Hedvig Operator User Guide

Table of Contents

Introduction

Hedvig Operator is the official operator to deploy and manage Hedvig components in container orchestrators such as Kubernetes.

Hedvig Operator deploys the Hedvig Storage Proxy and the Hedvig CSI Driver to manage the life cycle of persistent storage.

Supported Container Orchestrators

- Kubernetes 1.13 to 1.20
- Red Hat OpenShift 4.1 to 4.8

Supported Hedvig Releases

- Hedvig 4.0.x
- Hedvig 4.1.x
- Hedvig 4.2.x
- Hedvig 4.3.x
- Hedvig 4.4.x
- Hedvig 4.5.x

Required Network Setup

- If a firewall is enabled on the Kubernetes/OpenShift nodes, unblock the ports in the following table.

Table 1: Ports to be unblocked if firewall enabled

Protocol	Port range	Description
TCP	50022	ssh
TCP	2049	nfs
TCP	33333	nfs (mountd)
TCP	3260	iscsi
TCP	50000 - 50008	thrift
TCP	15000	thrift
TCP	8000	thrift

- Make sure that the Kubernetes/OpenShift nodes and the Hedvig Storage Cluster Nodes can communicate with each other.

OpenShift Prerequisites

- Make sure that the following SCCs (security context constraints) exist:
  - `hostnetwork`
  - `privileged`

- Make sure that the following capabilities are enabled on the aforementioned SCCs:
  - Allow Host Dir Volume Plugin: true
  - Allow Host Network: true
  - Allow Privileged Container: true

- In the namespace where you plan to install Hedvig components, create service accounts **hedvig-csi** and **hedvig-operator**.

  Add the aforementioned SCCs to the **default**, **hedvig-csi** and **hedvig-operator** service accounts using the following command:

  ```
  oc adm policy add-scc-to-user <scc-name>
  ```

  ```
  system:serviceaccount:<namespace>:<service-account-name>
  ```

Host Prerequisites

The location for installed Hedvig services depends on whether you are using Kubernetes or OpenShift:

- For Kubernetes, Hedvig services are installed on Kubernetes worker nodes.
- For OpenShift, Hedvig services are installed (by default) on OpenShift compute nodes.

Prerequisites must be met on these specific nodes, that is, worker nodes for Kubernetes, compute nodes for OpenShift.

Hedvig Storage Proxy (Block) Prerequisites

1. Install iSCSI initiator utilities (`iscsi-initiator-utils/open-iscsi`).
2. Make sure that the following processes are enabled and running:
   - `rpcbind`
   - `iscsid`
3. Make sure that the `iscsi` kernel modules are loaded. To list these modules, run:

```
lsmod | grep iscsi

scsi_tcp               18333  2
libiscsi_tcp           25146  1 iscsi_tcp
libiscsi               57233  2 libiscsi_tcp,iscsi_tcp
scsi_transport_iscsi   99909  3 iscsi_tcp,libiscsi
```

Multipath Configuration

Multipath is **not recommended** for iSCSI persistent volumes with Hedvig. If multipath is enabled by default on Kubernetes/OpenShift compute nodes, a blacklist must be created for Hedvig volumes using the vendor id as shown below:

```
# cat /etc/multipath.conf
blacklist {
       device {
               vendor  "_HEDVIG_"
       }
}
```

Hedvig Storage Proxy (NFS) Prerequisites

1. Install NFS utilities (`nfs-utils/nfs-common`).
2. Make sure that the `rpcbind` process is running.
3. Make sure that the `messagebus/dbus` process is running.
4. Create the `/etc/dbus-1/system.d/org.ganesha.nfsd.conf` file with the following contents, and restart `messagebus/dbus`.

```
<?xml version="1.0" encoding="UTF-8"?> <!-- -*- XML -*- -->

<!DOCTYPE busconfig PUBLIC
"-//freedesktop//DTD D-BUS Bus Configuration 1.0//EN"
"http://www.freedesktop.org/standards/dbus/1.0/busconfig.dtd">
<busconfig>
  <!-- Only root can own the service -->
  <policy user="root">
    <allow own="org.ganesha.nfsd"/>
    <allow send_destination="org.ganesha.nfsd"/>
  </policy>
  <policy context="default">
    <deny own="org.ganesha.nfsd"/>

    <allow send_destination="org.ganesha.nfsd"
           send_interface="org.freedesktop.DBus.Introspectable"/>

    <allow send_destination="org.ganesha.nfsd"
           send_interface="org.ganesha.nfsd.CBSIM"/>

    <allow send_destination="org.ganesha.nfsd"
           send_interface="org.ganesha.nfsd.admin"/>

    <allow send_destination="org.ganesha.nfsd"
           send_interface="org.ganesha.nfsd.ExportMgr"/>
  </policy>
</busconfig>
```

