB	1001GB	47%
ORANFS-SVM	soedata02	aggr1_node2	online	RW	1.95TB	1.16TB	37%
ORANFS-SVM	soedata03	aggr1_node1	online	RW	1.95TB	1.06TB	42%
ORANFS-SVM	soedata04	aggr1_node2	online	RW	1.95TB	1.33TB	28%
ORANFS-SVM	soedata05	aggr1_node1	online	RW	1.95TB	1.29TB	30%
ORANFS-SVM	soedata06	aggr1_node2	online	RW	1.95TB	1.29TB	30%
ORANFS-SVM	soedata07	aggr1_node1	online	RW	1.95TB	1.28TB	30%
ORANFS-SVM	soedata08	aggr1_node2	online	RW	1.95TB	1.25TB	32%
ORANFS-SVM	soedata09	aggr1_node1	online	RW	1.95TB	1.21TB	34%
ORANFS-SVM	soedata10	aggr1_node2	online	RW	1.95TB	1.19TB	35%
ORANFS-SVM	soedata11	aggr1_node1	online	RW	1.95TB	1.21TB	34%
ORANFS-SVM	soedata12	aggr1_node2	online	RW	1.95TB	1.31TB	29%
ORANFS-SVM	soedata13	aggr1_node1	online	RW	1.95TB	1.35TB	27%
ORANFS-SVM	soedata14	aggr1_node2	online	RW	1.95TB	1.37TB	26%
ORANFS-SVM	soedata15	aggr1_node1	online	RW	1.95TB	1.33TB	28%
ORANFS-SVM	soedata16	aggr1_node2	online	RW	1.95TB	1.37TB	26%
ORANFS-SVM	soelog01	aggr1_node1	online	RW	100GB	68.68GB	27%

```
ORANFS-SVM soelog02    aggr1_node2  online     RW         100GB    67.73GB   28%
ORANFS-SVM soelog03    aggr1_node1  online     RW         100GB    68.49GB   27%
ORANFS-SVM soelog04    aggr1_node2  online     RW         100GB    69.14GB   27%
ORANFS-SVM shdata01    aggr1_node1  online     RW           1TB    643.2GB   33%
ORANFS-SVM shdata02    aggr1_node2  online     RW           1TB    284.9GB   70%
ORANFS-SVM shdata03    aggr1_node1  online     RW           1TB    282.2GB   70%
ORANFS-SVM shdata04    aggr1_node2  online     RW           1TB    667.4GB   31%
ORANFS-SVM shdata05    aggr1_node1  online     RW           1TB    662.9GB   31%
ORANFS-SVM shdata06    aggr1_node2  online     RW           1TB    657.3GB   32%
ORANFS-SVM shdata07    aggr1_node1  online     RW           1TB    653.9GB   32%
ORANFS-SVM shdata08    aggr1_node2  online     RW           1TB    649.9GB   33%
ORANFS-SVM shdata09    aggr1_node1  online     RW           1TB    739.0GB   24%
ORANFS-SVM shdata10    aggr1_node2  online     RW           1TB    753.0GB   22%
ORANFS-SVM shdata11    aggr1_node1  online     RW           1TB    744.6GB   23%
ORANFS-SVM shdata12    aggr1_node2  online     RW           1TB    736.7GB   24%
ORANFS-SVM shdata13    aggr1_node1  online     RW           1TB    732.8GB   24%
ORANFS-SVM shdata14    aggr1_node2  online     RW           1TB    730.0GB   24%
ORANFS-SVM shdata15    aggr1_node1  online     RW           1TB    726.2GB   25%
ORANFS-SVM shdata16    aggr1_node2  online     RW           1TB    721.1GB   25%
ORANFS-SVM shlog01     aggr1_node1  online     RW          50GB    34.64GB   27%
ORANFS-SVM shlog02     aggr1_node2  online     RW          50GB    37.83GB   20%
ORANFS-SVM shlog03     aggr1_node1  online     RW          50GB    47.50GB    0%
ORANFS-SVM shlog04     aggr1_node2  online     RW          50GB    47.50GB    0%
```

Table 11 lists the database volume configuration for this solution where we deployed all three databases to validate SLOB and SwingBench workloads.

Table 11. Database volume configuration

Database Name	Volume	Size (GB)	Aggregate	Notes
OCRVOTE	ocrvote	100	aggr1_node1	OCR & Voting Disk

Database Name	Volume	Size (GB)	Aggregate	Notes
FINCDB (Container FINCDB with Pluggable Database as FINPDB)	findata01	400	aggr1_node1	SLOB Database Data Files
	findata02	400	aggr1_node2	
	findata03	400	aggr1_node1	
	findata04	400	aggr1_node2	
	findata05	400	aggr1_node1	
	findata06	400	aggr1_node2	
	findata07	400	aggr1_node1	
	findata08	400	aggr1_node2	
	findata09	400	aggr1_node1	
	findata10	400	aggr1_node2	
	findata11	400	aggr1_node1	
	findata12	400	aggr1_node2	
	findata13	400	aggr1_node1	
	findata14	400	aggr1_node2	
	findata15	400	aggr1_node1	
	findata16	400	aggr1_node2	
	finlog01	50	aggr1_node1	SLOB Database Redo Log Files
	finlog02	50	aggr1_node2	
	finlog03	50	aggr1_node1	
	finlog04	50	aggr1_node2	
	soedata01	2000	aggr1_node1	
	soedata02	2000	aggr1_node2	

Database Name	Volume	Size (GB)	Aggregate	Notes
SOECDB (Container SOECDB with Two Pluggable Database as SOEPDB and ENGPDB)	soedata03	2000	aggr1_node1	SOE Database Data Files
	soedata04	2000	aggr1_node2	
	soedata05	2000	aggr1_node1	
	soedata06	2000	aggr1_node2	
	soedata07	2000	aggr1_node1	
	soedata08	2000	aggr1_node2	
	soedata09	2000	aggr1_node1	
	soedata10	2000	aggr1_node2	
	soedata11	2000	aggr1_node1	
	soedata12	2000	aggr1_node2	
	soedata13	2000	aggr1_node1	
	soedata14	2000	aggr1_node2	
	soedata15	2000	aggr1_node1	
	soedata16	2000	aggr1_node2	
	soelog01	100	aggr1_node1	SOE Database Redo Log Files
	soelog02	100	aggr1_node2	
	soelog03	100	aggr1_node1	
	soelog4	100	aggr1_node2	
	shdata01	1000	aggr1_node1	
	shdata02	1000	aggr1_node2	
	shdata03	1000	aggr1_node1	
	shdata04	1000	aggr1_node2	

Database Name	Volume	Size (GB)	Aggregate	Notes
SHCDB (Container SHCDB with One Pluggable Database as SHPDB)	shdata05	1000	aggr1_node1	SH Database Data Files
	shdata06	1000	aggr1_node2	
	shdata07	1000	aggr1_node1	
	shdata08	1000	aggr1_node2	
	shdata09	1000	aggr1_node1	
	shdata10	1000	aggr1_node2	
	shdata11	1000	aggr1_node1	
	shdata12	1000	aggr1_node2	
	shdata13	1000	aggr1_node1	
	shdata14	1000	aggr1_node2	
	shdata15	1000	aggr1_node1	
	shdata16	1000	aggr1_node2	
	shlog01	50	aggr1_node1	SH Database Redo Log Files
	shlog02	50	aggr1_node2	
	shlog03	50	aggr1_node1	
	shlog04	50	aggr1_node2	

We used the widely adopted SLOB and Swingbench database performance test tools to test and validate throughput, IOPS, and latency for various test scenarios as explained below. These databases were configured and run workload after configuring dNFS as explained below.

Oracle dNFS Configuration

We recommend configuring the Oracle Database to access NFS V3 servers directly using an Oracle internal Direct NFS client instead of using the operating system kernel NFS client.

To enable Oracle Database to use Direct NFS Client, the NFS file systems must be mounted and available over regular NFS mounts before you start installation. Direct NFS Client manages settings after installation. If Oracle Database cannot open an NFS server using Direct NFS Client, then Oracle Database uses the platform operating system kernel NFS client. You should still set the kernel mount options as a backup, but for normal operation, Direct NFS Client uses its own NFS client.

Direct NFS Client supports up to four network paths to the NFS server. Direct NFS Client performs load balancing across all specified paths. If a specified path fails, then Direct NFS Client reissues I/O commands over any remaining paths.

Create an "oranfstab" File for Direct NFS Client

Direct NFS uses a configuration file, "oranfstab," to determine the available mount points. Create an "oranfstab" file with appropriate attributes for each NFS server that you want to access using Direct NFS Client according to your environment. Refer to the Oracle documentation for more information: https://docs.oracle.com/en/database/oracle/oracle-database/21/ladbi/creating-an-oranfstab-file-for-direct-nfs-client.html#GUID-C16A1AF8-CCC5-46C2-875E-4276C2CCCF22

If you use Direct NFS Client, then you can use a new file specific for Oracle data file management, "**oranfstab,**" to specify additional options specific for Oracle Database to Direct NFS Client. For example, you can use "**oranfstab**" to specify additional paths for a mount point. You can add the "**oranfstab**" file either to "**/etc**" or to "**$ORACLE_HOME/dbs**"

With shared Oracle homes, when the "**oranfstab**" file is placed in "**$ORACLE_HOME/dbs,**" the entries in the file are specific to a single database. In this case, all nodes running an Oracle RAC database use the same "**$ORACLE_HOME/dbs/oranfstab**" file. In non-shared Oracle RAC installs, "**oranfstab**" must be replicated on all nodes. The "**oranfstab**" configuration in "**$ORACLE_HOME/dbs**" is local to the database under "**$ORACLE_HOME,**" whereas the "**oranfstab**" in "**/etc/oranfstab**" applies to all Oracle databases on that server.

When the "**oranfstab**" file is placed in "**/etc,**" then it is globally available to all Oracle databases and can contain mount points used by all Oracle databases running on nodes in the cluster, including standalone databases. However, on Oracle RAC systems, if the "**oranfstab**" file is placed in "**/etc,**" then you must replicate the file "**/etc/oranfstab**" file on all nodes and keep each "**/etc/oranfstab**" file synchronized on all nodes, just as you must with the "**/etc/fstab**" file.

Note: In all cases, mount points must be mounted by the kernel NFS system, even when they are being served using Direct NFS Client. Refer to your vendor documentation to complete operating system NFS configuration and mounting. Refer to the Oracle document for more information: https://docs.oracle.com/en/database/oracle/oracle-database/21/ladbi/deploying_dnfs.html#GUID-D06079DB-8C71-4F68-A1E3-A75D7D96DCE2

Direct NFS Client searches for mount entries in the following order.

1. $ORACLE_HOME/dbs/oranfstab

2. /etc/oranfstab

3. /etc/mtab

Note: If a volume is not listed in oranfstab, Oracle will look through the OS mount tab to find a match. If that fails, control is handed back to the database and file access is made through Kernel NFS.

The syntax for the "oransftab" is as follows:

```
server: MyDataServer1

local: 192.0.2.0

path: 192.0.2.1

local: 192.0.100.0

path: 192.0.100.1

export: /vol/oradata1 mount: /mnt/oradata1
```

Note: Oracle dNFS was enabled at the RDBMS level on all the database nodes, and the “oranfstab” was updated to reflect the same across all nodes. The following is sample “oranfstab” configuration from Oracle RAC Node 1:

```
[oracle@flex1 ~]$ cat /u01/app/oracle/product/21.3.0/dbhome_1/dbs/oranfstab

Server: NetApp-A800

path: 10.10.21.41

path: 10.10.22.41

path: 10.10.23.41

path: 10.10.24.41

path: 10.10.21.42

path: 10.10.22.42

path: 10.10.23.42

path: 10.10.24.42

nfs_version: nfsv3

export: /soedata01 mount: /soedata01

export: /soedata02 mount: /soedata02

export: /soedata03 mount: /soedata03

export: /soedata04 mount: /soedata04

export: /soedata05 mount: /soedata05

export: /soedata06 mount: /soedata06

export: /soedata07 mount: /soedata07

export: /soedata08 mount: /soedata08

export: /soedata09 mount: /soedata09
```

```
export: /soedata10 mount: /soedata10

export: /soedata11 mount: /soedata11

export: /soedata12 mount: /soedata12

export: /soedata13 mount: /soedata13

export: /soedata14 mount: /soedata14

export: /soedata15 mount: /soedata15

export: /soedata16 mount: /soedata16

export: /soelog01 mount: /soelog01

export: /soelog02 mount: /soelog02

export: /soelog03 mount: /soelog03

export: /soelog04 mount: /soelog04

export: /findata01 mount: /findata01

export: /findata02 mount: /findata02

export: /findata03 mount: /findata03

export: /findata04 mount: /findata04

export: /findata05 mount: /findata05

export: /findata06 mount: /findata06

export: /findata07 mount: /findata07

export: /findata08 mount: /findata08

export: /findata09 mount: /findata09

export: /findata10 mount: /findata10

export: /findata11 mount: /findata11

export: /findata12 mount: /findata12

export: /findata13 mount: /findata13

export: /findata14 mount: /findata14

export: /findata15 mount: /findata15

export: /findata16 mount: /findata16

export: /finlog01 mount: /finlog01

export: /finlog02 mount: /finlog02
```

```
export: /finlog03 mount: /finlog03

export: /finlog04 mount: /finlog04

export: /shdata01 mount: /shdata01

export: /shdata02 mount: /shdata02

export: /shdata03 mount: /shdata03

export: /shdata04 mount: /shdata04

export: /shdata05 mount: /shdata05

export: /shdata06 mount: /shdata06

export: /shdata07 mount: /shdata07

export: /shdata08 mount: /shdata08

export: /shdata09 mount: /shdata09

export: /shdata10 mount: /shdata10

export: /shdata11 mount: /shdata11

export: /shdata12 mount: /shdata12

export: /shdata13 mount: /shdata13

export: /shdata14 mount: /shdata14

export: /shdata15 mount: /shdata15

export: /shdata16 mount: /shdata16

export: /shlog01 mount: /shlog01

export: /shlog02 mount: /shlog02

export: /shlog03 mount: /shlog03

export: /shlog04 mount: /shlog04
```

When the “oranfstab” file is created on all the RAC nodes, you need to enable the direct NFS client ODM library on all nodes. Shutdown the databases before this step. Run the following commands:

```
cd $ORACLE_HOME/rdbms/lib

make -f ins_rdbms.mk dnfs_on
```

This completes the dNFS setup.

Note: Oracle dNFS is by default enabled on Oracle 12c onwards. To disable dNFS, the RDBMS should be rebuilt with the dnfs_off option. Check the Best Practices section for enabling/disabling Oracle dNFS:

https://docs.oracle.com/en/database/oracle/oracle-database/19/ladbi/enabling-and-disabling-direct-nfs-client-control-of-nfs.html#GUID-27DDB55B-F79E-4F40-8228-5D94456E620B

Verify that the Oracle dNFS is enabled at the database level and working as expected. Run a SQL query against v$dnfs_servers that should show the details of the dNFS mounts as shown below.

```
SQL>
SQL> select svrname, dirname, mntport, nfsport, wtmax, rtmax, nfsversion from v$dnfs_servers;

SVRNAME                 DIRNAME                   MNTPORT    NFSPORT      WTMAX      RTMAX NFSVERSION
----------------------- ------------------------ ---------- ---------- ---------- ---------- ----------------
192.168.21.201          /oradata01                     2049       2049     524288     524288 NFSv3.0
192.168.21.201          /oraredo                       2049       2049     524288     524288 NFSv3.0
192.168.22.201          /oradata02                     2049       2049     524288     524288 NFSv3.0
192.168.23.201          /oradata03                     2049       2049     524288     524288 NFSv3.0
192.168.24.201          /oradata04                     2049       2049     524288     524288 NFSv3.0
```

SLOB Test

The Silly Little Oracle Benchmark (SLOB) is a toolkit for generating and testing I/O through an Oracle database. SLOB is very effective in testing the I/O subsystem with genuine Oracle SGA-buffered physical I/O. SLOB supports testing physical random single-block reads (db file sequential read) and random single block writes (DBWR flushing capability). SLOB issues single block reads for the read workload that are generally 8K (as the database block size was 8K).

For testing the SLOB workload, we have created one container database as FINCDB. For SLOB database, we have created total 20 file system (16 file system for data files and 4 file system for log files) and mounted on all eight nodes.

These file system volumes provided the storage required to create the tablespaces for the SLOB Database. We loaded SLOB schema on data volumes of up to 3 TB in size. We used SLOB2 to generate our OLTP workload. Each database server applied the workload to Oracle database, log, and temp files. The following tests were performed and various metrics like IOPS and latency were captured along with Oracle AWR reports for each test scenario.

User Scalability Test

SLOB2 was configured to run against all the eight Oracle RAC nodes and the concurrent users were equally spread across all the nodes. We tested the environment by increasing the number of Oracle users in database from a minimum of 128 users up to a maximum of 512 users across all the nodes. At each load point, we verified that the storage system and the server nodes could maintain steady-state behavior without any issues. We also made sure that there were no bottlenecks across servers or networking systems.

The User Scalability test was performed with 128, 256, 384 and 512 users on 8 Oracle RAC nodes by varying read/write ratio as follows:

- 100% read (0% update)
- 90% read (10% update)
- 70% read (30% update)
- 50% read (50% update)

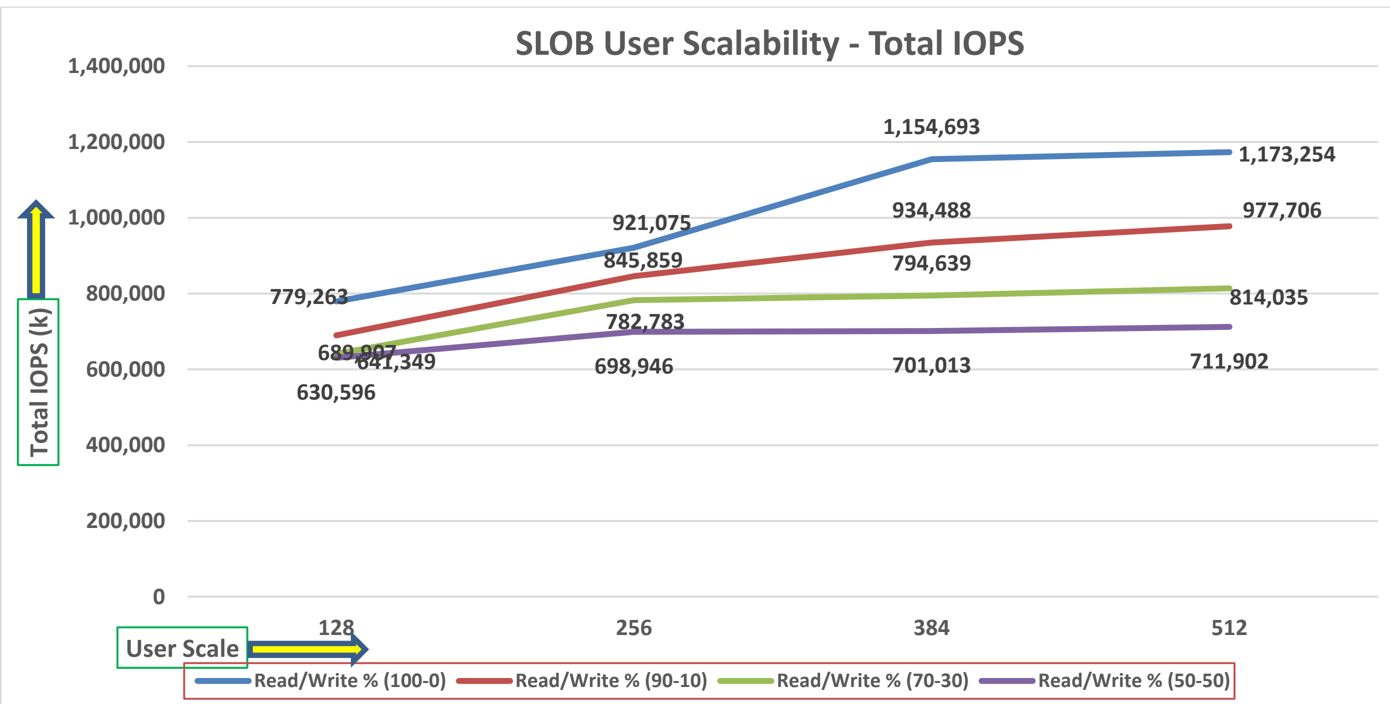
Table 12 lists the total number of IOPS (both read and write) available for user scalability test when run with 128, 256, 384 and 512 Users on the SLOB database.

Table 12. Total number of IOPS

Users	Read/Write % (100-0)	Read/Write % (90-10)	Read/Write % (70-30)	Read/Write % (50-50)
128	779,263	689,907	611,349	630,596
256	921,075	845,859	782,783	698,946
384	1,154,693	934,488	794,639	701,013
512	1,173,254	977,706	814,035	711,902

The following graphs demonstrate the total number of IOPS while running SLOB workload for various concurrent users for each test scenario.

The graph below shows the linear scalability with increased users and similar IOPS from 128 users to 512 users with 100% Read/Write, 90% Read/Write, 70% Read/Write and 50% Read/Write.



The AWR screenshot below was captured from a 100% Read (0% update) Test scenario while running SLOB test for 512 users for 4 hours. The screenshot shows a section from the Oracle AWR report from the run that highlights Physical Reads/Sec and Physical Writes/Sec for each instance. It highlights that IO load is distributed across all the cluster nodes performing workload operations. Due to variations in workload randomness, we conducted multiple runs to ensure consistency in behavior and test results.

System Statistics - Per Second DB/Inst: FINCDB/fincdb1 Snaps: 66-67

I#	Logical Reads/s	Physical Reads/s	Physical Writes/s	Redo Size (k)/s	Block Changes/s	User Calls/s	Execs/s	Parses/s	Logons/s	Txns/s
1	162,070.56	146,503.3	1.3	5.1	23.4	3.3	2,494.7	11.2	0.67	0.1
2	162,298.26	146,713.2	0.9	3.2	9.2	3.5	2,483.6	4.0	0.67	0.1
3	160,558.75	145,108.3	1.0	3.2	9.6	3.3	2,456.0	4.8	0.68	0.1
4	176,506.16	146,717.0	1.0	3.1	9.0	3.3	2,482.2	4.0	0.67	0.1
5	163,175.82	147,431.5	1.0	3.2	9.2	3.3	2,499.3	4.0	0.67	0.1
6	161,570.84	145,985.8	1.0	3.1	8.8	3.3	2,482.7	3.7	0.67	0.1
7	163,279.42	147,529.7	0.9	3.0	8.2	3.3	2,513.9	3.5	0.68	0.1
8	165,982.38	147,254.2	1.1	2.9	10.7	3.3	2,495.2	5.1	0.68	0.1
Sum	1,315,442.18	1,173,242.9	8.1	26.8	88.1	26.6	19,907.7	40.5	5.39	0.6
Avg	164,430.27	146,655.4	1.0	3.3	11.0	3.3	2,488.5	5.1	0.67	0.1
Std	5,132.71	810.4	0.1	0.7	5.0	0.1	16.9	2.5	0.00	0.0

The screenshot below was captured from a 100% Read Test scenario while running SLOB test with 512 users for sustained 12 Hours.

WORKLOAD REPOSITORY REPORT (RAC)

Database Summary

Id	Name	Unique Name	Role	Edition	RAC	CDB	Block Size	Snapshot Ids Begin	Snapshot Ids End	Number of Instances In Report	Number of Instances Total	Number of Hosts In Report	Number of Hosts Total	Report Total (minutes) DB time	Report Total (minutes) Elapsed time
3096272561	FINCDB	fincdb	PRIMARY	EE	YES	YES	8192	101	114	8	8	8	8	368,749.52	720.49

Database Instances Included In Report
-> Listed in order of instance number, I#

I#	Instance	Host	Startup	Begin Snap Time	End Snap Time	Release	Elapsed Time(min)	DB time(min)	Up Time(hrs)	Avg Active Sessions	Platform
1	fincdb1	flex1	25-Jan-23 15:24	27-Jan-23 00:23	27-Jan-23 12:23	21.0.0.0.0	720.10	46,100.56	44.97	64.02	Linux x86 64-bi
2	fincdb2	flex2	25-Jan-23 15:24	27-Jan-23 00:22	27-Jan-23 12:23	21.0.0.0.0	720.10	46,093.49	44.97	64.01	Linux x86 64-bi
3	fincdb3	flex3	25-Jan-23 15:24	27-Jan-23 00:23	27-Jan-23 12:23	21.0.0.0.0	720.10	46,092.25	44.97	64.01	Linux x86 64-bi
4	fincdb4	flex4	25-Jan-23 17:42	27-Jan-23 00:23	27-Jan-23 12:23	21.0.0.0.0	720.10	46,092.71	42.69	64.01	Linux x86 64-bi
5	fincdb5	flex5	25-Jan-23 15:24	27-Jan-23 00:23	27-Jan-23 12:23	21.0.0.0.0	720.10	46,091.92	44.97	64.01	Linux x86 64-bi
6	fincdb6	flex6	25-Jan-23 15:24	27-Jan-23 00:22	27-Jan-23 12:23	21.0.0.0.0	720.10	46,091.14	44.97	64.01	Linux x86 64-bi
7	fincdb7	flex7	25-Jan-23 15:24	27-Jan-23 00:22	27-Jan-23 12:23	21.0.0.0.0	720.12	46,093.37	44.97	64.01	Linux x86 64-bi
8	fincdb8	flex8	25-Jan-23 15:24	27-Jan-23 00:23	27-Jan-23 12:23	21.0.0.0.0	720.13	46,094.08	44.98	64.01	Linux x86 64-bi

Open Pluggable Databases at Begin Snap: 3, End Snap: 3

The screenshot below shows a section from AWR report from the run that highlights Physical Reads/Sec and Physical Writes/Sec for each instance for 12 Hour sustained periods.

