



The bridge to possible

Design and Deployment Guide
Cisco Public

FlexPod Datacenter with Red Hat OpenShift Virtualization

Design and Deployment Guide

Published: December 2024



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

The FlexPod Datacenter solution is a validated approach for deploying Cisco and NetApp technologies and products to build shared private and public cloud infrastructure. Cisco and NetApp have partnered to deliver a series of FlexPod solutions that enable strategic datacenter platforms. The success of the FlexPod solution is driven through its ability to evolve and incorporate both technology and product innovations in the areas of management, compute, storage, and networking. This document explains the design and deployment details of adding Red Hat OpenShift Virtualization on the latest FlexPod Datacenter with Red Hat OpenShift on Bare Metal design. This solution allows you to run and manage virtual machine workloads alongside container workloads. Some of the key advantages of OpenShift Virtualization on FlexPod Datacenter are:

- Create, manage, clone, and import Linux and Windows virtual machines (VMs).
- Run containerized and VM workloads alongside each other in a cluster.
- Manage VM network interfaces and storage disks.
- Live migrate VMs between cluster nodes.
- Attach and use hardware devices such as GPUs.
- Migrate VMs from other virtualization environments into OpenShift Virtualization.

In addition to the compute-specific hardware and software innovations, the integration of the Cisco Intersight cloud platform with NetApp Active IQ Unified Manager and Cisco Nexus switches delivers monitoring, and orchestration capabilities for different layers (storage and networking) of the FlexPod infrastructure. The modular nature of the Cisco Intersight platform also provides an easy upgrade path to additional services as they become available.

For information about 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>

Solution Overview and Design

This chapter contains the following:

- [Introduction](#)
- [Audience](#)
- [Purpose of this Document](#)
- [What's New in this Release?](#)
- [Solution Summary](#)

Introduction

Red Hat OpenShift Virtualization is an extension of Red Hat OpenShift that enables you to run and manage virtual machines alongside containerized applications in a unified environment. This integration allows organizations to modernize their IT infrastructure by bridging the gap between traditional virtual machine workloads and modern cloud-native applications. OpenShift Virtualization leverages Kubernetes orchestration to manage both containers and virtual machines, providing a seamless and consistent user experience. By doing so, it helps reduce the complexity of managing diverse workloads, enabling developers and IT operations teams to work more efficiently and effectively.

One of the key benefits of OpenShift Virtualization is its ability to facilitate a smoother transition to cloud-native architectures without requiring a complete overhaul of existing applications. Organizations can gradually refactor or replatform their applications while still running them in virtual machines, providing a flexible approach to modernization. Additionally, OpenShift Virtualization supports hybrid cloud strategies by allowing workloads to be easily moved between on-premises and cloud environments. This flexibility and interoperability make it a valuable tool for businesses looking to optimize their IT operations, improve resource utilization, and accelerate application development and deployment.

Audience

The intended audience of this document includes but is not limited to IT architects, sales engineers, field consultants, professional services, IT managers, partner engineering, and customers who want to take advantage of an infrastructure built to deliver IT efficiency and enable IT innovation.

Purpose of this Document

This document provides design and deployment guidance around implementing FlexPod Datacenter with Red Hat OpenShift Virtualization. This document assumes the FlexPod has already been configured with Red Hat OpenShift on Bare Metal as specified in [FlexPod Datacenter with Red Hat OCP Bare Metal Manual Configuration with Cisco UCS X-Series Direct - Cisco](#).

What's New in this Release?

The following design elements are added to FlexPod Datacenter with Red Hat OpenShift on Bare Metal to build the OpenShift Virtualization Platform:

-
- Configuration updates to the Cisco Nexus switches, the NetApp ONTAP storage, and Cisco Intersight Managed Mode policies and templates.
 - The OpenShift Virtualization Operator
 - The Migration Toolkit for Virtualization Operator
 - Demonstration of how to create, manage, clone, import, and migrate Linux and Windows VMs

Solution Summary

The FlexPod Datacenter solution as a platform for OpenShift Virtualization offers the following key benefits:

- The ability to run containerized and VM workloads alongside each other in a cluster.
- Save time and reduce errors with deployment ready Ansible playbooks for the base FlexPod setup and for FlexPod additions in the future.
- 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

Like all other FlexPod solution designs, FlexPod Datacenter with OpenShift Virtualization is configurable according to demand and usage. You can purchase exactly the infrastructure you need for your current application requirements and can then scale-up by adding more resources to the FlexPod system or scale-out by adding more FlexPod instances. This platform is also highly flexible, allowing you to run both containerized and VM-based workloads as needed on the same infrastructure.

Technology Overview

This chapter contains the following:

- [FlexPod Datacenter](#)
- [Red Hat OpenShift Virtualization](#)

FlexPod Datacenter

The IP-based FlexPod Datacenter with Red Hat Openshift on Bare Metal was used as the basis for this solution, and is specified here: [FlexPod Datacenter with Red Hat OCP Bare Metal Manual Configuration with Cisco UCS X-Series Direct – Cisco](#). This document will not repeat information from that document but will instead present what was updated or added for each component. The base FlexPod used in this validation used the following specific components:

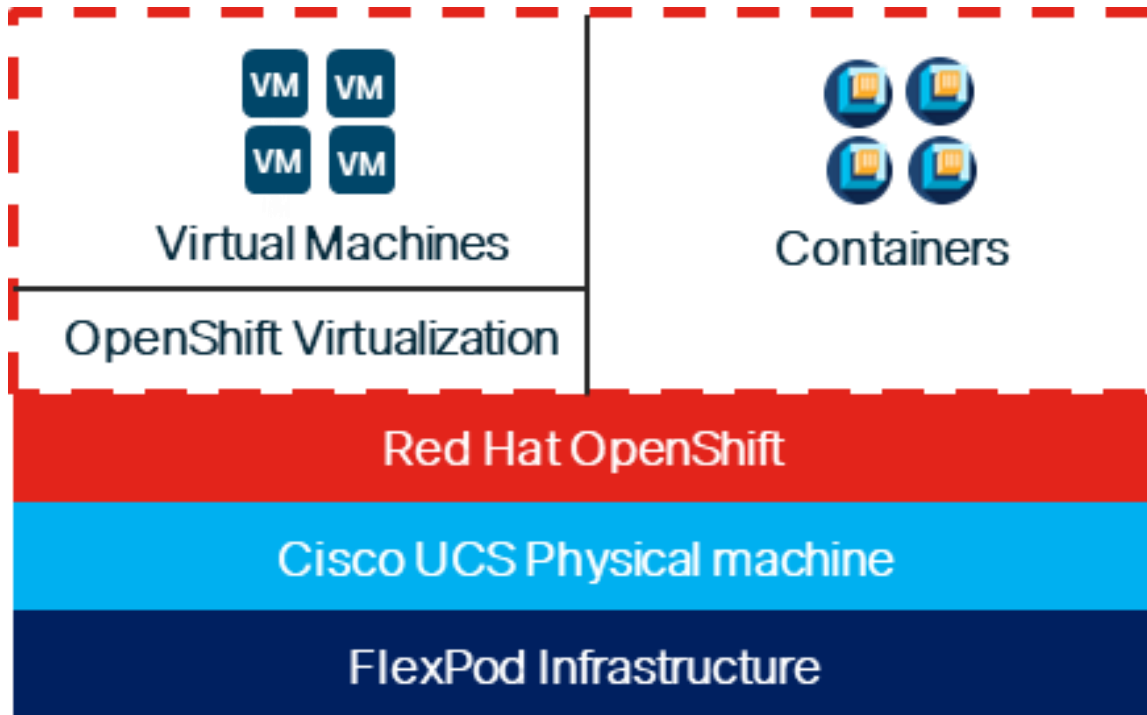
- Cisco UCS X9508 Chassis with Cisco UCSX-S9108-100G Fabric Interconnect Modules, Cisco UCS 9416 X-Fabric Modules and Cisco UCS X210c M7 Compute Nodes
- High-speed Cisco NX-OS-based Cisco Nexus 93600CD-GX switching designed to support up to 400GE connectivity
- NetApp AFF C800 end-to-end NVMe storage with up to 100GE connectivity
- NVIDIA L40S GPUs in Cisco UCS X440p PCIe Nodes connected to Cisco UCS X210c M7 Servers by Cisco UCS X-Fabric

Red Hat OpenShift Virtualization

OpenShift Virtualization integrates virtual machines into the OpenShift environment using KVM, the Linux kernel hypervisor, allowing them to be managed as native objects. This setup benefits from OpenShift's cluster management and scalability features. KVM, a core part of the Red Hat Enterprise Linux kernel, has been a trusted component in various Red Hat products for over 15 years, such as Red Hat Virtualization and Red Hat OpenStack Platform. The KVM, QEMU, and libvirt used by OpenShift are consistent with those platforms, making OpenShift a reliable type-1 hypervisor for virtualization tasks. Virtual machines on OpenShift are managed using the same scheduling system that handles non-virtual machine workloads, inheriting features like host affinity, resource awareness, load balancing, and high availability from OpenShift. Red Hat OpenShift Virtualization uses the same NetApp Trident Container Storage Interface (CSI) for persistent storage for VM disks that is used for persistent storage to containers.

When Red Hat OpenShift Virtualization is added to a Red Hat OpenShift environment, both containers and VMs can now be run side-by-side on the same infrastructure as shown below. When applications are being containerized, if there are any issues with containerization of the application, the application can be left to run in a VM.

Figure 1. Containers and VMs in the Same Infrastructure



Solution Design

This chapter contains the following:

- [Design Requirements](#)
- [Physical Topology](#)
- [FlexPod Multi-Tenant Configuration](#)
- [Red Hat OpenShift Virtualization](#)
- [Design Summary](#)

Design Requirements

The FlexPod Datacenter with Cisco UCS and Cisco Intersight meets the following general design requirements:

- Resilient design across all layers of the infrastructure with no single point of failure
- Scalable design with the flexibility to add compute capacity, storage, or network bandwidth as needed
- Modular design that can be replicated to expand and grow as the needs of the business grow
- Flexible design that can support different models of various components with ease
- Simplified design with ability to integrate and automate with external automation tools
- Cloud-enabled design which can be configured, managed, and orchestrated from the cloud using GUI or APIs

To deliver a solution which meets all these design requirements, various solution components are connected and configured as explained in the following sections.

Physical Topology

The FlexPod Datacenter solution with OpenShift Virtualization is built using the following hardware components:

- Cisco UCS X-Series Direct X9508 Chassis with Cisco UCSX-S9108-100G Fabric Interconnects, up to eight Cisco UCS X210C M7 Compute Nodes with 4th or 5th Generation Intel Xeon Scalable CPUs, and up to four Cisco UCS X440p PCIe Nodes each with up to two NVIDIA L40S GPUs (each X440p would go in place of an X210c)
- High-speed Cisco NX-OS-based Nexus 93600CD-GX switching design to support 100GE and 400GE connectivity
- NetApp AFF C800 end-to-end NVMe storage with 100G or 25G Ethernet

Note: Even though this solution validation was built with Cisco UCS X-Series Direct, all Cisco UCS components that support Red Hat CoreOS 4.16 on the Cisco UCS Hardware Compatibility List (HCL) are supported with this solution.

The software components of this solution consist of:

- Cisco Intersight to deploy, maintain, and support the Cisco UCS server components
- Cisco Intersight SaaS platform to maintain and support the FlexPod hardware components

- Cisco Intersight Assist Virtual Appliance to help connect NetApp ONTAP and Cisco Nexus switches with Cisco Intersight
- NetApp Active IQ Unified Manager to monitor and manage the storage and for NetApp ONTAP integration with Cisco Intersight
- Red Hat OpenShift to manage a Kubernetes containerized environment
- The Red Hat OpenShift Virtualization Operator to deploy and manage the OpenShift Virtualization environment
- The Red Hat Migration Toolkit for Virtualization Operator to migrate VMs from other virtualization environments to Red Hat OpenShift Virtualization or to migrate VMs between Red Hat OpenShift Virtualization environments
- NetApp Trident to provide persistent storage for VM disks

FlexPod Datacenter with OpenShift Virtualization

Figure 2 shows various hardware components and the network connections for the IP-based FlexPod design for FlexPod Datacenter with OpenShift Virtualization with Cisco UCS X-Series Direct.

