

# **Deployment Options for Kubernetes Container Applications on Unified Compute Platform CI with Hitachi VSP Series**

Reference Architecture Guide

By Jeff Chen and Henry Chu  
Contributors: Minami Morita, Wataru Ono

December 2019

# 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-182-00	Initial release	December 13, 2019

# Table of Contents

<b>Introduction</b>	<b>1</b>
<b>Solution Overview</b>	<b>2</b>
Bare Metal Container Hosts	4
Virtualized Container Hosts	5
Mixed Hybrid Container Hosts	7
<b>Solution Components</b>	<b>8</b>
Hardware Components	8
Software Components	9
<b>Solution Design</b>	<b>10</b>
Deploy a Kubernetes Cluster on Bare Metal Hosts	13
Deploy a Kubernetes Cluster on VMware vSphere	16
Deploy a Kubernetes Cluster on Mixed Hybrid Environments (VM and Bare Metal Hosts)	21
<b>Solution Validation</b>	<b>23</b>
Create Persistent Volumes for Bare Metal Hosts	23
Create Persistent Volumes from VMware vVol	25
Create Persistent Volumes from a VMFS Datastore	30
Create Persistent Volumes from a vSAN Datastore	35
<b>Conclusion</b>	<b>37</b>
<b>Appendix A - Product Descriptions</b>	<b>37</b>
Hardware Components	37
Software Components	39

# Deployment Options for Kubernetes Container Applications on Unified Compute Platform CI with Hitachi VSP Series

## Reference Architecture Guide

### Introduction

Businesses are under pressure to deal with increased customer demands and increasing competition. Digital transformation is necessary to manage these pressures where agility, modern processes, and architecture are challenges that need to be addressed. 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 has a long history of providing highly available and high-performance infrastructures, from end to end complete converged infrastructures to a build-your-own approach using Hitachi Virtual Storage Platforms (VSP). Some specific challenges of providing an infrastructure for container platform are:

- **Data Persistence** – Data is core for any application. Numerous applications require data persistence for example: Postgres, MongoDB, and MySQL, among others. Continuous Integration/Continuous Delivery (CI/CD) pipelines requires data persistency at every level as well as extending CI/CD pipeline. Hitachi Storage Plug-in for Containers (HSPC) integrates Hitachi Virtual Storage Platform (VSP) with Kubernetes 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 is 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 also provide both 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 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 will only increase operation in-efficiencies. Hinder the ability to increase innovation and agility. Orchestration and automation are the key to operational efficiencies. Hitachi provides 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 Hitachi VSP for storage management.

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

- Hitachi VSP Gx00/VSP Fx00/VSP 5000 for storage
- Hitachi Advanced Server DSxx0 for compute
- Hitachi UCP Advisor for end to end management
- Cisco Nexus 3000 and 9000, or Arista 7000 for Ethernet networking
- Brocade G620 for Fibre Channel SAN

This reference architecture also provides the reference design for a build-your-own container platform where only Hitachi VSP is used. Although a converged system is used as an example, the Hitachi VSP storage reference design still applies to the building-your-own-container platform.

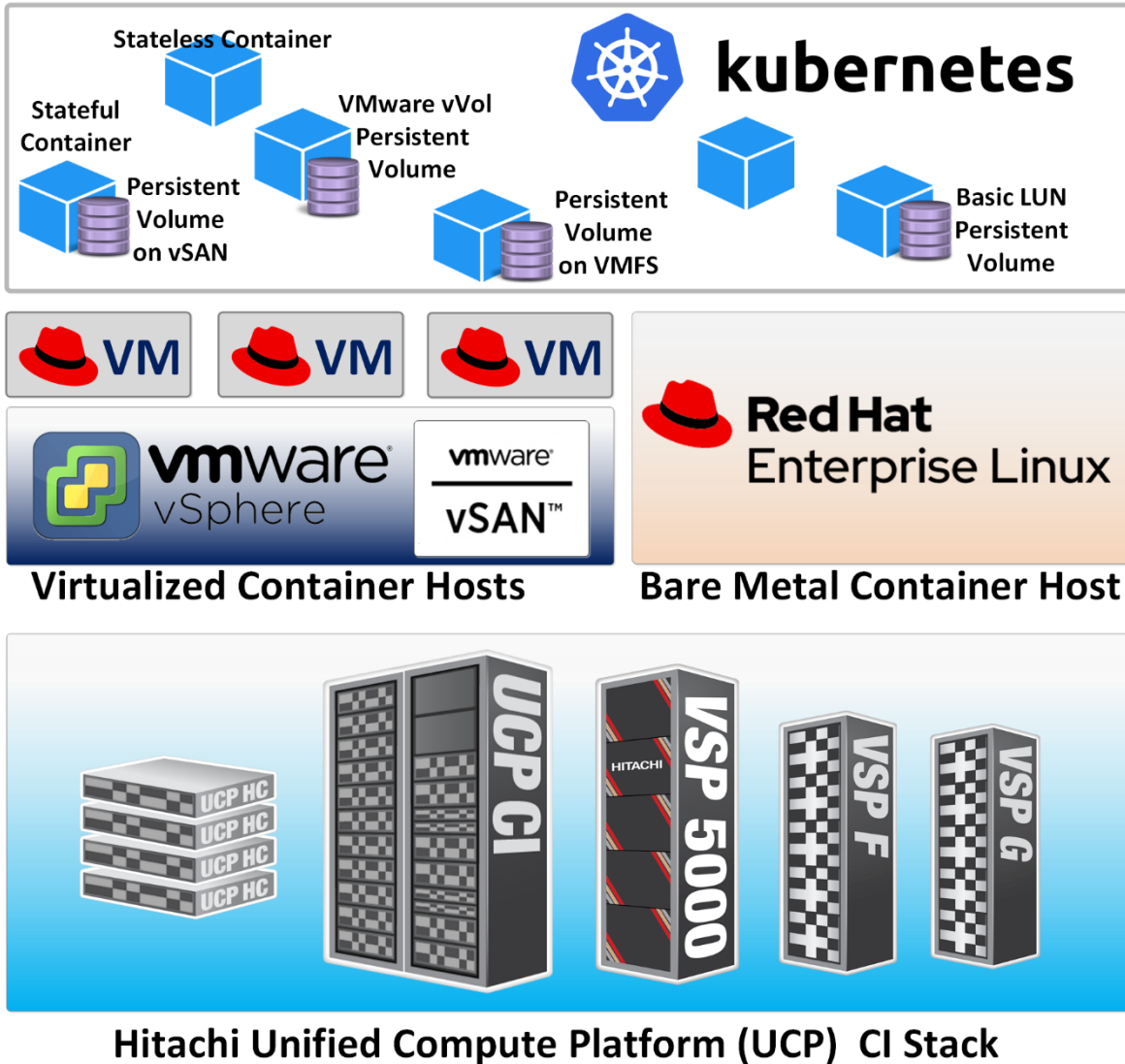
The intended audience of this document is IT administrators, system architects, consultants, and sales engineers to assist in planning, designing, and implementing UCP CI with container solutions.

## **Solution Overview**

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

Kubernetes is a powerful container orchestration platform. Providing a Kubernetes platform is one of the solutions for UCP CI. Figure 1 shows a high-level diagram of Kubernetes managing containers and persistent volumes on the UCI CI stack or with Hitachi VSP systems.

Figure 1



Kubernetes can be deployed on both bare metal hosts or virtual hosts. Depending on the deployment purposes and wide range of personas involved, different deployments can be used. Kubernetes deployment options on UCP CI are listed below.

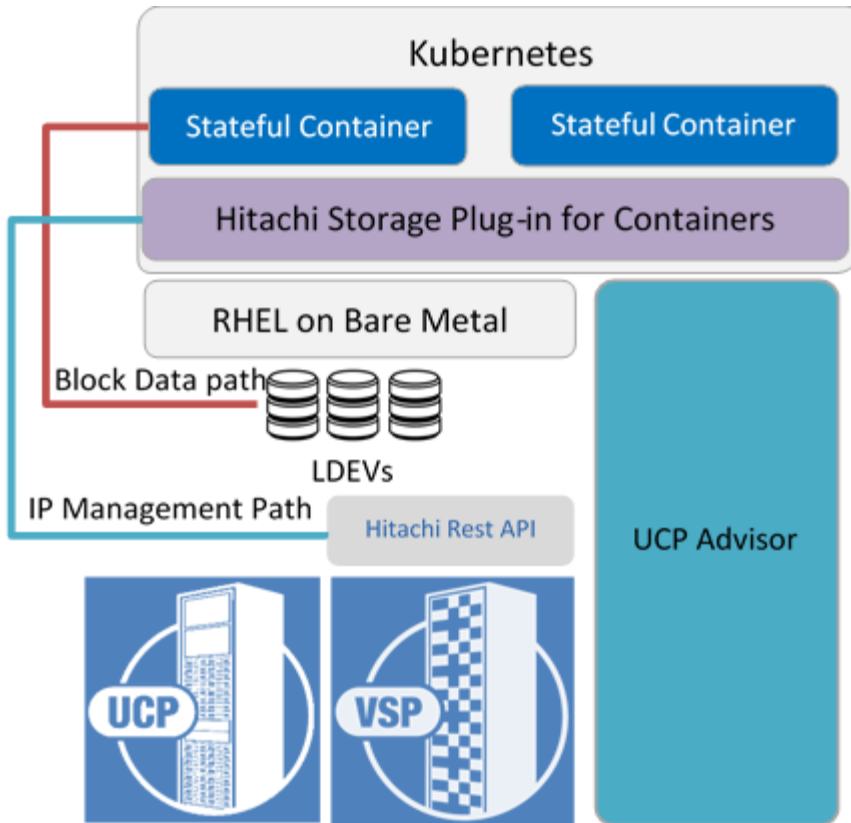
1. Bare metal Red Hat Enterprise Linux (RHEL) container hosts where the Kubernetes master controller nodes and the worker nodes are running on bare metal hosts.
  - Persistent volumes for stateful containers are dynamically provisioned from Hitachi storage.
2. Virtualized RHEL container hosts on VMware vSphere where the Kubernetes master controller nodes and the worker nodes are running on vSphere ESXi hosts. The following persistent volume options are available for this configuration:
  - Use VMware Virtual Volumes (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 UCP HC nodes
3. Mixed hybrid container hosts where Kubernetes master controller nodes (and some worker nodes) are VM + some bare metal worker nodes.

### **Bare Metal Container Hosts**

In an all bare metal deployment, Kubernetes master controller nodes and the worker nodes are deployed on 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 number of objects to management and also ease troubleshooting processes. These benefits cater to environments where service level agreements (SLAs) are more stringent, for example production environments.

Figure 2 shows an overview of a bare metal Kubernetes architecture.

**Figure 2**



The storage platforms can be any Hitachi VSP family storage system. Hitachi VSP provides a REST API for Hitachi Storage Plug-in for Containers (HSPC) to provision persistent volumes. HSPC is deployed within the Kubernetes cluster in the respective master controller nodes and worker nodes. Containers can access the persistent volumes through a local point inside the worker node. The persistent volumes are provided by Hitachi VSP LDEVs/LUNs through block protocol to the worker nodes.

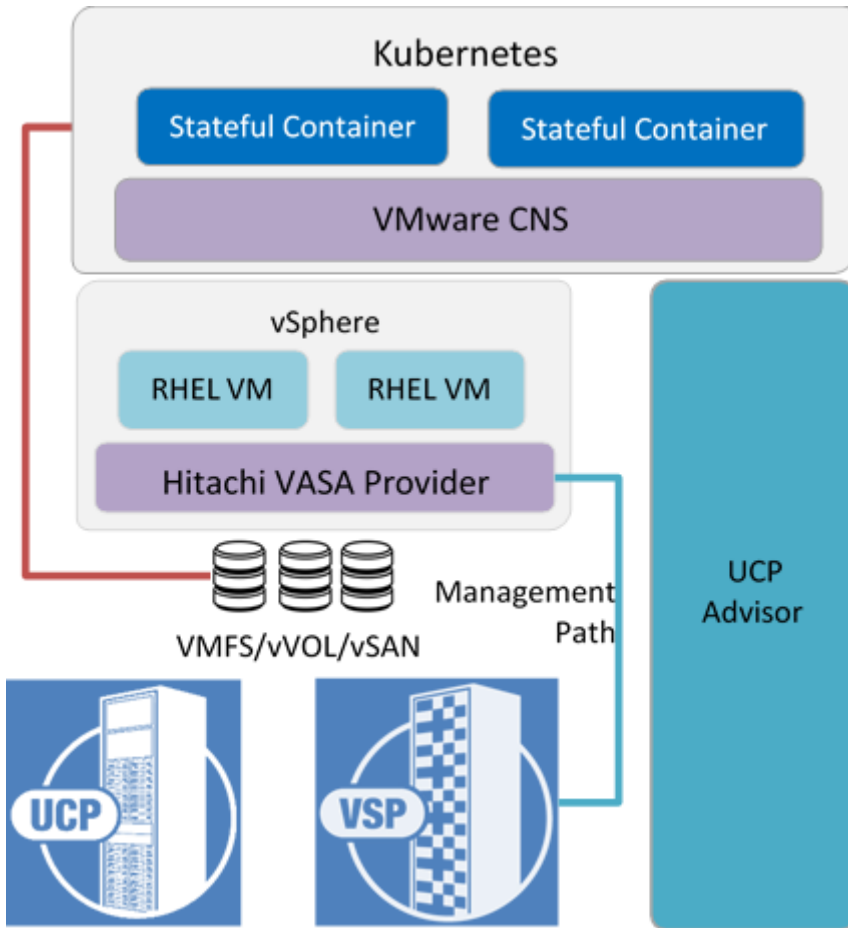
### Virtualized Container Hosts

In an all virtualized deployment, Kubernetes master controller nodes and worker nodes are all deployed on RHEL VM hosted on VMware vSphere hosts. A virtualized Kubernetes deployment is highly flexible and dynamic. Due to vSphere's ability to deploy VMs quickly, new virtual worker nodes can be added to the Kubernetes cluster quickly. No physical deployment in the datacenter is required to have a functional Kubernetes node. A new set of RHEL VMs can also be deployed quickly to deploy another Kubernetes cluster if there's a need to separate Kubernetes tenants into their own cluster. VMware vSphere can also help consolidate the Kubernetes nodes, reducing the rack space footprint. VMware's CPU scheduling ability and memory overcommit can further consolidate nodes. VMware's HA/DRS can also help protect the Kubernetes nodes and enforce the SLA.



Figure 3 shows an overview of a virtualized Kubernetes architecture.