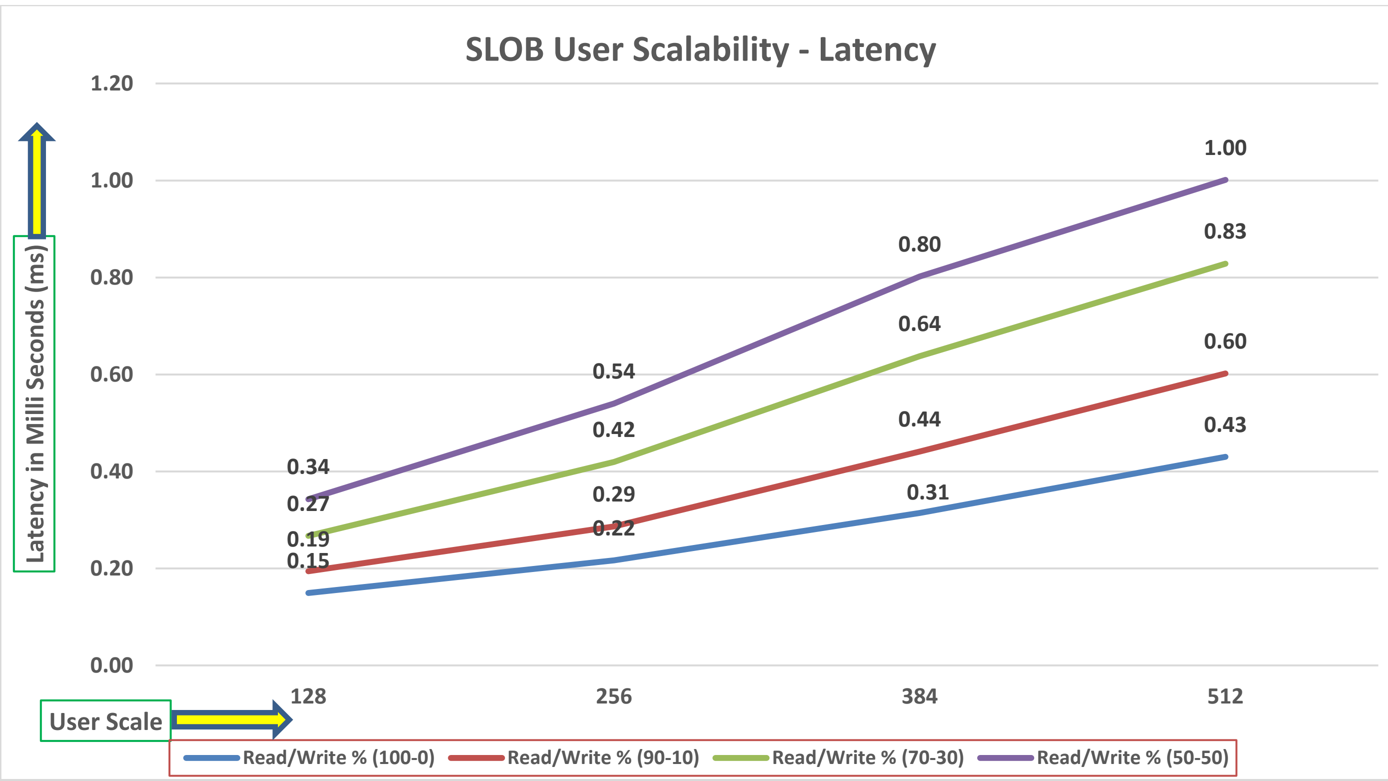
System Statistics - Per Second DB/Inst: FINCDB/fincdb1 Snaps: 101-114

I#	Logical Reads/s	Physical Reads/s	Physical Writes/s	Redo Size (k)/s	Block Changes/s	User Calls/s	Execs/s	Parses/s	Logons/s	Txns/s
1	151,302.57	138,416.0	1.8	6.2	35.5	3.6	2,353.8	13.0	0.70	0.1
2	151,379.27	138,633.0	1.4	3.7	9.7	3.7	2,388.3	3.1	0.70	0.1
3	151,388.46	138,750.0	1.3	3.6	9.2	3.5	2,335.3	3.0	0.70	0.1
4	171,157.68	138,930.6	1.4	3.7	73.3	3.5	2,342.6	3.4	0.70	0.1
5	149,896.25	137,324.9	1.3	3.6	9.3	3.5	2,354.3	2.8	0.70	0.1
6	151,182.60	138,214.4	1.5	3.7	9.5	3.5	2,535.8	3.1	0.70	0.1
7	150,782.57	137,869.3	1.6	3.7	9.5	3.5	2,501.7	3.4	0.70	0.1
8	166,323.78	139,719.0	1.4	3.1	13.3	3.5	2,416.7	3.4	0.70	0.1
Sum	1,243,413.18	1,107,857.2	11.6	31.3	169.4	28.1	19,228.4	35.3	5.62	0.7
Avg	155,426.65	138,482.2	1.5	3.9	21.2	3.5	2,403.6	4.4	0.70	0.1
Std	8,333.01	718.8	0.2	0.9	22.9	0.1	76.4	3.5	0.00	0.0

The screenshot below shows Top Timed Events and Wait Time during this 12 Hour sustained test while running with 512 Users.

```
Top Timed Events                     DB/Inst: FINCDB/fincdb1  Snaps: 101-114
-> Instance '*'  - cluster wide summary
->          '*'  Waits, %Timeouts, Wait Time Total(s)      : Cluster-wide total for the wait event
->          '*'  'Wait Time Avg'                         : Cluster-wide average computed as (Wait Time Total / Event Waits)
->          '*'  Summary 'Avg Wait Time '                 : Per-instance 'Wait Time Avg ' used to compute the following statistics
->          '*'  [Avg/Min/Max/Std Dev]                     : average/minimum/maximum/standard deviation of per-instance 'Wait Time Avg'
->          '*'  Cnt                                       : count of instances with wait times for the event

                               Wait                                   Event                      Wait Time                          Summary Avg Wait Time
       -------------------------------------------------   -----------------------  -------------------------------   ----------------------------------------------------
 I#    Class      Event                                          Waits %Timeouts        Total(s) Avg Wait %DB time        Avg        Min        Max    Std Dev  Cnt
----   ---------- ------------------------------------------   ----------- ---------  -------------- -------- --------   ---------- ---------- ---------- ---------- ----
  *    User I/O   db file sequential read                      4.739877E+10       0.0  2.1268329E+07 448.71us    96.13     448.72us   444.94us   452.38us     2.21us    8
                  DB CPU                                               N/A       N/A   1,366,984.20                6.18                                                  8
       Applicatio enq: TX - row lock contention                         56       0.0        2,335.91  41.71 s     0.01      41.07 s    34.30 s    45.89 s  6039.17ms    8
       System I/O control file sequential read                   1,701,849       0.0          753.53 442.77us     0.00     443.04us   432.24us   456.70us     9.83us    8
       Other      PX Deq: reap credit                           88,158,154     100.0          333.61   3.78us     0.00       3.78us     3.62us     3.93us    86.63ns    8
       Other      IMR slave acknowledgement msg                  1,111,494       0.0          223.31 200.91us     0.00     200.42us   177.89us   212.83us    10.84us    8
       Cluster    gc cr block 2-way                              2,422,399       0.0          191.42  79.02us     0.00      79.04us    77.70us    80.09us   792.32ns    8
       Concurrenc resmgr:internal state change                      10,453      17.5          190.60  18.23ms     0.00      18.26ms    16.81ms    20.45ms     1.23ms    8
       Applicatio enq: TM - contention                               1,241       0.0          171.41 138.12ms     0.00     137.97ms   108.18ms   171.22ms    22.94ms    8
       Other      enq: PS - contention                             528,101      22.5          119.36 226.02us     0.00     226.50us   198.10us   251.59us    19.07us    8
```

The graph below illustrates the latency exhibited by the NetApp AFF A800 Storage across different workloads. All the workloads experienced less than 1 millisecond latency and it varies based on the workloads. As expected, the 50% read (50% update) test exhibited higher latencies as the user counts increases.



SwingBench Test

SwingBench is a simple to use, free, Java-based tool to generate various types of database workloads and perform stress testing using different benchmarks in Oracle database environments. SwingBench can be used to demonstrate and test technologies such as Real Application Clusters, Online table rebuilds, Standby databases, online backup, and recovery, and so on. In this solution, we used SwingBench tool for running various type of workload and check the overall performance of this reference architecture.

SwingBench provides four separate benchmarks, namely, Order Entry, Sales History, Calling Circle, and Stress Test. For the tests described in this solution, SwingBench Order Entry (SOE) benchmark was used for representing OLTP type of workload and the Sales History (SH) benchmark was used for representing DSS type of workload.

The Order Entry benchmark is based on SOE schema and is TPC-C like by types of transactions. The workload uses a very balanced read/write ratio around 60/40 and can be designed to run continuously and test the performance of a typical Order Entry workload against a small set of tables, producing contention for database resources.

The Sales History benchmark is based on the SH schema and is like TPC-H. The workload is query (read) centric and is designed to test the performance of queries against large tables.

The first step after the databases creation is calibration; about the number of concurrent users, nodes, throughput, IOPS and latency for database optimization. For this solution, we ran the SwingBench workloads on various combination of databases and captured the system performance as follows:

Typically encountered in the real-world deployments, we tested a combination of scalability and stress related scenarios that ran across all the 8-node Oracle RAC cluster, as follows:

- OLTP database user scalability workload representing small and random transactions.
- DSS database workload representing larger transactions.
- Mixed databases (OLTP and DSS) workloads running simultaneously.

For this SwingBench workload tests, we created two Container Database as SOECDB and SHCDB. We configured the SOECDB container database and created two Pluggable Databases as SOEPDB and ENGPDB to run the SwingBench SOE workload representing OLTP type of workload characteristics. We configured the SHCDB container databases and created one Pluggable Databases as SHPDB to run the SwingBench SH workload representing DSS type of workload characteristics.

For this solution, we deployed multiple pluggable databases (SOEPDB and ENGPDB) plugged into one container (SOECDB) database and one pluggable database (SHPDB) plugged into one container (SHCDB) database to demonstrate the multitenancy capability, performance, and sustainability for this reference architecture.

In SOECDB container database, we created two pluggable databases as both the databases have similar workload characteristics. By consolidating multiple pluggable databases under the same container database allows easier management, efficiently sharing computational and memory resources, separation of administrative tasks, easier database upgrades as well as fewer patches and upgrades.

For the OLTP databases, we created and configured SOE schema of 3.5 TB for the SOEPDB Database and 2.5 TB for the ENGPDB Database. For the DSS database, we created and configured SH schema of 4 TB for the SHPDB Database:

- One OLTP Database Performance
- Multiple (Two) OLTP Databases Performance
- One DSS Database Performance
- Multiple OLTP & DSS Databases Performance

One OLTP Database Performance

For one OLTP database workload featuring Order Entry schema, we created one container database SOECDB and one pluggable database SOEPDB as explained earlier. We used 64 GB size of SGA for this database and, we ensured that the HugePages were in use. We ran the SwingBench SOE workload with varying the total number of users on this database from 256 Users to 896 Users. Each user scale iteration test was run for at least 3 hours and for each test scenario, we captured the Oracle AWR reports to check the overall system performance below:

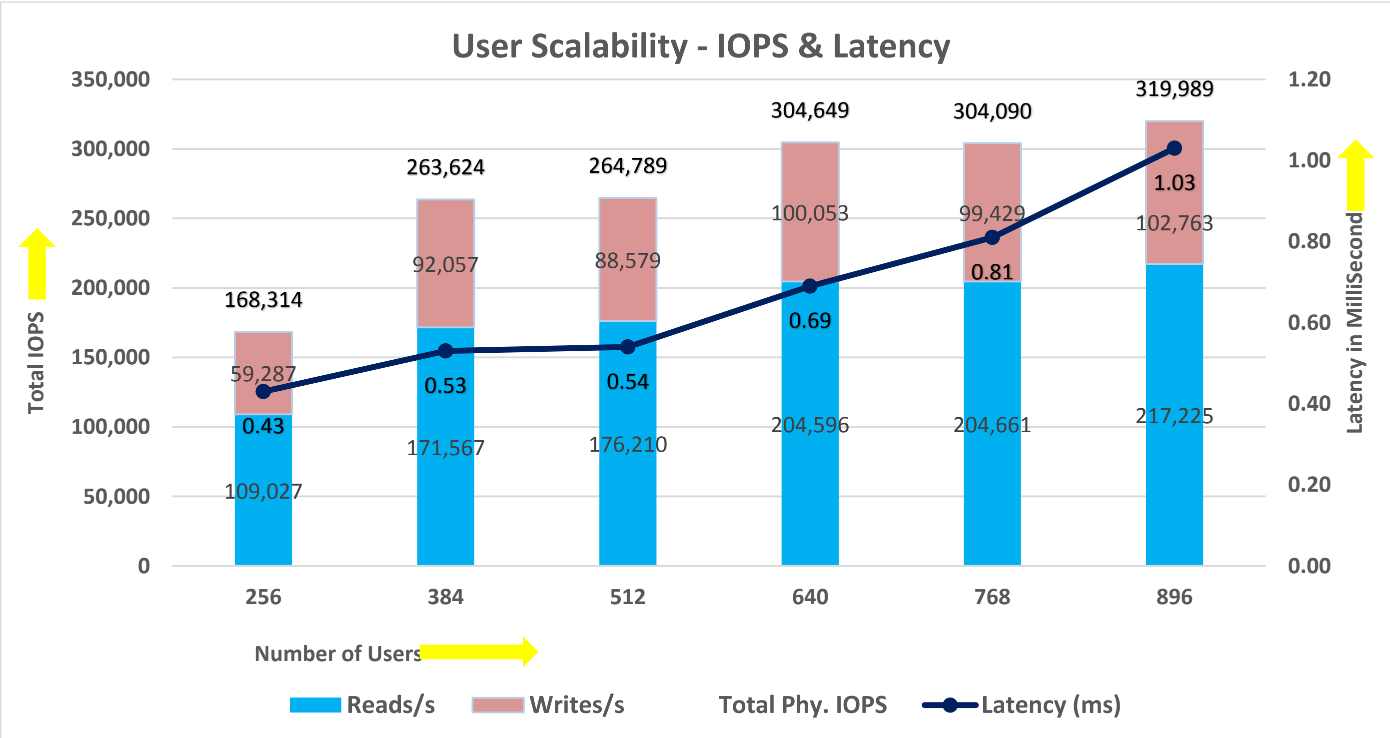
User Scalability

Table 13 lists the Transaction Per Minutes (TPM), IOPS, Latency and System Utilization for the SOECDB Database while running the workload from 256 users to 896 users across all the eight RAC nodes.

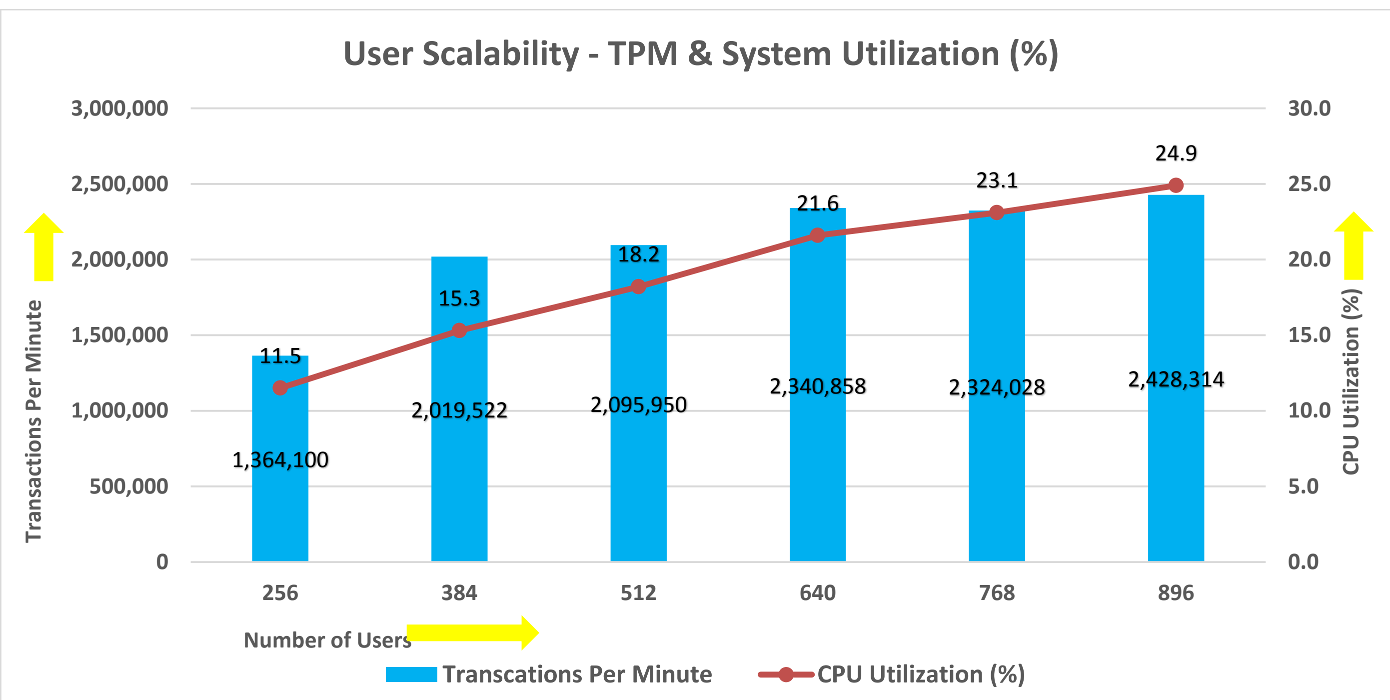
Table 13. User Scale Test on One OLTP Database

Number of Users	Transactions		Storage IOPS			Latency (milliseconds)	CPU Utilization (%)
	Per Seconds (TPS)	Per Minutes (TPM)	Reads/Sec	Writes/Sec	Total IOPS		
256	22,735	1,364,100	109,027	59,287	168,314	0.43	11.5
384	33,659	2,019,522	171,567	92,057	263,624	0.53	15.3
512	34,933	2,095,950	176,210	88,579	264,789	0.54	18.2
640	39,014	2,340,858	204,596	100,053	304,649	0.69	21.6
768	38,734	2,324,028	204,661	99,429	304,090	0.81	23.1
896	40,472	2,428,314	217,225	102,763	319,989	1.03	24.9

The following chart shows the IOPS and Latency for the SOECDB Database while running the workload from 256 users to 896 users across all eight RAC nodes.



The chart below shows the TPM and System Utilization for the same above tests on SOECDB Database for running the workload from 256 users to 896 users:



The screenshot below captured from the Oracle AWR report highlights the Physical Reads/Sec, Physical Writes/Sec and Transactions per Seconds for the Container SOECDB Database. We captured about 320k IOPS (217k Reads/s and 102k Writes/s) with the 40k TPS while running this workload on one database.

System Statistics - Per Second DB/Inst: SOECDB/soecdb1 Snaps: 90-92

I#	Logical Reads/s	Physical Reads/s	Physical Writes/s	Redo Size (k)/s	Block Changes/s	User Calls/s	Execs/s	Parses/s	Logons/s	Txns/s
1	599,839.62	25,534.2	12,354.3	17,692.4	106,084.5	14,724.4	51,003.4	5,855.8	0.71	4,906.6
2	600,101.22	25,667.6	12,733.5	17,922.3	107,221.1	14,800.6	51,267.7	5,884.0	0.71	4,932.1
3	661,781.52	25,688.4	12,416.4	17,791.5	106,350.2	14,776.5	51,171.0	5,874.9	0.71	4,924.1
4	699,612.72	27,055.3	13,056.4	18,824.4	112,422.7	15,621.5	54,084.7	6,209.7	0.71	5,205.8
5	828,194.07	30,537.0	12,876.0	18,849.6	113,181.5	15,739.6	54,509.1	6,254.8	0.71	5,245.1
6	609,042.88	25,868.3	12,849.9	17,995.7	107,922.3	14,910.7	51,642.5	5,928.4	0.71	4,968.8
7	619,770.35	25,649.0	12,602.0	17,926.3	107,349.0	14,910.9	51,638.4	5,927.5	0.71	4,968.9
8	590,630.47	31,225.5	13,874.7	18,944.5	114,725.8	15,966.2	55,302.8	6,347.5	0.71	5,320.6
Sum	5,208,972.84	217,225.4	102,763.3	145,946.8	875,257.3	121,450.3	420,619.6	48,282.6	5.68	40,471.9
Avg	651,121.61	27,153.2	12,845.4	18,243.3	109,407.2	15,181.3	52,577.4	6,035.3	0.71	5,059.0
Std	80,591.83	2,358.3	478.4	530.4	3,448.2	505.0	1,746.7	200.0	0.00	168.4

The screenshot below captured from the Oracle AWR report shows the SOECDB database "IO Profile" for the "Reads/s" and "Writes/s" requests for the entire duration of the test. The Total Requests (Read and Write Per Second) were around "345k" with Total (MB) Read+Write Per Second was around "2729" MB/s for the SOECDB database while running the workload test on one database.

IO Profile (Global) DB/Inst: SOECDB/soecdb1 Snaps: 90-92

Statistic	Read+Write/s	Reads/s	Writes/s
Total Requests	345,404.99	216,136.70	129,268.29
Database Requests	296,920.69	215,983.03	80,937.66
Optimized Requests	0.00	0.00	0.00
Redo Requests	23,555.01	N/A	23,555.01
Total (MB)	2,729.41	1,772.78	956.63
Database (MB)	2,486.13	1,687.71	798.41
Optimized Total (MB)	0.00	0.00	0.00
Redo (MB)	141.74	N/A	141.74
Database (blocks)	318,224.11	216,027.50	102,196.61
Via Buffer Cache (blocks)	318,205.13	216,017.84	102,187.28
Direct (blocks)	19.00	9.67	9.33

The screenshot below captured from the Oracle AWR report shows the "Top Timed Events" and average wait time for the SOECDB database for the entire duration of the test running with 896 Users.

```
Top Timed Events                        DB/Inst: SOECDB/soecdb1  Snaps: 90-92
-> Instance '*'  - cluster wide summary
->         '*'  Waits, %Timeouts, Wait Time Total(s)      : Cluster-wide total for the wait event
->         '*'  'Wait Time Avg'                          : Cluster-wide average computed as (Wait Time Total / Event Waits)
->         '*'  Summary 'Avg Wait Time '                  : Per-instance 'Wait Time Avg ' used to compute the following statistics
->         '*'  [Avg/Min/Max/Std Dev]                         : average/minimum/maximum/standard deviation of per-instance 'Wait Time Avg'
->         '*'  Cnt                                           : count of instances with wait times for the event

                         Wait                                  Event                       Wait Time                             Summary Avg Wait Time
       -------------------------------------------------  ----------------------  ---------------------------------  ---------------------------------------------
 I#    Class      Event                                          Waits %Timeouts        Total(s) Avg Wait %DB time        Avg        Min        Max    Std Dev  Cnt
----   ---------  ---------------------------------------  ------------ ---------  ------------- --------- -------  ---------- ---------- ---------- ---------- ----
  *               DB CPU                                             N/A       N/A      379,838.07               12.83                                                  8
       User I/O   db file sequential read                    753,521,837       0.0      344,401.40 457.06us      11.63    457.65us   440.75us   464.58us     9.02us    8
       Cluster    gc cr block congested                       17,382,340       0.0      302,962.62  17.43ms      10.23     17.16ms    14.37ms    18.57ms     1.64ms    8
       Cluster    gc current block congested                  23,929,745       0.0      290,106.86  12.12ms       9.80     11.95ms     9.95ms    12.84ms   940.34us    8
       Cluster    gc current grant busy                       31,340,038       0.0      212,954.99   6.79ms       7.19      6.79ms     5.31ms     7.30ms   721.60us    8
       Cluster    gc buffer busy acquire                       4,853,096       0.0      207,299.22  42.71ms       7.00     42.76ms    35.40ms    45.11ms     3.21ms    8
       Cluster    gc current block 3-way                     352,251,937       0.0      183,732.83 521.59us       6.20    521.14us   427.25us   558.05us    46.87us    8
       Cluster    gc current grant 2-way                     339,551,269       0.0      151,094.28 444.98us       5.10    435.80us   284.42us     1.03ms   260.84us    8
       Commit     log file sync                              144,861,010       0.0      141,117.51    .97ms       4.77       .97ms   915.69us     1.01ms    33.31us    8
       Cluster    gc cr block 3-way                          180,986,589       0.0      134,350.55 742.32us       4.54    747.42us   504.55us   850.95us   147.43us    8
```

The screenshot below shows the NetApp Storage array Q S P S (qos statistics performance show) when one OLTP database was running the workload. The screenshot shows the average IOPS “350k” with the average throughput of “2750 MB/s” with the average storage latency around “0.3 millisecond”.

```
Policy Group           IOPS      Throughput    Latency Is Adaptive? Is Shared?
-------------------- -------- --------------- ---------- ------------ ----------
-total-                353832   2737.13MB/s   317.00us - -
User-Best-Effort       352756   2737.06MB/s   318.00us false true
_System-Work             1076     73.62KB/s   166.00us false true
-total-                369716   2800.50MB/s   342.00us - -
User-Best-Effort       368413   2800.45MB/s   343.00us false true
_System-Work             1303     48.91KB/s   234.00us false true
-total-                371294   2816.74MB/s   320.00us - -
User-Best-Effort       360903   2797.96MB/s   325.00us false true
_System-Work            10391     18.78MB/s   129.00us false true
-total-                338746   2623.02MB/s   308.00us - -
User-Best-Effort       338157   2622.92MB/s   308.00us false true
_System-Work              589    103.49KB/s   249.00us false true
-total-                362948   2783.78MB/s   341.00us - -
User-Best-Effort       362866   2783.78MB/s   341.00us false true
_System-Work               82      8.68KB/s   195.00us false true
-total-                354500   2760.99MB/s   448.00us - -
User-Best-Effort       354467   2760.99MB/s   448.00us false true
_System-Work               33         0KB/s    60.00us false true
```

The storage cluster utilization during the above test was average around 62% which was an indication that storage hasn’t reached the threshold and could take more load by adding multiple databases.

```
FlexPod-A800: cluster.cluster: 2/2/2023 08:13:18
 cpu  cpu    total                      fcache    total    total data     data     data cluster  cluster  cluster     disk     disk     pkts     pkts
 avg busy      ops  nfs-ops cifs-ops       ops     recv     sent busy     recv     sent    busy     recv     sent     read    write     recv     sent
---- ---- -------- -------- -------- -------- -------- -------- ---- -------- -------- ------- -------- -------- -------- -------- -------- --------
 62%  74%   350191   350191        0        0   1.05GB   1.87GB   4%   1.05GB   1.87GB      0%   47.9KB   48.0KB   1.93GB    894MB   447426   390688
 62%  74%   351368   351368        0        0   1.02GB   1.85GB   4%   1.02GB   1.85GB      0%   45.3KB   45.3KB   1.98GB   1.03GB   440889   384713
 62%  72%   352978   352978        0        0   1013MB   1.85GB   4%   1013MB   1.85GB      0%   48.3KB   48.3KB   1.97GB    911MB   435548   383079
 62%  71%   340182   340182        0        0    973MB   1.83GB   4%    973MB   1.83GB      0%   41.1KB   41.2KB   1.96GB    991MB   429525   375344
 61%  71%   345008   345008        0        0   1.00GB   1.82GB   4%   1.00GB   1.82GB      0%   47.0KB   47.1KB   1.95GB   1004MB   436520   380513
 63%  72%   345026   345026        0        0   1.01GB   1.78GB   4%   1.01GB   1.78GB      0%   91.0KB   90.9KB   1.97GB   1.04GB   431126   374434
 65%  75%   364188   364188        0        0   1.12GB   1.93GB   4%   1.12GB   1.93GB      0%   38.3KB   38.3KB   2.15GB   1.12GB   463400   404244
 64%  77%   363463   363463        0        0   1.08GB   1.88GB   4%   1.08GB   1.88GB      0%   50.7KB   50.7KB   2.12GB   1.07GB   453342   395815
```

We also ran the maximum number of users (896) test for 24-hour period to check the system performance. For the entire 24-hour test, we observed the system performance (IOPS and Throughput) was consistent throughout and we did not observe any dips in performance while running one OLTP database stress test.