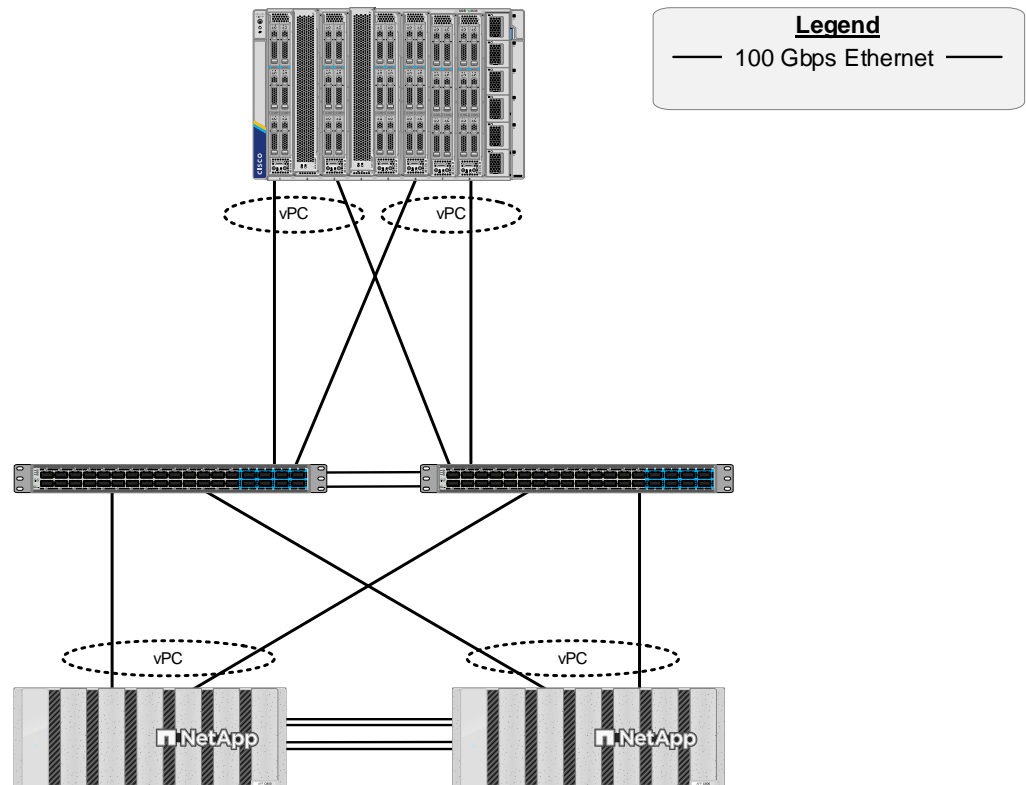
Figure 2. FlexPod Datacenter Physical Topology for IP-based Storage Access with Cisco UCS X-Series Direct

Cisco UCS X-Series Direct

Cisco UCS S9108 Fabric Interconnect in Cisco UCS 9508 Chassis, Cisco UCS M7 Servers

Cisco Nexus 93600CD-GX

NetApp storage controllers AFF-C800



The reference hardware configuration includes:

-
- Two Cisco Nexus 93600CD-GX Switches in Cisco NX-OS mode provide the switching fabric.
 - Two Cisco UCS S9108 Fabric Interconnects (FI) in the Cisco UCS X9508 chassis provide the chassis connectivity. Two 100 Gigabit Ethernet ports from each FI, configured as a Port-Channel, are connected to each Nexus 93600CD-GX.
 - The Cisco UCS X9508 Chassis is also equipped with a pair of Cisco UCS 9416 X-Fabric modules to accommodate GPUs.
 - One NetApp AFF C800 HA pair connects to the Cisco Nexus 93600CD-GX Switches using two 100 GE ports from each controller configured as a Port-Channel.
 - Up to two NVIDIA L40S GPUs are installed in the Cisco UCS X440p cards and are connected to the Cisco UCS X210C M7 in the adjacent slot by X-Fabric.
 - This reference configuration consists of 2 Cisco UCS X210c M7 servers each with an X440p card with 2 NVIDIA L40S GPUs, and 4 additional Cisco UCS X210c M7 servers.

Note: It is important to note that the Cisco UCS S9108 FI is a fully functional FI with 8 100G ports as shown in [Figure 3](#). The only difference between this FI and other FIs is the rules around how the ports can be configured. Currently, Cisco UCS C-Series servers and an additional X9508 chassis cannot be attached to the S9108 FI. All other FI policies can be applied to this FI in the same way as all the other FIs. In the datacenter, the Cisco UCS S9108 FIs provide a UCS Domain per X-Series Chassis. If using Cisco Intersight UCS Domain Profile Templates, all Cisco UCS S9108 Domains can be configured identically, and all servers behind the S9108s can be configured from a single pool.

Figure 3. Cisco UCS X9508 Chassis with 2 Cisco UCS S9108 FIs



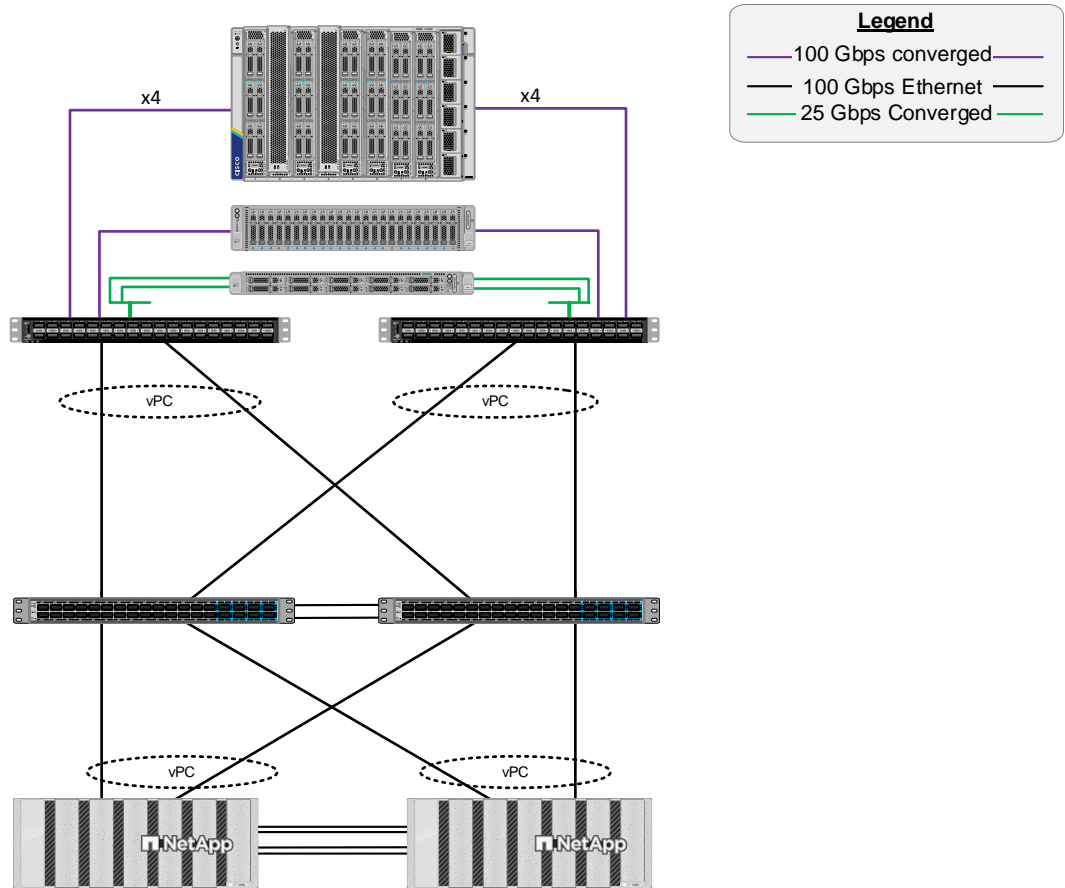
A standard FlexPod topology with a standalone FI such as the Cisco UCS 6536 FI as shown in [Figure 4](#) is also fully supported for OpenShift Virtualization.

Figure 4. FlexPod Datacenter Physical Topology for IP-based Storage Access with Cisco UCS 6536 FI

Cisco Unified Computing System
 Cisco UCS 6536 Fabric Interconnect, Cisco UCS 9508 Chassis with 9108-100G IFM, Cisco UCS M7 Servers

Cisco Nexus 9360CD-GX

NetApp storage controllers AFF-C800



FlexPod Multi-Tenant Configuration

In the deployment section of this document, [FlexPod Datacenter with Red Hat OCP Bare Metal Manual Configuration with Cisco UCS X-Series Direct](#) is first set up as a FlexPod tenant on top of FlexPod Base. This solution’s deployment layers OpenShift Virtualization on top of FlexPod Datacenter with Red Hat OpenShift on Bare Metal. With Red Hat OpenShift Virtualization, three Cisco virtual network interfaces (vNICs) are added, along with at least one VLAN for VM management. This VLAN is added to the Nexus switches, to the NetApp storage and to the UCS Domain Profile and worker node Server Profile.

Red Hat OpenShift Virtualization

Red Hat OpenShift Virtualization is layered onto a Red Hat OpenShift Bare Metal environment by installing and configuring the OpenShift Virtualization Operator and a HyperConverged Deployment as shown below. By default, the OpenShift Virtualization Operator is deployed in the openshift-cnv namespace and initial VMs can be configured there, but before VMs can be configured, VM networking and a place to store VMs will need to be set up.

Project: openshift-cnvd

Installed Operators > Operator details



OpenShift Virtualization
4.16.4 provided by Red Hat

Actions

Details | **YAML** | Subscription | Events | All instances | **OpenShift Virtualization Deployment** | HostPathProvisioner Deployment

HyperConvergeds

Create HyperConverged

Name Search by name...

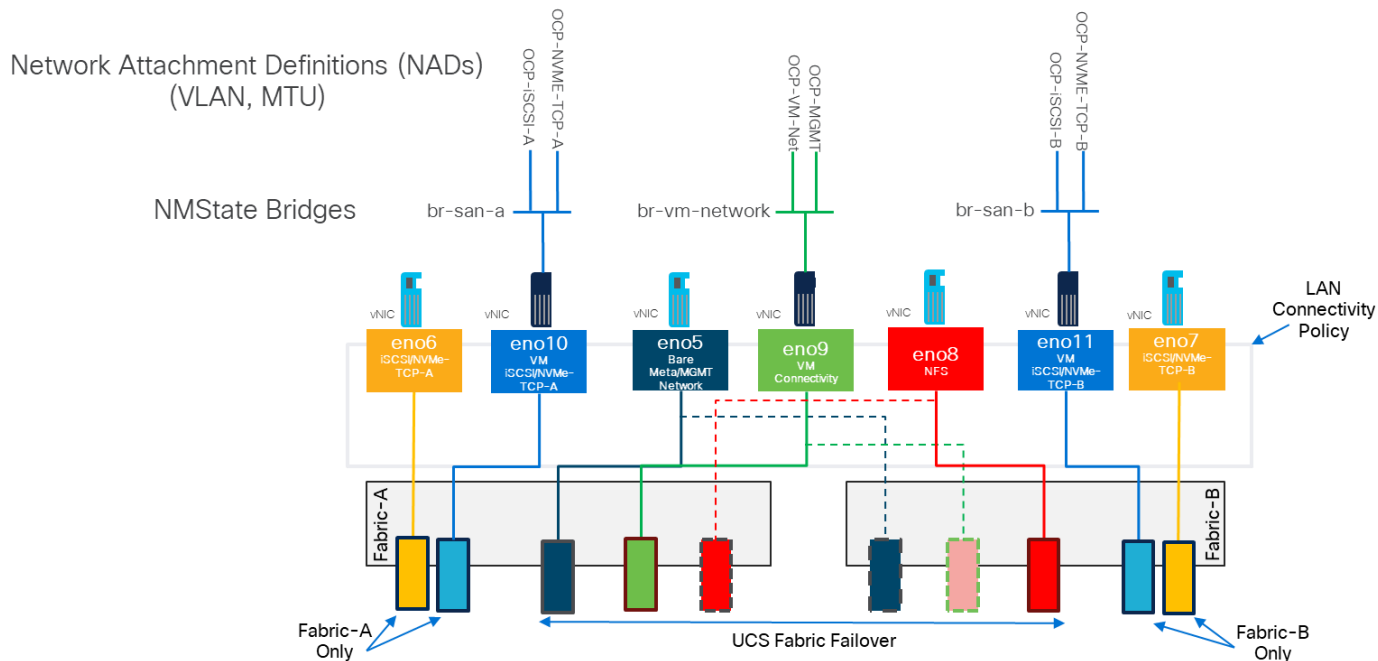
Name	Kind	Status	Labels	Last updated
kubevirt-hyperconverged	HyperConverged	Conditions: Reconcile Complete, Degraded	app=kubevirt-hyperconverged	Nov 4, 2024, 1:06 PM

OpenShift Virtualization VM Networking

In the Red Hat OpenShift Bare Metal environment, the NMState Operator was used to set up and maintain OpenShift server networking. With OpenShift Virtualization in this validation, NMState is used to create network bridges on vNICs added to the UCS Server Profiles for OpenShift Virtualization. Network Attachment Definitions (NADs) with specified VLAN tags and MTUs are then created on top of the bridges and attach VMs to the network.

