

**Use these Red Hat OpenShift deployment options for Hitachi Unified Compute Platform in a bare metal, virtualized, or mixed bare metal environment with virtualized OpenShift.**

---

ARCHITECTURE GUIDE

# **Deployment Options for Red Hat OpenShift on Hitachi Unified Compute Platform CI with Hitachi Virtual Storage Platform Family**

## **Reference Architecture Guide**

By Jeff Chen (Hitachi Vantara), Henry Chu (Hitachi Vantara), Tim Darnell (Hitachi Vantara), Mayur Shetty (Red Hat)

# Feedback

Hitachi Vantara welcomes your feedback. Please share your thoughts by sending an email message to [SolutionLab@hitachivantara.com](mailto:SolutionLab@hitachivantara.com). To assist the routing of this message, use the paper number in the subject and the title of this white paper in the text.

## Revision History

Revision	Changes	Date
MK-SL-210-00	Initial release	October 6, 2020
MK-SL-210-01	Update cover, authors	October 9, 2020

# Table of Contents

<b>Solution Overview</b>	<b>2</b>
Bare Metal Container Hosts	4
Virtualized Container Hosts	5
Mixed Hybrid Container Hosts	6
<b>Solution Components</b>	<b>8</b>
Hardware Components	8
Software Components	9
<b>Solution Design</b>	<b>10</b>
Deploy a OpenShift Container Platform Cluster on Bare Metal Hosts	12
Deploy a Red Hat OpenShift Container Platform Cluster on VMware vSphere	15
Deploy a Red Hat OpenShift Container Platform Cluster on Mixed Hybrid Environments (Virtualized and Bare Metal Hosts)	20
<b>Solution Validation</b>	<b>24</b>
Create Persistent Volumes for Red Hat OpenShift Container Platform Bare Metal Hosts	24
Create Persistent Volumes using VMware vVols	27
Create Persistent Volumes from a VMFS Datastore	31
<b>Product Descriptions</b>	<b>36</b>
Hardware Components	36
Software Components	39
<b>VMware vSphere CPI/CSI Setup Files for Red Hat OpenShift Container Platform 4</b>	<b>41</b>

# Deployment Options for Red Hat OpenShift on Hitachi Unified Compute Platform CI with Hitachi Virtual Storage Platform Family

## Reference Architecture Guide

Use these Red Hat OpenShift deployment options for Hitachi Unified Compute Platform in a bare metal, virtualized, or mixed bare metal environment with virtualized OpenShift.

Your business is under pressure to deal with increased customer demands and increasing competition. Using digital transformation manages these pressures where agility, processes, and architecture are challenges you need to address. Containerizing applications is one of the key initiatives in digital transformation.

Containers and their platform provide increased speed from development to deployment by increasing operational efficiency. From development, to building, to testing, and then to deployment, containers streamline these operations. Containers also provide less overhead compared to traditional units of hardware or virtual hardware. Containers are highly portable with less dependency on the operating system and underlying hardware platform.

A key element in the successful deployment of a container platform is having a robust and flexible infrastructure that can meet the wide variety of requirements in a highly dynamic environment. Hitachi Vantara with Red Hat provides highly available and high-performance infrastructures for container applications. Some specific challenges of providing an infrastructure for container platform are the following:

- **Data Persistence**

Data is core for any application. Many applications require data persistence, such as PostgreSQL, MongoDB, and MySQL, among others. Continuous integration and continuous delivery (CI/CD) pipelines require data persistency at every level.

Hitachi Storage Plug-in for Containers integrates Hitachi Virtual Storage Platform (VSP) with Red Hat OpenShift Container Platform to provide persistent storage for stateful container applications. It enables the containers to store and manage persistent data in Hitachi Vantara's robust storage systems. Hitachi Vantara provides the storage infrastructure and integrations needed to adopt container technology.

- **Computing Platform**

With a wide range of applications that are stateful or stateless, a wide range of flexible computing platforms are also necessary to match memory requirements as well as CPU requirements. Depending on the deployment purposes and wide range of personas involved, different computing technology might be necessary, from bare metal hosts to virtual hosts.

The type of computing technology is also a consideration when licensing costs are considered. Hitachi Vantara provides different computing options from the 1U dual socket Hitachi Advanced Server DS120 to the 2U quad socket Hitachi Advanced Server DS240.

Hitachi Vantara's compute platforms provide bare metal host deployment options and VMware vSphere deployment to fit the vast dynamic computing needs a container platform demands.

- **Network connectivity**

As with any infrastructure a reliable network infrastructure is needed to provide the network connectivity to allow enough bandwidth and security for north-bound and south-bound traffic. Hitachi Unified Compute Platform (UCP) addresses the network connectivity using spine and leaf design using Cisco Nexus or Arista switches with Arista Extensible Operating System.

## ▪ Infrastructure Management

Having a robust and flexible infrastructure without efficient management only increases operation inefficiencies. This hinders the ability to increase innovation and agility.

Orchestration and automation are the key to operational efficiencies. Hitachi provides Hitachi Unified Compute Platform Advisor (UCP Advisor) as a single pane-of-glass management for converged infrastructure, providing automation for compute, network, and storage infrastructure.

Hitachi Ops Center is also available with Virtual Storage Platform for storage management.

This reference architecture document describes a sample deployment of Red Hat OpenShift Container Platform on the [Hitachi Unified Compute Platform CI](#) (UCP CI) series. This is a highly configurable integrated infrastructure in which server, network, and storage can be scaled independently, to optimize performance and eliminate overprovisioning costs. The Unified Compute Platform CI architecture consists of the following hardware components:

- Hitachi Virtual Storage Platform 5000, VSP E990, VSP G series, VSP F series for storage
- Hitachi Advanced Server systems for compute
- Hitachi Unified Compute Platform Advisor for end-to-end management
- Cisco Nexus 9000 or Arista 7000 for Ethernet networking
- Brocade G620 switches for Fibre Channel SAN

This reference architecture also provides the reference design for a build-your-own Red Hat OpenShift Container Platform environment, only using Hitachi Virtual Storage Platform. Although a specific converged system is used as an example, this Virtual Storage Platform reference design still applies to building your own container platform.

The intended audience of this document is IT administrators, system architects, consultants, and sales engineers to assist in planning, designing, and implementing Unified Compute Platform CI with OpenShift Container Platform solutions.

---

**Note** – Testing of this configuration was in a lab environment. Many things affect production environments beyond prediction or duplication in a lab environment. Follow the recommended practice of conducting proof-of-concept testing for acceptable results in a non-production, isolated test environment that otherwise matches your production environment before your production implementation of this solution.

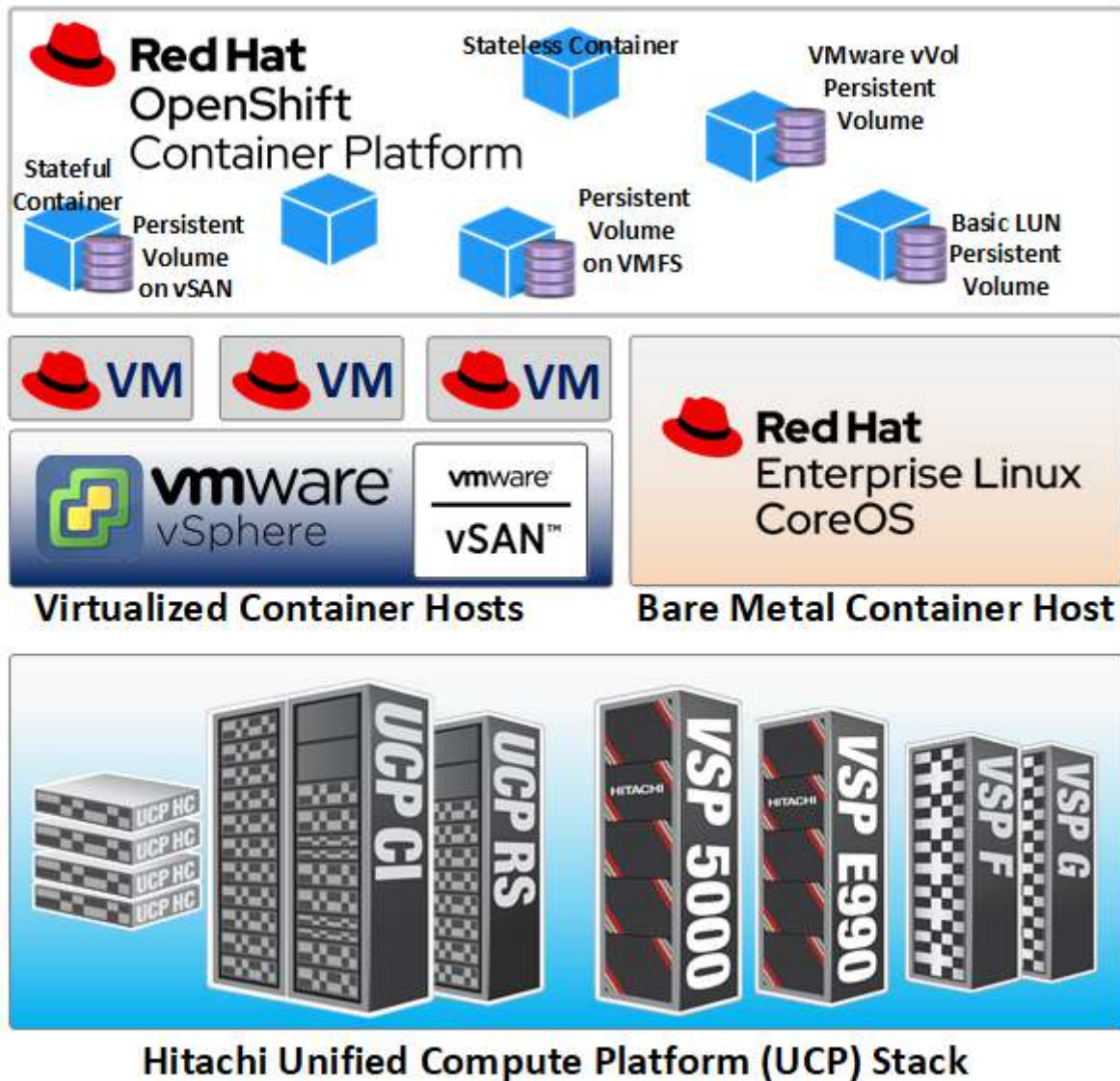
---

## Solution Overview

Hitachi Unified Compute Platform CI is an optimized and preconfigured converged infrastructure platform. It offers a broad range of compute and storage components that can be scaled and configured independently to eliminate overprovisioning. With Unified Compute Platform CI, you can optimize your data center to run any container application workload, at any scale.

Red Hat OpenShift is a powerful container orchestration platform. Providing OpenShift container platform is one of the solutions for Unified Compute Platform CI. Figure 1 on page 3 shows a high-level diagram of OpenShift managing containers and persistent volumes on the Unified Compute Platform CI stack or with Hitachi Virtual Storage Platform series systems.

Figure 1



You can deploy OpenShift on bare metal hosts and/or virtual hosts. Depending on the deployment purposes and wide range of personas involved, different deployments can be used. OpenShift deployment options on Unified Compute Platform CI are listed below.

- In a bare metal deployment, the master nodes run on Red Hat Enterprise Linux CoreOS, while the worker nodes can run on CoreOS or Red Hat Enterprise Linux. See “Bare Metal Container Hosts” on page 4.
  - Dynamically provision persistent volumes for stateful containers from Hitachi storage.
- Run virtualized Red Hat Enterprise Linux or CoreOS container hosts on VMware vSphere where the OpenShift master controller nodes and the worker nodes run on VMware vSphere ESXi hosts. See “Virtualized Container Hosts” on page 5.

The following persistent volume options are available for this configuration:

- Use VMware vVols to provision persistent volumes directly from Hitachi storage.
- Create persistent volumes from regular VMFS datastores.
- Create persistent volumes from VMware vSAN datastores hosted by Hitachi Unified Compute Platform HC nodes.
- Mix hybrid container hosts where OpenShift master controller nodes (and some worker nodes) are virtual machines plus some bare metal worker nodes. See “Mixed Hybrid Container Hosts” on page 6.

## Bare Metal Container Hosts

In an all bare metal deployment, deploy Red Hat OpenShift Container Platform (OCP) master controller nodes on Red Hat Enterprise Linux CoreOS and deploy the worker nodes on Red Hat Enterprise Linux (RHEL) hosts.

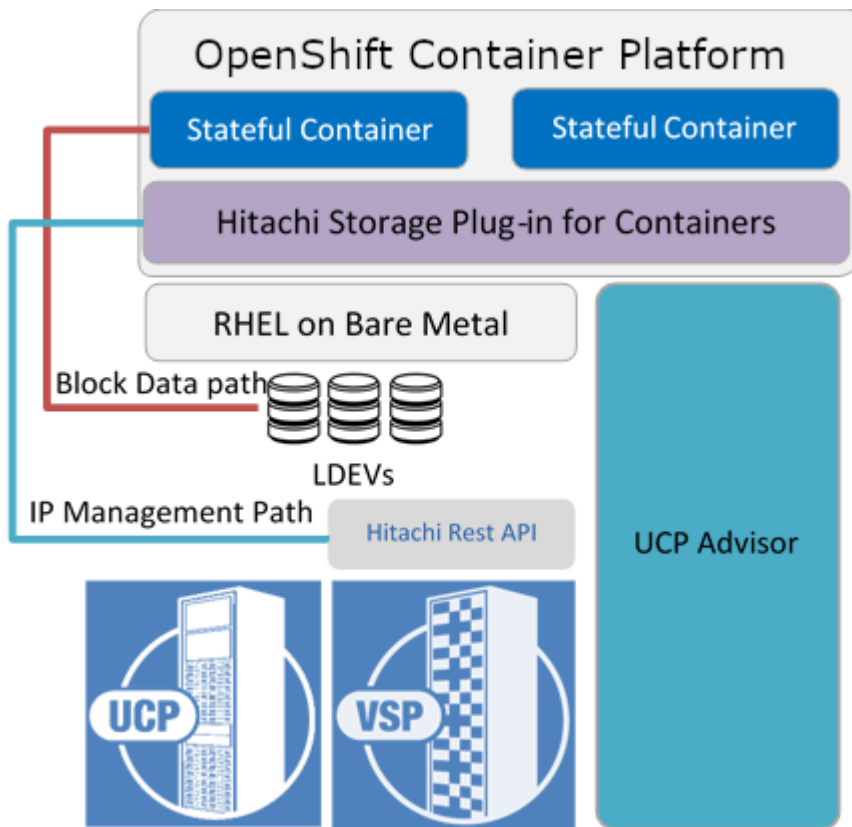
Bare metal hosts provide reliable and consistent access to compute resources. The compute resources are only shared with other containerized apps on the same worker nodes, but not with other worker nodes.

It also provides a more streamlined I/O stack, reducing overhead and complexity. Lowering complexity can help reduce the number of objects to manage and ease troubleshooting processes.

These benefits cater to environments where service level agreements (SLAs) are more stringent, as in production environments.

Figure 2 on page 5 shows an overview of a bare metal architecture for Red Hat OpenShift Container Platform.

**Figure 2**



The storage platforms can be any from the Hitachi Virtual Storage Platform family. Virtual Storage Platform provides a REST API for Hitachi Storage Plug-in for Containers to provision persistent volumes. Deploy Storage Plug-in for Containers within the OpenShift Container Platform cluster in the respective master controller nodes and worker nodes. Containers can access the persistent volumes through a local mount point inside the worker node. The persistent volumes are provided by Virtual Storage Platform-hosted LUNs through block protocol to the worker nodes.

## Virtualized Container Hosts

In an all virtualized deployment, deploy the Red Hat OpenShift Container Platform master controller nodes and worker nodes on virtual machines running Red Hat Enterprise Linux CoreOS hosted on VMware vSphere hosts.

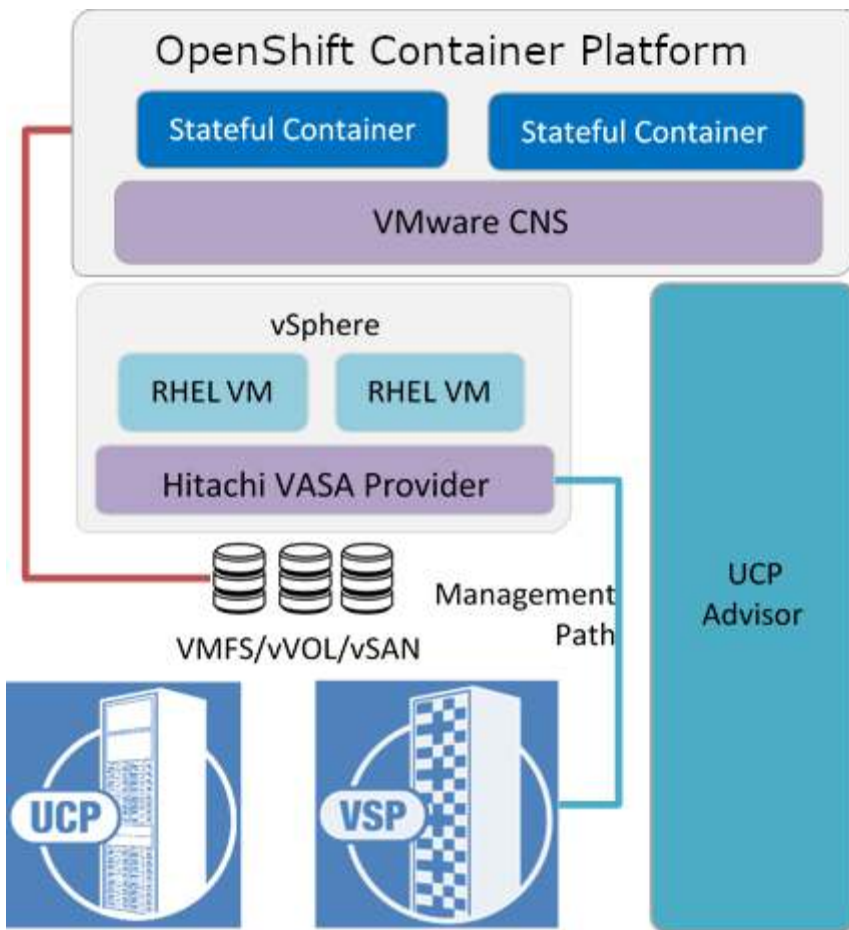
A virtualized OpenShift Container Platform deployment is highly flexible and dynamic. Due to vSphere's ability to deploy virtual machines quickly, new virtual worker nodes can be added to the cluster quickly. No physical deployment in the data center is required to have a functional OpenShift Container Platform node.