**Figure 3**



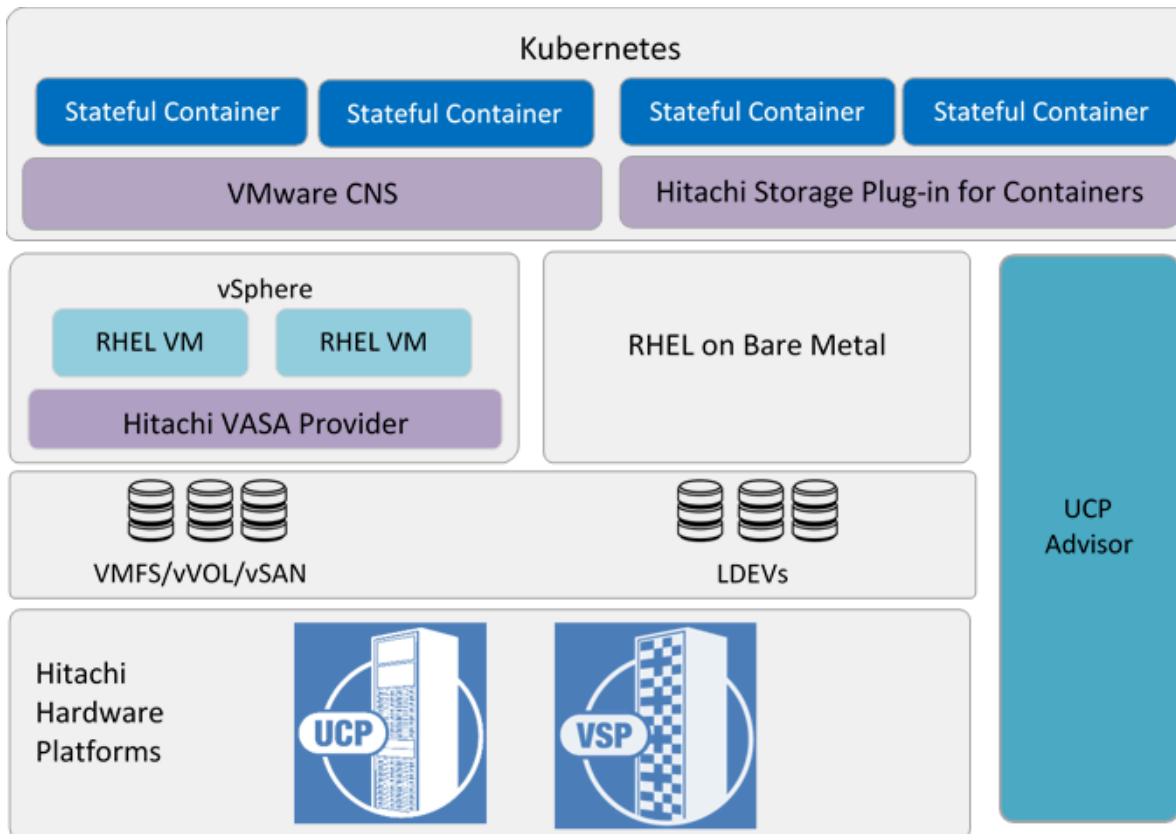
The storage platforms can be any Hitachi VSP family storage system, as well as Hitachi UCP HC vSAN-ready nodes. Hitachi VASA Provider for VMware vCenter provides Hitachi VSP storage capabilities awareness to VMware vCenter where it can be used with VMware Storage Policy Based Management (SPBM). VMware Cloud Native Storage (CNS) provides the persistent storage provisioning capabilities utilizing the VMware storage stack. Containers can access the persistent volumes through a local point inside the worker node VMs. The persistent volumes are provided by VMFS LDEVs or with vVols from Hitachi VSP storage. Persistent volumes can also be provided through Hitachi UCP HC vSAN-ready nodes.

## Mixed Hybrid Container Hosts

In a mixed hybrid deployment, both bare metal hosts and virtualized hosts are used. This combines the benefits of bare metal and virtualized hosts. RHEL deployed on bare metal systems that host Kubernetes nodes provides a simple but reliable and consistent access to compute resources. Stringent SLAs are easier to achieve. RHEL deployed on VM hosting Kubernetes nodes provides a highly dynamic and flexible Kubernetes deployment. RHEL VMs can be dynamically deployed to fit various needs. Combining bare metal hosts and virtualized hosts provides the benefits of both deployment models. As an example, 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, whereas a virtualized worker node is a better choice for container workloads and is considered more scale-out. Another example is a production-developer environment where production workloads are more static, but the development environment has more spin-up and spin-down scenarios. In both examples, Kubernetes master controller nodes are running on RHEL VMs. Kubernetes master controller nodes running on RHEL VMs provides the ability to create a new Kubernetes cluster more efficiently if there's a need to separate tenants at a cluster level.

Figure 4 shows an overview of a mixed hybrid Kubernetes architecture.

Figure 4



The storage platforms can be any Hitachi VSP family storage system, as well as Hitachi UCP HC vSAN-ready nodes. Hitachi VASA Provider for VMware vCenter with VMware CNS provides persistent storage provisioning capabilities for virtualized Kubernetes worker nodes, resulting in persistent storage provided by traditional VMFS volumes or vVols. VMware CNS also provides persistent storage for virtualized Kubernetes worker nodes running on UCP HC vSAN-ready nodes. Hitachi Storage Plug-in for Containers (HSPC) provides persistent storage for both bare metal and virtualized Kubernetes worker nodes.

---

**Note** — For all deployment options, when using HSPC with virtualized Kubernetes nodes, iSCSI must be used. Additionally, with a non-Hitachi UCP infrastructure, the data path can be either iSCSI or Fibre Channel. With the Hitachi UCP CI stack, only Fibre Channel is supported.

---

The computing platform used in this reference architecture is provided by Hitachi UCP, 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 UCP Advisor provides a single pane of glass management for both bare metal and vSphere infrastructures. It provides automation and management of the end-to-end hardware infrastructure including day-0 operations.

## Solution Components

This section lists the Unified Compute Platform (UCP) 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 2.10GHz v5 processor</li> <li>■ 8 × 16 GB DIMM, 128 GB memory</li> <li>■ 32 Gb SATADOM (boot)</li> <li>■ Emulex LPe3200 32 Gb/sec dual port PCIe HBA</li> <li>■ Intel X527-DA4 10G quad-port NIC</li> <li>■ vSAN Cache Tier: 2 × Intel Optane SSD DC P4800X (375GB, U.2) NVMe</li> <li>■ vSAN Capacity Tier: 6 × Intel SSD DC P4510 (4TB, U.2) NVMe</li> </ul>	BMC: 4.62.06  BIOS: S5BH3B14.H01	8
VSP 5500	<ul style="list-style-type: none"> <li>■ 2 TB cache</li> <li>■ 8 × 1.9 TB NVMe drives used</li> <li>■ 4 × 32 Gbps Fibre Channel ports used</li> </ul>	90-01-61-00/00	1

TABLE 1. HARDWARE COMPONENTS (CONTINUED)

Hardware	Description	Version	Quantity
VSP G1500	<ul style="list-style-type: none"> <li>■ 1 TB cache</li> <li>■ 8 × 1.9 TB SSD used</li> <li>■ 4 × 16 Gbps Fibre Channel ports used</li> </ul>	80-06-69-00/00	1
Cisco Nexus 93180LC-EX	<ul style="list-style-type: none"> <li>■ 18-port 100 GbE</li> <li>■ 24-port 40 GbE</li> </ul>	7.0(3)I7(6)	2
Cisco Nexus 93180YC-EX	<ul style="list-style-type: none"> <li>■ 48-port 25 GbE</li> <li>■ 6-port 100 GbE</li> </ul>	7.0(3)I7(6)	2
Cisco Nexus 3048TP	<ul style="list-style-type: none"> <li>■ 48-port 1 GbE</li> <li>■ 4-port 10 GbE</li> </ul>	7.0(3)I7(6)	1
Brocade G620	<ul style="list-style-type: none"> <li>■ 48-port 16/32 Gbps Fibre Channel switch</li> </ul>	8.2.1c	2

UCP CI hardware options are described in “Appendix A - Product Descriptions” on page 37.

## Software Components

Table 2 lists the key software components.

TABLE 2. SOFTWARE COMPONENTS

Software	Version
Hitachi Storage Virtualization Operating System RF	90-01-61-00/00 80-06-69-00/00
Hitachi Unified Compute Platform Advisor	3.5
Hitachi Storage Provider for VMware vCenter (VASA)	3.5.6
Hitachi Storage Plug-in for Containers	2.5.0
VMware vSphere	6.7u3
Red Hat Enterprise Linux	7.7
Docker Engine	1.13.1
Kubernetes	1.16.1

UCP CI software options are described in “Appendix A - Product Descriptions” on page 37.

## Solution Design

This is the detailed solution example of a UCP CI configured in a spine-leaf network topology.

Figure 5 shows a spine-leaf network topology for UCP CI with Cisco switches.

Figure 5

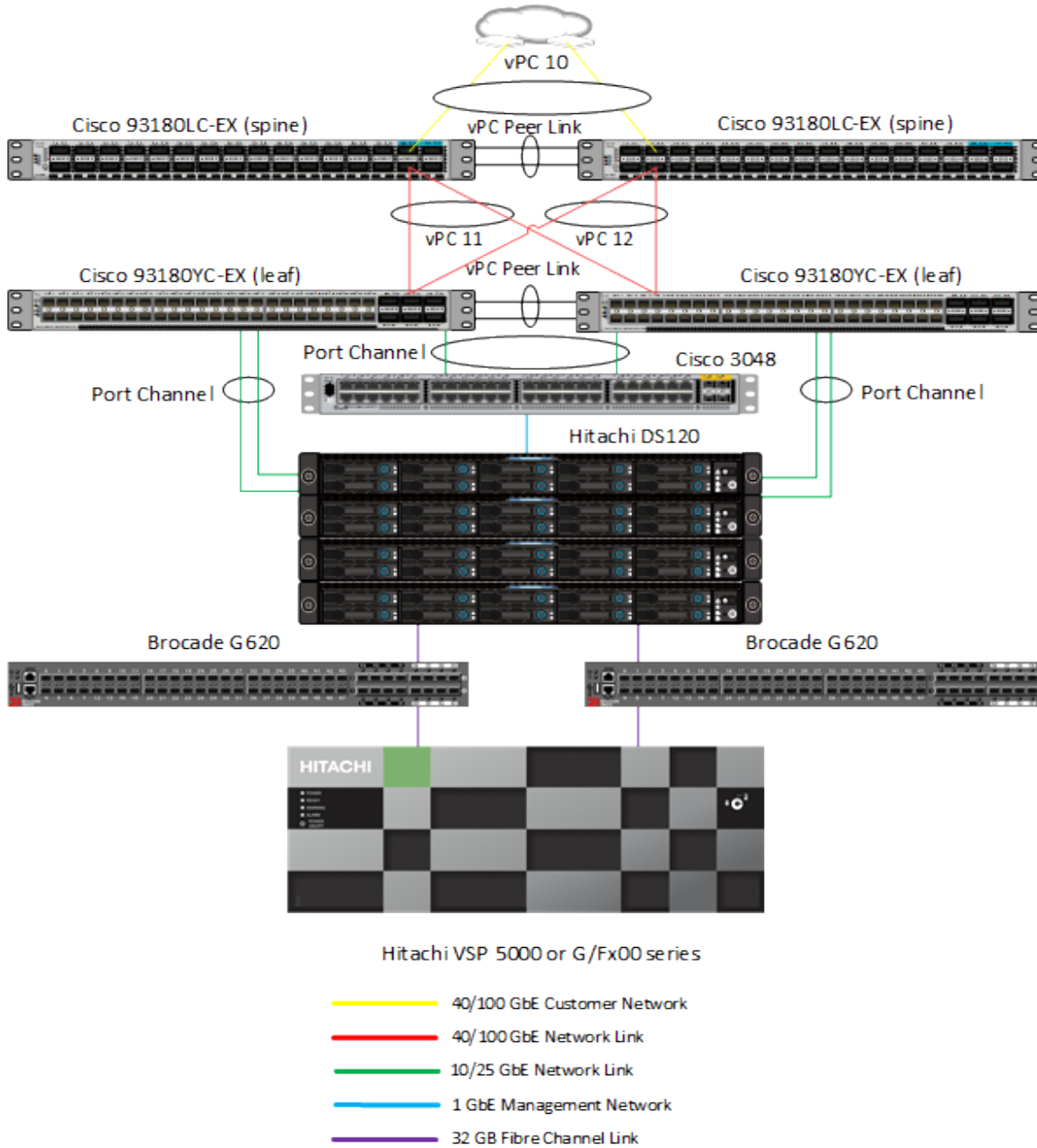
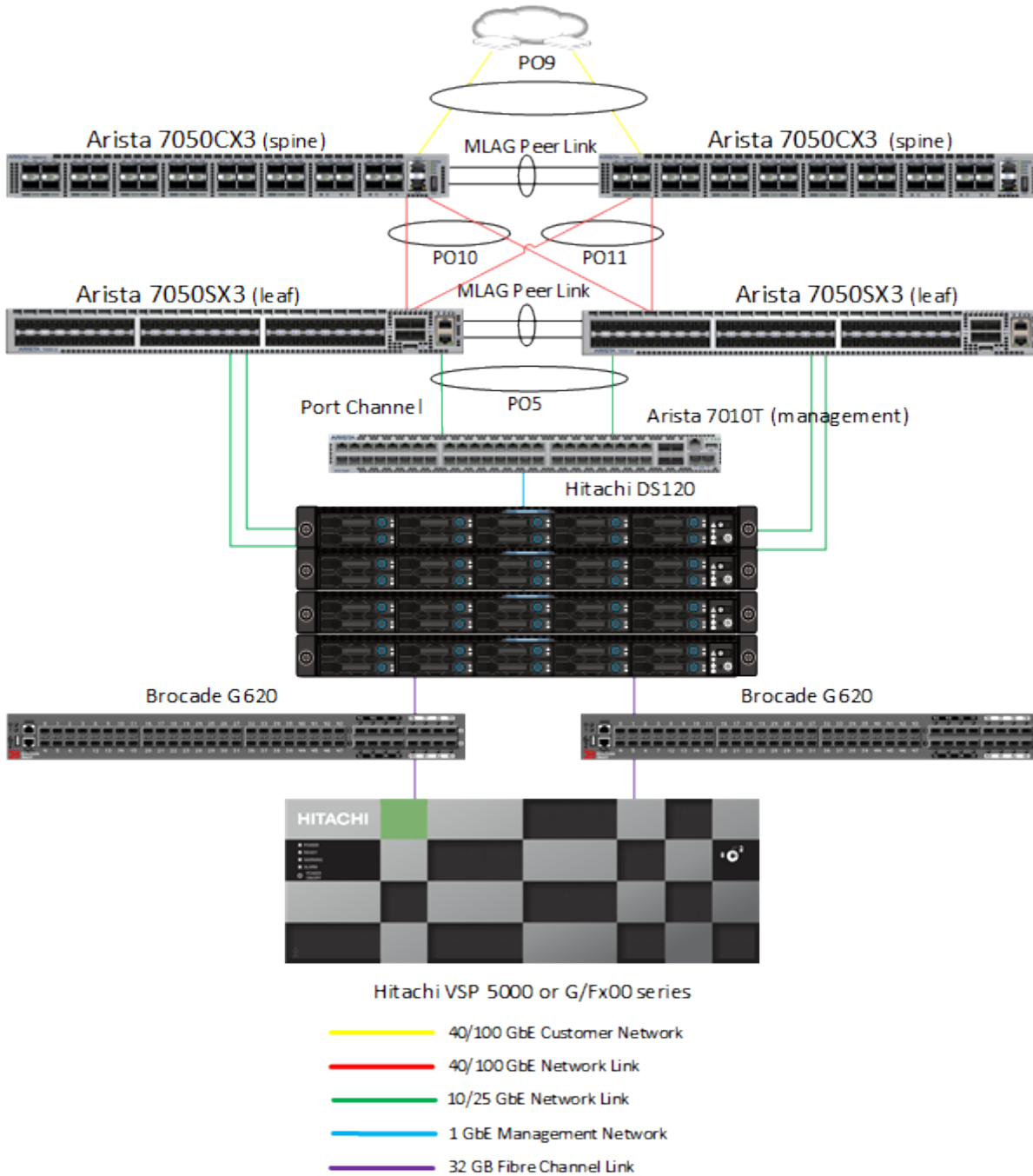


Figure 6 shows a spine-leaf network topology for UCP CI with Arista switches.