[Figure 5](#) illustrates that 3 vNICs (eno9, eno10, and eno11) are added to the Worker Server Profile Template in addition to the 4 vNICs that were used for OpenShift on Bare Metal. vNIC eno9 is used for VM front end connectivity and multiple VLANs can be configured on that vNIC. It is required that those VLANs are configured in the Cisco Nexus switches and also in the Cisco UCS Domain Profile VLAN policy and in the Ethernet Network Group policy attached to the vNIC in the LAN Connectivity policy. This vNIC is configured with Cisco UCS Fabric Failover which will fail the vNIC over to the other FI in case of an FI failure or reboot. vNICs eno10 and eno11 are identical to vNICs eno6 and eno7 and provide VM in-guest iSCSI and NVMe-TCP connectivity. These vNICs do not have Fabric Failover configured. Additional iSCSI and NVMe-TCP VLANs can be added to these vNICs in the same way VLANs are added to vNIC eno9. NMState Bridges are configured on eno9, eno10, and eno11 using Node Network Configuration Policies (NNCPs). Then, Network Attachment Definitions (NADs) are added, specifying the VLAN tag and the MTU (if 9000). VM NICs then reference the NAD and are attached to the correct VLAN. NADs configured in the default namespace or project are globally available to VMs in any namespace. NADs can also be configured in a specific namespace and are then only visible to VMs in that namespace. All three of the added vNICs use the Ethernet Adapter policy with 16 RX queues and Receive Side Scaling (RSS) enabled to allow the RX queues to be serviced by different CPU cores.

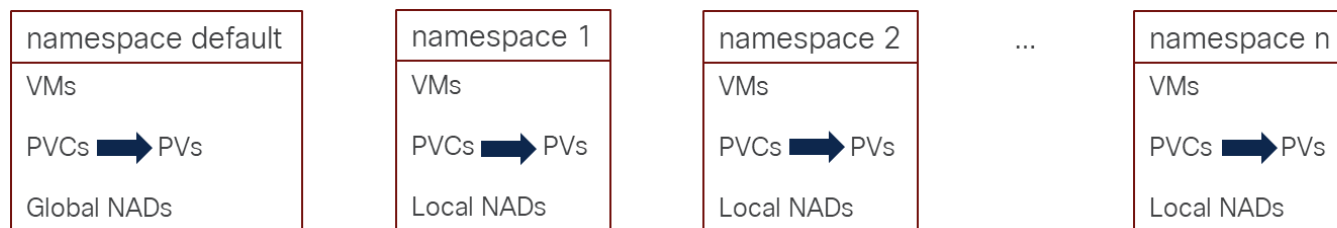
Figure 5. OpenShift Virtualization VM Attachment to the Network



Grouping and Separating VMs with namespaces or Projects

VMs can be created in the default openshift-cnv namespace or project, or in any other namespace. Using namespaces in this way allows VMs to be both grouped together and separated from other VMs. [Figure 6](#) shows this grouping. With respect to NADs, as described above, NADs created within the default namespace are globally visible to VMs, where NADs created within other namespaces are only visible to VMs within that namespace. Storage persistent volume claims (PVCs) for VM disks are also contained within the namespace, but it is important to note that the corresponding persistent volumes (PVs) are tracked at the cluster level and not within namespaces.

Figure 6. VM Grouping and Separation Using namespaces



VM Disk Storage with NetApp Trident

Just as in the FlexPod OpenShift with Bare Metal CVD, persistent storage for VM disks is provided by NetApp Trident. NFS, iSCSI, and NVMe-TCP storage can be provided by Trident, but only NFS and iSCSI can be used for VMs that support Live Migration of VMs between nodes. Live Migration requires that storage be configured with the ReadWriteMany access mode which NFS provides by default. With NetApp ONTAP iSCSI, ReadWriteMany can be configured by using the Block volume mode instead of Filesystem. NVMe-TCP VM disks can also be configured

with the Block volume mode and ReadWriteMany access mode, but after a Live Migration, the VM will go into a Paused state and the only way to recover the VM is to Force Stop it and then power it back on. When the VM comes back up, it will be on the node it was on before the Live Migration. It is important to note that with OpenShift Virtualization, each VM disk is stored in a separate volume in the NetApp storage, as opposed to the VMware vSphere environment where many VMs and their disks are stored within a datastore volume.

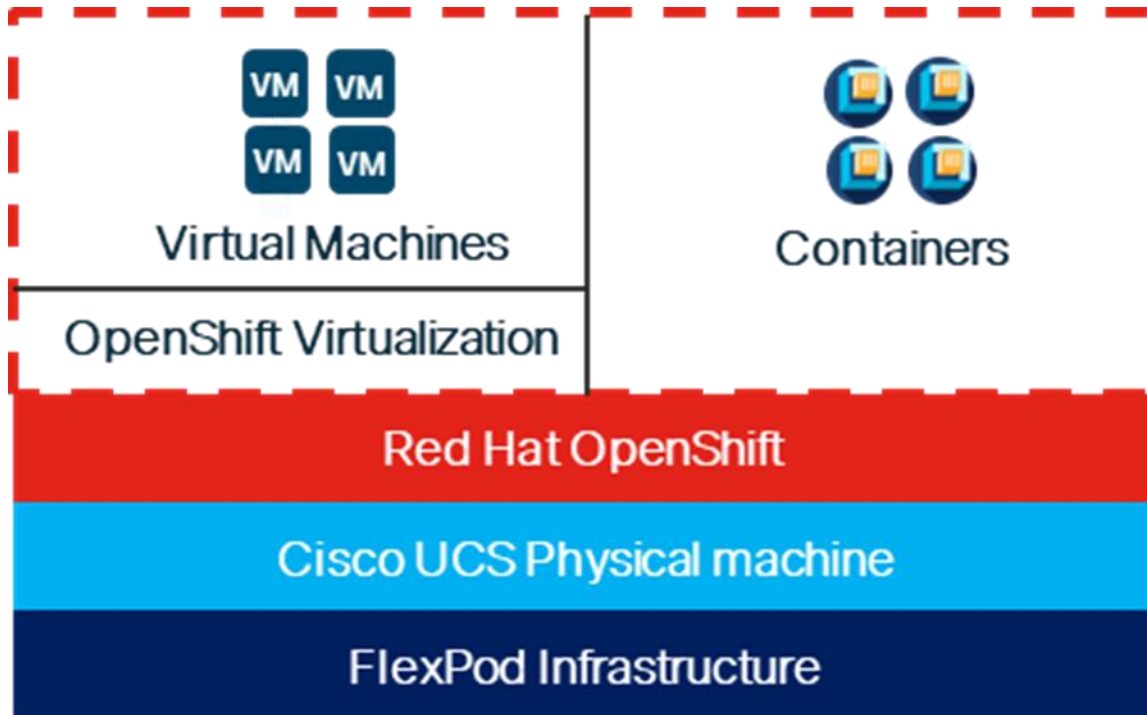
VM Migration into OpenShift Virtualization

The OpenShift Migration Toolkit for Virtualization allows VMs to be imported from VMware vSphere and Red Hat OpenStack into OpenShift as well as migrating OpenShift VMs between clusters. VMs for OVA format can also be imported. In this validation, VMware vSphere VMs were imported to OpenShift. Only RHEL 8 and 9 and Windows VMs were successfully imported. It was also found that VMs with SCSI disk controllers (preferably VMware para-virtual disk controllers) would successfully migrate. VMs with NVMe based disk controllers would migrate but could not be successfully started in OpenShift. All migrated VMs were brought into OpenShift with both virtio network interfaces and virtio disk controllers. In this migration, Windows 11 VMs (which are required to be encrypted in VMware) could not successfully be migrated to OpenShift VMs. It was also found that it is best to set up a direct Layer 2 (in the same VLAN) connection between the VMware ESXi hosts and OpenShift nodes for VM disk migration.

Design Summary

FlexPod Base was first setup on the infrastructure shown in [Figure 7](#). Then FlexPod Datacenter with Red Hat OpenShift on Bare Metal was installed as a tenant on top of FlexPod Base. Next, OpenShift Virtualization was installed and configured on top of OpenShift on Bare Metal along with the OpenShift Migration Toolkit for Virtualization. This results in a single modern platform that supports both containerized applications and VMs. In the journey to containerization, if an application cannot be fully containerized, the VM is a fallback for that application.

Figure 7. A Single Environment for both Containers and VMs



Solution Deployment

This chapter contains the following:

- [VLAN Configuration](#)
- [Software Revisions](#)
- [Deploy FlexPod](#)
- [Deploy OpenShift Virtualization](#)

Note: The NetApp storage controller and disk shelves should be connected according to best practices for the specific storage controller and disk shelves. For disk shelf cabling, go to [NetApp Support: https://docs.netapp.com/us-en/ontap-systems/index.html](https://docs.netapp.com/us-en/ontap-systems/index.html)

VLAN Configuration

[Table 1](#) lists VLANs configured for setting up the FlexPod environment along with their usage.

Table 1. VLAN Usage

VLAN ID	Name	Usage	IP Subnet used in this deployment
2	Native-VLAN	Use VLAN 2 as native VLAN instead of default VLAN (1).	
1020	OOB-MGMT-VLAN	Out-of-band management VLAN to connect management ports for various devices	10.102.0.0/24; GW: 10.102.0.254
1022	OCP-BareMetal-MGMT	OpenShift Bare Metal or Management VLAN for the Openshift Nodes and Cluster	10.102.2.0/24; GW: 10.102.2.254
1023*	VM-MGMT	OpenShift VM Management VLAN - for VM front end interfaces	10.102.3.0/24; GW: 10.102.3.254
3052	OCP-NFS	NFS VLAN for persistent storage	192.168.52.0/24 **
3012	OCP-iSCSI-A	iSCSI-A path for persistent storage	192.168.12.0/24 **
3022	OCP-iSCSI-B	iSCSI-B path for persistent storage	192.168.22.0/24 **
3032	OCP-NVMe-TCP-A	NVMe-TCP-A path for persistent storage	192.168.32.0/24 **
3042	OCP-NVMe-TCP-B	NVMe-TCP-B path for persistent storage	192.168.42.0/24 **

* Added for OpenShift Virtualization. All other VLANs were already added in OpenShift on Bare Metal.

** IP gateway is not needed since no routing is required for these subnets

Software Revisions

[Table 2](#) lists the software revisions for various components of the solution.

Table 2. Software Revisions

Layer	Device	Image Bundle	Comments
Compute	Cisco UCS	4.3(4.240078)	Cisco UCS GA release for infrastructure including FIs and IFM
	Cisco UCS X210C M7	5.2(2.240053)	
GPU	NVIDIA L40S	550.127.08	
Network	Cisco Nexus 93600CD-GX NX-OS	10.3(6)M	
Storage	NetApp AFF C800	ONTAP 9.14.1	Latest patch release
Software	Red Hat CoreOS	4.16	Latest patch release
	Red Hat OpenShift	4.16	Latest patch release
	NetApp Trident	24.10.0	

FlexPod Cabling

The information in this section is provided as a reference for cabling the physical equipment in a FlexPod environment. To simplify cabling requirements, a cabling diagram was used.