VMware vSphere can also help consolidate the OpenShift Container Platform nodes, reducing the rack space footprint. VMware's CPU scheduling ability and memory overcommit can further consolidate nodes. VMware High Availability and VMware Distributed Resource Scheduler can also help protect the OpenShift Container Platform cluster and enforce the SLA.

Figure 3 on page 6 shows an overview of a virtualized OpenShift Container Platform architecture.



Figure 3



The storage platforms can be any Hitachi Virtual Storage Platform family system, as well as Hitachi Unified Compute Platform HC with VMware vSAN-ready nodes. Hitachi Storage Provider for VMware vCenter provides Virtual Storage Platform capabilities awareness to VMware vCenter, where it can be used with VMware Storage Policy Based Management.

VMware Cloud Native Storage (CNS) provides the persistent storage provisioning capabilities using the VMware storage stack. Containers can access the persistent volumes through a local mount point inside the worker node virtual machines. The persistent volumes are provided by VMDKs in VMFS or with VMware vVols from Virtual Storage Platform. You can also provide persistent volumes through Unified Compute Platform HC with VMware vSAN-ready nodes.

## Mixed Hybrid Container Hosts

A mixed hybrid deployment uses bare metal hosts and virtualized hosts. Red Hat Enterprise Linux worker nodes deployed on bare metal systems provides simple, reliable, and consistent access to compute resources. Stringent SLAs are easier to achieve.

Red Hat Enterprise Linux or Red Hat CoreOS deployed on virtual machines hosting OpenShift Container Platform nodes provides a highly dynamic and flexible OpenShift Container Platform deployment. Virtual machines can be dynamically deployed to fit various needs.

Combining bare metal hosts and virtualized hosts provides the benefits of both deployment models.

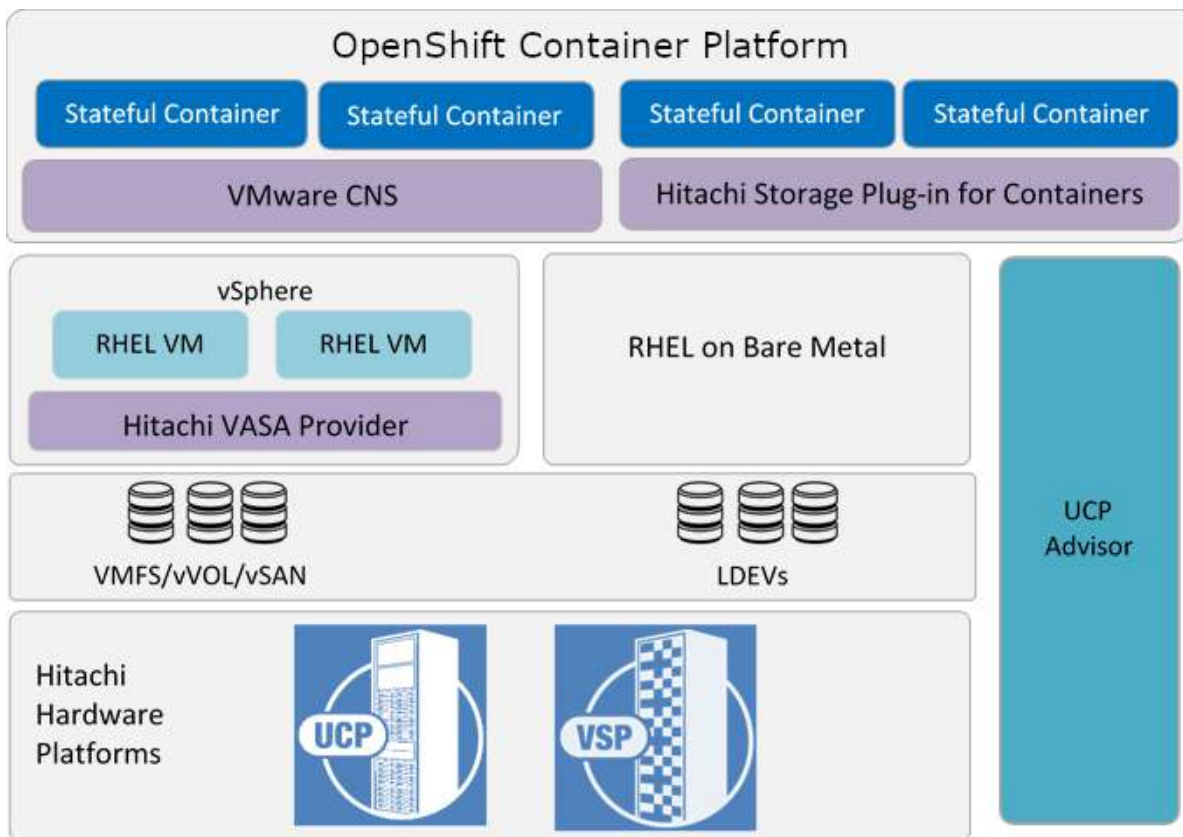
- Bare metal worker nodes are a better choice for containerized workloads that are more static in nature or are considered scale-up in nature with high SLA needs.
- A virtualized worker node is a better choice for container workloads and is considered more scale-out.
- A production-developer environment where production workloads are more static, but the development environment has more spin-up and spin-down scenarios.

In these examples, master nodes are running on virtual machines.

If there is a need to separate tenants at a cluster level, it is faster and efficient to create a new OpenShift Container Platform cluster with virtualized control plain.

Figure 4 shows an overview of a mixed hybrid OpenShift Container Platform architecture.

**Figure 4**



The storage can be any Hitachi Virtual Storage Platform family system, as well as Hitachi Unified Compute Platform HC with VMware vSAN-ready nodes.

Hitachi Provider for VMware vCenter with VMware CNS provides persistent storage provisioning capabilities for virtualized OpenShift Container Platform worker nodes, resulting in persistent storage provided by traditional VMFS volumes or vVols.

VMware CNS also provides persistent storage for virtualized worker nodes running on Unified Compute Platform HC with VMware vSAN-ready nodes. Hitachi Storage Plug-in for Containers provides persistent storage for bare metal and virtualized worker nodes.

---

**Note** – For all deployment options, when using Storage Plug-in for Containers with virtualized OpenShift Container Platform nodes, you must use iSCSI. Additionally, with a non-Unified Compute Platform infrastructure, the data path can be either iSCSI or Fibre Channel. The Unified Compute Platform CI stack only supports Fibre Channel.

---

The computing platform used in this reference architecture is provided by Hitachi Unified Compute Platform, where a wide range of 1U and 2U servers are available for the master controller nodes and the worker nodes, regardless of whether they are virtualized or bare metal. Hitachi Unified Compute Platform Advisor provides a single pane of glass management for bare metal and VMware vSphere infrastructures. It provides automation and management of the end-to-end hardware infrastructure including day-0 operations.

## Solution Components

This is the Hitachi Unified Compute Platform CI configuration used for this reference architecture.

## Hardware Components

Table 1 lists the key hardware components.

**TABLE 1. HARDWARE COMPONENTS**

Hardware	Description	Version	Quantity
Hitachi Advanced Server DS120 (compute)	<ul style="list-style-type: none"><li>2 Intel Xeon 4110 8-core, 2.10 GHz v5 processor</li><li>8 × 16 GB DIMM, 128 GB memory</li><li>128 GB SATADOM (boot)</li><li>Emulex LPe3200 32 Gb/s dual-port PCIe HBA</li><li>Intel X527-DA4 10 GbE quad-port NIC</li><li>If VMware vSAN is configured:<ul style="list-style-type: none"><li>The cache tier has 2 Intel Optane SSD DC P4800X (375GB, U.2) NVMe drives</li><li>The capacity tier had 6 Intel SSD DC P4510 (4TB, U.2) NVMe drives</li></ul></li></ul>	BMC: 4.65.06 BIOS: S5BH3B16.H01	5
Hitachi Virtual Storage Platform 5500	<ul style="list-style-type: none"><li>2 TB cache</li><li>24 × 1.9 TB NVMe drives used</li><li>4 × 32 Gb/s Fibre Channel ports used</li></ul>	90-03-02-00/00	1

**TABLE 1. HARDWARE COMPONENTS (CONTINUED)**

Hardware	Description	Version	Quantity
Hitachi Virtual Storage Platform E990	<ul style="list-style-type: none"> <li>1 TB cache</li> <li>24 × 1.9 TB NVMe drives used</li> <li>4 × 32 Gb/s Fibre Channel ports used</li> </ul>	93-02-01-60/00	1
Cisco Nexus 9332C switch (spine)	<ul style="list-style-type: none"> <li>32-port 40/100 GbE</li> <li>2-port 1/10 GbE</li> </ul>	9.2.3	2
Cisco Nexus 93180YC-FX switch	<ul style="list-style-type: none"> <li>48-port 10/25 GbE</li> <li>6-port 40/100 GbE</li> </ul>	7.0(3)I7(6)	2
Cisco Nexus 92348 switch	<ul style="list-style-type: none"> <li>48-port 1 GbE</li> <li>4-port 1/10/25 GbE</li> <li>2-port 40/100 GbE</li> </ul>	9.3.1	1
Brocade G620 switch	<ul style="list-style-type: none"> <li>48-port 16/32 Gbps Fibre Channel switch</li> </ul>	8.2.1c	2

Hitachi Unified Compute Platform CI hardware options are in “Hardware Components,” starting on page 36.

## Software Components

Table 2 lists the key software components.

**TABLE 2. SOFTWARE COMPONENTS**

Software	Version
Hitachi Storage Virtualization Operating System RF	90-03-02-00/00 93-02-01-60/00
Hitachi Unified Compute Platform Advisor	3.5
Hitachi Storage Provider for VMware vCenter	3.5.8
Hitachi Storage Plug-in for Containers	3.1.0
Red Hat OpenShift Container Platform	4.4.11
Red Hat Enterprise Linux	7.6
VMware vSphere	6.7u3
The Linux Foundation Kubernetes	1.17.1

Hitachi Unified Compute Platform CI software options are described in “Software Components,” starting on page 39.

## Solution Design

This is a detailed solution example of Hitachi Unified Compute Platform CI configured in a spine-leaf network topology.

Figure 5 shows a spine-leaf network topology for Unified Compute Platform CI with Cisco switches.

**Figure 5**

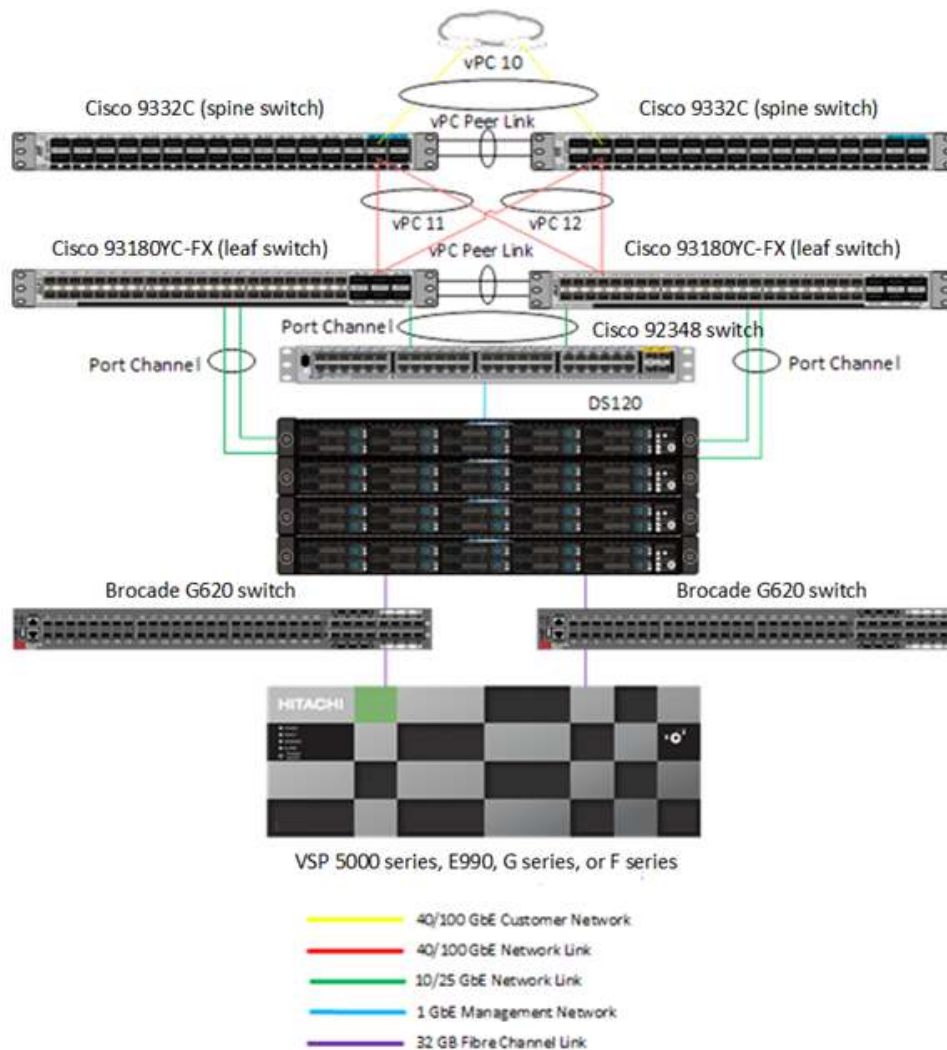
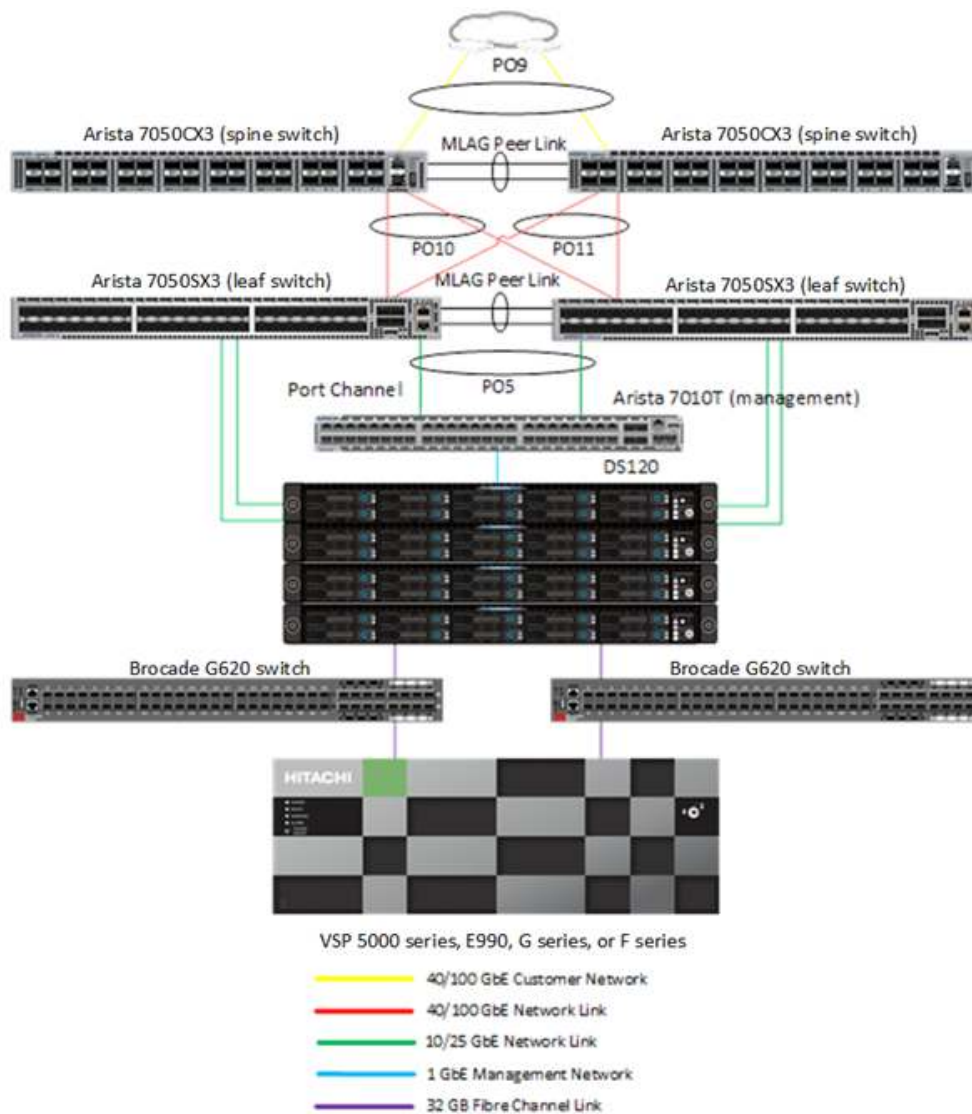


Figure 6 shows a spine-leaf network topology for Unified Compute Platform CI with Arista switches.

