## Dependencies

Installing any of the Dell EMC CSI Drivers requires a few utilities to be installed on the system running the installation.

|  |  |
| --- | --- |
| Dependency | Usage |
| kubectl | Kubectl is used to validate that the Kubernetes system meets the requirements of the driver. |
| helm | Helm v3 is used as the deployment tool for Charts. See, Install Helm 3 for instructions to install Helm 3. |
| sshpass | sshpass is used to check certain pre-requisities in worker nodes (in chosen drivers). |

In order to use these tools, a valid KUBECONFIG is required. Ensure that either a valid configuration is in the default location or that the KUBECONFIG environment variable points to a valid confiugration before using these tools.

## Capabilities

This project provides the following capabilitites, each one is discussed in detail later in this document.

* Install a driver. When installing a driver, options are provided to specify the target namespace as well as options to control the types of verifications to be performed on the target system.
* Upgrade a driver. Upgrading a driver is an effective way to either deploy a new version of the driver or to modify the parameters used in an initial deployment.
* Uninstall a driver. This removes the driver and any installed storage classes.
* Verify a Kubernetes system for suitability with a driver. These verification steps differ, slightly, from driver to driver but include verifiying version compatibility, namespace availability, existance of required secrets, and validating worker node compatibility with driver protocols such as iSCSI, Fibre Channel, NFS, etc

Most of these usages require the creation/specification of a values file. These files specify configuration settings that are passed into the driver and configure it for use. To create one of these files, the following steps should be followed:

* Download a template file for the driver to a new location, naming this new file is at the users discretion. The template files are always found at https://github.com/dell/helm-charts/raw/csi-powerstore-2.15.0/charts/csi-powerstore/values.yaml
* Edit the file such that it contains the proper configuration settings for the specific environment. These files are yaml formatted so maintaining the file structure is important.

For example, to create a values file for the PowerStore driver the following steps can be executed

# cd to the installation script directory

cd dell-csi-helm-installer

# download the template file

wget -O my-powerstore-settings.yaml https://github.com/dell/helm-charts/raw/csi-powerstore-2.15.0/charts/csi-powerstore/values.yaml

# edit the newly created values file

vi my-powerstore-settings.yaml

These values files can then be archived for later reference or for usage when upgrading the driver.

## Install A Driver

Installing a driver is performed via the csi-install.sh script. This script requires a few arguments: the target namespace and the user created values file. By default, this will verify the Kubernetes environment and present a list of warnings and/or errors. Errors must be addressed before installing, warning should be examined for their applicability. For example, in order to install the PowerStore driver into a namespace called "powerstore", the following command should be run:

./csi-install.sh --namespace powerstore --values ./my-powerstore-settings.yaml

For usage information:

[dell-csi-helm-installer]# ./csi-install.sh -h

Help for ./csi-install.sh

Usage: ./csi-install.sh options...

Options:

Required

--namespace[=]<namespace> Kubernetes namespace containing the CSI driver

--values[=]<values.yaml> Values file, which defines configuration values

Optional

--release[=]<helm release> Name to register with helm, default value will match the driver name

--upgrade Perform an upgrade of the specified driver, default is false

--node-verify-user[=]<username> Username to SSH to worker nodes as, used to validate node requirements. Default is root

--skip-verify Skip the kubernetes configuration verification to use the CSI driver, default will run verification

--skip-verify-node Skip worker node verification checks

-h Help

## Upgrade A Driver

# Powerstore

## Prerequisite

The following requirements must be met before installing the CSI Driver for PowerStore:

* A Kubernetes or OpenShift cluster (see supported versions)
* Install Helm 3.x
* Refer to the sections below for protocol specific requirements.
* If you want to use pre-configured iSCSI/FC hosts be sure to check that they are not part of any host group.
* Linux multipathing requirements (described later).
* Mount propagation is enabled on the container runtime that is being used.
* If using Snapshot feature, satisfy all Volume Snapshot requirements.
* Insecure registries are defined in Docker or other container runtime for CSI drivers that are hosted in a non-secure location.
* Ensure that your nodes support mounting NFS volumes if using NFS.
* For NVMe support the preferred multipath solution is NVMe native multipathing. The Dell Host Connectivity Guide describes the details of each configuration option.