Multiple (Two) OLTP Databases Performance

For running multiple OLTP database workload, we have created one container database SOECDB and two pluggable database SOEPDB and ENGPDB as explained earlier. We ran the SwingBench SOE workload on both the databases at the same time with varying the total number of users on both the databases from 384 Users to 896 Users. Each user scale iteration test was run for at least 3 hours and for each test scenario, we captured the Oracle AWR reports to check the overall system performance below.

Table 14 lists the IOPS and System Utilization for each of the pluggable databases while running the workload from total of 384 users to 896 users across all the eight RAC nodes.

Table 14. IOPS and System Utilization for Pluggable Databases

Users	IOPS for SOE	IOSP for OLTP	Total IOPS	System Utilization (%)
256	148,159	144,873	293,032	17.8
384	164,374	159,703	324,077	20.1
512	172,622	165,106	337,728	21.9
640	181,610	180,649	362,259	23.4
768	199,981	193,273	393,254	25.8

The chart below shows the IOPS and System Utilization for the overall CDBDB Database while running the database workload on both the databases at the same time. We observed both databases were linearly scaling the IOPS after increasing and scaling more users. We observed average 393k IOPS with overall system utilization around 27% when scaling maximum number of users on multiple database workload test. After increasing users beyond certain level, we observed more GC cluster events and overall similar IOPS around 395k.

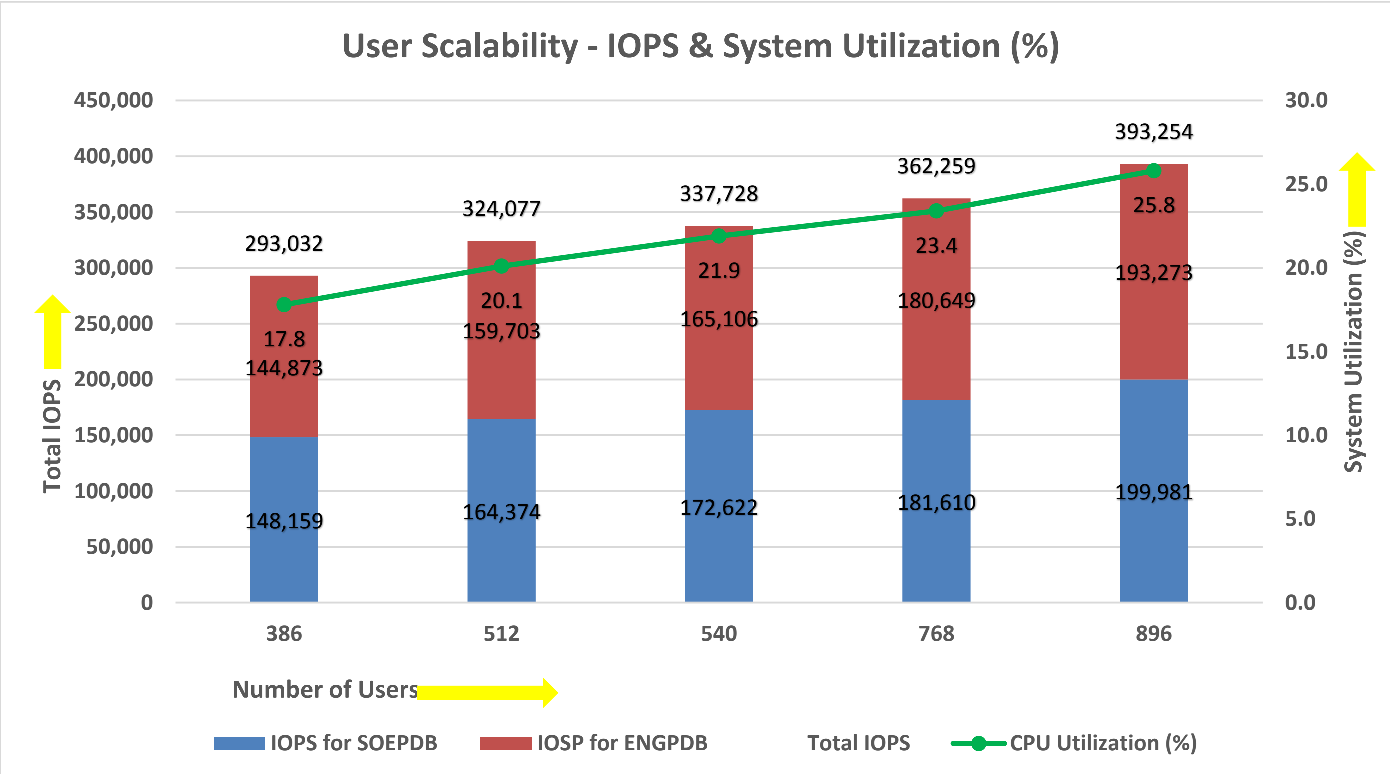
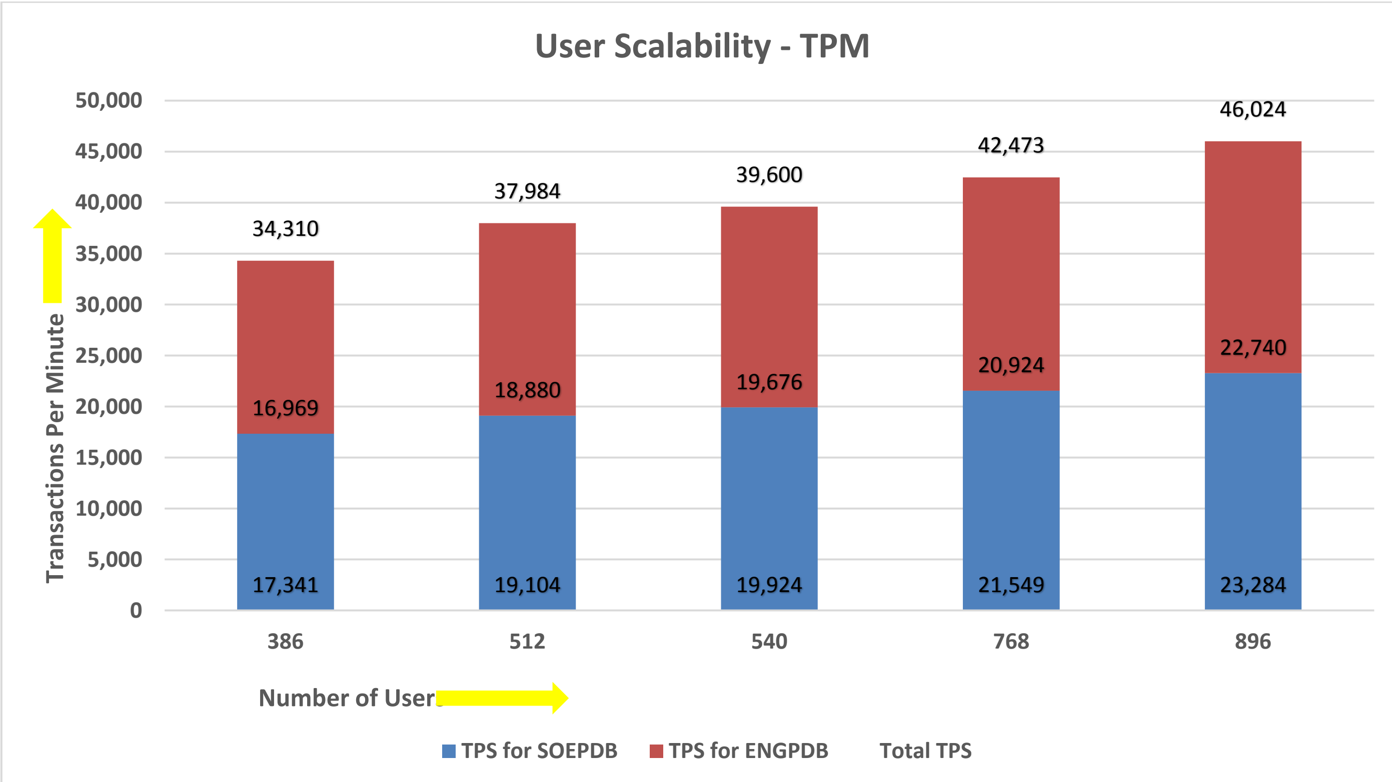


Table 15 lists the Transactions per Seconds (TPS) and Transactions per Minutes (TPM) for each of the pluggable databases while running the workload from total of 386 users to 896 users across all the eight RAC nodes.

Table 15. Transactions per Seconds and Transactions per Minutes

Users	TPS for SOE	TPS for OLTP	Total TPS	Total TPM
386	17,341	16,969	34,310	2,058,612
512	19,104	18,880	37,984	2,279,034
540	19,924	19,676	39,600	2,375,970
768	21,549	20,924	42,473	2,548,368
896	23,284	22,740	46,024	2,761,416

The chart below shows the Transactions per Seconds (TPS) for the same tests (above) on CDBDB Database for running the workload on both pluggable databases.



The screenshot below was captured from the Oracle AWR report, highlights the Physical Reads/Sec, Physical Writes/Sec and Transactions per Seconds for the Container Database while running total of 896 users on both pluggable databases. We captured about 393k IOPS (264k Reads/s and 129k Writes/s) with the 46k TPS (2,761,416 TPM) while running multiple OLTP databases workloads.

System Statistics - Per Second DB/Inst: SOECDB/soecdb1 Snaps: 1157-1159

I#	Logical Reads/s	Physical Reads/s	Physical Writes/s	Redo Size (k)/s	Block Changes/s	User Calls/s	Execs/s	Parses/s	Logons/s	Txns/s
1	670,087.02	32,702.3	15,805.5	19,920.0	122,125.4	16,955.1	58,899.8	6,746.8	0.87	5,650.2
2	651,324.65	33,530.5	16,401.0	20,516.7	125,841.6	17,544.3	60,936.1	6,974.3	0.86	5,846.5
3	722,595.63	33,446.6	16,287.2	20,612.3	125,842.9	17,498.7	60,777.3	6,956.7	0.85	5,831.3
4	732,436.41	33,840.0	16,506.7	20,787.8	127,365.1	17,757.5	61,676.0	7,063.4	0.86	5,917.5
5	793,976.53	33,648.2	16,378.6	20,720.3	126,712.7	17,643.2	61,282.2	7,012.9	0.86	5,879.4
6	684,610.88	33,511.2	16,715.1	20,566.5	126,834.4	17,588.1	61,082.2	6,992.0	0.86	5,861.1
7	607,877.01	29,778.0	14,771.2	18,316.6	112,535.5	15,589.6	54,154.7	6,201.7	0.86	5,194.9
8	651,585.43	33,574.8	16,357.1	20,943.6	126,296.8	17,533.5	60,897.3	6,971.0	0.87	5,842.8
Sum	5,514,493.57	264,031.6	129,222.6	162,383.8	993,554.4	138,110.0	479,705.6	54,918.9	6.87	46,023.6
Avg	689,311.70	33,004.0	16,152.8	20,298.0	124,194.3	17,263.7	59,963.2	6,864.9	0.86	5,753.0
Std	58,324.06	1,345.3	614.5	855.5	4,977.5	716.9	2,486.4	283.6	0.00	239.0

We also ran the 768 number of users test for 12-hour period to check the system overall performance. The screenshot below highlights the database summary while running the SwingBench SOE workload for 12-hour test duration on container database and the container database “SOECDB” was running with two pluggable databases as “SOEPDB” and “ENGPDB.”

```
WORKLOAD REPOSITORY REPORT (RAC)

Database Summary
~~~~~~~~~~~~~~~~
                              Database                                  Snapshot Ids     Number of Instances    Number of Hosts    Report Total (minutes)
---------------------------------------------------------------------- ---------------- -------------------- ------------------ -----------------------
        Id Name      Unique Name Role             Edition RAC CDB Block Size   Begin      End  In Report    Total  In Report    Total     DB time Elapsed time
----------- -------- ----------- ---------------- ------- --- --- ----------   -------- -------- --------- -------- --------- -------- ----------- ------------
 1630447391 SOECDB   soecdb      PRIMARY          EE      YES YES       8192     1116     1128         8        8         8        8  498,227.22       721.41
Database Instances Included In Report
-> Listed in order of instance number, I#
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
                                                                                                                             Avg Active
  I# Instance  Host      Startup         Begin Snap Time End Snap Time   Release         Elapsed Time(min) DB time(min) Up Time(hrs)  Sessions Platform
---- --------- --------- --------------- --------------- --------------- --------------- ----------------- ------------ ------------ ---------- ---------------
   1 soecdb1   flex1     09-Mar-23 15:10 15-Mar-23 18:52 16-Mar-23 06:52 21.0.0.0.0                 720.07    62,445.15       159.70      86.72 Linux x86 64-bi
   2 soecdb2   flex2     09-Mar-23 15:09 15-Mar-23 18:51 16-Mar-23 06:51 21.0.0.0.0                 720.07    62,157.71       159.70      86.32 Linux x86 64-bi
   3 soecdb3   flex3     09-Mar-23 15:51 15-Mar-23 18:52 16-Mar-23 06:52 21.0.0.0.0                 720.05    62,251.40       159.03      86.45 Linux x86 64-bi
   4 soecdb4   flex4     09-Mar-23 15:10 15-Mar-23 18:52 16-Mar-23 06:52 21.0.0.0.0                 720.07    62,139.93       159.70      86.30 Linux x86 64-bi
   5 soecdb5   flex5     09-Mar-23 15:10 15-Mar-23 18:52 16-Mar-23 06:52 21.0.0.0.0                 720.05    62,160.38       159.70      86.33 Linux x86 64-bi
   6 soecdb6   flex6     09-Mar-23 15:09 15-Mar-23 18:51 16-Mar-23 06:51 21.0.0.0.0                 720.05    62,192.32       159.70      86.37 Linux x86 64-bi
   7 soecdb7   flex7     09-Mar-23 15:09 15-Mar-23 18:51 16-Mar-23 06:51 21.0.0.0.0                 720.05    62,692.96       159.70      87.07 Linux x86 64-bi
   8 soecdb8   flex8     09-Mar-23 15:09 15-Mar-23 18:51 16-Mar-23 06:51 21.0.0.0.0                 720.05    62,187.36       159.70      86.37 Linux x86 64-bi
Open Pluggable Databases at Begin Snap: 4, End Snap: 4
~~~~~~~~~~~~~~~~~~~~~~~~~
```

The screenshot below, "OS Statistics by Instance" while the system was running mixed workload. As shown below, the workload was equally spread across all the databases clusters while the average CPU utilization was around 25% overall.

```
OS Statistics By Instance          DB/Inst: SOECDB/soecdb1  Snaps: 1116-1128
-> Listed in order of instance number, I#
-> End values are diplayed only if different from begin values

          CPU               Load                   % CPU                                       Time (s)                        Memory          End Values
     -----------------  -------------  ----------------------------------  --------------------------------------------  ----------  ------------------------
  I# #CPUs #Core #Sckt   Begin    End  % Busy  % Usr  % Sys  % WIO  % Idl            Busy           Idle          Total          MB  #CPU #Cor #Sck Memory (M)
---- ----- ----- -----  ------ ------  ------ ------ ------ ------ ------  -------------- -------------- --------------  ----------  ---- ---- ---- ----------
   1   112    56     2    19.8   19.6    26.7   18.0    5.2     .0   73.3     1,282,455.1    3,514,930.8    4,797,385.9   514,983.8
   2   112    56     2    18.2   17.7    21.4   14.7    3.9     .0   78.6     1,026,980.2    3,777,428.1    4,804,408.3   514,983.8
   3   112    56     2    14.6   20.6    21.5   14.8    3.9     .0   78.5     1,034,651.0    3,770,795.1    4,805,446.1   514,983.8
   4   112    56     2    13.4   14.8    21.5   14.9    3.9     .0   78.5     1,030,269.1    3,772,594.0    4,802,863.1   514,983.7
   5   112    56     2    21.0   18.4    22.4   15.5    4.0     .0   77.6     1,078,416.4    3,725,434.7    4,803,851.1   514,984.0
   6   112    56     2    17.8   16.1    21.5   14.9    3.9     .0   78.5     1,034,910.6    3,768,363.4    4,803,274.0   514,983.8
   7   112    56     2    20.2   21.6    22.7   15.6    4.1     .0   77.3     1,089,798.8    3,716,870.6    4,806,669.4   514,983.8
   8   112    56     2    15.9   21.1    21.8   15.0    3.9     .0   78.2     1,048,767.4    3,755,334.7    4,804,102.1   514,984.0
 ~~~                                                                       ~~~~~~~~~~~~~~ ~~~~~~~~~~~~~~ ~~~~~~~~~~~~~~
 Sum                                                                           8,626,248.5   29,801,751.4   38,427,999.9
                    -------------------------------------------------------------------------------------------------------------------
```

The screenshot below was captured from the Oracle AWR report shows the "Top Timed Events" for the container database for the entire 12-hour duration of the test.

```
Top Timed Events                    DB/Inst: SOECDB/soecdb1  Snaps: 1116-1128
-> Instance '*'  - cluster wide summary
->          '*'  Waits, %Timeouts, Wait Time Total(s)      : Cluster-wide total for the wait event
->          '*'  'Wait Time Avg'                         : Cluster-wide average computed as (Wait Time Total / Event Waits)
->          '*'  Summary 'Avg Wait Time '                 : Per-instance 'Wait Time Avg ' used to compute the following statistics
->          '*'  [Avg/Min/Max/Std Dev]                     : average/minimum/maximum/standard deviation of per-instance 'Wait Time Avg'
->          '*'  Cnt                                       : count of instances with wait times for the event

