IBM® Storage

IBM Storage for Red Hat OpenShift Blueprint

IBM Storage Team

Contents

About this document . . . 1
Executive summary . . . 1
Scope . . . 2
Prerequisites . . . 2
Virtual machine resources . . . 2
Network connectivity requirements . . . 3
Internet access . . . 3
vCenter requirements . . . 3
vCenter root CA certificate . . . 3
Lab topology . . . 4
Red Hat OpenShift Container Platform installation overview . . . 4
DNS . . . 5
SSH . . . 5
vCenter CA certificates . . . 5
Cluster deployment . . . 5
Accessing cluster by using the OC command-line client . . . 6
Adding Red Hat Enterprise Linux 7.6 worker nodes . . . 7
Configuring iSCSI/Fibre Channel for worker nodes . . . 8
(Optional) Manually configuring worker nodes running Red Hat Enterprise Linux . . . 8
Configuring worker nodes running Red Hat Enterprise Linux or Red Hat Enterprise Linux CoreOS . . . 9
Configuring for OpenShift Container Platform users (RHEL and RHCOS) with multipathing 9
Configuring iSCSI connectivity . . . 11
Configuring Fibre Channel connectivity . . . 12
Installing IBM block storage CSI driver on Red Hat OpenShift Container Platform . . . 13
Installing from the OpenShift web console . . . 13
Creating a secret for authentication to your storage devices . . . 19
Creating a block.csi.ibm.com StorageClass . . . 19
Installing IBM Spectrum Scale CSI Driver on Red Hat OpenShift Container Platform . . . 20
Installing IBM Spectrum Scale . . . 20
Creating a namespace for the Spectrum Scale CSI Driver . . . 21
Creating a secret for authentication to your storage device . . . 21
IBM Spectrum Scale storage class definitions . . . 28
Deploying CockroachDB Operator and Database instance . . . 30
Creating a subscription to the CockroachDB Operator . . . 30
Using the CockroachDB operator to deploy a StatefulSet . . . 33
Verifying successful deployment of CockroachDB . . . 35
Creating volume snapshots with IBM Block CSI Driver . . . 37
Appendix A: DNS configuration . . . 38
Appendix B: DHCP configuration . . . 40
Appendix C: Firewall rules . . . 40
Appendix D: IBM Spectrum Scale usage restrictions . . . 41

Notices . . . 43
Trademarks . . . 44
Terms and conditions for product documentation . . . 45
Applicability . . . 45

Commercial use 45
Rights 45
Privacy policy considerations 45

About this document

This IBM® Blueprint is intended to facilitate the deployment of IBM Storage for Red Hat OpenShift Container Platform by using detailed hardware specifications to build a system. It describes the associated parameters for configuring persistent storage within a Red Hat OpenShift Container Platform environment. To complete the tasks, you must understand Red Hat OpenShift, IBM Storage, the IBM block storage Container Storage Interface (CSI) driver, and the IBM Spectrum Scale CSI driver.

The information in this document is distributed on an "as is" basis without any warranty that is either expressed or implied. Support assistance for the use of this material is limited to situations where IBM Storwize® or IBM FlashSystem® storage devices, Enterprise Storage Server®, and IBM Spectrum® Scale are supported and entitled, and where the issues are not specific to a blueprint implementation.

IBM Storage Suite for IBM Cloud® Paks is an offering bundle that includes software-defined storage from IBM and Red Hat. Use this document for more information about how to deploy IBM Storage product licenses that are obtained through Storage Suite for Cloud Paks (IBM Spectrum Virtualize and IBM Spectrum Scale).

For more information about how to deploy Red Hat Storage, see the following web pages:

- Red Hat OpenShift Container Storage
- Red Hat Ceph Storage

Executive summary

Most organizations soon will be operating in a hybrid multicloud environment. Container technology help drive this rapid evolution from applications and data that is anchored on-premises in siloed systems, to applications and data easily moving when and where needed to gain the most insight and advantage.

IBM Storage unifies traditional and container-ready storage, and provides cloud-native agility with the reliability, availability, and security to manage enterprise containers in production. As clients scale containerized applications beyond experimental or departmental use, IBM's award-winning storage solutions enable mission-critical infrastructure that delivers shared-storage operational efficiency, price-performance leadership, and container data protection.

Through integration with the automation capabilities of Kubernetes and IBM Cloud Paks, IBM enables IT infrastructure and operations to improve developer speed and productivity, while delivering data reduction, disaster recovery, and data availability with enterprise storage. IBM Storage for Red Hat OpenShift is a comprehensive, container-ready solution that includes all of the elements and expertise that is needed for implementing the technologies that are driving businesses in the 21st century.

Scope

Starting OpenShift release 4.5, the installation of Red Hat OpenShift can be done in two ways: user provisioned infrastructure (UPI) and installer provisioned infrastructure (IPI).

As the name suggests, tasks in a UPI installation, such as deploying virtual machines from ova template, bootstrapping the installation and eventually cluster creation, and certificate signing requests must be done by user. For an IPI-based installation, many of these tasks are completed on user's behalf by the installer. Although the installation procedure is simplified, the prerequisites must be met for successful completion of the installation procedure.

This document focuses on the use of an IPI installation. The instructions for setting up prerequisites, such as domain name server (DNS), dynamic host configuration protocol (DHCP), web server (HTTP), and HAProxy server are mentioned only for convenience. The OpenShift administrator can choose to create these services or integrate the installation into the organization's services.

The instructions that are provided here are not a replacement for any official documentation that is released by OpenShift or Linux operating system providers. For your OpenShift release, see the official documentation release that is available at this web page.

Prerequisites

The lab setup of OpenShift was created as installer provisioned infrastructure that uses VMware vSphere 7.0. Users who want to deploy Red Hat OpenShift Container Platform cluster with VMware NSX-T or VMware vSAN must use VMware vSphere version 6.7 update 2 (6.7U2).

All-flash storage arrays can be used for the persistent storage needs of the OpenShift registry and the containerized applications IBM storage systems, such as Spectrum Scale, IBM Enterprise Storage Server®, IBM Storwize®, or IBM FlashSystem. A working knowledge of these storage systems is highly recommended.

Virtual machine resources

The IPI deployment creates the following types of resources in a vCenter instance:

- One each:
  - Folder
  - Tag category
  - Tag
  - Template
  - Bootstrap node/boot node
- Three each:
  - Control plane nodes/master nodes
  - Compute/worker nodes

Approximately 800 - 900 GB of storage space is required for a single OpenShift cluster that is deployed by using IPI.

Network connectivity requirements

All of the RHCOS machines require network in initramfs during boot to fetch ignition configuration files from the machine configuration server. The required IP addresses can be provisioned by using DHCP server.

After the initial boot, the machines can be configured to use static IP addresses. For more information about the DNS and DHCP configuration that are used in lab, see “Appendix A: DNS configuration” on page 38.

Intra-cluster communication must be enabled on several network ports. For more information about the required ports, see “Appendix C: Firewall rules” on page 40.

Internet access

Internet access is required for the installation process and eventual updating of the cluster environment. During the installation, internet access is used to complete the following tasks:

- Download the installation program
- Obtain packages that are required to install or update the cluster
- Perform subscription management
- Configure password-less SSH

For intra-cluster communication and for logging on to the cluster nodes, ssh-keys are used. These keys were generated on `rhel-host` by using the **`ssh-keygen`** command. The public key is automatically added to `authorized_keys` on all the OpenShift nodes by the installer.

vCenter requirements

The installation program requires access to an account with privileges to read and create the required resources in vCenter. Table 1 lists the necessary permissions.

Table 1 vCenter user permissions

Resource	Permissions
Data store	Browse, allocate space, low-level file operations, and remove file
Folder	Create and delete
vSphere tagging	All privileges
Network	Assign network
Resource	Assign virtual machines to resource pool
Profile driven storage	All privileges
vApp	All privileges
Virtual machines	All privileges

vCenter root CA certificate

The vCenter’s trusted root CA certificate must be added to the system that is designated to run the `openshift-installer` program because the installer must access vCenter API to create the OpenShift cluster.

Lab topology

The lab topology is shown in Figure 1.