5. *If SELinux is enabled,* set the appropriate security context for the file. You can verify the security context by running the `ls -lZ` command.

```
$ ls -lZ
Total 36
-rw-r--r--. 1 root root system_u:object_r:dbusd_etc_t:s0  475 Apr 12
22:00 dnsmasq.conf

-rw-r--r--. 1 root root system_u:object_r:dbusd_etc_t:s0  491 Apr 12
22:00 nm-dispatcher.conf

-rw-r--r--. 1 root root system_u:object_r:dbusd_etc_t:s0  354 Apr 12
22:00 nm-ifcfg-rh.conf

-rw-r--r--. 1 root root system_u:object_r:dbusd_etc_t:s0 8150 Apr 12
22:00 org.freedesktop.NetworkManager.conf

-rw-r--r--. 1 root root system_u:object_r:dbusd_etc_t:s0  638 Apr 12
22:00 org.freedesktop.PolicyKit1.conf

-rw-r--r--. 1 root root system_u:object_r:dbusd_etc_t:s0  890 Apr 16
17:44 org.ganesha.nfsd.conf

-rw-r--r--. 1 root root system_u:object_r:dbusd_etc_t:s0  535 Apr 12
22:00 org.selinux.conf

-rw-r--r--. 1 root root system_u:object_r:dbusd_etc_t:s0  409 Apr 12
22:00 teamd.conf
```

Hedvig Storage Proxy Caching Prerequisites

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.

Setting up the metacache on tmpfs

The following steps describe the metacache setup on `tmpfs` mounted on host path **`/hedvig/cache`**. If you choose to change the host path, update the `metacache.service` file accordingly.

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

```
[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
```

Image Repositories

The following table provides the image repository locations for the Hedvig Storage Proxies (Block and NFS), the Hedvig CSI Driver, and the Hedvig Operator.

Table 2: Hedvig Storage Proxy, Hedvig CSI Driver, and Hedvig Operator Image Repositories

Hedvig Storage Proxy	*https://hub.docker.com/r/hedvig/hedvig-proxy/tags*
Hedvig CSI Driver	*https://hub.docker.com/r/hedvig/hedvig-csi-volume-driver/tags*
Hedvig Operator	*https://hub.docker.com/r/hedvig/hedvig-operator/tags*

Install Hedvig Operator using kubectl/oc

To install the Hedvig Operator using kubectl/oc:

Download Hedvig Operator Configuration Files

1. Download the Hedvig Operator from the Commvault Store:

   https://cloud.commvault.com/webconsole/softwarestore/#!/

2. Copy it to any machine where `kubectl/oc` is installed.

3. Extract the downloaded package, and verify that it has the following files:
   - `crds/hedvig.io_hedvigdeploys.yaml`
   - `service_account.yaml`
   - `role_binding.yaml`
   - `role.yaml`
   - `operator.yaml`

Install the Hedvig Operator

1. Create a CRD (custom resources definition).

   ```
   kubectl create -f crds/hedvig.io_hedvigdeploys.yaml
   ```

   Create a namespace for the operator (and other Hedvig components).

   ```
   kubectl create ns <namespace>
   ```

2. Create a service account.

   Update `service_account.yaml` with the namespace, and create the service account.

   ```
   kubectl create -f service_account.yaml
   ```

3. Create a role.

   ```
   kubectl create -f role.yaml
   ```

4. Create a role binding.

   Update `role_binding.yaml` with the namespace, and create the role binding.

