SDL*

[Hitachi Adapters] HSPC[HPE]

version 8
Printed on 09/01/2025 19:14:57

Contents

1. Hewlett Packard Enterprise Development LP 3

2. Overview..... 4
[CHANGED] 2.1. About HPE XP Storage Plug-in for Containers..... 4
2.2. About the environment setup tasks..... 5
2.3. Requirements..... 6
[CHANGED] 2.3.1. Container orchestrators to be supported..... 6
[CHANGED] 2.3.2. Server requirements..... 6
[CHANGED] 2.3.3. Storage requirements..... 6
2.3.4. Network requirements..... 6
2.4. Pre-installation tasks..... 7
[CHANGED] 2.4.1. Server pre-installation..... 7
[CHANGED] 2.4.2. Storage pre-installation for HPE Storage..... 10

3. Installation..... 12
[CHANGED] 3.1. Installation on OpenShift..... 12
[CHANGED] 3.2. Installation on Kubernetes..... 12
[CHANGED] 3.3. Configuration of Storage Plug-in for Containers instance..... 13
[CHANGED] 3.4. Environment variables..... 14

4. Usage..... 16
[CHANGED] 4.1. Secret settings..... 16
[CHANGED] 4.2. StorageClass settings..... 16
[CHANGED] 4.3. PersistentVolumeClaim settings..... 18
4.4. Pod settings..... 19
4.5. Command examples..... 19
[CHANGED] 4.6. Volume snapshot..... 21
[CHANGED] 4.7. Volume cloning..... 23
[CHANGED] 4.8. Volume expansion..... 24
4.9. Raw block volume..... 24
4.10. ReadOnlyMany..... 25
[CHANGED] 4.11. Resource partitioning..... 26
4.12. Static provisioning..... 29
[CHANGED] 4.12.1. Requirements for using static provisioning..... 29
[CHANGED] 4.12.2. Creating Secret and StorageClass..... 30
[CHANGED] 4.12.3. Creating a PV..... 30
[CHANGED] 4.12.4. Creating a PVC..... 32
[CHANGED] 4.12.5. Troubleshooting when using static provisioning..... 33

[CHANGED] 5. Upgrade..... 35
5.1. Upgrade on OpenShift..... 35
[CHANGED] 5.2. Upgrade on Kubernetes..... 35

6. Re-creation..... 36
[CHANGED] 6.1. Deleting Storage Plug-in for Containers..... 36
6.2. Re-creating Storage Plug-in for Containers..... 36

7. Uninstallation..... 37
[CHANGED] 7.1. Uninstallation on OpenShift..... 37

[CHANGED] 7.2. Uninstallation on Kubernetes..... 37

8. Troubleshooting.....38
8.1. Collecting information for troubleshooting.....38
[CHANGED] 8.1.1. Information needed when contacting support.....38
[CHANGED] 8.1.2. Collecting logs for Storage Plug-in for Containers.....38
8.1.3. Collecting storage system information for HPE Storage.....39
[CHANGED] 8.2. Viewing the volume properties of PersistentVolume..... 39
8.3. Notes on forcibly deleting a Pod.....40
[CHANGED] 8.4. Creating and deleting PersistentVolumeClaim simultaneously 40
8.5. Host group settings..... 40
[NEW] 8.6. Timeout errors when creating or deleting Pods..... 41

9. Websites..... 42

[NEW] 10. Conventions for capacity values.....43

11. Websites..... 44

12. Support and other resources.....45
12.1. Accessing Hewlett Packard Enterprise Support.....45
12.2. Accessing updates..... 45
12.3. Remote support.....45
12.4. Warranty information..... 46
12.5. Regulatory information..... 46
12.6. Documentation feedback.....47

1. Hewlett Packard Enterprise Development LP

Acknowledgments

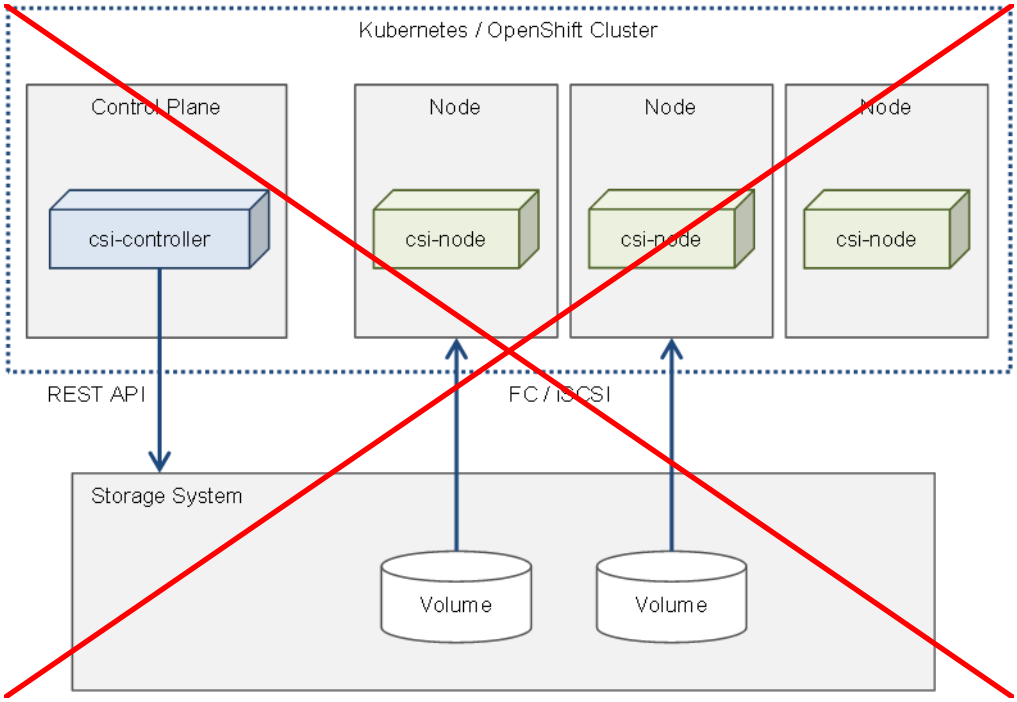
2. Overview

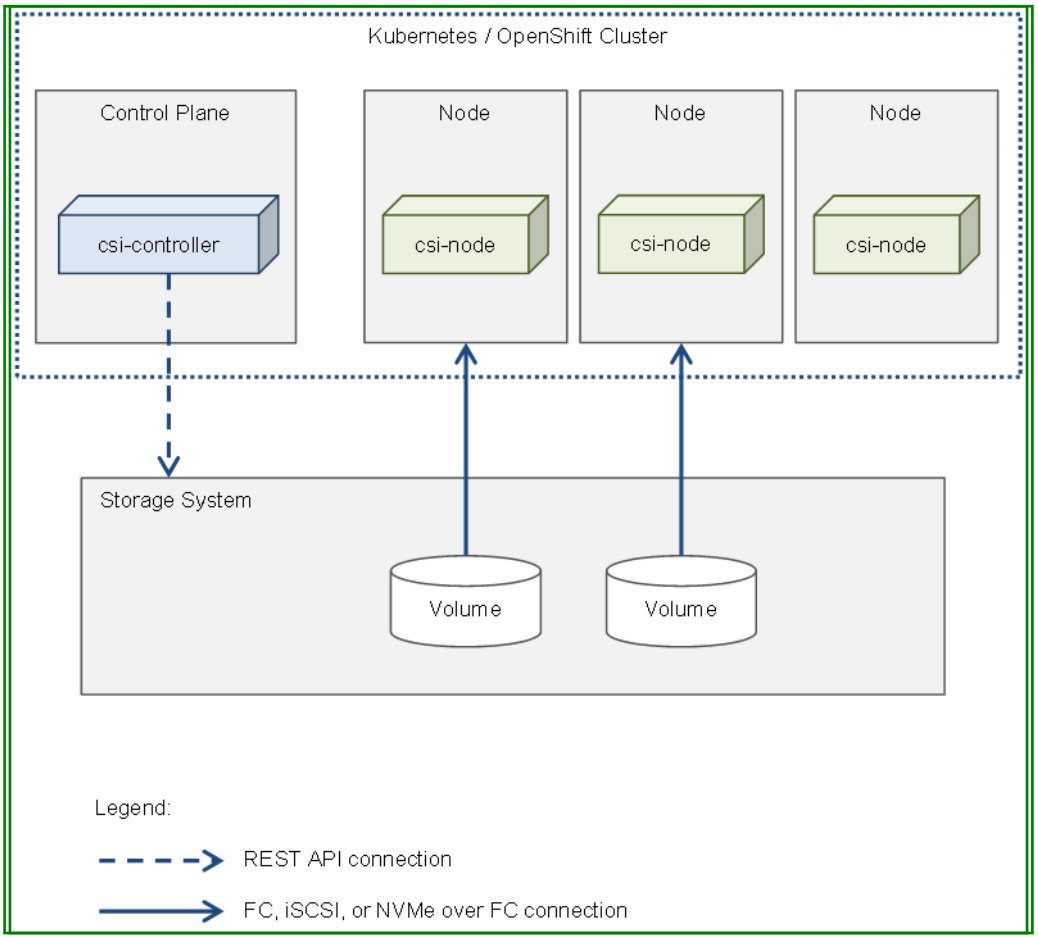
Storage Plug-in for Containers is software for creating and managing persistent volumes for HPE XP storage systems in a Kubernetes environment. By using Storage Plug-in for Containers, you can use HPE XP storage volumes from stateful applications running on Kubernetes. For persistent volumes, in addition to creation and deletion, operations such as snapshots and clones are supported.

2.1. About HPE XP Storage Plug-in for Containers

Storage Plug-in for Containers integrates Kubernetes or OpenShift with HPE storage using Container Storage Interface (CSI).

The following diagram illustrates a container environment where Storage Plug-in for Containers is deployed.





The following table lists and describes the components of Storage Plug-in for Containers.

Component	Purpose
csi-controller	Implements the CSI controller service, which mainly uses the REST API for storage operations. This is deployed as Deployment and starts on a control plane. If the csi-controller cannot start on a control plane, it might start on a node.
csi-node	Implements the CSI node service, which primarily manages volumes on each node. This is deployed as DaemonSet, and all nodes must have this component.
HPE Storage	Provides storage volumes for the containers.

2.2. About the environment setup tasks

Storage Plug-in for Containers enables dynamic operation of storage systems when containers are used. In order to use Storage Plug-in for Containers pre-installation tasks must be completed.

1. Check and apply the requirements for the server (where you plan to install Storage Plug-in for Containers, Kubernetes, and OpenShift), HPE storage, Kubernetes, and OpenShift.
2. Execute pre-installation tasks.
   a. Set up the Kubernetes and OpenShift environment.
   b. Configure the HPE storage.
3. Install Storage Plug-in for Containers.

2.3. Requirements

Before you install Storage Plug-in for Containers, check that the system requirements meet the following minimum requirements.

2.3.1. Container orchestrators to be supported

Container orchestrator	Remarks
~~Red Hat OpenShift Container Platform~~ Red Hat OpenShift Container Platform	~~—~~ —
~~Kubernetes~~ Kubernetes	~~—~~ —
~~Rancher Kubernetes Engine 2 (RKE2)~~ Rancher Kubernetes Engine 2 (RKE2)	~~If you use RKE2, in this manual, read "Kubernetes" as "RKE2" and act accordingly.~~ If you use RKE2, in this manual, read "Kubernetes" as "RKE2" and act accordingly.

For details on supported versions, see the Release Notes.

2.3.2. Server requirements

Component	Requirement
CPU	x86_64
Operating system	Refer to the release notes for ~~details.~~ details.  ~~You can also use Red Hat Enterprise Linux CoreOS as a worker node of OpenShift environments.~~
Interface	Fibre Channel, iSCSI, and NVMe over FC for bare metal servers iSCSI for virtual machines

2.3.3. Storage requirements

Storage requirements for HPE Storage.