                         Wait                                       Event                      Wait Time                              Summary Avg Wait Time
       ---------------------------------------------------  ----------------------  ---------------------------------  --------------------------------------------------
  I#   Class      Event                                          Waits %Timeouts        Total(s) Avg Wait %DB time         Avg        Min        Max    Std Dev  Cnt
----   ---------- ----------------------------------------  ------------ ---------  ---------------- --------- --------  ---------- ---------- ---------- ---------- ----
   *   User I/O   db file sequential read                   1.027440E+10       0.0  6,765,753.39 658.51us     22.63    658.57us   654.79us   665.58us     3.24us    8
                  DB CPU                                             N/A       N/A  4,616,989.83              15.44                                                 8
       Commit     log file sync                             1.837786E+09       0.0  3,342,745.04    1.82ms    11.18      1.82ms     1.79ms     1.89ms    34.11us    8
       Cluster    gc current block congested                 156,977,709       0.0  2,160,028.46   13.76ms     7.23     12.62ms     3.01ms    14.87ms     3.90ms    8
       Cluster    gc cr block busy                            92,026,212       0.0  1,935,056.26   21.03ms     6.47     21.03ms    20.34ms    21.50ms   397.91us    8
       Cluster    gc cr block congested                      121,323,662       0.0  1,785,800.87   14.72ms     5.97     13.42ms     3.09ms    15.52ms     4.18ms    8
       Cluster    gc current grant 2-way                    4.552465E+09       0.0  1,525,217.93 335.03us      5.10    340.87us   274.14us   618.98us   114.62us    8
       Cluster    gc current grant congested                 104,437,287       0.0  1,461,346.33   13.99ms     4.89     13.57ms    10.84ms    15.97ms     1.46ms    8
       Cluster    gc current block 3-way                    4.028686E+09       0.0  1,387,101.85 344.31us      4.64    343.56us   300.22us   357.35us    18.97us    8
       Cluster    gc current grant busy                      434,040,137       0.0  1,191,774.49    2.75ms     3.99      2.75ms     1.92ms     2.98ms   341.11us    8
```

The screenshot below was captured from the Oracle AWR report, highlights the Physical Reads/Sec, Physical Writes/Sec and Transactions per Seconds for the container database. We captured about 364k IOPS (244k Reads/s and 120k Writes/s) with the 43k TPS while running multiple databases workloads for 12 hours.

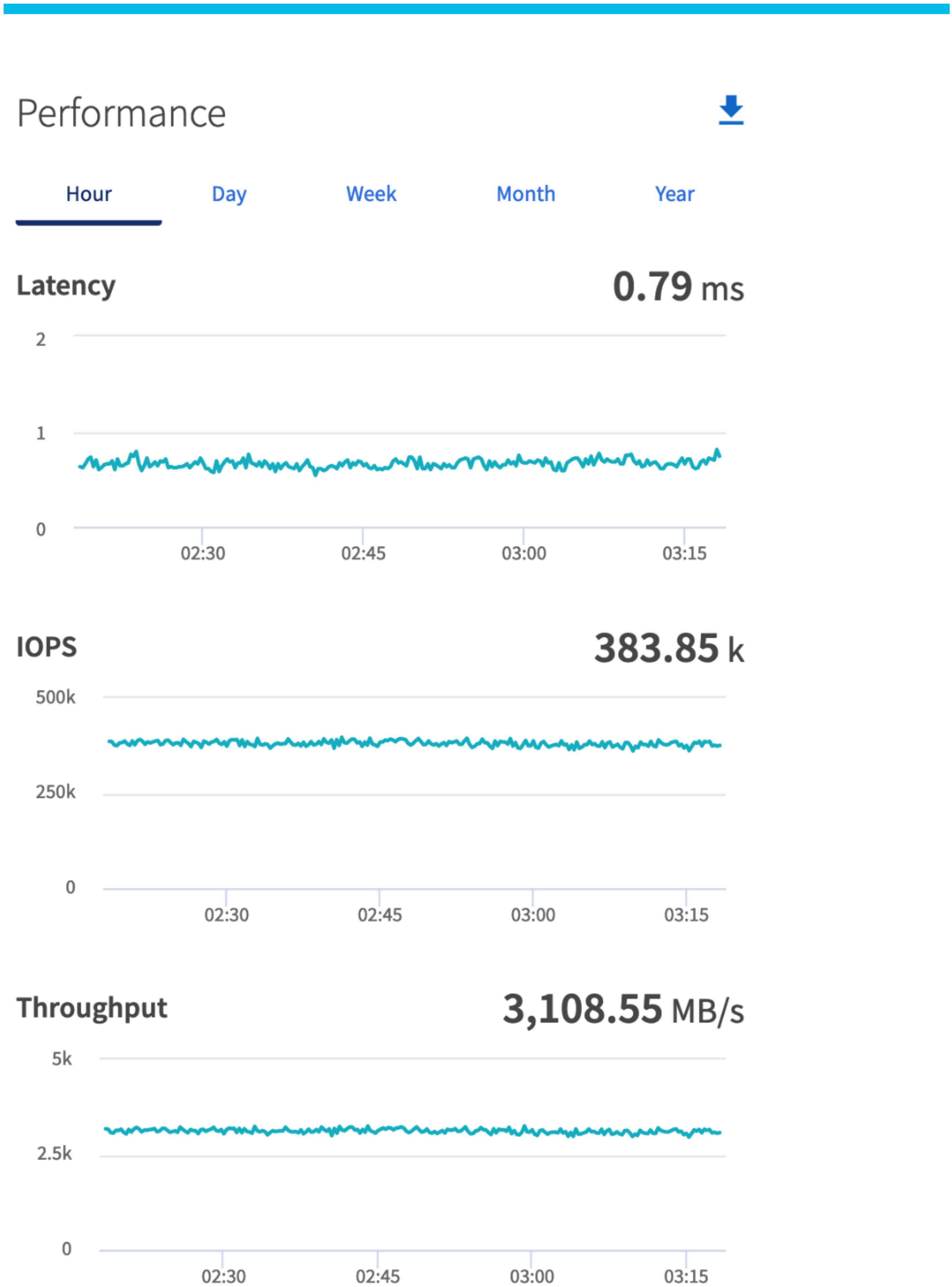
System Statistics - Per Second DB/Inst: SOECDB/soecdb1 Snaps: 1116-1128

I#	Logical Reads/s	Physical Reads/s	Physical Writes/s	Redo Size (k)/s	Block Changes/s	User Calls/s	Execs/s	Parses/s	Logons/s	Txns/s
1	657,678.58	30,628.4	14,876.3	18,986.7	115,246.1	16,034.1	55,803.3	6,380.1	1.89	5,341.5
2	606,470.58	30,998.8	15,211.3	19,381.7	117,487.0	16,353.9	56,870.6	6,500.1	1.89	5,447.9
3	599,526.66	30,605.2	14,945.5	19,122.6	115,941.1	16,136.7	56,120.1	6,414.3	1.89	5,375.5
4	742,544.56	31,014.9	15,197.9	19,452.5	117,802.5	16,404.4	57,057.8	6,523.0	1.89	5,464.8
5	665,268.90	30,722.2	15,054.7	19,310.1	116,968.5	16,278.0	56,610.9	6,469.8	1.89	5,422.6
6	634,732.79	30,697.5	15,352.9	19,368.5	117,552.8	16,254.3	56,524.4	6,461.1	1.89	5,414.7
7	592,387.02	28,532.0	14,228.9	18,023.2	109,344.6	15,103.1	52,544.8	6,006.0	1.89	5,031.0
8	603,722.47	30,848.9	15,052.2	19,330.8	116,968.0	16,278.1	56,621.9	6,472.6	1.89	5,422.6
Sum	5,102,331.57	244,047.9	119,919.9	152,976.0	927,310.7	128,842.6	448,153.8	51,226.8	15.11	42,920.7
Avg	637,791.45	30,506.0	14,990.0	19,122.0	115,913.8	16,105.3	56,019.2	6,403.4	1.89	5,365.1
Std	50,427.50	812.8	343.5	469.4	2,791.7	421.5	1,459.0	166.8	0.00	140.5

The screenshot below shows the NetApp Storage array “Q S P S (qos statistics performance show)” when two OLTP database was running the workload at the same time. The screenshot shows the average IOPS “370k” with the average throughput of “3.4 MB/s” with the average latency around “0.6 millisecond”.

(qos statistics performance show)

Policy Group	IOPS	Throughput	Latency	Is Adaptive?	Is Shared?
-total-	437853	3755.46MB/s	346.00us	-	-
User-Best-Effort	437482	3755.44MB/s	346.00us	false	true
_System-Work	371	18.63KB/s	359.00us	false	true
-total-	388085	3065.11MB/s	677.00us	-	-
User-Best-Effort	387172	3065.02MB/s	677.00us	false	true
_System-Work	913	85.72KB/s	470.00us	false	true
-total-	402871	3228.58MB/s	469.00us	-	-
User-Best-Effort	402491	3228.52MB/s	468.00us	false	true
_System-Work	380	68.58KB/s	1357.00us	false	true
-total-	389893	3128.84MB/s	660.00us	-	-
User-Best-Effort	389544	3128.82MB/s	661.00us	false	true
_System-Work	349	18.53KB/s	266.00us	false	true
-total-	352735	2835.98MB/s	546.00us	-	-
User-Best-Effort	351820	2835.90MB/s	547.00us	false	true
_System-Work	915	82.51KB/s	202.00us	false	true
-total-	343014	2785.67MB/s	604.00us	-	-
User-Best-Effort	342690	2785.66MB/s	603.00us	false	true
_System-Work	324	15.62KB/s	1.62ms	false	true


Performance
Hour
Day
Week
Month
Year
Latency
0.79 ms
2
1
0
02:30
02:45
03:00
03:15
IOPS
383.85 k
500k
250k
0
02:30
02:45
03:00
03:15
Throughput
3,108.55 MB/s
5k
2.5k
0
02:30
02:45
03:00
03:15

The screenshot below shows the NetApp Storage array cluster statistics performance when two OLTP database was running the workload at the same time. In the multiple OLTP database use-case the same behavior of storage cluster utilization (70%) was observed.

FlexPod-A800: cluster.cluster: 3/15/2023 18:59:56

cpu avg	cpu busy	total ops	nfs-ops	cifs-ops	fcache ops	total recv	total sent	data busy	data recv	data sent	cluster busy	cluster recv	cluster sent	disk read	disk write	pkts recv	pkts sent
72%	85%	392295	392295	0	0	1.15GB	2.01GB	4%	1.15GB	2.01GB	0%	48.8KB	48.8KB	3.25GB	1.31GB	475239	426488
72%	84%	389851	389851	0	0	1.15GB	2.02GB	4%	1.15GB	2.02GB	0%	44.9KB	44.9KB	3.28GB	1.39GB	477406	428175
69%	83%	383723	383723	0	0	1.14GB	2.02GB	4%	1.14GB	2.02GB	0%	46.0KB	46.0KB	3.29GB	1.31GB	472791	427127
70%	86%	401362	401362	0	0	1.18GB	2.06GB	4%	1.18GB	2.06GB	0%	48.8KB	48.8KB	3.36GB	1.16GB	482809	438689
70%	85%	389050	389050	0	0	1.15GB	2.02GB	4%	1.15GB	2.02GB	0%	42.6KB	42.6KB	3.33GB	1.29GB	474054	427451
69%	85%	383734	383734	0	0	1.14GB	1.99GB	4%	1.14GB	1.99GB	0%	44.6KB	44.6KB	3.27GB	1.22GB	468286	421539
70%	83%	378727	378727	0	0	1.13GB	1.98GB	4%	1.13GB	1.98GB	0%	50.7KB	50.7KB	3.33GB	1.40GB	467984	420112
69%	82%	382668	382668	0	0	1.12GB	1.95GB	4%	1.12GB	1.95GB	0%	46.6KB	46.4KB	3.16GB	1.19GB	464958	416813
71%	82%	395961	395961	0	0	1.17GB	2.07GB	4%	1.17GB	2.07GB	0%	47.8KB	47.8KB	3.20GB	1.27GB	487275	440078
67%	80%	376050	376050	0	0	1.10GB	1.94GB	4%	1.10GB	1.94GB	0%	52.4KB	52.5KB	3.05GB	1.18GB	459027	412471
68%	79%	379025	379025	0	0	1.12GB	1.97GB	4%	1.12GB	1.97GB	0%	41.9KB	41.8KB	3.10GB	1.14GB	468349	420238

The screenshot below captured from Oracle AWR report shows the SOECDB database "IO Profile" for the "Reads/s" and "Writes/s" requests for the entire 12-hour duration of the test. As the screenshots shows, the Total Requests (Read and Write Per Second) were around "379k" with Total (MB) Read+Write Per Second was around "3090" MB/s for the Container database while running the workload test on two databases at the same time.

IO Profile (Global) DB/Inst: SOECDB/soecdb1 Snaps: 1116-1128

Statistic	Read+Write/s	Reads/s	Writes/s
Total Requests	379,415.51	243,646.60	135,768.92
Database Requests	338,534.17	243,486.55	95,047.62
Optimized Requests	0.00	0.00	0.00
Redo Requests	18,179.11	N/A	18,179.11
Total (MB)	3,090.38	1,991.15	1,099.23
Database (MB)	2,838.17	1,903.05	935.12
Optimized Total (MB)	0.00	0.00	0.00
Redo (MB)	149.11	N/A	149.11
Database (blocks)	363,286.23	243,590.88	119,695.35
Via Buffer Cache (blocks)	363,250.30	243,567.39	119,682.92
Direct (blocks)	35.91	23.47	12.44

The screenshot below captured from the Oracle AWR report, shows the Container database "Interconnect Client Statistics Per Second" for the entire 12-hour duration of the test. As the screenshots shows, Interconnect Sent and Received Statistics were average around "1800 MB/s" while running both the OLTP database workload test.

Interconnect Client Statistics (per Second)DB/Inst: SOECDB/soecdb1 Snaps: 111

	Sent (MB/s)						Received (MB/s)					
I#	Total	Cache	IPQ	DLM	PNG	Misc	Total	Cache	IPQ	DLM	PNG	Misc
1	226.5	164.5	0.1	56.1	0.0	5.9	221.0	164.6	0.1	55.1	0.0	1.3
2	222.1	165.4	0.0	56.5	0.0	0.2	224.8	169.3	0.1	54.6	0.0	0.9
3	219.2	163.0	0.1	56.0	0.0	0.2	221.8	166.2	0.1	54.7	0.0	0.9
4	210.9	159.5	0.1	51.1	0.0	0.2	216.2	166.2	0.1	49.0	0.0	0.9
5	226.9	168.5	0.0	58.1	0.0	0.2	224.8	166.7	0.1	57.2	0.0	0.9
6	224.2	168.7	0.1	55.2	0.0	0.2	222.3	167.7	0.1	53.7	0.0	0.9
7	242.9	174.7	0.1	67.8	0.0	0.2	230.3	159.7	0.1	69.6	0.0	0.9
8	225.7	167.9	0.1	57.4	0.0	0.2	225.9	169.0	0.1	55.9	0.0	0.9
Sum	1,798.4	1,332.2	0.7	458.2	0.0	7.3	1,787.1	1,329.2	0.7	449.9	0.0	7.3
Avg	224.8	166.5	0.1	57.3	0.0	0.9	223.4	166.2	0.1	56.2	0.0	0.9
Std	9.0	4.6	0.0	4.7	0.0	2.0	4.1	3.0	0.0	5.9	0.0	0.1

For the entire 12-hour test, we observed the system performance (IOPS, Latency and Throughput) was consistent throughout and we did not observe any dips in performance while running multiple OLTP database stress test.

One DSS Database Performance

DSS database workloads are generally sequential in nature, read intensive and exercise large IO size. DSS database workload runs a small number of users that typically exercise extremely complex queries that run for hours. For running oracle database multitenancy architecture, we configured one container database as SHCDB and into that container, we created one pluggable database as SHPDB as explained earlier.

We configured 4 TB of SHPDB pluggable database by loading Swingbench "SH" schema into Datafile Tablespace. The screenshot below shows the database summary for the "SHCDB" database running for 12-hour duration. The container database "SHCDB" was also running with one pluggable databases "SHPDB" and the pluggable database was running the Swingbench SH workload for the entire 12-hour duration of the test.

```
Database Summary
~~~~~~~~~~~~~~~~
                              Database                                    Snapshot Ids     Number of Instances    Number of Hosts     Report Total (minutes)
---------------------------------------------------------------------   ----------------- --------------------- ------------------- -----------------------
        Id Name      Unique Name Role              Edition RAC CDB Block Size    Begin      End  In Report    Total  In Report    Total    DB time Elapsed time
---------- --------- ----------- ----------------- ------- --- --- ----------  -------- -------- --------- -------- --------- -------- ---------- -----------
3205256294 SHCDB     shcdb       PRIMARY           EE      YES YES       8192        14       27         8        8         8        8  11,551.09      725.45
Database Instances Included In Report
-> Listed in order of instance number, I#
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

                                                                                                                       Avg Active
 I# Instance  Host       Startup         Begin Snap Time End Snap Time   Release         Elapsed Time(min) DB time(min) Up Time(hrs)  Sessions Platform
---- -------- ---------- --------------- --------------- --------------- --------------- ----------------- ------------ ------------ --------- ---------------
   1 shcdb1   flex1      08-Mar-23 23:55 09-Mar-23 00:28 09-Mar-23 12:28 21.0.0.0.0                 720.07     1,444.25        12.55      2.01 Linux x86 64-bi
   2 shcdb2   flex2      08-Mar-23 23:50 09-Mar-23 00:23 09-Mar-23 12:23 21.0.0.0.0                 720.07     1,443.81        12.55      2.01 Linux x86 64-bi
   3 shcdb3   flex3      08-Mar-23 23:50 09-Mar-23 00:23 09-Mar-23 12:23 21.0.0.0.0                 720.07     1,443.88        12.55      2.01 Linux x86 64-bi
   4 shcdb4   flex4      08-Mar-23 23:51 09-Mar-23 00:23 09-Mar-23 12:24 21.0.0.0.0                 720.08     1,443.77        12.55      2.01 Linux x86 64-bi
   5 shcdb5   flex5      08-Mar-23 23:50 09-Mar-23 00:23 09-Mar-23 12:23 21.0.0.0.0                 720.08     1,443.71        12.55      2.00 Linux x86 64-bi
   6 shcdb6   flex6      08-Mar-23 23:50 09-Mar-23 00:22 09-Mar-23 12:22 21.0.0.0.0                 720.07     1,443.76        12.55      2.01 Linux x86 64-bi
   7 shcdb7   flex7      08-Mar-23 23:50 09-Mar-23 00:23 09-Mar-23 12:23 21.0.0.0.0                 720.07     1,443.74        12.55      2.01 Linux x86 64-bi
   8 shcdb8   flex8      08-Mar-23 23:50 09-Mar-23 00:23 09-Mar-23 12:23 21.0.0.0.0                 720.07     1,444.17        12.55      2.01 Linux x86 64-bi
Open Pluggable Databases at Begin Snap: 3, End Snap: 3
~~~~~~~~~~~~~~~~~~~~~~~~~
```

The screenshot below captured from Oracle AWR report shows the SHCDB database "IO Profile" for the "Reads/s" and "Writes/s" requests for the entire duration of the test. As the screenshots shows, the Total MB (Read and Write Per Second) were around "7543 MB/s" for the SHPDB database while running this test.

```
IO Profile (Global)                         DB/Inst: SHCDB/shcdb1  Snaps: 14-27

Statistic                         Read+Write/s        Reads/s      Writes/s
------------------------------ --------------- -------------- --------------
Total Requests                        9,129.92       8,035.57       1,094.35
Database Requests                     9,031.07       7,950.12       1,080.96
Optimized Requests                        0.00           0.00           0.00
Redo Requests                             2.39            N/A           2.39
Total (MB)                            7,543.78       7,356.44         187.34
Database (MB)                         7,542.28       7,355.11         187.16
Optimized Total (MB)                      0.00           0.00           0.00
Redo (MB)                                 0.01            N/A           0.01
Database (blocks)                   965,411.38     941,454.63      23,956.75
Via Buffer Cache (blocks)             4,093.07       3,773.22         319.84
Direct (blocks)                     961,318.32     937,681.41      23,636.91
                          ----------------------------------------------------
```

The screenshot below shows the NetApp storage array performance (Q S CH S (qos statistics characteristics show)) captured while running Swingbench SH workload on one DSS database. The screenshot shows the average throughput of "7.5 GB/s" while running the one DSS database workload.

Policy Group	IOPS	Throughput	Request size	Read	Concurrency	Is Adaptive?	Is Shared?
-total-	18759	8520.17MB/s	476253B	90%	137	-	-
User-Best-Effort	18690	8520.08MB/s	478006B	90%	137	false	true
_System-Work	69	91.01KB/s	1350B	8%	0	false	true
-total-	20701	9019.20MB/s	456853B	88%	137	-	-
User-Best-Effort	20654	9019.20MB/s	457892B	89%	137	false	true
_System-Work	47	3.92KB/s	85B	0%	0	false	true
-total-	16637	7313.87MB/s	460969B	88%	125	-	-
User-Best-Effort	15941	7313.79MB/s	481090B	91%	125	false	true
_System-Work	564	73.46KB/s	133B	14%	0	false	true
_System-Best-Effort	132	0KB/s	0B	0%	0	false	true
-total-	18057	7783.54MB/s	451992B	88%	111	-	-
User-Best-Effort	17895	7783.41MB/s	456076B	89%	111	false	true
_System-Work	162	129.73KB/s	820B	3%	0	false	true
-total-	21777	9722.52MB/s	468145B	90%	113	-	-
User-Best-Effort	21582	9722.38MB/s	472368B	91%	113	false	true
_System-Work	100	144.74KB/s	1482B	8%	0	false	true
_System-Best-Effort	95	0KB/s	0B	0%	0	false	true

The screenshot below shows the NetApp Storage array cluster statistics performance when one DSS database was running the workload at the same time. In this one DSS database use-case we observed storage cluster utilization were around 28%. The database performance was consistent throughout the test, and we did not observe any dips in performance for entire period of 12-hour test.

FlexPod-A800: cluster.cluster: 3/9/2023 01:48:41

cpu avg	cpu busy	total ops	nfs-ops	cifs-ops	fcache ops	total recv	total sent	data busy	data recv	data sent	cluster busy	cluster recv	cluster sent	disk read	disk write	pkts recv	pkts sent
25%	31%	20038	20038	0	0	395MB	8.42GB	22%	395MB	8.42GB	0%	42.6KB	42.5KB	2.32GB	357MB	161277	1017908
27%	34%	21563	21563	0	0	391MB	9.07GB	24%	391MB	9.07GB	0%	53.6KB	53.9KB	2.73GB	429MB	171839	1094557
27%	35%	21665	21665	0	0	337MB	9.50GB	25%	337MB	9.50GB	0%	128KB	127KB	2.64GB	367MB	170767	1144721
29%	36%	22247	22247	0	0	359MB	9.56GB	26%	359MB	9.56GB	0%	55.0KB	55.8KB	2.80GB	455MB	177103	1152812
27%	33%	22437	22437	0	0	356MB	9.68GB	25%	356MB	9.68GB	0%	45.9KB	45.8KB	2.60GB	214MB	175492	1166453
28%	33%	21199	21199	0	0	341MB	9.34GB	24%	341MB	9.34GB	0%	53.4KB	53.4KB	2.88GB	469MB	169779	1125433
26%	31%	21649	21649	0	0	346MB	9.29GB	24%	346MB	9.29GB	0%	43.9KB	43.9KB	2.60GB	192MB	167320	1119755
28%	34%	21497	21497	0	0	341MB	9.32GB	25%	341MB	9.32GB	0%	43.8KB	43.7KB	2.73GB	619MB	167554	1122800
28%	35%	22034	22034	0	0	354MB	9.58GB	24%	354MB	9.58GB	0%	55.9KB	55.9KB	2.68GB	235MB	172114	1154498
26%	31%	20356	20356	0	0	337MB	8.77GB	25%	337MB	8.77GB	0%	41.0KB	41.0KB	2.63GB	523MB	158368	1058071
28%	33%	20255	20255	0	0	352MB	8.67GB	21%	352MB	8.67GB	0%	44.1KB	44.1KB	2.68GB	619MB	159410	1046691
27%	32%	20643	20643	0	0	345MB	8.97GB	24%	345MB	8.97GB	0%	52.9KB	53.1KB	2.66GB	274MB	162169	1081945
27%	32%	21203	21203	0	0	326MB	9.14GB	24%	326MB	9.14GB	0%	44.1KB	43.9KB	2.43GB	589MB	161619	1102001
30%	34%	22593	22593	0	0	349MB	9.98GB	26%	348MB	9.97GB	0%	60.9KB	60.9KB	2.67GB	535MB	174341	1202059

Multiple OLTP and DSS Database Performance

In this test, we ran Swingbench SOE workloads on both the OLTP (SOEPDB + ENGPDB) databases and Swingbench SH workload on one DSS (SHPDB) Database at the same time for 24-hour and captured the overall system performance. We captured the system performance on small random queries presented via OLTP databases as well as large and sequential transactions submitted via DSS database workload as documented below.

The screenshot below shows the database summary for the "SOECDB" database running for a 24-hour duration. The container database "SOECDB" was running with both the pluggable databases "SOEPDB" and "ENGPDB" and both the pluggable databases were running the Swingbench SOE workload for the entire 24-hour duration of the test.

```
Database Summary
~~~~~~~~~~~~~~~~
                              Database                                 Snapshot Ids     Number of Instances    Number of Hosts     Report Total (minutes)
----------------------------------------------------------------    -----------------  -------------------  -------------------  -----------------------
        Id Name      Unique Name Role             Edition RAC CDB Block Size    Begin      End   In Report    Total  In Report    Total     DB time Elapsed time
---------- -------- ----------- ---------------- ------- --- --- ----------  -------- --------  --------- --------  --------- --------  ---------- ------------
1630447391 SOECDB    soecdb      PRIMARY          EE      YES YES      8192     1241     1265           8        8          8        8  655,243.32     1,443.22
Database Instances Included In Report
-> Listed in order of instance number, I#
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

                                                                                                                          Avg Active
  I# Instance  Host       Startup         Begin Snap Time End Snap Time   Release         Elapsed Time(min) DB time(min) Up Time(hrs)  Sessions Platform
---- --------- ---------- --------------- --------------- --------------- --------------- ----------------- ------------ ------------ --------- ---------------
   1 soecdb1   flex1      17-Mar-23 18:38 20-Mar-23 15:41 21-Mar-23 15:42 21.0.0.0.0               1,440.97    82,160.71        93.07     57.02 Linux x86 64-bi
   2 soecdb2   flex2      17-Mar-23 17:01 20-Mar-23 15:41 21-Mar-23 15:42 21.0.0.0.0               1,440.98    81,718.70        94.68     56.71 Linux x86 64-bi
   3 soecdb3   flex3      09-Mar-23 15:51 20-Mar-23 15:40 21-Mar-23 15:41 21.0.0.0.0               1,441.00    81,843.24       287.85     56.80 Linux x86 64-bi
   4 soecdb4   flex4      09-Mar-23 15:10 20-Mar-23 15:40 21-Mar-23 15:41 21.0.0.0.0               1,441.02    81,928.82       288.53     56.85 Linux x86 64-bi
   5 soecdb5   flex5      09-Mar-23 15:10 20-Mar-23 15:40 21-Mar-23 15:41 21.0.0.0.0               1,440.98    81,895.32       288.52     56.83 Linux x86 64-bi
   6 soecdb6   flex6      09-Mar-23 15:09 20-Mar-23 15:39 21-Mar-23 15:40 21.0.0.0.0               1,440.97    81,974.38       288.52     56.89 Linux x86 64-bi
   7 soecdb7   flex7      09-Mar-23 15:09 20-Mar-23 15:39 21-Mar-23 15:40 21.0.0.0.0               1,440.98    81,853.99       288.52     56.80 Linux x86 64-bi
   8 soecdb8   flex8      17-Mar-23 17:04 20-Mar-23 15:41 21-Mar-23 15:42 21.0.0.0.0               1,440.98    81,868.16        94.63     56.81 Linux x86 64-bi
Open Pluggable Databases at Begin Snap: 4, End Snap: 4
~~~~~~~~~~~~~~~~~~~~~~~~~
```

The screenshot below shows the database summary for the "SHCDB" database running for a 24-hour duration. The container database "SHCDB" was also running with one pluggable databases "SHPDB" and the pluggable database was running the Swingbench SH workload for the entire 24-hour duration of the tests.

```
Database Summary
~~~~~~~~~~~~~~~~
                              Database                                 Snapshot Ids     Number of Instances    Number of Hosts     Report Total (minutes)
----------------------------------------------------------------    -----------------  -------------------  -------------------  -----------------------
        Id Name      Unique Name Role             Edition RAC CDB Block Size    Begin      End   In Report    Total  In Report    Total     DB time Elapsed time
---------- -------- ----------- ---------------- ------- --- --- ----------  -------- --------  --------- --------  --------- --------  ---------- ------------
3205256294 SHCDB     shcdb       PRIMARY          EE      YES YES      8192      299      323           8        8          8        8   17,439.66     1,443.51
Database Instances Included In Report
-> Listed in order of instance number, I#
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

                                                                                                                          Avg Active
  I# Instance  Host       Startup         Begin Snap Time End Snap Time   Release         Elapsed Time(min) DB time(min) Up Time(hrs)  Sessions Platform
---- --------- ---------- --------------- --------------- --------------- --------------- ----------------- ------------ ------------ --------- ---------------
   1 shcdb1    flex1      17-Mar-23 19:28 20-Mar-23 15:42 21-Mar-23 15:43 21.0.0.0.0               1,441.27     1,512.30        92.25      1.05 Linux x86 64-bi
   2 shcdb2    flex2      17-Mar-23 19:28 20-Mar-23 15:42 21-Mar-23 15:43 21.0.0.0.0               1,441.27     1,454.65        92.25      1.01 Linux x86 64-bi
   3 shcdb3    flex3      17-Mar-23 19:27 20-Mar-23 15:41 21-Mar-23 15:42 21.0.0.0.0               1,441.28     1,450.88        92.25      1.01 Linux x86 64-bi
   4 shcdb4    flex4      17-Mar-23 19:27 20-Mar-23 15:41 21-Mar-23 15:42 21.0.0.0.0               1,441.32     2,892.50        92.25      2.01 Linux x86 64-bi
   5 shcdb5    flex5      17-Mar-23 19:27 20-Mar-23 15:40 21-Mar-23 15:42 21.0.0.0.0               1,441.27     1,450.92        92.25      1.01 Linux x86 64-bi
   6 shcdb6    flex6      17-Mar-23 19:26 20-Mar-23 15:40 21-Mar-23 15:41 21.0.0.0.0               1,441.27     2,892.53        92.25      2.01 Linux x86 64-bi
   7 shcdb7    flex7      17-Mar-23 19:26 20-Mar-23 15:40 21-Mar-23 15:41 21.0.0.0.0               1,441.27     2,893.23        92.25      2.01 Linux x86 64-bi
   8 shcdb8    flex8      17-Mar-23 19:28 20-Mar-23 15:42 21-Mar-23 15:43 21.0.0.0.0               1,441.27     2,892.64        92.25      2.01 Linux x86 64-bi
Open Pluggable Databases at Begin Snap: 3, End Snap: 3
~~~~~~~~~~~~~~~~~~~~~~~~~
```

The screenshot below was captured from the Oracle AWR report while running the Swingbench SOE and SH workload tests on all the three databases for 24-hours. The screenshot shows the "OS Statistics by Instance" while the system was running mixed workload. As shown below, the workload was equally spread across all the databases clusters while the average CPU utilization was around 20% overall.