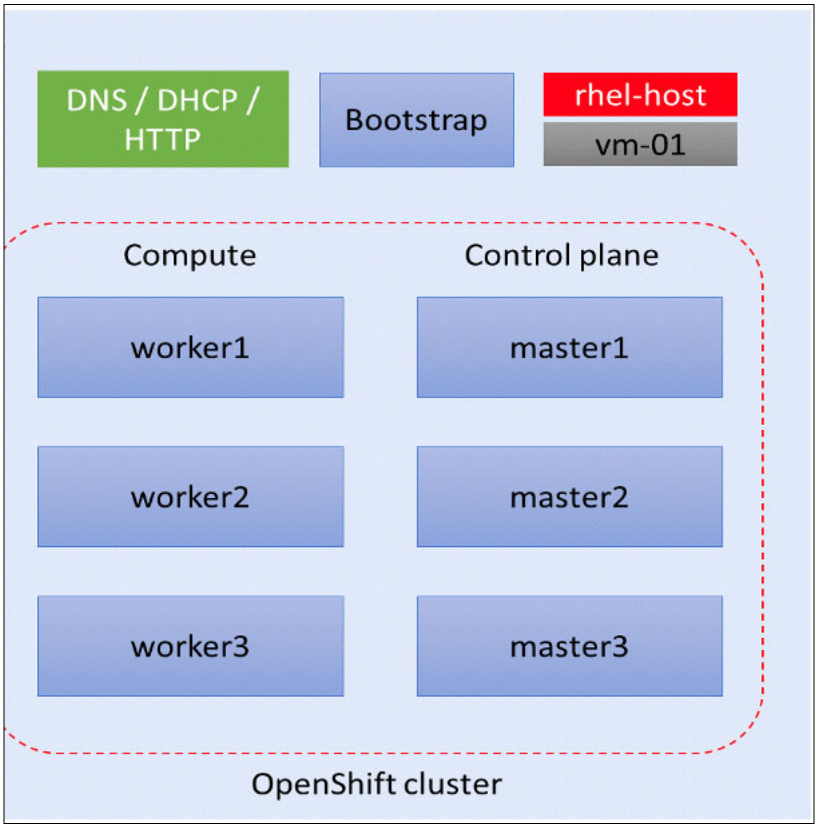

Figure 1 Lab topology

The lab environment is created by using VMware platform. The `rhel-host` (hereinafter referred to as the infrastructure node) is used to run the `openshift-installer` program for cluster creation and eventually to communicate with the cluster by using the OpenShift CLI tool (oc).

The installer program created various virtual machines, such as folder, tags, templates, bootstrap, and master and worker nodes in the lab vCenter instance. All of the cluster nodes rely on DHCP service to acquire IP addresses.

After the bootstrapping was complete, master nodes were created that formed the cluster. Then, the bootstrap node is no longer required and was discarded by the installer. The master node starts creating various cluster operators. Finally, the worker nodes were created and added to the cluster.

Red Hat OpenShift Container Platform installation overview

This section provides a checklist to confirm that the prerequisites were met and the installation procedure.

Use the next sections to confirm that the prerequisites are indeed met before starting the installation.

DNS

Example 2 lists the details of the API and Ingress DNS entries.

> **Important:** The OpenShift cluster name is configured as listed in Example 3 on page 6 as "Cluster name" and base domain in the same table.

Table 2 API and Ingress DNS entries

Endpoint	Example
api.<cluster_name>.<base_domain>	Sample DNS entry that is shown in "Appendix A: DNS configuration" on page 38.
*.apps.<cluster_name>.<base_domain>	Sample DNS entry that is shown in "Appendix A: DNS configuration" on page 38.

SSH

Complete the following steps to generate the a new SSH key:

1. Generate a new SSH key if such a key is not present, as shown in the following example:

   ```
   ssh-keygen -t rsa -b 4096 -N "" -f /$HOME/.ssh/id_rsa
   ```

2. Run the ssh-agent in background, as shown in the following example:

   ```
   eval "$(ssh-agent -s )"
   ```

3. Add the ssh private key to agent, as shown in the following example:

   ```
   ssh-add $HOME/.ssh/id_rsa
   ```

vCenter CA certificates

Complete the following steps to enable the openshift-installer to communicate with the vCenter server:

1. Download the CA bundle from `vCenter-IP-ADDRESS://certs/downloads.zip` and extract the file.
2. Copy the certificates to `/etc/pki/ca-trust/anchors`, as shown in the following example:

   ```
   cp certs/lin/* /etc/pki/ca-trust/source/anchors
   ```

3. Update system trust, as shown in the following example:

   ```
   update-ca-trust
   ```

Cluster deployment

Complete the following steps to create the Openshift ContainerPlatform cluster:

1. Create a directory to store all of the cluster-related files, as shown in the following example:

   ```
   mkdir /root/cluster1
   ```

2. Deploy the cluster by running the **openshift-installer create cluster** command, as shown in the following example:

```
openshift-install create cluster --dir=/root/cluster1 --log-level=info
```

3. Enter the required input at the installation prompt. Table 3 lists example questions and responses.

Table 3 OpenShift create cluster prompt and responses

Prompt	Response	Comment
Ssh-public-key	$HOME/.ssh/id_rsa.pub	Location of public key generated
Platform	vsphere	Platform type
vCenter	IP/FQDN of vCenter	vCenter IP address/name
Username	User to connect to vCenter	
Password	vCenter user's password	
Default data store	OpenShift-DS	
Network	OCP-Cluster-Net	
Virtual IP address for API	IP address for API endpoint	
Virtual IP address for Ingress	IP address for Ingress endpoint	
Base domain	Base domain for the cluster	
Cluster name	Name of the cluster	
Pull secret	Obtained from Red Hat Network	

The cluster installation process takes approximately 45 minutes to complete. After it is completed, the URL that is used to connect to the cluster console is displayed.

Several files are created in the directory that are provided on the command line. The `config/auth` directory contains the credentials for the `kubeadmin` user. The OpenShift console can be accessed by using these credentials.

Accessing cluster by using the OC command-line client

To monitor the cluster configuration creation, the oc command-line tool is used. Before running the oc tool, the cluster configuration is set in the environment by using the export command, as shown in the following example:

```
$ export KUBECONFIG=/root/cluster/config/auth/kubeconfig
$ oc whoami or oc get nodes
The output looks similar to following lines.
[root@cluster1-inf cluster1]# oc whoami
kube:admin
```

The output of the **oc get nodes** command is shown in Figure 2.

```
[root@cluster1-inf cluster1]# oc get nodes
NAME                                             STATUS   ROLES    AGE   VERSION
cluster1-m1.cluster1.storage-ocp.tuc.stglabs.ibm.com   Ready    master   31d   v1.18.3+47c0e71
cluster1-m2.cluster1.storage-ocp.tuc.stglabs.ibm.com   Ready    master   31d   v1.18.3+47c0e71
cluster1-m3.cluster1.storage-ocp.tuc.stglabs.ibm.com   Ready    master   31d   v1.18.3+47c0e71
cluster1-w1.cluster1.storage-ocp.tuc.stglabs.ibm.com   Ready    worker   31d   v1.18.3+47c0e71
cluster1-w2.cluster1.storage-ocp.tuc.stglabs.ibm.com   Ready    worker   31d   v1.18.3+47c0e71
```

Figure 2   Output from oc get nodes command

Adding Red Hat Enterprise Linux 7.6 worker nodes

When the cluster is online and operational, Red Hat Enterprise Linux 7.6 worker nodes can be added to the system. In our lab environment, we added one RHEL node (`oc-worker3-45`).

We completed the following steps that are provided in the Red Hat documentation to prepare the `rhel-host` and `oc-worker3-45` nodes with subscriptions for Red Hat OpenShift:

1. On the rhel-host, we installed the `openshift-ansible` and `openshift-clients` packages that are required to add an RHEL worker node, as shown in the following example:

   ```
   yum install openshift-ansible openshift-clients jq
   ```

   The `openshift-ansible` package provides installation program utilities and pulls in other packages that you require to add an RHEL compute node to your cluster, such as Ansible, playbooks, and related configuration files.

   The `openshift-clients` package provides the oc CLI; the jq package improves the display of JSON output on your command line.

2. Create an inventory directory in `/stage: mkdir /stage/inventory`.
3. Create the hosts file that will be used to add the RHEL worker node, as shown in Example 1.

Example 1   The /stage/inventory/hosts file

```
[all:vars]
ansible_user=root
#ansible_become=True
openshift_kubeconfig_path="/stage/auth/kubeconfig"
[new_workers]
oc-worker3-45
```

4. Run the playbook to add the new RHEL node:

   ```
   cd /usr/share/ansible/openshift-ansible
   ansible-playbook -i /stage/inventory/hosts playbooks/scaleup.yml
   ```

5. Approve any CSRs that are pending for the new machine:

   ```
   oc get csr
   ```

6. If all of the CSRs are valid, approve them all by running the following command:

   ```
   oc get csr -ojson | jq -r '.items[] | select(.status == {} ) | .metadata.name'
   | xargs oc adm certificate approve
   ```

7. Verify that the new node was added successfully:

   ```
   oc get nodes
   ```

Configuring iSCSI/Fibre Channel for worker nodes

Red Hat OpenShift Container Platform 4.4 worker nodes must be configured with the IBM storage system.

This section describes how to configure the storage system with the Red Hat OpenShift Container Platform worker nodes.

The test team performed the following steps in the solution lab environment to install and configure iSCSI or Fibre Channel:

1. Install the required packages on Red Hat Enterprise Linux.
2. For iSCSI configuration:
   a. Install `sg3_utils` utilities (which send SCSI commands)
   b. The iSCSI initiator server daemon
   c. The device mapper multipathing tool to configure multiple I/O paths between worker nodes and the storage array:

      ```
      yum install -y sg3_utils iscsi-initiator-utils device-mapper-multipath
      ```
3. For Fibre Channel configuration:
   a. Install `sg3_utils` utilities (which send SCSI commands)
   b. The device mapper multipathing tool to configure multiple I/O paths between worker nodes and the storage array

      ```
      yum install -y sg3_utils device-mapper-multipath
      ```

(Optional) Manually configuring worker nodes running Red Hat Enterprise Linux

Complete the following steps:

1. Use the preferred multipath settings that are shown in Example 2 for RHEL 7 and IBM FlashSystem V7200. The `multipath.conf` file is copied at `/etc/multipath.conf`.