**Figure 6**

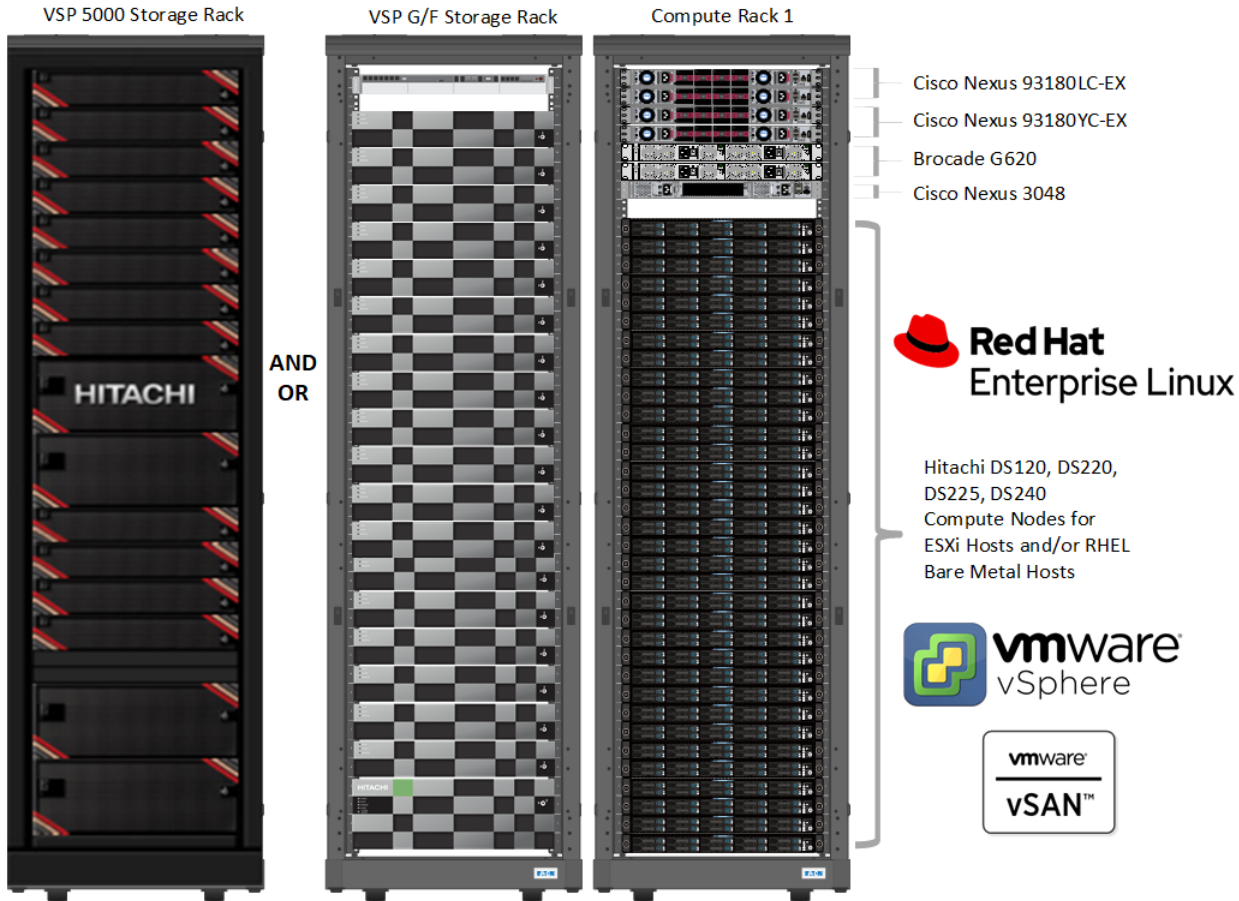


For more information regarding UCP CI configurations, see the following link:

[https://knowledge.hitachivantara.com/Documents/Application\\_Optimized\\_Solutions/VMWare/Unified\\_Compute\\_Platform\\_CI\\_for\\_VMware\\_vSphere\\_Reference\\_Architecture\\_Guide](https://knowledge.hitachivantara.com/Documents/Application_Optimized_Solutions/VMWare/Unified_Compute_Platform_CI_for_VMware_vSphere_Reference_Architecture_Guide)

Figure 7 shows an example rack of the UCP CI for Kubernetes cluster deployment. On the compute rack, DSxx0 compute nodes can be used as RHEL bare metal container hosts, VMware ESXi hosts, or a combination of both.

**Figure 7**



## Deploy a Kubernetes Cluster on Bare Metal Hosts

Figure 8 shows a high-level diagram of a bare metal Kubernetes cluster on UCP CI.

- Red Hat Enterprise Linux 7 (RHEL 7) was installed on seven DS120 compute nodes.
- Kubernetes cluster was deployed on RHEL servers.
- Hitachi Storage Plug-in for Containers (HSPC) was installed on Kubernetes as the CSI driver.
- On VSP 5500, a storage pool name GOLD Plus was created from NVMe drives.
- On Kubernetes, StorageClass representing GOLD Plus storage pool was defined
- When a stateful container was created, HSPC created a persistent volume dynamically from the GOLD Plus storage pool on VSP 5500.

Figure 8

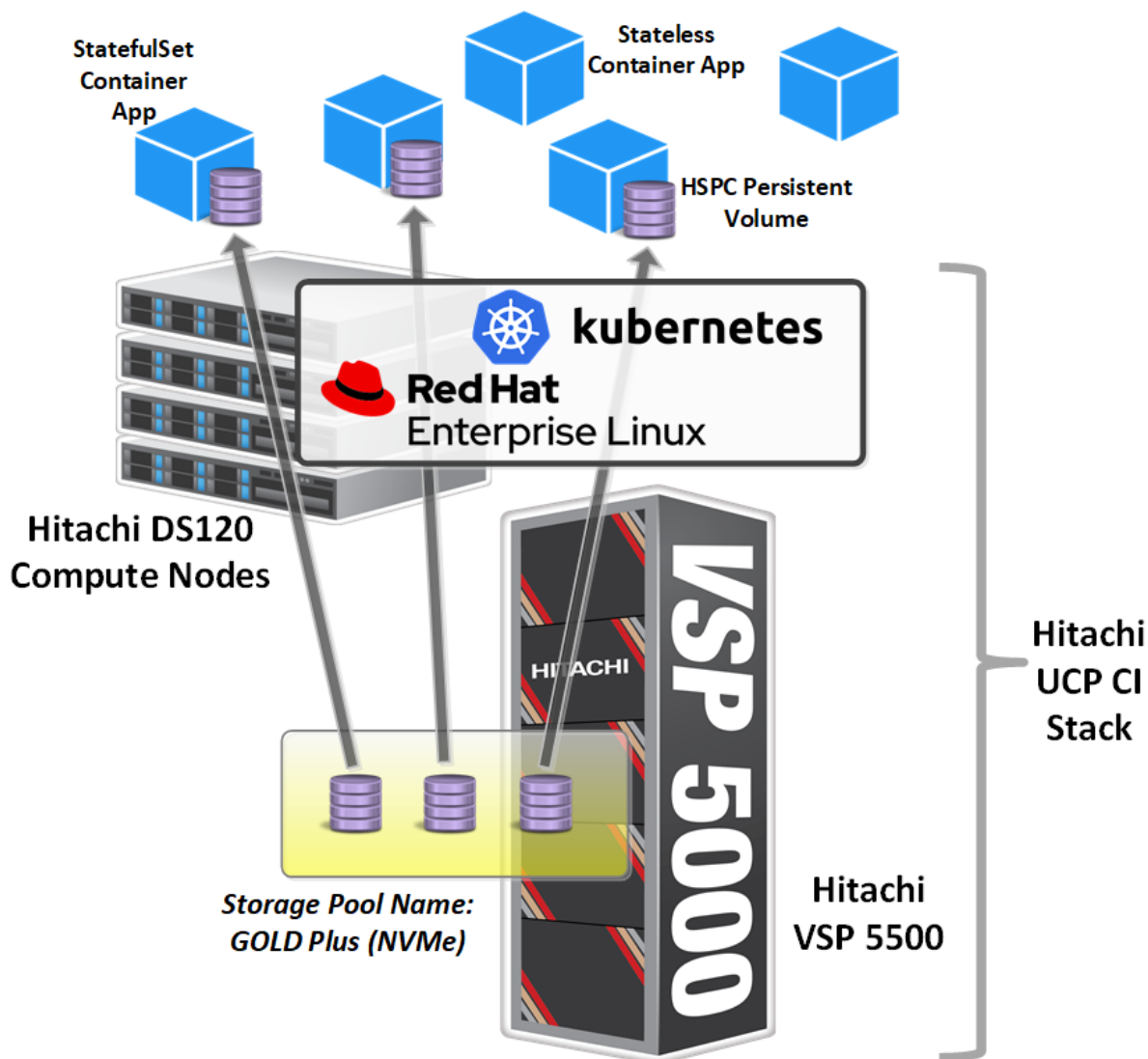




Table 3 lists bare metal RHEL Kubernetes hosts placement in UCP CI.

TABLE 3. KUBERNETES NODES AND IP ADDRESSES EXAMPLE FOR BARE METAL HOSTS

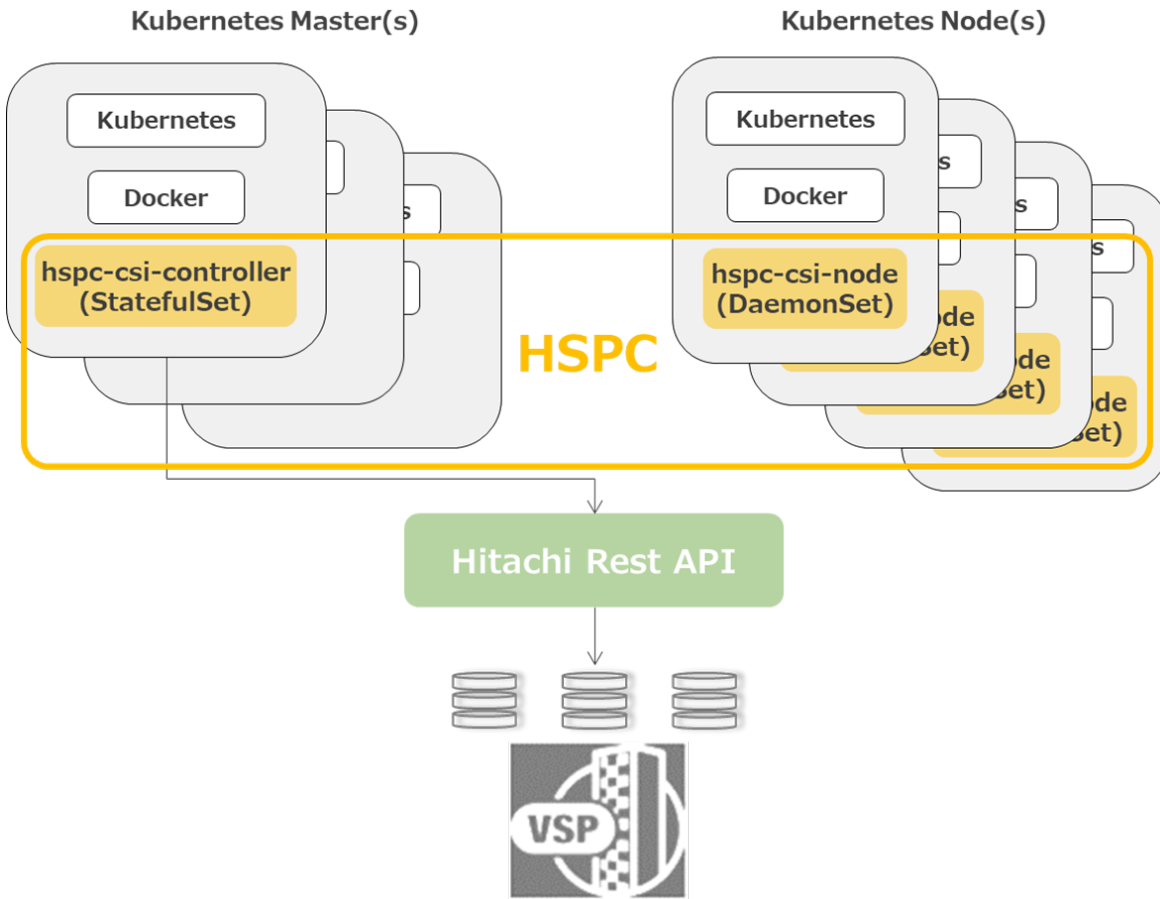
Node Name	Node Role	Device Type and Location	IP Address
Load Balancer Virtual IP	Load Balancer	Existing infrastructure outside of UCP CI	10.76.46.100
Master1.k8s.local	Master Node 1	DS120 Rack1 U1	10.76.46.101
Master2.k8s.local	Master Node 2	DS120 Rack1 U2	10.76.46.102
Master3.k8s.local	Master Node 3	DS120 Rack1 U3	10.76.46.103
Node1.k8s.local	Worker Node 1	DS120 Rack1 U4	10.76.46.104
Node2.k8s.local	Worker Node 2	DS120 Rack1 U5	10.76.46.105
Node3.k8s.local	Worker Node 3	DS120 Rack1 U6	10.76.46.106
Node4.k8s.local	Worker Node 4	DS120 Rack1 U7	10.76.46.107

[Hitachi Unified Compute Platform Advisor](#) (UCP Advisor) was used for VSP 5500 day-0 operations such as storage pool creation from NVMe drives.