   ```
   kubectl create -f role_binding.yaml
   ```

5. Create an operator.

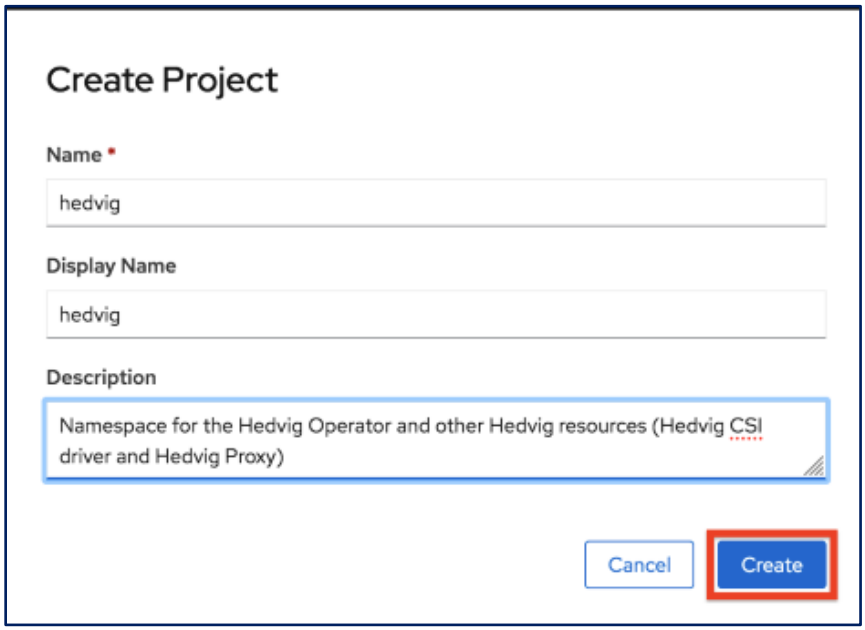
   Update `operator.yaml` with the namespace, and create the operator.

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

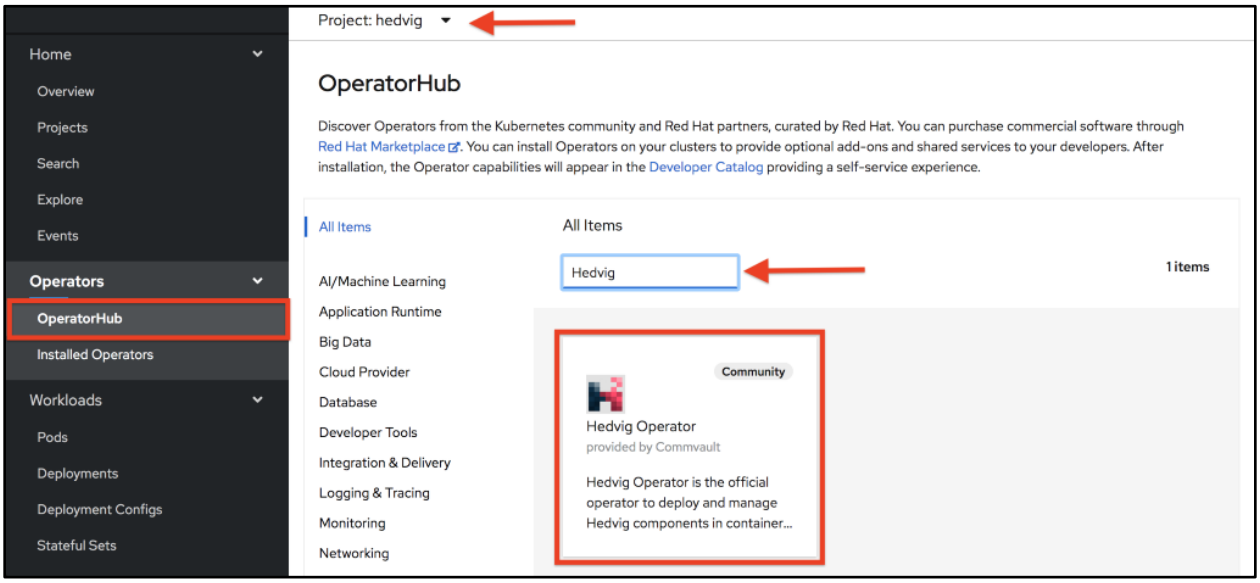
Install Hedvig Operator using OperatorHub

Hedvig Operator can be installed using OperatorHub if you are running OpenShift 4.x as your container orchestrator.