### Fibre Channel requirements

The following requirements must be fulfilled in order to successfully use the Fiber Channel protocol with the CSI PowerStore driver:

* Zoning of the Host Bus Adapters (HBAs) to the Fibre Channel ports on the PowerStore arrays must be done.
* If the number of volumes that will be published to nodes is high, then configure the maximum number of LUNs for your HBAs on each node. See the appropriate HBA document to configure the maximum number of LUNs

### iSCSI Requirements

The following requirements must be fulfilled in order to successfully use the iSCSI protocol with the CSI PowerStore driver:

* Ensure that the necessary iSCSI initiator utilities are installed on each Kubernetes worker node. This typically includes the iscsi-initiator-utils package for RHEL or open-iscsi package for Ubuntu.
* Enable and start the iscsid service on each Kubernetes worker node. This service is responsible for managing the iSCSI initiator. You can enable the service by running the following command on all worker nodes: systemctl enable --now iscsid
* Ensure that the unique initiator name is set in /etc/iscsi/initiatorname.iscsi.
* Ensure that the iSCSI initiators are available on all the nodes where the driver node plugin will be installed.
* Ensure that the unique initiator name is set in /etc/iscsi/initiatorname.iscsi.
* Kubernetes nodes must have network connectivity to an iSCSI port on the PowerStore array that has IP interfaces.
* Ensure that the iSCSI initiators on the nodes are not a part of any existing Host or Host Group on the PowerStore arrays. The driver will create host entries for the iSCSI initiators which adheres to the naming conventions required by the driver.

Refer to the Dell Host Connectivity Guide for more information

### Linux multipathing requirements

Supported Multipathing

* Dell PowerStore supports Linux multipathing (DM-MPIO) and NVMe native multipathing.
* Configure Linux multipathing before installing the CSI Driver.

FC/iSCSI

Configuration steps:

* Install the Device Mapper Multipathing package on all nodes:

# dnf install device-mapper-multipath

# apt install multipath-tools

* Enable multipathing: mpathconf --enable --with\_multipathd y
* Edit /etc/multipath.conf to enable user\_friendly\_names and find\_multipaths.
* Ensure the mpathconf command is available on all Kubernetes nodes.

The following is a simple sample multipath.conf file. For a detailed sample, refer Dell Technologies Host Connectivity.  
defaults {

polling\_interval 5

checker\_timeout 15

disable\_changed\_wwids yes

find\_multipaths no

}

devices {

device {

vendor DellEMC

product PowerStore

detect\_prio "yes"

path\_selector "queue-length 0"

path\_grouping\_policy "group\_by\_prio"

path\_checker tur

failback immediate

fast\_io\_fail\_tmo 5

no\_path\_retry 3

rr\_min\_io\_rq 1

max\_sectors\_kb 1024

dev\_loss\_tmo 10

hardware\_handler "1 alua"

}

device {

vendor .\*

product dellemc-powerstore

uid\_attribute ID\_WWN

prio ana

failback immediate

path\_grouping\_policy "group\_by\_prio"

path\_checker "none"

path\_selector "queue-length 0"

detect\_prio "yes"

fast\_io\_fail\_tmo 5

no\_path\_retry 3

rr\_min\_io\_rq 1

max\_sectors\_kb 1024

dev\_loss\_tmo 10

}

}

On some distributions the multipathd service for changes to the configuration and dynamically reconfigures itself. If you need to manually trigger a reload you can run the following command: sudo systemctl reload multipathd

### Replication feature Requirements (Optional)

Applicable only if you decided to enable the Replication feature in values.yaml

replication:

enabled: true

Replication CRD’s

The CRDs for replication can be obtained and installed from the csm-replication project on Github. Use csm-replication/deploy/replicationcrds.all.yaml located in csm-replication git repo for the installation.

CRDs should be configured during replication prepare stage with repctl as described in install-repctl

## Installation Guide

Set up a Kubernetes cluster following the official documentation.

* Proceed to the Prerequisite.
* Complete the base installation.
* Proceed with module installation.

Install Helm 3.x

Install Helm 3.x on the master node before you install the CSI Driver for PowerStore.