Example 2 Preferred multipath settings

```
devices {
device {
vendor "IBM"
product "2145"
path_grouping_policy "group_by_prio" path_selector "service-time 0"
prio "alua"
path_checker "true"
failback "immediate"
no_path_retry 5
rr_weight uniform
rr_min_io_rq "1"
dev_loss_tmo 120
}
}
```

For IBM FlashSystem A9000 systems, see this IBM knowledge Center web page for more information about the IBM Storage Host Attachment Kit.

2. Configure, start, and then verify the status of the multipath daemon service. Make sure that the multipath daemon service is in the active (running) state:

```
modprobe dm-multipath
systemctl start multipathd
systemctl status multipathd
systemctl enable multipathd
```

Configuring worker nodes running Red Hat Enterprise Linux or Red Hat Enterprise Linux CoreOS

In this section, MachineConfig is created to deploy `/etc/multipath.conf` and `/etc/udev/rules.d` to support connection to IBM Storage systems. In addition, iSCSI connectivity can be configured, as needed.

Configuring for OpenShift Container Platform users (RHEL and RHCOS) with multipathing

For this process and in this section, a `99-ibm-attach.yaml` file is provided, if needed.

> **Note:** The `99-ibm-attach.yaml` configuration file overrides any `multipath.conf` file that exists on your system. Use this file only if one is not created. If a file is created, edit the `multipath.conf` as necessary.

Complete the following steps if you need the `99-ibm-attach.yml` file:

1. Save the `99-ibm-attach.yaml` file that is shown in Example 3.

Example 3 99-ibm-attach.yaml file

```
apiVersion: machineconfiguration.openshift.io/v1 kind: MachineConfig
metadata:
  labels:
    machineconfiguration.openshift.io/role: worker
  name: 99-ibm-attach
spec:
  config:
    ignition:
      version: 2.2.0
    storage:
      files:
         - path: /etc/multipath.conf
           mode: 384
           filesystem: root
           contents:
             source:
data:,defaults%20%7B%0A%20%20%20%20path_checker%20tur%0A%20%20%20%20path_selector%
20%22roundrobin%200%22%0A%20%20%20%20rr_weight%20uniform%0A%20%20%20%20prio%20cons
t%0A%20%20%20%20rr_min_io_rq%201%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%0A%2
0%20%20%20polling_interval%2030%0A%20%20%20%20path_grouping_policy%20multibus%0A%2
0%20%20%20find_multipaths%20yes%0A%20%20%20%20no_path_retry%20fail%0A%20%20%20%20u
```

```
ser_friendly_names%20yes%0A%20%20%20%20failback%20immediate%0A%20%20%20%20checker_
timeout%2010%0A%20%20%20%20fast_io_fail_tmo%20off%0A%7D%0A%0Adevices%20%7B%0A%20%2
0%20%20device%20%7B%0A%20%20%20%20%20%20%20%20path_checker%20tur%0A%20%20%20%20%20
%20%20%20product%20%22FlashSystem%22%0A%20%20%20%20%20%20%20%20vendor%20%22IBM%22%
0A%20%20%20%20%20%20%20%20rr_weight%20uniform%0A%20%20%20%20%20%20%20%20rr_min_io_
rq%204%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%0A%20%20%20%20%20%20%20%20path
_grouping_policy%20multibus%0A%20%20%20%20%20%20%20%20path_selector%20%22round-rob
in%200%22%0A%20%20%20%20%20%20%20%20no_path_retry%20fail%0A%20%20%20%20%20%20%20%2
0failback%20immediate%0A%20%20%20%20%7D%0A%20%20%20%20device%20%7B%0A%20%20%20%20%
20%20%20%20path_checker%20tur%0A%20%20%20%20%20%20%20%20product%20%22FlashSystem-9
840%22%0A%20%20%20%20%20%20%20%20vendor%20%22IBM%22%0A%20%20%20%20%20%20%20%20fast
_io_fail_tmo%20off%0A%20%20%20%20%20%20%20%20rr_weight%20uniform%0A%20%20%20%20%20
%20%20%20rr_min_io_rq%201000%20%20%20%20%20%20%20%20%20%20%20%20%0A%20%20%20%20%20
%20%20%20path_grouping_policy%20multibus%0A%20%20%20%20%20%20%20%20path_selector%
20%22round-robin%200%22%0A%20%20%20%20%20%20%20%20no_path_retry%20fail%0A%20%20%20
%20%20%20%20%20failback%20immediate%0A%20%20%20%20%7D%0A%20%20%20%20device%20%7B%0
A%20%20%20%20%20%20%20%20vendor%20%22IBM%22%0A%20%20%20%20%20%20%20%20product%20%2
22145%22%0A%20%20%20%20%20%20%20%20path_checker%20tur%0A%20%20%20%20%20%20%20%20fe
atures%20%221%20queue_if_no_path%22%0A%20%20%20%20%20%20%20%20path_grouping_policy
%20group_by_prio%0A%20%20%20%20%20%20%20%20path_selector%20%22service-time%200%22%
20%23%20Used%20by%20Red%20Hat%207.x%0A%20%20%20%20%20%20%20%20prio%20alua%0A%20%20
%20%20%20%20%20%20rr_min_io_rq%201%0A%20%20%20%20%20%20%20%20rr_weight%20uniform%2
0%0A%20%20%20%20%20%20%20%20no_path_retry%20%225%22%0A%20%20%20%20%20%20%20%20dev_
loss_tmo%20120%0A%20%20%20%20%20%20%20%20failback%20immediate%0A%20%20%20%7D%0A%7D
%0A
             verification: {}
         - path: /etc/udev/rules.d/99-ibm-2145.rules
           mode: 420
           filesystem: root
           contents:
             source:
data:,%23%20Set%20SCSI%20command%20timeout%20to%20120s%20%28default%20%3D%3D%2030%
20or%2060%29%20for%20IBM%202145%20devices%0ASUBSYSTEM%3D%3D%22block%22%2C%20ACTION
%3D%3D%22add%22%2C%20ENV%7BID_VENDOR%7D%3D%3D%22IBM%22%2CENV%7BID_MODEL%7D%3D%3D%2
22145%22%2C%20RUN%2B%3D%22/bin/sh%20c%20%27echo%20120%20%3E/sys/block/%25k/device/
timeout%27%22%0A
             verification: {}
     systemd:
       units:
         - name: multipathd.service
           enabled: true
         # uncomment the following line if this MachineConfig will be used with
iSCSI connectivity
         #- name: iscsid.service
         #  enabled: true
```

2. Apply the yaml file by running the following command:
   `oc apply -f 99-ibm-attach.yaml`
3. RHEL users should verify that the `systemctl` status `multipathd` output indicates that the multipath status is active and error-free. Run the following commands to see whether multipath is correctly configured:

```
systemctl status multipathd
multipath -ll
```

Configuring iSCSI connectivity

Complete the following steps to configure iSCSI connectivity:

1. (Optional) Update the iSCSI initiator name in the /etc/iscsi/initiatorname.iscsi file with the worker node <hostname> inserted after the InitiatorName:

   ```
   InitiatorName=iqn.1994-05.com.redhat:<hostname>-<random generated number>
   ```

   For example:

   ```
   InitiatorName=iqn.1994-05.com.redhat:oc-worker1-410-74b436a728b6
   ```

2. Add host definitions to the IBM FlashSystem storage array by selecting hosts from the GUI console. Also, provide the iSCSI initiator name that is shown in Step 1 in the /etc/iscsi/initiatorname.iscsi file.
3. Click **Add** to add the host definition (see Figure 3).