Hitachi Storage Plug-in for Containers (HSPC) is a Kubernetes CSI driver that can be used to dynamically provision persistent volumes for stateful containers from Hitachi storage. The driver is deployed as a container for Kubernetes and works just like a regular application as shown in Figure 8. HSPC receives instructions for volume creation from Kubernetes and issues instructions to the Hitachi storage REST API. HSPC provides provisioned storage volumes to Kubernetes-managed containers and allows those volumes to be managed from Kubernetes.

HSPC makes it possible to fully utilize VSP storage system's high-performance and high-reliability features in a container environment.

**Figure 9**



For detailed HSPC installation procedures, see the following link:

[https://knowledge.hitachivantara.com/Documents/Adapters\\_and\\_Drivers/Storage\\_Adapters\\_and\\_Drivers/Containers](https://knowledge.hitachivantara.com/Documents/Adapters_and_Drivers/Storage_Adapters_and_Drivers/Containers)

The following article also provides details regarding HSPC Kubernetes CSI support:

<https://community.hitachivantara.com/s/article/container-storage-interface-csi-driver-for-hitachi-virtual-storage-platform-series/>

The following lists the maximum numbers for stateful containers:

- The maximum number of volumes on Hitachi VSP 5000 Series is 65,280.
  - For details, see [VSP 5000 series specifications](#)
- The maximum number of nodes is 5000.
- The maximum number of pods per node is 100, and total number of pods is 150,000.
  - For details, see [Kubernetes building large clusters](#)

## Deploy a Kubernetes Cluster on VMware vSphere

A virtualized Kubernetes cluster can easily be deployed on UCP CI for VMware. There are three deployment options for stateful container applications with UCP CI for VMware as shown in Figure 10.

1. Deploy stateful containers on VMware vVols. Hitachi Storage Provider for VMware vCenter is used to provision persistent volumes directly from VSP storage.
2. Deploy stateful containers on VMFS datastores. The persistent volumes are created as VMDK files.
3. Deploy stateful containers on VMware vSAN datastores hosted by UCP HC nodes.

Figure 10

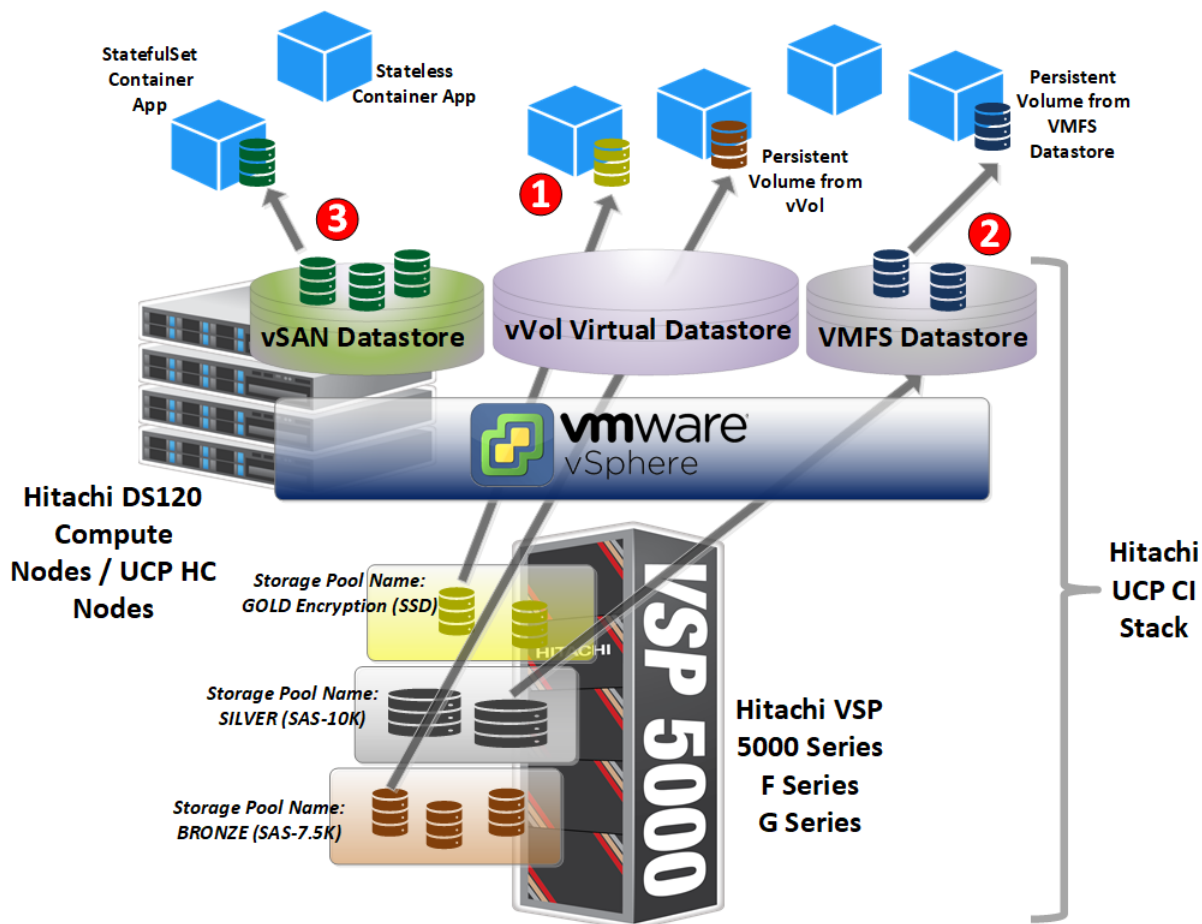


Figure 10 also shows the following:

- Three storage pools consist of different type of drives on VSP 5500H as listed in Table 4.
- One persistent volume is dynamically created form the GOLD Encryption storage pool as a VMware vVol and attached to a stateful container.
- The other persistent volume is dynamically created form the BRONZE storage pool as a VMware vVol and attached to a stateful container.
- A VMFS datastore is created by a large volume/LUN from the SILVER storage pool.

TABLE 4. VSP 5500H STORAGE POOLS

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

The different types of storage pools meet the following demands:

- VSP 5500H and VSP Gx00 storage provide different storage pools with different levels of capabilities as shown in Table 4.
- Container applications require different storage requirements.
- The container administrators create volumes that fit the container application requirements.

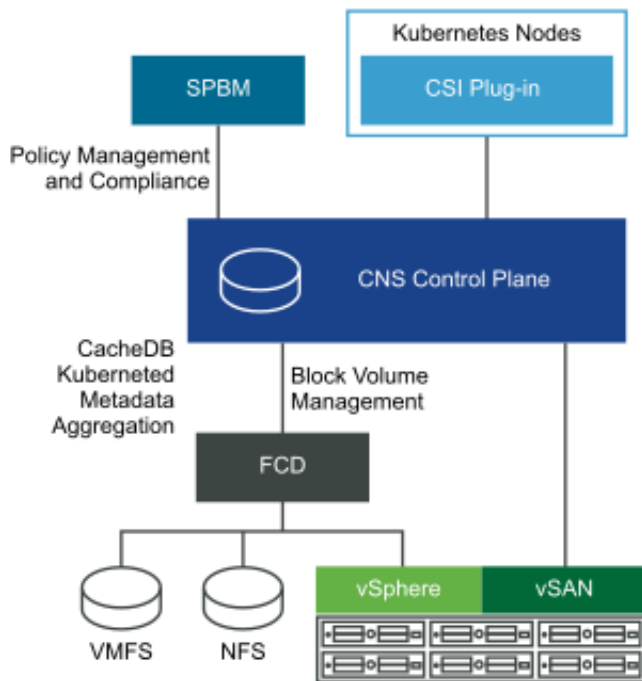
Table 5 lists Kubernetes cluster VM allocations on UCP CI for VMware.

TABLE 5. KUBERNETES NODES AND IP ADDRESSES EXAMPLE FOR VM HOSTS

Node Name	Node Role	Device Type and Location	IP Address
Load Balancer Virtual IP	Load Balancer	Existing infrastructure outside of UCP CI	10.76.46.100
Master1.k8s.local	Master Node 1	vSphere cluster hosted in DS120 Rack1 U1 – U4	10.76.46.101
Master2.k8s.local	Master Node 2		10.76.46.102
Master3.k8s.local	Master Node 3		10.76.46.103
Node1.k8s.local	Worker Node 1		10.76.46.104
Node2.k8s.local	Worker Node 2		10.76.46.105
Node3.k8s.local	Worker Node 3		10.76.46.106
Node4.k8s.local	Worker Node 4		10.76.46.107

VMware provides Cloud Native Storage (CNS) in order to create 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 vSphere Container Storage Interface (CSI) Plug-in which interfaces with Kubernetes clusters. [Storage Policy Based Management \(SPBM\)](#) is also used for container volume management. With CSI Plug-in and SBPM storage policies, the persistent volumes can be easily created from any datastore types that are supported by vSphere.

See the following link to set up a virtualized Kubernetes cluster with VMware Cloud Native Storage:

<https://docs.vmware.com/en/VMware-vSphere/6.7/Cloud-Native-Storage/GUID-51D308C7-ECFE-4C04-AD56-64B6E00A6548.html>

### *Use VMware vVol to Provision Persistent Volumes Directly from VSP Storage*

VMware vVol provides more granular volume management with SPBM. For each vVol, Hitachi storage creates a dedicated volume (LDEV) dynamically. The following link provides a detailed explanation of VMware vVol:

<https://kb.vmware.com/s/article/2113013>

In order to use VMware vVol with Hitachi storage, Hitachi Storage Provider for VMware vCenter needs to be installed. The deployment guide can be found at the following link:

[https://knowledge.hitachivantara.com/Documents/Adapters\\_and\\_Drivers/Storage\\_Adapters\\_and\\_Drivers/VMware/Storage\\_Provider\\_for\\_VMware\\_vCenter\\_\(VASA\)](https://knowledge.hitachivantara.com/Documents/Adapters_and_Drivers/Storage_Adapters_and_Drivers/VMware/Storage_Provider_for_VMware_vCenter_(VASA))

Figure 12 lists detailed capabilities that are included in the Hitachi VASA provider.

**Figure 12**

Capability Schema	
<b>Capability Schema</b>	
<b>Name</b>	Capability Schema for All
<b>Namespace</b>	com.hitachi.storageprovider.vvol
<b>Capability</b>	
<b>Available Values</b>	
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
<b>Additional tag based capabilities</b>	
<b>User Defined Capabilities</b>	
<b>Available Values</b>	
Location	Dublin,Frankfurt,Munich

Define the vVol storage policies that meet your storage requirements for the Kubernetes environment. Reference the VM storage policy name in the corresponding **StorageClass** YAML file to create a **StorageClass** for vVols. An example can be found in the Solution Validation section.

### Create Dynamic Persistent Volumes from Regular VMFS Datastores

In VMware CSI, there are two ways to specify regular VMFS datastores for Kubernetes to use for volume creation:

1. Use VMware SPBM to specify one or more datastores in the same category
  - If more than two datastores are in the same storage policy, the datastore with the most free space available is used.
2. Use the datastore's URL to specify a single datastore

For any storage management, VMware recommends using SPBM. To use SPBM for VMFS datastore, a tag-based VM storage policy needs to be created. The following link provides details regarding tag-based SPBM:

<https://blogs.vmware.com/virtualblocks/2018/07/26/using-tag-based-spbm-policies-to-manage-your-storage/>

Hitachi VASA provider also automatically tags datastores with a set of capabilities for LUNs that are provisioned from Hitachi storage.

Define the VM storage policies for VMFS datastore that meet your storage requirements for the Kubernetes environment. Reference the VM storage policy name in the corresponding **StorageClass** YAML file to create a **StorageClass**. An example can be found in the Solution Validation section.

## Create Persistent Volumes from VMware vSAN Datastores Hosted by UCP HC Nodes

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

The 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 the following link for default vSAN storage policy details:

<https://docs.vmware.com/en/VMware-vSphere/6.7/com.vmware.vsphere.virtualsan.doc/GUID-C228168F-6807-4C2A-9D74-E584CAF49A2A.html>

Define the VM storage policies for vSAN datastore that meets your storage requirements for the Kubernetes environment. Reference the VM 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 VM, vSphere deploys the VM into the ESXi host, that uses the least amount of resources by default. In the shared VMware environment, this creates a possibility that all the Kubernetes master VMs reside on the same ESXi host. If this ESXi host goes down, all the master nodes go down as well, and Kubernetes cluster operation is impacted. To avoid this situation, use the VM-VM affinity rules for vSphere environments.

Figure 13 shows an example of anti-affinity rules for the Kubernetes cluster VMs. With four ESXi hosts in the vSphere DRS cluster, 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 vCenter interface for configuring VM/Host Rules. The 'Cluster' page is open, and the 'Configure' tab is selected. The left sidebar shows the navigation menu with 'VM/Host Rules' highlighted. The main content area displays the 'VM/Host Rules' configuration page. At the top, there are buttons for '+ Add...', 'Edit...', and 'Delete'. Below this is a table with the following data:

Name	Type	Enabled
K8s-Workers	Separate Virtual Machines	Yes
K8s-Masters	Separate Virtual Machines	Yes

Below the table is the 'VM/Host Rule Details' section, which states: 'The listed 3 Virtual Machines must run on different hosts.' Below this is another table with the following data:

Rule Members	Conflicts
rhel7-node1-133	0
rhel7-node0-132	0
rhel7-node2-134	0

For more information regarding VM-VM affinity rules, see the following link:

<https://docs.vmware.com/en/VMware-vSphere/6.7/com.vmware.vsphere.resmgmt.doc/GUID-94FCC204-115A-4918-9533-BFC588338ECB.html?hWord=N4lghgNiBc4GZwJYDtEBcCelC+Q>

## Deploy a Kubernetes Cluster on Mixed Hybrid Environments (VM and Bare Metal Hosts)

All the previous deployment configurations can be combined and mixed together. This is also called a hybrid environment where Kubernetes master nodes and some worker nodes are virtualized, and some bare-metal worker nodes are also joined to the cluster. Table 6 provides an example of this configuration.