Component	Requirement
Model and Microcode version	Refer to the release notes for details.
Interface	Fibre Channel~~Fibre Channel,~~ , iSCSI, and NVMe over FC ~~for bare metal servers.~~ ~~iSCSI for vSphere virtual machines.~~
Host group	Must be dedicated to Storage Plug-in for Containers. Do not use a host group used for Storage Plug-in for Containers for purposes other than Storage Plug-in for Containers.
User account	The built-in Storage administrator (View & Modify) user group. If you are using a customized user group, make sure it has the same roles as the built-in Storage Administrator (View & Modify) user group.
License	The following licenses are required: • ~~Thin Provisioning~~ Thin Provisioning (THP) • Fast Snap (FS)
SVP	Single and dual SVP configurations are supported.

2.3.4. Network requirements

The network requirements for Storage Plug-in for Containers are as follows:

- Storage Plug-in for Containers uses the following ports. Use this information for reference when configuring the firewall.

Component	Port	Usage	Remarks
Storage	80 or 443	REST API connection	None

- Storage Plug-in for Containers does not support IPv6. Use IPv4.

2.4. Pre-installation tasks

Before you install Storage Plug-in for Containers, review and apply the server and storage pre-installation requirements.

2.4.1. Server pre-installation

The following table outlines the pre-installation tasks for each server component.

Component	Tasks
Hypervisor	If you want to use virtual machines, set up the hypervisor.  Storage Plug-in for Containers is tested with VMware vSphere 7.0.
Fibre Channel	Verify that HBA is installed on nodes that implement a Fibre Channel connection with the storage system.
iSCSI	Verify that iSCSI initiator software is installed on nodes that implement an iSCSI connection with the storage system. If the software is not installed, refer to: https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/7/html/storage_administration_guide/osm-create-iscsi-initiator  Storage Plug-in for Containers does not support IQNs that include uppercase alphabetic characters.

Component	Tasks
NVMe over FC	~~NVMe over FC connections are supported by the following OSs. Check the following before specifying settings:~~ NVMe over FC connections are supported by the following OSs. Check the following before specifying settings: • Ubuntu 20.04, 22.04 • Red Hat Enterprise Linux 9 • Red Hat Enterprise Linux CoreOS ~~(OpenShift 4.13 or later)~~ (OpenShift) Specify the following settings for a node that connects with the storage system through an NVMe over FC connection. The method for specifying settings differs depending on the OS. • Ubuntu Using the following command, install the `nvme-cli` tool: `apt-get install nvme-cli` • Red Hat Enterprise Linux • Broadcom (Emulex) Using the following website as reference, install the `nvme-cli` tool: https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/9/html/managing_storage_devices/configuring-nvme-over-fabrics-using-nvme-fc_managing-storage-devices • QLogic Using the following website as reference, install the `nvme-cli` tool and then reload the QLogic module (qla2xxx): https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/9/html/managing_storage_devices/configuring-nvme-over-fabrics-using-nvme-fc_managing-storage-devices • Red Hat Enterprise Linux CoreOS You do not need to install the `nvme-cli` tool. ~~Do not change a host NQN while the host is running. However, if a change while the host is running is necessary, perform the drain operation on the relevant node, and then change the host NQN. After changing the host NQN, restart the host.~~ • Do not change a host NQN while the host is running. However, if a change while the host is running is necessary, perform the drain operation on the relevant node, and then change the host NQN. After changing the host NQN, restart the host. • When there are multiple nodes, make sure that no duplicate host NQN exists.
Multipath function	~~For Fibre Channel and iSCSI, use Device Mapper Multipath. For NVMe over FC , use Native NVMe Multipath.~~ For Fibre Channel and iSCSI, use Device Mapper Multipath. For NVMe over FC, use Native NVMe Multipath. For details on ~~multipath function settings, see~~ multipath function settings, see Device Mapper Multipath settings ~~or Native NVMe Multipath settings~~ or Native NVMe Multipath settings.

2.4.1.1. Device Mapper Multipath settings

Enable Device Mapper Multipath and make sure that the `user_friendly_names` option is set to `yes`.

For example:

```
defaults {
        user_friendly_names yes
        find_multipaths yes
}
blacklist {
}
```

The setting values might differ depending on the environment. Also see the documentation for your OS.

- Red Hat Enterprise Linux 7: https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/7/html/dm_multipath/mpio_setup
- Red Hat Enterprise Linux 8: https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/8/html/configuring_device_mapper_multipath/configuring-dm-multipath_configuring-device-mapper-multipath
- Ubuntu: https://ubuntu.com/server/docs/device-mapper-multipathing-introduction

For OpenShift, you will need to use the MachineConfig YAML file. For details, see the official documentation: https://docs.openshift.com/container-platform/latest/machine_configuration/index.html

~~The~~ For OpenShift, the following is an ~~example:~~ example of the procedure:

1. Obtain `multipath-machineconfig-sample.yaml` from the provided sample files.
2. If necessary, change the multipath settings in `multipath-machineconfig-sample.yaml`.The following default multipath settings are specified in `multipath-sample.conf`. The character string obtained by encoding this file in base64 is specified in `multipath-machineconfig-sample.yaml`.

```
defaults {
user_friendly_names yes
find_multipaths yes
}
blacklist {
}
```

   a. Obtain `multipath-sample.conf` from the provided sample files.
   b. Edit `multipath-sample.conf` to change the multipath settings.
   c. Run the following command to obtain `multipath-sample.conf` encoded in base64:

```
# cat multipath-sample.conf | base64 -w0
```

   d. Change the `spec.config.storage.files.contents.source` setting in `multipath-machineconfig-sample.yaml`.
   The character string specified for `spec.config.storage.files.contents.source` corresponds to the multipath settings encoded in base64. Replace this character string with the base64-encoded character string obtained from `multipath-sample.conf`.

```
apiVersion: machineconfiguration.openshift.io/v1
kind: MachineConfig
metadata:
  name: multipath-machineconfig-sample
  labels:
    machineconfiguration.openshift.io/role: worker
spec:
  config:
    ignition:
      version: 3.2.0
    storage:
      files:
      - contents:
          source: data:text/
plain;charset=utf-8;base64,ZGVmYXVsdHMgewp1c2VyX2ZyaWVuZGx5X25hbWVzIHllcwpmaW5kX211b
HRpcGF0aHMgeWVzCn0KYmxhY2tsaXN0IHsKfQo=
          verification: {}
        filesystem: root
        mode: 400
        path: /etc/multipath.conf
```

3. Run the following command:

```
# oc apply -f multipath-machineconfig-sample.yaml
```

MachineConfig applies to compute nodes only. After MachineConfig is created, all compute nodes are automatically restarted one by ~~one.~~ one, and `/etc/multipath.conf` is created on all compute nodes.

4. ~~`/etc/multipath.conf` is created on all compute nodes. On each compute node, open~~ On each compute node, open `/etc/multipath.conf` and verify that the settings have been applied.

It might take time for the settings to be applied.

2.4.1.2. Native NVMe Multipath settings

The method for specifying Native NVMe Multipath settings differs depending on the OS.

Ubuntu

By default, Native NVMe Multipath is enabled. ~~For this reason, there is no need to specify Native NVMe Multipath settings.~~ If Native NVMe Multipath is disabled, enable it.

Red Hat Enterprise Linux

Using the following website as reference, enable Native NVMe Multipath:

https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/9/html/managing_storage_devices/enabling-multipathing-on-nvme-devices_managing-storage-devices

Red Hat Enterprise Linux CoreOS

By default, Native NVMe Multipath is enabled. ~~For this reason, there is no need to specify Native NVMe Multipath settings.~~ If Native NVMe Multipath is disabled, enable it.

2.4.2. Storage pre-installation for HPE Storage

The following table outlines the pre-installation tasks to be completed for each storage component.

Component	Task
Program products	• Enable Thin Provisioning (THP) license • Enable Fast Snap(FS) license
Pool	Create aTHP pool. Smart Tiers is not supported.
Fibre Channel connection	Use a Fibre Channel switch for communication between the storage and servers. Set the following parameters for storage ports using Remote Web Console: • Connection Type: P-to-P • Fabric: ON • Security: Enabled Storage Plug-in for Containers automatically performs the following actions: • Creates host groups for each host if there is no host group. If you want to use existing host groups, rename them according to the naming rule (see Host group and iSCSI target naming rules). • Adds the ~~WWN~~ WWNs for all of the HBA ports in each host to the host group created for each host. Storage Plug-in for Containers will overwrite host mode options even if existing host groups have other host mode options.

Component	Task
iSCSI connection	Enable port security by Remote Web Console. Storage Plug-in for Containers automatically performs the following actions: • Creates iSCSI targets for each host if there is no iSCSI target. If you want to use existing iSCSI targets, rename them according to the naming rule (see Host group and iSCSI target naming rules). • Adds the IQN to the iSCSI target corresponding to each host that will join the Kubernetes cluster. • Logs in to the iSCSI target on each host. If you want to use CHAP, do the following: • Create an iSCSI target (see Host group and iSCSI target naming rules). • Set CHAP for the port and iSCSI target. • Log in to the iSCSI target with CHAP authentication. Run login from each host.  Storage Plug-in for Containers will overwrite host mode options even if existing iSCSI targets have other host mode options.
NVMe over FC connection	Create an NVM subsystem to manage the path between the host and storage system. The NVM subsystem must be dedicated to Storage Plug-in for Containers. Do not use the NVM subsystem used for Storage Plug-in for Containers for any purpose other than Storage Plug-in for Containers. 1. Create the NVM subsystem for the storage system. Enable the namespace security, and specify Linux as the host mode. 2. Set the operation mode of the Fibre Channel port to NVMe mode. 3. Disable LUN security on the Fibre Channel port. 4. Set an NVM subsystem port. For details about each of the above steps, see the ~~*Provisioning for Open Systems User Guide*~~ *Provisioning for Open Systems User Guide*.

2.4.2.1. Host group and iSCSI target naming rules

Storage Plug-in for Containers automatically searches host groups and iSCSI targets based on the name.

If you want to use an already existing host group or iSCSI target, refer to either the naming rule of host groups or iSCSI targets depending on your storage connection:

Naming rule of host groups

Storage Plug-in for Containers searches host groups by the naming rule. If Storage Plug-in for Containers cannot find any host group in the port, it automatically creates the host group. If you already have host groups, you need to delete them or rename them according to the following naming rule:

"spc-<*wwn1*>-<*wwn2*>-<*wwn3*>"

Naming rule details:

- <*wwn1*>, <*wwn2*>, <*wwn3*> are the world wide name of each host.
- <*wwn1*>, <*wwn2*>, <*wwn3*> are sorted by name.
- If the host has more than three WWNs, Storage Plug-in for Containers sorts <*wwn1*>, <*wwn2*> ... <*wwnN*> and uses lower three names.
- If the host has only one or two WWNs, the names are "spc-<*wwn1*>" or "spc-<*wwn1*>-<*wwn2*>".

Naming rule of iSCSI targets

Storage Plug-in for Containers searches iSCSI targets by the naming rule. If Storage Plug-in for Containers cannot find any iSCSI target, it automatically creates the iSCSI target, "spc-<~~*hashed-IQN*~~ *hashed-IQN* >". If you already have iSCSI targets, you need to delete them or rename them according to the following naming rule: "spc-<~~*any-string*~~ *any-string* >"

3. Installation

This chapter describes how to install Storage Plug-in for Containers. The installation method depends on whether your environment is OpenShift or Kubernetes.

3.1. Installation on OpenShift

Storage Plug-in for Containers is easily deployed to OpenShift using the Operator, which can be installed from OperatorHub. To install Storage Plug-in for Containers, follow the steps below.