Steps

Run the command to install Helm 3.x.

curl https://raw.githubusercontent.com/helm/helm/master/scripts/get-helm-3 | bash

## Install Driver

### Steps

* Run git clone -b v2.14.1 https://github.com/dell/csi-powerstore.git to clone the git repository.
* Ensure that you have created namespace where you want to install the driver. You can run kubectl create namespace csi-powerstore to create a new one. “csi-powerstore” is just an example. You can choose any name for the namespace. But make sure to align to the same namespace during the whole installation.
* Edit samples/secret/secret.yaml file and configure connection information for your PowerStore arrays changing following parameters:
* endpoint: defines the full URL path to the PowerStore API.
* globalID: specifies what storage cluster the driver should use
* username, password: defines credentials for connecting to array.
* skipCertificateValidation: defines if we should use insecure connection or not.
* isDefault: defines if we should treat the current array as a default.
* blockProtocol: defines what transport protocol we should use (FC, ISCSI, NVMeTCP, NVMeFC, None, or auto).
* nasName: defines what NAS should be used for NFS volumes.
* nfsAcls (Optional): defines permissions - POSIX mode bits or NFSv4 ACLs, to be set on NFS target mount directory. NFSv4 ACls are supported for NFSv4 shares on NFSv4 enabled NAS servers only. POSIX ACLs are not supported and only POSIX mode bits are supported for NFSv3 shares.

Note : skipCertificateValidation flag is currently unsupported. Please refrain from changing its value.

Add more blocks similar to above for each PowerStore array if necessary. If replication feature is enabled, ensure the secret includes all the PowerStore arrays involved in replication.

### User Privileges

The username specified in secret.yaml must be from the authentication providers of PowerStore. The user must have the correct user role to perform the actions. The minimum requirement is Storage Operator.

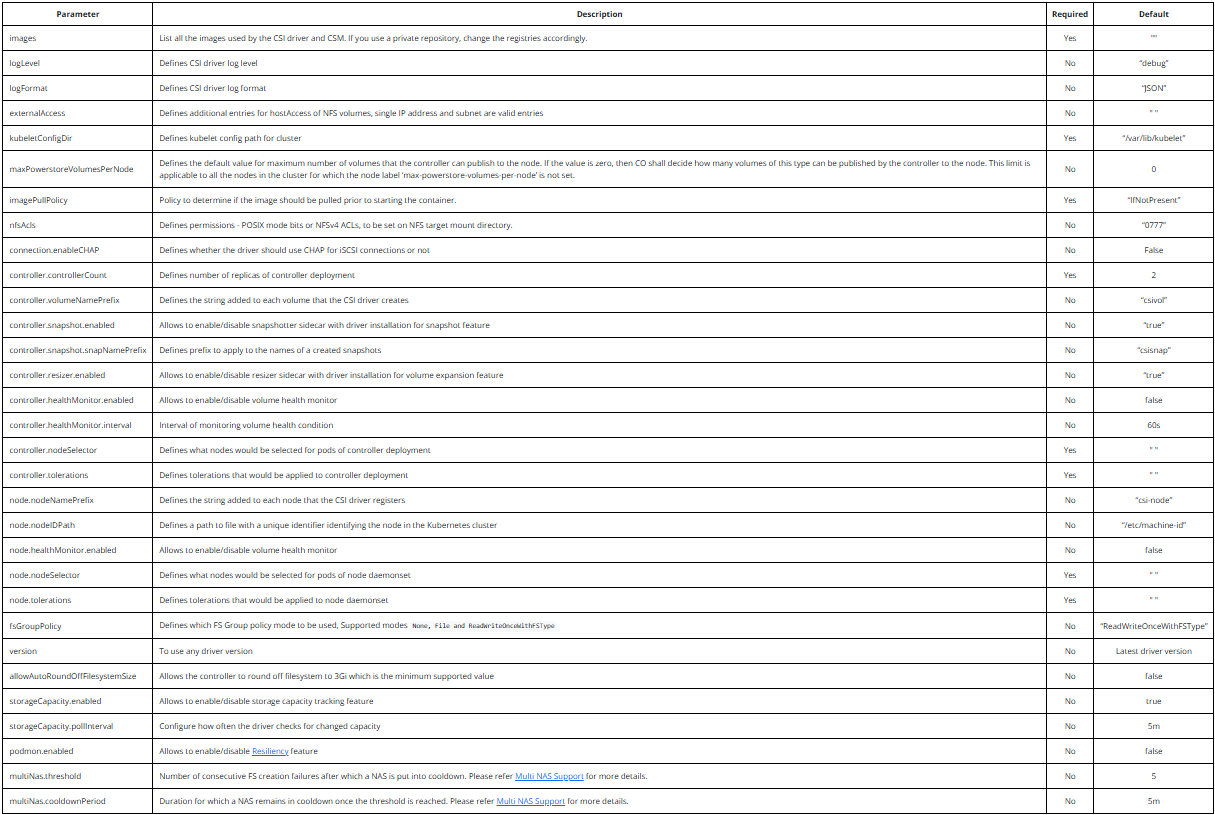
* Create the secret by running

kubectl create secret generic powerstore-config -n csi-powerstore --from-file=config=secret.yaml

