Hedvig CSI User Guide

Table of Contents

Prerequisites

- Hedvig Storage Cluster (Hedvig 4.0.0 or later)
- Kubernetes/OpenShift/D2IQ cluster:
  - Kubernetes version 1.13+
  - OpenShift version 4.1+

    Note: For prerequisites specific to OpenShift, see *Appendix A: OpenShift prerequisites*.
  - D2IQ Konvoy

    Note: For more information about setting up D2IQ Konvoy with Hedvig:

    https://docs.d2iq.com/ksphere/konvoy/partner-solutions/hedvig/
  - If a firewall is enabled on the Kubernetes/OpenShift/D2IQ nodes, for the list of ports that need to be unblocked, see *Appendix B: List of ports.*
- On the Kubernetes/OpenShift/D2IQ nodes:
  - Install `iscsi-initiator-utils` and `systemd`.
  - Ensure that the `rpcbind` process is running.
  - Ensure that the `iscsi` kernel modules are loaded. To list these modules, run:

    ```
    lsmod | grep iscsi
    ```
- Ensure that the Kubernetes/OpenShift/D2IQ nodes and the Hedvig Storage Cluster Nodes can communicate with each other.

Note:

- If you are using Kubernetes/D2IQ, the Hedvig services will be installed on the Kubernetes/D2IQ worker nodes.
- If you are using OpenShift, the Hedvig services will be installed (by default) on the OpenShift compute nodes.

Compatibility matrix

Hedvig version	Hedvig CSI Driver version
Hedvig 4.0.0	**1.0.0**

Setup instructions for the Hedvig Storage Proxies

Note: For OpenShift installations, if you plan to run the Hedvig Storage Proxies on **infra nodes**, perform the following actions on OpenShift **infra nodes**, as well.

IMPORTANT: Recommended settings for the Hedvig Storage Proxy are 4vCPUs, 8GB RAM.

Setting up caches on all Kubernetes/OpenShift/D2IQ nodes

1. Create the file `/etc/systemd/system/metacache.service`

```
[Unit]
Description=Setup Metacache
After=network.target tgt.service hedvigfsc.service

[Service]
Type=oneshot
ExecStart=/bin/bash -c "/bin/mount -t tmpfs -o size=4g tmpfs /hedvig/cache"
ExecStartPre=/bin/bash -c "/bin/mkdir -p /hedvig/cache"
RemainAfterExit=true
ExecStop=/bin/true
StandardOutput=journal

[Install]
WantedBy=multi-user.target
```

2. Run the following commands to set up the cache:

```
systemctl enable metacache
systemctl start metacache
```

3. Verify that `/hedvig/cache` is present by running:

```
df -kh
```

Installing the DaemonSet on all Kubernetes/OpenShift/D2IQ nodes

1. On any Linux client machine where `kubectl` is installed, download the Hedvig CSI Driver package:

   `hedvig-csi-installer.tar`

2. Untar the downloaded package, and change to the install directory:

   ```
   cd hedvig-csi-installer
   ```

3. Locate the manifest for the DaemonSet under:

   `manifests/ds/daemonset.yml.tmpl`

4. Copy this file into `daemonset.yml` and update the values for the following fields:

   - **`HEDVIG_SEED_1, HEDVIG_SEED_2 and HEDVIG_SEED_3`** — Hostnames of any three Hedvig Storage Cluster Nodes in the Hedvig Storage Cluster
   - **`KUBE_CLUSTER_HEDVIG_ID`** — Unique id for the Kubernetes/OpenShift/D2IQ cluster

5. Update the image name in `daemonset.yaml` to `hedviginc/hedvigblock:<tag>` and set the `<tag>` to the software version installed on the Hedvig Storage Cluster.

   A complete list of available versions can be found here:

   `https://hub.docker.com/r/hedviginc/hedvigblock/tags/`

6. Deploy the DaemonSet by running:

   ```
   kubectl create -f daemonset.yml
   ```

7. In the Hedvig WebUI, verify that the Hedvig Storage Proxies are up and running.

   In the Hedvig CLI, run the following command:

   ```
   hedvig> lskubecontrollers -i <KUBE_CLUSTER_HEDVIG_ID> -t block
   ```

   This command should list all Kubernetes/OpenShift/D2IQ nodes.

Setup instructions for the Hedvig CSI Driver

Setting up the Hedvig CSI Driver

1. Navigate to the `root` directory of the package.

2. Update the following configuration values in the `setup/backend.json` file to point to your Hedvig Storage Cluster:

   - **`StorageCluster`** — Name of the Hedvig Storage Cluster
   - **`StorageNodes`** — Comma-separated list of hostnames/IP addresses of the Hedvig Storage Cluster Nodes
   - **`KubeClusterID`** — Unique id for the Kubernetes/OpenShift/D2IQ cluster (should be same as the value for `KUBE_CLUSTER_HEDVIG_ID` used in the `daemonset.yml` configuration)

     Note: See *Installing the DaemonSet on all Kubernetes/OpenShift/D2IQ nodes.*

3. Update the image name in both `hedvig-csideployment.yaml` and `hedvig-csidaemonset.yaml` to `hedviginc/hedvig-csi-volume-driver:<tag>` and set the `<tag>` to the most recently released version of the Hedvig CSI Driver.