- If there is a previous version of Storage Plug-in for Containers, remove it before performing the installation procedure.
- If you want to install Storage Plug-in for Containers in an OpenShift Container Platform environment that does not have access to the internet, mirror the certified-operators catalog in advance. For details on the procedure, see ~~https://docs.openshift.com/container-platform/latest/installing/disconnected_install/installing-mirroring-installation-images.html#olm-mirror-catalog_installing-mirroring-installation-images~~https://docs.openshift.com/container-platform/latest/disconnected/mirroring/installing-mirroring-installation-images.html#olm-mirror-catalog_installing-mirroring-installation-images.
  For example, for OpenShift Container Platform version 4.10, the index image of the certified-operators catalog is registry.redhat.io/redhat/certified-operator-index:v4.10. For details, see https://docs.openshift.com/container-platform/latest/operators/understanding/olm-rh-catalogs.html.

1. Access OperatorHub from the OpenShift web console.
2. Search HPE XPStorage Plug-in for Containers and install the Operator.

Select the following settings in Operator Subscription:

- Installation mode: Select **A specific namespace on the cluster**and ~~<~~ specify any namespace. ~~*any namespace >*~~
- Update approval: Select **Manual**and approve the Install ~~Plan (see~~ Plan (seehttps://docs.openshift.com/).

3. Confirm the status of the Operator is **Succeeded**.
4. Confirm the status of the Operator Pod is **Running**.
5. Click **Create Instance** on the Operator Details.
6. Click **Create**. If you want to make an advanced configuration, refer to Configuration of Storage Plug-in for Containers instance.
7. Confirm the status READY is **true** using the following command:

```
# oc get ~~xspc -n <~~ xspc -n <~~namespace for xspc~~ Storage-Plug-in-for-Containers-namespace > NAME READY AGE ~~xspc~~ xspc    true    30s
```

3.2. Installation on Kubernetes

For Kubernetes, you can install Storage Plug-in for Containers using Operator. To install Storage Plug-in for Containers, perform the following procedure.

If there is a previous version of Storage Plug-in for Containers, remove it before performing the installation procedure.

1. Extract the Storage Plug-in for Containers package and move to the directory **yaml/operator**.
2. Create the namespace for the Operator:

```
# kubectl create -f xspc-operator-namespace.yaml
```

3. Create a Secret for Red Hat registry ~~and name the Secret~~ that will be required to create the Operator, and name the Secret `regcred-redhat-com`. (For details, see https://kubernetes.io/docs/tasks/configure-pod-container/pull-image-private-registry/#create-a-secret-by-providing-credentials-on-the-command-line.) For example:

```
# kubectl create secret docker-registry regcred-redhat-com \
  --namespace=xspc-operator-system \
  --docker-server=registry.connect.redhat.com \
  --docker-username=<user> \
  --docker-password=<password>
```

4. Create the Operator and confirm the Operator is running:

```
# kubectl create -f xspc-operator.yaml

# kubectl get deployment -n xspc-operator-system
NAME                                READY   UP-TO-DATE   AVAILABLE   AGE
xspc-operator-controller-manager    1/1     1            1           14s
```

5. Create ~~two Secrets for Red Hat registry and name the Secrets~~ `regcred-redhat-com` ~~and~~ ~~`regcred-redhat-io`~~ ~~.~~ a Secret for Red Hat registry that will be required to deploy Storage Plug-in for Containers, and name the Secret `regcred-redhat-com`. ~~For example:~~ For example:

```
# SPC_NAMESPACE=<any-namespace any-namespace >

# kubectl create secret docker-registry regcred-redhat-com \
--namespace=${SPC_NAMESPACE} \
--docker-server=registry.connect.redhat.com \
--docker-username=<user> \ --docker-password= password
>

# kubectl create secret docker-registry regcred-redhat-io \
--namespace=${SPC_NAMESPACE} \
--docker-server=registry.redhat.io \
--docker-username=<
user
> \
--docker-password=<
password
>

<password>
```

6. ~~Modify the namespace if you specified a namespace other than kube-system for~~ Modify the namespace if you specified a namespace other than kube-system for <~~SPC_NAMESPACE~~ *SPC_NAMESPACE-value* ~~:~~ >:

```
apiVersion: csi.hpe.com/v1
kind: XSPC
metadata:
  name: xspc
  namespace: <SPC_NAMESPACE SPC_NAMESPACE-value >
spec:
  imagePullSecrets:
  - regcred-redhat-com regcred-redhat- com
  -
regcred-redhat-io
```

~~If you want to make an advanced configuration, refer to~~ If you want to make an advanced configuration, refer to Configuration of Storage Plug-in for Containers instance.

7. Deploy Storage Plug-in for Containers and confirm the status READY is **true** using the following command:

```
# kubectl create -f xspc_v1_xspc.yaml

# kubectl get xspc -n ${SPC_NAMESPACE}
NAME READY AGE
xspc true 30s
```

3.3. Configuration of Storage Plug-in for Containers instance

You can configure Storage Plug-in for Containers by editing the `CustomResource` YAML file, which includes the following parameters:

Parameter	Description
spec.imagePullSecrets	Specify this parameter if a Secret is required to pull an image.
spec.controller.containers.name	Name of the Storage Plug-in for Containers that you want to configure in xspc-csi-controller pods. For example, xspc-csi-driver, csi-provisioner, and so on are the key to the container name inside the xspc-csi-controller. The ~~kubectl describe deployment xspc-csi-controller -n < SPC_NAMESPACE ></~~ kubectl describe deployment xspc-csi-controller -n <SPC_NAMESPACE-value> command is used to get the container names.
spec.controller.containers.image	The image name of xspc-csi-controller
spec.controller.containers.imagePullPolicy	The image pull policy of xspc-csi-controller. The default value is IfNotPresent.
spec.controller.containers.env	List of environment variables to set in xspc-csi-controller container. Refer to Environment variables.
spec.controller.containers.args	Arguments to the entry point for xspc-csi-controller. This replaces all parameters at spec.template.spec.containers.args in a deployment of the container xspc-csi-controller.
spec.controller.tolerations	Specify the toleration of the Pod that runs xspc-csi-controller. The same format as Kubernetes is to be used. For details, see https://kubernetes.io/docs/concepts/scheduling-eviction/taint-and-toleration/.
spec.controller.affinity.nodeAffinity	Specify the node affinity of the Pod that runs xspc-csi-controller. The same format as Kubernetes is to be used. For details, see https://kubernetes.io/docs/tasks/configure-pod-container/assign-pods-nodes-using-node-affinity/.
spec.node.containers.name	Name of the container that you want to configure in xspc-csi-node pods. For example, xspc-csi-driver, liveness-probe, and so on are the key to the container name inside xspc-csi-node. The ~~kubectl describe daemonset xspc-csi-node -n < SPC_NAMESPACE ></~~ kubectl describe daemonset xspc-csi-node -n <SPC_NAMESPACE-value> command is used to get the container names.
spec.node.containers.image	The image name of xspc-csi-node
spec.node.containers.imagePullPolicy	The image pull policy of xspc-csi-node. The default value is IfNotPresent.
spec.node.containers.env	List of environment variables to set in xspc-csi-node container.
spec.node.containers.args	Arguments to the entry point for xspc-csi-node. This replaces all parameters at spec.template.spec.containers.args in a deployment of the container xspc-csi-node.
spec.node.tolerations	Specify the toleration of the Pod that runs xspc-csi-node. The same format as Kubernetes is to be used. For details, see https://kubernetes.io/docs/concepts/scheduling-eviction/taint-and-toleration/.
spec.node.affinity.nodeAffinity	Specify the node affinity of the Pod that runs xspc-csi-node. The same format as Kubernetes is to be used. For details, see https://kubernetes.io/docs/tasks/configure-pod-container/assign-pods-nodes-using-node-affinity/.

3.4. Environment variables

The following is the environment variable of xspc-csi-driver on xpsc-csi-controller:

Environment variable name	Description
SPC_VERIFY_CERTIFICATE	If true, the TLS certificate of the storage is checked in the HTTPS connection (Default: false).
TZ	Timezone for logging. For example, Asia/Tokyo (Default: UTC).

The following is an example to enable certificate verification of the xspc-csi-driver.

1. Check the current settings using the following command:

```
# kubectl get deployment -n <SPC_NAMESPACE-value> xspc-csi-controller -o yaml
<…>
      - name: xspc-csi-driver
        env:
        - name: CSI_ENDPOINT
          value: unix:///csi/csi-controller.sock
        - name: KUBE_NODE_NAME
          valueFrom:
            fieldRef:
              apiVersion: v1
              fieldPath: spec.nodeName
<…>
```

2. Add env: SPC_VERIFY_CERTIFICATE to Storage Plug-in for Containers manifests.

```
apiVersion: csi.hpe.com/v1
kind: XSPC
metadata:
  name: xspc
  namespace: <SPC_NAMESPACE-value>
spec:
  controller:
    containers:
      - name: xspc-csi-driver
        env:
        - name: SPC_VERIFY_CERTIFICATE
          value: "true"
```

3. Uninstall and reinstall Storage Plug-in for Containers. For more information on how to uninstall and reinstall Storage Plug-in for Containers, see Installation and Uninstallation.
4. Check the changes.

```
# kubectl get deployment -n <SPC_NAMESPACE-value> xspc-csi-controller -o yaml
<…>
    - name: xspc-csi-driver
      env:
    - name: CSI_ENDPOINT
        value: unix:///csi/csi-controller.sock
      - name: KUBE_NODE_NAME
        valueFrom:
          fieldRef:
            apiVersion: v1
            fieldPath: spec.nodeName
      - name: SPC_VERIFY_CERTIFICATE
        value: "true"
<…>
```

4. Usage

This chapter describes the settings and command examples for each component used in Storage Plug-in for Containers.

4.1. Secret settings

The Secret file contains the storage URL, user name, and password settings that are necessary for Storage Plug-in for Containers to work with your environment. The following sample provides information about the required parameters.

Parameter references for secret-sample.yaml

```
apiVersion: v1
kind: Secret
metadata:
  name: secret-sample              #(1)
type: Opaque
data:
  url: aHR0cDovLzE3Mi4xNi4xLjE=    #(2)
  user: VXNlcjAx                   #(3)
  password: UGFzc3dvcmQwMQ==       #(4)
```

Legend:

(1) Secret name

(2) base64-encoded storage URL

Use the IP address of the ~~SVP for the following:~~ SVP. ~~XP8 and XP7 .~~

Example:

```
echo -n "http://172.16.1.1" | base64
```

(3) base64-encoded storage user name.

Example:

```
echo -n "User01" | base64
```

(4) base64-encoded storage password.

Example:

```
echo -n "Password01" | base64
```

4.2. StorageClass settings

The StorageClass file contains storage settings that are necessary for Storage Plug-in for Containers to work with your environment. The following sample provides information about the required parameters.

After creating a StorageClass and PVC, re-creating StorageClass will not affect the existing PVCs.

StorageClass for HPE Storage

~~Parameter references for sc-sample.yaml~~

Parameter references for sc-sample.yaml

```
~~apiVersion: storage.k8s.io/v1~~
~~kind: StorageClass~~
~~metadata:~~
```

```
  name:
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: sc-sample                          #(1)
  annotations:
    kubernetes.io/description: HPE XP Storage Plug-in for Containers
provisioner: xspc.csi.hpe.com
reclaimPolicy: Delete
volumeBindingMode: Immediate
allowVolumeExpansion: true
parameters:
  serialNumber: "54321"                    #(2)
  poolID: "1"                              #(3)
  portID : CL1-A,CL2-A                     #(4)
  connectionType: fc                       #(5)
  storageEfficiency: "CompressionDeduplication"      #(6)
  storageEfficiencyMode: "PostProcess"               #(7)
  csi.storage.k8s.io/fstype: ext4                    #(8)
  csi.storage.k8s.io/node-publish-secret-name: "secret-sample"        #(9)
  csi.storage.k8s.io/node-publish-secret-namespace: "default"         #(10)
  csi.storage.k8s.io/provisioner-secret-name: "secret-sample"         #(9)
  csi.storage.k8s.io/provisioner-secret-namespace: "default"          #(10)
  csi.storage.k8s.io/controller-publish-secret-name: "secret-sample"  #(9)
  csi.storage.k8s.io/controller-publish-secret-namespace: "default"   #(10)
  csi.storage.k8s.io/node-stage-secret-name: "secret-sample"          #(9)
  csi.storage.k8s.io/node-stage-secret-namespace: "default"           #(10)
  csi.storage.k8s.io/controller-expand-secret-name: "secret-sample"   #(9)
  csi.storage.k8s.io/controller-expand-secret-namespace: "default"    #(10)
```

Legend:

(1) StorageClass name

(2) Storage serial number

(3) THP pool ID

(4) Port ID. Use a comma separator for multipath. ~~If an NVMe over FC connection is used, this specification is unnecessary.~~ If an NVMe over FC connection is used, this specification is unnecessary.