The cabling diagram in this section contains the details for the prescribed and supported configuration of the NetApp AFF C800 running NetApp ONTAP 9.14.1.

Note: For any modifications of this prescribed architecture, see the [NetApp Interoperability Matrix Tool \(IMT\)](#).

Note: This document assumes that out-of-band management ports are plugged into an existing management infrastructure at the deployment site. These interfaces will be used in various configuration steps.

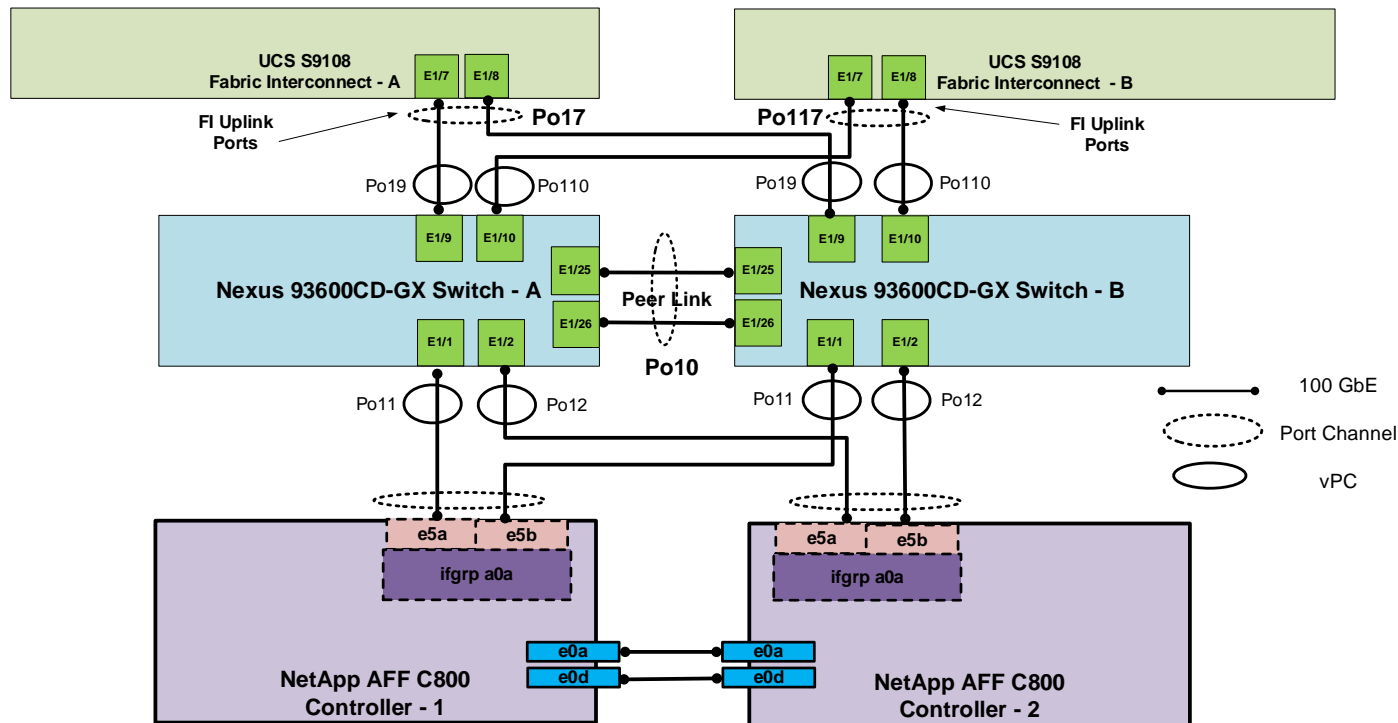
Note: Be sure to use the cabling directions in this section as a guide.

The NetApp storage controller and disk shelves should be connected according to best practices for the specific storage controller and disk shelves. For disk shelf cabling, go to [NetApp Support](#).

[Figure 8](#) details the cable connections used in the validation lab for the FlexPod topology based on the Cisco UCS S9108 fabric interconnect installed in the Cisco UCS X9508 chassis. Two 100Gb links connect each Cisco UCS Fabric Interconnect to the Cisco Nexus Switches and each NetApp AFF controller to the Cisco Nexus Switches. Additional 1Gb management connections will be needed for out-of-band network switches that sit apart from the FlexPod infrastructure. Each Cisco UCS fabric interconnect and Cisco Nexus switch is connected to the

out-of-band network switches, and each AFF controller has a connection to the out-of-band network switches. Layer 3 network connectivity is required between the Out-of-Band (OOB) and In-Band (IB) Management Subnets. This cabling diagram shows the iSCSI-boot configuration.

Figure 8. FlexPod Cabling with Cisco UCS 6536 Fabric Interconnect



Deploy FlexPod

Procedure 1. Deploy FlexPod with OpenShift on Bare Metal

Step 1. Using the information in the above tables and diagrams, use [FlexPod Datacenter Base Configuration using IaC with Cisco IMM and NetApp ONTAP](#) or [FlexPod Datacenter Base Manual Configuration with Cisco IMM and NetApp ONTAP](#) to deploy the Base FlexPod.

Step 2. Use [FlexPod Datacenter with Red Hat OCP Bare Metal Manual Configuration with Cisco UCS X-Series Direct](#) to deploy Red Hat OpenShift with Bare Metal on the environment.

Deploy OpenShift Virtualization

Use the following procedures and steps to add OpenShift Virtualization to the OpenShift on Bare Metal environment.

Procedure 1. Configure Nexus Switches for the Openshift Virtualization

Run the following commands to add at least one VLAN with a routable subnet for VM management connections. Also, optionally add NTP distribution interfaces in this subnet. Execute these steps in an ssh session on both switches.

```
config t
```

```
vlan 1023
name OCP-VM-MGMT
exit
int Po10,Po19,Po110,Po127 # Po127 is switch uplink port channel
switchport trunk allowed vlan add 1023 # Add OCP-VM-MGMT VLAN to vPC peer link, FI, and uplink port channels
exit
vrf context OCP-VM-MGMT
description VRF for routing OCP-VM-MGMT subnets/VLANs
ip route 0.0.0.0/0 10.102.3.254
interface VLAN1023
vrf member OCP-VM-MGMT
ip address 10.102.3.3/24 # Use 10.102.3.4/24 in the second switch
no shutdown
exit
copy r s
```

Procedure 2. Configure Cisco UCS IMM for OpenShift Virtualization


Use the following steps to add the OCP-VM-MGMT VLAN and three vNICs to IMM. Execute these steps from Cisco Intersight.

Step 1. In Cisco Intersight, select **Infrastructure Service > Policies**. Add a Filter of Type VLAN. Select and edit the UCS Domain VLAN policy (for example, AA02-S9108-VLAN). Click **Next**. As you did when building the FlexPod, add the OCP-VM-MGMT VLAN to the policy. Click **Save** to save the policy.

Edit

Add VLANs

Add VLANs to the policy

 VLANs should have one Multicast policy associated to it

Configuration

Prefix * ⓘ

OCP-VM-MGMT


VLAN IDs * ⓘ

1023

Auto Allow On Uplinks ⓘ

Enable VLAN Sharing ⓘ

Multicast Policy *

Selected Policy / ⓘ [Edit Selection](#) 

Step 2. In Cisco Intersight, select **Infrastructure Service > Templates > vNIC Templates**. Click **Create vNIC Template**. Select the correct Organization (for example, AA02-OCP) and name the template OCP-VM-MGMT. Click **Next**.

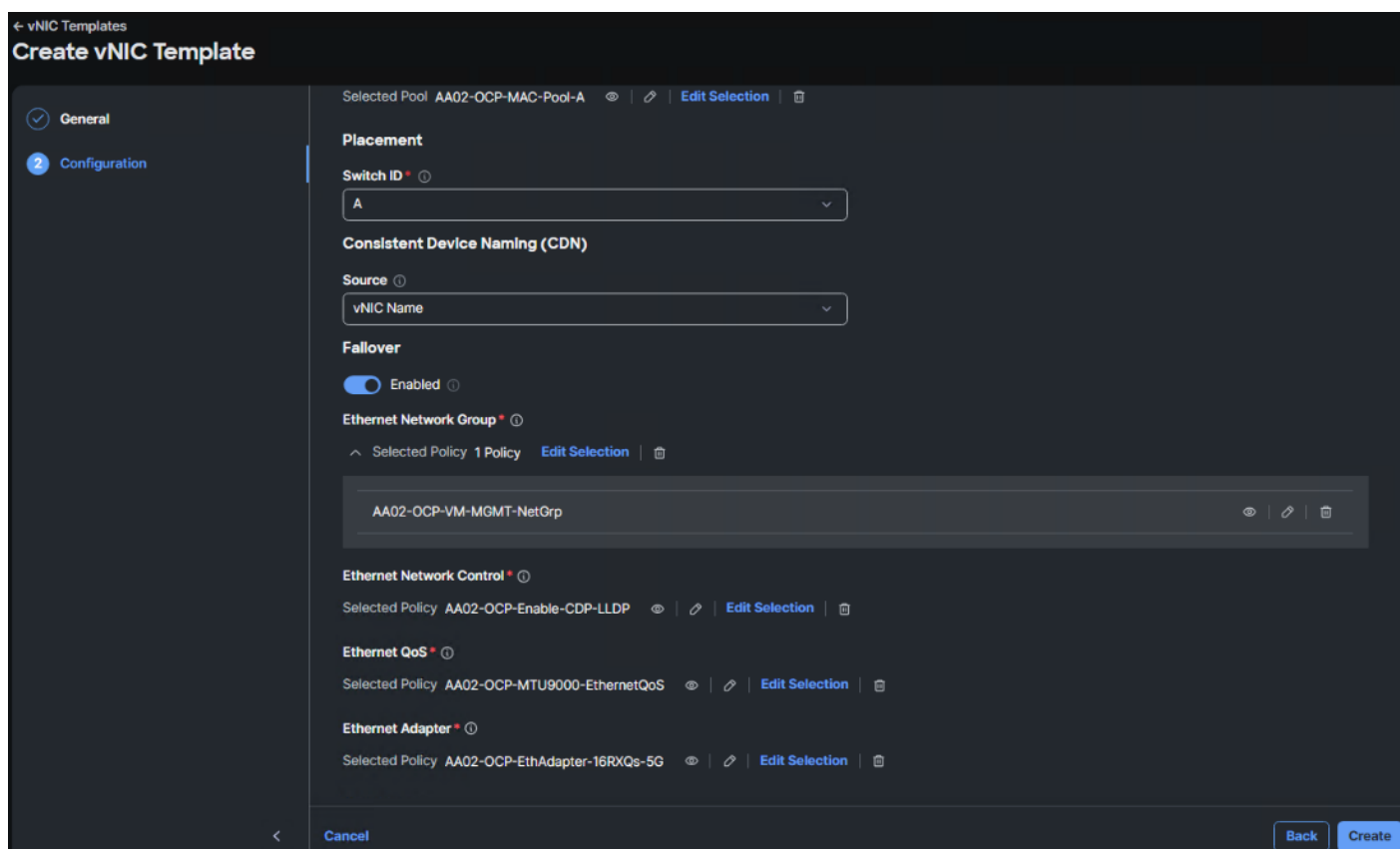
Step 3. The interface will be placed on Fabric A with Fabric Failover enabled. Under MAC Pool, select MAC-Pool-A. Leave Switch ID set to A. Leave the CDN Source set to vNIC Name. Enable Failover.

Step 4. Under Ethernet Network Group, click Select Policies and then click Create New. Make sure the correct Organization is selected and name the policy (for example, AA02-OCP-VM-MGMT-NetGrp). Click **Next**.

Step 5. Click **Add VLANs** then select **Enter Manually**. VLANs can be entered in a comma-separated list. Enter all VLANs that you want to attach VMs to with the exception of the iSCSI and NVMe-TCP VLANs. Click **Enter**. Do not set a Native VLAN in this policy. When all VLANs have been added, click **Create** to create the policy.

Note: All VLANs entered here should also be entered in the FI VLAN policy above and on the Nexus switch ports connected to the FI uplink ports.