**Figure 6**

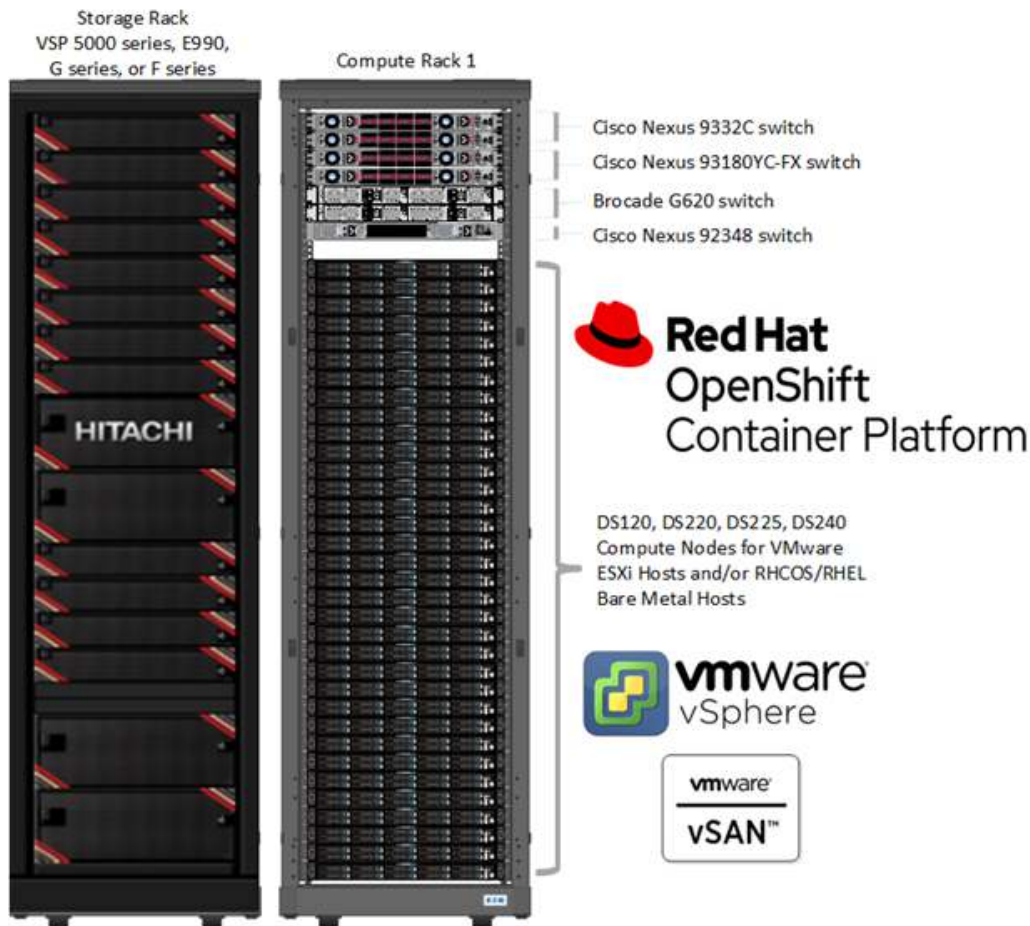


For more information regarding Unified Compute Platform CI configurations, see [Hitachi Unified Compute Platform CI for VMware vSphere Reference Architecture Guide](#).

Figure 7 on page 12 shows an example rack of the Unified Compute Platform CI for an OpenShift Container Platform cluster deployment. On the compute rack, Hitachi Advanced Server compute nodes can be used for Red Hat Enterprise Linux or Red Hat Enterprise Linux CoreOS bare metal container hosts, VMware ESXi hosts, or a combination of both.



Figure 7



## Deploy a OpenShift Container Platform Cluster on Bare Metal Hosts

Figure 8 on page 13 shows a high-level diagram of a bare metal Red Hat OpenShift Container Platform cluster on Hitachi Unified Compute Platform CI.

- Install three master nodes with Red Hat Enterprise Linux CoreOS 4.4 and two worker nodes with Red Hat Enterprise Linux 7 on Hitachi Advanced Server DS120 compute nodes. As of this writing, Hitachi Storage Plug-in for Containers supports only Red Hat Enterprise Linux worker nodes, with plans to support CoreOS soon.
- Deploy the Hitachi Storage Plug-in for Containers operator through OperatorHub.
- On Hitachi Virtual Storage Platform E990, create a storage pool name GOLD Plus from NVMe drives.
- On OpenShift Container Platform, define a StorageClass representing GOLD Plus storage pool.
- When creating a stateful container, Hitachi Storage Plug-in for Containers created a persistent volume dynamically from the GOLD Plus storage pool on Virtual Storage Platform E990.

Figure 8

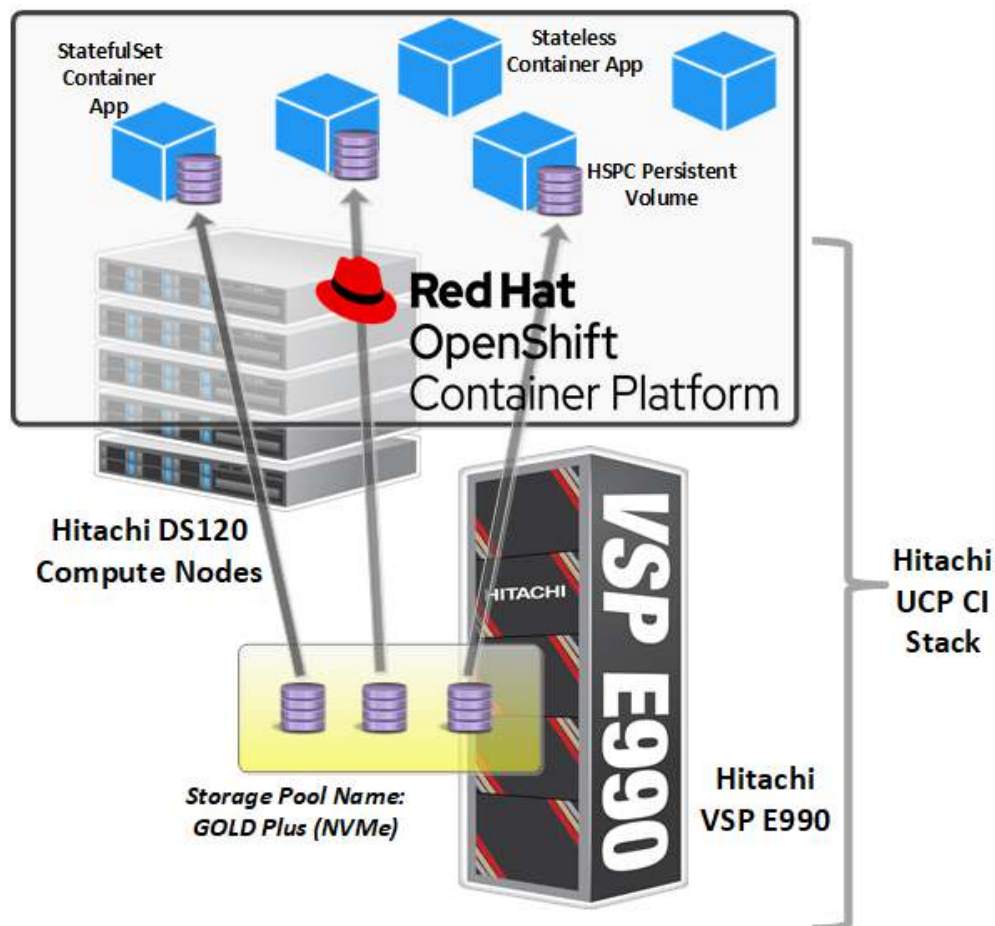


Table 3 lists bare metal OpenShift Container Platform hosts placement in Unified Compute Platform CI.

**TABLE 3. RED HAT OPENSIFT CONTAINER PLATFORM NODES AND IP ADDRESSES EXAMPLE FOR BARE METAL HOSTS**

Node Name	Node Role	Device Type and Location	IP Address
Router-LB	Load Balancer	Existing infrastructure outside of UCP CI	External: 10.76.46.100
	Router		Internal: 172.21.0.1
Ocp-master-1	Master Node 1	DS120 Rack1 U1	172.21.1.10
Ocp-master-2	Master Node 2	DS120 Rack1 U2	172.21.1.11
Ocp-master-3	Master Node 3	DS120 Rack1 U3	172.21.1.12
Ocp-worker-1	Worker Node 1	DS120 Rack1 U4	172.21.1.100
Ocp-worker-2	Worker Node 2	DS120 Rack1 U5	172.21.1.101

[Hitachi Unified Compute Platform Advisor](#) was used for Hitachi Virtual Storage Platform E990 day-0 operations, such as storage pool creation from NVMe drives.

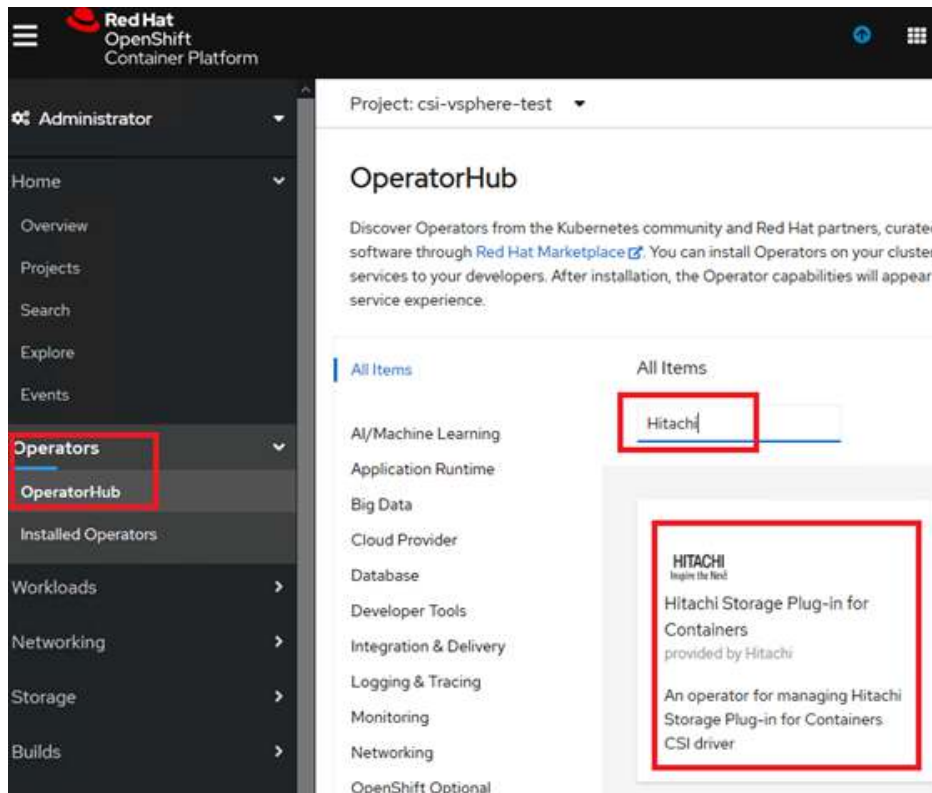
For detailed OpenShift Container Platform installation procedures, see [Installing a cluster on bare metal](#)



The Hitachi Storage Plug-in for Containers operator is a Kubernetes CSI driver that can be used to dynamically provision persistent volumes for stateful containers from Hitachi storage. Deploy the Hitachi Storage Plug-in for Containers operator through the OpenShift Container Platform OperatorHub web interface, as shown in Figure 9.

Hitachi Storage Plug-in for Containers makes it possible to fully utilize Virtual Storage Platform's high-performance and high-reliability features in a container environment.

**Figure 9**



For detailed Hitachi Storage Plug-in for Containers installation and configuration procedures, see [Containers](#) on the Hitachi Vantara Support website.

The following are the maximum values for stateful containers:

- The maximum number of volumes on Hitachi Virtual Storage Platform 5000 series is 65,280. For details, see [VSP 5000 series specifications](#).
- The maximum number of nodes is 2000.
- The maximum number of pods per node is 500, and total number of pods is 150,000. For details, see [OpenShift Container Platform tested cluster maximums](#).

# Deploy a Red Hat OpenShift Container Platform Cluster on VMware vSphere

Deploy a virtualized Red Hat OpenShift Container Platform cluster on Hitachi Unified Compute Platform CI with VMware vSphere. There are three deployment options for stateful container applications with Unified Compute Platform CI with VMware, as shown in Figure 10.

- **#1 in Diagram.** Deploy stateful containers on virtual volumes created with VMware vVols. Use Hitachi Storage Provider for VMware vCenter to provision persistent volumes directly from Virtual Storage Platform.
- **#2 in Diagram.** Deploy stateful containers on VMFS datastores. The persistent volumes are created as VMDK files.
- **#3 in Diagram.** Deploy stateful containers on datastores created with VMware vSAN hosted by Hitachi Unified Compute Platform HC nodes on Hitachi Advanced Server DS120.

This figure shows using Virtual Storage Platform 5000 series storage. In your implementation, you may use any Virtual Storage Platform F series, G series, or 5000 series system.

Figure 10

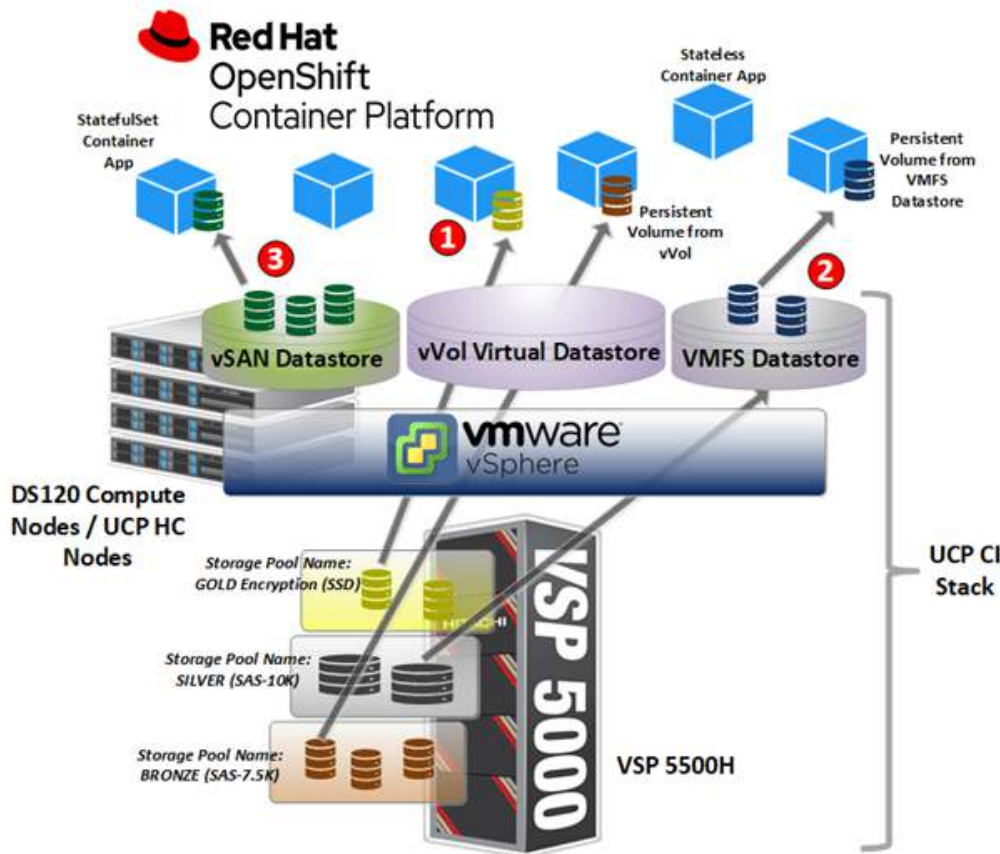


Figure 10 on page 15 also shows the following:

- Three storage pools consist of different type of drives on Virtual Storage Platform 5500H, as listed in Table 4.
- A **gold** encryption storage pool and **bronze** storage pool are placed into one VMware vVol storage container.
- One persistent volume is dynamically created from the **gold** encryption storage pool using VMware vVols and attached to a stateful container.
- The other persistent volume is dynamically created from the **bronze** storage pool using VMware vVols and attached to a stateful container.
- A VMFS datastore is created by a large volume/LUN from the **silver** storage pool.

**TABLE 4. HITACHI VIRTUAL STORAGE PLATFORM 5500H STORAGE POOLS**

Storage Pool Name	Disk Type	Usage and Capability
GOLD Encryption	SSD	High performance, high data protection with encryption.
SILVER	SAS 10K RPM	Mid performance. General use.
BRONZE	SAS 7.2K RPM	High capacity, lower performance, and cost effective.

The different types of storage pools meet the following demands:

- Virtual Storage Platform 5500H and VSP G series provide different storage pools with different capability levels, as shown in Table 4.
- Container applications require different storage requirements.
- The container administrators create volumes that fit the container application requirements.

Table 5 lists OpenShift Container Platform cluster virtual machine allocations on Unified Compute Platform CI with VMware.

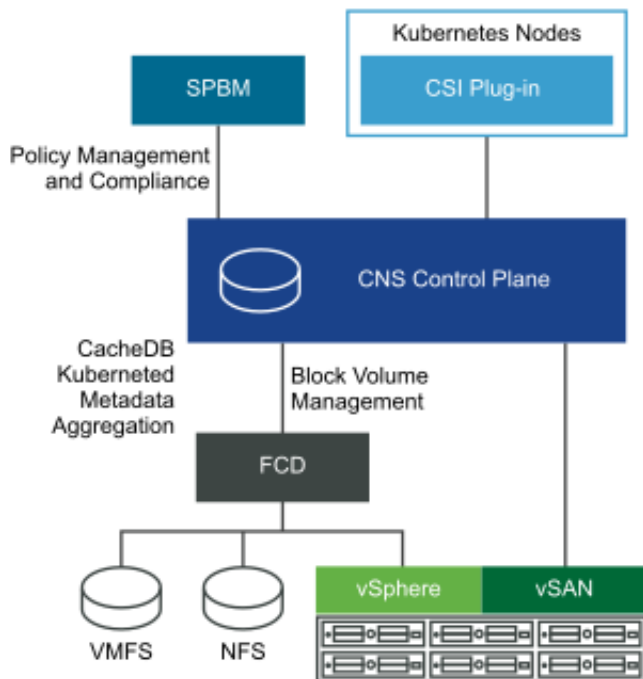
**TABLE 5. RED HAT OPENSIFT CONTAINER PLATFORM NODES AND IP ADDRESSES EXAMPLE FOR VIRTUAL MACHINE HOSTS**

Node Name	Node Role	Device Type and Location	IP Address
Router-LB	Load Balancer	Existing infrastructure outside of UCP CI	External: 10.76.46.100
	Router		Internal: 172.21.0.1
Ocp-master-1	Master Node 1	VMware vSphere cluster hosted in DS120, Rack 1, U1 – U4	172.21.1.10
Ocp-master-2	Master Node 2		172.21.1.11
Ocp-master-3	Master Node 3		172.21.1.12
Ocp-worker-1	Worker Node 1		172.21.1.100
Ocp-worker-2	Worker Node 2		172.21.1.101
Ocp-worker-3	Worker Node 3		172.21.1.102
Ocp-worker-4	Worker Node 4		172.21.1.103

For detailed OpenShift Container Platform installation procedures, see [Installing a cluster on vSphere](#).