```
OS Statistics By Instance          DB/Inst: SOECDB/soecdb1  Snaps: 1241-1265
-> Listed in order of instance number, I#
-> End values are diplayed only if different from begin values

           CPU              Load                % CPU                                  Time (s)                         Memory           End Values
      -----------------  -------------  ---------------------------------  ---------------------------------------------  ---------  -------------------------
  I#  #CPUs #Core #Sckt   Begin    End  % Busy  % Usr  % Sys  % WIO  % Idl             Busy           Idle          Total         MB  #CPU #Cor #Sck Memory (M)
----  ----- ----- -----  ------ ------  ------ ------ ------ ------ ------  --------------- -------------- --------------  ---------  ---- ---- ---- ----------
   1    112    56     2    17.5   13.1    20.0   13.4    3.8     .0   80.0      1,926,050.7    7,685,102.3    9,611,153.0  514,983.8
   2    112    56     2    13.2   17.7    17.9   12.5    3.2     .0   82.1      1,726,545.6    7,898,351.2    9,624,896.8  514,983.8
   3    112    56     2    13.2   13.6    17.1   11.6    3.2     .0   82.9      1,645,104.5    7,981,650.9    9,626,755.4  514,983.8
   4    112    56     2    13.7   10.3    16.9   11.4    3.2     .0   83.1      1,625,793.3    7,999,044.6    9,624,837.9  514,983.7
   5    112    56     2    16.7   11.9    17.5   11.7    3.4     .0   82.5      1,682,425.1    7,946,733.2    9,629,158.3  514,984.0
   6    112    56     2    18.1   11.5    17.9   12.1    3.3     .0   82.1      1,719,339.7    7,905,422.6    9,624,762.3  514,983.8
   7    112    56     2    17.3   13.0    19.5   13.0    3.7     .0   80.5      1,880,923.3    7,744,725.9    9,625,649.2  514,983.8
   8    112    56     2    13.3   13.2    18.0   12.2    3.4     .0   82.0      1,736,153.9    7,889,769.4    9,625,923.3  514,984.0
 ~~~                                                                        ~~~~~~~~~~~~~~~ ~~~~~~~~~~~~~~ ~~~~~~~~~~~~~~
 Sum                                                                           13,942,336.2   63,050,800.1   76,993,136.3
                    ------------------------------------------------------------------------------------------------------------------------
```

The screenshot below was captured from the Oracle AWR report shows the "Top Timed Events" for the SOEPDB database while running Swingbench SOE workloads on both the pluggable (SOEPDB and ENGPDB) databases for the entire 24-hour duration of the test.

```
Top Timed Events                    DB/Inst: SOECDB/soecdb1  Snaps: 1241-1265
-> Instance '*'  - cluster wide summary
->          '*'  Waits, %Timeouts, Wait Time Total(s)      : Cluster-wide total for the wait event
->          '*'  'Wait Time Avg'                         : Cluster-wide average computed as (Wait Time Total / Event Waits)
->          '*'  Summary 'Avg Wait Time '                 : Per-instance 'Wait Time Avg ' used to compute the following statistics
->          '*'  [Avg/Min/Max/Std Dev]                    : average/minimum/maximum/standard deviation of per-instance 'Wait Time Avg'
->          '*'  Cnt                                      : count of instances with wait times for the event
```

	Wait		Event		Wait Time			Summary Avg Wait Time				
I#	Class	Event	Waits	%Timeouts	Total(s)	Avg Wait	%DB time	Avg	Min	Max	Std Dev	Cnt
*	User I/O	db file sequential read	1.617582E+10	0.0	1.8970250E+07	1.17ms	48.25	1.17ms	1.17ms	1.19ms	8.87us	8
		DB CPU	N/A	N/A	6,879,488.94		17.50					8
	Commit	log file sync	2.841786E+09	0.0	5,845,650.04	2.06ms	14.87	2.06ms	1.95ms	2.14ms	68.86us	8
	System I/O	db file parallel write	2.158353E+09	0.0	2,848,549.79	1.32ms	7.25	1.32ms	1.27ms	1.36ms	25.04us	8
	Cluster	gc current grant 2-way	7.464309E+09	0.0	1,598,255.26	214.12us	4.07	214.31us	199.84us	237.43us	11.84us	8
	Cluster	gc current block 3-way	6.383037E+09	0.0	1,553,431.07	243.37us	3.95	243.15us	229.34us	249.65us	6.75us	8
	Cluster	gc cr block busy	66,901,097	0.0	1,492,476.15	22.31ms	3.80	22.31ms	22.07ms	22.57ms	173.85us	8
	Cluster	gc cr block 3-way	3.863840E+09	0.0	988,511.60	255.84us	2.51	255.52us	238.63us	262.83us	8.00us	8
	System I/O	log file parallel write	2.927532E+09	0.0	887,721.73	303.23us	2.26	303.03us	294.43us	313.94us	6.56us	8
	Cluster	gc cr grant 2-way	3.399139E+09	0.0	689,774.42	202.93us	1.75	202.93us	189.55us	222.58us	10.06us	8

The screenshot below was captured from the Oracle AWR report, highlights the Physical Reads/Sec, Physical Writes/Sec and Transactions per Seconds for the Container SOECDB Database. We captured around 288k IOPS (192k Reads/s and 96k Writes/s) with the 33k TPS while running multiple databases workloads.

System Statistics - Per Second   DB/Inst: SOECDB/soecdb1 Snaps: 1241-1265

I#	Logical Reads/s	Physical Reads/s	Physical Writes/s	Redo Size (k)/s	Block Changes/s	User Calls/s	Execs/s	Parses/s	Logons/s	Txns/s
1	500,214.40	24,005.5	12,122.3	15,448.7	90,563.8	12,506.0	43,514.2	4,971.1	2.39	4,164.5
2	629,970.51	22,842.1	12,201.3	15,771.7	92,061.1	12,718.5	44,236.7	5,061.0	2.39	4,235.4
3	467,538.51	24,323.8	11,950.5	15,451.9	90,334.5	12,470.0	43,370.8	4,955.5	2.38	4,152.5
4	497,447.73	24,110.0	11,858.2	15,384.1	89,525.4	12,351.6	42,970.8	4,912.2	2.38	4,113.0
5	498,257.05	24,206.6	11,901.8	15,441.5	90,045.6	12,422.1	43,218.5	4,939.7	2.39	4,136.5
6	527,803.62	23,924.9	12,014.7	15,335.4	89,666.0	12,279.4	42,718.8	4,882.9	2.38	4,088.9
7	476,616.56	24,094.1	12,117.9	15,462.9	90,537.9	12,414.9	43,176.7	4,933.7	2.38	4,134.1
8	502,280.42	24,629.1	12,067.7	15,439.0	89,911.2	12,413.6	43,174.0	4,933.4	2.38	4,133.7
Sum	4,100,128.80	192,135.9	96,234.5	123,735.2	722,645.4	99,576.0	346,380.4	39,589.5	19.07	33,158.6
Avg	512,516.10	24,017.0	12,029.3	15,466.9	90,330.7	12,447.0	43,297.5	4,948.7	2.38	4,144.8
Std	50,747.95	522.1	119.2	130.4	796.3	129.6	449.2	52.6	0.00	43.3

The screenshot below was captured from the Oracle AWR report shows the SOECDB database "IO Profile" for the "Reads/s" and "Writes/s" requests for the entire 24-hour duration of the test. As the screenshots shows, the Total Requests (Read and Write Per Second) were around "301k" with Total (MB) Read+Write Per Second was around "2455" MB/s for the SOECDB database while running the mixed workload test.

IO Profile (Global)   DB/Inst: SOECDB/soecdb1 Snaps: 1241-1265

Statistic	Read+Write/s	Reads/s	Writes/s
Total Requests	301,922.88	191,681.00	110,241.88
Database Requests	268,137.80	191,533.65	76,604.15
Optimized Requests	0.00	0.00	0.00
Redo Requests	16,889.73	N/A	16,889.73
Total (MB)	2,455.45	1,571.86	883.59
Database (MB)	2,249.41	1,498.74	750.67
Optimized Total (MB)	0.00	0.00	0.00
Redo (MB)	120.65	N/A	120.65
Database (blocks)	287,924.38	191,838.77	96,085.61
Via Buffer Cache (blocks)	287,684.11	191,610.17	96,073.93
Direct (blocks)	239.83	228.16	11.68

The screenshot below was captured from the Oracle AWR report shows the SOECDB database “Interconnect Client Statistics Per Second” for the entire 24-hour duration of the test. As the screenshots shows, Interconnect Sent and Received Statistics were average around “1400 MB/s” while running the mixed workload test.

Interconnect Client Statistics (per Second)DB/Inst: SOECDB/soecdb1 Snaps: 124

	Sent (MB/s)						Received (MB/s)					
I#	Total	Cache	IPQ	DLM	PNG	Misc	Total	Cache	IPQ	DLM	PNG	Misc
1	168.8	125.1	0.1	43.4	0.0	0.2	174.3	130.5	0.1	42.8	0.0	0.9
2	173.7	130.8	0.1	42.6	0.0	0.2	182.6	138.0	0.1	43.7	0.0	0.9
3	171.4	126.9	0.1	44.2	0.0	0.2	171.9	127.9	0.1	43.0	0.0	0.9
4	161.0	120.6	0.2	39.9	0.0	0.2	165.9	126.9	0.2	38.0	0.0	0.8
5	175.7	130.3	0.2	45.0	0.0	0.2	173.3	128.2	0.1	44.1	0.0	0.8
6	172.8	129.5	0.2	42.9	0.0	0.2	169.1	126.2	0.1	41.9	0.0	0.8
7	201.3	141.9	0.1	53.6	0.0	5.7	183.0	127.5	0.1	54.1	0.0	1.2
8	173.2	127.6	0.1	45.3	0.0	0.2	171.0	125.8	0.1	44.3	0.0	0.9
Sum	1,397.9	1,032.9	0.9	356.8	0.0	7.2	1,391.1	1,030.9	0.9	352.0	0.0	7.2
Avg	174.7	129.1	0.1	44.6	0.0	0.9	173.9	128.9	0.1	44.0	0.0	0.9
Std	11.6	6.1	0.1	4.0	0.0	2.0	6.1	4.0	0.0	4.6	0.0	0.1

The screenshot below was captured from the Oracle AWR report shows the “Top Timed Events” for the SHCDB database while running Swingbench SH workloads on the pluggable (SHPDB) database for the entire 24-hour duration of the test.

```
Top Timed Events                          DB/Inst: SHCDB/shcdb1  Snaps: 299-323
-> Instance '*'  - cluster wide summary
->          '*'  Waits, %Timeouts, Wait Time Total(s)      : Cluster-wide total for the wait event
->          '*'  'Wait Time Avg'                        : Cluster-wide average computed as (Wait Time Total / Event Waits)
->          '*'  Summary 'Avg Wait Time '                : Per-instance 'Wait Time Avg ' used to compute the following statistics
->          '*'  [Avg/Min/Max/Std Dev]                     : average/minimum/maximum/standard deviation of per-instance 'Wait Time Avg'
->          '*'  Cnt                                       : count of instances with wait times for the event
```

	Wait		Event		Wait Time			Summary Avg Wait Time				
I#	Class	Event	Waits	%Timeouts	Total(s)	Avg Wait	%DB time	Avg	Min	Max	Std Dev	Cnt
*		DB CPU	N/A	N/A	466,995.53		44.63					8
	User I/O	direct path read	389,283,284	0.0	462,621.96	1.19ms	44.21	1.19ms	1.14ms	1.24ms	32.00us	8
	User I/O	direct path read temp	77,735,811	0.0	86,721.18	1.12ms	8.29	1.12ms	1.07ms	1.15ms	25.65us	8
	User I/O	local write wait	14,497,394	0.0	11,386.12	785.39us	1.09	781.40us	699.73us	829.44us	40.36us	8
	System I/O	db file parallel write	14,678,823	0.0	10,382.72	707.33us	0.99	704.44us	639.45us	743.96us	33.26us	8
	User I/O	direct path write temp	43,540,717	0.0	8,421.04	193.41us	0.80	195.26us	179.37us	241.27us	20.76us	8
	User I/O	db file scattered read	1,787,287	0.0	7,054.41	3.95ms	0.67	4.03ms	3.19ms	4.40ms	452.33us	8
	Cluster	gc current block busy	280,943	0.0	2,797.36	9.96ms	0.27	10.23ms	8.59ms	13.96ms	1.72ms	8
	System I/O	control file sequential read	3,286,053	0.0	2,695.39	820.25us	0.26	820.32us	799.97us	842.91us	15.93us	8
	Cluster	gc cr multi block grant	2,172,458	0.0	2,517.97	1.16ms	0.24	1.14ms	901.20us	1.33ms	149.27us	8

The screenshot below was captured from the Oracle AWR report shows the SHCDB database “IO Profile” for the “Reads/s” and “Writes/s” requests for the entire 24-hour duration of the test. As the screenshots shows, the Total MB (Read and Write Per Second) were around “4071 MB/s” for the SHPDB database while running this test.

```
IO Profile (Global)                          DB/Inst: SHCDB/shcdb1  Snaps: 299-323

Statistic                        Read+Write/s        Reads/s       Writes/s
------------------------------ -------------- -------------- --------------
Total Requests                       4,992.91       4,355.49         637.42
Database Requests                    4,947.43       4,317.47         629.95
Optimized Requests                       0.00           0.00           0.00
Redo Requests                            1.83            N/A           1.83
Total (MB)                           4,071.23       3,958.30         112.93
Database (MB)                        4,070.53       3,957.70         112.83
Optimized Total (MB)                     0.00           0.00           0.00
Redo (MB)                                0.02            N/A           0.02
Database (blocks)                  521,027.64     506,585.55      14,442.08
Via Buffer Cache (blocks)            2,781.22       2,609.80         171.42
Direct (blocks)                    518,246.39     503,975.73      14,270.66
                          ----------------------------------------------------
```

The screenshot below shows the NetApp Storage array "Q S P S (qos statistics performance show)" when all the databases were running the workloads at the same time. The screenshot shows the average IOPS "350k" with the average throughput of "7 GB/s" with the average latency around "1 millisecond."

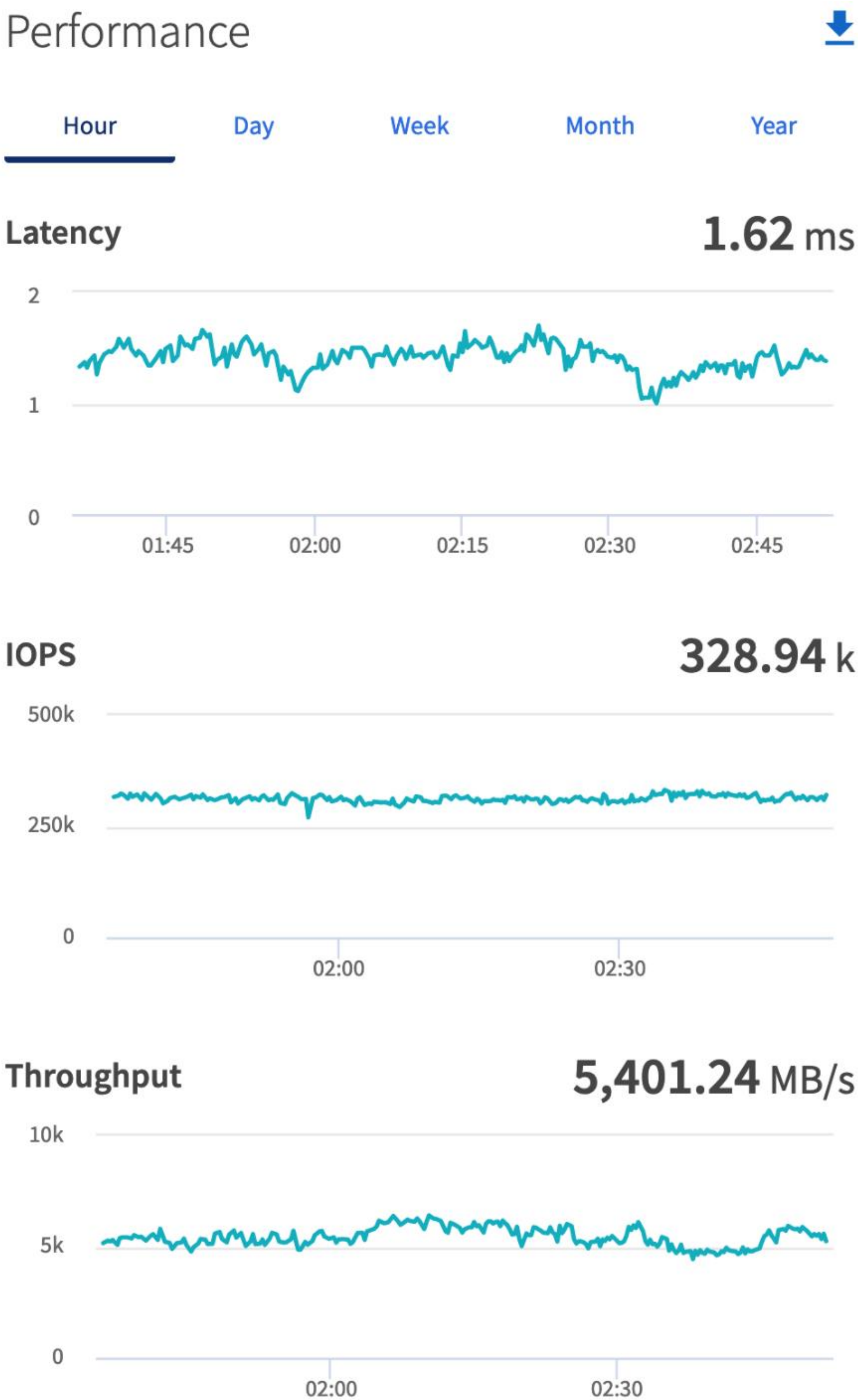
```
Policy Group             IOPS      Throughput    Latency Is Adaptive? Is Shared?
------------------- -------- --------------- ---------- ------------ ----------
-total-               380949     6507.10MB/s   842.00us -  -
User-Best-Effort      377008     6507.01MB/s   850.00us false true
_System-Work            3941       98.21KB/s    47.00us false true
-total-               355601     5104.02MB/s   831.00us -  -
User-Best-Effort      355276     5103.60MB/s   831.00us false true
_System-Work             325      423.61KB/s   163.00us false true
-total-               369202     6629.52MB/s   883.00us -  -
User-Best-Effort      369156     6629.45MB/s   883.00us false true
_System-Work              46       70.75KB/s   217.00us false true
-total-               354521     8371.03MB/s     1.56ms -  -
User-Best-Effort      354464     8371.02MB/s     1.56ms false true
_System-Work              57        8.38KB/s   508.00us false true
-total-               333099     9052.17MB/s  1428.00us -  -
User-Best-Effort      333015     9052.17MB/s  1428.00us false true
_System-Work              84        2.02KB/s    83.00us false true
-total-               330787     6390.37MB/s  1209.00us -  -
User-Best-Effort      330677     6390.36MB/s  1210.00us false true
_System-Work             110       15.98KB/s   472.00us false true
-total-               373053     6785.99MB/s   999.00us -  -
User-Best-Effort      372972     6785.95MB/s   999.00us false true
_System-Work              81       45.02KB/s   259.00us false true
```

The screenshot below shows the NetApp Storage array "statistics." The screenshot shows the average CPU busy around "73%" with "4.5 GB/s" disk read and "1.2 GB/s" disk write when all the databases were running the workloads at the same time. The storage cluster utilization was the highest with both OLTP and DSS running together generating around ~7GB/sec throughput.

FlexPod-A800: cluster.cluster: 3/20/2023 15:50:05

cpu avg	cpu busy	total ops	nfs-ops	cifs-ops	fcache ops	total recv	total sent	data busy	data recv	data sent	cluster busy	cluster recv	cluster sent	disk read	disk write	pkts recv	pkts sent
73%	84%	352571	352571	0	0	1.08GB	5.37GB	13%	1.08GB	5.36GB	0%	77.1KB	77.1KB	4.59GB	1.24GB	465568	804807
72%	82%	350587	350587	0	0	1.08GB	5.61GB	16%	1.08GB	5.61GB	0%	58.1KB	58.1KB	4.36GB	1.05GB	460282	831660
74%	84%	353611	353611	0	0	1.06GB	5.29GB	15%	1.06GB	5.29GB	0%	67.6KB	67.6KB	4.59GB	1.33GB	458407	795085
69%	79%	341532	341532	0	0	1.04GB	5.65GB	15%	1.04GB	5.65GB	0%	66.7KB	66.7KB	4.31GB	1.03GB	454534	833846
72%	83%	351221	351221	0	0	1.09GB	5.70GB	15%	1.09GB	5.70GB	0%	91.2KB	91.1KB	4.42GB	1.14GB	468155	847305
70%	82%	346933	346933	0	0	1.04GB	5.36GB	15%	1.04GB	5.36GB	0%	55.3KB	55.4KB	4.36GB	1.12GB	452892	799908
65%	74%	322698	322698	0	0	996MB	5.44GB	16%	996MB	5.44GB	0%	82.5KB	82.5KB	4.06GB	979MB	430916	799154
70%	80%	351542	351542	0	0	1.06GB	5.05GB	13%	1.06GB	5.05GB	0%	47.6KB	47.6KB	4.19GB	1.17GB	453642	765419
72%	82%	348965	348965	0	0	1.09GB	5.52GB	15%	1.09GB	5.52GB	0%	42.4KB	42.4KB	4.46GB	1.16GB	466517	826048
73%	82%	333069	333069	0	0	1.01GB	5.61GB	15%	1.01GB	5.61GB	0%	48.5KB	48.5KB	4.43GB	1.34GB	446342	825510
71%	81%	343457	343457	0	0	1.04GB	4.59GB	14%	1.04GB	4.59GB	0%	45.7KB	45.7KB	4.20GB	1.20GB	441046	704818
70%	81%	345948	345948	0	0	1.07GB	5.20GB	14%	1.07GB	5.20GB	0%	42.1KB	42.1KB	4.22GB	1.15GB	457987	783604

The screenshot below shows the NetApp Array GUI when all the databases were running the workloads at the same time.


Performance
Hour
Day
Week
Month
Year
Latency
1.62 ms
2
1
0
01:45
02:00
02:15
02:30
02:45
IOPS
328.94 k
500k
250k
0
02:00
02:30
Throughput
5,401.24 MB/s
10k
5k
0
02:00
02:30

When we ran multiple (OLTP and DSS) databases workloads together, we achieved average around “330k” IOPS, “5.5 GB/s” Throughput with the average latency around “2 milliseconds.” For the entire 24-hour tests, we observed the system performance (IOPS and Throughput) was consistent throughout and we did not observe any dips in performance while running these tests.

Resiliency and Failure Tests

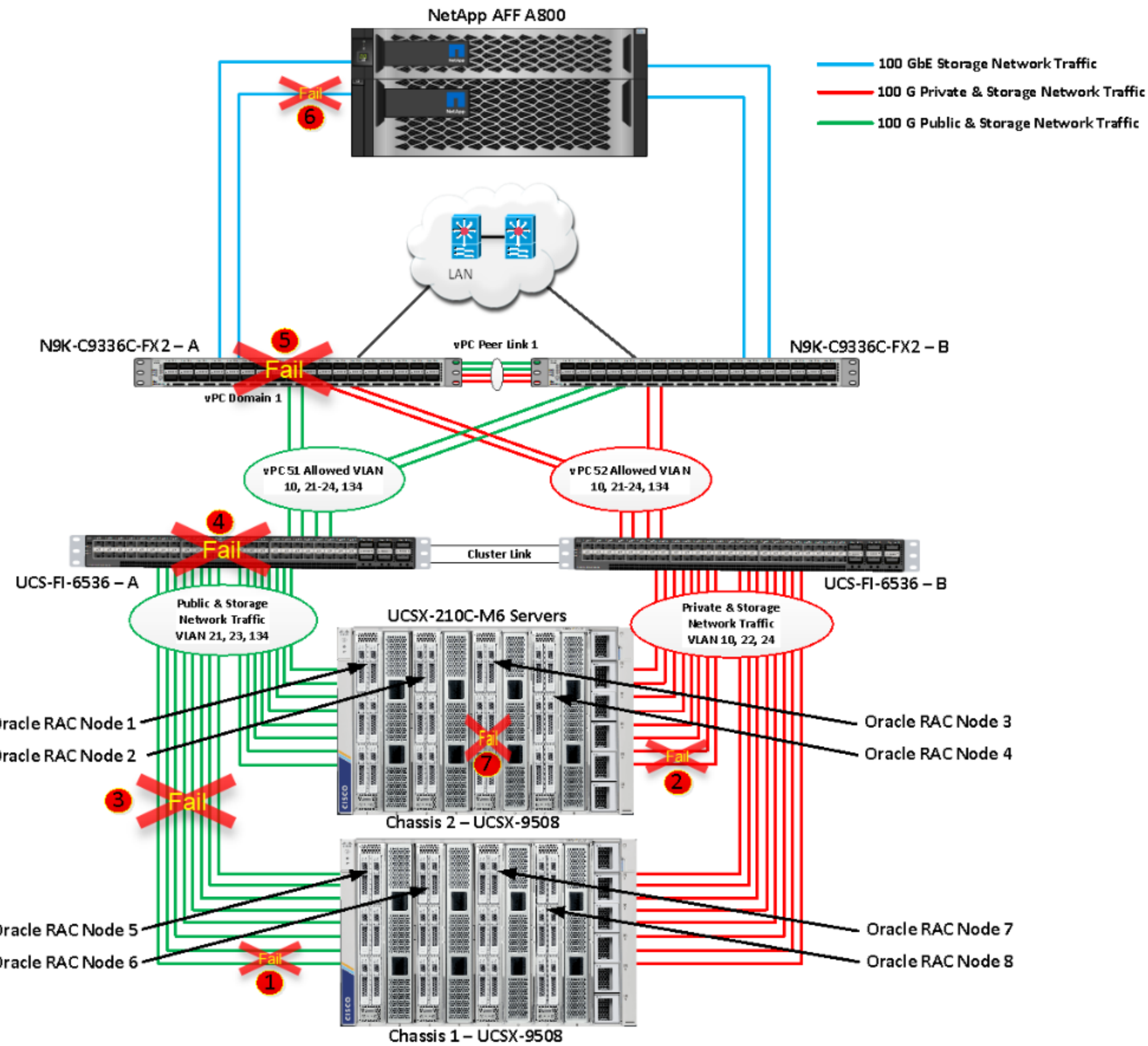
This chapter contains the following:

The goal of these tests was to ensure that the reference architecture withstands commonly occurring failures due to either unexpected crashes, hardware failures or human errors. We conducted many hardware (disconnect power), software (process kills) and OS specific failures that simulate the real world scenarios under stress conditions. In the destructive testing, we will also demonstrate the unique failover capabilities of Cisco UCS components used in this solution. Table 16 highlights the test cases.