TABLE 6. KUBERNETES NODES AND IP ADDRESSES EXAMPLE FOR VM AND BARE METAL HOSTS

Node Name	Node Role	Device Type and Location	IP Address
Load Balancer Virtual IP	Load Balancer	Existing infrastructure outside of UCP CI	10.76.46.100
vMaster1.k8s.local	Master Node 1 (VM)	vSphere cluster hosted in DS120 Rack1 U1 – U4	10.76.46.101
vMaster2.k8s.local	Master Node 2 (VM)		10.76.46.102
vMaster3.k8s.local	Master Node 3 (VM)		10.76.46.103
vNode1.k8s.local	Worker Node 1 (VM)		10.76.46.104
vNode2.k8s.local	Worker Node 2 (VM)		10.76.46.105
vNode3.k8s.local	Worker Node 3 (VM)		10.76.46.106
vNode4.k8s.local	Worker Node 4 (VM)		10.76.46.107
Node5.k8s.local	Worker Node 5	DS120 Rack1 U5	10.76.46.108
Node6.k8s.local	Worker Node 6	DS120 Rack1 U6	10.76.46.109
Node7.k8s.local	Worker Node 7	DS120 Rack1 U7	10.76.46.110
Node8.k8s.local	Worker Node 8	DS120 Rack1 U8	10.76.46.111

Note that once all the nodes are joined to the cluster, Kubernetes does not know if the worker nodes are bare metal hosts or VMs. 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 and HSPC CSI plug-in installations as well.

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

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

In this example, the following were used:

- VM worker nodes:
  - # kubectl label nodes vNode1 node-type=vm-cns
- Bare metal worker nodes:
  - # kubectl label nodes Node5 node-type=hspc-fc



When installing VMware CNS CSI and HSPC drivers, **nodeSelector** needs to be added to the YAML file. The following example shows the part of the HSPC YAML file for installation.

```
apiVersion: apps/v1
kind: DaemonSet
metadata:
  name: hspc-csi-node
  namespace: kube-system
spec:
  selector:
    matchLabels:
      app: hspc-csi-driver
  updateStrategy:
    type: RollingUpdate
  template:
    metadata:
      labels:
        app: hspc-csi-driver
    spec:
      serviceAccount: hspc-csi-sa
      # Only run on the bare-metal nodes with HBAs
      nodeSelector:
        node-type: hspc-fc
    containers:
      - name: driver-registrar
```

The following link provides detailed information regarding **nodeSelector**:

<https://kubernetes.io/docs/concepts/configuration/assign-pod-node/>

## Solution Validation

The Kubernetes cluster was deployed on UCP CI successfully with bare metal, virtualized, or hybrid configurations. The Hitachi Storage Plug-in for Containers (HSPC) and VMware's Cloud Native Storage drivers were also installed for persistent volume provisioning. The container volume operations described in this section were performed to validate this solution.

### Create Persistent Volumes for Bare Metal Hosts

The Hitachi Storage Plug-in for Containers (HSPC) was installed for bare metal container hosts to provision the persistent volumes directly from Hitachi storage. Scaling StatefulSet containers were validated in this section. The following lists the test procedure:

- Deploy a Kubernetes cluster with four DS120 bare metal worker nodes
- On VSP 5500, create a storage pool with NVMe drives
- Install the HSPC CSI driver on the Kubernetes cluster
- Deploy a MongoDB StatefulSet with a 512 GB persistent volume
- Scale MongoDB StatefulSet up to 8 and measure the container creation time
- Validate the volume creation on VSP 5500
- When scaling containers, Kubernetes creates containers sequentially. When the container status is **Running**, the next container creation starts. Table 7 lists the container creation time for MongoDB with a 512 GB volume.

TABLE 7. 512 GB MONGODB CONTAINER CREATION TIME

Replicas	Total Completion Time	Average Container Creation Time
1	19 sec.	19 sec.
4	91 sec.	22.75 sec
8	171 sec	21.35 sec

The average MongoDB container creation in this environment was about 20 seconds. From the HSPC logs, the storage operation took about 10 seconds to provision a volume, and the Kubernetes took another 10 seconds to get MongoDB up and running.

To map a persistent volume to the corresponding volume in the storage, use the following command:

```
kubectl describe pv <pv-name>
```

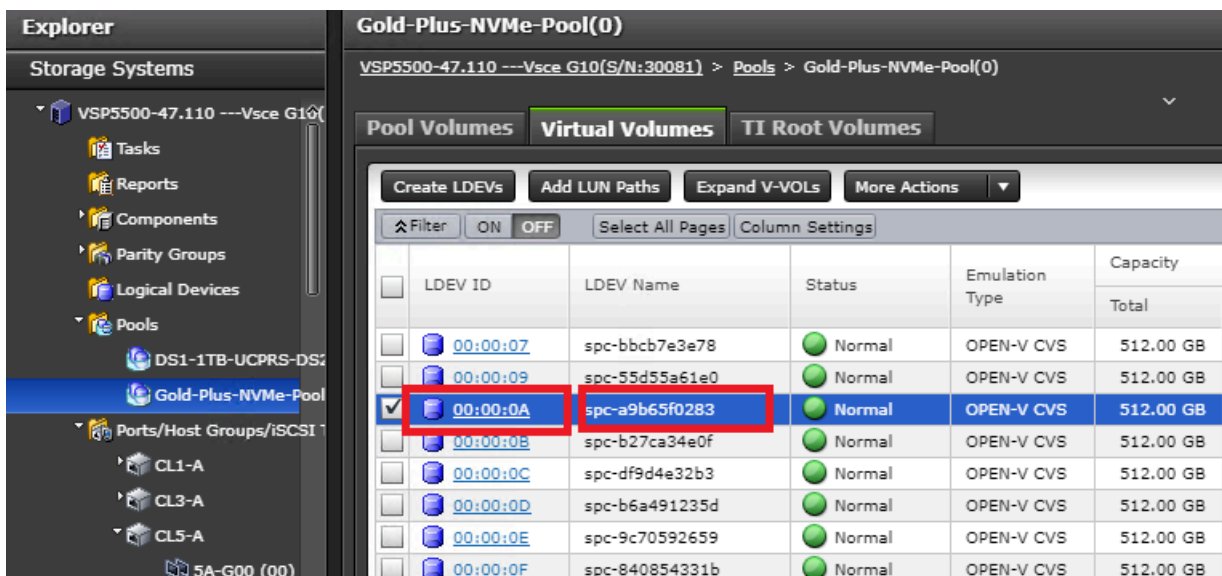
Figure 14 shows an output example with `ldevIDHex` and `nickname` attributes.

**Figure 14**

```
[root@master0 hspc]# kubectl describe pv pvc-3f1e9a1a-cdbc-46ff-a005-bbcf0a101f45
Name:          pvc-3f1e9a1a-cdbc-46ff-a005-bbcf0a101f45
Labels:        <none>
Annotations:   pv.kubernetes.io/provisioned-by: hspc.csi.hitachi.com
Finalizers:    [kubernetes.io/pv-protection external-attacher/hspc-csi-hitachi-com]
StorageClass:  sc-vsp5k-gold
Status:        Bound
Claim:         default/mongodb-persistent-storage-claim-hspc-mongod-hspc-2
Reclaim Policy: Delete
Access Modes:  RWO
VolumeMode:   Filesystem
Capacity:     512Gi
Node Affinity: <none>
Message:
Source:
  Type:        CSI (a Container Storage Interface (CSI) volume source)
  Driver:      hspc.csi.hitachi.com
  VolumeHandle: 60060e8008758100005075810000000a--spc-a9b65f0283
  ReadOnly:    false
  VolumeAttributes:
    autoHG=true
    connectionType=fc
    ldevIDDec=10
    ldevIDHex=00:0A
    mode=normal
    nickname=spc-a9b65f0283
    ports=CL5-A, CL5-B, CL3-A, CL4-A, CL1-A, CL2-A
    size=512Gi
```

Go to **Hitachi Device Manager – Storage Navigator** and look for the corresponding volume in the storage as shown in Figure 15.

**Figure 15**



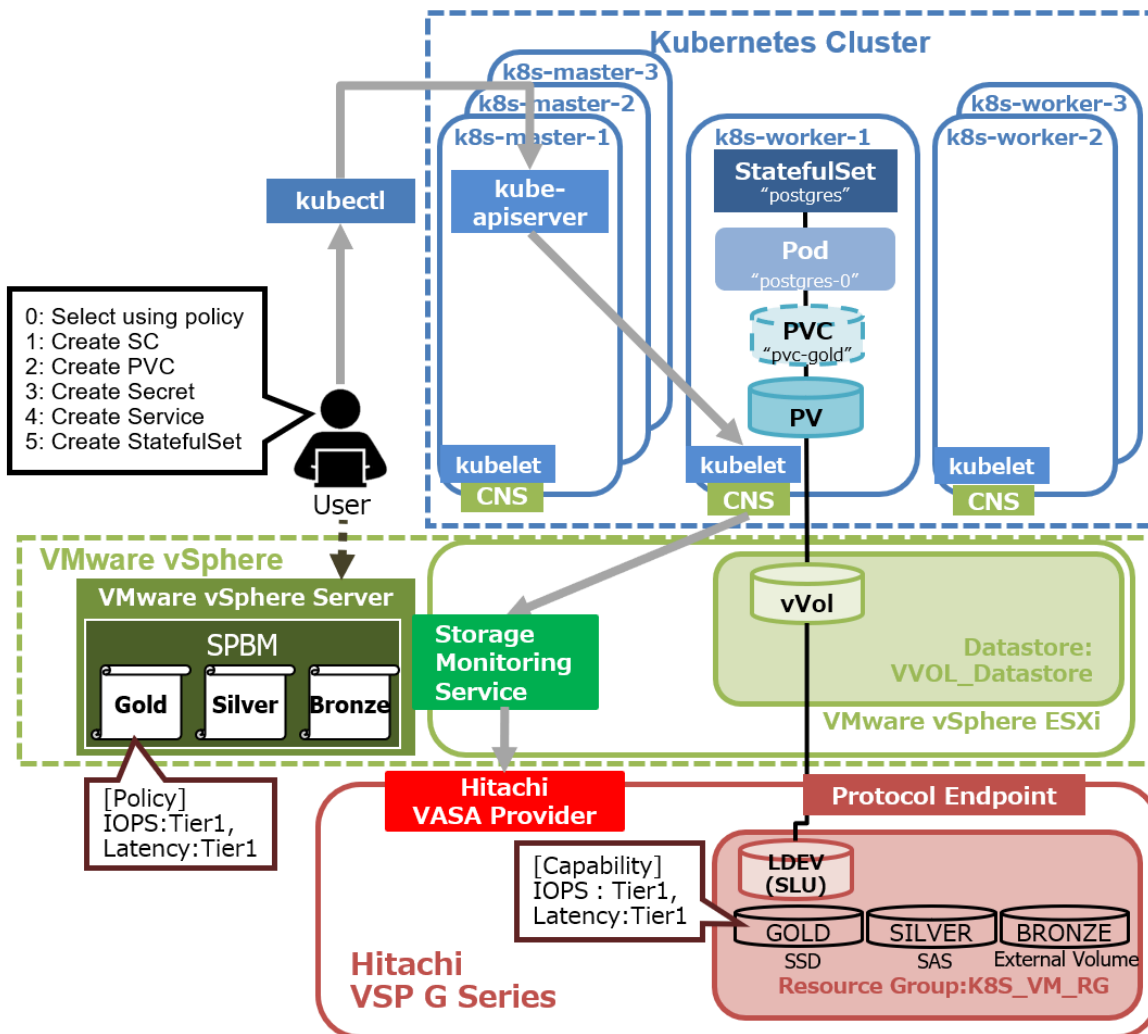
An example of creating StatefulSet applications can be found at the following link:

<https://community.hitachivantara.com/s/article/deploy-wordpress-and-mysql-in-kubernetes-using-hspsc/>

### Create Persistent Volumes from VMware vVol

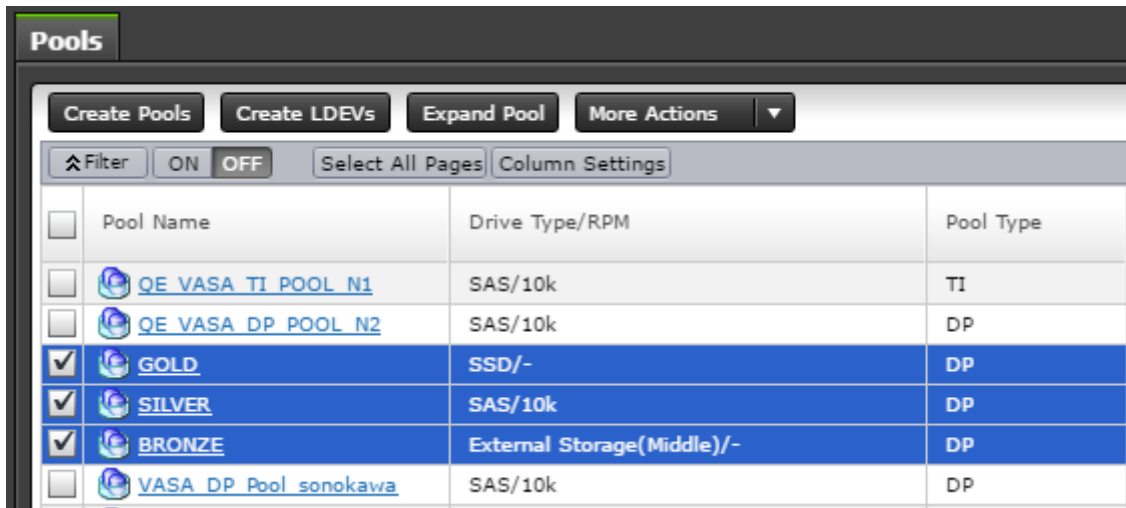
One of the common methods to manage the storage volume in VMware vSphere environment is using VMware vSphere Storage Policy Based Management (SPBM). The storage requirements such as IOPS and I/O latency required by the applications that make up the system vary from application to application. Policy-based storage management allows each application to have a persistent volume that meets their requirements. Figure 16 shows an overview of the test environment to deploy a Postgres StatefulSet container on a VMware vVol.