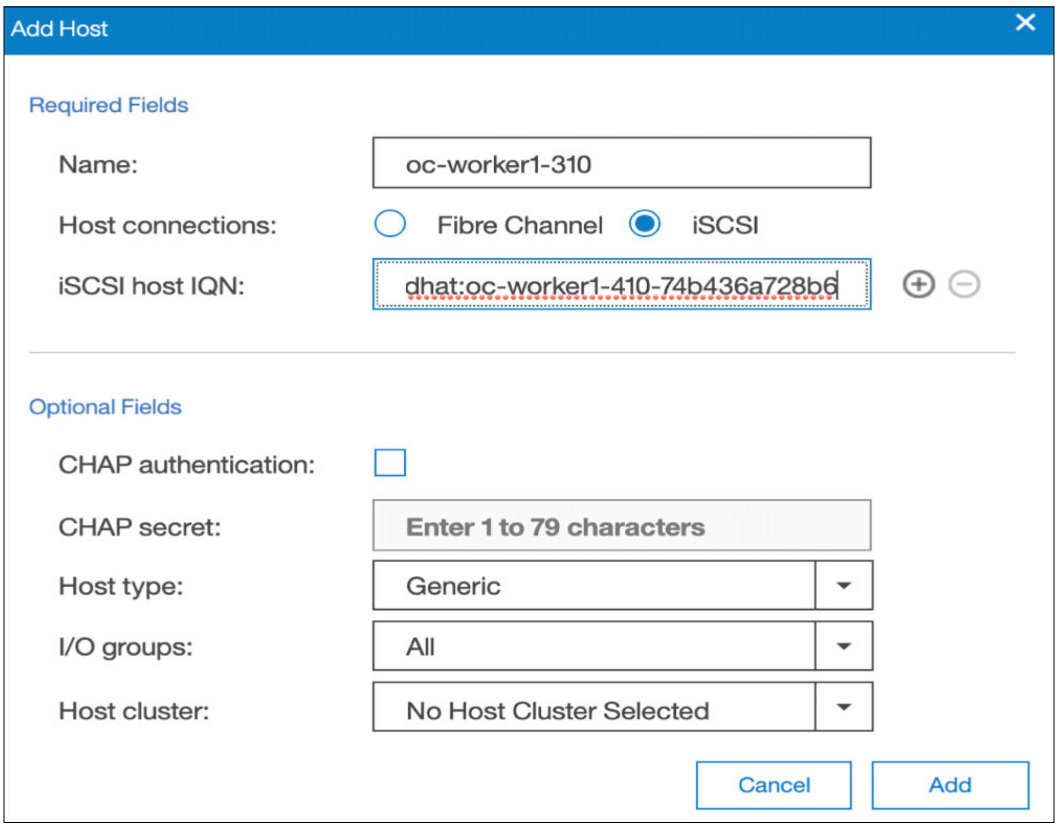

Figure 3 Adding iSCSI host

4. For the iSCSI initd script startup, set a session to automatic in /etc/iscsi/iscsid.conf: node.startup = automatic
5. Discover the iSCSI targets by using the iscsiadm CLI:

   ```
   iscsiadm -m discoverydb -t st -p <IP Address configured for iSCSI @ FlashSystem
   Storage Array>:3260 --discover
   ```

6. Verify the host by using the Storwize GUI console.

Configuring Fibre Channel connectivity

Complete the following steps by using the `systool` to get the Fibre Channel WWPN that will be associated with the host definition on the storage device on RHEL nodes:

1. Install `sysfsutils` to simplify getting the FC WWPN:

   ```
   yum -y install systools
   ```

2. Run `systool` against the `fc_host` to get the WWPN for each installed FC adapter:

   ```
   systool -c fc_host -v | grep port_name
   port_name = "0x10008c7cffb01b00"
   port_name = "0x10008c7cffb01b01"
   ```

3. Zone your host ports to the storage array storage ports on your FC switches.
4. Add host definitions to the FlashSystem storage array by selecting hosts from the GUI console. If zoning is used, you can select the host WWPN from the drop-down list. Click **Add** to add the host definition (see Figure 4).

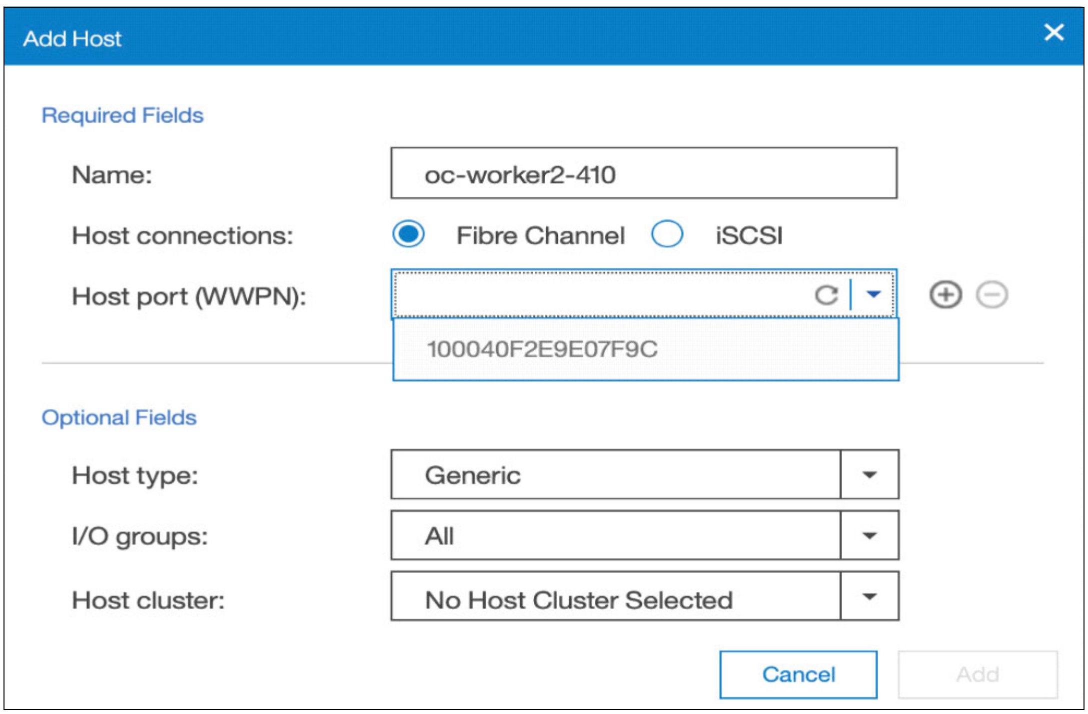

Figure 4 Adding FC host

Installing IBM block storage CSI driver on Red Hat OpenShift Container Platform

The following section describes how to install the IBM block storage CSI driver to work with OpenShift Container Platform 4.4. The source code and more information are available at this web page.

Installing from the OpenShift web console

When the Red Hat OpenShift Container Platform is used, the Operator for IBM block storage CSI driver can be installed directly from the OpenShift web console through the OperatorHub. Installing the Container Storage Interface (CSI) driver is part of the Operator installation process. The source code and more information are available at the following GitHub web pages:

- ibm-block-csi-driver
- ibm-block-csi-operator

Procedure

Complete the following steps:

1. From the Red Hat OpenShift container platform home, click **Projects** → **Create Project** and enter the following information:
   - Name: `ibm-block-csi`
   - Display name: `ibm-block-csi`
   - Description: `IBM Block CSI`
2. From the Red Hat OpenShift container platform home, click **OperatorHub** → **Project: ibm-block-csi**.

3.Search for IBM block storage CSI driver, as shown in Figure 5.

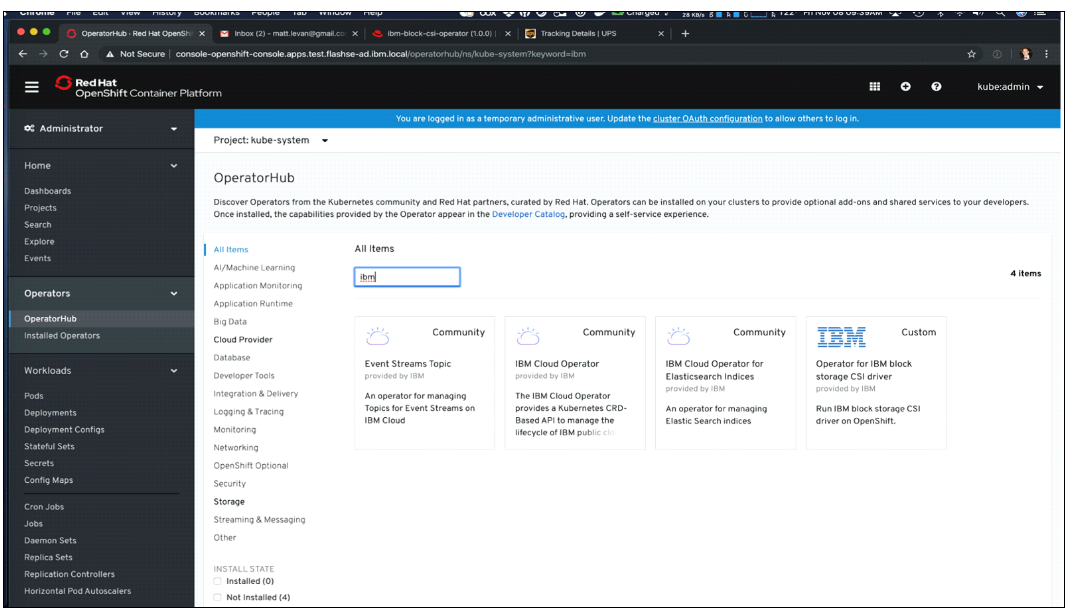

Figure 5 Searching IBM block storage driver in catalog

3. Select the Operator for IBM block storage CSI driver and click **Install**, as shown in Figure 6. The Operator Subscription is displayed.