(5) Connection type between storage and nodes. `fc`, `iscsi`, and `nvme-fc` are supported. If `connectionType` is not specified, `fc` is set.

If an NVMe over FC connection is used, add `nvmSubsystemID`, and then specify a value.

(6) Activation of adaptive data reduction. `"Compression"`, `"CompressionDeduplication"`, and `"Disabled"` are supported. The default is `"Disabled"`, and If `"Disabled"` is specified, adaptive data reduction is disabled. For a storage system where the compression accelerator module is installed, if you specify `"Compression"` or `"CompressionDeduplication"` for `storageEfficiency`, the compression function using the compression accelerator module is automatically activated.

(7) Execution mode of adaptive data reduction. You can specify this parameter when `storageEfficiency` is `"Compression"` or `"CompressionDeduplication"`, and `"Inline"` and `"PostProcess"` are supported for the parameter. If `storageEfficiencyMode` is not specified, adaptive data reduction runs in the default execution mode, which depends on the storage system model. For details on the parameter, see the description of adaptive data reduction in the *Provisioning for Open Systems User Guide*.

If the LDEV was created with Storage Plug-in for Containers, do not change the parameters related to adaptive data reduction.

(8) Filesystem type. `ext4` and `xfs` are supported. If `csi.storage.k8s.io/fstype` is not specified, `ext4` is set.

(9) Secret name

(10) Secret namespace

4.3. PersistentVolumeClaim settings

In this section, you will configure PersistentVolumeClaim settings, which are required by Storage Plug-in for Containers to dynamically create a new volume for a storage system.

The PersistentVolumeClaim file contains volume information that is used by Storage Plug-in for Containers to create PersistentVolumes. The following sample provides information about the required parameters.

~~If you use the procedure described in Creating a PV to create a static PV, perform the procedure described in Creating a PVC to complete the association with a PVC, and then specify these settings. If you specify these settings before completing the association, no PV will be dynamically created by Storage Plug-in for Containers , and a PVC might be associated with the static PV whose association is not complete. Complete the association or delete the PV if it is unnecessary.~~

- If you want to use the existing volume of the storage system as PersistentVolumeClaim, see Static provisioning.
- If you will use PersistentVolumeClaim to be configured in this section and the static provisioning function at the same time, a static PV created by following the procedure described in Creating a PV must be properly associated with a PVC by performing the procedure described in Creating a PVC. If you have not performed the procedure described in Creating a PVC, perform the following procedure before configuring the PersistentVolumeClaim settings in this section.
  a. Check PVs for which association has not been completed.

```
kubectl get pv
```

  PVs whose `STATUS` is `Available` have not been associated.
  b. For PVs for which association has not been completed, check whether `claimRef` is specified.

```
kubectl get <PV-name> -o yaml
```

  c. If there are any PVs for which `claimRef` is not specified, perform either of the following procedures for each PV.
    - Re-create the static PV, specify `claimRef`, and then perform the procedure described in Creating a PVC.
    - If you do not need the PV, delete it.

~~To check~~ If you configure PersistentVolumeClaim settings as described in this section when there is a PV for which ~~PVs have not been associated, run the `kubectl get pv claimRef` command. PVs whose STATUS~~ is ~~Available have~~ not ~~been associated.~~ specified, no PV will be dynamically created by Storage Plug-in for Containers, and a PVC might be associated with the static PV whose association is not complete.

Parameter references for pvc-sample.yaml

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: pvc-sample                #(1)
spec:
  accessModes:
  - ReadWriteOnce                 #(2)
  resources:
    requests:
```

```
      storage: 1Gi                #(3)
  storageClassName: sc-sample     #(4)
```

Legend:

(1) PersistentVolumeClaim name

(2) Specify `ReadWriteOnce` or `ReadOnlyMany`. To use ReadOnlyMany, see ReadOnlyMany.

(3) Volume size

(4) StorageClass name

Usage restrictions for a PersistentVolumeClaim

- If a failure occurs when creating a PersistentVolumeClaim, a PersistentVolumeClaim object will be created without the PersistentVolume. In this case, delete the PersistentVolumeClaim object using the `kubectl delete pvc <~~PVC_NAME~~ PVC-name >` command.
- If a failure occurs when deleting a PersistentVolumeClaim, a PersistentVolumeClaim object will be deleted but the PersistentVolume object will remain and any storage asset associated with the PersistentVolume object may also remain. In this case, see Viewing the volume properties of PersistentVolume and obtain the volume ID of the storage. Delete the PersistentVolume using the `kubectl delete pv <~~PV_NAME~~ PV-name >` command. Also, delete the storage asset (LDEV). For details, see the user guide for the storage system in your environment.

4.4. Pod settings

The Pod file contains volume information. Storage Plug-in for Containers mount volumes based on this information.

Parameter references for pod-sample.yaml

```
apiVersion: v1
kind: Pod
metadata:
  name: pod-sample            #(1)
spec:
  containers:
    - name: my-busybox
      image: busybox
      volumeMounts:
      - mountPath: "/data"    #(2)
        name: sample-volume
      command: ["sleep", "1000000"]
      imagePullPolicy: IfNotPresent
  volumes:
    - name: sample-volume
      persistentVolumeClaim:
        claimName: pvc-sample  #(3)
```

Legend:

(1) Pod name

(2) Path (path where the volume is mounted inside a container)

(3) PersistentVolumeClaim name

4.5. Command examples

Following are examples of creating and deleting a Secret, StorageClass, PersistentVolumeClaim, and Pod using commands in practice.

If your environment is OpenShift, replace Kubernetes Command Line Interface (CLI) with OpenShift CLI. For more information about OpenShift CLI, refer to the OpenShift CLI reference.

Create a Secret, StorageClass, PersistentVolumeClaim, and Pod

```
# kubectl create -f secret-sample.yaml
secret/secret-sample created

# kubectl get secret
NAME            TYPE     DATA   AGE
secret-sample   Opaque   3      34s

# kubectl create -f sc-sample.yaml
storageclass.storage.k8s.io/sc-sample created

# kubectl get sc
NAME        PROVISIONER            AGE
sc-sample   xspc.csi.hpe.com   21s

# kubectl create -f pvc-sample.yaml
persistentvolumeclaim/pvc-sample created

# kubectl get pvc
NAME         STATUS   VOLUME                                     CAPACITY   ACCESS
MODES   STORAGECLASS   AGE
pvc-sample   Bound    pvc-cf8c6089-0386-4c39-8037-e1520a986a7d   1Gi
RWO            sc-sample      28s

# kubectl create -f pod-sample.yaml
pod/pod-sample created

# kubectl get pod
NAME         READY   STATUS    RESTARTS   AGE
pod-sample   1/1     Running   0          20s
```

If the LDEV was created with Storage Plug-in for Containers, do not change the nickname.

Confirm a PersistentVolume information created by Storage Plug-in for Containers

```
# kubectl get pv
NAME                                       CAPACITY   ACCESS MODES   RECLAIM
POLICY   STATUS   CLAIM                 STORAGECLASS   REASON   AGE
pvc-0b76d44e-b584-4367-8fdf-1cb2a55aa9a0   1Gi        RWO
Delete           Bound    default/pvc-sample   sc-sample               5m7s

# kubectl describe pv pvc-0b76d44e-b584-4367-8fdf-1cb2a55aa9a0
Name:            pvc-0b76d44e-b584-4367-8fdf-1cb2a55aa9a0
Labels:          <none>
Annotations:     pv.kubernetes.io/provisioned-by: xspc.csi.hpe.com
                 volume.kubernetes.io/provisioner-deletion-secret-name: secret-
sample
                 volume.kubernetes.io/provisioner-deletion-secret-namespace: default
Finalizers:      [kubernetes.io/pv-protection external-attacher/xspc-csi-hpe-com]
StorageClass:    sc-sample
Status:          Bound
Claim:           default/pvc-sample
Reclaim Policy:  Delete
Access Modes:    RWO
VolumeMode:      Filesystem
Capacity:        1Gi
Node Affinity:   <none>
Message:
Source:
    Type:              CSI (a Container Storage Interface (CSI) volume source)
    Driver:            xspc.csi.hpe.com
    FSType:            ext4
    VolumeHandle:      01--scsi--800000070002--50--spc-2eb52f17b2
    ReadOnly:          false
    VolumeAttributes:      connectionType=fc
```

```
                            hostModeOption=
                            ldevIDDec=50
                            ldevIDHex=00:32
                            nickname=spc-2eb52f17b2
                            ports=CL7-B,CL8-B
                            size=1Gi
                            storage.kubernetes.io/
csiProvisionerIdentity=1685677499807-8081-xspc.csi.hpe.com
Events:                 <none>
```

Delete a Secret, StorageClass, PersistentVolumeClaim, and Pod

```
# kubectl get pod
NAME         READY   STATUS    RESTARTS   AGE
pod-sample   1/1     Running   0          30s

# kubectl delete pod pod-sample
pod "pod-sample" deleted

# kubectl get pvc
NAME         STATUS   VOLUME                                     CAPACITY   ACCESS
MODES   STORAGECLASS   AGE
pvc-sample   Bound    pvc-cf8c6089-0386-4c39-8037-e1520a986a7d   1Gi
RWO            sc-sample      46s

# kubectl delete pvc pvc-sample
persistentvolumeclaim "pvc-sample" deleted

# kubectl get sc
NAME        PROVISIONER        AGE
sc-sample   xspc.csi.hpe.com   53s

# kubectl delete sc sc-sample
storageclass.storage.k8s.io "sc-sample" deleted

# kubectl get secret
NAME            TYPE     DATA   AGE
secret-sample   Opaque   3      74s

# kubectl delete secret secret-sample
secret "secret-sample" deleted
```

4.6. Volume snapshot

This feature can create a snapshot that is a point-in-time image of a volume. A snapshot can be used to duplicate a previous state of an existing volume.

- If the volume is expanded, confirm for completion before executing this feature. See Volume expansion for more details.
- Flush the data before creating a snapshot for data consistency. For example, temporarily remove the ~~pod.~~ Pod.

This feature requires the following resources:

- StorageClass
- PersistentVolumeClaim

If your environment is Kubernetes, install Snapshot CRDs and Snapshot Controller per cluster (see https://github.com/kubernetes-csi/external-snapshotter). For Snapshot CRDs, use v1.

If Snapshot Alpha or Beta CRDs are present in your environment, remove them before installing Snapshot v1 CRDs.

Parameter references for volumesnapshotclass-sample.yaml

```
apiVersion: snapshot.storage.k8s.io/v1
kind: VolumeSnapshotClass
metadata:
```

```
  name: snapshotclass-sample    #(1)
driver: xspc.csi.hpe.com
deletionPolicy: Delete
parameters:
  poolID: "1"                    #(2)
  csi.storage.k8s.io/snapshotter-secret-name: "secret-sample" #(3)
  csi.storage.k8s.io/snapshotter-secret-namespace: "default"  #(4)
```

Legend:

(1) VolumeSnapshotClass name

(2) Same poolID as the StorageClass

(3) Same Secret name as the StorageClass

(4) Same Secret namespace as the StorageClass

Parameter references for volumesnapshot-sample.yaml

```
apiVersion: snapshot.storage.k8s.io/v1
kind: VolumeSnapshot
metadata:
  name: snapshot-sample                                  #(1)
spec:
  volumeSnapshotClassName: snapshotclass-sample          #(2)
  source:
    persistentVolumeClaimName: pvc-sample                #(3)
```

Legend:

(1) VolumeSnapshot name

(2) VolumeSnapshotClass name

(3) PersistentVolumeClaim name from which the snapshot is obtained

Parameter references for pvc-from-snapshot-sample.yaml

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: pvc-from-snapshot-sample    #(1)
spec:
  dataSource:
    name: snapshot-sample           #(2)
    kind: VolumeSnapshot
    apiGroup: snapshot.storage.k8s.io
  accessModes:
  - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi                  #(3)
  storageClassName: sc-sample       #(4)
```

Legend:

(1) PersistentVolumeClaim name

(2) VolumeSnapshot name

(3) Specify the size of the source volume. Obtain the size by using the `kubectl get pv <~~PV_NAME~~ PV-name> -o yaml` command, which is displayed in **size**.