Step 6. Select the policy just created and click **Select**. Under Ethernet Network Control, click **Select Policy** and select the already created policy (for example, AA02-OCP-Enable-CDP-LLDP). Under Ethernet QoS, click **Select Policy** and select the MTU9000 policy. Under Ethernet Adapter, click **Select Policy** and select the appropriate 16RXQs policy. Do not select a policy under iSCSI Boot. After all policies have been selected, click **Create** to create the vNIC Template.



Step 7. In Cisco Intersight, select **Infrastructure Service > Policies**. Add a Filter of Type **LAN Connectivity**. Select and edit the appropriate Worker LAN Connectivity Policy for worker nodes running OpenShift Virtualization. Click **Next**. Use the **Add** pulldown to add a **vNIC from Template**. Name the vNIC eno9 and click **Select vNIC Template**. Select the OCP-VM-MGMT vNIC Template created above and click **Select**. Enter 4 for PCI Order. Click **Add**.

Add vNIC from Template



Name * ⓘ

vNIC Template ⓘ
 Selected vNIC Template OCP-VM-MGMT  |  | [Edit Selection](#) | 

Pin Group Name ⓘ

MAC

Pool Static

Mac Pool * ⓘ
 Selected Pool AA02-OCP-MAC-Pool-A  | 

Placement

Simple Advanced

i When Simple Placement is selected, the Slot ID and PCI Link are automatically determined by the system. vNICs are deployed on the first VIC. The Slot ID determines the first VIC. Slot ID numbering begins with MLOM, and thereafter it keeps incrementing by 1, starting from 1.

Switch ID * ⓘ

PCI Order ⓘ

>= 0

Step 8. Add vNIC eno10 with the same “OCP-iSCSI-NVMe-TCP-A-vNIC” template used for eno6 and PCI Order 5. Also, add vNIC eno11 with the same “OCP-iSCSI-NVMe-TCP-B-vNIC” template used for eno7 and PCI Order 6. When all 3 vNICs are added, click **Save & Deploy** followed by **Save & Proceed**.

vNIC Configuration

Manual vNICs Placement | Auto vNICs Placement

Add ▾ Graphic vNICs Editor

🗑️ ✎ 🗑️ Filters 7 results Export

<input type="checkbox"/>	Name	Slot ID	Switch...	PCI Order	Failover	Pin Gro...	MAC P...	vNIC Temp...	Ter	⚙️
<input type="checkbox"/>	eno7	Auto	B	2	Disabled	-	AA02-OCP-...	OCP-ISCSI-...	OK	⋮
<input type="checkbox"/>	eno8	Auto	B	3	Enabled	-	AA02-OCP-...	OCP-NFS-v...	OK	⋮
<input type="checkbox"/>	eno11	Auto	B	6	Disabled	-	AA02-OCP-...	OCP-ISCSI-...	OK	⋮
<input type="checkbox"/>	eno5	Auto	A	0	Enabled	-	AA02-OCP-...	OCP-BareM...	OK	⋮
<input type="checkbox"/>	eno6	Auto	A	1	Disabled	-	AA02-OCP-...	OCP-ISCSI-...	OK	⋮
<input type="checkbox"/>	eno9	Auto	A	4	Enabled	-	AA02-OCP-...	OCP-VM-N...	OK	⋮
<input type="checkbox"/>	eno10	Auto	A	5	Disabled	-	AA02-OCP-...	OCP-ISCSI-...	OK	⋮

Step 9. Adding vNICs will require a server reboot, enable **Reboot Immediately to Activate** and click **Deploy**. Wait for all servers to reboot successfully.

Deploy OpenShift Virtualization

Procedure 1. Deploy the OpenShift Virtualization Operator

Step 1. In the OpenShift console, select **Operators > OperatorHub**. In the search box, type **Virtualization**. Select OpenShift Virtualization provided by Red Hat.

Step 2. Click **Install**. Click **Install** again to deploy OpenShift Virtualization in the openshift-cnv namespace.

Step 3. Once the operator is installed, click **Create Hyperconverged**. Scroll down and click **Create**.

Step 4. Refresh the web console as required and wait for the kubevirt-hyperconverged status to become Conditions: ReconcileComplete, Available, Upgradeable.

Installed Operators > Operator details



OpenShift Virtualization
4.16.3 provided by Red Hat

Actions ▾

Details | YAML | Subscription | Events | All instances | OpenShift Virtualization Deployment | HostPathProvisioner Deployment

HyperConvergeds

Create HyperConverged

Name ▾ /

Name	Kind	Status	Labels	Last updated
kubevirt-hyperconverged	HyperConverged	Conditions: ReconcileComplete, Available, Upgradeable	app=kubevirt-hyperconverged	Nov 13, 2024, 2:00 PM

Step 5. Connect and log into the RHEL Activation Keys page in the Red Hat Hybrid Cloud Console at <https://console.redhat.com/insights/connector/activation-keys#SIDs=&tags=>. If an appropriate Activation Key is not present and if your user permissions allow, use Create activation key to create a RHEL Activation Key for OpenShift Virtualization automatic subscription activation for RHEL VMs. Note the Key Name and Organization ID.

Step 6. In the OpenShift Console, select **Virtualization > Overview > Settings**. Expand **Guest management** and then expand **Automatic subscription of new RHEL VirtualMachines**. Fill in the **Activation key** and **Organization ID** and click **Apply**. Optionally, turn on **Enable auto updates for RHEL VirtualMachines** to enable automatic updates on all RHEL VMs and turn on **Enable guest system log access** to enable access to the VM guest system logs.

Procedure 2. Configure VM Network Connectivity

In this lab validation, virtual machines are connected to the network with NMState NodeNetworkConfigurationPolicy (NNCP) bridges and NetworkAttachmentDefinitions (NAD). A bridge interface can be created on a network interface (bond or NIC) that supports multiple allowed VLANs. An NAD can then be created with a VLAN tag and the MTU can be specified. The VM NIC is then connected to the NAD. Earlier, 3 vNICs (one for VM front-end networks and 2 for in-guest iSCSI and NVMe-TCP) were added to the worker nodes. Three NNCP bridges will be created, and NADs will be created on top of these. NADs created in the default namespace are available for VMs in any namespace. NADs can also be created in individual namespaces and are only available for VMs in that namespace.

Step 1. On the OpenShift installer VM, in the NMState directory, create the following YAML files:

```
cat vm-network-bridge.yaml
apiVersion: nmstate.io/v1
kind: NodeNetworkConfigurationPolicy
metadata:
  name: br-vm-network-policy
spec:
  nodeSelector:
    net-type: fi-attached
  desiredState:
    interfaces:
      - name: br-vm-network
        description: Linux bridge with eno9 as a port
        type: linux-bridge
        state: up
        ipv4:
          enabled: false
        ipv6:
          enabled: false
        bridge:
          options:
            stp:
              enabled: false
        port:
          - name: eno9

cat iscsi-nvme-a-bridge.yaml
apiVersion: nmstate.io/v1
kind: NodeNetworkConfigurationPolicy
metadata:
  name: br-vm-san-a-policy
spec:
  nodeSelector:
    node-role.kubernetes.io/worker: ''
  desiredState:
    interfaces:
      - name: br-vm-san-a
        description: Linux bridge with eno10 as a port
        type: linux-bridge
        state: up
        ipv4:
          enabled: false
        ipv6:
          enabled: false
```

```

    bridge:
      options:
        stp:
          enabled: false
      port:
        - name: eno10

cat iscsi-nvme-b-bridge.yaml
apiVersion: nmstate.io/v1
kind: NodeNetworkConfigurationPolicy
metadata:
  name: br-vm-san-b-policy
spec:
  nodeSelector:
    node-role.kubernetes.io/worker: ''
  desiredState:
    interfaces:
      - name: br-vm-san-b
        description: Linux bridge with eno11 as a port
        type: linux-bridge
        state: up
        ipv4:
          enabled: false
        ipv6:
          enabled: false
        bridge:
          options:
            stp:
              enabled: false
          port:
            - name: eno11

```

Step 2. Create the bridge interfaces:

```

oc create -f vm-network-bridge.yaml
oc create -f iscsi-nvme-a-bridge.yaml
oc create -f iscsi-nvme-b-bridge.yaml

```

Step 3. In the OpenShift Console, select **Networking > NodeNetworkConfigurationPolicy**. The three bridges should now be listed and should be pushed out to the worker nodes.

NodeNetworkConfigurationPolicy

Create ▾

Name ↑	Matched nodes	Enactment states
NNCP br-vm-network-policy	3 nodes	✓ 3 Available
NNCP br-vm-san-a-policy	3 nodes	✓ 3 Available
NNCP br-vm-san-b-policy	3 nodes	✓ 3 Available
NNCP ocp-iscsi-a-policy	3 nodes	✓ 3 Available
NNCP ocp-iscsi-b-policy	3 nodes	✓ 3 Available
NNCP ocp-nfs-policy	3 nodes	✓ 3 Available
NNCP ocp-nvme-tcp-a-policy	3 nodes	✓ 3 Available
NNCP ocp-nvme-tcp-b-policy	3 nodes	✓ 3 Available

Step 4. Select **Networking > NetworkAttachmentDefinitions**. At the top of the page, select the default Project. On the right, click **Create Network Attachment Definition**. In our example, we are creating a VM network

attachment on VLAN 1023. The Name is **vm-network-vlan-1023**. For Network Type, select **Linux bridge**. For Bridge name, enter **br-vm-network**. For VLAN tag number, enter **1023**. Uncheck the MAC spoof check checkbox. Click **Create**.

Project: default ▼

Create network attachment definition

[Edit YAML](#)

Name * ?

vm-network-vlan-1023

Description

Network Type *

Linux bridge ▼

Bridge name *

br-vm-network

VLAN tag number ?

1023

MAC spoof check

Create

Cancel

Step 5. As required, create Network Attachment Definitions for iSCSI A and B and for NVMe-TCP A and B. For the iSCSI definitions, do not specify a VLAN since the iSCSI VLANs are the native VLANs for those vNICs. For each of the in-guest iSCSI and NVMe-TCP interfaces, select the NAD and in the Network Attachment Definition details, select the YAML tab. In the YAML tab, on the last config line, add **“mtu”:9000**, as shown in the screenshot. Click **Save**.

Project: default ▾

Network Attachment Definitions > Network Attachment Definition details

NAD vm-ocp-iscsi-a

Actions ▾

Details YAML

Alt + F1 Accessibility help View shortcuts Show tooltips View sidebar

```
1 apiVersion: k8s.cni.cncf.io/v1
2 kind: NetworkAttachmentDefinition
3 metadata:
4   annotations:
5     k8s.v1.cni.cncf.io/resourceName: bridge.network.kubevirt.io/br-vm-san-a
6   creationTimestamp: '2024-11-04T19:45:34Z'
7   generation: 3
8   managedFields: ...
22  name: vm-ocp-iscsi-a
23  namespace: default
24  resourceVersion: '6136820'
25  uid: 14b64f39-c3b7-48ab-a896-c04ee81784e6
26 spec:
27   config: '{"name":"vm-ocp-iscsi-a","type":"bridge","cniVersion":"0.3.1","bridge":"br-vm-san-a","mtu":9000,"macspoofchk":false,"ipam":{},"preserveDe
28
```

Step 6. Add any additional need VM Network Attachment Definitions.

Project: default ▾

Network Attachment Definitions

Create Network Attachment Definition

Name ▾ Search by name... /

Name	Namespace	Type
NAD vm-network-vlan-1023	NS default	bridge
NAD vm-ocp-iscsi-a	NS default	bridge
NAD vm-ocp-iscsi-b	NS default	bridge
NAD vm-ocp-nvme-tcp-a	NS default	bridge
NAD vm-ocp-nvme-tcp-b	NS default	bridge

Procedure 3. Create a RHEL VM

For Red Hat Enterprise Linux (RHEL) 8 and 9, Fedora, and CentOS Stream 9 VMs, a VM disk image is already loaded by default. In the OpenShift Console, check **Virtualization > Overview > Settings > General settings > Automatic images download** to verify that these images are downloaded automatically. Red Hat recommends not installing VMs in the openshift-cnv namespace. VMs can be created in the default namespace or in any other namespace. Namespaces can be used to group and separate VMs, and specific Network Attachment Definitions can be created within a namespace and only used by VMs within that namespace.