VMware Cloud Native Storage (CNS) creates persistent volumes for containers. Figure 11 shows the components of VMware CNS, which is explained in more detail at [VMware Docs Cloud Native Storage Components](#).

**Figure 11**



VMware CNS contains the VMware vSphere Container Storage Interface (CSI), which interfaces with Kubernetes clusters. Also use [Storage Policy Based Management \(SPBM\)](#) for container volume management. With Container Storage Interface and Storage Policy Based Management, create the persistent volumes from any of the datastore types that are supported by vSphere.

## Use VMware vVols to Provision Persistent Volumes Directly from Hitachi Virtual Storage Platform

VMware vVols provides more granular volume management with Storage Policy Based Management. For each virtual volume, the Hitachi storage creates a dedicated volume (DP-Vol) dynamically. For a detailed explanation of VMware vVols, see [Understanding Virtual Volumes \(vVols\) in VMware vSphere 6.7/7.0 \(2113013\)](#).

To use VMware vVols with Hitachi storage, install Hitachi Storage Provider for VMware vCenter. For more information, see [VMware vSphere Virtual Volumes \(vVols\) with Hitachi Virtual Storage Platform Quick Start and Reference Guide](#).

To deploy this environment, see [Storage Provider for VMware vCenter \(VASA\)](#).

Figure 12 is a screenshot that shows detailed capabilities that are included in Storage Provider for VMware vCenter.

**Figure 12**

Capability Schema	
Capability Schema	
Name	Capability Schema for All
Namespace	com.hitachi.storageprovider.vvol
Capability	Available Values
Performance IOPS - Class	Tier1_IOPS,Tier2_IOPS,Tier3_IOPS
Performance Latency - Class	Tier1_Latency,Tier2_Latency,Tier3_Latency
Availability - Class	Tier1,Tier2,Tier3
Cost - Class	10 - 1000
Snapshot Backup Importance - Class	Critical,Important,Best Effort
Drive Type/Drive Speed	FMD DC2,FMD HDE,FMD,SSD,SAS 15K,SAS 10K,SAS 7.2K,External
Pool Type	HDP,HDT
RAID Level	RAID6(14D+2P),RAID6(12D+2P),RAID6(6D+2P),RAID5(7D+1P),RAID5(6D+1P)
Encryption	Yes,No
Snapshot	Yes,No
Deduplication	Yes,No
Compression	Accelerated,Normal,No
Additional tag based capabilities	
User Defined Capabilities	Available Values
Location	Dublin,Frankfurt,Munich

Define the virtual machine and VMDK storage policies using the capabilities above that meet your data and apps requirements for the Red Hat OpenShift Container Platform environment. Reference the virtual machine (VM) storage policy name in the corresponding **StorageClass** YAML file to create a **StorageClass** for vVols. An example can be found in "Solution Validation" on page 24.

## Create Dynamic Persistent Volumes from Regular VMFS Datastores

In VMware CSI, there are two ways to specify regular VMFS datastores for Red Hat OpenShift Container Platform to use for volume creation:

- Use VMware Storage Policy Based Management (SPBM) to specify one or more datastores in the same category.  
If more than two datastores are in the same storage policy, the datastore used is the one with the most amount of free space available.
- Use the datastore's URL to specify a single datastore.

For any storage management, VMware recommends using SPBM. To use SPBM for VMFS datastore, create a tag-based virtual machine storage policy. See [Using Tag-Based SPBM Policies to Manage Your Storage](#).

Hitachi Storage Provider for VMware vCenter also automatically tags datastores with a set of capabilities for LUNs that are provisioned from Hitachi storage.

Define the virtual machine storage policies for VMFS datastore that meet your storage requirements for the Red Hat OpenShift Container Platform environment. Reference the virtual machine storage policy name in the corresponding **StorageClass** YAML file to create a **StorageClass**. An example can be found in Solution Validation.

## Create Persistent Volumes from VMware vSAN Datastores Hosted by Hitachi Unified Compute Platform HC Nodes

[Hitachi Unified Compute Platform HC series](#) is a certified VMware vSAN-ready node that can be combined with the Unified Compute Platform CI stack. The vSAN datastore and VMFS datastore can coexist in the same Unified Compute Platform stack, providing more storage options for your VMware environment.

VMware vSAN comes with a default storage policy with data protection of one mirroring copy. Create your own vSAN storage policy to meet your storage and data protection requirements. See [About the vSAN Default Storage Policy](#).

Define the virtual machine storage policies for the vSAN datastore that meets your storage requirements for the Red Hat OpenShift Container Platform environment. Reference the virtual machine storage policy name in the corresponding **StorageClass** YAML file to create a **StorageClass**.

## Configure High-Availability for Virtualized Master Nodes and Worker Nodes

When deploying a virtual machine, VMware vSphere deploys the virtual machine into the VMware vSphere ESXi host that is using the least amount of resources, by default. In the shared VMware environment, this creates a possibility that all the Red Hat OpenShift Container Platform master virtual machines reside on the same ESXi host. If this ESXi host goes down, all the master nodes go down as well, affecting the OpenShift Container Platform cluster operation. To avoid this situation, use the VM-VM affinity rules for vSphere environments.

Figure 13 on page 20 shows an example of anti-affinity rules for the cluster virtual machines. With four ESXi hosts in the VMware vSphere Distributed Resource Scheduler (DRS) cluster, the three master nodes and four worker nodes are placed into separate ESXi hosts. One master node and one worker node can be placed in the same ESXi host.

Figure 13

The screenshot shows the vSphere ClusterB configuration interface. The left sidebar contains a navigation tree with categories: Services (vSphere DRS, vSphere Availability), Configuration (Quickstart, General, Licensing, VMware EVC, VM/Host Groups, **VM/Host Rules**, VM Overrides, Host Options, Host Profile, I/O Filters), More (Alarm Definitions, Scheduled Tasks), and vSAN (Services). The main panel is titled 'VM/Host Rules' and has tabs for Summary, Monitor, Configure (selected), Permissions, Hosts, VMs, and Datastores. Below the title are buttons for '+ Add...', 'Edit...', and 'Delete'. A table lists the rules:

Name	Type	Enabled
ocp-master-nodes	Separate Virtual Machines	Yes
ocp-worker-nodes	Separate Virtual Machines	Yes

Below this is the 'VM/Host Rule Details' section, which states: 'The listed 3 Virtual Machines must run on different hosts.' It includes buttons for '+ Add...', 'Details...', and 'Remove'. A table shows the rule members and their conflicts:

Rule Members	Conflicts
ocp4-master-3	0
ocp4-master-1	0
ocp4-master-2	0

For more information regarding VM-VM affinity rules, see [VM-VM Affinity Rules](#).

## Deploy a Red Hat OpenShift Container Platform Cluster on Mixed Hybrid Environments (Virtualized and Bare Metal Hosts)

All the previous deployment configurations can be combined and mixed. A hybrid environment on Hitachi Unified Compute Platform CI is where Red Hat OpenShift Container Platform master nodes and some worker nodes are virtualized, and some bare-metal worker nodes are also joined to the cluster. Table 6, "Red Hat OpenShift Container Platform Nodes and IP Addresses Example for Virtualized and Bare Metal Hosts," on page 21 provides an example of this configuration.

**TABLE 6. RED HAT OPENSIFT CONTAINER PLATFORM NODES AND IP ADDRESSES EXAMPLE FOR VIRTUALIZED AND BARE METAL HOSTS**

Node Name	Node Role	Device Type and Location	IP Address
Router-LB	Load Balancer	Existing infrastructure outside of UCP CI	External: 10.76.46.100
	Router		Internal: 172.21.0.1
Ocp-master-1	Master Node 1 (VM)	VMware vSphere cluster hosted in DS120 Rack1 U1 – U3	172.21.1.10
Ocp-master-2	Master Node 2 (VM)		172.21.1.11
Ocp-master-3	Master Node 3 (VM)		172.21.1.12
Ocp-worker-1	Worker Node 1 (VM)	DS120 Rack1 U4	172.21.1.100
Ocp-worker-2	Worker Node 2 (VM)		172.21.1.101
Ocp-worker-3	Worker Node 3 (Bare Metal)		172.21.1.102
Ocp-worker-4	Worker Node 4 (Bare Metal)		172.21.1.103

For detailed hybrid OpenShift Container Platform installation procedures, see [Installing OCP 4.3 on VMware with UPI](#).

To add more Red Hat Enterprise Linux bare metal nodes, see [Adding RHEL compute machines to an OpenShift Container Platform cluster](#)

---

**Note** – For OpenShift Container Platform 4.4, if you have a OpenShift Container Platform virtual machine cluster deployed by a vSphere installer, as showed in “Configure High-Availability for Virtualized Master Nodes and Worker Nodes” on page 19, you will not be able to add bare metal Red Hat Enterprise Linux worker nodes using the link above. You need to use bare metal user-provisioned infrastructure (UPI) installation as in [Installing OCP 4.3 on VMware with UPI](#).

---

Once all the nodes are joined to the cluster, OpenShift Container Platform does not know if the worker nodes are bare metal hosts or virtual machines. One way to help distinguish these node types is to assign labels to each node and use the **nodeSelector** attribute while deploying container apps. This also applies to VMware Container Storage Interface (CSI) plug-in installations as well.

Before installing the CSI plug-in, use the following command to assign a label to each node:

```
# oc label nodes <node-name> <label-key>=<label-value>
```

This example uses the following labels:

- Virtual machine worker nodes:

```
# oc label nodes ocp-worker-1 node-type=vm-cns
```

- Bare metal worker nodes:

```
# oc label nodes ocp-worker-3 node-type=hspc-fc
```

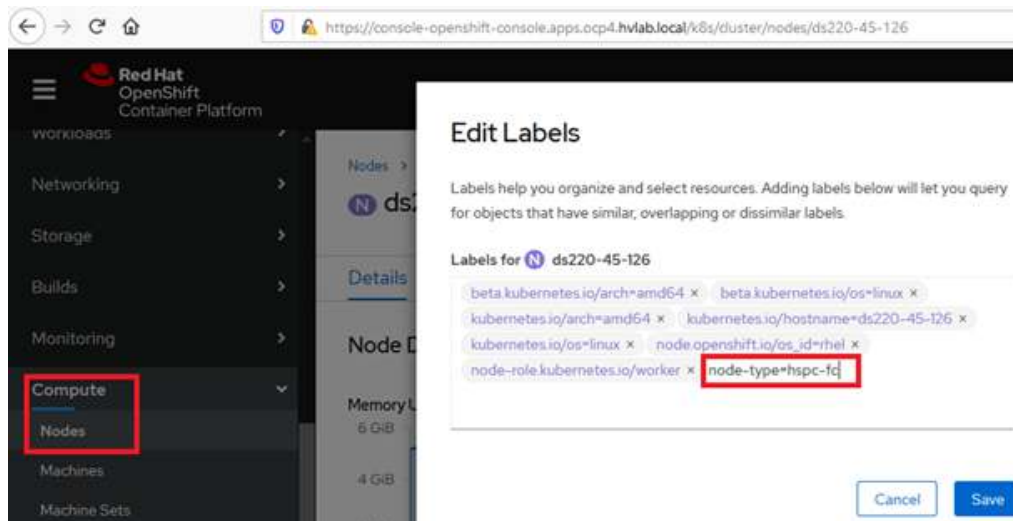


You can add the node labels from the OpenShift Container Platform web console as well, as this screenshot in Figure 14 shows.

To add a custom label, do the following:

1. From **Compute**, select **Nodes** and then select a node.
2. Select **Action**, select **Edit Labels**.
3. Add a custom label and select **Save**.

**Figure 14**



When installing VMware CNS and the Hitachi Storage Plug-in for Containers CSI drivers, nodeSelector can be added to the YAML file. These CSI drivers deploy a daemon set to each worker nodes.

The Storage Plug-in for Containers daemon set (**hspc-csi-node**) can also be deployed to virtual machine worker nodes with iSCSI configuration.

If you choose to use Storage Plug-in for Containers with only bare metal nodes, follow this procedure to only install **hspc-csi-node**:

1. Install Hitachi Storage Plug-in for Containers from OperatorHub. This installs **hspc-csi-node** to all worker nodes, including virtual machine worker nodes.
2. As shown in Figure 15, open **hspc-csi-node**, and add **node-type: hspc-fc** in the YAML file and **Save**. This removes **hspc-csi-node** from virtual machine worker nodes.

Figure 15

The screenshot shows the Red Hat OpenShift Container Platform console. On the left, the 'Workloads' menu is expanded, and 'Daemon Sets' is highlighted. The main panel shows the 'hspc-csi-node' Daemon Set details for the 'hspc-test' project. The 'YAML' tab is selected, displaying the following configuration:

```
6 selfLink: /apis/apps/v1/namespaces/hspc-test/daemonsets/
7 resourceVersion: '204279'
8 name: hspc-csi-node
9 uid: 6ced2cc0-e4b3-4725-b864-59eb1b15a740
10 creationTimestamp: '2020-08-29T04:39:14Z'
11 generation: 1
12 namespace: hspc-test
13 ownerReferences:
14   - apiVersion: csi.hitachi.com/v1
15     kind: HSPC
16     name: hspc
17     uid: 3e286bd5-af7f-4a7e-a8f4-baf3f72f0878
18     controller: true
19     blockOwnerDeletion: true
20 spec:
21   selector:
22     matchLabels:
23       app: hspc-csi-driver
24   template:
25     metadata:
26       creationTimestamp: null
27     labels:
28       app: hspc-csi-driver
29     spec:
30       nodeSelector:
31         node-type: hspc-fc
32       restartPolicy: Always
33       serviceAccountName: hspc-csi-sa
```

For detailed information regarding nodeSelector, see [Placing pods on specific nodes using node selectors](#).

See VMware vSphere CPI/CSI Setup Files for Red Hat OpenShift Container Platform 4 for more information about installing VMware vSphere CSI into OpenShift Container Platform in a hybrid configuration.

## Solution Validation

The Red Hat OpenShift Container Platform cluster was deployed on Hitachi Unified Compute Platform CI successfully with bare metal, virtualized, and hybrid configurations. Hitachi Storage Plugin for Containers and VMware Cloud Native Storage drivers were installed for persistent volume provisioning. These container volume operations were performed to validate this solution.

## Create Persistent Volumes for Red Hat OpenShift Container Platform Bare Metal Hosts

Hitachi Storage Plugin for Containers was installed for bare metal Red Hat OpenShift Container Platform hosts to provision the persistent volumes directly from Hitachi Virtual Storage Platform E990. The **sc-e990.yaml** file was used for storage class creation. See Figure 16.

Figure 16

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: sc-e990
  annotations:
    kubernetes.io/description: Hitachi Storage Plug-in for Containers
    storageclass.kubernetes.io/is-default-class: "true"
provisioner: hspc.csi.hitachi.com
reclaimPolicy: Delete
volumeBindingMode: Immediate
allowVolumeExpansion: true
parameters:
  serialNumber: "445617"
  poolID: "0"
  portID : CL4-C,CL2-A
  connectionType: fc
  csi.storage.k8s.io/fstype: ext4
  csi.storage.k8s.io/node-publish-secret-name: "secret-e990"
  csi.storage.k8s.io/node-publish-secret-namespace: "hspc-test"
  csi.storage.k8s.io/provisioner-secret-name: "secret-e990"
  csi.storage.k8s.io/provisioner-secret-namespace: "hspc-test"
  csi.storage.k8s.io/controller-publish-secret-name: "secret-e990"
  csi.storage.k8s.io/controller-publish-secret-namespace: "hspc-test"
  csi.storage.k8s.io/node-stage-secret-name: "secret-e990"
  csi.storage.k8s.io/node-stage-secret-namespace: "hspc-test"
  csi.storage.k8s.io/controller-expand-secret-name: "secret-e990"
  csi.storage.k8s.io/controller-expand-secret-namespace: "hspc-test"
```

With the **storageclass.kubernetes.io/is-default-class: "true"** setting, OpenShift Container Platform used this storage class by default for persistent volume creation. This makes sure that persistent volumes were created by Storage Plug-in for Containers for validation.

For new bare metal hosts, Storage Plug-in for Containers automatically created host groups in Hitachi Virtual Storage Platform using the specified Fibre Channel ports in the **portID** parameter.

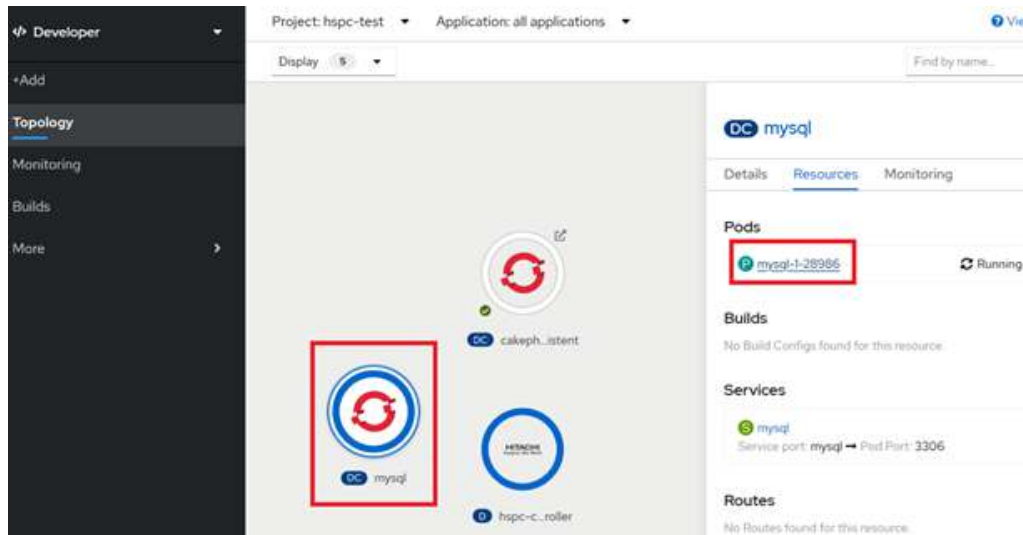
A storage class was created using the command line, from an administrator workstation with OpenShift Container Platform client software installed, entering the following command.