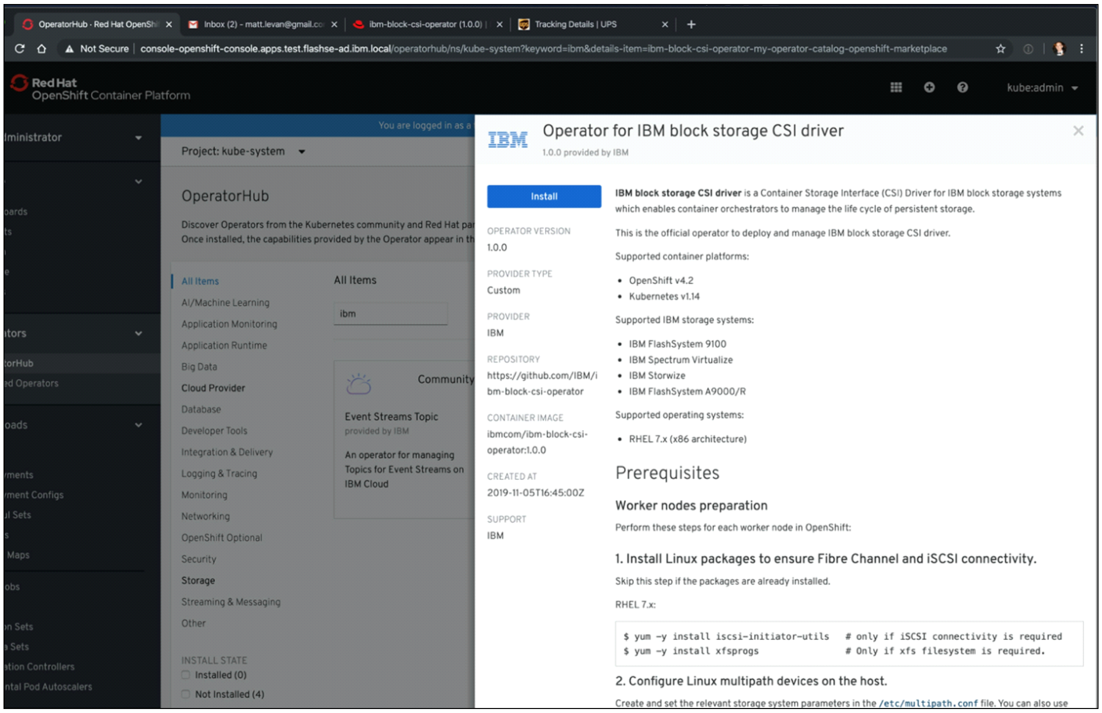

Figure 6 Installing the operator

4. Under the **A specific namespace on the cluster section**, select **ibm-block-csi** (see Figure 7).

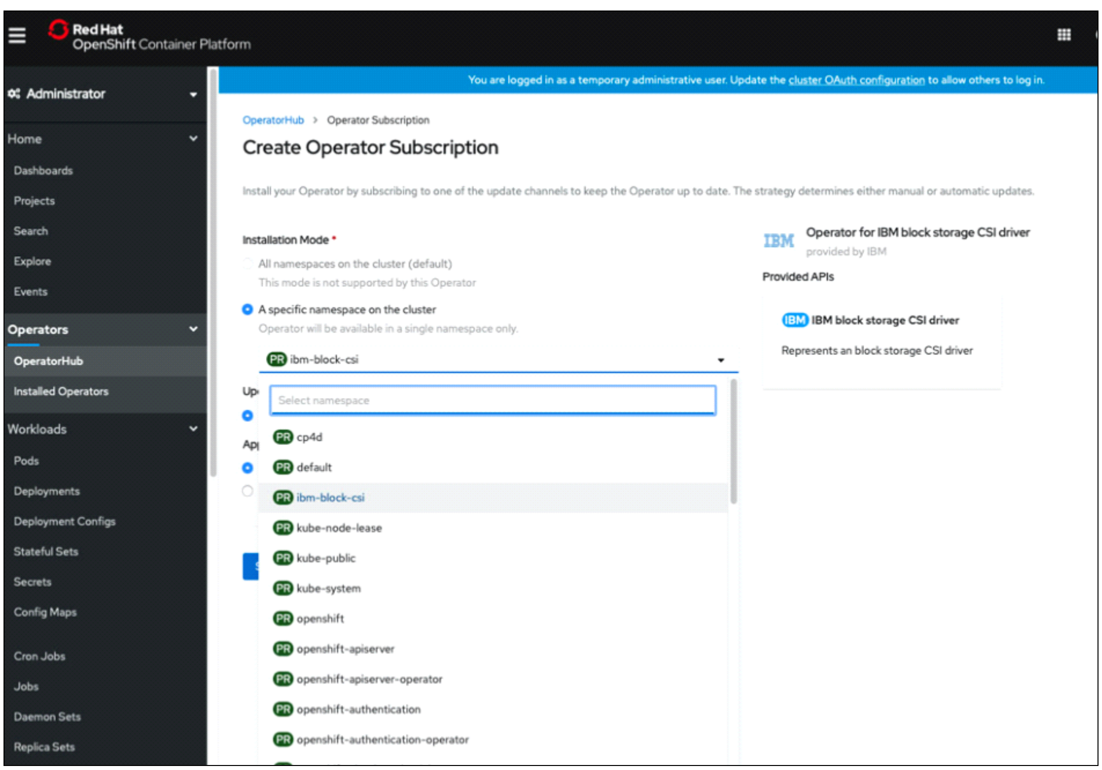

Figure 7 Selecting ibm-block-csi namespace

5. Click **Subscribe**, as shown in Figure 8.

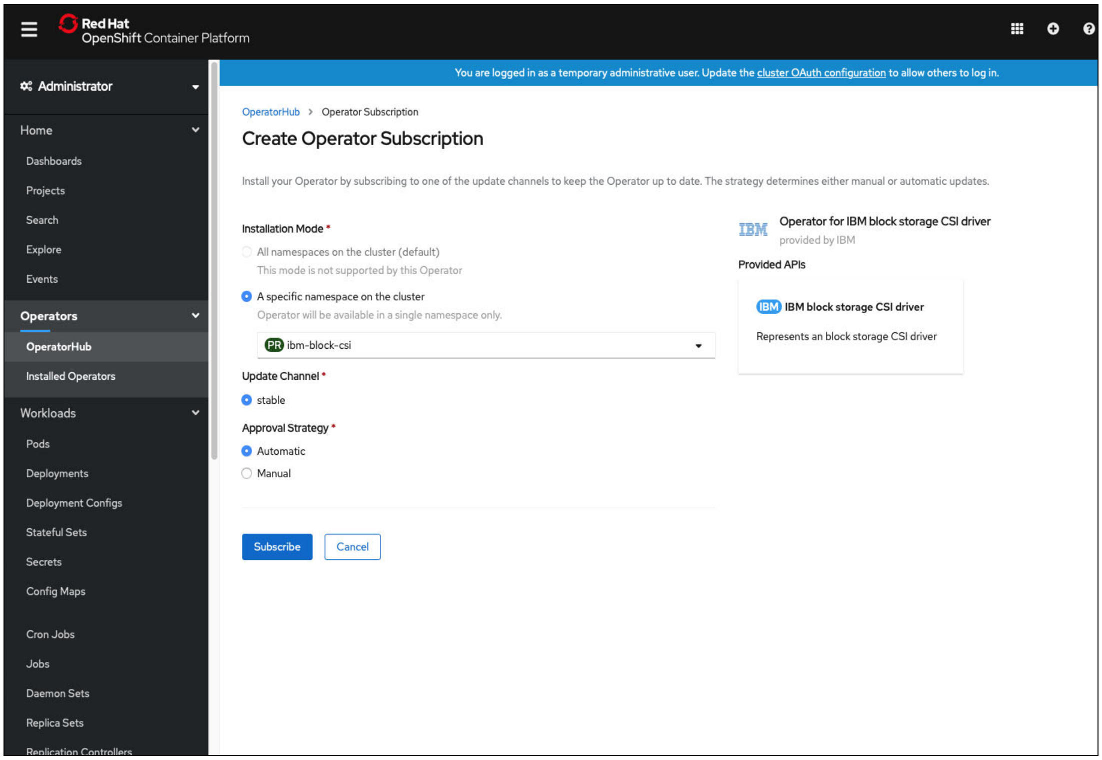

Figure 8 Subscribing to operator

6. From **Operators** → **Installed Operators**, check the status of the operator for IBM block storage CSI driver, as shown in Figure 9.