If the volume is expanded or it is a statically provisioned PersistentVolume, obtain the size by using the `kubectl get pv <~~PV_NAME~~ PV-name>` command. The size is displayed in **CAPACITY**.

(4) Specify the same StorageClass name as the one used for dataSource.

Command examples

- Create a VolumeSnapshotClass:

```
# kubectl create -f volumesnapshotclass-sample.yaml
```

- Create a VolumeSnapshot:

```
# kubectl create -f volumesnapshot-sample.yaml
```

- Verify that readyToUse is true. If it is true, the creation of VolumeSnapshot is complete.

```
# kubectl get volumesnapshot -o yaml
```

If readyToUse is false, confirm the cause and solution by following the steps:

a. Obtain the `boundVolumeSnapshotContentName` by using the command: kubectl get volumesnapshot -o yaml
b. Confirm the error message by using the command: **`kubectl describe volumesnapshotcontent <VolumeSnapshotContentName>`**

- Create a PersistentVolumeClaim from a snapshot:

```
# kubectl create -f pvc-from-snapshot-sample.yaml
```

4.7. Volume cloning

This feature can create a duplicate as a clone of an existing volume. A clone can be consumed in the same way as any standard volume.

- If the volume is expanded, confirm for completion before executing this feature. Refer to Volume expansion for details.
- Flush the data before cloning for data consistency. For example, temporarily remove the ~~pod.~~ Pod.

This feature requires the following resources:

- StorageClass
- PersistentVolumeClaim

Parameter references for pvc-from-pvc-sample.yaml

This YAML file is a manifest file for creating a clone from an existing volume "pvc-sample".

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: pvc-from-pvc-sample      #(1)
spec:
  dataSource:
    name: pvc-sample             #(2)
    kind: PersistentVolumeClaim
    apiGroup: ""
  accessModes:
  - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi               #(3)
  storageClassName: sc-sample    #(4)
```

Legend:

(1) PersistentVolumeClaim name of clone

(2) PersistentVolumeClaim name of source

(3) Specify the size of the source volume. Obtain the size by using the `kubectl get pv <~~PV_NAME~~ PV-name > -o yaml` command, which is displayed in **size**.

If the volume is expanded or it is a statically provisioned PersistentVolume, obtain the size by using the `kubectl get pv <~~PV_NAME~~ PV-name >` command, which is displayed in **CAPACITY**.

(4) Specify the same StorageClass name as the one used for dataSource.

Command examples

- Create a PersistentVolumeClaim for a clone:

```
# kubectl create -f pvc-from-pvc-sample.yaml
```

4.8. Volume expansion

This feature can expand the capacity of an existing volume. There is no need to delete and recreate the Pod for volume expansion.

- Confirm completion of volume expansion with the `kubectl get pvc` command, which is displayed in **CAPACITY**. Do not shut down the OS or drain the node before volume expansion completes.
- In NVMe over FC, volume expansion in the state in which the PVC is attached to a host is not supported. Expand a volume in the state in which the PVC is detached from the host.

This feature requires the following resources:

- StorageClass
- PersistentVolumeClaim

Volume expansion has the following restrictions:

- The minimum additional size for volume expansion is 1 GiB.
- The maximum additional size for volume expansion is 7 TiB or a value that does not exceed the warning threshold of pool capacity. If you add more than 7 TiB, execute the command again.
- Volume capacity cannot be reduced.
- While `allowVolumeExpansion` of StorageClass is set to `false`, a PersistentVolume created with this setting cannot be expanded.
- The size obtained by the `kubectl get pv <~~PV_NAME~~ PV-name > -o yaml` command is not updated after the volume is expanded. If the volume is expanded, obtain the size by using the `kubectl get pv <~~PV_NAME~~ PV-name >` command, which is displayed in **CAPACITY**.

Command examples

- Expand the capacity of an existing volume pvc-sample to 5GiB:

```
# kubectl patch pvc pvc-sample --patch \
'{"spec":{"resources":{"requests":{"storage": "5Gi"}}}}'
```

- Confirm the completion of volume expansion by looking at **CAPACITY**:

```
# kubectl get pv <~~PV_NAME~~ PV-name >
NAME            CAPACITY     ACCESS MODES      RECLAIM POLICY
STATUS CLAIM                 STORAGECLASS      REASON AGE
<~~PV_NAME~~ PV-name >      5Gi          RWO               Delete
Bound default/pvc-sample     sc-sample               35s
```

If you want to change the size of an LDEV created ~~with the Storage Plug-in for Containers, use the~~ with Storage Plug-in for Containers, use the `kubectl` command instead of using the storage system management software to change the size of the LDEV.

4.9. Raw block volume

Kubernetes supports raw block volumes in addition to filesystem volumes. This section describes how to apply a raw block volume.

This feature requires the StorageClass.

Parameter references for pvc-sample-block.yaml

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: pvc-sample-block    #(1)
spec:
  accessModes:
  - ReadWriteOnce    #(2)
  volumeMode: Block
  resources:
    requests:
      storage: 1Gi    #(3)
  storageClassName: sc-sample    #(4)
```

Legend:

(1) PersistentVolumeClaim name

(2) Specify `ReadWriteOnce` or `ReadWriteMany`.

(3) Volume size

(4) StorageClass name

Parameter references for pod-sample-block.yaml

```
apiVersion: v1
kind: Pod
metadata:
  name: pod-sample-block               #(1)
spec:
  containers:
    - name: my-busybox
      image: busybox
      volumeDevices:
      - devicePath: "/block"    #(2)
        name: sample-volume
      command: ["sleep", "1000000"]
      imagePullPolicy: IfNotPresent
  volumes:
    - name: sample-volume
      persistentVolumeClaim:
        claimName: pvc-sample-block    #(3)
```

Legend:

(1) Pod name

(2) Path (path where the volume is mounted in the container)

(3) PersistentVolumeClaim name

Command examples

- Create a PersistentVolumeClaim for a raw block volume:

```
# kubectl create -f pvc-sample-block.yaml
```

- Create a Pod for a raw block volume:

```
# kubectl create -f pod-sample-block.yaml
```

4.10. ReadOnlyMany

You can mount a volume on one or many nodes in your Kubernetes cluster and perform read-only operations.

To create a PersistentVolumeClaim with ReadOnlyMany, you must create the PersistentVolumeClaim from an existing PVC.

Use the PersistentVolumeClaim manifest file used in the Volume cloning section and specify ReadOnlyMany, as shown in the following example.

