CSI Driver for Dell EMC PowerStore

Version 1.0

Product Guide

May 2020

Dell EMC
Hopkinton, Massachusetts 01748-9103
1-508-435-1000 In North America 1-866-464-7381
www.DellEMC.com

CONTENTS

CHAPTER 1

Introduction

This chapter includes the following topics:

Product overview

The CSI Driver for Dell EMC PowerStore is a plug in that is installed into Kubernetes to provide storage provisioning for Dell EMC PowerStore storage arrays.

The CSI Driver for Dell EMC PowerStore and Kubernetes communicate using the Container Storage Interface protocol. The CSI Driver for Dell EMC PowerStore conforms to CSI specification v1.1.

Features of CSI Driver for Dell EMC PowerStore

The CSI Driver for Dell EMC PowerStore has the following features:

- Supports CSI 1.1
- Supports Kubernetes version 1.14 and 1.16
- Supports Fibre Channel
- Supports iSCSI
- Supports Linux native multipathing
- Supports CentOS versions 7.3, 7.5, and 7.6 as host operating system
- Supports Red Hat Enterprise Linux 7.6 as host operating system
- Supports Ubuntu 18.04 as host operating system
- Automatic Kubernetes version detection
- Dynamic and Static PV provisioning
- Helm 3 charts installer
- Persistent volume (PV) capabilities - create and delete
- Volume and host prefixes for easier identification in PowerStore Manager
- Volume mount as ext4 or xfs file system on the worker node

CHAPTER 2

Installation

This chapter includes the following topics:

Installation overview

The Helm chart installs the CSI Driver for Dell EMC PowerStore using a shell script. This script installs the CSI Driver container image along with the required Kubernetes sidecar containers.

The CSI Driver installation repository includes Helm charts that use a shell script to deploy the CSI Driver for Dell EMC PowerStore. The shell script installs the CSI Driver for Dell EMC PowerStore container image along with the required Kubernetes sidecar containers.

The controller section of the Helm chart installs the following components in a StatefulSet in the csi-powerstore namespace:

- CSI Driver for Dell EMC PowerStore
- Kubernetes Provisioner
- Kubernetes Attacher

The node section of the Helm chart installs the following components in a DaemonSet in the csi-powerstore namespace:

- CSI Driver for Dell EMC PowerStore
- Kubernetes Registrar

Prerequisites

This topic lists the prerequisites to install the CSI Driver for Dell EMC PowerStore.

Before you install the CSI Driver for Dell EMC PowerStore, you must complete the following tasks:

- Install Kubernetes
  The CSI Driver for Dell EMC PowerStore works with Kubernetes versions 1.14 and 1.16.
- Refer to either Fibre Channel requirements or Set up the iSCSI Initiator
  ⓘ **Note:** You can use either the Fibre Channel or iSCSI protocol, but you do not need both.
- Configure Docker service
- Install Helm 3.0
- Linux native multipathing requirements

Fibre Channel requirements

Dell EMC PowerStore supports Fibre Channel communication. If you will use the Fibre Channel protocol, ensure that the following requirement is met before you install the CSI Driver for Dell EMC PowerStore:

- Zoning of the Host Bus Adapters (HBAs) to the Fibre Channel port director must be done.

Set up the iSCSI Initiator

The CSI Driver for Dell EMC PowerStore v1.0 supports iSCSI connectivity.

If you will use the iSCSI protocol, set up the iSCSI initiators as follows:

- Make sure that the iSCSI initiators are available on both Master and Minions nodes.
- Kubernetes nodes should have access (network connectivity) to an iSCSI director on the Dell EMC PowerStore array that has IP interfaces. Manually create IP routes for each node that connects to the Dell EMC PowerStore.

- All Kubernetes nodes must have the *iscsi-initiator-utils* package installed, and the *iscsid* service must be enabled and running. To do this, run the `systemctl enable iscsid && systemctl start iscsid` command.
- Make sure that the iSCSI initiators on the nodes are not a part of any existing Host (Initiator Group) on the Dell EMC PowerStore.
- The CSI driver needs the port group names containing the required iSCSI director ports. These Port Groups must be set up on each Dell EMC PowerStore array. All the port groups names supplied to the driver must exist on each Dell EMC PowerStore with the same name.

For information about configuring iSCSI, see Dell EMC PowerStore documentation on Dell EMC Support.

Configure Docker service

This topic gives the procedure to configure docker service. Configure the mount propagation in Docker on all Kubernetes nodes before installing the CSI Driver for Dell EMC PowerStore.

Procedure

1. Edit the service section of */etc/systemd/system/multi-user.target.wants/docker.service* file to add the following lines:

```
docker.service
[Service]...
MountFlags=shared
```

2. Restart the docker service with *systemctl daemon-reload* and *systemctl restart docker* on all the nodes.

Install Helm 3.0

Install Helm 3.0 on the master node before you install the CSI Driver for Dell EMC PowerStore.

Procedure

1. Run the `curl https://raw.githubusercontent.com/helm/helm/master/scripts/get-helm-3 | bash` command to install Helm 3.0.

Linux multipathing requirements

Dell EMC PowerStore supports Linux multipathing. Configure Linux multipathing before installing the CSI Driver for Dell EMC PowerStore.

Set up Linux multipathing as follows:

- All the nodes must have *Device Mapper Multipathing* package installed.