   A complete list of available versions can be found here:

   `https://hub.docker.com/r/hedviginc/hedvig-csi-volume-driver/tags/`

Installing the Hedvig CSI Driver

1. Navigate to the `root` directory of the package.

2. To install the Hedvig CSI Driver:

   `./install_hedvig.sh`

   Sample templates for Storage Classes and applications can be found here:

   `./manifests`

Post-installation verification

1. Verify that the storage proxy pods (deployed as a part of the DaemonSet) and the Hedvig CSI Driver pods are in `Running` state.

   Run the following command to verify the installation:

   ```
   kubectl get pods -n kube-system
   ```

   If the installation is successful, the output should look like this:

   ```
   kubectl get pods -n kube-system
   NAME                                      READY    STATUS    RESTARTS   AGE
   hedvig-csi-controller-f69cf7f65-hxhlp     4/4      Running   0          68s
   hedvig-csi-node-cjd4q                     2/2      Running   0          68s
   hedvig-csi-node-f82km                     2/2      Running   0          68s
   hedvig-csi-node-nrclg                     2/2      Running   0          68s
   ```

2. On the Hedvig CLI, run the following commands:

   ```
   lsallkubeclusters
   ```

   This command should list the KubeClusterID. This verifies that the Hedvig CSI Driver pods have successfully registered with the Hedvig Storage Cluster.

   ```
   lskubecontrollers -i <KubeClusterID> -t block
   ```

   This command should list all Kubernetes/OpenShift/D2IQ nodes, along with their respective IQNs. This verifies that the storage proxy pods have successfully registered with the Hedvig Storage Cluster.

   ```
   getallkubedata -i <KubeClusterID> -t KubeBackend
   ```

   This command should list the available backends for the given Kubernetes/OpenShift/D2IQ cluster. By default, `hedvig-block` should be available.

Storage Classes and Hedvig virtual disk attributes

Creating a Default Storage Class for Persistent Volumes (PVs)

The following manifests create a Storage Class for Persistent Volumes (PVs) backed by Hedvig virtual disks of type Block, depending on the replication policy of the cluster:

Cluster with Agnostic (default) replication policy

If the cluster is set up with the **Agnostic** replication policy (the default), then the default storage class should be created using the following manifest:

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: sc-hedvig-block
provisioner: io.hedvig.csi
parameters:
  backendType: "hedvig-block"
  rp: "Agnostic"
```

Cluster with RackAware replication policy

If the cluster is set up across multiple racks (**RackAware** replication policy), then the default storage class should be created using the following manifest:

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
    name: sc-hedvig-block
provisioner: io.hedvig.csi
parameters:
    backendType: "hedvig-block"
    rp: "RackAware"
```

Cluster with DataCenterAware replication policy

If the cluster is set up across multiple data centers (**DataCenterAware** replication policy), then the default storage class should be created using the following manifest:

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
    name: sc-hedvig-block
provisioner: io.hedvig.csi
parameters:
    backendType: "hedvig-block"
    rp: "DataCenterAware"
    dcNames: "<comma-separated-list-of-datacenter-names>"
```

Customizing Storage Classes with Hedvig virtual disk attributes

Storage Classes can be customized by specifying Hedvig virtual disk attributes in the `parameters` section, as shown below:

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: sc-hedvig-block-compressed
provisioner: io.hedvig.csi
parameters:
  backendType: "hedvig-block"
  compressed: "true"
```

Note: See the next section, *Hedvig virtual disk attribute keys*, for a table of the keys and their possible values.

Hedvig virtual disk attribute keys

key	values	default value	notes
dedupEnable	true/false	false	
compressed	true/false	false	
cacheEnable	true/false	false	
rf (replication factor)	1 to 6	3	
rp (replication policy)	Agnostic RackAware DataCenterAware	Agnostic	
dcNames	comma-separated list of data center names		This applies only to a replication policy (**rp**) of **DataCenterAware**.
diskResidence	flash/hdd	hdd	In an all-flash cluster, **diskResidence** should always be set to **flash**.
encryptionEnable	true/false	false	
blockSize	512/4096	4096	
description	any string	""	

Appendix A: OpenShift prerequisites

- Ensure that the following SCC (SecurityContextConstraints) exist:
  - **hostnetwork**
  - **privileged**

- Create a new SCC configuration called **hostpath** with the following configuration parameters, and save it as **hostpath.yaml**

```
kind: SecurityContextConstraints
apiVersion: v1
metadata:
  name: hostpath
allowPrivilegedContainer: true
runAsUser:
  type: RunAsAny
seLinuxContext:
  type: RunAsAny
fsGroup:
  type: RunAsAny
supplementalGroups:
  type: RunAsAny
users:
- my-admin-user
groups:
- my-admin-group
```

- Create the new SCC **hostpath** by running the command:

```
oc create -f hostpath.yaml
```

- Additional configurations on SCC **hostpath**:

```
oc patch scc hostpath -p '{"allowHostDirVolumePlugin": true}'

oc adm policy add-scc-to-group hostpath system:authenticated

oc adm policy add-scc-to-group hostpath system:admin
```

- Ensure that the following settings are enabled on **hostnetwork**, **privileged**, and SCC **hostpath**:

```
allowHostDirVolumePlugin: true

allowHostNetwork: true

allowPrivilegedContainer: true
```

Appendix B: List of ports

protocol	port range	description
TCP	50022	ssh
TCP	3260	iscsi
TCP	50000 – 50008	thrift