Note: The default VM images are stored as Volume Snapshots in the openshift-virtualization-os-images namespace or project. A Volume Snapshot Class must be defined for these images to be downloaded.

Step 1. In the OpenShift Console, select **Virtualization > VirtualMachines**. Select Project: **default** or use any other custom project or namespace where you want to place the VM. If using a custom project or namespace, create any specific Network Attachment Definitions within that namespace, remembering that the VLANs used in

those Network Attachment Definitions need to be configured in the Cisco Nexus switches, in the Cisco UCS Fabric Interconnects VLAN policy, and in the vNIC Ethernet Network Group policy.

Step 2. Click the **Create VirtualMachine** drop-down list and select **From template**. Click **Red Hat Enterprise Linux 9 VM**.

Step 3. Do not select Boot from CD. Also, leave Disk source set to Template default. Adjust the Disk size as necessary. Adjust CPU|Memory as necessary. Click **Optional Parameters** and adjust as necessary. Change the VirtualMachine name and **Customize VirtualMachine**.



Red Hat Enterprise Linux 9 VM

rhel9-server-small



Template info

Operating system

Red Hat Enterprise Linux 9 VM

Workload type

Server (default)

Description

Template for Red Hat Enterprise Linux 9 VM or newer. A PVC with the RHEL disk image must be available.

Documentation

[Refer to documentation](#)

CPU | Memory

2 CPU | 8 GiB Memory

Network interfaces (1)

Name	Network	Type
default	Pod networking	Masquerade

Disks (2)

Name	Drive	Size
rootdisk	Disk	60 GiB
cloudinitdisk	Disk	-

Storage

Boot from CD

Disk source *

Template default

Disk size *

- 60 + GiB

Drivers

Mount Windows drivers disk

[Optional parameters](#)

Quick create VirtualMachine

VirtualMachine name *

rhel9-test-01

Project Public SSH key

default Not configured

Start this VirtualMachine after creation

Quick create VirtualMachine

Customize VirtualMachine

Cancel

Step 4. On the Customize and create VirtualMachine page, select the **Network Interfaces** tab. Click the three dots to the right of the default network interface and select **Edit**. Change the Name of the network interface if desired and using the Network pulldown, select an appropriate Network Attachment Definition. Add any additional network interfaces as needed. Click **Save**.

Edit network interface ×

Name *

Model

Network * ⓘ

> [Advanced](#)

Step 5. Click the **Disks** tab and adjust as necessary.

Step 6. Click the **Scripts** tab. **Edit** Cloud-init, select Script and adjust as necessary (for example, add “ssh_pwauth: True” right after the password line in the script to allow ssh password authentication login). Click **Save** and **Apply**.

Step 7. Once all tabs have been reviewed, click **Create VirtualMachine**.

Step 8. Once the VM has been Provisioned and Started, click the **Console** tab. You can click Guest login credentials to see the default user name and password. This user has sudo privileges. Login and configure the VM. If you configured the automatic subscription key insertion, you can use “sudo dnf” to add and upgrade packages.

Step 9. To clone the VM, from **Virtualization > VirtualMachines > VirtualMachine details**, use the **Actions** pulldown to select **Clone**. Give the VM a new name, which will be configured as the new hostname by CloudInit and select the checkbox for **Start VirtualMachine once created**. Click **Clone**.

Procedure 4. Create a Windows VM

For Window VMs first time creation, a Windows Installation ISO and installation key are needed. Once the first VM for a given version of windows is created, the boot disk can be stored as a boot image to create more VMs in the future. In this example, Windows Server 2022 Standard will be installed.

Step 1. In the OpenShift Console, select **Virtualization > VirtualMachines**. Select Project: **default** or use any other custom project or namespace where you want to place the VM. If using a custom project or namespace, create any specific Network Attachment Definitions within that namespace, remembering that the VLANs used in those Network Attachment Definitions need to be configured in the Cisco Nexus switches, in the Cisco UCS Fabric Interconnects VLAN policy, and in the vNIC Ethernet Network Group policy.

Step 2. Click the **Create VirtualMachine** pulldown and choose **From template**. Click **Microsoft Windows Server 2022 VM**.

Step 3. Select **Boot from CD**. For CD source, from the drop-down list select **Upload (Upload a new file to a PVC)**. Click **Browse** and browse to and **Open** the Windows Server 2022 ISO. Adjust the CD Disk size as necessary to be a little larger than the ISO. Leave Disk source set to Blank and set Disk size to 30GiB. Make sure **Mount Windows drivers disk** is selected. Adjust CPU|Memory as necessary. Click on Optional Parameters and change the DATA_SOURCE_NAME to windows2k22. Change the VirtualMachine name and click **Customize VirtualMachine**.



Microsoft Windows Server 2022 VM



windows2k22-server-medium

Template info

Operating system

Microsoft Windows Server 2022 VM

Workload type

Server (default)

Description

Template for Microsoft Windows Server 2022 VM. A PVC with the Windows disk image must be available.

Documentation

[Refer to documentation](#)

CPU | Memory

2 CPU | 8 GiB Memory

Network interfaces (1)

Name	Network	Type
default	Pod networking	Masquerade

Disks (3)

Name	Drive	Size
rootdisk	Disk	30 GiB
windows-drivers-disk	CD-ROM	-
installation-cdrom	CD-ROM	10 GiB

Storage

Boot from CD

CD source *

Upload (Upload a new file to a PVC)

Upload data *

en-us_windows_server_2022_... Browse... Clear

Disk size *

- 10 + GiB

Disk source *

Blank

Disk size *

- 30 + GiB

Drivers

Mount Windows drivers disk

Optional parameters

DATA_SOURCE_NAME

windows2k22

Name of the DataSource to clone

DATA_SOURCE_NAMESPACE

openshift-virtualization-os-images

Quick create VirtualMachine

VirtualMachine name *

win2k22-test-01

Project Public SSH key
default Not configured

Start this VirtualMachine after creation

Quick create VirtualMachine

Customize VirtualMachine

Cancel

Step 4. If you get an Invalid certificate error, click the URL, click **Advanced** and then click **Proceed** to `cdi-uploadproxy-openshift-cnv.apps.ocp.flexpodb4.cisco.com` (unsafe). When you get to the message “This page isn’t working”, close the window. Click **Customize VirtualMachine** again. Wait a few minutes for the ISO to be uploaded to a PVC.

Step 5. On the Customize and create VirtualMachine page, select the **Network Interfaces** tab. Click the three dots to the right of the default network interface and select **Edit**. Change the Name of the network interface if desired. Also, change the Model to virtio for best network performance. Using the Network pulldown, select an appropriate Network Attachment Definition. Add any additional network interfaces as needed. Click **Save**.

Edit network interface ✕

Name *

Model

Network * ?

Bridge Binding▼

[> Advanced](#)

SaveCancel

Step 6. Click the **Disks** tab and using the three dots to the right of rootdisk, click **Edit**. Change the Interface to virtio. Also, if a Storage Class other than the default storage class is desired for the disk, select the new Storage Class. Click **Save**.

Note: For Windows, if virtio is used for rootdisk, it is critical that the windows-drivers-disk is mounted so that the virtio driver can be loaded during the installation process.

Edit disk



Use this disk as a boot source

Name *

rootdisk

Source *

Empty disk (blank) ▼

PersistentVolumeClaim size *

- 30 + GiB ▼

Type

Disk ▼

Hot plug is enabled only for "Disk" type

Interface *

VirtIO ▼

Hot plug is enabled only for "SCSI" interface

StorageClass

ontap-nfs ▼

Apply optimized StorageProfile settings

Optimized values Access mode: ReadWriteMany, Volume mode: Filesystem.

Enable preallocation

› [Advanced settings](#)

Save

Cancel

- Step 7.** Once all tabs have been reviewed, click **Create VirtualMachine**.
- Step 8.** Once the VM has been Provisioned and Started, click the **Console** tab. Click on the console window and hit a key when you see “Press any key to boot from CD or DVD...”
- Step 9.** Select the appropriate language and click **Next**.
- Step 10.** Click **Install now**.
- Step 11.** Enter an appropriate product key or click “I don’t have a product key” and click **Next**.
- Step 12.** Select “Windows Server 2022 Standard (Desktop Experience)” or the appropriate version and click **Next**.
- Step 13.** Click the Accept checkbox and click **Next**.
- Step 14.** Click “Custom:...”.
- Step 15.** If you selected the virtio rootdisk, click **Load driver**. Click **Browse**. Expand the **virtio-win CD Drive**. Expand **amd64** and select the **2k22** folder. Click **OK**. Leave “Red Hat VirtIO SCSI controller” selected and click **Next**.
- Step 16.** Make sure **Drive 0 Unallocated Space** is selected and click **Next**.
- Step 17.** The Windows installation will complete and do not hit a key to boot from CD or DVD when the VM reboots. The VM will reboot more than once.
- Step 18.** Enter and confirm an Administrator password and click **Finish**.
- Step 19.** Use the Send key pulldown to send **Ctrl + Alt + Delete** the VM. Enter the Administrator password to login.
- Step 20.** Close the Server Manager. Use the Folder icon in the Task Bar to open **File Explorer**. Select **CD Drive virtio-win** on the left. Scroll down and double-click **virtio-win-guest-tools**. Select the checkbox to agree to the license terms and conditions and click **Install**. Complete the installation. Click **Restart** to reboot the VM.
- Step 21.** Log back into the VM and if necessary configure the network interface. Type update in the search box, install all Windows updates, and restart the VM.
- Step 22.** Log back into the VM and make any other adjustments, such as setting the VM timezone. Then, use the VirtualMachine details > Actions pulldown to Stop the VM.
- Step 23.** Select the **Configuration** tab. On the left, select **Storage**. Uncheck “Mount Windows drivers disk”. Using the three dots to the right of the installation-cdrom, select **Detach**. Choose whether to delete the installation-cdrom PVC (it can be used for other installations) and click **Detach**. The rootdisk should now be the VM’s only disk and it should show bootable.
- Step 24.** Using the Actions pulldown, Start the VM. Select the Console tab. Log into the VM. Open File Explorer and navigate to This PC\C:\Windows\System32\Sysprep. Double-click sysprep. Change the Shutdown Options to Shutdown and click OK. When the VM begins to Start, use the Actions pulldown to Stop it.
- Step 25.** To create a Windows Server 2022 Standard Bootable volume, select **Virtualization > Bootable volumes**. Switch to the openshift-virtualization-os-images Project. Use the **Add volume** pulldown to select **With form**. For Source type, select **Use existing volume**. For PVC project, select **default** or the project where the Windows Server 2022 VM is located. For PVC name, select the PVC for the Windows Server 2022 VM rootdisk. Select the appropriate StorageClass. Do not change the disk size. For volume name, enter **windows2k22**. For Destination project, select **openshift-virtualization-os-images**. For Preference, select windows.2k22.virtio. For DefaultInstanceType, select **Red Hat provided > U series > large: 2 CPUs, 8 GiB Memory** or an appropriate DefaultInstanceType. Click **Save**.

Add volume ✕

Upload a new volume, or use an existing PersistentVolumeClaim (PVC), VolumeSnapshot or DataSource.

Source type

Use existing volume ▼

Source details

PVC project *

PR default ▼

Location of the existing PVC

PVC name *

PVC win2k22-test-01 ▼

Clone existing PVC ⓘ

Destination details

StorageClass

SC ontap-nfs ▼

Apply optimized StorageProfile settings

Optimized values Access mode: ReadWriteMany, Volume mode: Filesystem.

Disk size *

− 32 + GiB ▼

Volume name *

windows2k22

Destination project

PR openshift-virtualization-os-images ▼

Volume metadata ⓘ

Preference ⓘ *

VMCP windows.2k22.virtio ▼

Default InstanceType ⓘ

VMCI u1.large ▼

Save

Cancel

Step 26. The windows2k22 Data Source should now appear under Bootable volumes. Wait for the Clone in Progress to complete. When creating a VM from Template, the Microsoft Windows Server 2022 VM template should now show Source available. When using this template the Disk source can be left at Template default without checking Boot from CD. The disk size specified must be larger than the disk size used to create the initial VM. Under Optional parameters, change the DATA_SOURCE_NAME to windows2k22. When Customizing the VM, it will be necessary to go into the YAML tab and change the rootdisk bus: to virtio under spec: and then click Save. Also, make sure to click the Network tab can configure the Network Interface accordingly.