  > ⓘ **Note:** This package is installed by default and creates a multipath configuration file. This file is located in `/etc/multipath.conf`.

- Enable multipathing using the `mpathconf --enable --with_multipathd y` command.
- Enable `user_friendly_names` and `find_multipaths` in the *multipath.conf* file.
- Ensure that the `multipath` command for *multipath.conf* is available on all Kubernetes nodes.

Install CSI Driver for Dell EMC PowerStore

Install the CSI Driver for Dell EMC PowerStore using this procedure.

Before you begin

Ensure that you meet the following prerequisites before you install the CSI Driver for Dell EMC PowerStore:

- You have the downloaded files ready for this procedure.
- You have the Helm chart from the Dell EMC GitHub repository, ready for this procedure.
- The top-level helm directory contains the *install* and *uninstall* shell scripts. The scripts perform certain preinstallation and postinstallation operations (like creating Custom Resource Definitions), which cannot be performed in the helm chart.
- The iSCSI initiators are available on all nodes, including the master and minion nodes.
- The Kubernetes feature gates are enabled.
- The mount propagations are configured in Docker.
- The nonsecure registries are defined in Docker, for CSI drivers that are hosted in a nonsecure location.

Procedure

1. Run `git clone https://github.com/dell/csi-powerstore.git` to clone the git repository to the master node of the Kubernetes cluster.
2. Run `cd csi-powerstore/helm && cp csi-powerstore/values.yaml ./myvalues.yaml` to change the directory to the top-level helm directory and copy the values file for driver configuration.
3. Provide values in the *myvalues.yaml* file for the following parameters:
    - `powerStoreApi`: This value defines the full URL path to the PowerStore API.
    - `powerStoreApiUser`: This value defines the username to log in to the PowerStore API.
    - `powerStoreApiPassword`: This value defines the password to log in to the PowerStore API.
    - `nodeIDPath`: This parameter defines a path to file with unique identifier identifying the node in the Kubernetes cluster.
    - `volumeNamePrefix`: This parameter defines the string added to each volume that the CSI driver creates.
    - `nodeNamePrefix`: This parameter defines the string added to each node that the CSI driver registers.
    - `transportProtocol`: This parameter defines which transport protocol to use (FC, ISCSI, or auto).

    By default, the driver scans available SCSI adapters and tries to register them with the storage array under the SCSI host name using `nodeNamePrefix` and the ID read from the file pointed to by `nodeIDPath`. If an adapter is already registered with the storage under a different host name, the adapter is not used by the driver.

    A host name the driver uses for registration of adapters is in the form `<nodeNamePrefix>-<nodeID>`. By default, these are `csi-node` and the machine ID read from the file `/etc/machine-id`. To customize the host name, 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-192.168.1.2`.

4. Run `install.sh` to install the driver.

   This script also runs the *verify.sh* script that is present in the same directory. You are prompted to enter the credentials for each of the Kubernetes nodes. The *verify.sh* script needs the credentials to check if the *kubelet* is configured with the appropriate feature gates on each of the Kubernetes nodes.

Results

The CSI Driver for Dell EMC PowerStore is installed.

CSI Driver Usage

Once you install the driver, it creates a default storage class, csi-powerstore, using parameters from *myvalues.yaml* file. You can configure it to your liking or create custom storage classes, which are using the driver as a provisioner.

You can create Persistent Volumes (PV) and PersistentVolumeClaims (PVC) using these storage classes. After that, use the PVC name in Kubernetes manifests, where you can specify which containers can use certain volumes and where they should be mounted. For a more detailed explanation of the pod configuration, see the official Kubernetes documentation.

Controller Plug-in query commands

This topic lists the commands to view the details of StatefulSet and check logs for Controller plug-in.

Procedure

1. Query details of the StatefulSet.

   ```
   kubectl get statefulset -n csi-powerstore
   ```

   ```
   kubectl describe statefulset csi-powerstore-controller -n csi-powerstore
   ```

2. Check the logs of the Controller.

   ```
   kubectl logs csi-powerstore-controller-0 driver -n csi-powerstore
   ```

Node Plug-in query commands

This topic lists the commands to view the details of DaemonSet and check logs for the Node plug-in.

Procedure

1. Query details of DaemonSet.

   ```
   kubectl get daemonset -n csi-powerstore
   ```

   ```
   kubectl describe daemonset csi-powerstore-node -n csi-powerstore
   ```

2. Check logs of the Node.

   ```
   kubectl logs csi-powerstore-node-<suffix> driver -n csi-powerstore
   ```

CHAPTER 3

Test

This chapter includes the following topics:

CSI Driver Testing

In the repository, a simple test manifest exists that creates two different PersistentVolumeClaims using default ext4 and xfs storage classes, and automatically mounts them to the pod.

About this task

Procedure

1. To run this test, run the `kubectl` command from the `root` directory of the repository:

   ```
   kubectl create -f ./tests/simple/
   ```

   You can find all the created resources in `testpowerstore` namespace.

2. Check if the pod is created and Ready and Running by running:

   ```
   kubectl get all -n testpowerstore
   ```

   If it's in `CrashLoopback` state then the driver installation wasn't successful. Check the logs of the node and the controller.

3. Go into the created container and verify that everything is mounted correctly.
4. After verifying, you can uninstall the testing PVCs and StatefulSet.

   ```
   kubectl delete -f ./tests/simple/
   ```