Figure 16



For the validation, GOLD, SILVER, and BRONZE storage pools were created in VSP G1500 as shown in Figure 17.

Figure 17



The following VM Storage Policies were created in vSphere as listed in Table 8.

TABLE 8. VM STORAGE POLICY MAPPING

VM Storage Policy Name	Performance IOPS-Class	Performance Latency-Class	VSPG1500 Storage Pool Name	Drive Type
Gold vVol Storage Policy	Tier1_IOPS	Tier1_Latency	GOLD	SSD
Silver vVol Storage Policy	Tier2_IOPS	Tier2_Latency	SILVER	SAS 10k
Bronze vVol Storage Policy	Tier3_IOPS	Tier3_Latency	BRONZE	External Storage

After VM Storage Policies are created, go to a Kubernetes Master node, and reference it in the StorageClass YAML file **sc-gold-vvol-vmw.yaml** as shown below.

```
kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
  name: sc-gold-vvol-vmw
  annotations:
    storageclass.kubernetes.io/is-default-class: "false"
provisioner: csi.vsphere.vmware.com
parameters:
  storagepolicyname: "Gold vVol Storage Policy"
```

Use the following command to create storage class:

```
# kubectl create -f sc-gold-vvol-vmw.yaml
```

Use the following **pvc-gold-vvol-vmw.yaml** file and command to create a persistent volume claim. This operation enables VMware CNS and Hitachi VASA provider to create a vVol dynamically in Hitachi VSP storage.

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: pvc-gold-vvol-vmw
  annotations:
    volume.beta.kubernetes.io/storage-class: sc-gold-vvol-vmw
spec:
  accessModes:
  - ReadWriteOnce
  resources:
  requests:
    storage: 10Gi
```

```
# kubectl create -f pvc-gold-vvol-vmw.yaml
```

Create a Secret in the Kubernetes cluster and set the password for PostgreSQL login. Use the following command:

```
# kubectl create secret generic postgres-pass --from-literal=password=postgres-pass
```

Use the following **svc-postgres.yaml** file and command to create a PostgreSQL service.

```
apiVersion: v1
kind: Service
metadata:
  name: svc-postgres
  labels:
    app: postgresql
spec:
  type: ClusterIP
  ports:
  - port: 5432
  selector:
    app: postgresql
```

```
# kubectl create -f svc-postgres.yaml
```

Use the following **postgres.yaml** file and command to create a Postgres StatefulSet.

```
apiVersion: apps/v1
kind: StatefulSet
metadata:
  name: postgres
  labels:
    app: postgresql
spec:
  serviceName: svc-postgres
  replicas: 1
  selector:
    matchLabels:
      app: postgresql
  updateStrategy:
    type: RollingUpdate
  template:
    metadata:
      labels:
        app: postgresql
    spec:
      containers:
        - name: postgres
          image: postgres:11.3
          imagePullPolicy: IfNotPresent
          env:
            - name: POSTGRES_PASSWORD
              valueFrom:
                secretKeyRef:
                  name: postgres-pass
                  key: password
            - name: PGDATA
              value: /var/lib/postgresql/data/pgdata
      ports:
        - containerPort: 5432
          name: postgresql
      volumeMounts:
        - name: postgresql-persistent-storage
          mountPath: /var/lib/postgresql/data
      volumes:
        - name: postgresql-persistent-storage
          persistentVolumeClaim: pvc-gold-vvol-vmw
```

```
# kubectl create -f postgres.yaml
```

Figure 18 shows that the same persistent volume can be found in the **Cloud Native Storage/Container Volumes** section.

**Figure 18**

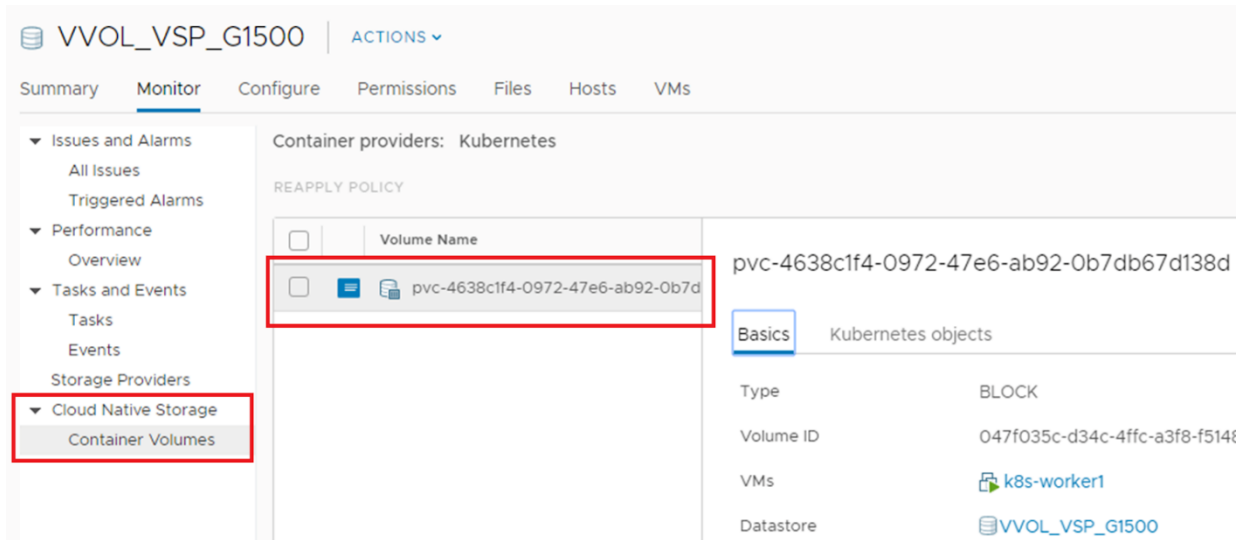


Figure 19 shows that the actual VMDK volumes were created.

**Figure 19**

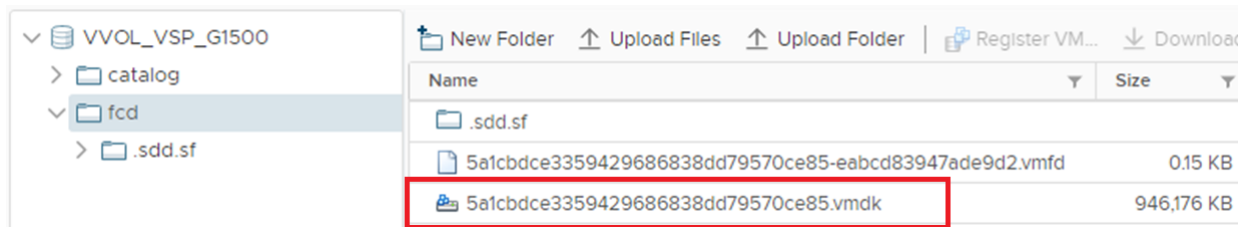


Figure 20 shows the vVol and corresponding VSP G1500 volume (LDEV) ID on the web interface of Hitachi Storage Provider for VMware vCenter.

**Figure 20**

VM Name	VMFS File	VVol Type	Volume
catalog	catalog	Config	00:02:E4
fcd	fcd	Config	00:02:E5
fcd	5a1cbdce3359429686838dd79570ce85-eabcd83947ade9d2.vmf	Other	00:02:E9
fcd	5a1cbdce3359429686838dd79570ce85.vmdk	Data	00:02:E8



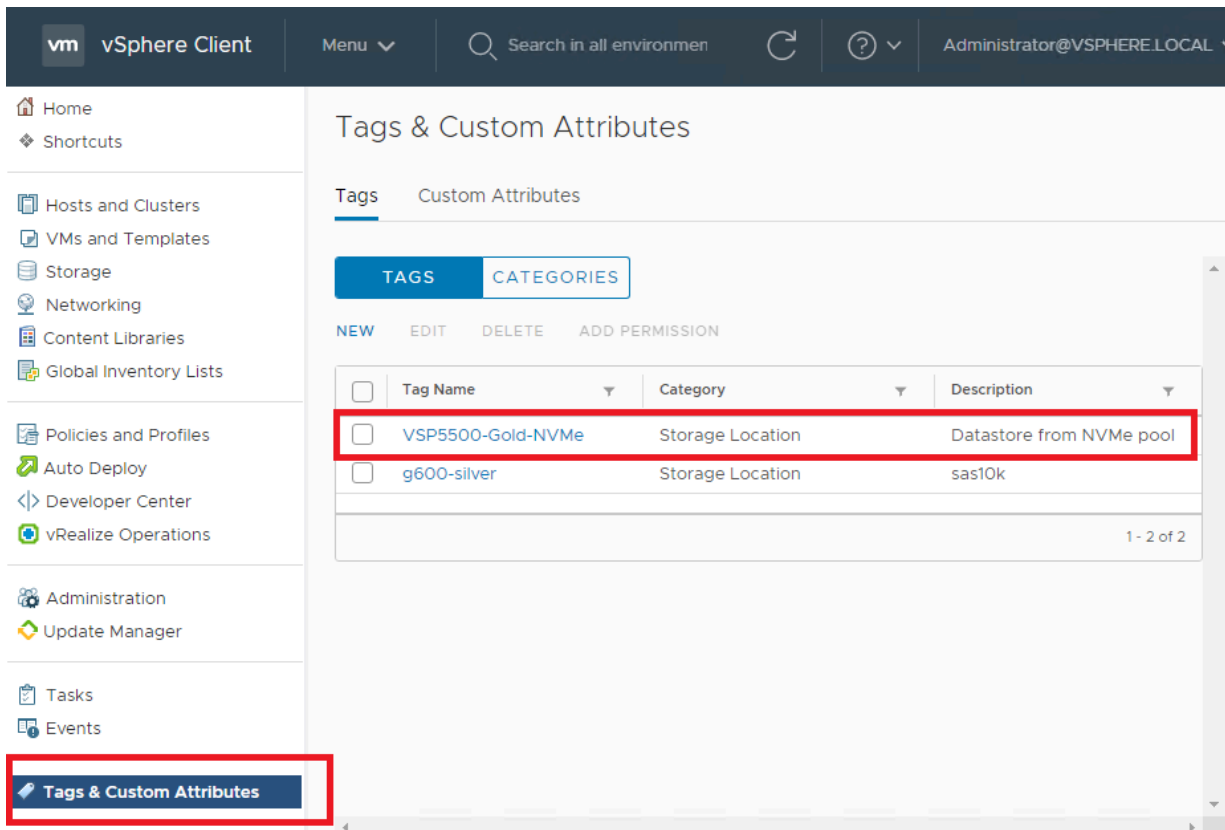
## Create Persistent Volumes from a VMFS Datastore

The VMware CNS CSI driver was installed for virtualized Kubernetes container hosts to provision the persistent volumes from a VMFS datastore. Deploying StatefulSet containers on a VMFS datastore was validated in this section. The following lists the test procedure:

- Virtualized Kubernetes cluster on vSphere hosted on DS120 nodes
- A VMFS datastore provisioned from a LUN/LDEV on VSP 5500 with a storage pool created by NVMe drives
- Create a tag-based VM Storage Policy in vSphere
- VMware CNS CSI driver was installed on the Kubernetes cluster
- Deploy 3 replica of MongoDB StatefulSet with a 100 GB persistent volume

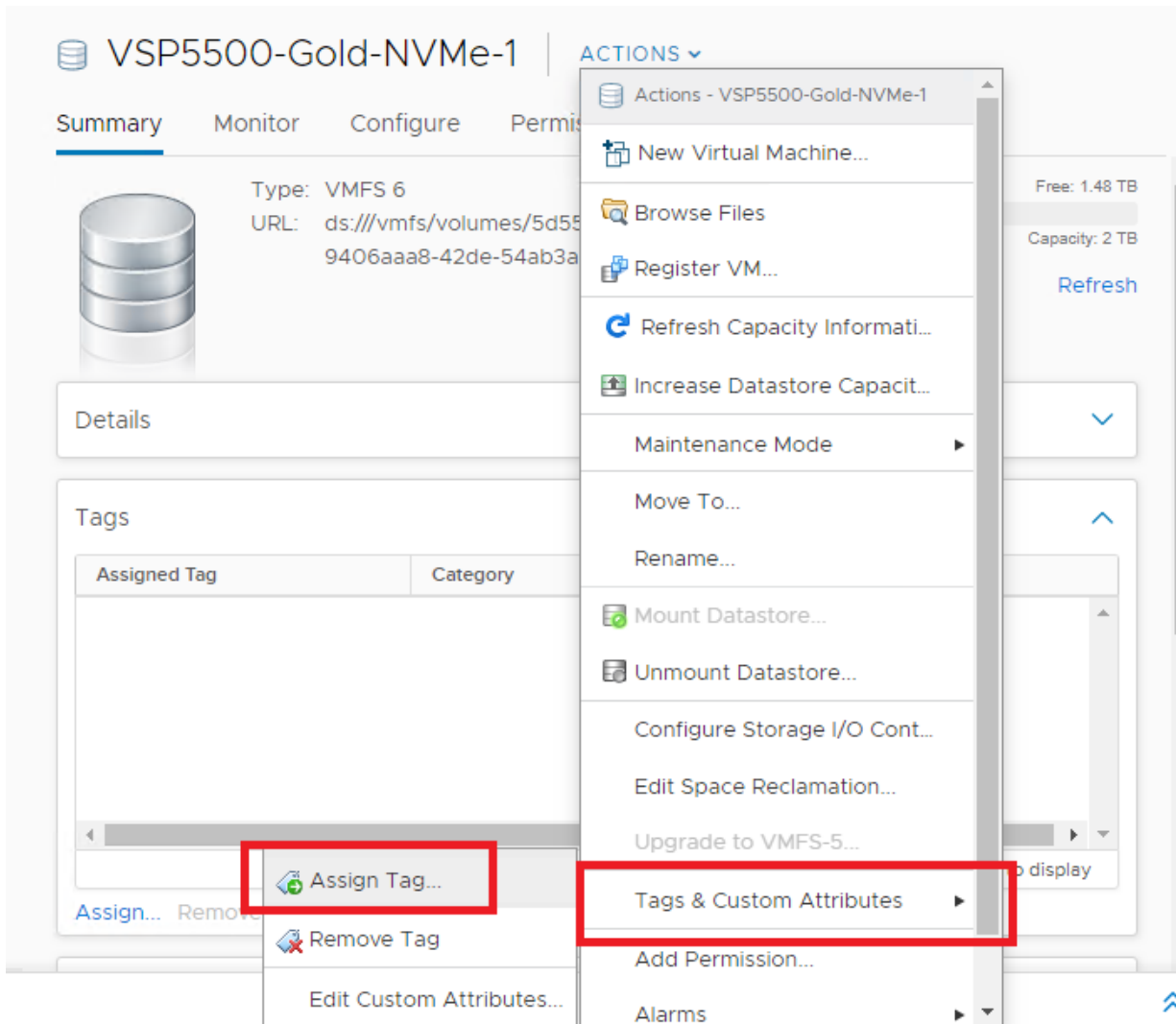
In order to use VMware's SPBM features for a VMFS datastore, the datastore needs to be tagged first. Figure 21 shows **VSP5500-Gold-NVMe** tag was created on vSphere Client.

Figure 21



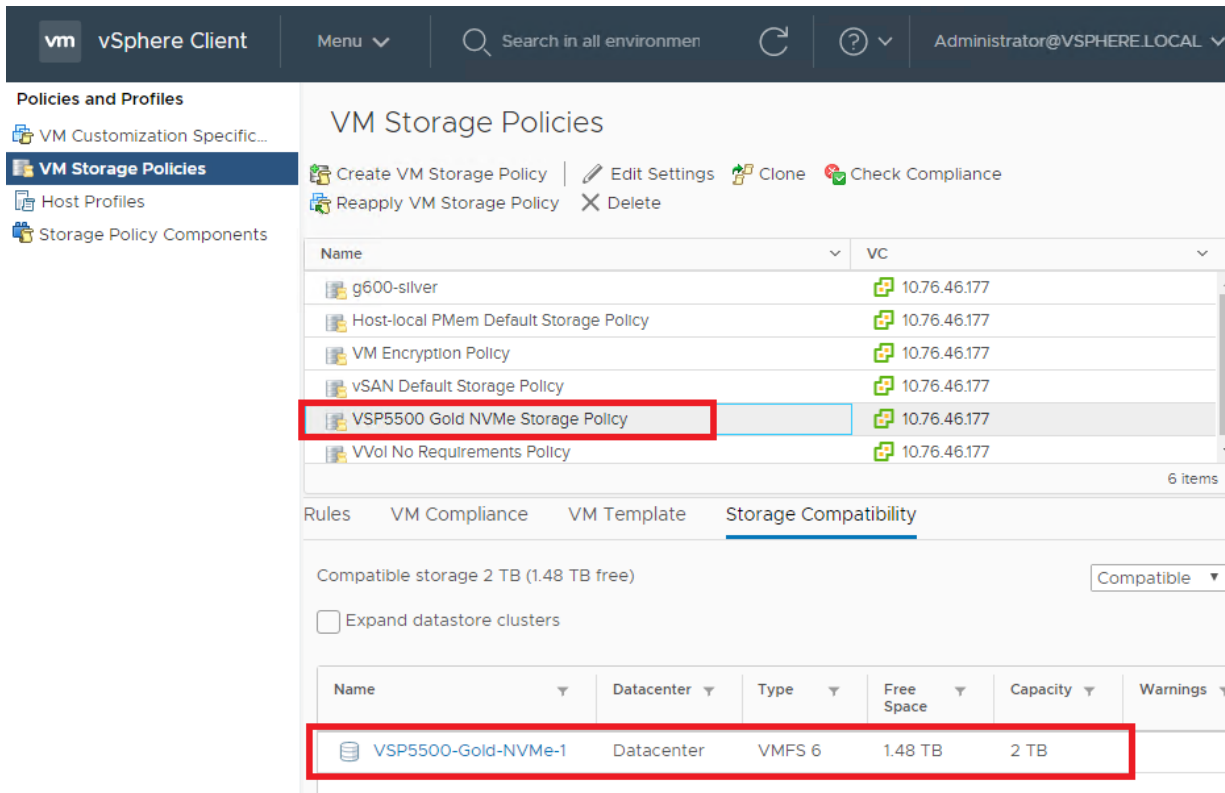
Go to the target datastore and assign a tag. As an example, Figure 22 shows how to assign a tag to the datastore named **VSP5500-Gold-NVMe-1**.

**Figure 22**



After the datastore is tagged, a VM Storage Policy can be created. In this case, as shown in Figure 23, the **VSP5500 Gold NVMe Storage Policy** was created with **VSP5500-Gold-NVMe-1** datastore.

Figure 23



Once the VM Storage Policy is created, go to a Kubernetes Master node and reference it in the StorageClass YAML file **sc-vsp5500-gold-nvme.yaml** as shown below. Use steps similar to the previous section to deploy stateful containers.