```
$ oc create -f sc-e990.yaml
```

To validate stateful container creation with a persistent volume, MySQL container was deployed from the OpenShift Container Platform web console Developer Catalog.

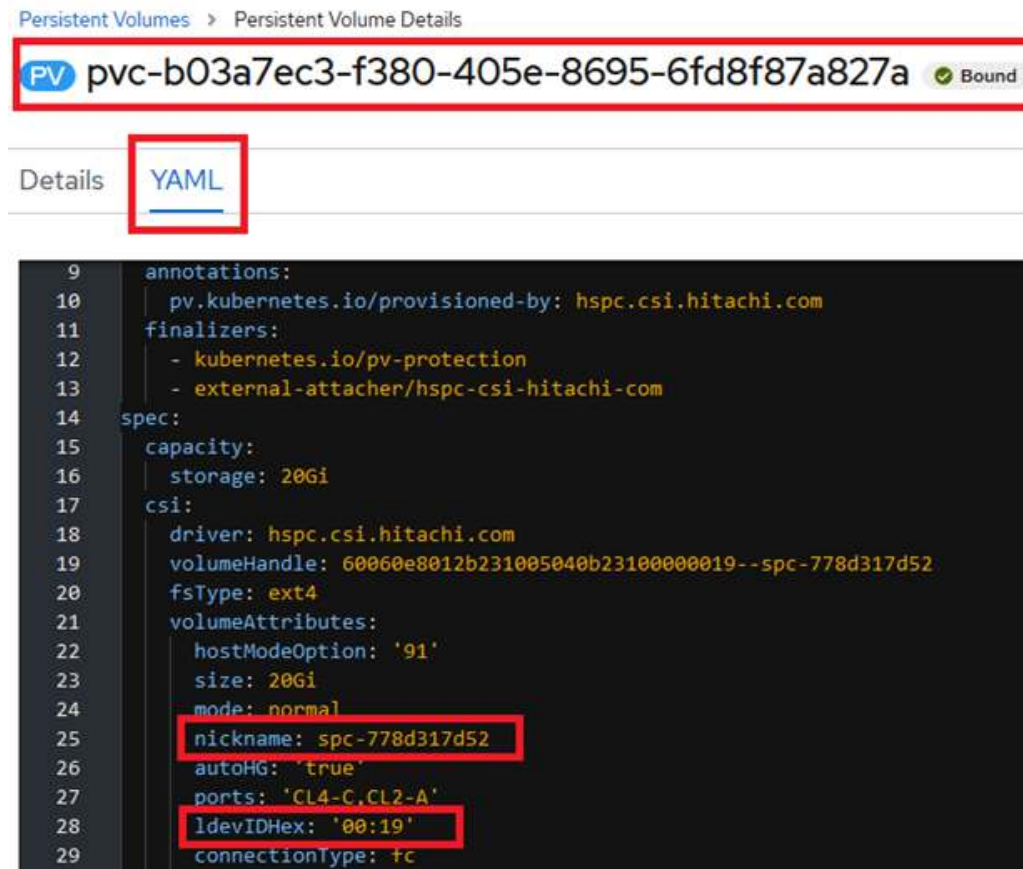
The **mysql** stateful container was successfully deployed and in running state as shown in the screenshot in Figure 17.

**Figure 17**



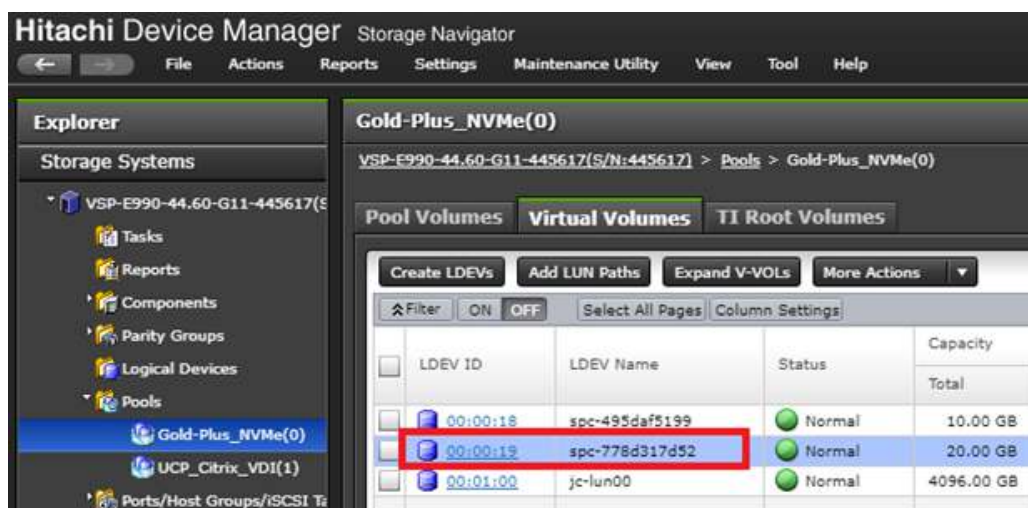
The persistent volume information can be found as you navigate into the mysql pod. To map a persistent volume to the corresponding volume in the storage, look into **YAML** section with **ldevIDHex** and **nickname** attributes, as shown in the screenshot in Figure 18 on page 26.

Figure 18



**Hitachi Device Manager – Storage Navigator** was used to look for the corresponding volume in the storage, as shown in the screenshot in Figure 19.

Figure 19

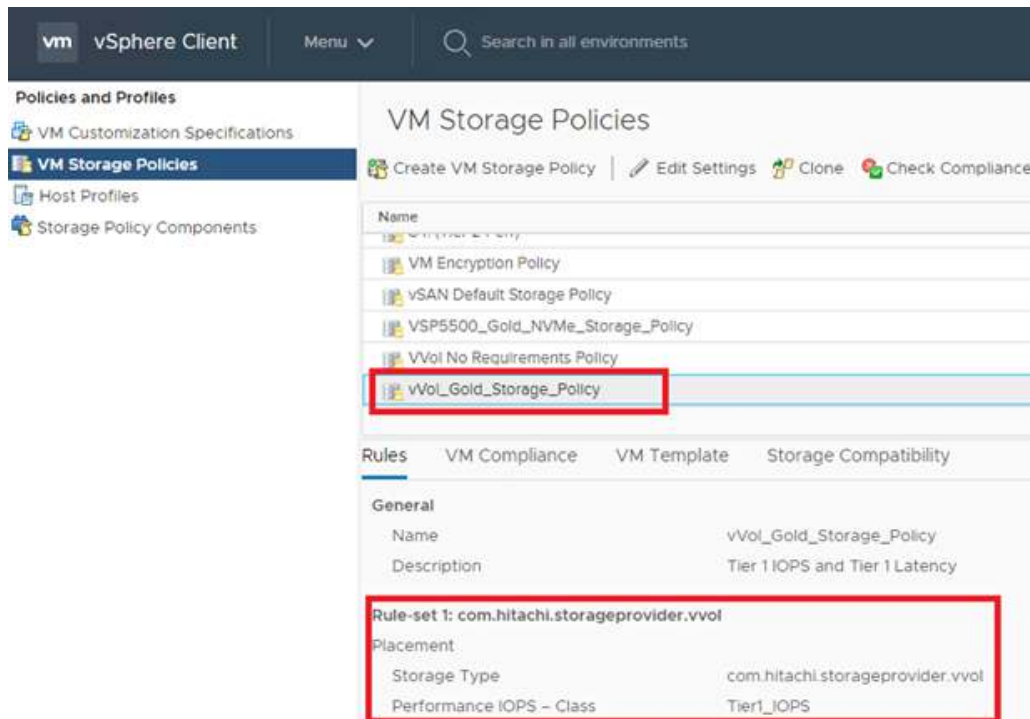


## Create Persistent Volumes using VMware vVols

One of the common methods to manage the storage volume in a VMware vSphere environment is using VMware vSphere Storage Policy Based Management (SPBM). The storage requirements, such as IOPS and I/O latency required by the applications that make up the system, vary from application to application.

Policy-based storage management allows each application to have a persistent volume that meets their requirements. Figure 20 shows that the storage policy **vVol\_Gold\_Storage\_Policy** was created and used for this validation.

**Figure 20**



The storage class was created next. Figure 21 on page 28 is a screenshot that shows the storage class creation using the OpenShift Container Platform web console.

Figure 21

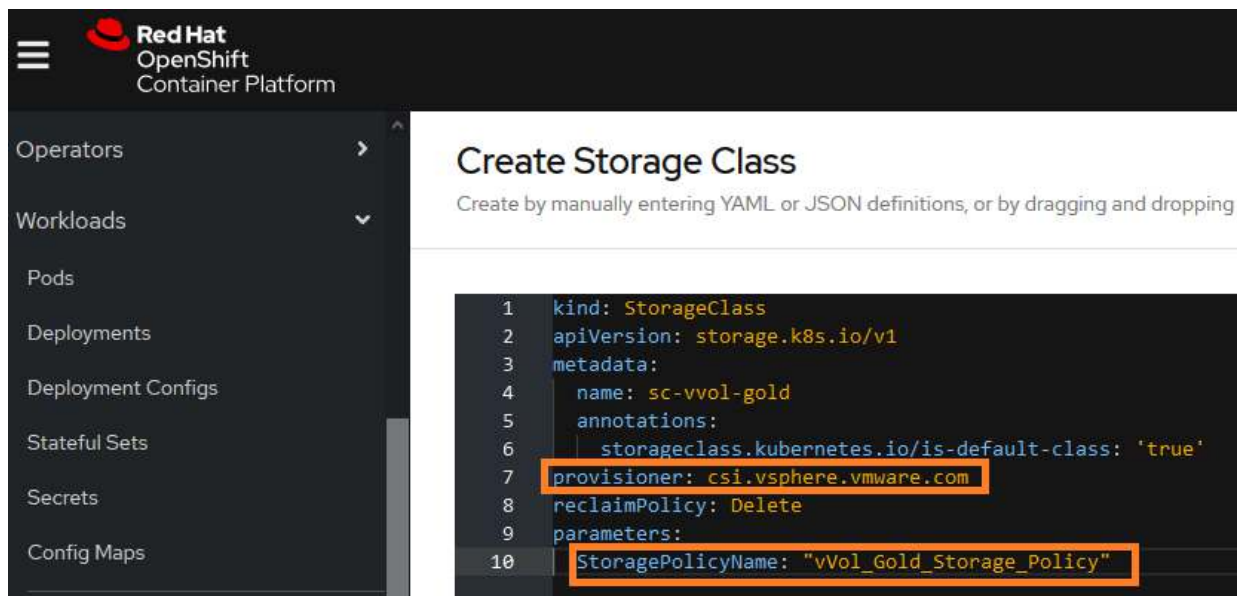


Figure 22 is a screenshot that shows PostgreSQL stateful database container deployment from the OpenShift Container Platform web console Developer Catalog.

Figure 22

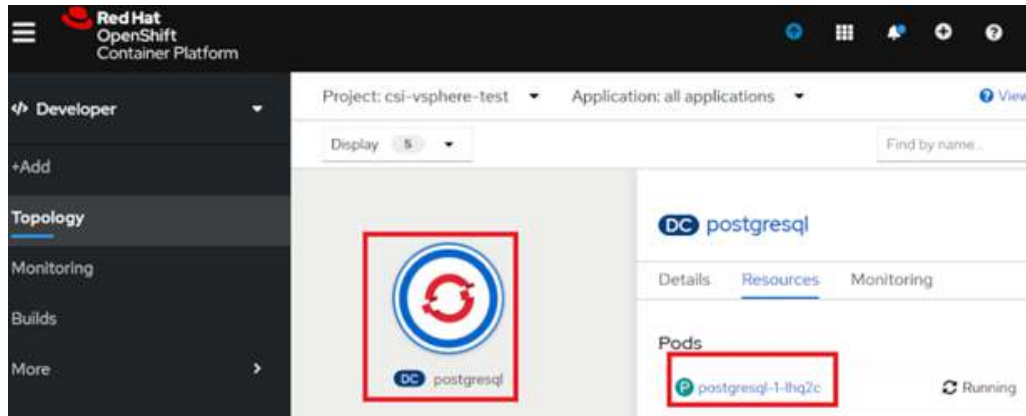




Figure 23 is a screenshot that shows the detail of persistent volume claim information and persistent volume name from the postgresl pod.

Figure 23

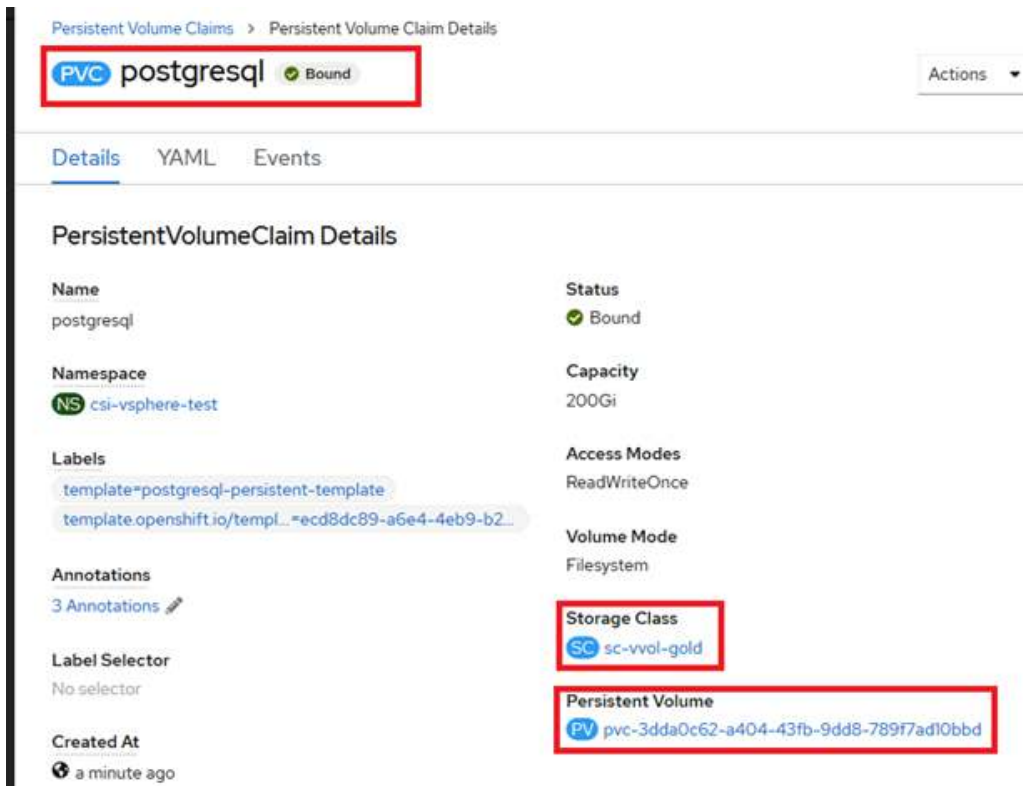