Procedure 5. Install the virtctl Command Line Utility

The virtctl client is a supplemental command-line utility for managing virtualization resources from the command line. For more information, see

https://docs.openshift.com/container-platform/4.16/virt/getting_started/virt-using-the-cli-tools.html.

Step 1. Using Chrome from the OpenShift Installer VM, in the OpenShift Console, select **Virtualization > Overview**. If necessary, close the Welcome to OpenShift Virtualization popup. In the upper right corner of the, click **Download the virtctl command-line utility**. On the Command Line Tools page, click **Download virtctl for Linux for x86_64**.

Step 2. From a terminal window, type the following commands:

```
cd ~/Downloads
tar -xzf virtctl.tar.gz
sudo cp virtctl /usr/local/bin/
rm virtctl*
```

Step 3. Use **virtctl** to restart a VM. Go to **Virtualization > VirtualMachines** in the default project to Monitor from the OpenShift Console.

```
oc project default
oc get vms

NAME                AGE   STATUS   READY
rhel9-test-01       21h   Running  True
rhel9-test-02       21h   Running  True
win2k22-test-01     17h   Running  True
win2k22-test-02     17h   Running  True

virtctl restart rhel9-test-01
```

Procedure 6. Deploy the Migration Toolkit for Virtualization Operator

The Migration Toolkit for Virtualization Operator is used to migrate VMs from VMware and other sources into OpenShift Virtualization and also to migrate VMs between OpenShift clusters.

Step 1. In the OpenShift Console, select **Operators > OperatorHub**. Choose **Migration Toolkit for Virtualization Operator provided by Red Hat**. Click **Install**. Leave all settings at their defaults and click **Install** again.

Step 2. Once the Operator is installed, click **Create ForkliftController**. Click **Create**. Wait for the Status to become “Condition: Running”. Click to **Refresh web console**.

Step 3. On the left, select **Migration > Overview**. The Migration Toolkit for Virtualization has been successfully installed.

Procedure 7. Deploy OpenShift Image Registry

The OpenShift internal Image Registry will be used here to store the VMware VDDK container that accelerates converting and migrating VMware VMs to OpenShift Virtualization VMs. This registry can also be used to locally store any additional customized containers used by applications.

Step 1. The image registry was removed during the OpenShift on Bare Metal installation. Change the Image Registry Operator configuration's managementState from Removed to Managed.

```
oc patch configs.imageregistry.operator.openshift.io cluster --type merge --patch '{"spec":{"managementState":"Managed"}}'
```

Step 2. Modify the registry configuration.

```
oc edit configs.imageregistry.operator.openshift.io
```

Step 3. Change the storage field to the format shown below (leaving the claim field blank) to automatically create a 100GB image-registry-storage PVC in the openshift-image-registry namespace and save the config with “:x” (vim save and exit).

```
storage:
  pvc:
    claim:
```

Step 4. Wait for the registry to become available.

```
oc get clusteroperator image-registry
```

NAME	VERSION	AVAILABLE	PROGRESSING	DEGRADED	SINCE	MESSAGE
image-registry	4.16.18	True	False	False	50s	

Step 5. Ensure that your registry is set to managed to enable building and pushing of images.

```
oc describe configs.imageregistry/cluster | grep "Management State"

Management State:    Managed
Management State:    Managed
Management State:    Managed
```

Step 6. To expose the registry using DefaultRoute:

```
oc patch configs.imageregistry.operator.openshift.io/cluster --patch '{"spec":{"defaultRoute":true}}' --type=merge
```

Procedure 8. Add VMware VDDK Container to Image Registry

Step 1. On your OpenShift installer VM, create and navigate to a temporary directory.

```
mkdir /tmp/vddk && cd /tmp/vddk
```

Step 2. Using Chrome, navigate to the [VMware VDDK version 8 download page](#). Select **Login** and log into the site. Select version 8.0.1 for Linux (or the version that matches your vCenter) and click **Download**.

Note: A Broadcom VMware login will be needed to download the VDDK. Also, we are assuming we are migrating from vSphere 8.0 Update 1 or later.

Step 3. Move the downloaded file to the temporary directory.

```
mv ~/Downloads/VMware-vix-disklib-8.0.1-21562716.x86_64.tar.gz /tmp/vddk/
ls
VMware-vix-disklib-8.0.1-21562716.x86_64.tar.gz
```

Step 4. Extract the VDDK archive:

```
tar -xzf VMware-vix-disklib-8.0.1-21562716.x86_64.tar.gz
```

Step 5. Create a Dockerfile:


```
cat > Dockerfile <<EOF
FROM registry.access.redhat.com/ubi8/ubi-minimal
USER 1001
COPY vmware-vix-disklib-distrib /vmware-vix-disklib-distrib
RUN mkdir -p /opt
ENTRYPOINT ["cp", "-r", "/vmware-vix-disklib-distrib", "/opt"]
EOF
```

Step 6. Get the exposed registry URL.

```
HOST=$(oc get route default-route -n openshift-image-registry --template='{{ .spec.host }}')
echo $HOST
```

```
default-route-openshift-image-registry.apps.ocp.flexpodb4.cisco.com
```

Step 7. Login to the registry with podman after logging into the cluster with a cluster-admin user.

```
oc login -u admin -p <password>
podman login -u admin -p $(oc whoami -t) --tls-verify=false $HOST
```

Step 8. Build and tag the VDDK image.

```
podman build . -t $HOST/openshift/vddk:8.0.1
```

```
Successfully tagged default-route-openshift-image-registry.apps.ocp.flexpodb4.cisco.com/openshift/vddk:8.0.1
```

Step 9. Push the image to the image registry.

```
podman push $HOST/openshift/vddk:8.0.1 --tls-verify=false
```

```
Getting image source signatures
Copying blob 12d5399f6d50 done |
Copying blob 325e71f5b538 done |
Copying blob 7e02eaad2ba1 done |
Copying config 120bbf6b2d done |
Writing manifest to image destination
```

Step 10. Restore the default oc login.

```
cp <path>/kubeconfig ~/.kube/config
```

Procedure 9. Add a VMware Migration Provider

Use the following steps to add a VMware Migration Provider to the Migration Providers for virtualization.

Step 1. In the OpenShift Console, select **Migration > Providers for virtualization**. Switch to the openshift-mtv project. Click **Create Provider**.

Step 2. Select **vSphere**. For Provider resource name, enter your vCenter hostname with all lowercase characters. Select the vCenter Endpoint type. For URL, enter <https://<vcenter-fqdn>/sdk>. For VDDK init image, the VDDK can be accessed internally in the cluster. Enter **image-registry.openshift-image-registry.svc:5000/openshift/vddk:8.0.1**. For Provider credentials, use administrator@vsphere.local and the corresponding password. Select to **Skip certificate validation**. Click **Create provider**.

Step 3. For VM migrations, it is best to have a direct connection in the same subnet between the VMware cluster and the OpenShift cluster. It is recommended to pipe the VLAN used for OpenShift Bare Metal / Management into the VMware cluster and configure a port group with this VLAN on vSwitch0 on each ESXi host. Then create a VMkernel port with the Management service on each ESXi host in the OpenShift Bare Metal / Management subnet with no default gateway. It is important that the Network Label for the VMkernel port is the same on all ESXi hosts. The following screenshot shows this VMkernel port on one ESXi host.

aa02-esxi-01.flexpodb4.cisco.com | ACTIONS

Summary Monitor **Configure** Permissions VMs Datastores Networks Updates

Storage

- Storage Adapters
- Storage Devices
- Host Cache Configuration
- Protocol Endpoints
- I/O Filters

Networking

- Virtual switches
- VMkernel adapters**
- Physical adapters
- TCP/IP configuration

VMkernel adapters

ADD NETWORKING... REFRESH

	Device	Network Label	Switch	IP Address	TCP/IP Stack	Enabled Services
⋮ >>	vmk0	Management Network	vSwitch0	10.102.1.101	Default	Management
⋮ >>	vmk1	AA02-VMW-NFS-A	vDS0	192.168.50.101	Default	--
⋮ >>	vmk2	AA02-VMW-NFS-B	vDS0	192.168.60.101	Default	--
⋮ >>	vmk3	AA02-VMW-vMotion	vDS0	192.168.0.101	Default	vMotion +1
⋮ >>	vmk4	VMkernel-OCP-Baremetal-MGMT	vSwitch0	10.102.2.101	Default	Management

Step 4. In the OpenShift Console, select **Migration > Providers for virtualization**. Select the vCenter provider to bring up the Provider Details page. In the tab bar, select **Hosts**. Using the checkboxes, select each ESXi host and then click **Select migration network**. Use the pulldown to select VMkernel portgroup created in the previous step. Enter root for the ESXi host admin username and the corresponding password. Click **Save**.

Select migration network

You can select a migration network for a source provider to reduce risk to the source environment and to improve performance.

3 hosts selected.

Network *

VMkernel-OCP-Baremetal-MGMT - 10.102.2.101/24 ✕ ▾

ESXi host admin username *

root ✓

The username for the ESXi host admin

ESXi host admin password *

..... ✓ 👁

The password for the ESXi host admin

Save **Cancel**

Step 5. Repeat this process for all ESXi hosts.

Project: openshift-mtv ▾

Providers > Provider Details

PR aa02-vc Ready

Actions ▾

Details YAML Credentials Virtual Machines Hosts

Hosts

Name ▾	Filter by name	→	☰	Select migration network			
<input type="checkbox"/>	Name ↑			Network for data transfer ↓		Bandwidth ↓	MTU ↓
<input type="checkbox"/>	aa02-esxi-01.flexpodb4.cisco.com ↗			VMkernel-OCF-Baremetal-MGMT - 10.102.2.101/24		100000 Mbps	1500
<input type="checkbox"/>	aa02-esxi-02.flexpodb4.cisco.com ↗			VMkernel-OCF-Baremetal-MGMT - 10.102.2.102/24		50000 Mbps	1500
<input type="checkbox"/>	aa02-esxi-03.flexpodb4.cisco.com ↗			VMkernel-OCF-Baremetal-MGMT - 10.102.2.103/24		50000 Mbps	1500

Procedure 10. Migrating VMware VMs to OpenShift Virtualization

Now that the Migration Toolkit for Virtualization Operator has been fully set up, VMware VMs can be migrated to any namespace within the OpenShift cluster.

- Step 1.** If a new namespace is required, create the namespace. Go to **Project > Create Project** in the OpenShift Console. If Network Attachment Definitions are needed within the namespace, create them within the namespace or Project.
- Step 2.** VM Migrations must occur within the openshift-mtv namespace. Select **Project > openshift-mtv** in the OpenShift Console. Then choose **Migration > Plans for virtualization**. In the upper right, click **Create Plan**.
- Step 3.** Select the vCenter as the source provider. Then, select any supported Windows or Linux VMs and click **Next**.
- Step 4.** Give the plan a name, such as migrate-**<vm name>**. Leave host as the Target provider. Select the appropriate Target namespace. Select the appropriate Target network under Network mappings and Target storage (Storage Class) under Storage mappings. Click **Create migration plan**.
- Step 5.** It will take a few minutes for the migration plan to get to a Status of Ready. Once the Status is Ready, click **Start migration**. Note that any running VMs will be shut down. Click **Start**.
- Step 6.** To monitor the status of the VM migration, click **0 of X VMs migrated**. Then, next to any VM, click **>** to monitor the migration status. Depending on the VM disk number and size, the migration will take some time.
- Step 7.** Once the migration has Succeeded, select **Virtualization > VirtualMachines**, then select the project you migrate the VM(s) into. Select a VM that was migrated. Select the **Configuration** tab and then select **Network**. The migration preserved the VMware MAC address. If you want to use an OpenShift MAC address, note the values of each field, then click the three dots to the right and choose **Delete** followed by **Delete** again. Then click **Add network interface**. Select or fill in the noted values and click **Save**. Now the VM should have an OpenShift MAC address.
- Step 8.** Select **Actions > Start** to power on the VM. Select the Console tab to interact with the VM. The VM networking will need to be set up for the VM in OpenShift if DHCP is not being used.