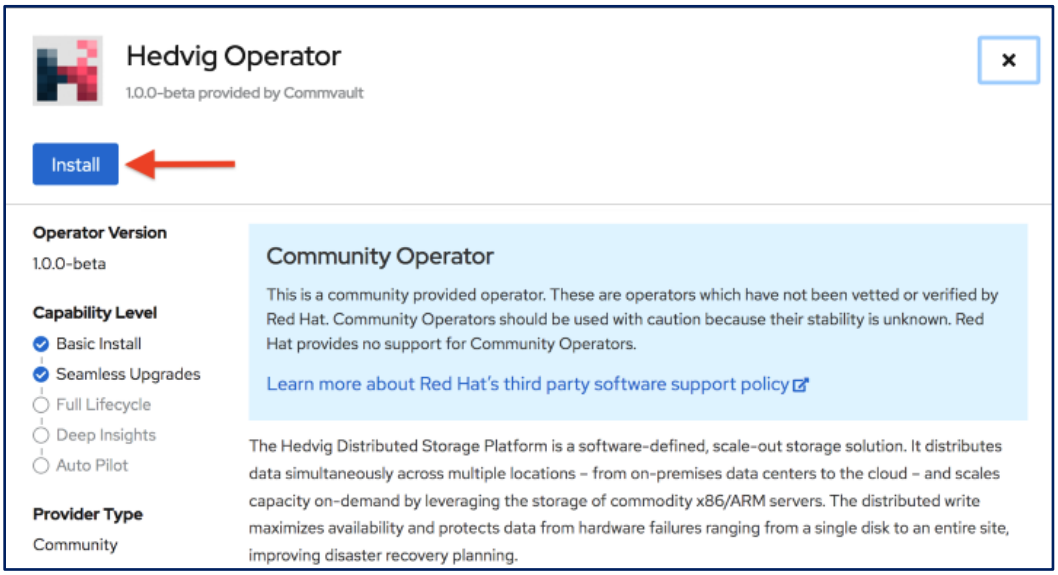
1. Create a namespace for the operator.



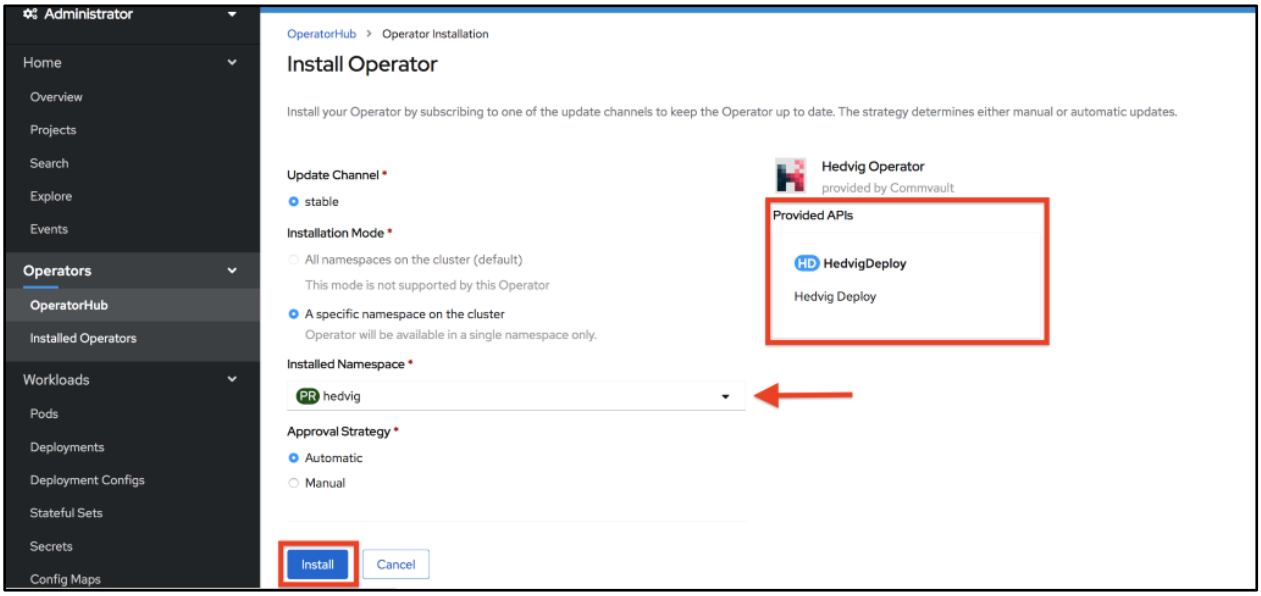
2. Navigate to the **OperatorHub** tab on the OpenShift cluster admin page, and search for **Hedvig**.