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: pvc-rox-sample
spec:
  dataSource:
    name: pvc-sample
    kind: PersistentVolumeClaim
    apiGroup: ""
  accessModes:
  - ReadOnlyMany # Specify "ReadOnlyMany" here.
  resources:
    requests:
      storage: 1Gi
  storageClassName: sc-sample
```

4.11. Resource partitioning

By using this function, you can partition storage system resources for each Kubernetes cluster.

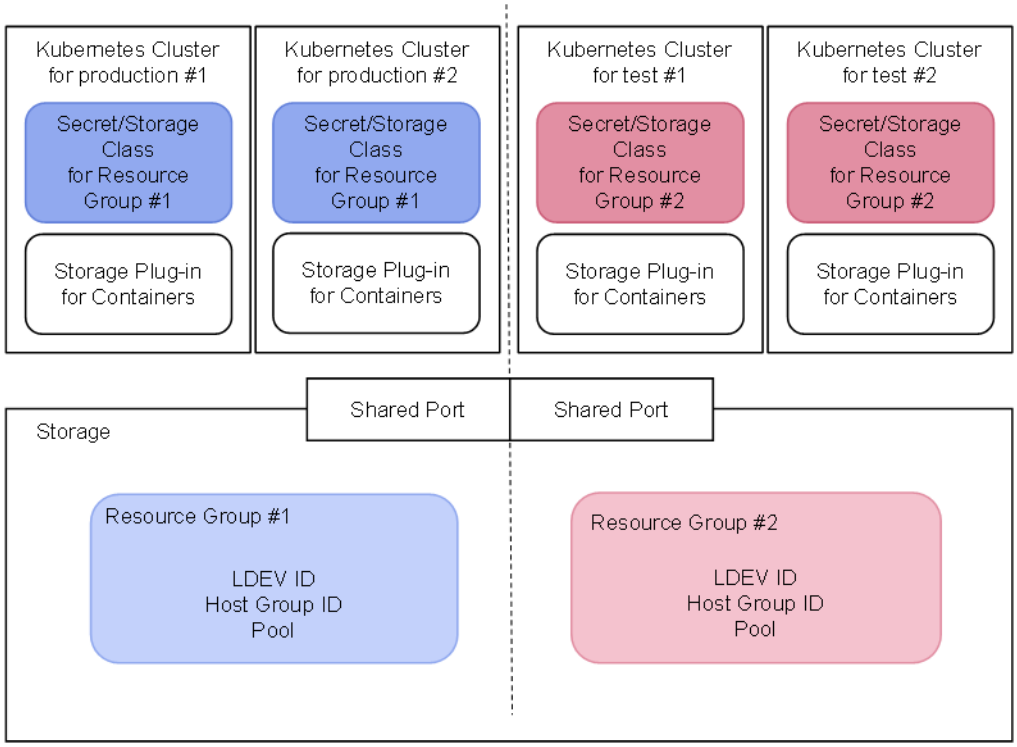
The following are examples of resource partitioning:

- You can restrict the range of LDEV IDs added to a resource group for a specific Kubernetes cluster.
- You can isolate the impacts between Kubernetes clusters.

Before you use the resource partitioning, the storage system settings, Secret and StorageClass settings, are required.

Supported configurations

The following are examples of configurations in which storage system resources can be partitioned.



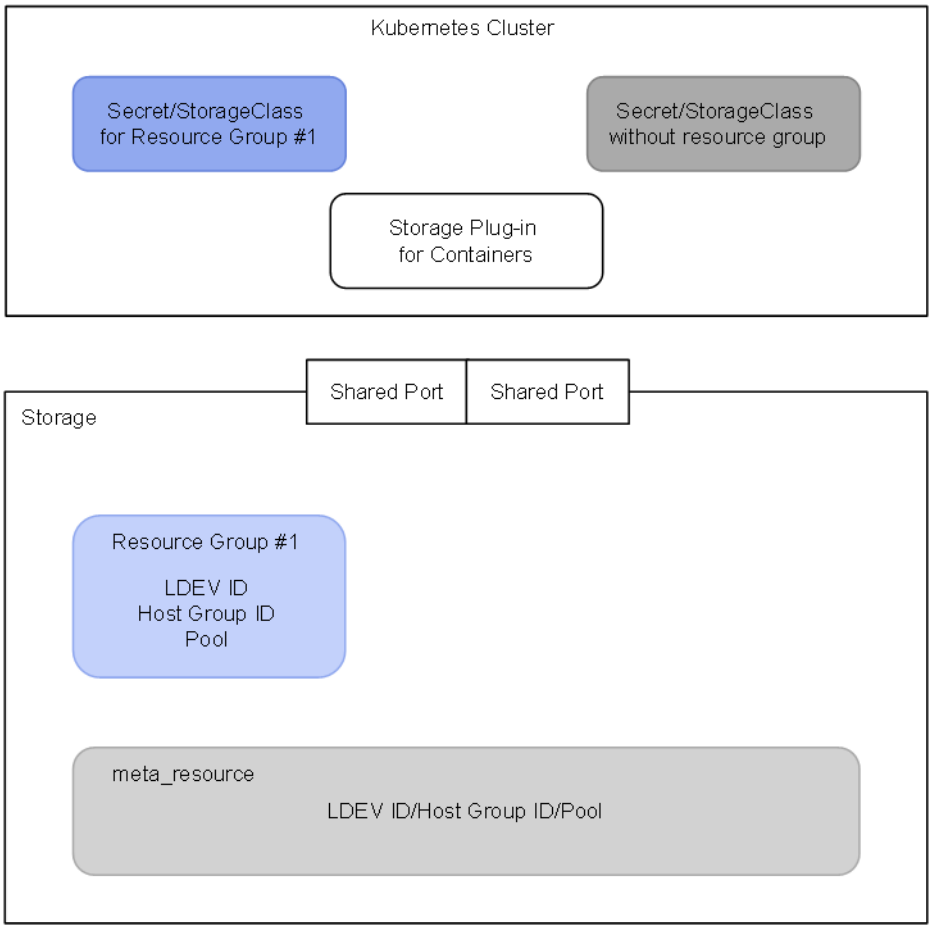
Unsupported configurations

The following are examples of configurations that are not supported.

Example 1

You cannot include both the following configurations in the same Kubernetes cluster.

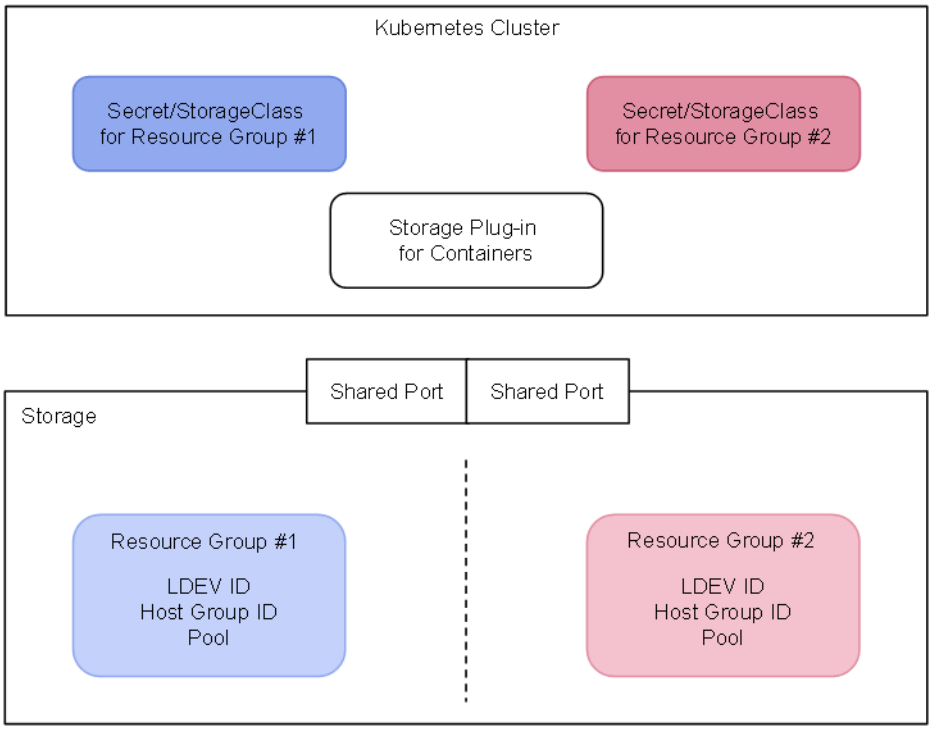
- StorageClass and Secret are configured for a resource group.
- StorageClass and Secret are temporarily configured for use with meta resource.



Example 2

If multiple resource groups are configured for a single storage system, each of those resource groups cannot correspond to a resource group in the same Kubernetes cluster.

Only one resource group (containing storageClass and Secret) per storage system can be configured for a Kubernetes cluster.



Storage system requirements and settings

Set your storage system to meet the following requirements:

Storage system resources	Descriptions
Resource group	You cannot use multiple resource groups for a single Kubernetes cluster. Virtual storage machines are not supported.
Storage system user group and Storage system user	Storage system users must have access only to the resource group that you created. The storage system user must not have access to other resource groups.
Pool	Create a pool from pool volumes with the resource group that you have created.
LDEV	Allocate the necessary number of unused LDEV IDs to the resource group. If you enable the adaptive data reduction function, a deduplication system data volume is created. Register the LDEV ID required to allocate this volume to each resource group. For details about the number of LDEV IDs that need to be registered, see the user guide for the storage system in your environment.
Host Group	For each port of a storage system defined in StorageClass, prepare the same number of host group IDs as the number of hosts. For example, if the number of hosts is 3 and the number of ports is 2, a total of 6 host group IDs are required. For each storage system port, allocate the prepared host group IDs to the resource groups.
NVM subsystem	If NVMe over FC is used, assign the NVM subsystem to the resource group.
Port	If NVMe over FC is used, assign the storage system port to the resource group.

Secret settings

Specify the resource group ID of the storage system.

Example of Secret settings:

```
apiVersion: v1
kind: Secret
metadata:
  name: secret-sample
type: Opaque
data:
  url: aHR0cDovLzE3Mi4xNi4xLjE=
  user: VXNlcjAx
  password: UGFzc3dvcmQwMQ==
stringData:
  resourceGroupID: "1"    # Specify resource group ID
```

StorageClass settings

If you use iSCSI as a storage system connection, specify the port IP address in number order. If you use Fibre Channel ~~or NVMe over FC~~ or NVMe over FC as a storage system connection, no additional setting is required for StorageClass.

Examples of StorageClass settings:

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: sc-sample
provisioner: xspc.csi.hpe.com
reclaimPolicy: Delete
volumeBindingMode: Immediate
allowVolumeExpansion: true
parameters:
  serialNumber: "54321"
  poolID: "1"
  portID : CL1-A,CL2-A
  connectionType: iscsi
  portIP: "192.168.10.10, 192.168.10.11"    # Specify iSCSI Port IP Addresses.
<...>
```

4.12. Static provisioning

This function allows existing volumes in a storage system to be used as PVCs by a container orchestrator. By using this function, you can perform operations on existing volumes in the same way as PVCs dynamically provisioned by using Storage Plug-in for Containers.

4.12.1. Requirements for using static provisioning

The requirements for using static provisioning are as follows:

Make sure that the volume meets the following requirements:

~~Make sure that the LDEV meets the following requirements:~~

- The LDEV has the THP attribute.
- The format of the nickname is ~~spc-*< 10-digit-hexadecimal-number >.*~~ spc-*<10-digit-hexadecimal-number>.*
  <10-digit-hexadecimal-number> needs to be a unique value for each LDEV.
  If the nickname is the same as that of another LDEV, the functions supported by Storage Plug-in for Containers might not work properly.
  ~~The following is an example~~ You can also use a command to generate a ~~random~~ unique character string for a ~~nickname:~~ nickname. The following is an example command:

```
# echo spc-$(cat /dev/urandom | tr -dc a-f0-9 | head -c 10)
```

- The LDEV is not mapped to a port.
- No pairs are formed.

For other requirements, see Requirements.

If the LDEV is assigned to a specific resource group, also see the storage system requirements in Resource partitioning.

4.12.2. Creating Secret and StorageClass

Create the Secret and StorageClass to be specified when creating a PV and PVC.

For details about the settings in the YAML files for Secret and StorageClass, see Secret settings and StorageClass settings. ~~If the LDEV is assigned to a specific resource group, also see the descriptions of Secret settings and StorageClass settings in~~ The volume is assigned to a specific resource group, also see the descriptions of Secret settings and StorageClass settings in Resource partitioning.

For the parameters of StorageClass, specify values based on the status of the target ~~LDEV.~~ volume. If the values specified for the parameters do not match the actual ~~LDEV~~ volume status, the functions supported by Storage Plug-in for Containers might not work properly.

4.12.3. Creating a PV

Create a PV to be associated with a PVC.

~~After performing this procedure to create a PV, you must perform the procedure described in Creating a PVC to complete the association between the created PV and a PVC.~~
~~If you perform PersistentVolumeClaim settings before performing the procedure described in Creating a PVC, the created PV might be associated with an unintended PVC.~~

- You can create only one PV for one volume.
  If you create more than one PV for one volume, the functions supported by Storage Plug-in for Containers might not work properly.
- If values specified for the parameters of the PV are incorrect, an unexpected error message might be displayed.

1. Create a YAML file.
   For the parameters of the PV, specify values based on the status of the target ~~LDEV~~ volume and the settings of the created StorageClass. If the values specified for the parameters do not match the actual ~~LDEV~~ volume status, the functions supported by Storage Plug-in for Containers might not work properly.
   Example of the YAML file:

```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: static-pv
  annotations:
    pv.kubernetes.io/provisioned-by: xspc.csi.hpe.com
spec:
  persistentVolumeReclaimPolicy: Delete
  accessModes:
  - ReadWriteOnce
  capacity:
    storage: 1Gi
  volumeMode: Filesystem
  csi:
    fsType: ext4
    volumeAttributes:
      connectionType: fc
      ports: CL1-A, CL2-A
    volumeHandle: 01--scsi--900000070010--50000--spc-c3d46c5a71
    driver: xspc.csi.hpe.com controllerExpandSecretRef: name: secret-sample
namespace: default controllerPublishSecretRef: name: secret-sample namespace:
default storageClassName: sc-sample claimRef: name: static-pvc namespace: default
```

   Parameters:

Parameter	Description	Required or optional
metadata.name	Specify the PV name.	Required

Parameter	Description	Required or optional
metadata.annotations.pv.kubernetes.io/provisioned-by	Specify xspc.csi.hpe.com.	Required if Delete is specified for persistentVolumeReclaimPolicy
spec.persistentVolumeReclaimPolicy	Specify the reclaim policy. If you specify Retain, the PV and LDEV will not be deleted even when the PVC is deleted. Instead, the PV will be in the released state, so if you want to reuse the LDEV, you will need to re-create the PV. For details, see https://kubernetes.io/docs/concepts/storage/persistent-volumes/#retain. If you specify Delete, the PV and LDEV will be deleted when the PVC is deleted. The default value is Retain.	Optional
spec.accessModes	Specify the access mode. For details about supported access modes, see PersistentVolumeClaim settings.	Required
spec.capacity.storage	Specify the LDEV size.	Required
spec.volumeMode	Specify Filesystem or Block. If the target LDEV was used as a raw block volume, be sure to specify Block. If the LDEV was used as a raw block volume and you specify Filesystem, it will be formatted as a file system and the existing data will be deleted. The default value is Filesystem.	Optional
spec.csi.fsType	Specify the file system type of the target LDEV. The default value is ext4. If you specify Block for volumeMode, this parameter is disabled. For details about supported file system types, see StorageClass settings.	Optional
spec.csi.volumeAttributes.connectionType	Specify the type of connection between the storage system and the node. For details about supported connection types, see StorageClass settings.	Required
spec.csi.volumeAttributes.ports	Specify the storage port ID. For multipath configurations, use commas to delimit the storage port IDs.	Required
spec.csi.volumeAttributes.portIPs	Specify the storage port IP address. For multipath configurations, use commas to delimit the storage port IP addresses.	Required if iscsi is specified for connectionType and the LDEV is assigned to a specific resource group
spec.csi.volumeAttributes.nvmSubsystemID	Specify the NVM subsystem ID of the storage system.	Required if nvme-fc is specified for connectionType

Parameter	Description	Required or optional
spec.csi.volumeHandle	Specify the value in the following format: `01--<IO-protocol>--<storage-device-ID>--<LDEV-ID>--<LDEV-nickname>` ***<IO-protocol>*** Specify the value as follows: • If fc or iscsi is specified for connectionType: scsi • If nvme-fc is specified for connectionType: nvme ***<storage-device-ID>*** Check this value by referring to the *REST API Reference Guide* for each storage model. The storage device ID is a 12-digit value and the format is as follows: *<6-digit-fixed-value-for-each-storage-model><6-digit-serial-number>* For example, the fixed value for XP8 is 900000. ***<LDEV-ID>*** Check these values by using the storage system management software. Specify a value by using a decimal number. ***<LDEV-nickname>*** Check these values by using the storage system management software.	Required
spec.csi.driver	Specify xspc.csi.hpe.com.	Required
spec.csi.controllerExpandSecretRef.name	Specify the name of the Secret.	Required
spec.csi.controllerExpandSecretRef.namespace	Specify the namespace of the Secret.	Required
spec.csi.controllerPublishSecretRef.name	Specify the name of the Secret.	Required
spec.csi.controllerPublishSecretRef.namespace	Specify the namespace of the Secret.	Required
spec.storageClassName	Specify the StorageClass name.	Required
spec.claimRef.name	Specify the PVC name to be created in Creating a PVC.	Required If you do not specify this parameter, the PV might be associated with an unintended PVC and the Storage Plug-in for Containers functions might not work properly.
spec.claimRef.namespace	Specify the namespace of the PVC to be created in Creating a PVC.	Required

2. Deploy the YAML file.

```
# kubectl apply -f <YAML-file-name>
```

4.12.4. Creating a PVC

Create a PVC so that Storage Plug-in for Containers functions can be used.