Table 16. Hardware Failover Tests

Test Scenario	Tests Performed
Test 1: UCS-X Chassis IFM Link/Links Failure	Run the system on full Database workload. Disconnect one or two links from each Chassis 1 IFM and Chassis 2 IFM by pulling it out and reconnect it after 10-15 minutes. Capture the impact on overall database performance.
Test 2: One of the FI Failure	Run the system on full Database workload. Power Off one of the Fabric Interconnects and check the network traffic on the other Fabric Interconnect and capture the impact on overall database performance.
Test 3: One of the Nexus Switch Failure	Run the system on full Database workload. Power Off one of the Cisco Nexus switches and check the network and storage traffic on the other Nexus switch. Capture the impact on overall database performance.
Test 4: Storage Controller Links Failure	Run the system on full Database workload. Disconnect one link from each of the NetApp Storage Controllers by pulling it out and reconnect it after 10-15 minutes. Capture the impact on overall database performance.
Test 5: RAC Server Node Failure	Run the system on full Database workload. Power Off one of the Linux Hosts and check the impact on database performance.

The architecture below illustrates various failure scenario which can be occurred due to either unexpected crashes or hardware failures. The failure scenario 1 and/or scenario 2 represents the Chassis IFM link failures. Also, scenario 3 represents the Chassis all IFM links failure. Scenario 4 represents one of the Cisco UCS FI failure and similarly, scenario 5 represents one of the Cisco Nexus Switch failures. Scenario 6 represents the NetApp Storage Controllers link failures and Scenario 7 represents one of the Server Node Failures.



Note: All the Hardware failover tests were conducted with all three databases (SOEPDB, ENGPDB and SHPDB) running Swingbench mixed workloads.

As previously explained, we configured to carry Oracle Public Network traffic on "VLAN 134" through FI – A and Oracle Private Interconnect Network traffic on "VLAN 10" through FI – B under normal operating conditions before the failover tests.

The screenshots below show a complete infrastructure details of MAC address and VLAN information for Cisco UCS FI - A and FI - B Switches before failover test. Log into FI - A and type “connect nxos” then type “show mac address-table” to see all the VLAN connection on the switch:

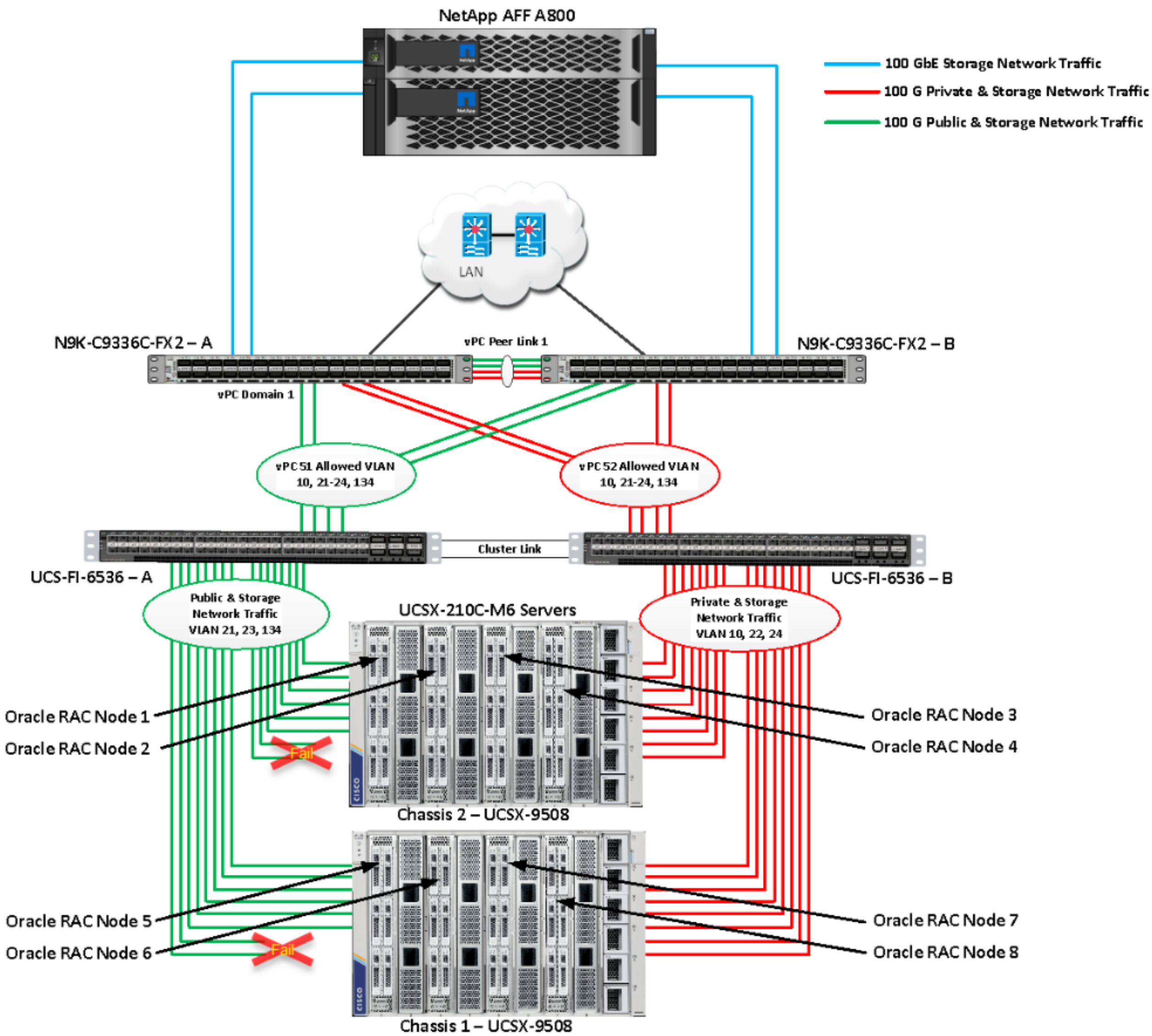
```
[ORA21C-FI-A(nx-os)# show mac address-table | grep static
*   21     0025.b5a1.a104   static   -        F     F    Veth806
*   21     0025.b5a1.a107   static   -        F     F    Veth818
*   21     0025.b5a1.a109   static   -        F     F    Veth830
*   21     0025.b5a1.a10b   static   -        F     F    Veth842
*   21     0025.b5a1.a10d   static   -        F     F    Veth854
*   21     0025.b5a1.a10f   static   -        F     F    Veth866
*   21     0025.b5a1.a111   static   -        F     F    Veth878
*   21     0025.b5a1.a113   static   -        F     F    Veth890
*   23     0025.b5a1.a105   static   -        F     F    Veth802
*   23     0025.b5a1.a106   static   -        F     F    Veth814
*   23     0025.b5a1.a108   static   -        F     F    Veth826
*   23     0025.b5a1.a10a   static   -        F     F    Veth838
*   23     0025.b5a1.a10c   static   -        F     F    Veth850
*   23     0025.b5a1.a10e   static   -        F     F    Veth862
*   23     0025.b5a1.a110   static   -        F     F    Veth874
*   23     0025.b5a1.a112   static   -        F     F    Veth886
*  134     0025.b513.4a24   static   -        F     F    Veth810
*  134     0025.b513.4a25   static   -        F     F    Veth822
*  134     0025.b513.4a26   static   -        F     F    Veth834
*  134     0025.b513.4a27   static   -        F     F    Veth846
*  134     0025.b513.4a28   static   -        F     F    Veth858
*  134     0025.b513.4a29   static   -        F     F    Veth870
*  134     0025.b513.4a2a   static   -        F     F    Veth882
[*  134     0025.b513.4a2b   static   -        F     F    Veth894
```

Similarly, log into FI - B and type “connect nxos” then type “show mac address-table” to see all the VLAN connection on the switch as follows:

```
[ORA21C-FI-B(nx-os)# show mac address-table | grep static
*   10     0025.b513.4b02   static   -        F      F     Veth804
*   10     0025.b513.4b03   static   -        F      F     Veth816
*   10     0025.b513.4b04   static   -        F      F     Veth828
*   10     0025.b513.4b05   static   -        F      F     Veth840
*   10     0025.b513.4b06   static   -        F      F     Veth852
*   10     0025.b513.4b07   static   -        F      F     Veth864
*   10     0025.b513.4b08   static   -        F      F     Veth876
*   10     0025.b513.4b09   static   -        F      F     Veth888
*   22     0025.b5b1.b104   static   -        F      F     Veth808
*   22     0025.b5b1.b107   static   -        F      F     Veth820
*   22     0025.b5b1.b108   static   -        F      F     Veth832
*   22     0025.b5b1.b10b   static   -        F      F     Veth844
*   22     0025.b5b1.b10d   static   -        F      F     Veth856
*   22     0025.b5b1.b10e   static   -        F      F     Veth868
*   22     0025.b5b1.b111   static   -        F      F     Veth880
*   22     0025.b5b1.b112   static   -        F      F     Veth892
*   24     0025.b5b1.b105   static   -        F      F     Veth800
*   24     0025.b5b1.b106   static   -        F      F     Veth812
*   24     0025.b5b1.b109   static   -        F      F     Veth824
*   24     0025.b5b1.b10a   static   -        F      F     Veth836
*   24     0025.b5b1.b10c   static   -        F      F     Veth848
*   24     0025.b5b1.b10f   static   -        F      F     Veth860
*   24     0025.b5b1.b110   static   -        F      F     Veth872
*   24     0025.b5b1.b113   static   -        F      F     Veth884
```

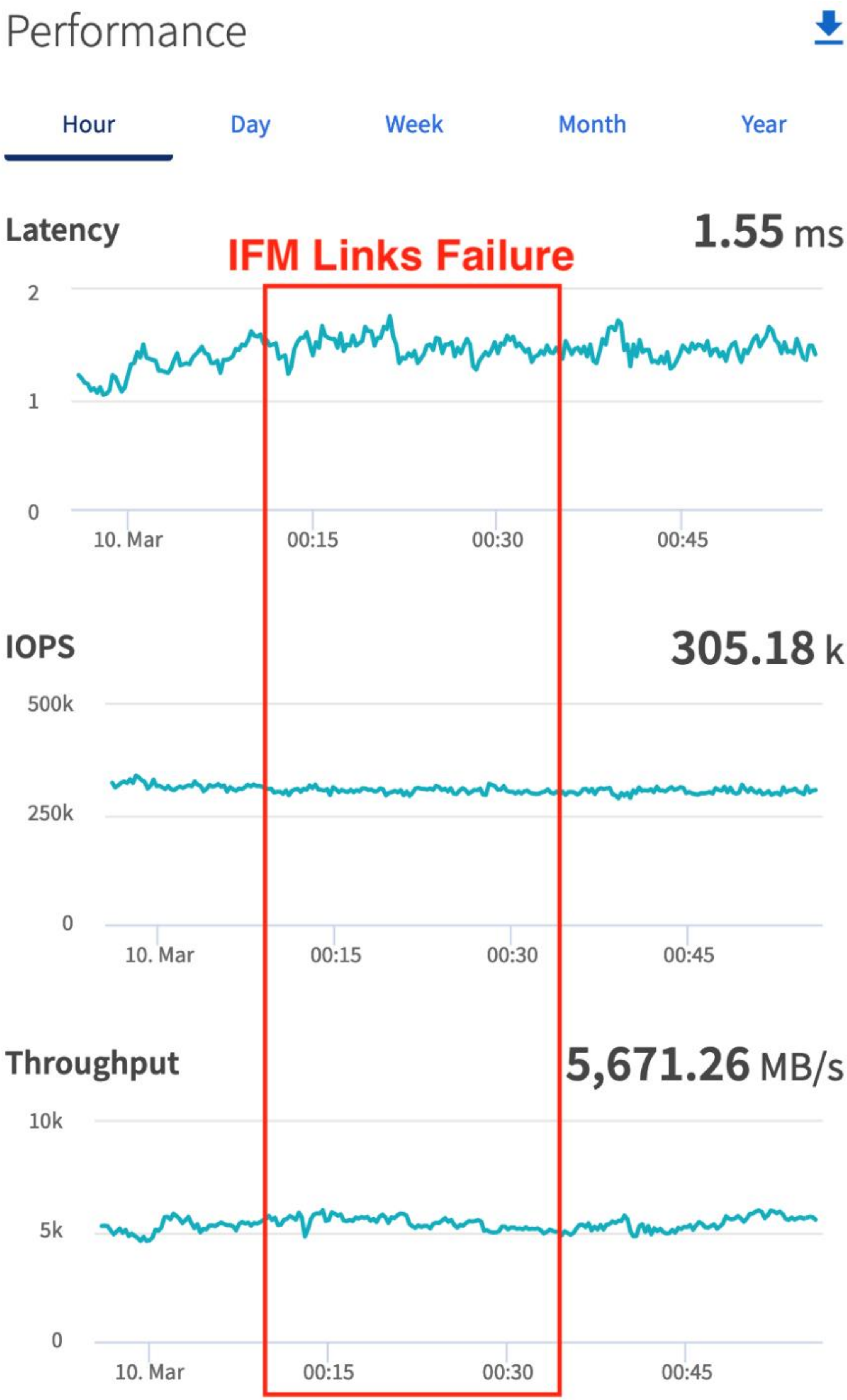
Test 1 – Cisco UCS-X Chassis IFM Links Failure

We conducted the chassis IFM Links failure test on Cisco UCS Chassis 1 and Chassis 2 by disconnecting one of the server port link cables from both chassis as shown below:



Unplug two server port cables from Chassis 1 and Chassis 2 each and check all the VLAN traffic information on both Cisco UCS FIs, Database and NetApp Storage. The screenshot below shows the database workload performance from the storage array when multiple chassis links failed.

We noticed no disruption in any of the network traffic and the database kept running under normal working conditions even after multiple IFM links failed from both the Chassis because of the Cisco UCS Port-Channel Feature. We kept the chassis links down for at least an hour and then reconnected those failed links and observed no disruption in network traffic and database operation.

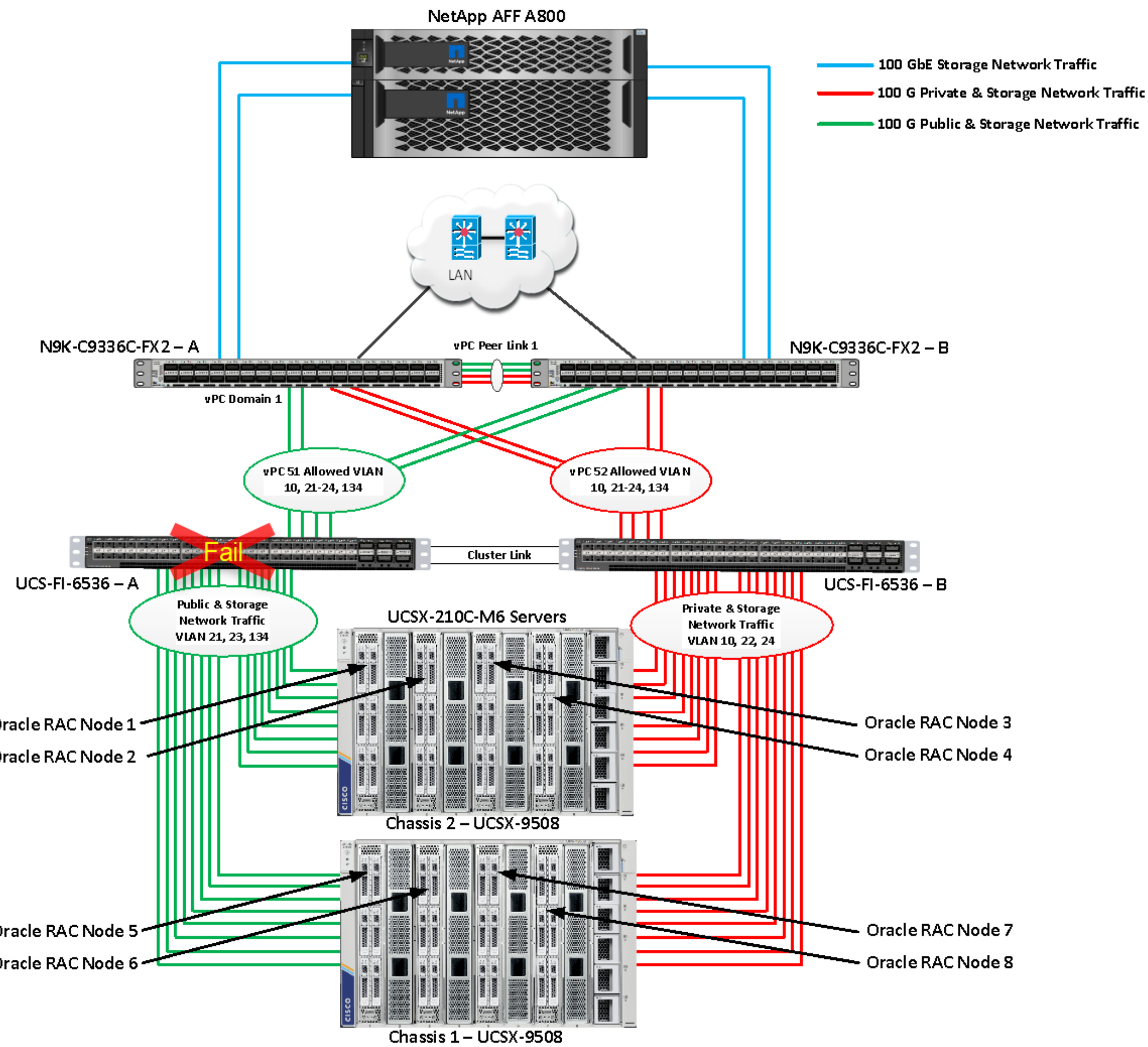


Performance
Hour
Day
Week
Month
Year
Latency
1.55 ms
IFM Links Failure
2
1
0
10. Mar
00:15
00:30
00:45
IOPS
305.18 k
500k
250k
0
10. Mar
00:15
00:30
00:45
Throughput
5,671.26 MB/s
10k
5k
0
10. Mar
00:15
00:30
00:45

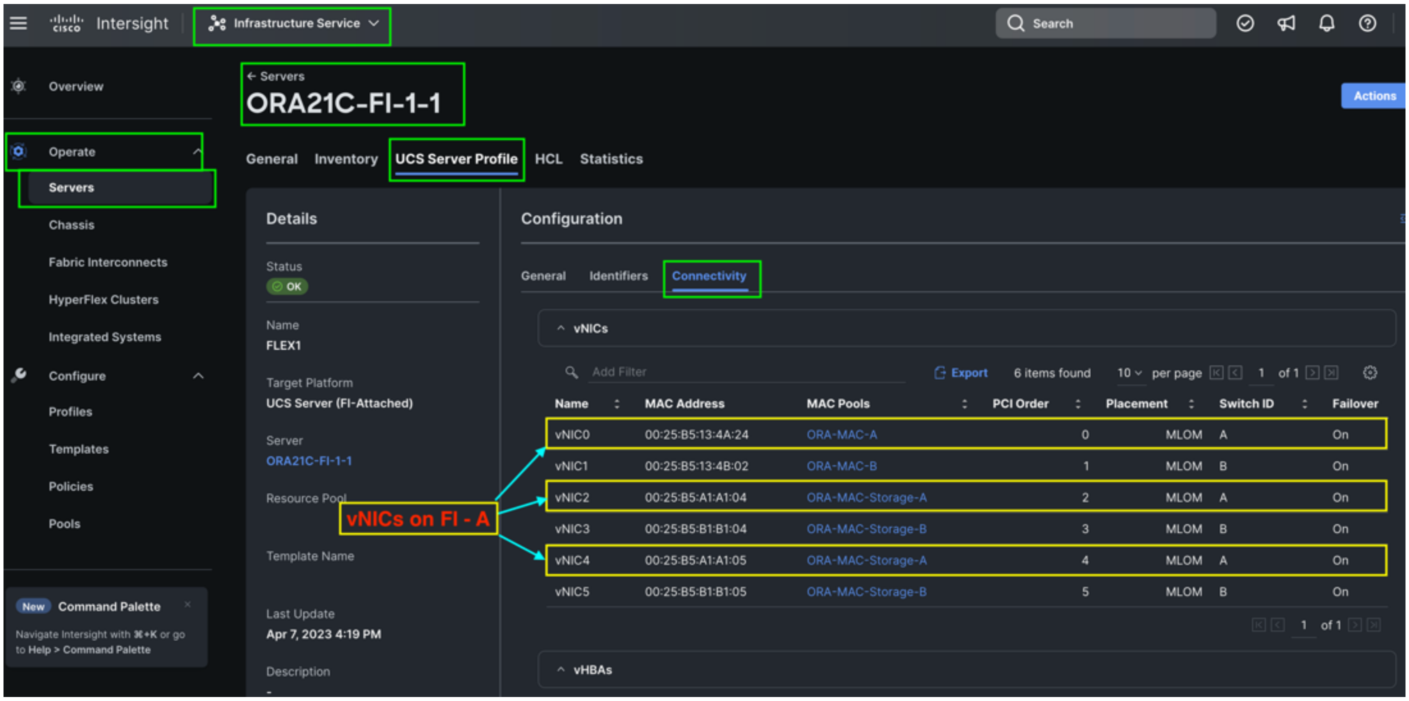
Test 2 – One FI Failure

We conducted a hardware failure test on FI-A by disconnecting the power cable to the fabric interconnect switch.

The figure below illustrates how during FI-A switch failure, the respective nodes (flex1, flex2, flex3 and flex4) on chassis 1 and nodes (flex5, flex6, flex7 and flex8) on chassis 2 will re-route the VLAN (134 – Management Network, 21 and 23 – Storage Network) traffic through the healthy Fabric Interconnect Switch FI-B.



As shown below, log into Cisco Intersight and go to Infrastructure Service > Operate > Servers > Server 1 (ORA21C-FI-1-1) > UCS Server Profile > Connectivity > and check all vNIC which were on FI-A as shown below. We will login into FI-B and check that those vNICs on FI-A and network traffic failed over to other FI while FI-A went down.

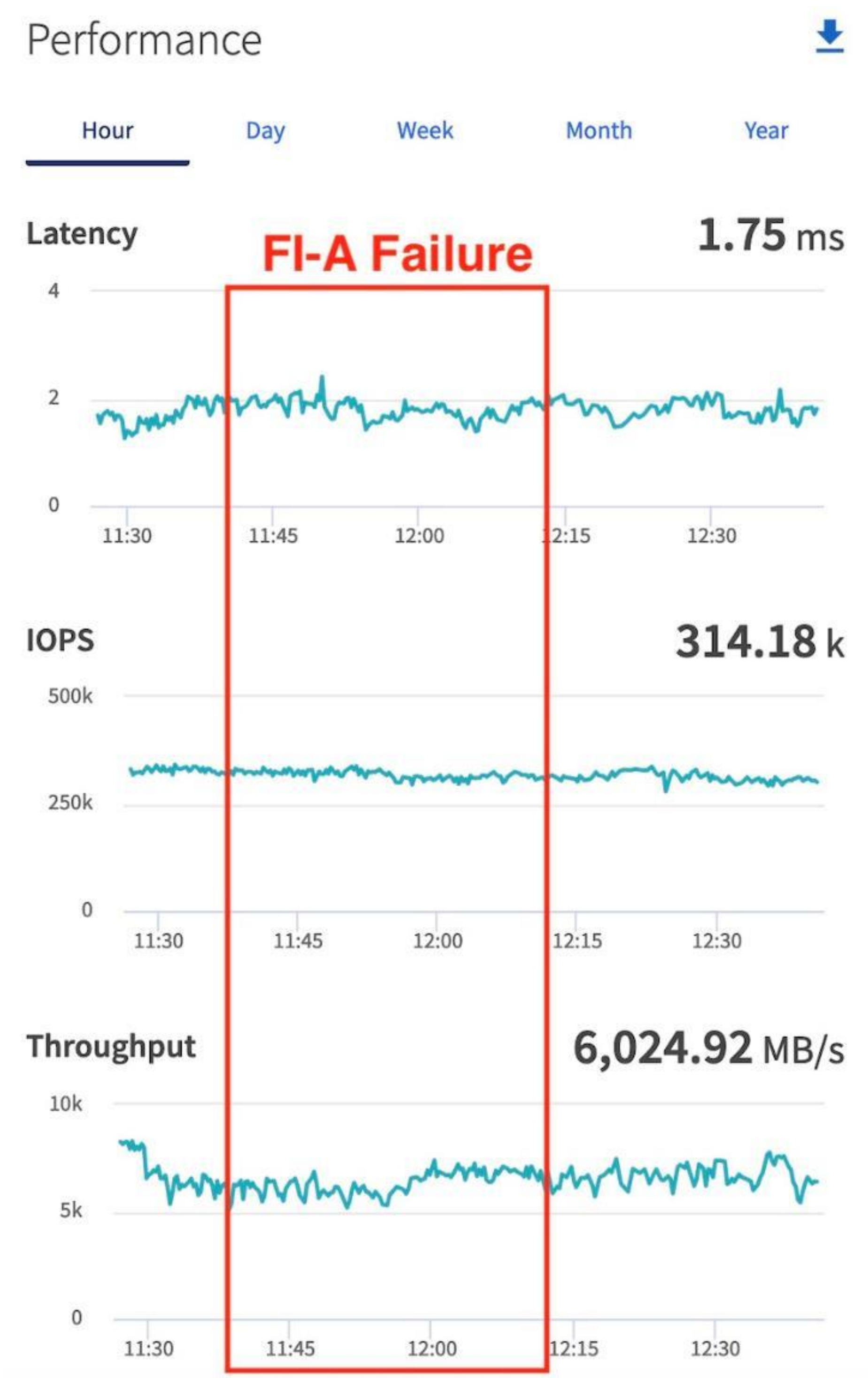


Log into FI – B and type "connect nxos" then type "show mac address-table" to see all VLAN connection on FI – B.