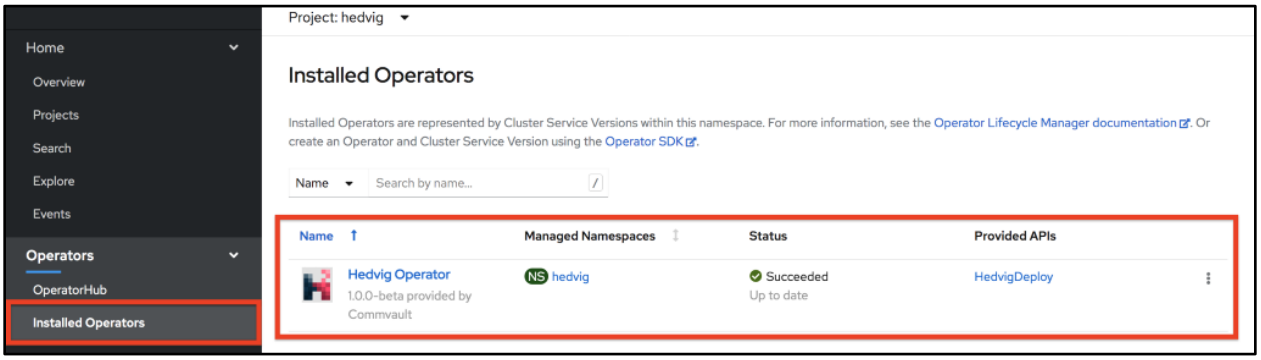
3. Click on the Hedvig Operator to open the install page, and then click on **Install**.

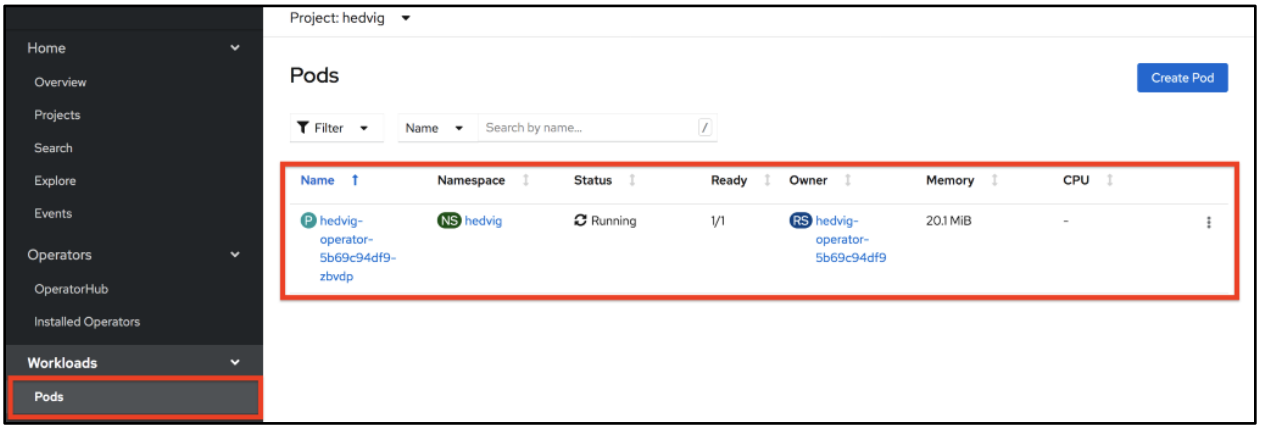


4. Verify the namespace for the Hedvig Operator and click on **Install**. A new CustomResourceDefinition called **HedvigDeploy** will also be created, which will be used to configure Hedvig components.



5. Hedvig Operator should now be deployed in the configured namespace.





Create the Hedvig Deployment Configuration

Create a `HedvigDeploy` resource (`hedvig-deploy.yaml`) with the following information. The numbers refer to descriptions in *Table 3: Hedvig Deployment Configuration Resource Definitions.*

```
apiVersion: hedvig.io/v1
kind: HedvigDeploy
metadata:
  name: <deployment-name>  1
  namespace: <namespace>  2
spec:
  cluster:
    name: <hedvig-cluster-name>  3
    seeds:
    - <hedvig-cluster-seed-1>  4
    - <hedvig-cluster-seed-2>  5
    - <hedvig-cluster-seed-3>  6
  k8s:
   name: <k8s-cluster-name>  7
  proxy:
    repository: hedviginc/hedvigblock  8
    tag: <image-tag>  9
    imagePullPolicy: IfNotPresent
    block: true/false
    nfs: true/false
  controller:
    repository: hedvig/hedvig-csi-volume-driver  10
    tag: <image-tag>  11
    imagePullPolicy: IfNotPresent
  node:
    repository: hedvig/hedvig-csi-volume-driver  12
    tag: <image-tag>  13
    imagePullPolicy: IfNotPresent
  sidecars:  14
  - name: csi-provisioner
    repository: quay.io/k8scsi/csi-provisioner
    tag: "v1.6.0"
    imagePullPolicy: IfNotPresent
  - name: csi-snapshotter
    repository: quay.io/k8scsi/csi-snapshotter
    tag: "v2.1.0"
    imagePullPolicy: IfNotPresent
  - name: csi-resizer
    repository: quay.io/k8scsi/csi-resizer
    tag: "v0.5.0"
    imagePullPolicy: IfNotPresent
  - name: csi-node-driver-registrar
    repository: quay.io/k8scsi/csi-node-driver-registrar
    tag: "v1.0-canary"
    imagePullPolicy: IfNotPresent
```