* Create storage classes using ones from samples/storageclass folder as an example and apply them to the Kubernetes cluster by running kubectl create -f <path\_to\_storageclass\_file>
* If you do not specify arrayID parameter in the storage class then the array that was specified as the default would be used for provisioning volumes.
* Download the default values.yaml file

cd dell-csi-helm-installer && wget -O my-powerstore-settings.yaml https://github.com/dell/helm-charts/raw/csi-powerstore-2.14.1/charts/csi-powerstore/values.yaml

* Edit the newly created values file and provide values for the following parameters vi my-powerstore-settings.yaml:



* Install the driver using csi-install.sh bash script by running

./csi-install.sh --namespace csi-powerstore --values ./my-powerstore-settings.yaml --helm-charts-version <version>

After that the driver should be installed, you can check the condition of driver pods by running kubectl get all -n csi-powerstore

NOTE:

* The parameter --helm-charts-version is optional and if you do not specify the flag, by default the csi-install.sh script will clone the version of the helm chart that is specified in the driver’s csi-install.sh file. If you wish to install the driver using a different version of the helm chart, you need to include this flag. Also, remember to delete the helm-charts repository present in the csi-powerstore directory if it was cloned before.
* For detailed instructions on how to run the install scripts, refer to the readme document in the dell-csi-helm-installer folder.
* By default, the driver scans available SCSI adapters and tries to register them with the storage array under the SCSI hostname using node.nodeNamePrefix and the ID read from the file pointed to by node.nodeIDPath. If an adapter is already registered with the storage under a different hostname, the adapter is not used by the driver.
* A hostname the driver uses for registration of adapters is in the form <nodeNamePrefix>-<nodeID>-<nodeIP>. By default, these are csi-node and the machine ID read from the file /etc/machine-id.
* To customize the hostname, for example if you want to make them more user friendly, adjust nodeIDPath and nodeNamePrefix accordingly. For example, you can set nodeNamePrefix to k8s and nodeIDPath to /etc/hostname to produce names such as k8s-worker1-192.168.1.2.
* (Optional) Enable additional Mount Options - A user is able to specify additional mount options as needed for the driver.

Mount options are specified in storageclass yaml under mountOptions.

WARNING: Before utilizing mount options, you must first be fully aware of the potential impact and understand your environment’s requirements for the specified option.

### Storage Classes

The CSI driver for PowerStore version 1.3 and later, dell-csi-helm-installer does not create any storage classes as part of the driver installation. A wide set of annotated storage class manifests have been provided in the samples/storageclass folder. Use these samples to create new storage classes to provision storage.

#### What happens to my existing storage classes?

Upgrading from an older version of the driver: The storage classes will be deleted if you upgrade the driver. If you wish to continue using those storage classes, you can patch them and apply the annotation “helm.sh/resource-policy”: keep before performing an upgrade.

Note: If you continue to use the old storage classes, you may not be able to take advantage of any new storage class parameter supported by the driver.

#### Steps to create storage class:

There are samples storage class yaml files available under samples/storageclass. These can be copied and modified as needed.

Edit the sample storage class yaml file and update following parameters:

* arrayID: specifies what storage cluster the driver should use, if not specified driver will use storage cluster specified as default in samples/secret/secret.yaml
* csi.storage.k8s.io/fstype: specifies what filesystem type driver should use, possible variants ext3, ext4, xfs, nfs, if not specified driver will use ext4 by default.
* nfsAcls (Optional): defines permissions - POSIX mode bits or NFSv4 ACLs, to be set on NFS target mount directory.
* allowedTopologies (Optional): If you want you can also add topology constraints.

allowedTopologies:

- matchLabelExpressions:

- key: csi-powerstore.dellemc.com/12.34.56.78-iscsi

# replace "-iscsi" with "-fc", "-nvmetcp" or "-nvmefc" or "-nfs" at the end to use FC, NVMeTCP, NVMeFC or NFS enabled hosts

# replace "12.34.56.78" with PowerStore endpoint IP

values:

- "true"

Create your storage class by using kubectl:

kubectl create -f <path\_to\_storageclass\_file>

NOTE: Deleting a storage class has no impact on a running Pod with mounted PVCs. You cannot provision new PVCs until at least one storage class is newly created.