Procedure 11. Migrating an Openshift VM to a Different Namespace

The virtctl export command can be used to export a VM disk and VM configuration, and then the disk can be exported into a different namespace along with a modified VM configuration. This method can also be used to migrate VMs between OpenShift clusters and for VM Backup and Restore.

Step 1. A storage location will need to be set up that is large enough to hold large numbers of VM disks. It is recommended to use the OpenShift Installer VM for this since it already contains configurations to access the OpenShift cluster. One way to set up a large storage location is to create a volume on the NetApp storage and mount it with NFS from this VM. You will need to create a Network Attachment Definition (NAD) with the NFS VLAN ID and MTU 9000 to connect to the VM. Once the NFS mount is setup, create a directory for each VM and switch into this directory.

Step 2. For the VM to export, if you want to leave it running, in the OpenShift Console select **Virtualization > VirtualMachines**. Then select the VM to export. On the right, click **Take snapshot**. Name the snapshot snapshot-for-export and click **Save**.

Step 3. On the OpenShift Installer VM from the VM backup directory, export the VM disk and VM manifest. Depending on the disk size, this can take some time.

```
oc project <vm-namespace>
virtctl vmexport download <vm-name> --snapshot=snapshot-for-export --output=disk.img.gz
virtctl vmexport download <vm-name> --snapshot=snapshot-for-export --manifest > <vm-name>.yaml
```

Note: If the VM has more than one disk, you will need to add “-volume=<pvc name>” to the disk export command along with a disk number in the -output field and run this command once for each disk.

Step 4. Delete the snapshot from the source VM.

Step 5. Import the VM disk(s) into the new namespace. The disk(s) will be uploaded using the default Storage Class and will create a PVC with the same name as the dv.

```
oc project <new-vm-namespace>
virtctl image-upload dv <vm-name> --image-path=./disk.img.gz --size=<source disk size>Gi -insecure
```

Note: If the VM has more than one disk, each disk will need to be uploaded to a unique PVC or DV.

Step 6. Using a text editor, edit the manifest .yaml file. Change the VM name if necessary and change the namespace to the new namespace name. Under interfaces, delete the macAddress line to set a new MAC address for this VM. If a different NAD is to be used with the new VM, change the networkName field. Next, change the persistentVolumeClaim > claimName to match the dv field in the command above. Finally, delete all the lines in the DataVolume section of the yaml file between the sets of “---” leaving only one “---” at the bottom of the file.

Step 7. Create the new VM in the new namespace.

```
oc create -f <vm-name>.yaml
```

Note: If you get an error message stating that the VM could not be created, run the command a second time and the VM should be successfully created.

Procedure 12. Attaching a GPU to a VM

This procedure can be used to attach PCIe passthrough GPUs to VMs, which will pass an entire datacenter GPU in compute mode to a VM. Although it is possible to define vGPUs, which are subsets of full GPUs with a part of the GPU framebuffer, and pass through to VMs, that is not defined here. For defining and passing through vGPUs to VMs, see

https://docs.openshift.com/container-platform/4.16/virt/virtual_machines/advanced_vm_management/virt-configuring-virtual-gpus.html.

Step 1. Node labeling can be used to designate worker node GPUs to be used for containerized applications or for VMs. To label a node to passthrough its GPUs to VMs, from the OpenShift Installer VM, run the following command:

```
oc label node <node_name> nvidia.com/gpu.deploy.operands=false
```

Step 2. Verify that the label was added to the node:

```
oc describe node <node_name>
```

Step 3. This will cause the NVIDIA GPU operator to unload the NVIDIA GPU driver pods from that node. This can be verified with the following commands. The NVIDIA GPU Operator pods should no longer be running on the labeled node.

```
oc get pods -n nvidia-gpu-operator -o wide
```

Step 4. Create a custom resource (CR) to enable either Intel Virtualization Technology for Directed I/O extensions (IOMMU) or AMD IOMMU on the worker nodes.

```
cat 100-worker-kernel-arg-iommu.yaml
apiVersion: machineconfiguration.openshift.io/v1
kind: MachineConfig
metadata:
  labels:
    machineconfiguration.openshift.io/role: worker
  name: 100-worker-iommu
spec:
  config:
    ignition:
      version: 3.2.0
  kernelArguments:
    - intel_iommu=on
```

Step 5. Create the new MachineConfig object:

```
oc create -f 100-worker-kernel-arg-iommu.yaml
```

Step 6. Verify that the new MachineConfig was created:

```
oc get MachineConfig
```

Step 7. Ssh to the worker node with the passthrough GPU(s) and obtain the vendor-ID and the device-ID for the GPU device. In this case the vendor-ID is 10de and the device-ID is 26b9.

```
ssh core@<worker-baremetal-ip>
lspci -nnv | grep -i nvidia
38:00.0 3D controller [0302]: NVIDIA Corporation AD102GL [L40S] [10de:26b9] (rev a1)
      Subsystem: NVIDIA Corporation Device [10de:1851]
d8:00.0 3D controller [0302]: NVIDIA Corporation AD102GL [L40S] [10de:26b9] (rev a1)
      Subsystem: NVIDIA Corporation Device [10de:1851]
exit
```

Step 8. In your machine-configs directory, create 100-worker-vfiopci.bu to bind the GPU(s) to the VFIO driver and substituting your GPU's vendor-ID and device-ID.

```
cat 100-worker-vfiopci.bu
variant: openshift
version: 4.16.0
metadata:
  name: 100-worker-vfiopci
  labels:
    machineconfiguration.openshift.io/role: worker
storage:
  files:
    - path: /etc/modprobe.d/vfio.conf
      mode: 0644
      overwrite: true
      contents:
```

```
inline: |
  options vfio-pci ids=10de:26b9
- path: /etc/modules-load.d/vfio-pci.conf
mode: 0644
overwrite: true
contents:
  inline: vfio-pci
```

Step 9. Use butane to generate a MachineConfig yaml file.

```
./butane 100-worker-vfiopci.bu -o 100-worker-vfiopci.yaml
```

Step 10. Apply the MachineConfig to the worker nodes to attach any unattached GPUs to the VFIO driver.

```
oc apply -f 100-worker-vfiopci.yaml
```

Step 11. Verify that the MachineConfig object was added.

```
oc get MachineConfig
```

Step 12. Ssh back to the worker node and verify the GPU is bound to the VFIO driver.

```
ssh core@<worker-baremetal-ip>
lspci -nnk -d 10de:
38:00.0 3D controller [0302]: NVIDIA Corporation AD102GL [L40S] [10de:26b9] (rev a1)
  Subsystem: NVIDIA Corporation Device [10de:1851]
  Kernel driver in use: vfio-pci
  Kernel modules: nouveau
d8:00.0 3D controller [0302]: NVIDIA Corporation AD102GL [L40S] [10de:26b9] (rev a1)
  Subsystem: NVIDIA Corporation Device [10de:1851]
  Kernel driver in use: vfio-pci
  Kernel modules: nouveau
```

Step 13. To attach a GPU to a VM, in the OpenShift Console select **Virtualization > VirtualMachines**. Then select the appropriate project. Select the VM and then select the **Configuration** tab. Select **GPU devices**. Give the GPU a name then select an available GPU under Device name. You can use the Add GPU device link to add multiple GPUs to a VM. Click **Save** to complete adding the GPU to the VM.

GPU devices



 Restart the VirtualMachine to apply changes.

Enter a name for the device to be assigned and select it from the dropdown menu.

Click **Save**.

Click **+ Add GPU device** to add another devices.

Name

Device name *



 Add GPU device

PCI host devices

nvidia.com/NVIDIA-L40S ✓

Save

Cancel

Step 14. You will need to restart the VM to use the GPU. You will also need to install GPU guest drivers in the VM to use the GPU. In this validation, an NVIDIA L40S GPU was connected to a Windows 11 VM, NVIDIA GPU drivers installed, and the [sdbds/InstantID-for-windows: InstantID : Zero-shot Identity-Preserving Generation in Seconds](#) application was installed and run to test the GPU.

About the Authors

John George, Technical Marketing Engineer, Cisco Systems, Inc.

John has been involved in designing, developing, validating, and supporting the FlexPod Converged Infrastructure since it was developed more than 13 years ago. Before his role with FlexPod, he supported and administered a large worldwide training network and VPN infrastructure. John holds a master's degree in Computer Engineering from Clemson University.

Kamini Singh, Technical Marketing Engineer, Hybrid Cloud Infra & OEM Solutions, NetApp

Kamini Singh is a Technical Marketing engineer at NetApp. She has more than five years of experience in data center infrastructure solutions. Kamini focuses on FlexPod hybrid cloud infrastructure solution design, implementation, validation, automation, and sales enablement. Kamini holds a bachelor's degree in Electronics and Communication and a master's degree in Communication Systems.

Acknowledgements

For their support and contribution to the design, validation, and creation of this Cisco Validated Design, the authors would like to thank:

- Archana Sharma, Technical Marketing Engineer, Cisco Systems, Inc.
- Paniraja Koppa, Technical Marketing Engineer, Cisco Systems, Inc.

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

ALL DESIGNS, SPECIFICATIONS, STATEMENTS, INFORMATION, AND RECOMMENDATIONS (COLLECTIVELY, "DESIGNS") IN THIS MANUAL ARE PRESENTED "AS IS," WITH ALL FAULTS. CISCO AND ITS SUPPLIERS DISCLAIM ALL WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE. IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THE DESIGNS, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

THE DESIGNS ARE SUBJECT TO CHANGE WITHOUT NOTICE. USERS ARE SOLELY RESPONSIBLE FOR THEIR APPLICATION OF THE DESIGNS. THE DESIGNS DO NOT CONSTITUTE THE TECHNICAL OR OTHER PROFESSIONAL ADVICE OF CISCO, ITS SUPPLIERS OR PARTNERS. USERS SHOULD CONSULT THEIR OWN TECHNICAL ADVISORS BEFORE IMPLEMENTING THE DESIGNS. RESULTS MAY VARY DEPENDING ON FACTORS NOT TESTED BY CISCO.

CCDE, CCENT, Cisco Eos, Cisco Lumin, Cisco Nexus, Cisco StadiumVision, Cisco TelePresence, Cisco WebEx, the Cisco logo, DCE, and Welcome to the Human Network are trademarks; Changing the Way We Work, Live, Play, and Learn and Cisco Store are service marks; and Access Registrar, Aironet, AsyncOS, Bringing the Meeting To You, Catalyst, CCDA, CCDP, CCIE, CCIP, CCNA, CCNP, CCSP, CCVP, Cisco, the Cisco Certified Internetwork Expert logo, Cisco IOS, Cisco Press, Cisco Systems, Cisco Systems Capital, the Cisco Systems logo, Cisco Unified Computing System (Cisco UCS), Cisco UCS B-Series Blade Servers, Cisco UCS C-Series Rack Servers, Cisco UCS S-Series Storage Servers, Cisco UCS Manager, Cisco UCS X-Series, Cisco UCS Management Software, Cisco Unified Fabric, Cisco Application Centric Infrastructure, Cisco Nexus 9000 Series, Cisco Nexus 7000 Series, Cisco Prime Data Center Network Manager, Cisco NX-OS Software, Cisco MDS Series, Cisco Unity, Collaboration Without Limitation, EtherFast, EtherSwitch, Event Center, Fast Step, Follow Me Browsing, FormShare, GigaDrive, HomeLink, Internet Quotient, IOS, iPhone, iQuick Study, LightStream, Linksys, MediaTone, MeetingPlace, MeetingPlace Chime Sound, MGX, Networkers, Networking Academy, Network Registrar, PCNow, PIX, PowerPanels, ProConnect, ScriptShare, SenderBase, SMARTnet, Spectrum Expert, SE1tackWise, The Fastest Way to Increase Your Internet Quotient, TransPath, WebEx, and the WebEx logo are registered trademarks of Cisco Systems, Inc. and/or its affiliates in the United States and certain other countries. (LDW_P1)

All other trademarks mentioned in this document or website are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (0809R)

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>.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: <https://www.cisco.com/go/trademarks>. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1110R)