Table 3: Hedvig Deployment Configuration Resource Definitions

Number	Description
1	Deployment name
2	Namespace for deploying Hedvig components (Storage Proxy and CSI Driver)
3	Hedvig Cluster name
4 5 6	Hostnames of any three Hedvig Storage Cluster Nodes in the Hedvig Storage Cluster
7	Kubernetes/OpenShift Cluster name
8 9	Hedvig Storage Proxy repository and image tag corresponding to the software version installed on the Hedvig Storage Cluster
10 11	Hedvig CSI repository and image tag corresponding to the CSI driver version
12 13	Hedvig CSI repository and image tag corresponding to the CSI driver version
14	Hedvig CSI sidecar container repositories have been set to their default locations

Install the HedvigDeploy Resource using kubectl/oc

1. Create the `secret` for the Hedvig Storage Cluster.

   ```
   kubectl create secret generic hedvig-cluster-credentials --from-literal=username='<username>' --from-literal=password='<password>' -n <namespace>
   ```

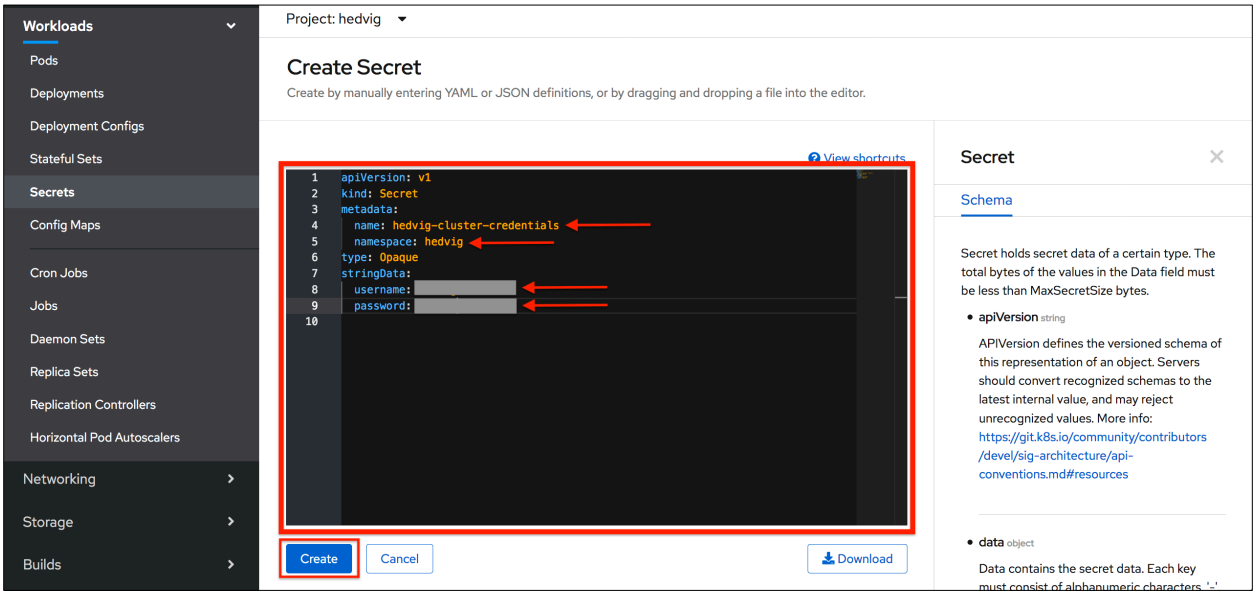
   <**username**> and <**password**> should correspond to a valid user account on the Hedvig Storage Cluster.

