

Getting Started with VMware Cloud Native Storage

Update 3

Modified on 25 MAR 2020

VMware vSphere 6.7

vCenter Server 6.7

VMware ESXi 6.7



vmware®

You can find the most up-to-date technical documentation on the VMware website at:

<https://docs.vmware.com/>

If you have comments about this documentation, submit your feedback to

docfeedback@vmware.com

VMware, Inc.
3401 Hillview Ave.
Palo Alto, CA 94304
www.vmware.com

Copyright © 2019-2020 VMware, Inc. All rights reserved. [Copyright and trademark information.](#)

Contents

About Getting Started with VMware Cloud Native Storage 4

Updated Information 5

- 1 Understanding Cloud Native Storage 6**
 - Cloud Native Storage Concepts and Terminology 6
 - Cloud Native Storage Users 8
 - Cloud Native Storage Components 9
 - Cloud Native Storage Roles and Privileges 10

- 2 Cloud Native Storage for vSphere Administrators 12**
 - Requirements for Cloud Native Storage 12
 - Create a Storage Policy 13
 - Configure Kubernetes Cluster Virtual Machines 14
 - Monitor Container Volumes Across Kubernetes Clusters 15

About Getting Started with VMware Cloud Native Storage

The *Getting Started with VMware Cloud Native Storage* documentation provides information about VMware[®] Cloud Native Storage, a vSphere and Kubernetes solution that offers comprehensive data management for stateful applications in the vSphere environment.

This information includes a brief overview of the Cloud Native Storage concepts and components. It also covers tasks that vSphere administrators perform to provide persistent storage resources to a Kubernetes cluster and to monitor those resources using the vSphere Client.

Intended Audience

The information is intended for vSphere administrators who have a basic understanding of Kubernetes and are familiar with container deployment concepts.

Kubernetes users who want to run Kubernetes clusters and containerized applications on vSphere can refer to the [Kubernetes vSphere Cloud Provider](#) documentation on GitHub.

VMware Technical Publications Glossary

VMware Technical Publications provides a glossary of terms that might be unfamiliar to you. For definitions of terms as they are used in VMware technical documentation, go to <http://www.vmware.com/support/pubs>.

Updated Information

This *Getting Started with VMware Cloud Native Storage* is updated with each release of the product or when necessary.

This table provides the update history of the *Getting Started with VMware Cloud Native Storage*.

| Revision | Description |
|-------------|---|
| 25 MAR 2020 | Sections that included tasks for Kubernetes uses have been removed from the <i>Getting Started with VMware Cloud Native Storage</i> documentation. Kubernetes users who want to run Kubernetes clusters and containerized applications on vSphere can refer to the Kubernetes vSphere Cloud Provider documentation on GitHub. This document also includes information about the vSphere CSI driver installation, deployment, and sample YAMLS. |
| 04 MAR 2020 | Minor edits. |
| 22 OCT 2019 | Sections related to installation of the vSphere CSI plug-in and deployment of a Kubernetes cluster have been removed from this document. For information, see the Kubernetes vSphere Cloud Provider documentation. |
| 01 OCT 2019 | Cloud Native Storage Components has been corrected to state that the vSphere Container Storage Interface supports a single vCenter Server. |
| 20 AUG 2019 | Initial release. |

Understanding Cloud Native Storage



Cloud Native Storage is a solution that provides comprehensive data management for stateful applications. When you use Cloud Native Storage, you can create the containerized stateful applications capable of surviving restarts and outages. Stateful containers leverage storage exposed by vSphere while using such primitives as standard volume, persistent volume, and dynamic provisioning.

With Cloud Native Storage, you can create persistent container volumes independent of virtual machine and container life cycle. vSphere storage backs the volumes, and you can set a storage policy directly on the volumes. After you create the volumes, you can review them and their backing virtual disks in the vSphere Client, and monitor their storage policy compliance.

This chapter includes the following topics:

- [Cloud Native Storage Concepts and Terminology](#)
- [Cloud Native Storage Users](#)
- [Cloud Native Storage Components](#)
- [Cloud Native Storage Roles and Privileges](#)

Cloud Native Storage Concepts and Terminology

Be familiar with several concepts essential to the vSphere Cloud Native Storage environment.

| | |
|-------------------------------|--|
| Kubernetes Cluster | A cluster of VMs where Kubernetes control plane and worker services are running. On top of the Kubernetes cluster, you deploy your containerized applications. Applications can be stateful and stateless. |
| Pod | A pod is a group of one or more containers that share such resources as storage and network. Containers inside a pod are started, stopped, and replicated as a group. |
| Container Orchestrator | Open-source platforms, such as Kubernetes, for deployment, scaling, and management of containerized applications across clusters of hosts. The platforms provide a container-centric infrastructure. |
| Stateful Application | As containerized applications evolve from stateless to stateful, they require persistent storage. Unlike stateless applications that do not save data between sessions, stateful applications save data to persistent storage. |

The retained data is called the application's state. You can later retrieve the data and use it in the next session. Most applications are stateful. A database is as an example of a stateful application.