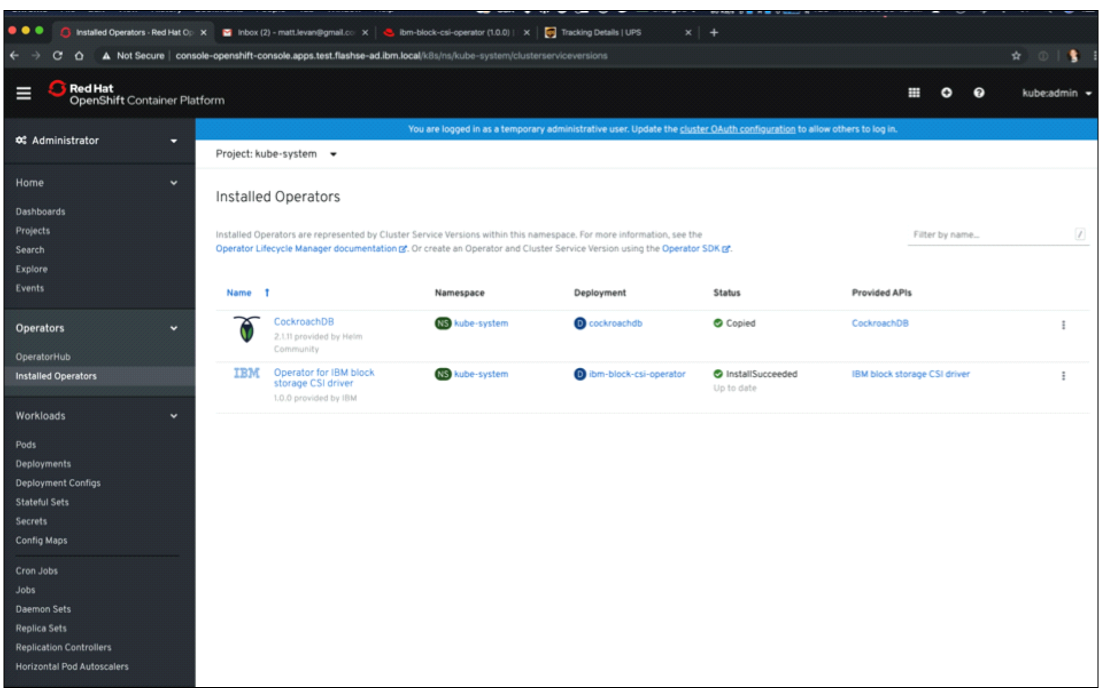

Figure 9 Operator is installed

7. Wait until the status is shows as “Up to date” and then “Install Succeeded”.

8. When the operator installation progress completes, click the installed operator for IBM block storage CSI driver.
9. Click **Create Instance** to create IBM Block CSI (see Figure 10).

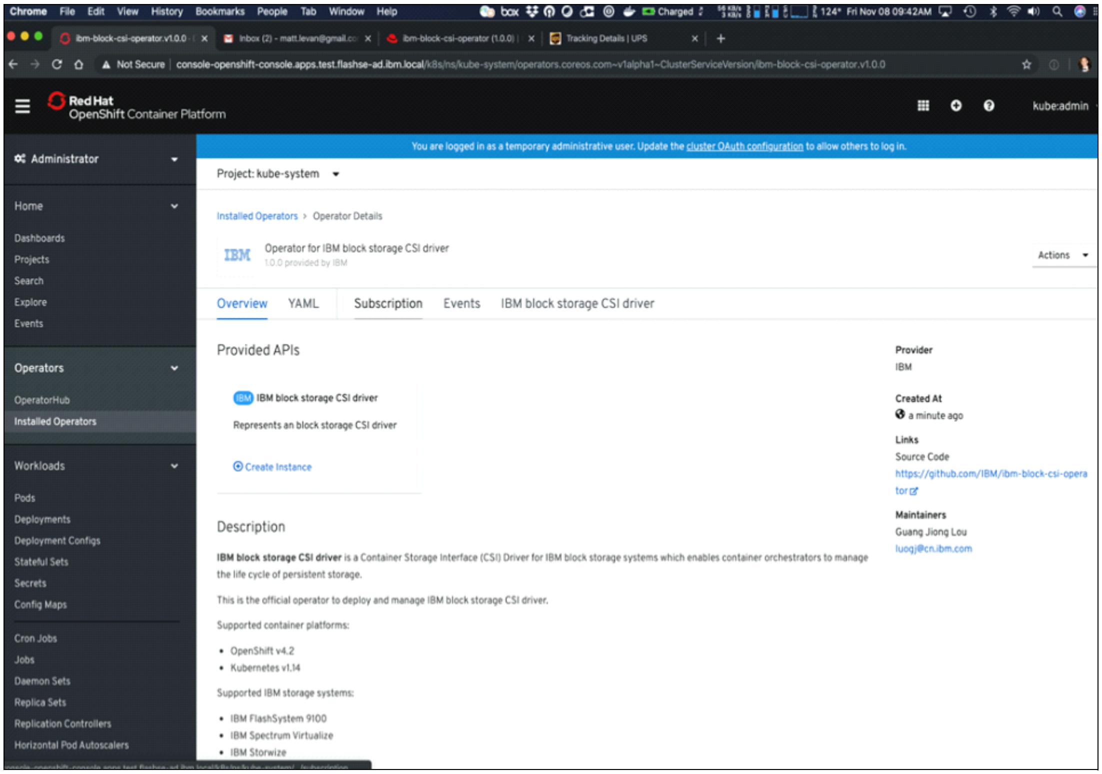

Figure 10 Operator overview

10. If needed, edit the `.yaml` file in the web console, as shown in Figure 11.

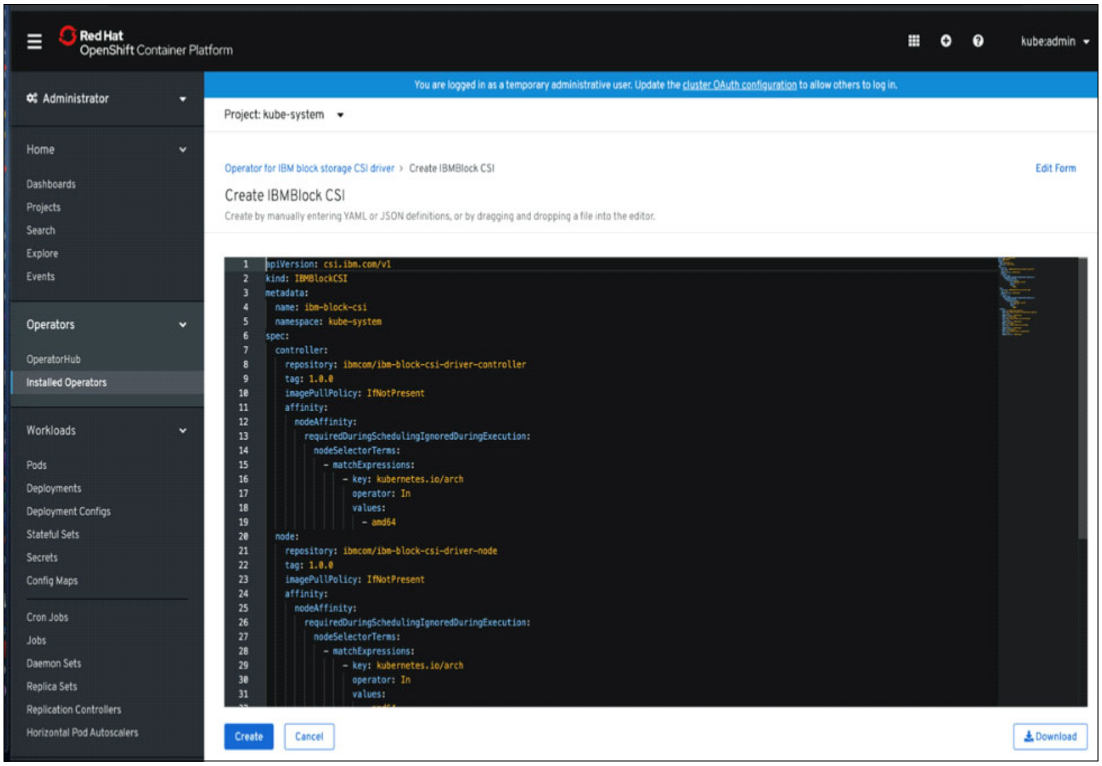

Figure 11 Editing IBM Block CSI driver yaml

The `.yaml` file that was used in the lab configuration is shown in Example 4 on page 18.

Example 4 The .yaml file that was used in the lab configuration


```yaml
apiVersion: csi.ibm.com/v1
kind: IBMBlockCSI
metadata:
   name: ibm-block-csi
   namespace: ibm-block-csi
 spec:
   controller:
     affinity:
       nodeAffinity:
         requiredDuringSchedulingIgnoredDuringExecution:
           nodeSelectorTerms:
             - matchExpressions:
                 - key: kubernetes.io/arch
                   operator: In
                   values:
                     - amd64
                     - s390x
     imagePullPolicy: IfNotPresent
     repository: ibmcom/ibm-block-csi-driver-controller
     tag: 1.4.0
   node:
     affinity:
       nodeAffinity:
         requiredDuringSchedulingIgnoredDuringExecution:
           nodeSelectorTerms:
             - matchExpressions:
                 - key: kubernetes.io/archå
                   operator: In
                   values:
                     - amd64
                     - s390x
     imagePullPolicy: IfNotPresent
     repository: ibmcom/ibm-block-csi-driver-node
     tag: 1.4.0
   sidecars:
     - imagePullPolicy: IfNotPresent
       name: csi-node-driver-registrar
       repository: k8s.gcr.io/sig-storage/csi-node-driver-registrar
       tag: v2.0.1
     - imagePullPolicy: IfNotPresent
       name: csi-provisioner
       repository: k8s.gcr.io/sig-storage/csi-provisioner
       tag: v2.0.2
     - imagePullPolicy: IfNotPresent
       name: csi-attacher
       repository: k8s.gcr.io/sig-storage/csi-attacher
       tag: v3.0.0
     - imagePullPolicy: IfNotPresent
       name: csi-snapshotter
       repository: k8s.gcr.io/sig-storage/csi-snapshotter
       tag: v3.0.0
     - imagePullPolicy: IfNotPresent
       name: csi-resizer
       repository: k8s.gcr.io/sig-storage/csi-resizer
       tag: v1.0.0
     - imagePullPolicy: IfNotPresent
       name: livenessprobe
       repository: k8s.gcr.io/sig-storage/livenessprobe
       tag: v2.1.0
 status:
   controllerReady: true
   nodeReady: true
   phase: Running
   version: 1.4.0
```

11.Click **Create** and wait until the status is running.