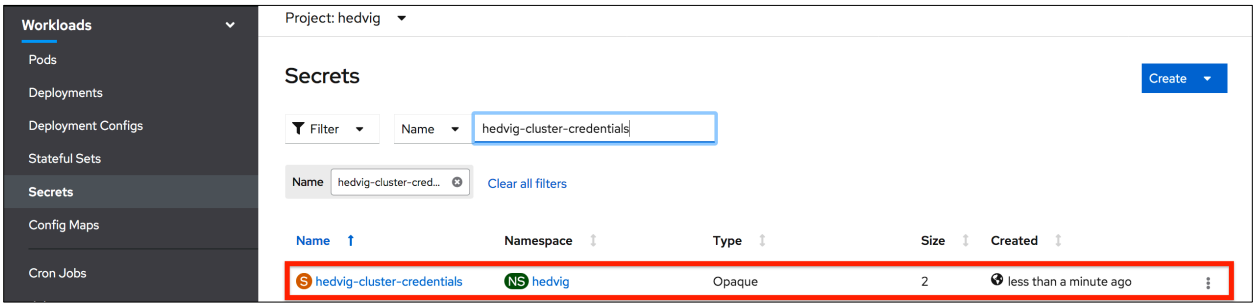
2. Install the `HedvigDeploy` resource.

   ```
   kubectl create –f hedvig-deploy.yaml
   ```

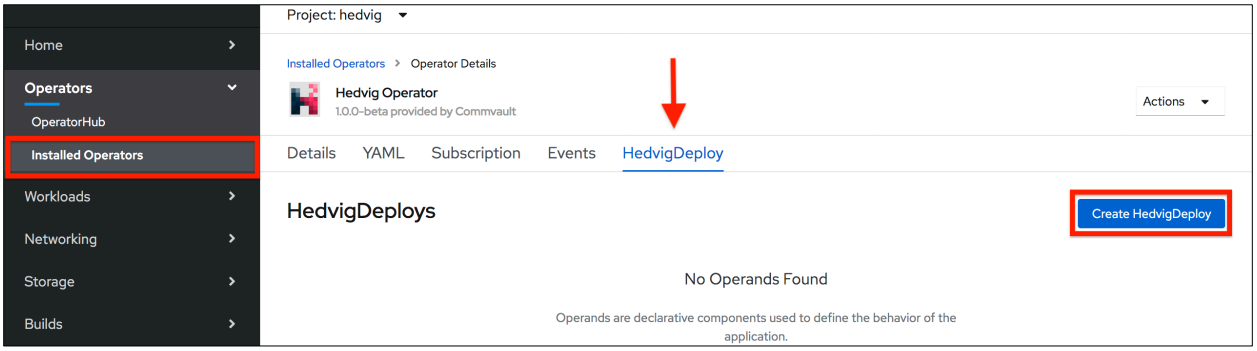
Install the HedvigDeploy Resource using OpenShift UI

1. Create the secret for the Hedvig Storage Cluster with the name `hedvig-cluster-credentials` in the namespace where the Hedvig components will be installed.