In the screenshot below, we noticed when the FI-A failed, all the MAC addresses of the redundant vNICs kept their VLANs network traffic going through FI-B. We observed that total 24 vNICs (each server having 3 vNIC for VLAN 134, 21 and 23) were failed over to other FI and database network traffic kept running under normal conditions even after failure of one of the FI.

```
ORA21C-FI-B(nx-os)# show mac address-table | grep static
*   10      0025.b513.4b02   static   -        F       F      Veth804
*   10      0025.b513.4b03   static   -        F       F      Veth816
*   10      0025.b513.4b04   static   -        F       F      Veth828
*   10      0025.b513.4b05   static   -        F       F      Veth840
*   10      0025.b513.4b06   static   -        F       F      Veth852
*   10      0025.b513.4b07   static   -        F       F      Veth864
*   10      0025.b513.4b08   static   -        F       F      Veth876
*   10      0025.b513.4b09   static   -        F       F      Veth888
*   21      0025.b5a1.a104   static   -        F       F      Veth807
*   21      0025.b5a1.a107   static   -        F       F      Veth819
*   21      0025.b5a1.a109   static   -        F       F      Veth831
*   21      0025.b5a1.a10b   static   -        F       F      Veth843
*   21      0025.b5a1.a10d   static   -        F       F      Veth855
*   21      0025.b5a1.a10f   static   -        F       F      Veth867
*   21      0025.b5a1.a111   static   -        F       F      Veth879
*   21      0025.b5a1.a113   static   -        F       F      Veth891
*   22      0025.b5b1.b104   static   -        F       F      Veth808
*   22      0025.b5b1.b107   static   -        F       F      Veth820
*   22      0025.b5b1.b108   static   -        F       F      Veth832
*   22      0025.b5b1.b10b   static   -        F       F      Veth844
*   22      0025.b5b1.b10d   static   -        F       F      Veth856
*   22      0025.b5b1.b10e   static   -        F       F      Veth868
*   22      0025.b5b1.b111   static   -        F       F      Veth880
*   22      0025.b5b1.b112   static   -        F       F      Veth892
*   23      0025.b5a1.a105   static   -        F       F      Veth803
*   23      0025.b5a1.a106   static   -        F       F      Veth815
*   23      0025.b5a1.a108   static   -        F       F      Veth827
*   23      0025.b5a1.a10a   static   -        F       F      Veth839
*   23      0025.b5a1.a10c   static   -        F       F      Veth851
*   23      0025.b5a1.a10e   static   -        F       F      Veth863
*   23      0025.b5a1.a110   static   -        F       F      Veth875
*   23      0025.b5a1.a112   static   -        F       F      Veth887
*   24      0025.b5b1.b105   static   -        F       F      Veth800
*   24      0025.b5b1.b106   static   -        F       F      Veth812
*   24      0025.b5b1.b109   static   -        F       F      Veth824
*   24      0025.b5b1.b10a   static   -        F       F      Veth836
*   24      0025.b5b1.b10c   static   -        F       F      Veth848
*   24      0025.b5b1.b10f   static   -        F       F      Veth860
*   24      0025.b5b1.b110   static   -        F       F      Veth872
*   24      0025.b5b1.b113   static   -        F       F      Veth884
*  134      0025.b513.4a24   static   -        F       F      Veth811
*  134      0025.b513.4a25   static   -        F       F      Veth823
*  134      0025.b513.4a26   static   -        F       F      Veth835
*  134      0025.b513.4a27   static   -        F       F      Veth847
*  134      0025.b513.4a28   static   -        F       F      Veth859
*  134      0025.b513.4a29   static   -        F       F      Veth871
*  134      0025.b513.4a2a   static   -        F       F      Veth883
*  134      0025.b513.4a2b   static   -        F       F      Veth895
[* 4043     4006.d5b3.0458   static   -        F       F      Po1391
[* 4043     b8a3.7701.eea8   static   -        F       F      Po1382
[* 4043     b8a3.7701.f020   static   -        F       F      Po1369
[* 4043     b8a3.7701.f058   static   -        F       F      Po1383
[* 4043     b8a3.7701.f120   static   -        F       F      Po1384
[* 4043     c44d.8471.5e38   static   -        F       F      Po1366
* 4043      f87a.41ec.cab0   static   -        F       F      Po1367
[* 4043     f87a.41ec.cab8   static   -        F       F      Po1363
[G    -     0008.310f.6dab   static   -        F       F      sup-eth1(R)
```

The screenshot below shows the NetApp Storage Array performance of the mixed workloads on all the databases while one of the FI failed.



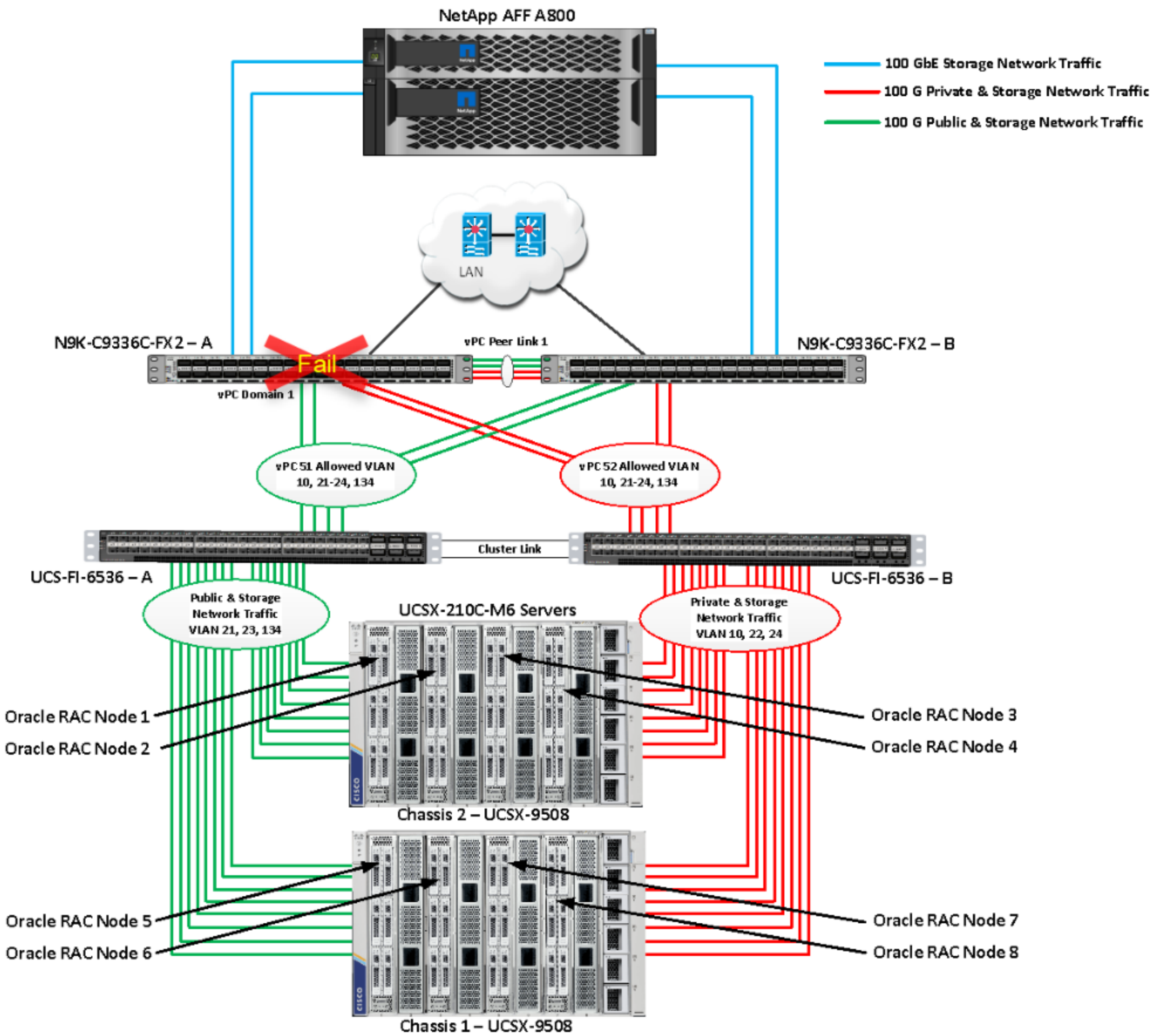
We also monitored and captured databases and its performance during this FI failure test through database alert log files and AWR reports. When we disconnected the power from FI – A, it caused a momentary impact on performance on the overall total IOPS, latency on OLTP as well as throughput on the DSS database for a few seconds but noticed that we did not see any interruption in any Private Server to Server Oracle RAC Interconnect Network, Management Public Network and Storage network traffic on IO Service Requests to the storage. We observed the database workload kept running under normal conditions throughout duration of FI failure.

We noticed this behavior because each server node has vNICs configured as failover enabled on LAN connectivity policy so that during FI failure, vNIC can failover to another active FI. Therefore, in case of any one FI failure, all the vNIC and its MAC address of the servers would route their traffic through another FI.

After plugging back power cable to FI-A Switch, the respective nodes (flex1, flex2, flex3 and flex4) on chassis 1 and nodes (flex5, flex6, flex7 and flex8) on chassis 2 will route back the MAC addresses and its VLAN public network and storage network traffic to FI-A.

Test 3 – Cisco Nexus Switch Failure

We conducted a hardware failure test on Cisco Nexus Switch-A by disconnecting the power cable to the Cisco Nexus Switch and checking the storage network traffic on Cisco Nexus Switch-B and the overall system as shown below:



The screenshot below shows the vpc summary on Cisco Nexus Switch B while Cisco Nexus A was down.

```
ORA21C-N9K-B# show vpc brief
Legend:
                (*) - local vPC is down, forwarding via vPC peer-link

vPC domain id                     : 1
Peer status                       : peer link is down
vPC keep-alive status             : Suspended (Destination IP not reachable)
Configuration consistency status  : success
Per-vlan consistency status       : success
Type-2 consistency status         : success
vPC role                          : primary
Number of vPCs configured         : 4
Peer Gateway                      : Disabled
Dual-active excluded VLANs        : -
Graceful Consistency Check        : Enabled
Auto-recovery status              : Disabled
Delay-restore status              : Timer is off.(timeout = 30s)
Delay-restore SVI status          : Timer is off.(timeout = 10s)
Operational Layer3 Peer-router    : Disabled
Virtual-peerlink mode             : Disabled

vPC Peer-link status
---------------------------------------------------------------------
id     Port   Status Active vlans
--     ----   ------ -------------------------------------------------
1      Po1    down   -

vPC status
----------------------------------------------------------------------------
Id     Port          Status Consistency Reason                Active vlans
--     ------------  ------ ----------- ------                ---------------
13     Po13          up     success     success               21-24

14     Po14          up     success     success               21-24

51     Po51          up     success     success               1,10,21-24,134

52     Po52          up     success     success               1,10,21-24,134
```

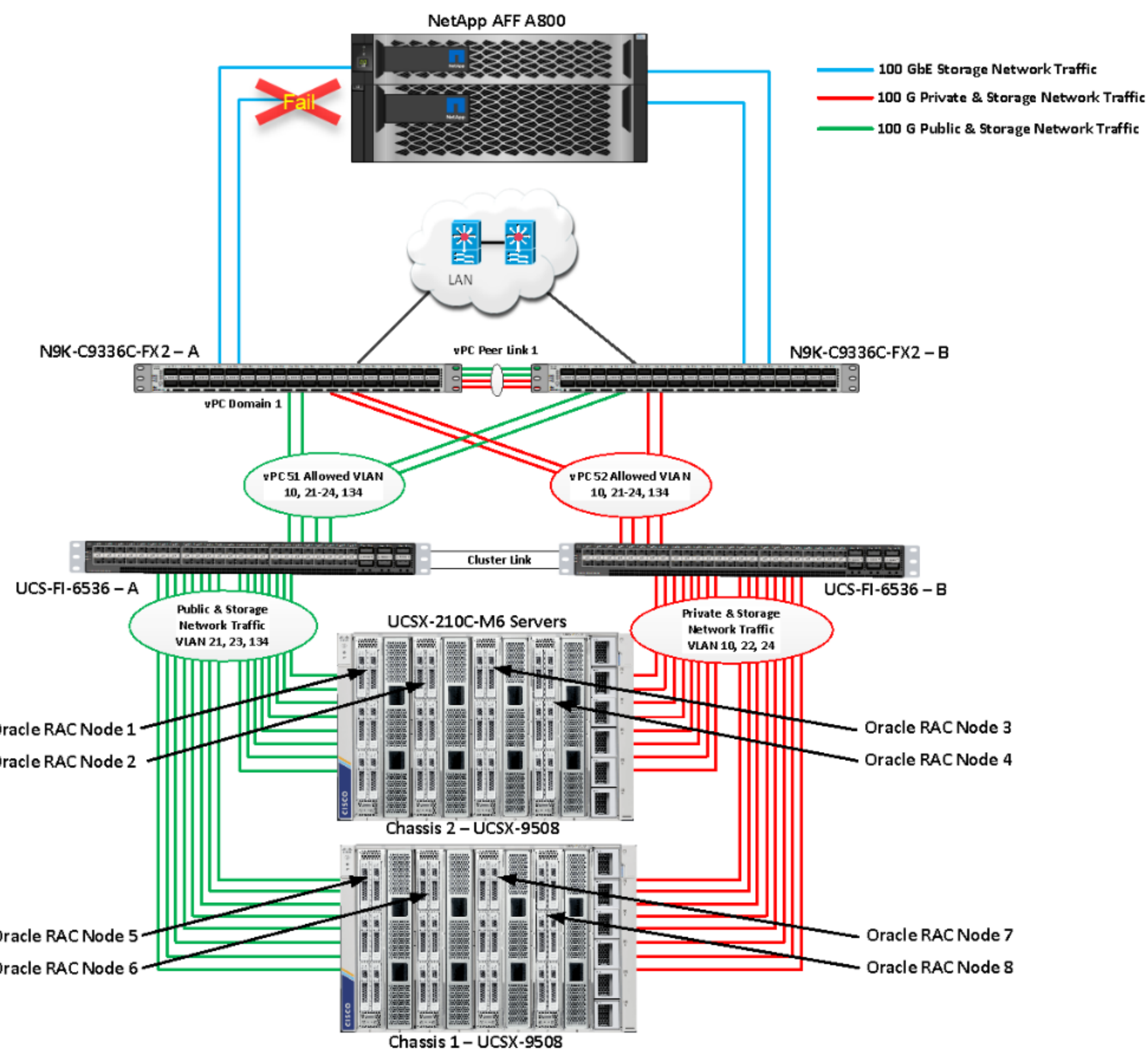
When we disconnected the power from Cisco Nexus-A Switch, it caused a very momentary impact on performance of the overall total IOPS, latency on OLTP as well as throughput of the DSS database for a few seconds but noticed that we did not see any interruption in the overall Private Server to Server Oracle RAC Interconnect Network, Management Public Network, and storage network traffic on I/O Service Requests to the storage as shown below:

Performance
Hour
Day
Week
Month
Year
Latency
2.29 ms
Nexus A Failure
4
2
0
11:00
11:15
11:30
11:45
IOPS
309.09 k
500k
250k
0
11:00
11:15
11:30
11:45
Throughput
7,229.18 MB/s
10k
5k
0
11:00
11:15
11:30
11:45

Like FI failure tests, we observed no impact overall on all three databases performance and all the VLAN network traffic were going through other active Cisco Nexus switch B and databases workload kept running under normal conditions throughout the duration of Nexus failure. After plugging back the power cable back into Cisco Nexus-A Switch, Nexus Switch returns to normal operating state and database performance will resume at peak performance.

Test 4 – Storage Controller Links Failure

We performed storage controller link failure test by disconnecting one of the 100G links from the NetApp Array from one of the storage controller as shown below:



As explained previously in the storage configuration section, we created one interface group (a0a) across both the controller by adding all the physical storage ports into the group. This logical interface group provides increased resiliency, increased availability, and load sharing. Like Chassis link failure tests, we noticed no disruption in any of the network and storage traffic and the database kept running under normal working conditions even after storage link failed.

After plugging back into the storage to Cisco Nexus link into storage controller, the Cisco Nexus Switch and Storage array links comes back online, and database performance resumed to peak performance.

Test 5 – RAC Server Node Failure

In this test, we started the SwingBench workload test run on all of the RAC nodes, and then during run, we powered down one node from the RAC cluster to check the overall system performance. We didn't observe any performance impact on overall database IOPS, latency and throughput after losing one node from the system.

We completed an additional failure scenario and validated that there is no single point of failure in this reference design.

Summary

The Cisco Unified Computing System (Cisco UCS) is a next-generation data center platform that unites computing, network, storage access, and virtualization into a single cohesive system. Cisco UCS is an ideal platform for the architecture of mission critical database workloads such as Oracle RAC. The combination of Cisco UCS, NetApp and Oracle Real Application Cluster Database architecture can accelerate your IT transformation by enabling faster deployments, greater flexibility of choice, efficiency, high availability, and lower risk. The FlexPod Data-center solution is a validated approach for deploying Cisco and NetApp technologies and products to build shared private and public cloud infrastructure.

If you're interested in understanding the FlexPod design and deployment details, including the configuration of various elements of design and associated best practices, refer to Cisco Validated Designs for FlexPod, here: https://www.cisco.com/c/en/us/solutions/design-zone/data-center-design-guides/flexpod-design-guides.html .

The FlexPod Datacenter solution with Cisco UCS X-Series and NetApp AFF Storage using NetApp ONTAP 9.12.1 offers the following key customer benefits:

- Simplified cloud-based management of solution components.
- Hybrid-cloud-ready, policy-driven modular design.
- Highly available and scalable platform with flexible architecture that supports various deployment models.
- Cooperative support model and Cisco Solution Support.
- Easy to deploy, consume, and manage architecture, which saves time and resources required to research, procure, and integrate off-the-shelf components.
- Support for component monitoring, solution automation and orchestration, and workload optimization.

About the Authors

Hardikkumar Vyas, Technical Marketing Engineer, CSPG UCS Product Management and Data Center Solutions Engineering Group, Cisco Systems, Inc.

Hardikkumar Vyas is a Solution Architect in Cisco System's Cloud and Compute Engineering Group for configuring, implementing, and validating infrastructure best practices for highly available Oracle RAC databases solutions on Cisco UCS Servers, Cisco Nexus Products, and various Storage Technologies. Hardikkumar Vyas holds a master's degree in electrical engineering and has over 10 years of experience working with Oracle RAC Databases and associated applications. Hardikkumar Vyas's focus is developing database solutions on different platforms, perform benchmarks, prepare reference architectures, and write technical documents for Oracle RAC Databases on Cisco UCS Platforms.

Tushar Patel, Distinguished Technical Marketing Engineer , CSPG UCS Product Management and Data Center Solutions Engineering Group, Cisco Systems, Inc.

Tushar Patel is a Distinguished Technical Marketing Engineer in Cisco System's CSPG UCS Product Management and Data Center Solutions Engineering Group and a specialist in Flash Storage technologies and Oracle RAC RDBMS. Tushar has over 27 years of experience in Flash Storage architecture, Database architecture, design, and performance. Tushar also has strong background in Intel X86 architecture, hyper converged systems, Storage technologies and Virtualization. He has worked with large number of enterprise customers, to evaluate, and deploy mission critical database solutions. Tushar has presented to both internal and external audiences at various conferences and customer events.

Acknowledgements

For their support and contribution to the design, validation, and creation of this Cisco Validated Design, the authors would like to thank:

- Bobby Oommen, Sr. Manager FlexPod Solutions, NetApp

Appendix

This appendix is organized into the following:

Compute

Cisco Intersight: https://www.intersight.com

Cisco Intersight Managed Mode: https://www.cisco.com/c/en/us/td/docs/unified_computing/Intersight/b_Intersight_Managed_Mode_Configuration_Guide.html

Cisco Unified Computing System: http://www.cisco.com/en/US/products/ps10265/index.html

Cisco UCS 6536 Fabric Interconnects: https://www.cisco.com/c/en/us/products/collateral/servers-unified-computing/ucs6536-fabric-interconnect-ds.html

Network

Cisco Nexus 9000 Series Switches: http://www.cisco.com/c/en/us/products/switches/nexus-9000-series-switches/index.html

Cisco MDS 9132T Switches: https://www.cisco.com/c/en/us/products/collateral/storage-networking/mds-9100-series-multilayer-fabric-switches/datasheet-c78-739613.html

Storage

NetApp ONTAP: https://docs.netapp.com/ontap-9/index.jsp

NetApp Active IQ Unified Manager: https://community.netapp.com/t5/Tech-ONTAP-Blogs/Introducing-NetApp-Active-IQ-Unified-Manager-9-11/ba-p/435519

ONTAP Storage Connector for Cisco Intersight:
https://www.netapp.com/pdf.html?item=/media/25001-tr-4883.pdf

ONTAP tools for VMware vSphere: https://docs.netapp.com/us-en/ontap-tools-vmware-vsphere/index.html

NetApp SnapCenter: https://docs.netapp.com/us-en/snapcenter/index.html

Interoperability Matrix

Cisco UCS Hardware Compatibility Matrix: https://ucshcltool.cloudapps.cisco.com/public/

VMware and Cisco Unified Computing System: http://www.vmware.com/resources/compatibility