```
kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
  name: sc-vsp5500-gold-nvme
  annotations:
    storageclass.kubernetes.io/is-default-class: "false"
provisioner: csi.vsphere.vmware.com
parameters:
  storagepolicysname: "VSP5500 Gold NVMe Storage Policy"
  fstype: ext4
```

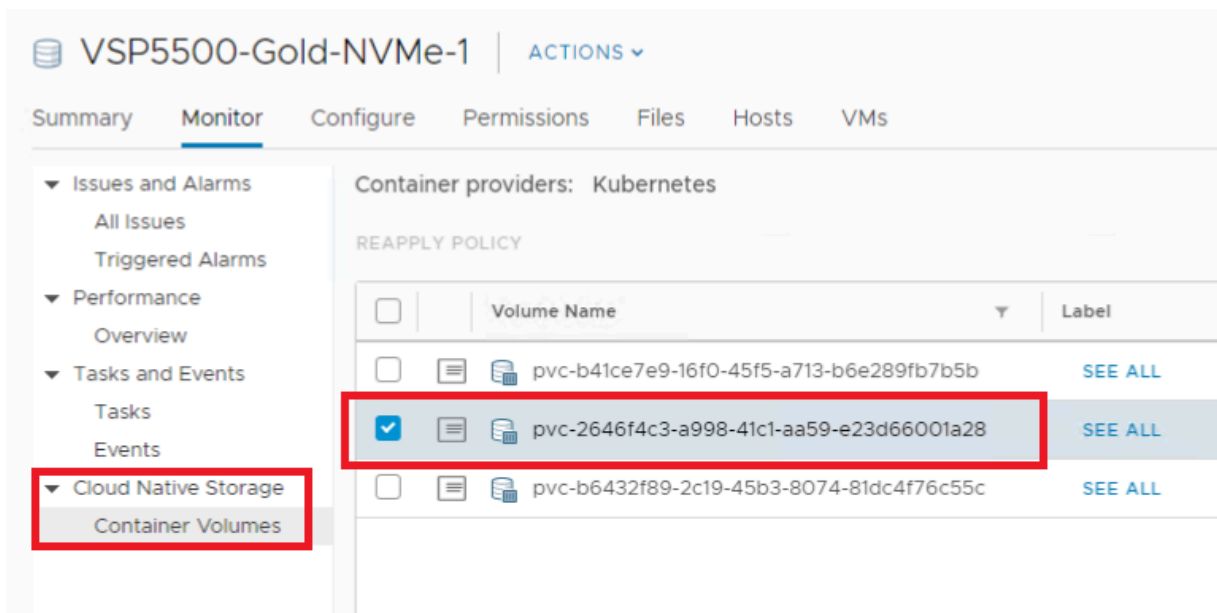
Figure 24 lists the persistent volumes after the MongoDB StatefulSet containers were created.

Figure 24