PersistentVolume

Stateful applications use PersistentVolumes to store their data. A PersistentVolume is a Kubernetes volume capable of retaining its state and data. It is independent of a pod and can continue to exist even when the pod is deleted or reconfigured. In the vSphere environment, the PersistentVolume objects use virtual disks (VMDKs) as their backing storage.

StorageClass

Kubernetes uses a StorageClass to define different tiers of storage and to describe different types of requirements for storage backing the PersistentVolume. In the vSphere environment, a storage class can be linked to a storage policy. As a vSphere administrator, you create storage policies that describe different storage requirements. The VM storage policies can be used as a part of StorageClass definition for dynamic volume provisioning.

The following sample YAML file references the **Gold** storage policy that you created earlier using the vSphere Client. The resulting persistent volume VMDK is placed on a compatible datastore that satisfies the **Gold** storage policy requirements.

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

PersistentVolumeClaim

Typically, applications or pods can request persistent storage through a PersistentVolumeClaim. The PersistentVolumeClaim specifies the type and class of storage, the access mode, either ReadWriteOnce or ReadWriteMany, and other parameters for the PersistentVolume. The request can then dynamically provision the corresponding PersistentVolume object and the underlying virtual disk in the vSphere environment.

Once the claim is created, the PersistentVolume is automatically bound to the claim. Pods use the claim to mount the PersistentVolume and access storage.

When you delete this claim, the corresponding PersistentVolume object and the underlying storage are deleted.

```
kind: PersistentVolumeClaim
metadata:
  name: persistent-VMDK
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 5Gi
  storageClassName: gold-sc
```

StatefulSet

A StatefulSet manages the deployment and scaling of your stateful applications. The StatefulSet is valuable for applications that require stable identifiers or stable persistent storage. You can configure the StatefulSet to include a volumeClaimTemplates entry that automatically generates the PersistentVolumeClaim objects.

Cloud Native Storage Users

The types of users involved in the process of creating and monitoring Kubernetes volumes in the vSphere Cloud Native Storage environment generally fall into two categories, a Kubernetes user and a vSphere administrator. Both types of users have access to different tools and perform different tasks.

CNS Kubernetes User

The Kubernetes user might be a Kubernetes developer and an application owner, a Kubernetes administrator, or combine functions of both. The tasks that the Kubernetes user performs in the Cloud Native Storage environment include the following:

- Deploy and manage the vSphere CSI. For information, see the [Deploying a Kubernetes Cluster on vSphere with CSI and CPI](#) section of the [Kubernetes vSphere Cloud Provider](#) documentation on GitHub.
- Deploy and manage stateful applications. For information, see the [Sample manifests to test CSI driver functionality](#) section of the [Kubernetes vSphere Cloud Provider](#) documentation on GitHub.
- Perform life cycle operations for persistent volumes.
- Perform life cycle operations for storage classes.

CNS vSphere User

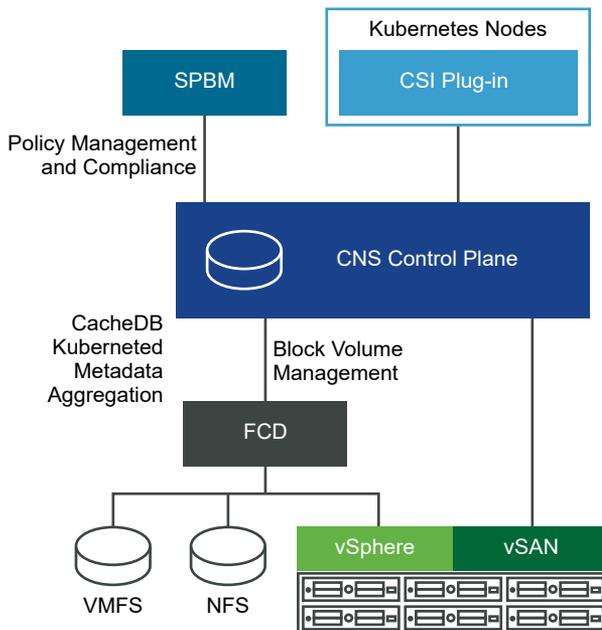
A CNS vSphere user, or a vSphere administrator, has access to the vSphere Client to perform the following tasks:

- Perform life cycle operations for the VM storage policies. For example, create a VM storage policy to be used for a Kubernetes storage class and communicate its name to the Kubernetes user. See [Create a Storage Policy](#).
- Use the Cloud Native Storage section of the vSphere Client to monitor health and storage policy compliance of the container volumes across the Kubernetes clusters. See [Monitor Container Volumes Across Kubernetes Clusters](#).

Cloud Native Storage Components

Cloud Native Storage uses several components to integrate with vSphere storage.

The following illustration shows how these components interact.



Kubernetes Cluster

In the Cloud Native Storage environment, a Kubernetes cluster is a cluster of virtual machines, or nodes, deployed in vSphere. A Kubernetes user directly interacts with the cluster when deploying stateful applications on top of it.

vSphere Container Storage Interface (CSI)

The vSphere CSI is an interface that exposes vSphere storage to containerized workloads on container orchestrators, such as Kubernetes. It enables vSAN and other types of vSphere storage.

On Kubernetes, the CSI driver is used with the out of tree vSphere Cloud Controller Manager (CCM).

The Container Storage Interface supports dynamic provisioning of container volumes.

The interface supports the following functionalities:

- The vSphere First Class Disk functionality.
- Kubernetes zones.
- Conventional and raw mounts.
- Single vCenter Server, and multiple data centers and clusters.
- Provisioning from multiple datastores or datastore clusters.

On Kubernetes, the CSI driver is used with the out-of-tree vSphere Cloud Provider Interface (CPI). The CSI driver is shipped as a container image and must be deployed by the cluster administrator. For information, see the [Deploying a Kubernetes Cluster on vSphere with CSI and CPI](#) section of the [Kubernetes vSphere Cloud Provider](#) documentation on GitHub.

Cloud Native Storage Server Component

The CNS server component resides in vCenter Server. It is an extension of vCenter Server management that implements the provisioning and life cycle operations for the container volumes.

When provisioning container volumes, it interacts with the First Class Disk functionality to create virtual disks that back the volumes. In addition, the CNS server component communicates with the Storage Policy Based Management to guarantee a required level of service to the disks.

The CNS also performs query operations that allow you to manage and monitor container volumes and their backing virtual disks through vCenter Server.

First Class Disk (FCD)

Also called Improved Virtual Disk. It is a named virtual disk unassociated with a VM. These disks reside on a VMFS, NFS, or vSAN datastore and back container volumes.

Storage Policy Based Management

Storage Policy Based Management is a vCenter Server service that supports provisioning of persistent volumes according to specified storage requirements. After provisioning, the service monitors compliance of the volume with the required policy characteristics.

Cloud Native Storage Roles and Privileges

The vSphere user must have specific privileges to perform operations related to Cloud Native Storage.

You can create several roles to assign sets of permissions on the objects that participate in the Cloud Native Storage environment.

For more information about roles and permissions in vSphere, and how to create a role, see the *vSphere Security* documentation.

| Role Name | Privilege Name | Description | Required On |
|---------------|--|---|---|
| CNS-SPBM | Profile-driven storage > Profile-driven storage update | Allows changes to be made to VM storage policies, such as creating and updating storage VM storage policies. | Root vCenter Server |
| | Profile-driven storage > Profile-driven storage view | Allows viewing of defined storage policies. | |
| CNS-VM | Virtual machine > Configuration > Add existing disk | Allows adding an existing virtual disk to a virtual machine. | All cluster node VMs |
| | Virtual Machine > Configuration > Add or remove device | Allows addition or removal of any non-disk device. | |
| CNS-Datastore | Datastore > Low level file operations | Allows performing read, write, delete, and rename operations in the datastore browser. | Shared datastore where persistent volumes reside |
| Read-only | Default role | <p>Users with the Read Only role for an object are allowed to view the state of the object and details about the object. For example, users with this role can find the shared datastore accessible to all node VMs.</p> <p>For zone and topology-aware environments, all ancestors of node VMs, such as a host, cluster, and data center must have the Read-only role set for the vSphere user configured to use the CSI driver and CCM. This is required to allow reading tags and categories to prepare the nodes' topology.</p> | All hosts where the nodes VMs reside Data center |
| CNS UI | privilege.Cns.label > privilege.Cns.Searchable.label | Allows storage administrator to see CNS UI. | |

Cloud Native Storage for vSphere Administrators

2

A vSphere administrator delivers storage resources to the Kubernetes team and creates VM storage policies that describe different storage requirements and classes of services. After the Kubernetes workloads with persistent storage are provisioned, the vSphere administrator can monitor the life cycle of the backing storage resources and their compliance to the requirements.