1. Create a YAML file.
   Example of the YAML file:

```
apiVersion: v1 kind: PersistentVolumeClaim metadata: name: static-pvc namespace: default spec: accessModes: - ReadWriteOnce resources: requests: storage: 1Gi storageClassName: sc-sample volumeMode: Filesystem volumeName: static-pv
```

Parameters:

Parameter	Description	Required or optional
`metadata.name`	Specify the PVC name.	Required
`metadata.namespace`	Specify the namespace of the PVC.	Optional
`spec.accessModes`	Specify the same value as the value of `accessModes` for the PV.	Required
`spec.resources.requests.storage`	Specify the same value as the value of `capacity.storage` for the PV.	Required
`spec.storageClassName`	Specify the same value as the value of `storageClassName` for the PV.	Required
`spec.volumeName`	Specify the PV name.	Required If you do not specify this parameter, the PVC might be associated with an unintended PV and the Storage Plug-in for Containers functions might not work properly.
`spec.volumeMode`	If you specified `volumeMode` for the PV, specify the same value for this parameter as you did for the PV. The default value is `Filesystem`.	Required if you specified `Block` for `volumeMode` of the PV

2. Deploy the YAML file.

```
# kubectl apply -f <YAML-file-name>
```

3. Confirm that STATUS of the PVC is Bound.When STATUS of the PVC is Bound, you can use the functions supported by Storage Plug-in for Containers.

4.12.5. Troubleshooting when using static provisioning

A VolumeAttachment remains after you delete the Pod

If there are mistakes in the format or values of parameters in the YAML file for the PV, the creation of the Pod will fail. If you delete this Pod, a VolumeAttachment might remain.

The following are examples of failed attempts to create a Pod:

- Incorrect format of `volumeHandle` (error code: `0x0000c002`)
- Insufficient permissions for LDEV (error code: `0x00001007`)
- `controllerPublishSecretRef` is not specified (error code: `0x0000c00f`)

To delete a VolumeAttachment, run the following command:

```
# kubectl patch volumeattachments <VolumeAttachment-name> --type merge -p '{"metadata":{"finalizers":null}}'
```

The PV remains after you delete the PVC

If there are mistakes in the format or values of parameters in the YAML file for the PV, even if you specify `Delete` for `persistentVolumeReclaimPolicy`, the PV might remain after you delete the PVC.

The following shows examples of a PV remaining after a PVC is deleted:

- Incorrect format of `volumeHandle` (error code: `0x0000c002`)
- Insufficient permissions for LDEV (error code: `0x00001007`)

To delete the PV, perform the following procedure:

1. If you need to delete the ~~LDEV, use the~~ volume, use the `volumeHandle`information to check the target ~~LDEV.~~ volume.

```
# kubectl get pv <PV-name> -o yaml
```

2. Delete the PV.

```
# kubectl delete pv <PV-name>
```

3. If you need to delete the ~~LDEV, use the management software for the storage system to delete it.~~ volume, use the management software for the storage system to delete it.

The volume remains after you delete the PVC

For the LDEV specified by *<LDEV-ID>* of `volumeHandle`, if an incorrect value is specified for the value of *<LDEV-nickname>* and if you specify `Delete` for `persistentVolumeReclaimPolicy`, the PV will be deleted after the PVC is deleted, but the volume will not be deleted.

To delete the volume, use the management software for the storage system to delete it.

5. Upgrade

This chapter describes how to upgrade Storage Plug-in for Containers. The upgrade method you use depends on whether your environment is OpenShift or Kubernetes.

~~If alpha versions of VolumeSnapshotClass and VolumeSnapshot are present in your environment, remove them before upgrading Storage Plug-in for Containers .~~

- If alpha versions of VolumeSnapshotClass and VolumeSnapshot are present in your environment, remove them before upgrading Storage Plug-in for Containers.
- If you are using an old version of OpenShift or Kubernetes, the latest Storage Plug-in for Containers might not operate. If you want to use the latest Storage Plug-in for Containers, upgrade OpenShift or Kubernetes first.
- Before upgrading Storage Plug-in for Containers, back up the settings configured in Configuration of Storage Plug-in for Containers instance. The settings that are backed up might also be used for the new version of Storage Plug-in for Containers. If you do not have the `xspc_v1_xspc.yaml` file that was used when configuring the settings, run `kubectl get xspc -A -o yaml` to obtain the configured settings.

5.1. Upgrade on OpenShift

For OpenShift, you can upgrade Storage Plug-in for Containers using OpenShift web console.

1. Delete XSPC on the Storage Plug-in for Containers tab for Operator Details.
2. Uninstall the Operator of Storage Plug-in for Containers.
3. Install new Storage Plug-in for Containers. See Installation on OpenShift.

5.2. Upgrade on Kubernetes

For Kubernetes, you can upgrade Storage Plug-in for Containers by following the steps below.

1. ~~Delete the Storage Plug-in for Containers and the Operator by using the previous version package:~~ Delete Storage Plug-in for Containers and the Operator by using the previous version package:

```
# kubectl delete -f xspc_v1_xspc.yaml
# kubectl delete -f xspc-operator.yaml
```

2. ~~Install the new~~ Install newStorage Plug-in for Containers. See the topic Installation on Kubernetes.

6. Re-creation

This chapter describes how to re-create Storage Plug-in for Containers. The re-creation method depends on whether your environment is OpenShift or Kubernetes. To re-create Storage Plug-in for Containers, delete Storage Plug-in for Containers, and then create it again.

6.1. Deleting Storage Plug-in for Containers

OpenShift

Access the OpenShift web console, and delete the Storage Plug-in for Containers instance.

Kubernetes

Run the following command to delete the Storage Plug-in for Containers instance:

```
# kubectl delete xspc -n <SPC_NAMESPACE SPC_NAMESPACE-value > xspc
```

For <SPC_NAMESPACE SPC_NAMESPACE-value >, specify the namespace specified in `xspc_v1_xspc.yaml`.

6.2. Re-creating Storage Plug-in for Containers

OpenShift

Access the OpenShift web console, and create a Storage Plug-in for Containers instance.

For details, see Installation on OpenShift.

Kubernetes

Run the following command to create a Storage Plug-in for Containers instance:

```
# kubectl create -f xspc_v1_xspc.yaml
```

For details, see Installation on Kubernetes.

7. Uninstallation

This chapter describes how to uninstall Storage Plug-in for Containers. This step includes removing any PersistentVolumeClaims, PersistentVolumes, StorageClasses, Storage Plug-in for Containers, and other elements. The uninstallation method you use depends on whether your environment is OpenShift or Kubernetes.

7.1. Uninstallation on OpenShift

For OpenShift, you can uninstall Storage Plug-in for Containers using OpenShift web console.

1. Delete all Pods which are using the volumes created by Storage Plug-in for Containers.
2. Delete the ~~PersistentVolumeClaim, the StorageClass and the Secret related to~~ VolumeSnapshotClass, VolumeSnapshot, PersistentVolumeClaim, the StorageClass, and the Secret that were created in relation toStorage Plug-in for Containers.
3. Delete XSPC on the HPE XP Storage Plug-in for Containers tab of the Operator Details.
4. Uninstall the Operator of Storage Plug-in for Containers.

7.2. Uninstallation on Kubernetes

For Kubernetes, you can uninstall Storage Plug-in for Containers by following the steps below.

1. Delete all Pods which are using the volumes created by Storage Plug-in for Containers.
2. Delete the ~~PersistentVolumeClaim, the StorageClass, and the Secret related to~~ VolumeSnapshotClass, VolumeSnapshot, PersistentVolumeClaim, the StorageClass, and the Secret that were created in relation toStorage Plug-in for Containers.
3. ~~Delete the Storage Plug-in for Containers and resource for the Operator:~~ Delete Storage Plug-in for Containers and the resources for the Operator:

```
# kubectl delete -f xspc_v1_xspc.yaml
# kubectl delete -f xspc-operator.yaml
# kubectl delete -f xspc-operator-namespace.yaml
```

4. Delete the Secret for Red Hat registry.

8. Troubleshooting

When troubleshooting, you must understand what information to collect when an error occurs, the cases where an error occurs, and what action to take in each case.

8.1. Collecting information for troubleshooting

If a failure occurs in Storage Plug-in for Containers, collect the following information. Provide the collected information to customer support when you make an inquiry.

8.1.1. Information needed when contacting support

You can provide the following information for Storage Plug-in for Containers and the storage system to customer support for advanced troubleshooting.

Information	Procedure
Command execution logs	Retrieve the command that you ran and the result of running that command.
Result of running the `kubectl describe` command for the operation target resource	Run the following command for the resource you operated. `# kubectl describe <resource> -n <Storage-Plug-in-for-Containers-namespace> <resource-name>`
Cluster information	Run the following command: `# kubectl cluster-info dump -A > dump.txt`
Pod information	Retrieve the command that you ran in step 1 of Collecting logs for Storage Plug-in for Containers and the result of running that command.
Operator logs	See Collecting logs for Storage Plug-in for Containers.
CSI controller logs	See Collecting logs for Storage Plug-in for Containers.
CSI node logs	See Collecting logs for Storage Plug-in for Containers.
PVC-related manifests	Get the YAML files for StorageClass, Secret, and PersistentVolumeClaim.
Snapshot-related manifests	Get the YAML files for VolumeSnapshotClass, Secret, and VolumeSnapshot.
Snapshot-related logs	Collect the snapshot controller logs that you installed in the Volume snapshot chapter.
Application manifests	Get the YAML files for applications that uses Storage Plug-in for Containers PVCs.
Storage logs	See Collecting storage system information for HPE Storage.

8.1.2. Collecting logs for Storage Plug-in for Containers

You can retrieve logs for your running containers using the `kubectl logs` command. To collect Storage Plug-in for Containers logs, you need to collect logs from the Operator, CSI controller, and CSI node.

If necessary, set up cluster-level logging to save logs:
https://kubernetes.io/docs/concepts/cluster-administration/logging/

~~To retrieve logs from the operator, run the following command:~~

```
~~# kubectl logs -n <~~
~~namespace~~
~~> xspc-operator-controller-manager-<~~
~~id~~
~~>~~
```

~~To retrieve logs from the CSI controller, run the following command:~~

```
~~# kubectl logs -n <~~
~~namespace~~
~~> xspc-csi-controller-<~~
~~id~~
~~> -c~~
~~xspc-csi-driver~~
```

~~To retrieve logs from the CSI node, run the following command:~~

```
~~# kubectl logs -n <~~
~~namespace~~
~~> xspc-csi-node-<~~
~~id~~
~~> -c~~
~~xspc-csi-driver~~
```

~~You will see multiple CSI node Pods since this is deployed as a DaemonSet. Collect logs from all these Pods.~~

1. Before retrieving logs, run the following command to check the Pod name.

```
# kubectl get pod -A -o wide
```

2. Run the following command to retrieve logs.
   - Operator

```
# kubectl logs -n <Storage-Plug-in-for-Containers-namespace> xspc-operator-controller-manager-<id>
```

   - CSI controller

```
# kubectl logs -n <Storage-Plug-in-for-Containers-namespace> xspc-csi-controller-<id> -c xspc-csi-driver
```

You must specify `-c xspc-csi-driver` to run the command.