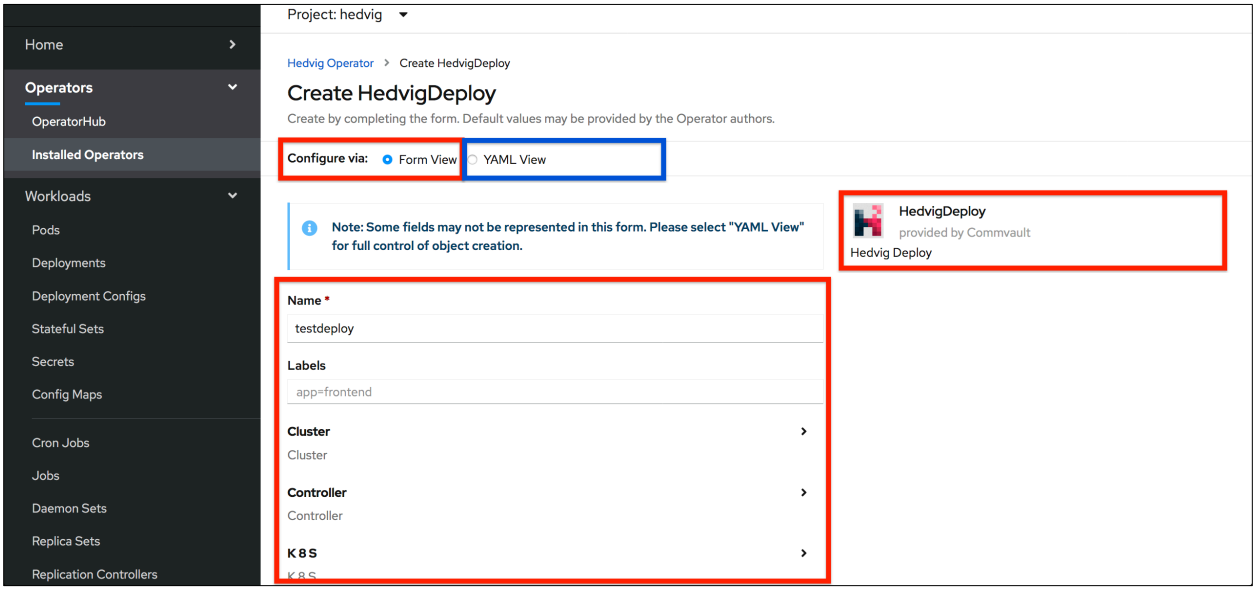




2. Navigate to the Hedvig Operator under the **Installed Operators** tab, and create **HedvigDeploy** resource.



3. HedvigDeploy resource can be created by entering the Hedvig Deployment configuration details through a form or by uploading the `hedvig-deploy.yaml` file.



Verify the Installation

After installing the `HedvigDeploy` resource, verify that the following Hedvig components are created in the specified namespace.

1. Hedvig Proxy Daemonset

```
kubectl get daemonset -n hedvig

NAME                DESIRED CURRENT READY UP-TO-DATE AVAILABLE NODE SELECTOR

hedvig-proxy-block  2       2       2     2          2         <none>

hedvig-proxy-nfs    2       2       2     2          2         <none>
```

2. Hedvig CSI Driver

```
kubectl get deployment -n hedvig

NAME                    READY   UP-TO-DATE   AVAILABLE

hedvig-csi-controller   1/1     1            1


kubectl get daemonset -n hedvig

NAME              DESIRED CURRENT READY UP-TO-DATE AVAILABLE NODE SELECTOR

hedvig-csi-node   2       2       2     2          2         <none>
```

Upgrade Hedvig Components using the Operator

Upgrade the Hedvig components using the Operator:

- *Hedvig Storage Proxy Upgrade*
- *CSI Driver Upgrade*

Hedvig Storage Proxy Upgrade

To upgrade the Hedvig Storage Proxies deployed, edit the `HedvigDeploy` resource, and specify the new image tag for the storage proxy.

```
spec:
…
…
  proxy:
    block: true
    imagePullPolicy: IfNotPresent
    nfs: true
    repository: hedvig/hedvig-proxy
    tag: <new-image-tag>
```

Alternately, you can also use the following `kubectl` command:

```
kubectl patch hedvigdeploy <deployment-name> -p
'{"spec":{"proxy":{"tag":"<new-image-tag>"}}}' -n <namespace> --type=merge
```

CSI Driver Upgrade

To upgrade the Hedvig CSI Driver deployed, edit the `HedvigDeploy` resource, and specify the new image tag for the CSI controller and node.

```
spec:
…
…
  controller:
    imagePullPolicy: IfNotPresent
    repository: hedvig/hedvig-csi-volume-driver
    tag: <new-image-tag>
  node:
    imagePullPolicy: IfNotPresent
    repository: hedvig/hedvig-csi-volume-driver
    tag: <new-image-tag>
```

Alternately, you can also use the following `kubectl` commands:

```
kubectl patch hedvigdeploy <deployment-name> -p
'{"spec":{"controller":{"tag":"<new-image-tag>"}}}' -n <namespace>
--type=merge

kubectl patch hedvigdeploy <deployment-name> -p
'{"spec":{"node":{"tag":"<new-image-tag>"}}}' -n <namespace> --type=merge
```

Appendix A: Hedvig Block Volumes with Rancher Kubernetes Clusters

In a Kubernetes cluster setup using Rancher, the iSCSI initiator is embedded in the kubelet, which is created using the rancher/hyperkube Docker image.

In most situations, the kubelet should be able to discover and create iSCSI sessions with the Hedvig iSCSI target (Hedvig proxy daemonset) for dynamically provisioned Hedvig Block volumes.

In some instances, the iSCSI initiator embedded in the kubelet might not be compatible with the Hedvig iSCSI target. In these instances, it is recommended that you follow the steps detailed in the Rancher documentation to reconfigure the kubelet.