Figure 24 on page 30 is a screenshot that shows that the same persistent volume can be found in the **Cloud Native Storage/ Container Volumes** section in VMware vCenter.



Figure 24

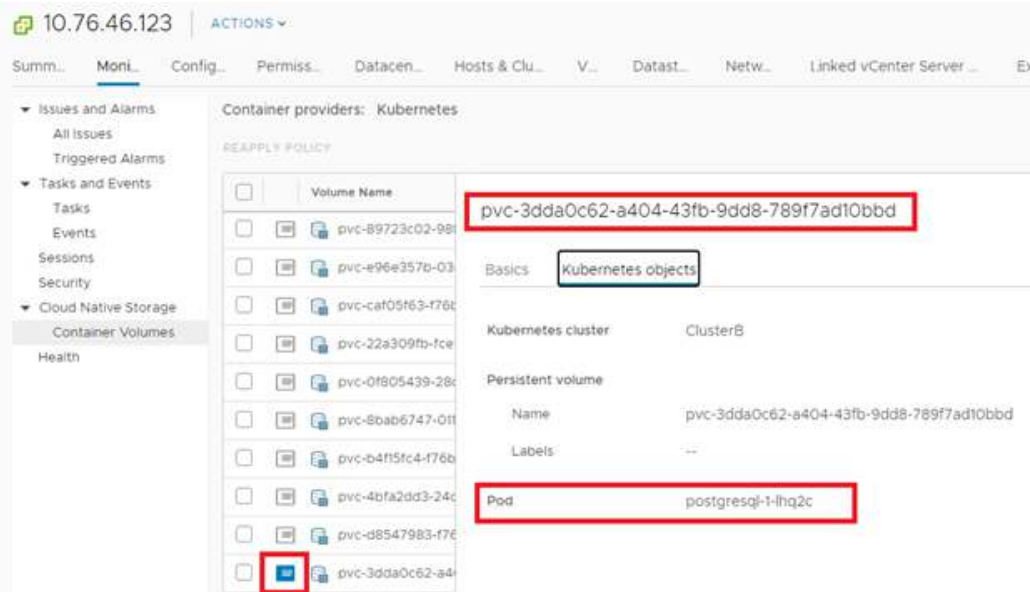
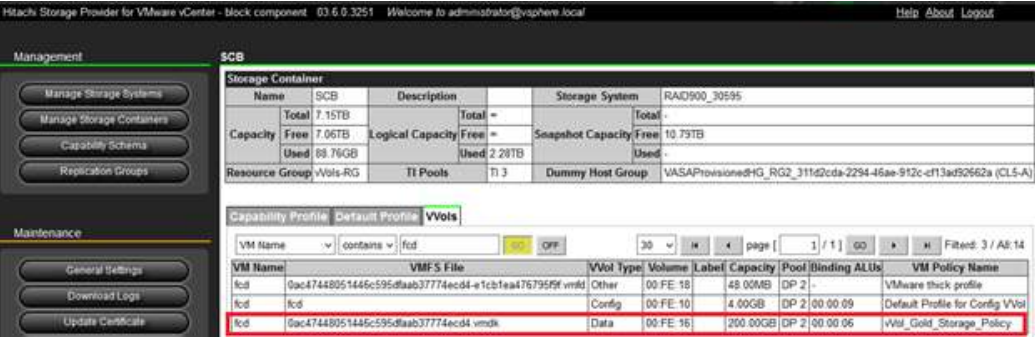


Figure 25 is a screenshot that shows the specific virtual volume created with VMware vVol and the corresponding Hitachi Virtual Storage Platform 5500 volume (LDEV) ID on the web interface of Hitachi Storage Provider for VMware vCenter.

Figure 25

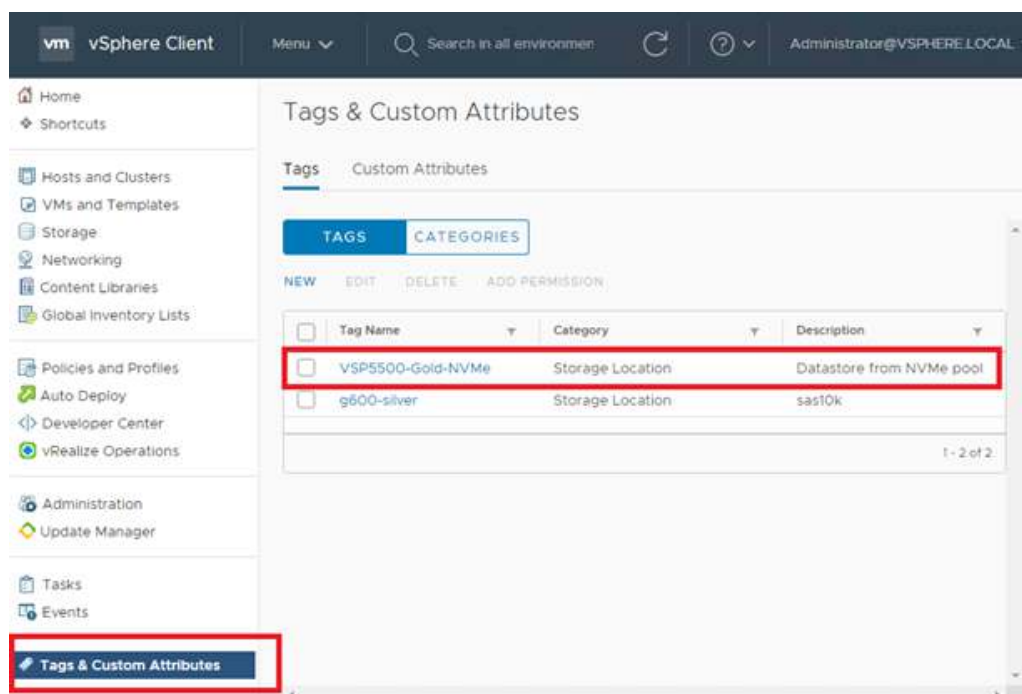


## Create Persistent Volumes from a VMFS Datastore

The VMware CNS CSI driver was installed for virtualized Red Hat OpenShift Container Platform hosts to provision the persistent volumes from a VMFS datastore. Deploying a stateful container on a VMFS datastore validated this action. The following is the test procedure:

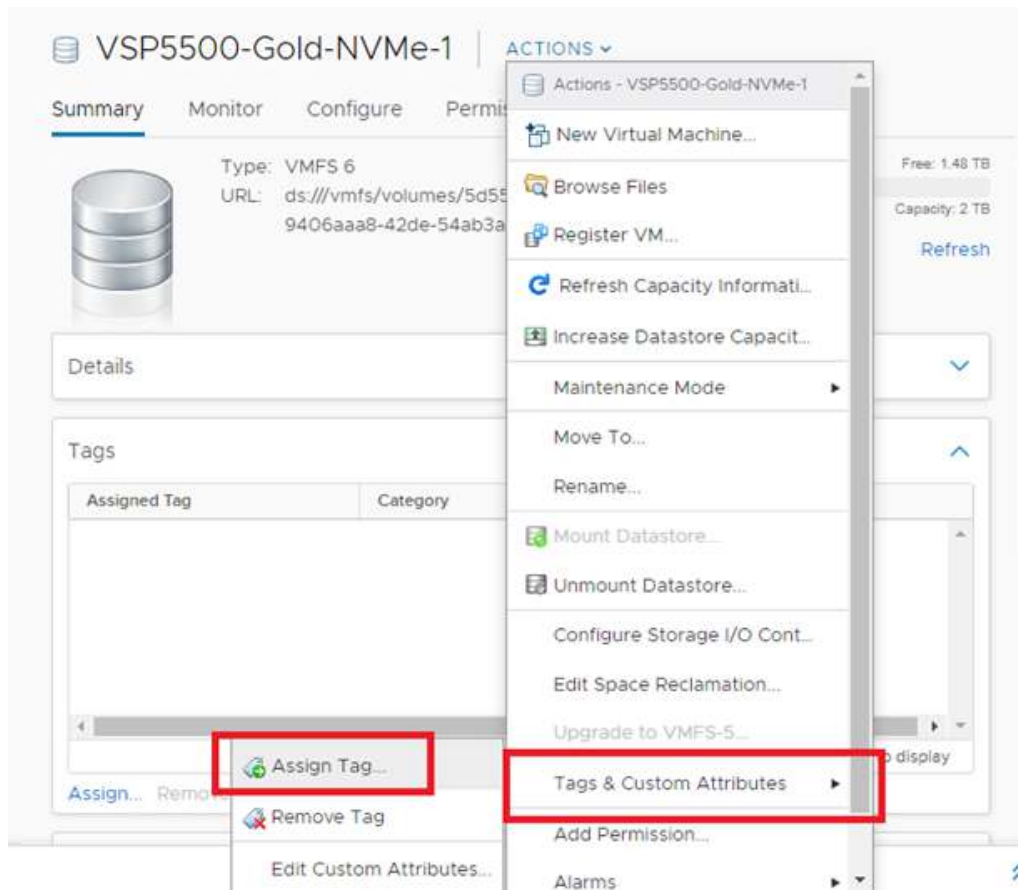
1. Create a virtualized OpenShift Container Platform cluster on VMware vSphere hosted on Hitachi Advanced Server DS120 nodes.
2. Create a VMFS datastore provisioned from a LUN/LDEV on Virtual Storage Platform 5500 with a storage pool created from NVMe drives.
3. Create a tag-based VM Storage Policy in VMware vSphere.
4. Install the VMware CNS CSI driver on the OpenShift Container Platform cluster.
5. Use Developer Catalog to deploy a MongoDB stateful application with an 80 GB persistent volume.
6. To use the VMware Storage Policy Based Management features for a VMFS datastore, tag the datastore first. Figure 26 is a screenshot that shows the **VSP5500-Gold-NVMe** tag that was created on the vSphere client for validation.

**Figure 26**



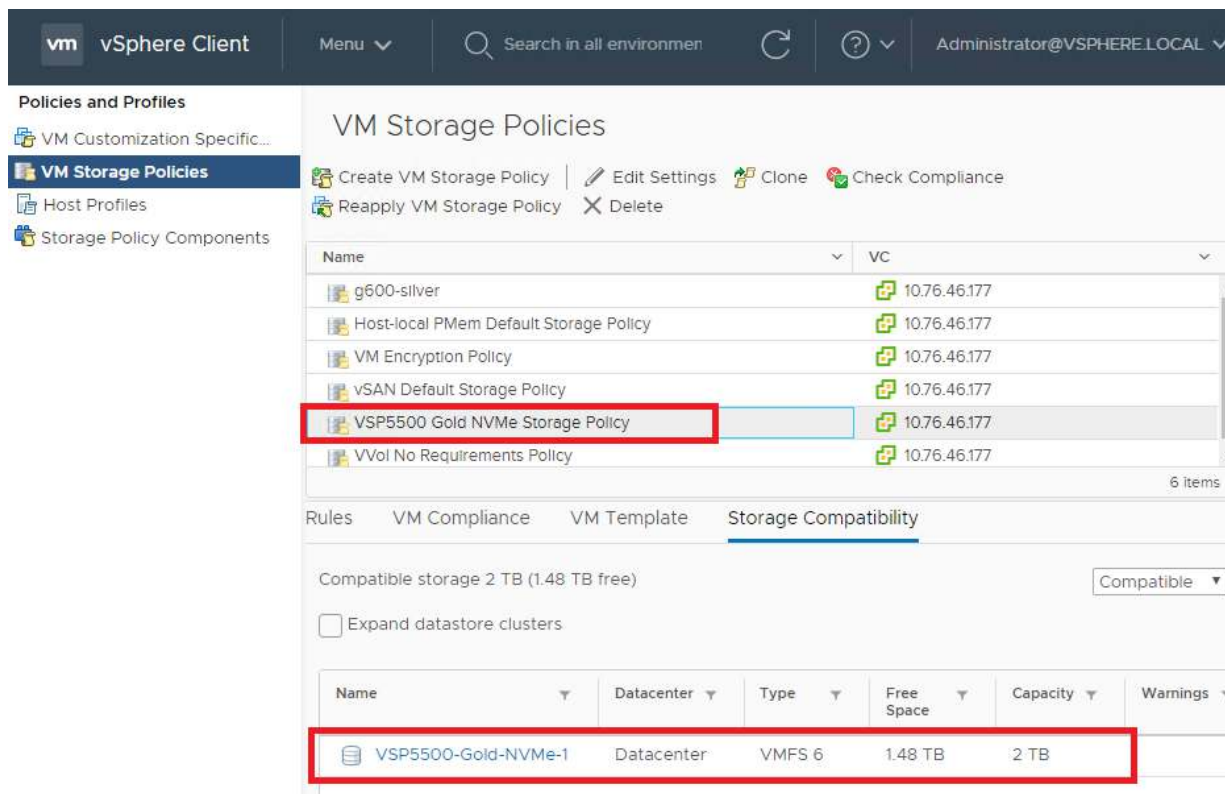
7. From the target datastore, assign a tag. Figure 27 shows assigning a tag to the **VSP5500-Gold-NVMe-1** datastore.

**Figure 27**



8. Create a VM storage policy. Figure 28 shows the **VSP5500 Gold NVMe Storage Policy** created with the **VSP5500-Gold-NVMe-1** datastore.

**Figure 28**



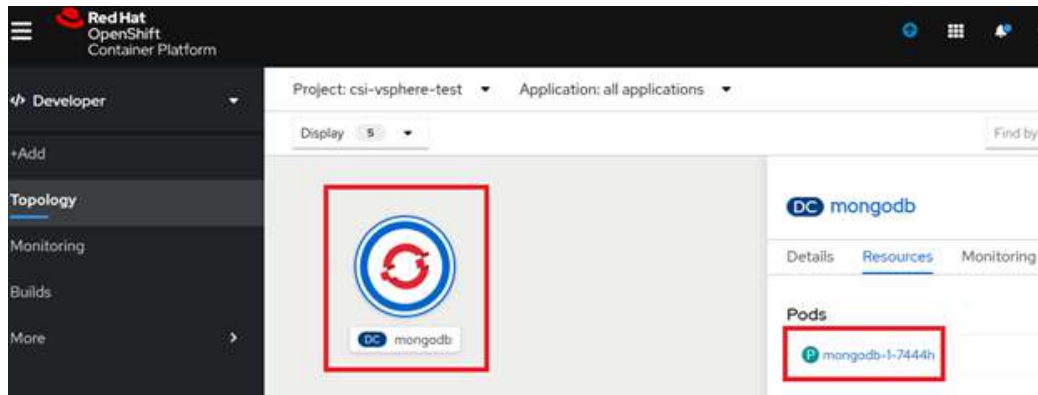
9. Use the `sc-vsp5k-nvme.yaml` file to create a storage class.

**Figure 29**