This chapter includes the following topics:

- [Requirements for Cloud Native Storage](#)
- [Create a Storage Policy](#)
- [Configure Kubernetes Cluster Virtual Machines](#)
- [Monitor Container Volumes Across Kubernetes Clusters](#)

Requirements for Cloud Native Storage

Your Cloud Native Storage environment and virtual machines that participate in the Kubernetes cluster must meet several requirements.

Cloud Native Storage Requirements

- vSphere 6.7 Update 3 or later.
- Kubernetes version 1.14 and later.
- A Kubernetes cluster deployed on the virtual machines. For details about deploying the vSphere CSI plug-in and running the Kubernetes cluster on vSphere, see the [Kubernetes vSphere Cloud Provider](#) documentation in GitHub.

Requirements for Kubernetes Cluster Virtual Machines

- Virtual machines with hardware version 15 or later. Install VMware Tools on each node virtual machine.
- Virtual machine hardware recommendations:
 - Set CPU and memory adequately based on workload requirements.
 - Use the VMware Paravirtual SCSI controller for the primary disk on the Node VM.
- All virtual machines must have access to a shared datastore, such as vSAN.

- Set the `disk.EnableUUID` parameter on each node VM. See [Configure Kubernetes Cluster Virtual Machines](#).
- To avoid errors and unpredictable behavior, do not take snapshots of CNS node VMs.

Create a Storage Policy

The virtual disk (VMDK) that will back your containerized application needs to meet specific storage requirements. As a vSphere user, you create a VM storage policy based on the requirements provided to you by the Kubernetes user.

The storage policy will be associated with the VMDK backing your application.

If you have multiple vCenter Server instances in your environment, create the VM storage policy on each instance. Use the same policy name across all instances.

Prerequisites

- The Kubernetes user identifies the Kubernetes cluster where the stateful containerized application will be deployed.
- The Kubernetes user collects storage requirements for the containerized application and communicates them to the vSphere user.
- Required privileges: **VM storage policies. Update** and **VM storage policies. View**.

Procedure

- 1 In the vSphere Client, open the **Create VM Storage Policy** wizard.
 - a Click **Menu > Policies and Profiles**.
 - b Under **Policies and Profiles**, click **VM Storage Policies**.
 - c Click **Create VM Storage Policy**.
- 2 Enter the policy name and description, and click **Next**.

| Option | Action |
|-----------------------|--|
| vCenter Server | Select the vCenter Server instance. |
| Name | Enter the name of the storage policy, for example Space-Efficient . |
| Description | Enter the description of the storage policy. |

- 3 On the **Policy structure** page under Datastore-specific rules, select **Enable rules for vSAN storage** and click **Next**.
- 4 On the **vSAN** page, define the policy rule set and click **Next**.
 - a On the **Availability** tab, define the **Site disaster tolerance** and **Failures to tolerate**.
 - b On the **Advanced Policy Rules** tab, define advanced policy rules, such as number of disk stripes per object and flash read cache reservation.

- 5 On the **Storage compatibility** page, review the list of vSAN datastores that match this policy and click **Next**.
- 6 On the **Review and finish** page, review the policy settings, and click **Finish**.

| General | |
|----------------|---|
| Name | Space-Efficient |
| Description | vCenter Server |
| vCenter Server | sc2-rdops-vm08-dhcp-23-199.eng.vmware.com |

| vSAN | |
|-------------------------|-------------------------|
| Availability | |
| Site disaster tolerance | None - standard cluster |
| Failures to tolerate | No data redundancy |

| Advanced Policy Rules | |
|-----------------------------------|-------------------|
| Number of disk stripes per object | 1 |
| IOPS limit for object | 0 |
| Object space reservation | Thin provisioning |
| Flash read cache reservation | 0% |
| Disable object checksum | No |
| Force provisioning | No |

CANCEL BACK FINISH

What to do next

You can now inform the Kubernetes user of the storage policy name. The VM storage policy you created will be used as a part of storage class definition for dynamic volume provisioning.