   - CSI node

```
# kubectl logs -n <Storage-Plug-in-for-Containers-namespace> xspc-csi-node-<id> -c xspc-csi-driver
```

   - You must specify `-c xspc-csi-driver` to run the command.
   - You will see multiple CSI node Pods because this is deployed as a DaemonSet. Collect logs from all these Pods.

3. Perform the following procedure to retrieve the directories as old logs might be rotated and removed from the retrieved logs.
   a. Check the target node.
      From the result of step 1, find the line where the value of `NAME` includes `xspc-csi-controller`, and check the value of `NODE` in the same line.
   b. Retrieve the directories stored under `/var/log/pods` for the node you checked in step a.

8.1.3. Collecting storage system information for HPE Storage

If you are using an SVP, collect the regular dump files.

If you are not using an SVP, collect system dumps using the maintenance utility. For details about how to collect the dump files of storage systems, see the *Remote Web Console User Guide*.

8.2. Viewing the volume properties of PersistentVolume

When a volume is dynamically created by Storage Plug-in for Containers, information about the created volume is set in the `spec.csi.volumeAttributes` of the PersistentVolume. You can view these properties using the `kubectl get pv <~~PV-name~~ PV-name > -o yaml` command.

These properties are mainly used for internal purposes. The following tables describe some properties that can be helpful when troubleshooting.

Volume properties for HPE storage.

Property	Description
ldevIDDec	Decimal LDEV ID
ldevIDHex	Hexadecimal LDEV ID
size	Capacity of the volume Capacity shown here is the original capacity used when creating the volume.

8.3. Notes on forcibly deleting a Pod

If you forcibly delete a Pod from a specific node, the deleted Pod and the information on the PVC associated with the Pod might remain on the relevant node, and an unexpected error might occur.

To properly delete this information, you must restart the relevant node before using the node again.

8.4. Creating and deleting PersistentVolumeClaim simultaneously

When PersistentVolumeClaims are created or deleted simultaneously, the storage might get overloaded and cause errors `0x0000100b`, `0x0000100f`, `0x0000101a`, or `0x0000f007`. This problem can be reduced by specifying the `--worker-threads` argument to the `csi-provisioner` container. This argument limits the number of simultaneously running create and delete operations. The default value is 20.

The following example shows how to reduce the number of `--worker-threads` to 10. For the YAML configuration, refer to Configuration of Storage Plug-in for Containers instance.

```
apiVersion: csi.hpe.com/v1
kind: XSPC
metadata:
  name: xspc
  namespace: <SPC_NAMESPACE SPC_NAMESPACE-value > spec: imagePullSecrets: - regcred-redhat-com - regcred-redhat-io controller: containers: - name: csi-provisioner args: - --csi-address=/csi/csi-controller.sock - --timeout=300s - --v=5 - --worker-threads=10 - --default-fstype=ext4 regcred-redhat- com
  controller:
    containers:
      - name: csi-provisioner
        args:
        - --csi-address=/csi/csi-controller.sock
        - --timeout=300s
        - --v=5
        - --worker-threads=10
        - --default-fstype=ext4
```

If the problem persists, contact technical support.

8.5. Host group settings

If you encounter error `0x00001023`, you must modify the host group in the storage. Storage Plug-in for Containers searches the host group named "spc-<*wwn1*>-<*wwn2*>-<*wwn3*>", based on the naming rules (see Host group and iSCSI target naming rules). The error was likely generated because the host group's name may not follow the "spc-<*wwn1*>-<*wwn2*>-<*wwn3*>" naming format. To resolve the issue, delete the host group shown in the error message and rename the host group that has host WWNs.

1. Delete Storage Plug-in for Containers.
   For details, see Deleting Storage Plug-in for Containers.
2. Delete the host group that is specified in the error message.

3. Search host groups that have WWNs for each host, and delete them or rename them to "spc-<*wwn1*>-<*wwn2*>-<*wwn3*>".
4. Create Storage Plug-in for Containers.
   For details, see Re-creating Storage Plug-in for Containers.

8.6. Timeout errors when creating or deleting Pods

Depending on the status of the storage system or host OS, processing of Storage Plug-in for Containers might slow down and a timeout error might occur.

If a timeout error continues to occur when you create or delete a Pod, see Re-creation and re-create a Storage Plug-in for Containers instance. If the timeout error is not resolved, restart the node to which the Pod where the timeout error occurs is assigned. Restarting a node might affect other environments. For this reason, sufficiently check the scope of the impact, see the Kubernetes and OpenShift documentation, and then carefully restart the node while checking the procedures one by one.

9. Websites

General websites

Single Point of Connectivity Knowledge (SPOCK) Storage compatibility matrix

https://www.hpe.com/storage/spock

Storage white papers and analyst reports

https://www.hpe.com/storage/whitepapers

For additional websites, see Support and other resources.

Documentation websites for XP

XP Configuration Manager and XP Intelligent Management Suite do not have product pages. XP Configuration Manager deliverables are available on the XP Command View Advanced Edition pages. XP Intelligent Management Suite deliverables are available on the Automation Director, Data Protection Manager, and Intelligent Storage Manager product pages.

XP8 Storage

https://www.hpe.com/support/XP8manuals

XP7 Storage

https://www.hpe.com/support/XP7manuals

XP8 Command View Advanced Edition

https://www.hpe.com/support/CVAE8manuals

XP7 Command View Advanced Edition

https://www.hpe.com/support/CVAE7/manuals

XP8 Automation Director

https://www.hpe.com/support/XP8-AutomationDirector-manuals

XP7 Automation Director

https://www.hpe.com/support/XP7-AutomationDirector-manuals

XP8 Data Protection Manager

https://www.hpe.com/support/XP8-DataProtectionMgr-manuals

XP7 Data Protection Manager

https://www.hpe.com/support/XP7-DataProtectionMgr-manuals

XP Intelligent Storage Manager

https://www.hpe.com/support/XP-IntelligentStorageMgr-manuals

10. Conventions for capacity values

Logical capacity units (for example, logical device capacity, cache memory capacity) are calculated based on the values that are outlined in the following table.

Logical capacity unit	Value
1 KiB	1,024 (2^{10}) bytes
1 MiB	1,024 KiB or $1{,}024^2$ bytes
1 GiB	1,024 MiB or $1{,}024^3$ bytes
1 TiB	1,024 GiB or $1{,}024^4$ bytes
1 PiB	1,024 TiB or $1{,}024^5$ bytes
1 EiB	1,024 PiB or $1{,}024^6$ bytes

11. Websites

General websites

Single Point of Connectivity Knowledge (SPOCK) Storage compatibility matrix

https://www.hpe.com/storage/spock

Storage white papers and analyst reports

https://www.hpe.com/storage/whitepapers

For additional websites, see Support and other resources.

Documentation websites for XP

XP Configuration Manager and XP Intelligent Management Suite do not have product pages. XP Configuration Manager deliverables are available on the XP Command View Advanced Edition pages. XP Intelligent Management Suite deliverables are available on the Automation Director, Data Protection Manager, and Intelligent Storage Manager product pages.

XP8 Storage

https://www.hpe.com/support/XP8manuals

XP7 Storage

https://www.hpe.com/support/XP7manuals

XP8 Command View Advanced Edition

https://www.hpe.com/support/CVAE8manuals

XP7 Command View Advanced Edition

https://www.hpe.com/support/CVAE7/manuals

XP8 Automation Director

https://www.hpe.com/support/XP8-AutomationDirector-manuals

XP7 Automation Director

https://www.hpe.com/support/XP7-AutomationDirector-manuals

XP8 Data Protection Manager

https://www.hpe.com/support/XP8-DataProtectionMgr-manuals

XP7 Data Protection Manager

https://www.hpe.com/support/XP7-DataProtectionMgr-manuals

XP Intelligent Storage Manager

https://www.hpe.com/support/XP-IntelligentStorageMgr-manuals

12. Support and other resources

12.1. Accessing Hewlett Packard Enterprise Support

- For live assistance, go to the Contact Hewlett Packard Enterprise Worldwide website:
  https://www.hpe.com/assistance
- To access documentation and support services, go to the Hewlett Packard Enterprise Support Center website:
  https://www.hpe.com/support/hpesc

Information to collect

- Technical support registration number (if applicable)
- Product name, model or version, and serial number
- Operating system name and version
- Firmware version
- Error messages
- Product-specific reports and logs
- Add-on products or components
- Third-party products or components

12.2. Accessing updates

- Some software products provide a mechanism for accessing software updates through the product interface. Review your product documentation to identify the recommended software update method.
- To download product updates:

  Hewlett Packard Enterprise Support Center

  https://www.hpe.com/support/hpesc

  Hewlett Packard Enterprise Support Center: Software downloads

  https://www.hpe.com/support/downloads

  My HPE Software Center

  https://www.hpe.com/software/hpesoftwarecenter
- To subscribe to eNewsletters and alerts:
  https://www.hpe.com/support/e-updates
- To view and update your entitlements, and to link your contracts and warranties with your profile, go to the Hewlett Packard Enterprise Support Center **More Information on Access to Support Materials** page:
  https://www.hpe.com/support/AccessToSupportMaterials

Access to some updates might require product entitlement when accessed through the Hewlett Packard Enterprise Support Center. You must have an HPE Passport set up with relevant entitlements.

12.3. Remote support

Remote support is available with supported devices as part of your warranty or contractual support agreement. It provides intelligent event diagnosis, and automatic, secure submission of hardware event notifications to Hewlett Packard Enterprise, which initiates a fast and accurate resolution based on the service level of your product. Hewlett Packard Enterprise strongly recommends that you register your device for remote support.

If your product includes additional remote support details, use search to locate that information.

HPE Get Connected

https://www.hpe.com/services/getconnected

HPE Pointnext Tech Care

https://www.hpe.com/services/techcare

HPE Datacenter Care services

https://www.hpe.com/services/datacentercare

12.4. Warranty information

To view the warranty information for your product, see the links provided below:

HPE ProLiant and IA-32 Servers and Options

https://www.hpe.com/support/ProLiantServers-Warranties

HPE Enterprise and Cloudline Servers

https://www.hpe.com/support/EnterpriseServers-Warranties

HPE Storage Products

https://www.hpe.com/support/Storage-Warranties

HPE Networking Products

https://www.hpe.com/support/Networking-Warranties

12.5. Regulatory information

To view the regulatory information for your product, view the *Safety and Compliance Information for Server, Storage, Power, Networking, and Rack Products*, available at the Hewlett Packard Enterprise Support Center:

https://www.hpe.com/support/Safety-Compliance-EnterpriseProducts

Additional regulatory information

Hewlett Packard Enterprise is committed to providing our customers with information about the chemical substances in our products as needed to comply with legal requirements such as REACH (Regulation EC No 1907/2006 of the European Parliament and the Council). A chemical information report for this product can be found at:

https://www.hpe.com/info/reach

For Hewlett Packard Enterprise product environmental and safety information and compliance data, including RoHS and REACH, see:

https://www.hpe.com/info/ecodata

For Hewlett Packard Enterprise environmental information, including company programs, product recycling, and energy efficiency, see:

https://www.hpe.com/info/environment

12.6. Documentation feedback

HPE is committed to providing documentation that meets your needs. To help us improve the documentation, use the **Feedback** button and icons (located at the bottom of an opened document) on the Hewlett Packard Enterprise Support Center portal (https://www.hpe.com/support/hpesc) to send any errors, suggestions, or comments. All document information is captured by the process.

Index

, 43

R

regulatory information, 46

W

warranty information, 46