```
kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
  name: sc-vsp5k-nvme
  annotations:
    storageclass.kubernetes.io/is-default-class: 'true'
provisioner: csi.vsphere.vmware.com
parameters:
  StoragePolicyName: "VSP5500 Gold NVMe Storage Policy"
```

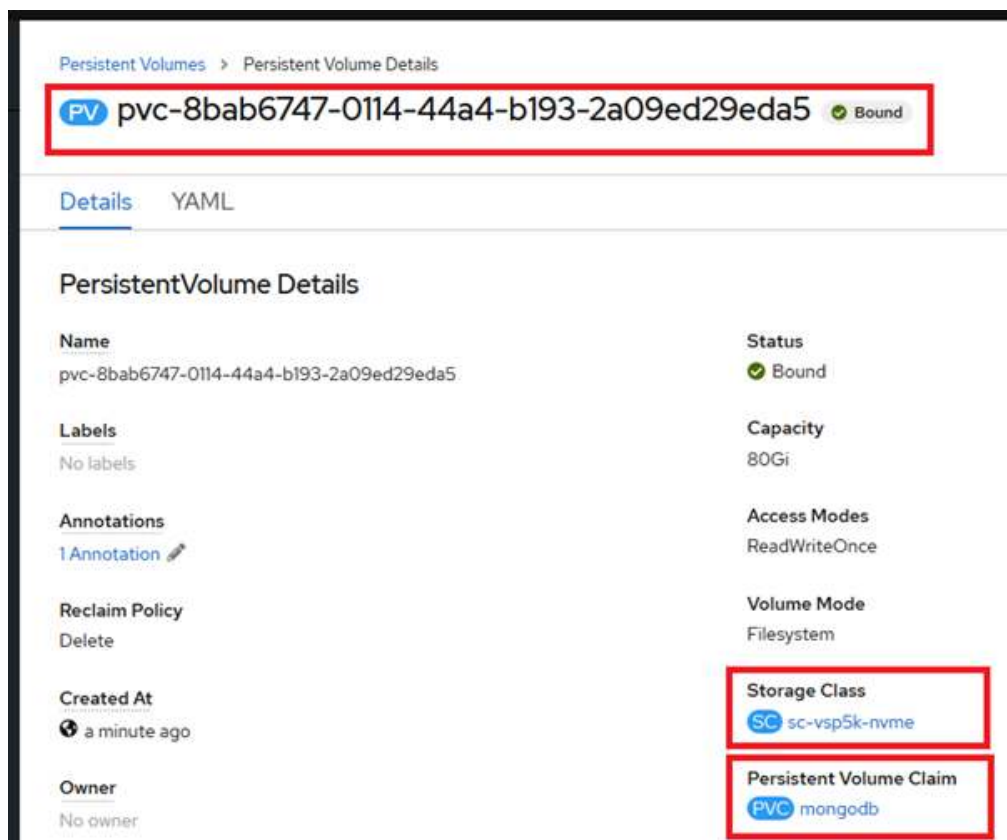
10. From the OpenShift Container Platform web console Developer Catalog, deploy a MongoDB stateful application.
  - Figure 30 is a screenshot that shows the mongodb stateful container was successfully deployed and was running during validation.

**Figure 30**



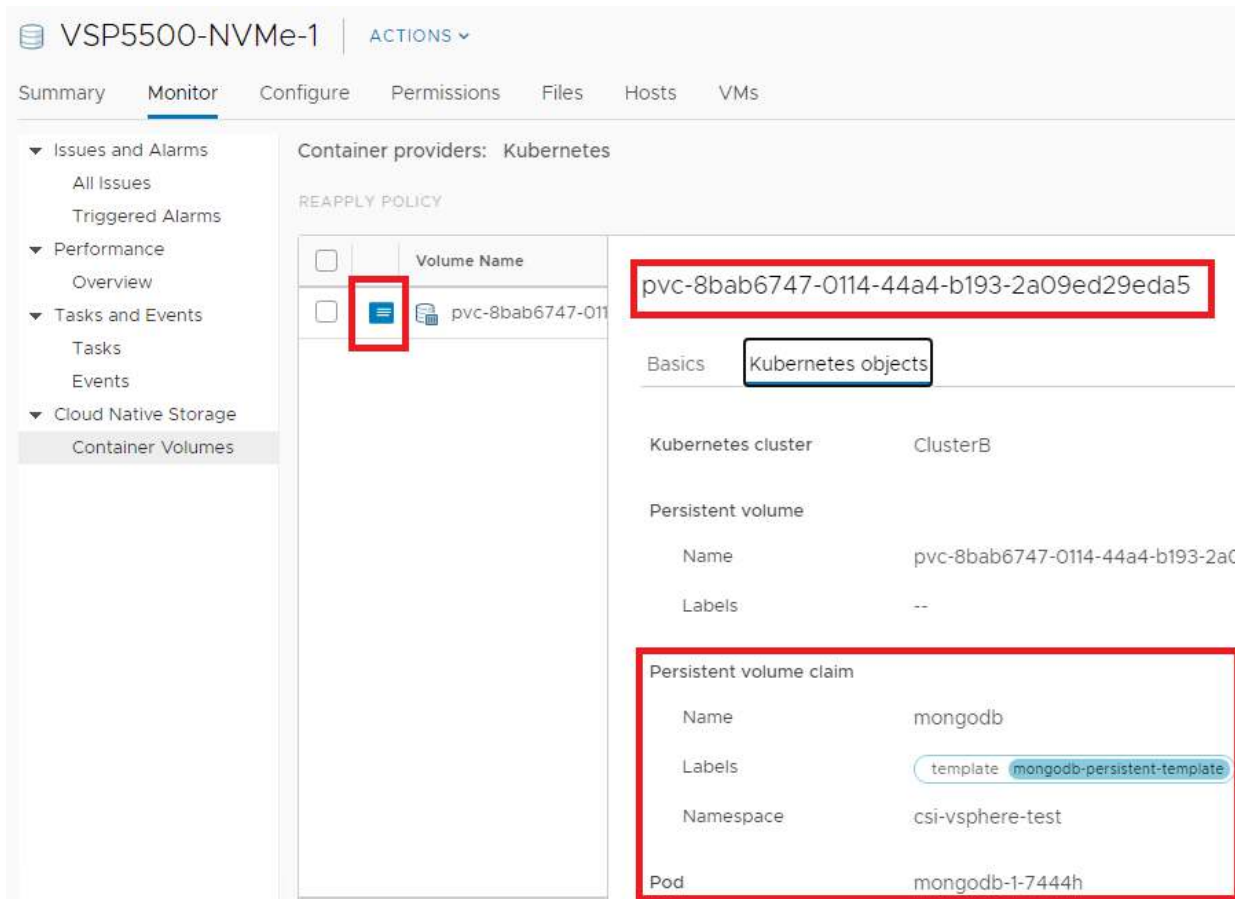
- This screenshot shows selecting the mongodb pod displayed the persistent volume information during validation (Figure 31).

**Figure 31**



- Figure 32 shows that in vCenter, the same persistent volume was found in the **Cloud Native Storage/Container Volumes**.

**Figure 32**

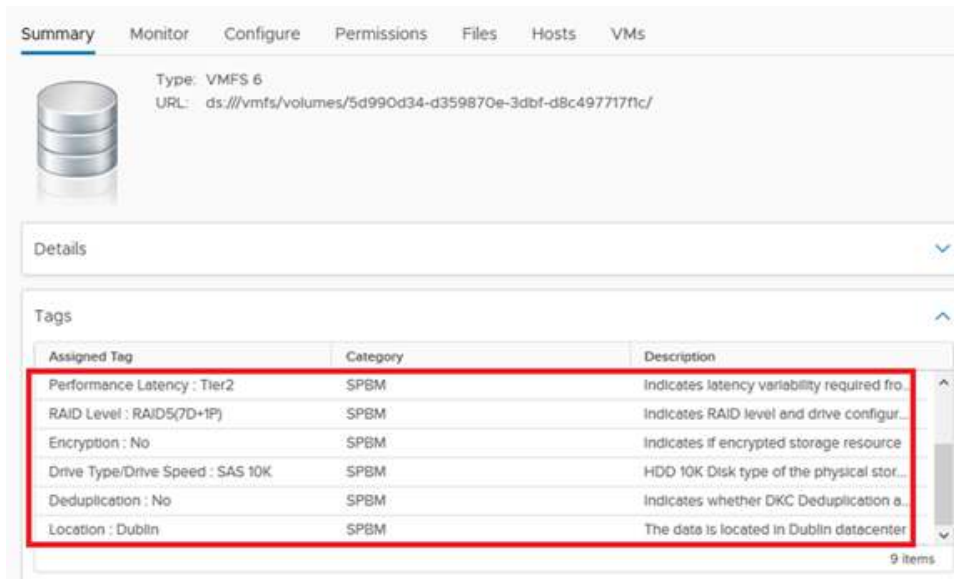


## Automatic VMFS Datastore Tagging with Hitachi Storage Provider for VMware vCenter

The previous test validated tagging the datastore manually in the VMware vSphere client.

Alternatively, with Hitachi Storage Provider for VMware vCenter, datastores can be tagged automatically as shown in Figure 33 on page 36. These tags describe the datastore capability in greater details, including performance level, RAID level, disk type, location, and more. This provides a granular and effective storage management tool in a large VMware environment.

**Figure 33**



## Product Descriptions

This is information about the hardware and software components used in this converged solution for Red Hat OpenShift Container Platform on Hitachi Unified Compute Platform CI.

## Hardware Components

These are the hardware components available for Hitachi Unified Compute Platform CI.

### Hitachi Advanced Server DS120

Optimized for performance, high density, and power efficiency in a dual-processor server, [Hitachi Advanced Server DS120](#) delivers a balance of compute and storage capacity. This 1U rack mounted server has the flexibility to power a wide range of solutions and applications.

The highly scalable memory supports up to 3 TB RAM using 24 slots of 2666 MHz DDR4 RDIMM. Advanced Server DS120 is powered by the Intel Xeon Scalable processor family for complex and demanding workloads. There are flexible OCP and PCIe I/O expansion card options available. This server supports up to 12 small form factor storage devices with up to 4 NVMe drives.

### Hitachi Advanced Server DS220

With a combination of two Intel Xeon Scalable processors and high storage capacity in a 2U rack-space package, [Hitachi Advanced Server DS220](#) delivers the storage and I/O to meet the needs of converged solutions and high-performance applications in the data center.

The Intel Xeon Scalable processor family is optimized to address the growing demands on today's IT infrastructure. The server provides 24 slots for high-speed DDR4 memory, allowing up to 3 TB of memory per node when 128 GB DIMMs are used. This server supports up to 12 large form factor storage devices and an additional 2 small form factor storage devices.

This server has three storage configuration options:

- 12 large form factor storage devices and an additional 2 small form factor storage devices in the back of the chassis
- 16 SAS or SATA drives, 8 NVMe drives, and an additional 2 small form factor storage devices in the back of the chassis
- 24 SFF devices and an additional 2 SFF storage devices in the back of the chassis

## Hitachi Advanced Server DS225

Choose [Hitachi Advanced Server DS225](#) to ensure you have the flexibility and performance you need to support your business-critical enterprise applications.

Advanced Server DS225 delivers compute density and efficiency to meet the needs of your most demanding high-performance applications. It takes full advantage of the Intel Xeon scalable processor family with up to four dual-width 300 W graphic accelerator cards, up to 3 TB memory capacity, and additional PCIe 3.0 expansion slots in a 2U rack space package.

Front-side accessible storage bays supports up to eight hot-pluggable, serial-attached SCSI (SAS) or serial-ATA (SATA) devices. These bays also support flexible configuration, which allows Advanced Server DS225 to deliver high I/O performance and high capacity.

## Hitachi Advanced Server DS240

Meet the needs of your most demanding high-performance applications with [Hitachi Advanced Server DS240](#). With up to four Intel Xeon Scalable Processors and up to 6 TB memory capacity in a 2U rack-space package, this server delivers unparalleled compute density and efficiency.

The Advanced Server DS240 architecture takes full advantage of the groundbreaking Intel Xeon Scalable Processor family, including the highest performance options, to address the growing demands of your IT infrastructure.

## Hitachi Virtual Storage Platform 5000 Series

This enterprise-class, flash array evolution storage, [Hitachi Virtual Storage Platform 5000](#) series (VSP) has an innovative, scale-out design optimized for NVMe and storage class memory. It achieves the following:

- **Agility using NVMe** – Speed, massive scaling with no performance slowdowns, intelligent tiering, and efficiency.
- **Resilience** – Superior application availability and flash resilience. Your data is always available, mitigating business risk.
- **Storage simplified** – Do more with less, integrate AI (artificial intelligence) and ML (machine learning), simplify management, and save money and time with consolidation.

The Virtual Storage Platform 5000 series starts as small as 3.8 TB and scales up to 69 PB of raw capacity and 21 million IOPS of performance. This allows for massive consolidation of workloads for cost savings.

Our patented Hitachi Accelerated Fabric allows Hitachi Storage Virtualization Operating System RF (SVOS RF) to offload I/O traffic between blocks. It uses an architecture that provides immediate processing power without wait time or interruption to maximize I/O throughput. As a result, your applications suffer no latency increases since access to data is accelerated between nodes, even when you scale your system out.

Your environment can use either Virtual Storage Platform 5100 or VSP 5500.



## Hitachi Virtual Storage Platform E990

[Hitachi Virtual Storage Platform E990](#) supercharges business application performance with all-NVMe storage. It uses Hitachi Ops Center, so you can improve IT operations with the latest AI and ML capabilities. Advanced data reduction in Virtual Storage Platform E990 enables you to run data reduction with even the most performance hungry applications.

The all-NVMe architecture in Virtual Storage Platform E990 delivers consistent, low-microsecond latency to reduce latency costs for critical applications. This predictable performance optimizes storage resources.

With Virtual Storage Platform E990 and the rest of Hitachi's midrange storage family, you have agile and automated data center technology. These systems allow you to cost-effectively meet your current digital expectations and give you the ability to address future challenges, as your application data needs and service levels evolve. With time-tested, proven availability and scalability, Hitachi Vantara delivers infrastructure solutions that help you maximize your data center advantage.

## Hitachi Virtual Storage Platform F Series Storage

Use [Hitachi Virtual Storage Platform F series storage systems](#) for a flash-powered cloud platform for your mission critical applications. This storage meets demanding performance and uptime business needs. Extremely scalable, its 4.8 million random read IOPS allows you to consolidate more applications for more cost savings.

## Hitachi Virtual Storage Platform G Series Storage

The [Hitachi Virtual Storage Platform G series storage systems](#) enable the seamless automation of the data center. It has a broad range of efficiency technologies that deliver maximum value while making ongoing costs more predictable. You can focus on strategic projects and consolidating more workloads while using a wide range of media choices.

## Arista Data Center Switches

[Arista Networks](#) builds software-driven cloud networks for data center, cloud, and campus environments. Arista delivers efficient, reliable, and high-performance Universal Cloud Network architectures, based on 10 GbE, 25 GbE, 40 GbE, 50 GbE, and 100 GbE platforms delivered with an extensible operating system – Arista EOS.

- [Arista 7050CX3-32S](#) is a 1RU-sized spine switch with 32 (downlink) and 4 (uplink) 100 GbE QSFP ports for multiple-rack solutions. Each QSFP port supports a choice of five speeds, with flexible configuration between 100 GbE, 40 GbE, 4 × 10 GbE, 4 × 25 GbE, or 2 × 50 GbE modes.
- [Arista 7050SX3-48YC8](#) is a 1RU-sized switch with 48 × 25 GbE SFP and 8 × 100 GbE QSFP ports. The high density SFP ports can be configured in groups of 4 to run either at 25 GbE or a mix of 10 GbE/1 GbE speeds. The QSFP ports allow 100 GbE or 40 GbE high speed network uplinks.
- [Arista 7010T](#) is a 1RU-sized, 48-port 1 GbE management switch for single-rack and multiple-rack solutions.

## Cisco Nexus Data Center Switches

[Cisco Nexus data center switches](#) are built for scale, industry-leading automation, programmability, and real-time visibility.

This solution uses the following Cisco switches to provide Ethernet connectivity:

- [Cisco Nexus 9332C](#) is a 1U-sized spine switch with 32 ports 40/100 GbE for multiple-rack solutions. Uplink ports also support 4 × 10 GbE or 4 × 25 GbE breakouts, if needed.
- [Cisco Nexus 93180YC-FX](#) is a 1U-sized top-of-rack or leaf switch with 48 ports 10/25 GbE (downlink) and 6 ports 40/100 GbE (uplink) for single-rack and multiple-rack solutions.
- [Cisco Nexus 92348GC-X](#) is a 1U-sized management switch with 48 ports 1 GbE, 4 ports 1/10/25 GbE and 2 ports 40/100 GbE for single-rack and multiple-rack solutions.

## Brocade Fibre Channel Switches from Broadcom

[Brocade Fibre Channel switches](#) deliver industry-leading performance, simplifying scale-out network architectures. Get the high-performance, availability, and ease of management you need for a solid foundation to grow the storage network you want.

- [Broadcom G620](#) is a 1U-sized QSFP Fibre Channel switch with 48 ports 16/32 Gb/s and 4 ports 128 Gb/s for multiple-rack solutions.

## Software Components

These are the software components used in this reference architecture.

### Hitachi Storage Virtualization Operating System RF

[Hitachi Storage Virtualization Operating System RF](#) (SVOS RF) spans and integrates multiple platforms. It integrates storage system software to provide system element management and advanced storage system functions. Used across multiple platforms, Storage Virtualization Operating System includes storage virtualization, thin provisioning, storage service level controls, dynamic provisioning, and performance instrumentation.

Storage Virtualization Operating System RF uses Hitachi Dynamic Provisioning (HDP) to provide wide striping and thin provisioning. Dynamic Provisioning provides one or more wide-striping pools across many RAID groups. Each pool has one or more dynamic provisioning virtual volumes (DP-VOLs) without initially allocating any physical space. Deploying Dynamic Provisioning avoids the routine issue of hot spots that occur on logical devices (LDEVs).

Storage Virtualization Operating System RF has been optimized for Hitachi Virtual Storage Platform 5000 series for the scale-out, NVMe design. It incorporates artificial intelligence and machine learning to reduce costs with intelligent tiering. This automates data placement to assure that your data always resides on the most optimized tier.

### Hitachi Unified Compute Platform Advisor

[Hitachi Unified Compute Platform Advisor](#) (UCP Advisor) brings simplified IT administration to virtualized, converged, and hyperconverged systems from Hitachi. Unified Compute Platform Advisor supports guided life-cycle management to the server, network, and storage elements within supported Unified Compute Platform systems.

Unified Compute Platform Advisor is used to discover and provision servers initially, and later to manage the compute nodes:

- Identify Unified Compute Platform servers for remote management.
- Provision servers.
- Image the custom BIOS settings on the server
- Install the operating system.
- Upgrade the installed firmware,
- Power cycle a compute node remotely.
- Launch a remote console for a server.
- Provides remote access to general system information.

## Hitachi Storage Provider for VMware vCenter

When you want to support policy-based automation and improve operational insight into the storage or converged platform hosting that environment, use [Hitachi Storage Provider for VMware vCenter](#). This allows a unique implementation of VMware vSphere API for Storage Awareness (VASA), supporting traditional-based datastores (VMFS and NFS) and VMware vVols-based datastores.

Storage Provider for VMware vCenter, as part of the infrastructure, communicates with VMware vCenter to indicate storage capabilities and state information. It supports policy-based management, operations management, and resource scheduling functionality.

## Hitachi Storage Plug-in for Containers

Hitachi Storage Plug-in for Containers provides connectivity between Docker, Kubernetes, or Kubernetes Container Storage Interface (CSI) containers and Hitachi Virtual Storage Platform 5000 series, G series, and F series systems. With the compatibility plug-in, your organization can deliver shared storage for containers that persists beyond the timeline of a single container host.

## Red Hat OpenShift

[Red Hat OpenShift](#) offers automated installation, upgrades, and lifecycle management throughout the container stack—the operating system, Kubernetes and cluster services, and applications—on any cloud.

## Red Hat Enterprise Linux

[Red Hat Enterprise Linux](#) delivers military-grade security, 99.999% uptime, support for business-critical workloads, and so much more. Ultimately, the platform helps you reallocate resources from maintaining the status quo to tackling new challenges.

## VMware vSphere

[VMware vSphere](#) is a virtualization platform that provides a datacenter infrastructure. It helps you get the best performance, availability, and efficiency from your infrastructure and applications. Virtualize applications with confidence using consistent management.

VMware vSphere has the following components:

- [VMware vSphere ESXi](#)  
This hypervisor loads directly on a physical server. ESXi provides a robust, high-performance virtualization layer that abstracts server hardware resources and makes them shareable by multiple virtual machines.
- [VMware vCenter Server](#)  
This management software provides a centralized platform for managing your VMware vSphere environments so you can automate and deliver a virtual infrastructure with confidence:
  - VMware vSphere vMotion
  - VMware vSphere Storage vMotion
  - VMware vSphere Distributed Resource Scheduler
  - VMware vSphere High Availability
  - VMware vSphere Fault Tolerance

# VMware vSphere CPI/CSI Setup Files for Red Hat OpenShift Container Platform 4

This is the VMware vSphere CSI installation used for Red Hat OpenShift Container Platform hybrid cluster configuration. While it is a similar installation procedure for native Kubernetes, some YAML files needed to be modified to fit the OpenShift Container Platform environment. The vSphere CSI installation method for native Kubernetes can be found in [Kubernetes vSphere Cloud Provider](#).

Make sure to add all bare metal nodes into the OpenShift Container Platform hybrid cluster and install the Hitachi Storage Plug-in for Containers CSI operator before installing vSphere CSI.

The following procedure was used to install vSphere CSI.

The install directory is the following: /home/ocpinstall/ocp-upi-install/

To install VMware vSphere CSI, do the following.

1. On a Red Hat Enterprise Linux administrator workstation, configure govc to access the target vSphere cluster.
2. Taint all the nodes for CPI install with following command:

```
oc adm taint nodes --selector='!node-role.kubernetes.io/master' node.cloudprovider.kubernetes.io/uninitialized=true:NoSchedule
```

3. Create the vsphere.conf and cpi-global-secret.yaml files for CPI.

- vsphere.conf

```
[Global]  
  
port = "443"  
  
insecure-flag = "true"  
  
secret-namespace = "kube-system"  
  
secret-name = "cpi-global-secret"  
  
[VirtualCenter "<IP or RQDN of vCenter>"]  
  
datacenters = "yourDatacenter"
```

- cpi-global-secret.yaml

```
apiVersion: v1  
  
kind: Secret  
  
metadata:  
  
name: cpi-global-secret  
  
namespace: kube-system  
  
stringData:  
  
<yourVcenterIP>.username: "administrator@vsphere.local"  
  
<yourVcenterIP>.password: "<yourPassword>"
```

4. Create the configmap and secret in OpenShift Container Platform for CPI.

```
oc create configmap cloud-config --from-file=/home/ocpinstall/ocp-upi-install/vsphere.conf --  
namespace=kube-system
```

```
oc create -f /home/ocpinstall/ocp-upi-install/cpi-global-secret.yaml
```

5. Create the YAML files for roles/rolebindings for CPI.

- cpi-roles.yaml

```
apiVersion: v1
```

```
items:
```

```
- apiVersion: rbac.authorization.k8s.io/v1
```

```
kind: ClusterRole
```

```
metadata:
```

```
name: system:cloud-controller-manager
```

```
rules:
```

```
- apiGroups:
```

```
- ""
```

```
resources:
```

```
- events
```

```
verbs:
```

```
- create
```

```
- patch
```

```
- update
```

```
- apiGroups:
```

```
- ""
```

```
resources:
```

```
- nodes
```

```
verbs:
```

```
- '*'
```

```
- apiGroups:
```

```
- ""
```

```
resources:
```

```
- nodes/status
```

**verbs:**

- patch

**- apiGroups:**

- ""

**resources:**

- services

**verbs:**

- list

- patch

- update

- watch

**- apiGroups:**

- ""

**resources:**

- services/status

**verbs:**

- patch

**- apiGroups:**

- ""

**resources:**

- serviceaccounts

**verbs:**

- create

- get

- list

- watch

- update

**- apiGroups:**

- ""

**resources:**

- persistentvolumes

```
verbs:
- get
- list
- update
- watch
- apiGroups:
- ""
resources:
- endpoints
verbs:
- create
- get
- list
- watch
- update
- apiGroups:
- ""
resources:
- secrets
verbs:
- get
- list
- watch
kind: List
metadata: {}
```



- `apiVersion: v1`  
`items:`
  - `apiVersion: rbac.authorization.k8s.io/v1`  
`kind: RoleBinding`  
`metadata:`
    - `name: servicecatalog.k8s.io:apiserver-authentication-reader`
    - `namespace: kube-system``roleRef:`
    - `apiGroup: rbac.authorization.k8s.io`
    - `kind: Role`
    - `name: extension-apiserver-authentication-reader``subjects:`
    - `apiGroup: ""`  
`kind: ServiceAccount`  
`name: cloud-controller-manager`  
`namespace: kube-system`
    - `apiGroup: ""`  
`kind: User`  
`name: cloud-controller-manager`
  - `apiVersion: rbac.authorization.k8s.io/v1`  
`kind: ClusterRoleBinding`  
`metadata:`
    - `name: system:cloud-controller-manager``roleRef:`
    - `apiGroup: rbac.authorization.k8s.io`
    - `kind: ClusterRole`
    - `name: system:cloud-controller-manager`

```
subjects:
- kind: ServiceAccount
  name: cloud-controller-manager
  namespace: kube-system
- kind: User
  name: cloud-controller-manager
kind: List
metadata: {}
```

■ cpi-daemonset.yaml

```
---
apiVersion: v1
kind: ServiceAccount
metadata:
  name: cloud-controller-manager
  namespace: kube-system
---
apiVersion: apps/v1
kind: DaemonSet
metadata:
  name: vsphere-cloud-controller-manager
  namespace: kube-system
  labels:
    k8s-app: vsphere-cloud-controller-manager
spec:
  selector:
    matchLabels:
      k8s-app: vsphere-cloud-controller-manager
  updateStrategy:
    type: RollingUpdate
```

```
template:
  metadata:
    labels:
      k8s-app: vsphere-cloud-controller-manager
  spec:
    nodeSelector:
      node-role.kubernetes.io/master: ""
    securityContext:
      runAsUser: 0
    tolerations:
      - key: node.cloudprovider.kubernetes.io/uninitialized
        value: "true"
        effect: NoSchedule
      - key: node-role.kubernetes.io/master
        effect: NoSchedule
    serviceName: cloud-controller-manager
    containers:
      - name: vsphere-cloud-controller-manager
        image: gcr.io/cloud-provider-vsphere/cpi/release/manager:latest
        args:
          - --v=2
          - --cloud-provider=vsphere
          - --cloud-config=/etc/cloud/vsphere.conf
        volumeMounts:
          - mountPath: /etc/cloud
            name: vsphere-config-volume
            readOnly: true
        resources:
          requests:
            cpu: 200m
    hostNetwork: true
```

```

volumes:
  - name: vsphere-config-volume

  configMap:
    name: cloud-config
---
apiVersion: v1
kind: Service
metadata:
  labels:
    component: cloud-controller-manager
    name: vsphere-cloud-controller-manager
    namespace: kube-system
spec:
  type: NodePort
  ports:
    - port: 43001
      protocol: TCP
      targetPort: 43001
  selector:
    component: cloud-controller-manager

```

6. Create the clusterroles, clusterrolebindings, service account privileges, and CPI daemonset.

```

oc create -f /home/ocpininstall/ocp-upi-install/cpi-roles.yaml
oc create -f /home/ocpininstall/ocp-upi-install/cpi-role-bindings.yaml
oc adm policy add-scc-to-user privileged -z cloud-controller-manager
oc create -f /home/ocpininstall/ocp-upi-install/cpi-daemonset.yaml

```

7. Set the ProviderID on the master nodes manually since this is not running with cloud-provider=vsphere.

- `setprovid.sh`

```
#!/bin/bash

export KUBECONFIG=/home/ocpinstall/ocp-upi-install/auth/kubeconfig

oc login -u cluster_admin -p <yourPassword>

export GOVC_USERNAME='administrator@vsphere.local'

export GOVC_INSECURE=1

export GOVC_PASSWORD='<yourPassword>'

export GOVC_URL='<vcenterIP-FQDN>'

DATACENTER='DatacenterB'

VM_PREFIX='ocp4-master'

IFS=$'\n'

for vm in $(govc ls "/$DATACENTER/vm" | grep $VM_PREFIX); do

    MACHINE_INFO=$(govc vm.info -json -dc=$DATACENTER -vm.ipath="/$vm" -e=true)

    VM_NAME=$(jq -r '.VirtualMachines[] | .Name' <<< $MACHINE_INFO | awk '{print tolower($0)}')

    # UUIDs come in lowercase, upper case them

    VM_UUID=$(jq -r '.VirtualMachines[] | .Config.Uuid' <<< $MACHINE_INFO | awk '{print toupper($0)}')

    echo "Patching $VM_NAME with UUID:$VM_UUID"

    oc patch node $VM_NAME -p "{\"spec\":{\"providerID\":\"vsphere://$VM_UUID\"}}"

done
```

8. Execute `setprovid.sh`.

```
chmod 777 setprovid.sh

./setprovid.sh
```

9. Create the vsphere.conf, RBAC, deployment, and daemonset YAML (CSI).

- csi-vsphere.conf

```
[Global]  
  
cluster-id = "<yourCluster>"  
  
[VirtualCenter "vcenterIP-FQDN"]  
  
insecure-flag = "true"  
  
user = "administrator@vsphere.local"  
  
password = "<yourPassword>"  
  
port = "443"  
  
datacenters = "<yourDatacenter>"
```

- csi-rbac.yaml

```
kind: ServiceAccount  
  
apiVersion: v1  
  
metadata:  
  
  name: vsphere-csi-controller  
  
  namespace: kube-system  
  
---  
  
kind: ClusterRole  
  
apiVersion: rbac.authorization.k8s.io/v1  
  
metadata:  
  
  name: vsphere-csi-controller-role  
  
rules:  
  
  - apiGroups: [""]  
  
    resources: ["nodes", "persistentvolumeclaims", "pods"]  
  
    verbs: ["get", "list", "watch"]  
  
  - apiGroups: [""]  
  
    resources: ["persistentvolumeclaims/status"]  
  
    verbs: ["update", "patch"]
```

- apiGroups: [""]
- resources: ["persistentvolumes"]
- verbs: ["get", "list", "watch", "create", "update", "delete", "patch"]
- apiGroups: [""]
- resources: ["events"]
- verbs: ["get", "list", "watch", "create", "update", "patch"]
- apiGroups: ["coordination.k8s.io"]
- resources: ["leases"]
- verbs: ["get", "watch", "list", "delete", "update", "create"]
- apiGroups: ["storage.k8s.io"]
- resources: ["storageclasses", "csinodes"]
- verbs: ["get", "list", "watch"]
- apiGroups: ["storage.k8s.io"]
- resources: ["volumeattachments"]
- verbs: ["get", "list", "watch", "update", "patch"]

---

**kind:** ClusterRoleBinding

**apiVersion:** rbac.authorization.k8s.io/v1

**metadata:**

name: vsphere-csi-controller-binding

**subjects:**

- kind: ServiceAccount
- name: vsphere-csi-controller
- namespace: kube-system

**roleRef:**

kind: ClusterRole

name: vsphere-csi-controller-role

apiGroup: rbac.authorization.k8s.io



- `csi-controller-deploy.yaml`
  - # Minimum Kubernetes version - 1.16**
  - # For prior releases make sure to add required --feature-gates flags**
  - kind: Deployment**
  - apiVersion: apps/v1**
  - metadata:**
    - name: vsphere-csi-controller**
    - namespace: kube-system**
  - spec:**
    - replicas: 1**
    - strategy:**
      - type: RollingUpdate**
      - rollingUpdate:**
        - maxUnavailable: 1**
        - maxSurge: 0**
    - selector:**
      - matchLabels:**
        - app: vsphere-csi-controller**
    - template:**
      - metadata:**
        - labels:**
          - app: vsphere-csi-controller**
          - role: vsphere-csi**
      - spec:**
        - serviceAccountName: vsphere-csi-controller**
        - nodeSelector:**
          - node-role.kubernetes.io/master: ""**

**tolerations:**

- operator: "Exists"

effect: NoSchedule

- operator: "Exists"

effect: NoExecute

**dnsPolicy: "Default"**

**containers:**

- name: csi-attacher

image: quay.io/k8scsi/csi-attacher:v2.0.0

args:

- "--v=4"

- "--timeout=300s"

- "--csi-address=\$(ADDRESS)"

- "--leader-election"

env:

- name: ADDRESS

value: /csi/csi.sock

**volumeMounts:**

- mountPath: /csi

name: socket-dir

- name: vsphere-csi-controller

image: gcr.io/cloud-provider-vsphere/csi/release/driver:v2.0.0

**lifecycle:**

preStop:

exec:

command: ["/bin/sh", "-c", "rm -rf /var/lib/kubelet/plugins\_registry/csi.vsphere.vmware.com"]

**imagePullPolicy: "Always"**

**env:**

- name: CSI\_ENDPOINT  
value: unix:///var/lib/kubelet/plugins\_registry/csi.sock
- name: X\_CSI\_MODE  
value: "controller"
- name: VSPHERE\_CSI\_CONFIG  
value: "/etc/cloud/csi-vsphere.conf"
- name: LOGGER\_LEVEL  
value: "PRODUCTION" # Options: DEVELOPMENT, PRODUCTION

**volumeMounts:**

- mountPath: /etc/cloud  
name: vsphere-config-volume  
readOnly: true
- mountPath: /var/lib/kubelet/plugins\_registry/  
name: socket-dir

**ports:**

- name: healthz  
containerPort: 9808  
protocol: TCP

**livenessProbe:**

- httpGet:
  - path: /healthz
  - port: healthz
- initialDelaySeconds: 10
- timeoutSeconds: 3
- periodSeconds: 5
- failureThreshold: 3

- name: liveness-probe  
image: quay.io/k8scsi/livenessprobe:v1.1.0

```

args:
  - "--csi-address=$(ADDRESS)"
env:
  - name: ADDRESS
    value: /var/lib/kubelet/plugins_registry/csi.sock
volumeMounts:
  - mountPath: /var/lib/kubelet/plugins_registry/
    name: socket-dir
- name: vsphere-syncer
  image: gcr.io/cloud-provider-vsphere/csi/release/syncer:v2.0.0
  args:
    - "--leader-election"
  imagePullPolicy: "Always"
  env:
    - name: FULL_SYNC_INTERVAL_MINUTES
      value: "30"
    - name: VSPHERE_CSI_CONFIG
      value: "/etc/cloud/csi-vsphere.conf"
    - name: LOGGER_LEVEL
      value: "PRODUCTION" # Options: DEVELOPMENT, PRODUCTION
  volumeMounts:
    - mountPath: /etc/cloud
      name: vsphere-config-volume
      readOnly: true
- name: csi-provisioner
  image: quay.io/k8scsi/csi-provisioner:v1.4.0

```

```

args:
  - "--v=4"
  - "--timeout=300s"
  - "--csi-address=$(ADDRESS)"
  - "--feature-gates=Topology=true"
  - "--strict-topology"
  - "--enable-leader-election"
  - "--leader-election-type=leases"
env:
  - name: ADDRESS
    value: /csi/csi.sock
volumeMounts:
  - mountPath: /csi
    name: socket-dir
volumes:
  - name: vsphere-config-volume
secret:
  secretName: vsphere-config-secret
  secretNamespace: kube-system
  - name: socket-dir
  hostPath:
    path: /var/lib/kubelet/plugins_registry/csi.vsphere.vmware.com
    type: DirectoryOrCreate
---
apiVersion: storage.k8s.io/v1beta1
kind: CSIDriver
metadata:
  name: csi.vsphere.vmware.com
spec:
  attachRequired: true
  podInfoOnMount: false

```

- `csi-daemonset.yaml`
  - # Minimum Kubernetes version - 1.16**
  - # For prior releases make sure to add required --feature-gates flags**
  - kind: DaemonSet**
  - apiVersion: apps/v1**
  - metadata:**
    - name: vsphere-csi-node**
    - namespace: kube-system**
  - spec:**
    - selector:**
      - matchLabels:**
        - app: vsphere-csi-node**
    - updateStrategy:**
      - type: "RollingUpdate"**
    - template:**
      - metadata:**
        - labels:**
          - app: vsphere-csi-node**
          - role: vsphere-csi**
      - spec:**
        - dnsPolicy: "Default"**
        - serviceAccountName: vsphere-csi-controller**
        - # Only install this csi on VM nodes**
        - nodeSelector:**
          - node-type: vm-cns**
        - containers:**
          - name: node-driver-registrar**
            - image: quay.io/k8scsi/csi-node-driver-registrar:v1.2.0**

```

lifecycle:
  preStop:
    exec:
      command: ["/bin/sh", "-c", "rm -rf /registration/csi.vsphere.vmware.com-reg.sock /csi/csi.sock"]
  args:
    - "--v=5"
    - "--csi-address=$(ADDRESS)"
    - "--kubelet-registration-path=$(DRIVER_REG_SOCK_PATH)"
  env:
    - name: ADDRESS
      value: /csi/csi.sock
    - name: DRIVER_REG_SOCK_PATH
      value: /var/lib/kubelet/plugins/csi.vsphere.vmware.com/csi.sock
  securityContext:
    privileged: true
  volumeMounts:
    - name: plugin-dir
      mountPath: /csi
    - name: registration-dir
      mountPath: /registration
    - name: vsphere-csi-node
      image: gcr.io/cloud-provider-vsphere/csi/release/driver:v2.0.0
      imagePullPolicy: "Always"
    env:
      - name: NODE_NAME
        valueFrom:
          fieldRef:
            fieldPath: spec.nodeName
      - name: CSI_ENDPOINT
        value: unix:///csi/csi.sock
      - name: X_CSI_MODE

```



```

    value: "node"

- name: X_CSI_SPEC_REQ_VALIDATION
  value: "false"

# needed only for topology aware setups

#- name: VSPHERE_CSI_CONFIG

  # value: "/etc/cloud/csi-vsphere.conf" # here csi-vsphere.conf is the name of the file used for creating
  # secret using "--from-file" flag

- name: X_CSI_DEBUG
  value: "true"

- name: LOGGER_LEVEL
  value: "PRODUCTION" # Options: DEVELOPMENT, PRODUCTION

securityContext:
  privileged: true

  capabilities:
    add: ["SYS_ADMIN"]

  allowPrivilegeEscalation: true

volumeMounts:

# needed only for topology aware setups

#- name: vsphere-config-volume

# mountPath: /etc/cloud

# readOnly: true

- name: plugin-dir
  mountPath: /csi

- name: pods-mount-dir
  mountPath: /var/lib/kubelet

# needed so that any mounts setup inside this container are

# propagated back to the host machine.

mountPropagation: "Bidirectional"

- name: device-dir
  mountPath: /dev

```

```
ports:
  - name: healthz
    containerPort: 9808
    protocol: TCP
livenessProbe:
  httpGet:
    path: /healthz
    port: healthz
  initialDelaySeconds: 10
  timeoutSeconds: 3
  periodSeconds: 5
  failureThreshold: 3
- name: liveness-probe
  image: quay.io/k8scsi/livenessprobe:v1.1.0
  args:
    - --csi-address=/csi/csi.sock
  volumeMounts:
    - name: plugin-dir
      mountPath: /csi
  volumes:
    # needed only for topology aware setups
    #- name: vsphere-config-volume
    # secret:
    #  secretName: vsphere-config-secret
    - name: registration-dir
      hostPath:
        path: /var/lib/kubelet/plugins_registry
        type: Directory
```

```

- name: plugin-dir
  hostPath:
    path: /var/lib/kubelet/plugins/csi.vsphere.vmware.com/
    type: DirectoryOrCreate
- name: pods-mount-dir
  hostPath:
    path: /var/lib/kubelet
    type: Directory
- name: device-dir
  hostPath:
    path: /dev
tolerations:
- effect: NoExecute
  operator: Exists
- effect: NoSchedule
  operator: Exists

```

10. Create the secret, RBAC, service account privileges, and CSI deployment, and daemonset.

```
oc create secret generic vsphere-config-secret --from-file=/home/ocpininstall/ocp-upi-install/csi-vsphere.conf --namespace=kube-system
```

```
oc create -f /home/ocpininstall/ocp-upi-install/csi-rbac.yaml
```

```
oc adm policy add-scc-to-user privileged -z vsphere-csi-controller
```

```
oc create -f /home/ocpininstall/ocp-upi-install/csi-controller-deploy.yaml
```

```
oc create -f /home/ocpininstall/ocp-upi-install/csi-daemonset.yaml
```

This completes the vSphere CPI/CSI installation on Red Hat OpenShift Container Platform.

## For More Information

Hitachi Vantara Global Services offers experienced storage consultants, proven methodologies and a comprehensive services portfolio to assist you in implementing Hitachi products and solutions in your environment. For more information, see the [Services](#) website.

Demonstrations and other resources are available for many Hitachi products. To schedule a live demonstration, contact a sales representative or partner. To view on-line informational resources, see the [Resources](#) website.

Hitachi Academy is your education destination to acquire valuable knowledge and skills on Hitachi products and solutions. Our Hitachi Certified Professional program establishes your credibility and increases your value in the IT marketplace. For more information, see the Hitachi Vantara [Training and Certification](#) website.

For more information about Hitachi products and services, contact your sales representative, partner, or visit the [Hitachi Vantara](#) website.

## Hitachi Vantara



Corporate Headquarters  
2535 Augustine Drive  
Santa Clara, CA 95054 USA  
[www.HitachiVantara.com](http://www.HitachiVantara.com) | [community.HitachiVantara.com](http://community.HitachiVantara.com)

Regional Contact Information  
USA: 1-800-446-0744  
Global: 1-858-547-4526  
[HitachiVantara.com/contact](http://HitachiVantara.com/contact)

© Hitachi Vantara LLC 2020. All rights reserved. HITACHI is a trademark or registered trademark of Hitachi, Ltd. VSP is a trademark or registered trademark of Hitachi Vantara LLC. All other trademarks, service marks and company names are properties of their respective owners.

Notice: This document is for informational purposes only, and does not set forth any warranty, expressed or implied, concerning any equipment or service offered or to be offered by Hitachi Vantara.

MK-SL-210-01. October 2020.