NetApp Interoperability Matrix Tool: http://support.netapp.com/matrix/

Cisco Nexus A Configuration

```
ORA21C-N9K-A# show running-config
!Command: show running-config
!Running configuration last done at: Mon Apr 10 22:04:13 2023
!Time: Fri May 2 07:51:54 2023
version 9.2(3) Bios:version 05.33
switchname ORA21C-N9K-A
policy-map type network-qos jumbo
  class type network-qos class-default
    mtu 9216
vdc ORA21C-N9K-A id 1
  limit-resource vlan minimum 16 maximum 4094
  limit-resource vrf minimum 2 maximum 4096
  limit-resource port-channel minimum 0 maximum 511
  limit-resource u4route-mem minimum 248 maximum 248
  limit-resource u6route-mem minimum 96 maximum 96
  limit-resource m4route-mem minimum 58 maximum 58
  limit-resource m6route-mem minimum 8 maximum 8
cfs eth distribute
feature interface-vlan
feature hsrp
feature lacp
feature vpc
feature lldp
no password strength-check
username admin password 5 $5$QyO36Ye4$xKHjJmPA/zgfNSpblJPcbu7GgNA0GweKS/xOzUjCcK4  role
network-admin
ip domain-lookup
```

```
system default switchport
system qos
  service-policy type network-qos jumbo
copp profile strict
snmp-server user admin network-admin auth md5 0xab8f5da7966d49de676779a717fb6b92 priv
0xab8f5da7966d49de676779a717fb6b92 localizedkey
rmon event 1 description FATAL(1) owner PMON@FATAL
rmon event 2 description CRITICAL(2) owner PMON@CRITICAL
rmon event 3 description ERROR(3) owner PMON@ERROR
rmon event 4 description WARNING(4) owner PMON@WARNING
rmon event 5 description INFORMATION(5) owner PMON@INFO
ntp server 72.163.32.44 use-vrf default
vlan 1,10,21-24,134
vlan 10
  name Oracle_RAC_Private_Traffic
vlan 21
  name Storage_Traffic_A1
vlan 22
  name Storage_Traffic_B1
vlan 23
  name Storage_Traffic_A2
vlan 24
  name Storage_Traffic_B2
vlan 134
  name Oracle_RAC_Public_Traffic
spanning-tree port type edge bpduguard default
spanning-tree port type network default
vrf context management
  ip route 0.0.0.0/0 10.29.134.1
vpc domain 1
  peer-keepalive destination 10.29.134.44 source 10.29.134.43
interface Vlan1
interface Vlan134
  no shutdown
interface port-channel1
  description VPC peer-link
  switchport mode trunk
  switchport trunk allowed vlan 1,10,21-24,134
  spanning-tree port type network
  vpc peer-link
```

```
interface port-channel13
  description PC-NetApp-A
  switchport mode trunk
  switchport trunk allowed vlan 21-24
  spanning-tree port type edge trunk
  mtu 9216
  vpc 13
interface port-channel14
  description PC-NetApp-B
  switchport mode trunk
  switchport trunk allowed vlan 21-24
  spanning-tree port type edge trunk
  mtu 9216
  vpc 14
interface port-channel51
  description connect to ORA21C-FI-A
  switchport mode trunk
  switchport trunk allowed vlan 1,10,21-24,134
  spanning-tree port type edge trunk
  mtu 9216
  vpc 51
interface port-channel52
  description connect to ORA21C-FI-B
  switchport mode trunk
  switchport trunk allowed vlan 1,10,21-24,134
  spanning-tree port type edge trunk
  mtu 9216
  vpc 52
interface Ethernet1/1
  description Peer link connected to ORA21C-N9K-B-Eth1/1
  switchport mode trunk
  switchport trunk allowed vlan 1,10,21-24,134
  channel-group 1 mode active
interface Ethernet1/2
  description Peer link connected to ORA21C-N9K-B-Eth1/2
  switchport mode trunk
  switchport trunk allowed vlan 1,10,21-24,134
  channel-group 1 mode active
interface Ethernet1/3
```

```
  description Peer link connected to ORA21C-N9K-B-Eth1/3
  switchport mode trunk
  switchport trunk allowed vlan 1,10,21-24,134
  channel-group 1 mode active
interface Ethernet1/4
  description Peer link connected to ORA21C-N9K-B-Eth1/4
  switchport mode trunk
  switchport trunk allowed vlan 1,10,21-24,134
  channel-group 1 mode active
interface Ethernet1/5
interface Ethernet1/6
interface Ethernet1/7
interface Ethernet1/8
interface Ethernet1/9
  description Fabric-Interconnect-A-27
  switchport mode trunk
  switchport trunk allowed vlan 1,10,21-24,134
  spanning-tree port type edge trunk
  mtu 9216
  channel-group 51 mode active
interface Ethernet1/10
  description Fabric-Interconnect-A-28
  switchport mode trunk
  switchport trunk allowed vlan 1,10,21-24,134
  spanning-tree port type edge trunk
  mtu 9216
  channel-group 51 mode active
interface Ethernet1/11
  description Fabric-Interconnect-B-27
  switchport mode trunk
  switchport trunk allowed vlan 1,10,21-24,134
  spanning-tree port type edge trunk
  mtu 9216
  channel-group 52 mode active
interface Ethernet1/12
  description Fabric-Interconnect-B-28
  switchport mode trunk
  switchport trunk allowed vlan 1,10,21-24,134
  spanning-tree port type edge trunk
```

```
  mtu 9216
  channel-group 52 mode active
interface Ethernet1/13
interface Ethernet1/14
interface Ethernet1/15
interface Ethernet1/16
interface Ethernet1/17
  description FlexPod-A800-CT1:e5a
  switchport mode trunk
  switchport trunk allowed vlan 21-24
  mtu 9216
  channel-group 13 mode active
interface Ethernet1/18
  description FlexPod-A800-CT2:e5a
  switchport mode trunk
  switchport trunk allowed vlan 21-24
  mtu 9216
  channel-group 14 mode active
interface Ethernet1/19
interface Ethernet1/20
interface Ethernet1/21
interface Ethernet1/22
interface Ethernet1/23
interface Ethernet1/24
interface Ethernet1/25
interface Ethernet1/26
interface Ethernet1/27
interface Ethernet1/28
interface Ethernet1/29
  description To-Management-Uplink-Switch
  switchport access vlan 134
  speed 1000
interface Ethernet1/30
interface Ethernet1/31
interface Ethernet1/32
interface Ethernet1/33
interface Ethernet1/34
interface Ethernet1/35
interface Ethernet1/36
```

```
interface mgmt0
  vrf member management
  ip address 10.29.134.43/24
line console
line vty
boot nxos bootflash:/nxos.9.2.3.bin
no system default switchport shutdown
```

Configuration of “sysctl.conf”

```
[root@flex1 ~]# cat /etc/sysctl.conf
# sysctl settings are defined through files in
# /usr/lib/sysctl.d/, /run/sysctl.d/, and /etc/sysctl.d/.
# Vendors settings live in /usr/lib/sysctl.d/.
# To override a whole file, create a new file with the same in
# /etc/sysctl.d/ and put new settings there. To override
# only specific settings, add a file with a lexically later
# name in /etc/sysctl.d/ and put new settings there.
# For more information, see sysctl.conf(5) and sysctl.d(5).
vm.nr_hugepages=120000
net.core.netdev_max_backlog = 300000
net.ipv4.tcp_moderate_rcvbuf = 1
net.ipv4.tcp_no_metrics_save = 1
net.ipv4.tcp_rmem = 4096 87380 134217728
net.ipv4.tcp_sack = 0
net.ipv4.tcp_syncookies = 0
net.ipv4.tcp_timestamps = 0
net.ipv4.tcp_window_scaling = 1
net.ipv4.tcp_wmem = 4096 65536 134217728
sunrpc.tcp_slot_table_entries = 128
# oracle-database-preinstall-21c setting for fs.file-max is 6815744
fs.file-max = 6815744
# oracle-database-preinstall-21c setting for kernel.sem is '250 32000 100 128'
kernel.sem = 250 32000 100 128
# oracle-database-preinstall-21c setting for kernel.shmmni is 4096
kernel.shmmni = 4096
# oracle-database-preinstall-21c setting for kernel.shmall is 1073741824 on x86_64
kernel.shmall = 1073741824
# oracle-database-preinstall-21c setting for kernel.shmmax is 4398046511104 on x86_64
kernel.shmmax = 4398046511104
# oracle-database-preinstall-21c setting for kernel.panic_on_oops is 1 per Orabug 19212317
```

```
kernel.panic_on_oops = 1
# oracle-database-preinstall-21c setting for net.core.rmem_default is 262144
net.core.rmem_default = 134217728
# oracle-database-preinstall-21c setting for net.core.rmem_max is 4194304
net.core.rmem_max = 134217728
# oracle-database-preinstall-21c setting for net.core.wmem_default is 262144
net.core.wmem_default = 134217728
# oracle-database-preinstall-21c setting for net.core.wmem_max is 1048576
net.core.wmem_max = 134217728
# oracle-database-preinstall-21c setting for net.ipv4.conf.all.rp_filter is 2
net.ipv4.conf.all.rp_filter = 2
# oracle-database-preinstall-21c setting for net.ipv4.conf.default.rp_filter is 2
net.ipv4.conf.default.rp_filter = 2
# oracle-database-preinstall-21c setting for fs.aio-max-nr is 1048576
fs.aio-max-nr = 1048576
# oracle-database-preinstall-21c setting for net.ipv4.ip_local_port_range is 9000 65500
net.ipv4.ip_local_port_range = 9000 65500
```

Configuration of “oracle-database-preinstall-21c.conf”

```
[root@flex1 ~]# cat /etc/security/limits.d/oracle-database-preinstall-21c.conf
# oracle-database-preinstall-21c setting for nofile soft limit is 1024
oracle   soft   nofile    1024
# oracle-database-preinstall-21c setting for nofile hard limit is 65536
oracle   hard   nofile    65536
# oracle-database-preinstall-21c setting for nproc soft limit is 16384
# refer orabug15971421 for more info.
oracle   soft   nproc    16384
# oracle-database-preinstall-21c setting for nproc hard limit is 16384
oracle   hard   nproc    16384
# oracle-database-preinstall-21c setting for stack soft limit is 10240KB
oracle   soft   stack    10240
# oracle-database-preinstall-21c setting for stack hard limit is 32768KB
oracle   hard   stack    32768
# oracle-database-preinstall-21c setting for memlock hard limit is maximum of 128GB on x86_64 or
3GB on x86 OR 90 % of RAM
oracle   hard   memlock    474609060
# oracle-database-preinstall-21c setting for memlock soft limit is maximum of 128GB on x86_64 or
3GB on x86 OR 90% of RAM
oracle   soft   memlock    474609060
# oracle-database-preinstall-21c setting for data soft limit is 'unlimited'
```

```
oracle   soft   data    unlimited
# oracle-database-preinstall-21c setting for data hard limit is 'unlimited'
oracle   hard   data    unlimited
```

Configuration of "fstab"

```
[root@flex1 ~]# cat /etc/fstab
# /etc/fstab
# Created by anaconda on Fri Jan 13 19:58:12 2023
# Accessible filesystems, by reference, are maintained under '/dev/disk/'.
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info.
# After editing this file, run 'systemctl daemon-reload' to update systemd
# units generated from this file.
/dev/mapper/ol-root     /                       xfs     defaults        0 0
UUID=2300cce7-826b-48d8-9540-c9d4fc6c733e /boot                   xfs     defaults        0 0
UUID=7D1B-6D3C          /boot/efi               vfat    umask=0077,shortname=winnt 0 2
/dev/mapper/ol-swap     none                    swap    defaults        0 0
###10.10.21.41:/fiodata1   /fiodata1        nfs
rw,bg,hard,rsize=524288,wsize=524288,nfsvers=3,actimeo=0,nointr,timeo=600,tcp,nconnect=16
###10.10.22.41:/fiodata3   /fiodata3        nfs
rw,bg,hard,rsize=524288,wsize=524288,nfsvers=3,actimeo=0,nointr,timeo=600,tcp,nconnect=16
###10.10.21.41:/fiodata5   /fiodata5        nfs
rw,bg,hard,rsize=524288,wsize=524288,nfsvers=3,actimeo=0,nointr,timeo=600,tcp,nconnect=16
###10.10.22.41:/fiodata7   /fiodata7        nfs
rw,bg,hard,rsize=524288,wsize=524288,nfsvers=3,actimeo=0,nointr,timeo=600,tcp,nconnect=16
###10.10.23.41:/fiodata2   /fiodata2        nfs
rw,bg,hard,rsize=524288,wsize=524288,nfsvers=3,actimeo=0,nointr,timeo=600,tcp,nconnect=16
###10.10.24.41:/fiodata4   /fiodata4        nfs
rw,bg,hard,rsize=524288,wsize=524288,nfsvers=3,actimeo=0,nointr,timeo=600,tcp,nconnect=16
###10.10.23.41:/fiodata6   /fiodata6        nfs
rw,bg,hard,rsize=524288,wsize=524288,nfsvers=3,actimeo=0,nointr,timeo=600,tcp,nconnect=16
###10.10.24.41:/fiodata8   /fiodata8        nfs
rw,bg,hard,rsize=524288,wsize=524288,nfsvers=3,actimeo=0,nointr,timeo=600,tcp,nconnect=16
10.10.21.41:/ocrvote    /ocrvote        nfs
rw,bg,hard,rsize=32768,wsize=32768,nfsvers=3,actimeo=0,nointr,timeo=600,tcp
10.10.21.41:/slobdata1  /slobdata1      nfs
rw,bg,hard,rsize=32768,wsize=32768,nfsvers=3,actimeo=0,nointr,timeo=600,tcp
10.10.22.42:/slobdata2  /slobdata2      nfs
rw,bg,hard,rsize=32768,wsize=32768,nfsvers=3,actimeo=0,nointr,timeo=600,tcp
10.10.23.41:/slobdata3  /slobdata3      nfs
rw,bg,hard,rsize=32768,wsize=32768,nfsvers=3,actimeo=0,nointr,timeo=600,tcp
10.10.24.42:/slobdata4  /slobdata4      nfs
rw,bg,hard,rsize=32768,wsize=32768,nfsvers=3,actimeo=0,nointr,timeo=600,tcp
```

```
 10.10.21.41:/sloblog1   /sloblog1       nfs
rw,bg,hard,rsize=32768,wsize=32768,nfsvers=3,actimeo=0,nointr,timeo=600,tcp
 10.10.22.42:/sloblog2   /sloblog2       nfs
rw,bg,hard,rsize=32768,wsize=32768,nfsvers=3,actimeo=0,nointr,timeo=600,tcp
 10.10.21.41:/findata01   /findata01       nfs
rw,bg,hard,rsize=524288,wsize=524288,nfsvers=3,actimeo=0,nointr,timeo=600,tcp
 10.10.22.42:/findata02   /findata02       nfs
rw,bg,hard,rsize=524288,wsize=524288,nfsvers=3,actimeo=0,nointr,timeo=600,tcp
 10.10.23.41:/findata03   /findata03       nfs
rw,bg,hard,rsize=524288,wsize=524288,nfsvers=3,actimeo=0,nointr,timeo=600,tcp
 10.10.24.42:/findata04   /findata04       nfs
rw,bg,hard,rsize=524288,wsize=524288,nfsvers=3,actimeo=0,nointr,timeo=600,tcp
 10.10.21.41:/findata05   /findata05       nfs
rw,bg,hard,rsize=524288,wsize=524288,nfsvers=3,actimeo=0,nointr,timeo=600,tcp
 10.10.22.42:/findata06   /findata06       nfs
rw,bg,hard,rsize=524288,wsize=524288,nfsvers=3,actimeo=0,nointr,timeo=600,tcp
 10.10.23.41:/findata07   /findata07       nfs
rw,bg,hard,rsize=524288,wsize=524288,nfsvers=3,actimeo=0,nointr,timeo=600,tcp
 10.10.24.42:/findata08   /findata08       nfs
rw,bg,hard,rsize=524288,wsize=524288,nfsvers=3,actimeo=0,nointr,timeo=600,tcp
 10.10.21.41:/findata09   /findata09       nfs
rw,bg,hard,rsize=524288,wsize=524288,nfsvers=3,actimeo=0,nointr,timeo=600,tcp
 10.10.22.42:/findata10   /findata10       nfs
rw,bg,hard,rsize=524288,wsize=524288,nfsvers=3,actimeo=0,nointr,timeo=600,tcp
 10.10.23.41:/findata11   /findata11       nfs
rw,bg,hard,rsize=524288,wsize=524288,nfsvers=3,actimeo=0,nointr,timeo=600,tcp
 10.10.24.42:/findata12   /findata12       nfs
rw,bg,hard,rsize=524288,wsize=524288,nfsvers=3,actimeo=0,nointr,timeo=600,tcp
 10.10.21.41:/findata13   /findata13       nfs
rw,bg,hard,rsize=524288,wsize=524288,nfsvers=3,actimeo=0,nointr,timeo=600,tcp
 10.10.22.42:/findata14   /findata14       nfs
rw,bg,hard,rsize=524288,wsize=524288,nfsvers=3,actimeo=0,nointr,timeo=600,tcp
 10.10.23.41:/findata15   /findata15       nfs
rw,bg,hard,rsize=524288,wsize=524288,nfsvers=3,actimeo=0,nointr,timeo=600,tcp
 10.10.24.42:/findata16   /findata16       nfs
rw,bg,hard,rsize=524288,wsize=524288,nfsvers=3,actimeo=0,nointr,timeo=600,tcp
 10.10.21.41:/finlog01   /finlog01       nfs
rw,bg,hard,rsize=524288,wsize=524288,nfsvers=3,actimeo=0,nointr,timeo=600,tcp
 10.10.22.42:/finlog02   /finlog02       nfs
rw,bg,hard,rsize=524288,wsize=524288,nfsvers=3,actimeo=0,nointr,timeo=600,tcp
 10.10.23.41:/finlog03   /finlog03       nfs
rw,bg,hard,rsize=524288,wsize=524288,nfsvers=3,actimeo=0,nointr,timeo=600,tcp
 10.10.24.42:/finlog04   /finlog04       nfs
rw,bg,hard,rsize=524288,wsize=524288,nfsvers=3,actimeo=0,nointr,timeo=600,tcp
```

```
 10.10.21.41:/soedata01   /soedata01       nfs
rw,bg,hard,rsize=524288,wsize=524288,nfsvers=3,actimeo=0,nointr,timeo=600,tcp
 10.10.22.42:/soedata02   /soedata02       nfs
rw,bg,hard,rsize=524288,wsize=524288,nfsvers=3,actimeo=0,nointr,timeo=600,tcp
 10.10.23.41:/soedata03   /soedata03       nfs
rw,bg,hard,rsize=524288,wsize=524288,nfsvers=3,actimeo=0,nointr,timeo=600,tcp
 10.10.24.42:/soedata04   /soedata04       nfs
rw,bg,hard,rsize=524288,wsize=524288,nfsvers=3,actimeo=0,nointr,timeo=600,tcp
 10.10.21.41:/soedata05   /soedata05       nfs
rw,bg,hard,rsize=524288,wsize=524288,nfsvers=3,actimeo=0,nointr,timeo=600,tcp
 10.10.22.42:/soedata06   /soedata06       nfs
rw,bg,hard,rsize=524288,wsize=524288,nfsvers=3,actimeo=0,nointr,timeo=600,tcp
 10.10.23.41:/soedata07   /soedata07       nfs
rw,bg,hard,rsize=524288,wsize=524288,nfsvers=3,actimeo=0,nointr,timeo=600,tcp
 10.10.24.42:/soedata08   /soedata08       nfs
rw,bg,hard,rsize=524288,wsize=524288,nfsvers=3,actimeo=0,nointr,timeo=600,tcp
 10.10.21.41:/soedata09   /soedata09       nfs
rw,bg,hard,rsize=524288,wsize=524288,nfsvers=3,actimeo=0,nointr,timeo=600,tcp
 10.10.22.42:/soedata10   /soedata10       nfs
rw,bg,hard,rsize=524288,wsize=524288,nfsvers=3,actimeo=0,nointr,timeo=600,tcp
 10.10.23.41:/soedata11   /soedata11       nfs
rw,bg,hard,rsize=524288,wsize=524288,nfsvers=3,actimeo=0,nointr,timeo=600,tcp
 10.10.24.42:/soedata12   /soedata12       nfs
rw,bg,hard,rsize=524288,wsize=524288,nfsvers=3,actimeo=0,nointr,timeo=600,tcp
 10.10.21.41:/soedata13   /soedata13       nfs
rw,bg,hard,rsize=524288,wsize=524288,nfsvers=3,actimeo=0,nointr,timeo=600,tcp
 10.10.22.42:/soedata14   /soedata14       nfs
rw,bg,hard,rsize=524288,wsize=524288,nfsvers=3,actimeo=0,nointr,timeo=600,tcp
 10.10.23.41:/soedata15   /soedata15       nfs
rw,bg,hard,rsize=524288,wsize=524288,nfsvers=3,actimeo=0,nointr,timeo=600,tcp
 10.10.24.42:/soedata16   /soedata16       nfs
rw,bg,hard,rsize=524288,wsize=524288,nfsvers=3,actimeo=0,nointr,timeo=600,tcp
 10.10.21.41:/soelog01   /soelog01       nfs
rw,bg,hard,rsize=524288,wsize=524288,nfsvers=3,actimeo=0,nointr,timeo=600,tcp
 10.10.22.42:/soelog02   /soelog02       nfs
rw,bg,hard,rsize=524288,wsize=524288,nfsvers=3,actimeo=0,nointr,timeo=600,tcp
 10.10.23.41:/soelog03   /soelog03       nfs
rw,bg,hard,rsize=524288,wsize=524288,nfsvers=3,actimeo=0,nointr,timeo=600,tcp
 10.10.24.42:/soelog04   /soelog04       nfs
rw,bg,hard,rsize=524288,wsize=524288,nfsvers=3,actimeo=0,nointr,timeo=600,tcp
 10.10.21.41:/shdata01   /shdata01       nfs
rw,bg,hard,rsize=524288,wsize=524288,nfsvers=3,actimeo=0,nointr,timeo=600,tcp
 10.10.22.42:/shdata02   /shdata02       nfs
rw,bg,hard,rsize=524288,wsize=524288,nfsvers=3,actimeo=0,nointr,timeo=600,tcp
```

```
10.10.23.41:/shdata03   /shdata03       nfs
rw,bg,hard,rsize=524288,wsize=524288,nfsvers=3,actimeo=0,nointr,timeo=600,tcp
10.10.24.42:/shdata04   /shdata04       nfs
rw,bg,hard,rsize=524288,wsize=524288,nfsvers=3,actimeo=0,nointr,timeo=600,tcp
10.10.21.41:/shdata05   /shdata05       nfs
rw,bg,hard,rsize=524288,wsize=524288,nfsvers=3,actimeo=0,nointr,timeo=600,tcp
10.10.22.42:/shdata06   /shdata06       nfs
rw,bg,hard,rsize=524288,wsize=524288,nfsvers=3,actimeo=0,nointr,timeo=600,tcp
10.10.23.41:/shdata07   /shdata07       nfs
rw,bg,hard,rsize=524288,wsize=524288,nfsvers=3,actimeo=0,nointr,timeo=600,tcp
10.10.24.42:/shdata08   /shdata08       nfs
rw,bg,hard,rsize=524288,wsize=524288,nfsvers=3,actimeo=0,nointr,timeo=600,tcp
10.10.21.41:/shdata09   /shdata09       nfs
rw,bg,hard,rsize=524288,wsize=524288,nfsvers=3,actimeo=0,nointr,timeo=600,tcp
10.10.22.42:/shdata10   /shdata10       nfs
rw,bg,hard,rsize=524288,wsize=524288,nfsvers=3,actimeo=0,nointr,timeo=600,tcp
10.10.23.41:/shdata11   /shdata11       nfs
rw,bg,hard,rsize=524288,wsize=524288,nfsvers=3,actimeo=0,nointr,timeo=600,tcp
10.10.24.42:/shdata12   /shdata12       nfs
rw,bg,hard,rsize=524288,wsize=524288,nfsvers=3,actimeo=0,nointr,timeo=600,tcp
10.10.21.41:/shdata13   /shdata13       nfs
rw,bg,hard,rsize=524288,wsize=524288,nfsvers=3,actimeo=0,nointr,timeo=600,tcp
10.10.22.42:/shdata14   /shdata14       nfs
rw,bg,hard,rsize=524288,wsize=524288,nfsvers=3,actimeo=0,nointr,timeo=600,tcp
10.10.23.41:/shdata15   /shdata15       nfs
rw,bg,hard,rsize=524288,wsize=524288,nfsvers=3,actimeo=0,nointr,timeo=600,tcp
10.10.24.42:/shdata16   /shdata16       nfs
rw,bg,hard,rsize=524288,wsize=524288,nfsvers=3,actimeo=0,nointr,timeo=600,tcp
10.10.21.41:/shlog01   /shlog01       nfs
rw,bg,hard,rsize=524288,wsize=524288,nfsvers=3,actimeo=0,nointr,timeo=600,tcp
10.10.22.42:/shlog02   /shlog02       nfs
rw,bg,hard,rsize=524288,wsize=524288,nfsvers=3,actimeo=0,nointr,timeo=600,tcp
10.10.23.41:/shlog03   /shlog03       nfs
rw,bg,hard,rsize=524288,wsize=524288,nfsvers=3,actimeo=0,nointr,timeo=600,tcp
10.10.24.42:/shlog04   /shlog04       nfs
rw,bg,hard,rsize=524288,wsize=524288,nfsvers=3,actimeo=0,nointr,timeo=600,tcp
```

Configuration of “oranfstab”

```
[root@flex1 ~]# cat /u01/app/oracle/product/21.3.0/dbhome_1/dbs/oranfstab
Server: NetApp-A800
path: 10.10.21.41
path: 10.10.22.41
path: 10.10.23.41
path: 10.10.24.41
```

```
path: 10.10.21.42
path: 10.10.22.42
path: 10.10.23.42
path: 10.10.24.42
nfs_version: nfsv3
export: /soedata01 mount: /soedata01
export: /soedata02 mount: /soedata02
export: /soedata03 mount: /soedata03
export: /soedata04 mount: /soedata04
export: /soedata05 mount: /soedata05
export: /soedata06 mount: /soedata06
export: /soedata07 mount: /soedata07
export: /soedata08 mount: /soedata08
export: /soedata09 mount: /soedata09
export: /soedata10 mount: /soedata10
export: /soedata11 mount: /soedata11
export: /soedata12 mount: /soedata12
export: /soedata13 mount: /soedata13
export: /soedata14 mount: /soedata14
export: /soedata15 mount: /soedata15
export: /soedata16 mount: /soedata16
export: /soelog01 mount: /soelog01
export: /soelog02 mount: /soelog02
export: /soelog03 mount: /soelog03
export: /soelog04 mount: /soelog04
export: /findata01 mount: /findata01
export: /findata02 mount: /findata02
export: /findata03 mount: /findata03
export: /findata04 mount: /findata04
export: /findata05 mount: /findata05
export: /findata06 mount: /findata06
export: /findata07 mount: /findata07
export: /findata08 mount: /findata08
export: /findata09 mount: /findata09
export: /findata10 mount: /findata10
export: /findata11 mount: /findata11
export: /findata12 mount: /findata12
export: /findata13 mount: /findata13
export: /findata14 mount: /findata14
```

```
export: /findata15 mount: /findata15
export: /findata16 mount: /findata16
export: /finlog01 mount: /finlog01
export: /finlog02 mount: /finlog02
export: /finlog03 mount: /finlog03
export: /finlog04 mount: /finlog04
export: /shdata01 mount: /shdata01
export: /shdata02 mount: /shdata02
export: /shdata03 mount: /shdata03
export: /shdata04 mount: /shdata04
export: /shdata05 mount: /shdata05
export: /shdata06 mount: /shdata06
export: /shdata07 mount: /shdata07
export: /shdata08 mount: /shdata08
export: /shdata09 mount: /shdata09
export: /shdata10 mount: /shdata10
export: /shdata11 mount: /shdata11
export: /shdata12 mount: /shdata12
export: /shdata13 mount: /shdata13
export: /shdata14 mount: /shdata14
export: /shdata15 mount: /shdata15
export: /shdata16 mount: /shdata16
export: /shlog01 mount: /shlog01
export: /shlog02 mount: /shlog02
export: /shlog03 mount: /shlog03
export: /shlog04 mount: /shlog04
```

Feedback

For comments and suggestions about this guide and related guides, join the discussion on [Cisco Community](https://cs.co/en-cvds) at https://cs.co/en-cvds.

CVD Program

Americas Headquarters
Cisco Systems, Inc.
San Jose, CA

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

Europe Headquarters
Cisco Systems International BV Amsterdam,
The Netherlands

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