```
[root@master0 ~]# kubectl get pv
NAME                                     CAPACITY  ACCESS MODES  RECLAIM POLICY  STATUS  CLAIM
STORAGECLASS                            REASON       AGE
pvc-2646f4c3-a998-41c1-aa59-e23d66001a28 100Gi      RWO           Delete          Bound   default/mongodb-persiste
nt-storage-claim-mongod-1                sc-vsp5500-gold-nvme 11m
pvc-3edf5427-3d07-4b06-a148-f9d4765e45c5 512Gi      RWO           Delete          Bound   default/mongodb-persiste
nt-storage-claim-hspc-mongod-hspc-2     sc-vsp5k-gold        28h
pvc-a1660eca-7f28-47ea-9535-23a239648765 512Gi      RWO           Delete          Bound   default/mongodb-persiste
nt-storage-claim-hspc-mongod-hspc-0     sc-vsp5k-gold        28h
pvc-b41ce7e9-16f0-45f5-a713-b6e289fb7b5b 100Gi      RWO           Delete          Bound   default/mongodb-persiste
nt-storage-claim-mongod-0                sc-vsp5500-gold-nvme 11m
pvc-b6432f89-2c19-45b3-8074-81dc4f76c55c 100Gi      RWO           Delete          Bound   default/mongodb-persiste
nt-storage-claim-mongod-2                sc-vsp5500-gold-nvme 10m
pvc-be2be915-60ed-4b2a-b014-7b20da761888 512Gi      RWO           Delete          Bound   default/mongodb-persiste
nt-storage-claim-hspc-mongod-hspc-1     sc-vsp5k-gold        28h
pvc-f44ffdd6-1b29-4be1-80e2-6154dd756895 512Gi      RWO           Delete          Bound   default/mongodb-persiste
nt-storage-claim-hspc-mongod-hspc-3     sc-vsp5k-gold        28h
```

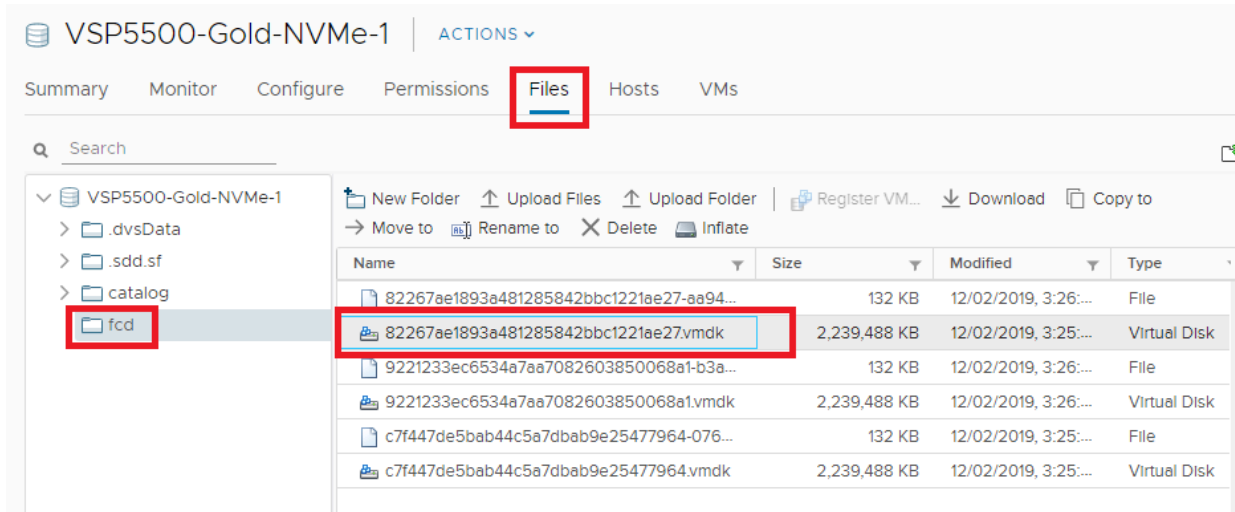
Figure 25 shows that the same persistent volume can be found in the **Cloud Native Storage/Container Volumes** section.

Figure 25



Actual VMDK files can be found under the **fcd** folder of the datastore as shown in Figure 26.

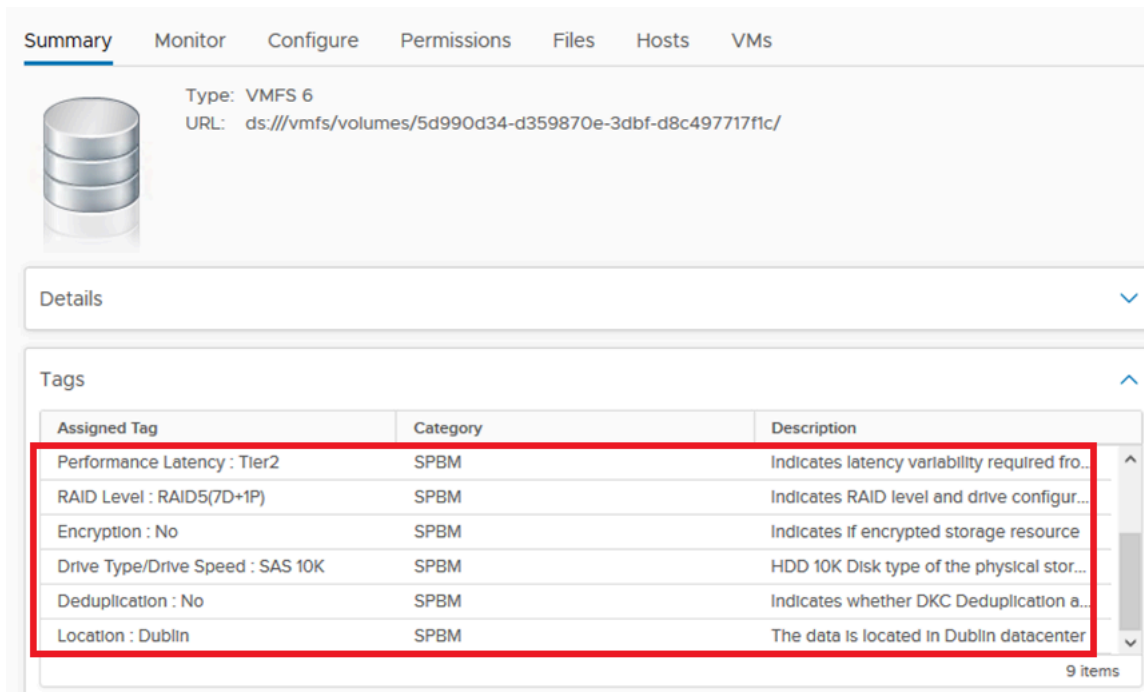
**Figure 26**



### Automatic VMFS Datastore Tagging with Hitachi VASA Provider

The previous steps showed how to tag datastore manually in vSphere client. Alternatively, with Hitachi VASA provider, datastores can be tagged automatically as shown in Figure 27. These tags describe the datastore capability in greater details including performance level, RAID level, disk type, location and more. This provides granular yet effective storage management in large VMware environments.

**Figure 27**



## Create Persistent Volumes from a vSAN Datastore

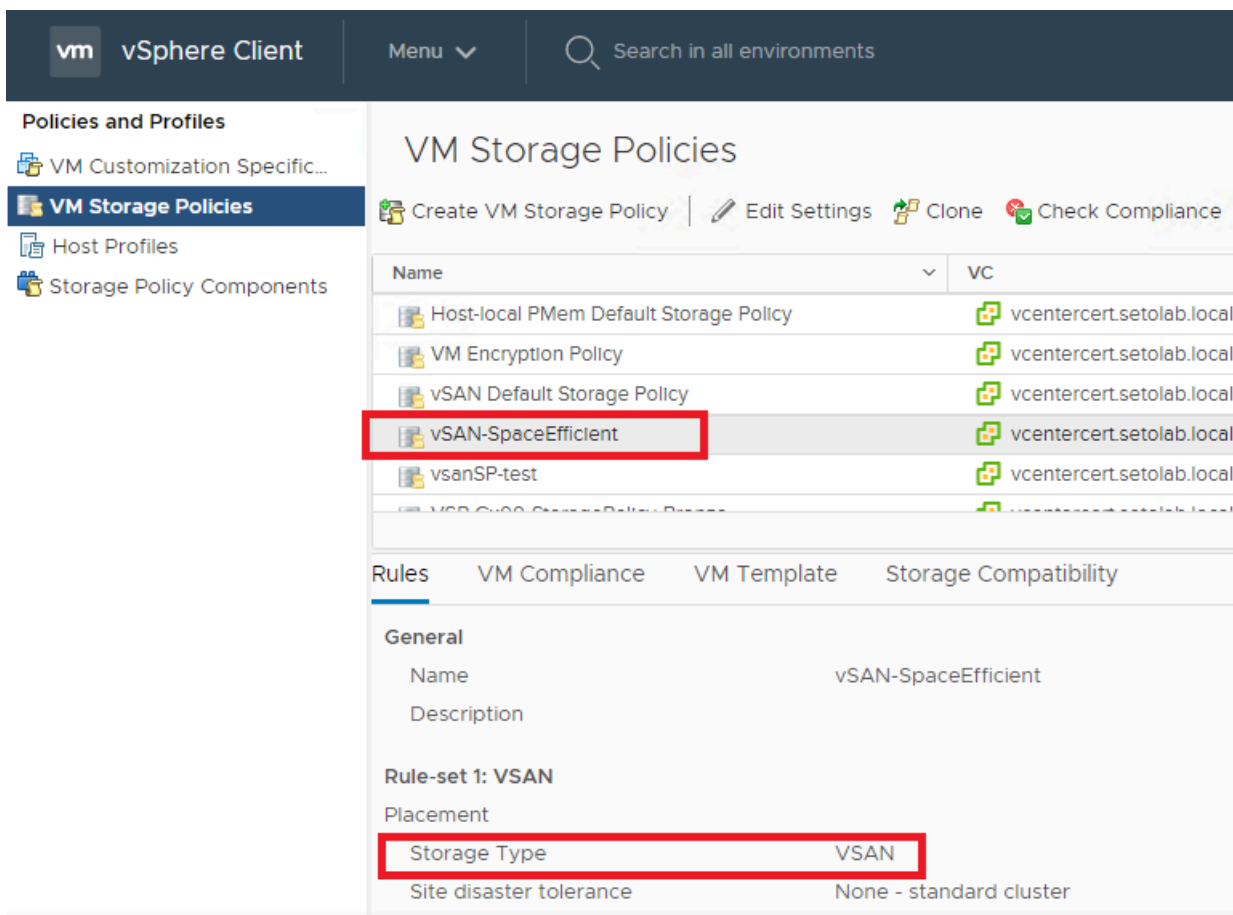
The VMware CNS CSI driver was installed for virtualized Kubernetes container hosts to provision the persistent volumes from a vSAN datastore. Deploying StatefulSet containers on a vSAN datastore was validated in this section. The following lists the test procedure:

- Deploy a virtualized Kubernetes cluster on vSphere with a vSAN datastore hosted on four DS120 nodes
- Create a VM Storage Policy for vSAN datastore
- Install VMware CNS CSI driver on the Kubernetes cluster
- Deploy 3 replica of MongoDB StatefulSet with a 100 GB persistent volume

The vSAN datastore has a default VM Storage Policy with RAID 1 (mirroring) data protection level. However, a RAID 5 or RAID 6 configuration consumes less capacity compared to RAID 1. Figure 28 shows a new VM Storage Policy: **vSAN-SpaceEfficient** was created with RAID 5 configuration. For more information about vSAN RAID settings, see the following VMware documentation link:

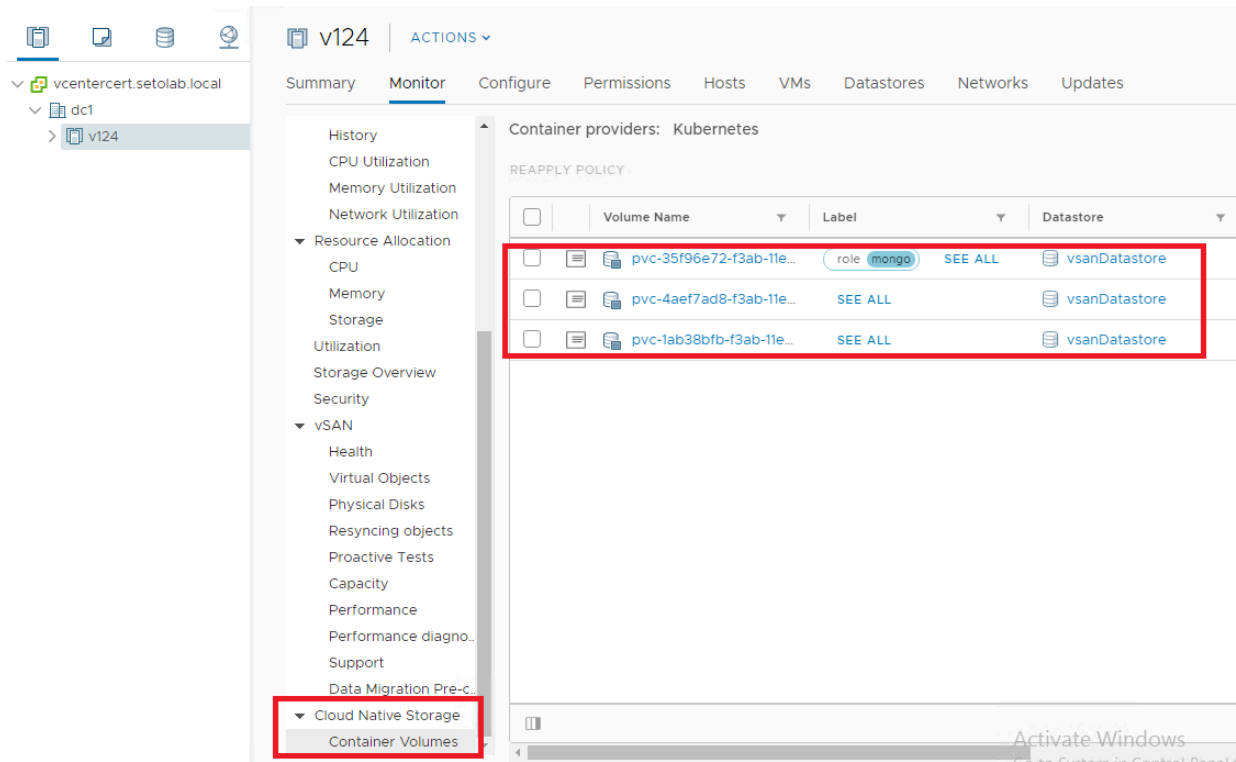
[Using RAID 5 or RAID 6 Erasure Coding](#)

Figure 28



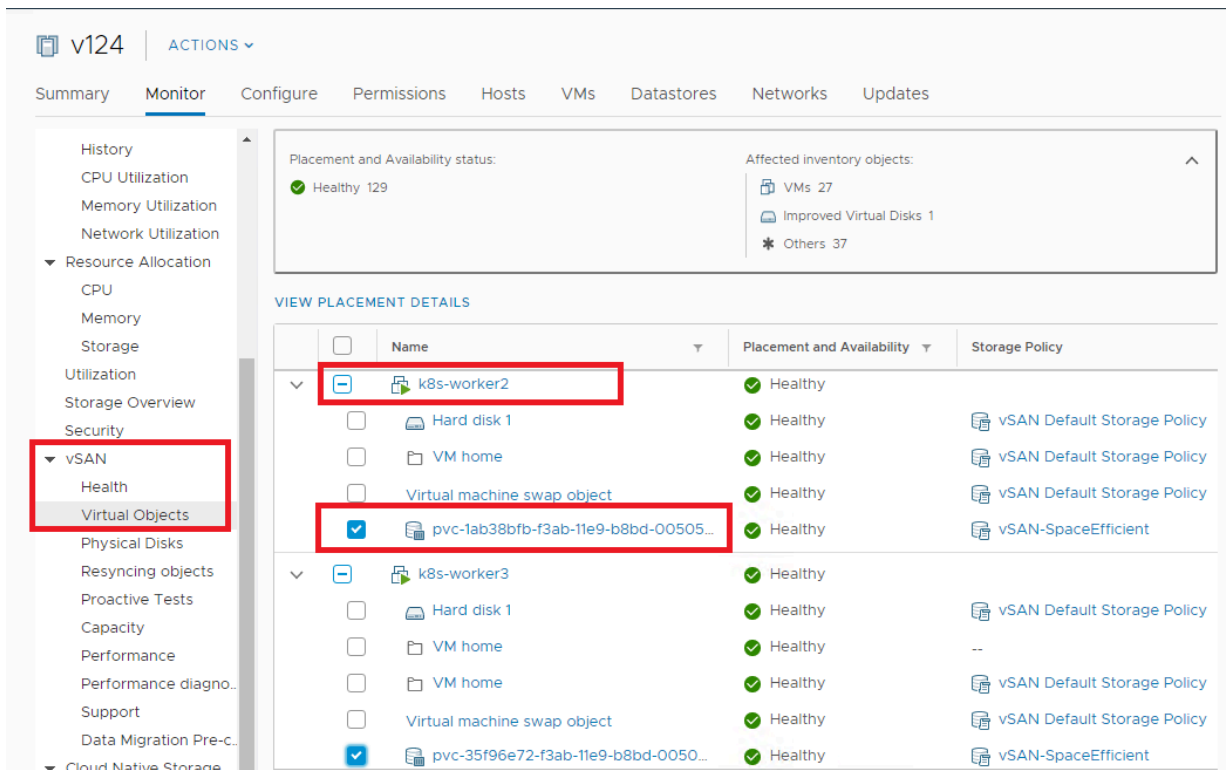
Use steps similar to the previous section to deploy MongoDB containers on vSAN datastores. Figure 27 shows three persistent volumes were created from vSAN datastore.

**Figure 29**



For vSAN datastore, there is more detailed information under the vSAN/Virtual Objects. Figure 30 shows one of the persistent volumes was mounted on **k8s-worker2** VM with healthy status.

**Figure 30**



## Conclusion

Hitachi UCP CI with Kubernetes gives container platforms enterprise scale, efficiency, flexibility, performance, and resiliency. Hitachi VSP storage systems meet and exceed persistent volume needs for any container workloads. This solution provides the foundation a modern IT Infrastructure needs to support a private cloud environment.

## Appendix A - Product Descriptions

### Hardware Components

This section describes the hardware components available for UCP 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 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. 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 solid state disks.

#### *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.

#### *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 Storage*

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.

### *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 to consolidate 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 93180LC-EX](#) is a 1U-sized spine switch with 14 ports 100 GbE (downlink), 28 ports 40 GbE (downlink), and 6 ports 100 GbE (uplink) for multiple-rack solutions. Uplink ports also support 4 × 10 GbE or 4 × 25 GbE breakouts, if needed.
- [Cisco Nexus 93180YC-EX](#) 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 3048TP](#) is a 1U-sized management switch with 48 ports 1 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**

This section describe 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 includes standards-based management software on a Hitachi Command Suite (HCS) base. This provides storage configuration and control capabilities for you.

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).

### *Hitachi Unified Compute Platform Advisor*

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

The key management and deployment benefits include:

- Provide a single management interface for all physical and virtual infrastructure resources
- Provision and manage both physical and virtual storage
- Manage virtual networking, including synchronization up to the physical network switch
- Deploy and manage physical compute servers
- Provision onboard storage, compute and network configuration, and import existing configuration via a .csv file

- Provide simple and efficient firmware upgrade and management
- Facilitate configuration of VMware software, including vRealize Orchestrator and vRealize Log Insight
- Accomplish complex IT operations and workflows such as smart provisioning, data replication, and VMware provisioning through orchestration
- Manage business continuity aspects with regards to backup and data replication scenarios

### *Hitachi Storage Provider for VMware vCenter (VASA)*

[Hitachi Storage Provider for VMware vCenter](#) enables organizations to deploy software-defined storage solutions for VMware Virtual Volumes (vVols) on Hitachi Virtual Storage Platform (VSP) systems and the Hitachi NAS Platform. It provides application-centric granular management and storage-policy-based management for agile, simple and efficient IT services. It reduces the operational burden between VI administrators and storage administrators for faster and smarter VM and application services provisioning.

### *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 (VSP) G series and F series enterprise-class hybrid and all-flash storage systems. With the compatibility plug-in, your organization can deliver shared storage for containers that persists beyond the timeline of a single container host.

### *Kubernetes*

[Kubernetes](#) is an open-source system for automating deployment, scaling, and management of containerized applications. Its groups containers that make up an application into logical units for easy management and discovery. Kubernetes builds upon 15 years of experience of running production workloads at Google, combined with best-of-breed ideas and practices from the community. It has feature such as storage orchestration, self-healing, automated rollouts and rollbacks and horizontal scaling.

### *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

### *Red Hat Enterprise Linux*

[Red Hat Enterprise Linux](#) (RHEL) 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.

### *Docker Engine*

[Docker Engine](#) is the industry's de facto container runtime that runs on various Linux (CentOS, Debian, Fedora, Oracle Linux, RHEL, SUSE, and Ubuntu) and Windows Server operating systems.

## Hitachi Vantara



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

Contact Information  
Phone: 1-800-446-0744  
Sales: 1-858-225-2095  
[HitachiVantara.com/contact](http://HitachiVantara.com/contact)

© Hitachi Vantara Corporation, 2019. All rights reserved. HITACHI is a trademark or registered trademark of Hitachi, Ltd. Microsoft and Windows Server are trademarks or registered trademarks of Microsoft Corporation. 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 Corporation.

MK-SL-182-00, December 2019