Configure Kubernetes Cluster Virtual Machines

On each node VM, enable the `disk.EnableUUID` parameter, so that the VMs can successfully mount the virtual disks.

Perform these steps for each of the VM nodes that participate in the cluster.

Prerequisites

- Create several VMs for your Kubernetes cluster. For the VM requirements, see [Requirements for Cloud Native Storage](#).
- Required privilege: **Virtual machine. Configuration. Settings**.

Note To avoid errors and unpredictable behavior, do not take snapshots of CNS node VMs.

Procedure

- 1 In the vSphere Client, right-click the virtual machine and select **Edit Settings**.

- 2 Click the **VM Options** tab and expand the **Advanced** menu.
- 3 Click **Edit Configuration** next to Configuration Parameters.
- 4 Configure the `disk.EnableUUID` parameter.

If the parameter exists, make sure that its value is set to True. If the parameter is not present, add it and set its value to True.

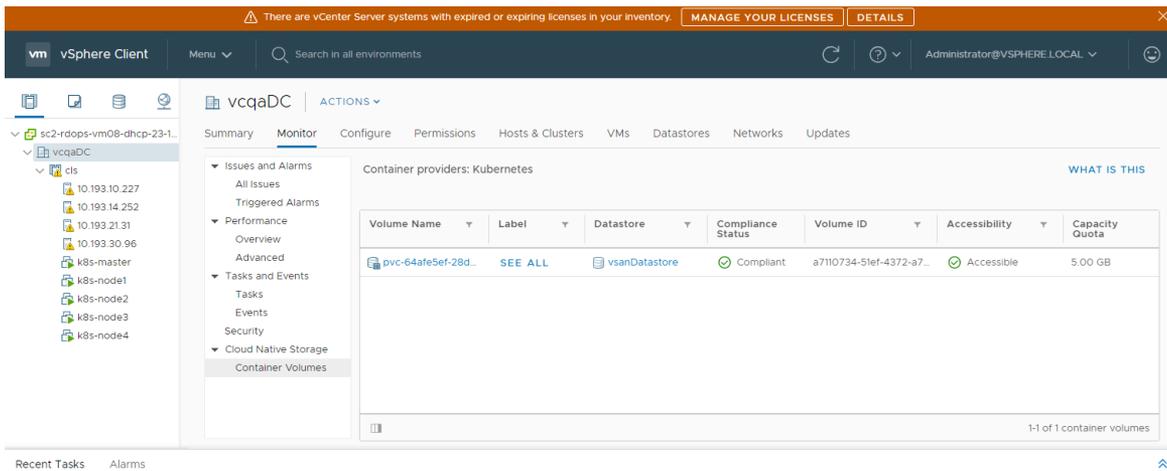
| Name | Value |
|------------------------------|-------|
| <code>disk.EnableUUID</code> | True |

Monitor Container Volumes Across Kubernetes Clusters

Display container volumes in your environment and monitor their storage policy compliance status.

Procedure

- 1 Navigate to the vCenter Server instance, a data center, or a datastore.
- 2 Click the **Monitor** tab and click **Container Volumes** under **Cloud Native Storage**.
- 3 Observe the container volumes available in your environment and monitor their storage policy compliance status.



- 4 Click the **SEE ALL** link in the Label column to view additional details.
- 5 Click the link in the Volume Name column to review such details as placement, compliance, and storage policy.

Note This view is available only when the underlying datastore is vSAN.

The screenshot displays the vSphere Client interface. At the top, a notification bar indicates that there are vCenter Server systems with expired or expiring licenses in the inventory, with buttons for 'MANAGE YOUR LICENSES' and 'DETAILS'. The main header shows 'vSphere Client' and a search bar. The left sidebar contains a tree view of the environment, including a cluster named 'cls' with several hosts listed below it.

The main content area is divided into several sections:

- Summary:** Shows 'Placement and Availability status: Healthy 22' and 'Affected inventory objects: VMs 5, Improved Virtual Disks 1'.
- Monitor:** The active tab, showing a table of storage objects and their policies.
- Table:** A table with columns for 'Name', 'Placement and Availability', and 'Storage Policy'. It lists various objects like 'VM home', 'Virtual machine swap object', and 'Hard disk 1' and '2' for hosts 'k8s-node3' and 'k8s-node4'. All objects are shown as 'Healthy'.

At the bottom of the interface, there are tabs for 'Recent Tasks' and 'Alarms'.