Creating a secret for authentication to your storage devices

In this section, we describe the two options that can be used to create a secret for authentication to your storage devices:

- From a YAML file
- By using the OC command line

Option 1: Creating a storage device secret from a YAML file

> **Note:** Data values must be encoded as base64 for entry into the yaml file. The output from base64 is entered in the `data.Password` field.

Run following command to create a base64 encoded value:

```
echo -n superuser | base64
echo -n <password-for-storge-array> | base64

File: flashsystem-secret.yaml

apiVersion: v1
data:
  management_address: aXN2N2sxMW1nbXQxLnR1Yy5zdGdsYWJzLmlibS5jb20=
  password: cGFzc3cwcmQ=      # base64 encoded storage array password
  username: c3vwUiGoc2eY      # base64 encoded storage array username
kind: Secret
metadata:
  creationTimestamp: "2020-05-11T08:59:32Z"
  name: storwize
  namespace: ibm-block-csi
  resourceVersion: "2685627"
  selfLink: /api/v1/namespaces/ibm-block-csi/secrets/storwize
  uid: 5979fbb4-3244-4d51-b721-89ebc81c601c
type: Opaque
```

Option 2: Create the secret by using oc command line

Run the following command to create the secret by using the OC command line:

```
oc create secret generic flashsystem --from-literal=management_address=
flashv7k-1.isvcluster.net --from-literal=username=superuser
--from-literal=password=passw0rd -n ibm-block-csi
```

Creating a block.csi.ibm.com StorageClass

To create the StorageClass, use following yaml:

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: ibmc-block-gold
parameters:
  SpaceEfficiency: thin
  csi.storage.k8s.io/controller-publish-secret-name: flashsystem
```

```
  csi.storage.k8s.io/controller-publish-secret-namespace: ibm-block-csi
  csi.storage.k8s.io/provisioner-secret-name: flashsystem
  csi.storage.k8s.io/provisioner-secret-namespace: ibm-block-csi
  pool: isvFS_1
provisioner: block.csi.ibm.com
reclaimPolicy: Delete
volumeBindingMode: Immediate
```

Installing IBM Spectrum Scale CSI Driver on Red Hat OpenShift Container Platform

In this section, we describe how to install the IBM Spectrum Scale CSI Driver on Red Hat OpenShift Container Platform.

IBM Spectrum Scale is a parallel, scale-out, high-performance solution that consolidates traditional file-based and new-era workloads to support artificial intelligence, data lake and object storage, Hadoop, Spark, and analytics use cases.

IBM Spectrum Scale helps clients optimize for cost and performance by using intelligent data management that automates movement of data to the optimal storage tier without affecting the user. Known for performance and reliability, IBM Spectrum Scale provides data storage for some of the largest compute clusters in the world.

IBM Spectrum Scale v5.0 is required on all IBM Spectrum Scale nodes in the Kubernetes cluster (all nodes that run the IBM Spectrum Scale CSI Driver code). For more information about recommendations for the exact IBM Spectrum Scale v5.4 level to use, see IBM Support's IBM Spectrum Scale Software Version Recommendation Preventive Service Planning.

Table 4 lists the software requirements for the solution that was used in our lab test environment.

Table 4 Software requirements for the solution lab

Software solution requirements	**Version**
IBM Spectrum Scale	V5.0.5.1+
IBM Elastic Storage® Server[a]	V5.3.4.2+

a. IBM Elastic Storage Server is shown to illustrate storage compatibility with IBM Spectrum Scale and IBM Spectrum Scale CSI Driver. However, IBM Spectrum Scale CSI Driver, Kubernetes, or Red Hat OpenShift code cannot be installed directly on the IBM Elastic Storage Server Elastic Management Server or IBM Elastic Storage Server I/O nodes. IBM Elastic Storage Server is managed by the larger IBM Spectrum Scale cluster. IBM Spectrum Scale CSI Driver, Kubernetes, or Red Hat OpenShift is installed on IBM Spectrum Scale nodes in the larger cluster.

Installing IBM Spectrum Scale

Complete the following steps to install IBM Spectrum Scale:

1. Set up your IBM Spectrum Scale cluster.

   For more information, see these IBM knowledge Center web pages:

   – IBM Spectrum Scale cluster configurations

   - Steps for establishing and starting your IBM Spectrum Scale cluster
2. You must add all Red Hat OpenShift Container Platform 4.4 worker nodes as IBM Spectrum Scale client nodes.

   For more information, see this IBM Knowledge Center web page.
3. Next, you must create a file system in your IBM Spectrum Scale cluster.

   For more information about creating a file system, see this IBM Knowledge Center web page.

   For more information about the command that is used to create a file system, see this IBM Knowledge Center web page.
4. Finally, mount the file system on all worker nodes in the Red Hat OpenShift Container Platform cluster.

   For more information, see this IBM Knowledge Center web page.

Before using IBM Spectrum Scale CSI Driver with IBM Spectrum Scale, make note of the conditions that are described in "Appendix D: IBM Spectrum Scale usage restrictions" on page 40.

Creating a namespace for the Spectrum Scale CSI Driver

From the Red Hat OpenShift Container Platform Home, select **Projects** → **Create Project** and enter the following information:

- Name: `ibm-spectrum-scale-csi-driver`
- Display Name: `ibm-spectrum-scale-csi-driver`
- Description: `IBM Spectrum Scale CSI`

Creating a secret for authentication to your storage device

In this section, we describe the two options that can be used to create a secret for authentication to your storage device.

Option 1: Create a storage device secret from yaml file

Use the `scalegui-secret.yaml` file:

```
apiVersion: v1
data:
  password: #base64 encoded password for the spectrum scale gui
  username: #base64 encoded username for the spectrum scale gui
kind: Secret
metadata:
  name: scalegui
  labels:
    product: ibm-spectrum-scale-csi
type: Opaque
```

Apply the configuration:

```
Oc apply -f scalegui-secret.yaml
```

Option 2: Create a storage device secret with the oc command line

Run the following command:

```
oc create secret generic scalegui --from-literal=username=csi-admin
--from-literal=password=passw0rd -n ibm-spectrum-scale-csi-driver
```

Procedure

Complete following steps:

1. For Red Hat OpenShift container platform operators, from OperatorHub, select **Projects** → **ibm-spectrum-scale-csi-driver**.
2. Search for IBM Spectrum Scale CSI plug-in operator (see Figure 12).

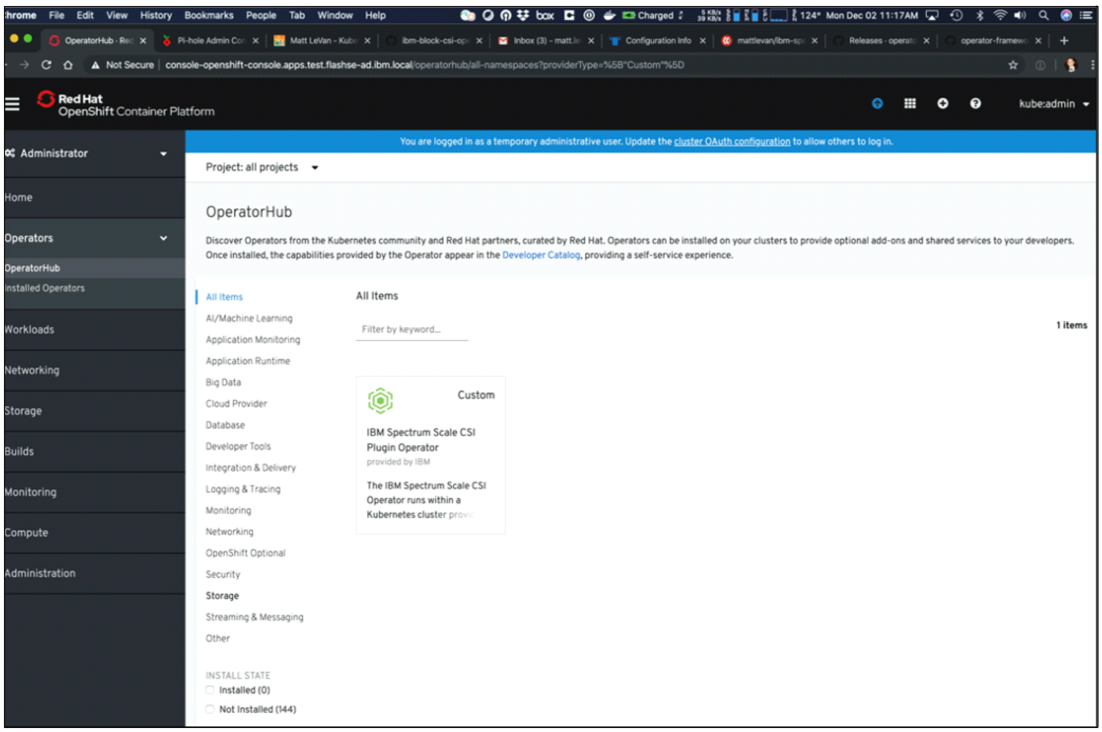

Figure 12 Operator in OperatorHub

3. Select the IBM Spectrum Scale CSI plug-in operator and click **Install** (see Figure 13).

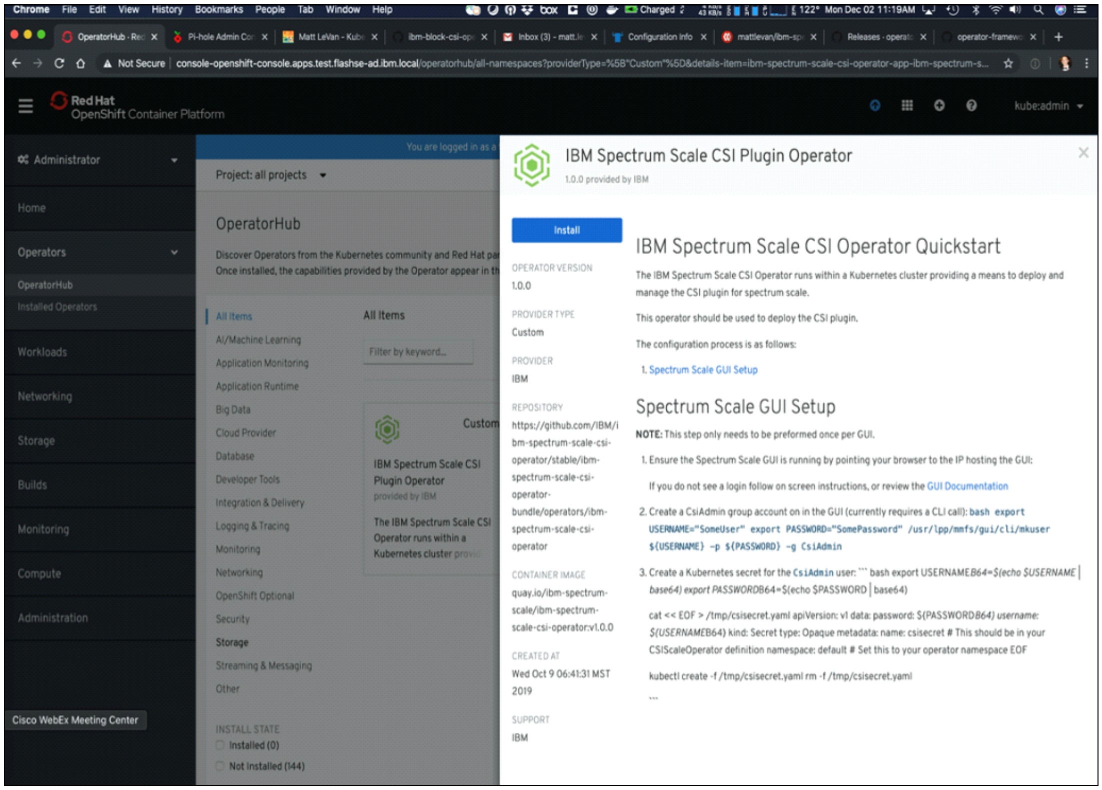

Figure 13 Installation window

4. Set the Installation Mode to `ibm-spectrum-scale-csi-driver` under A specific namespace on the cluster, as shown in Figure 14.

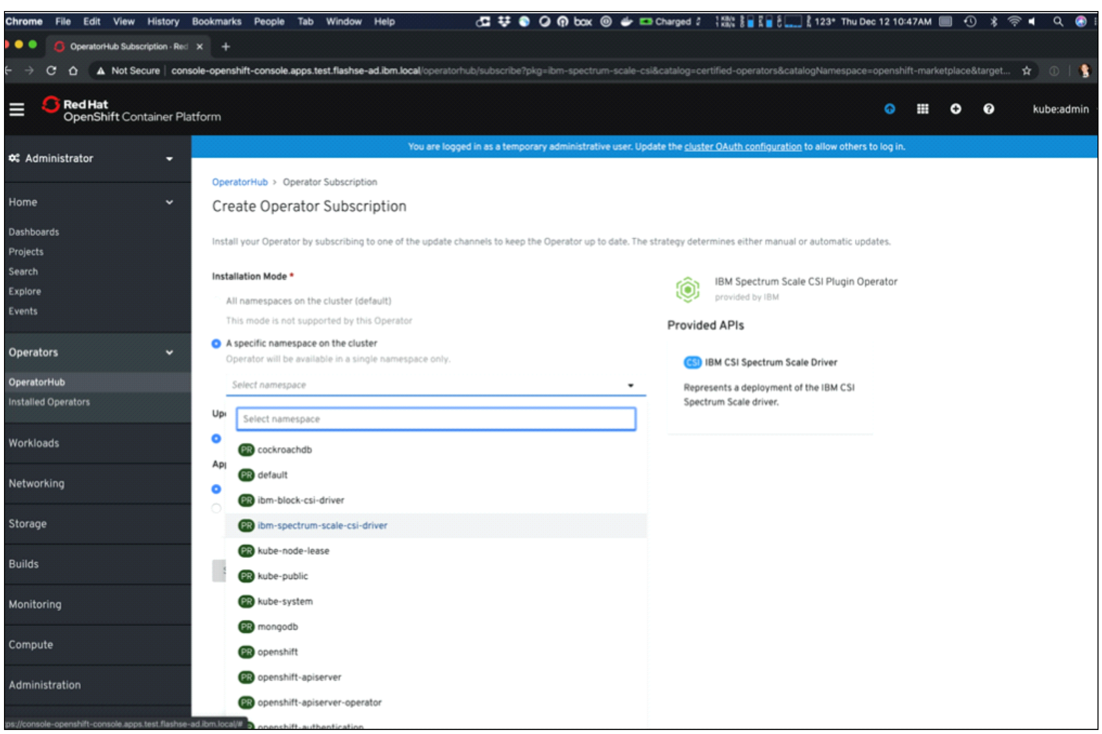

Figure 14 Selecting namespace

5. Click **Subscribe** (see Figure 15).

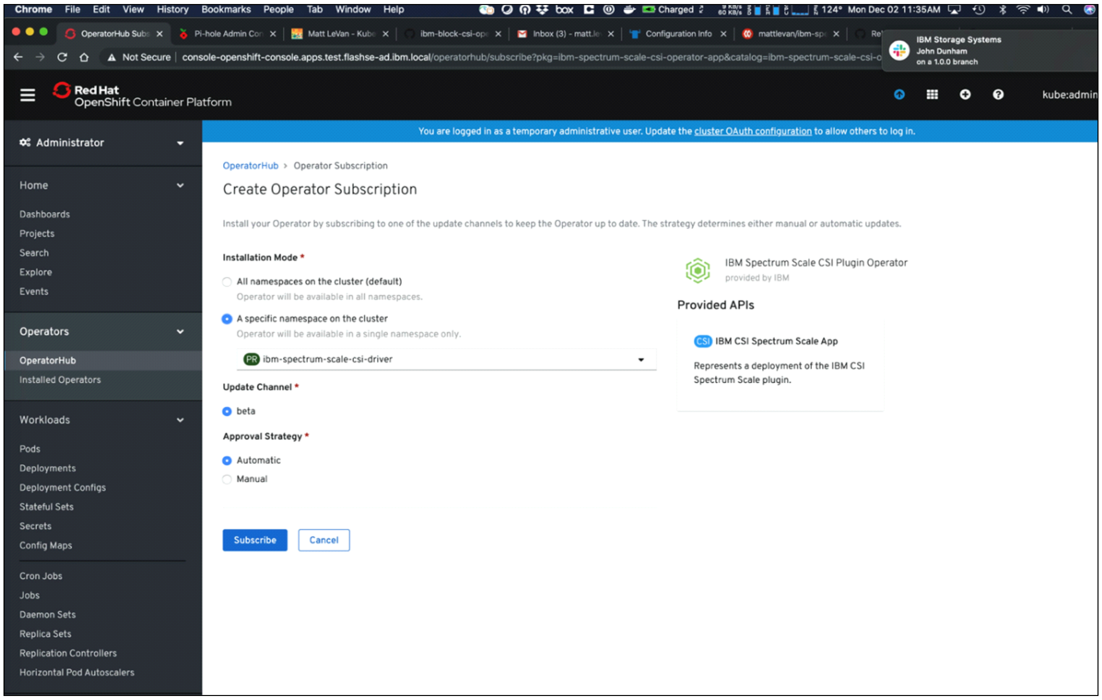

Figure 15 Adding subscription

6. From **Operators** → **Installed Operators**, check the status of the IBM Spectrum Scale CSI Plug-in Operator, as shown in Figure 16.

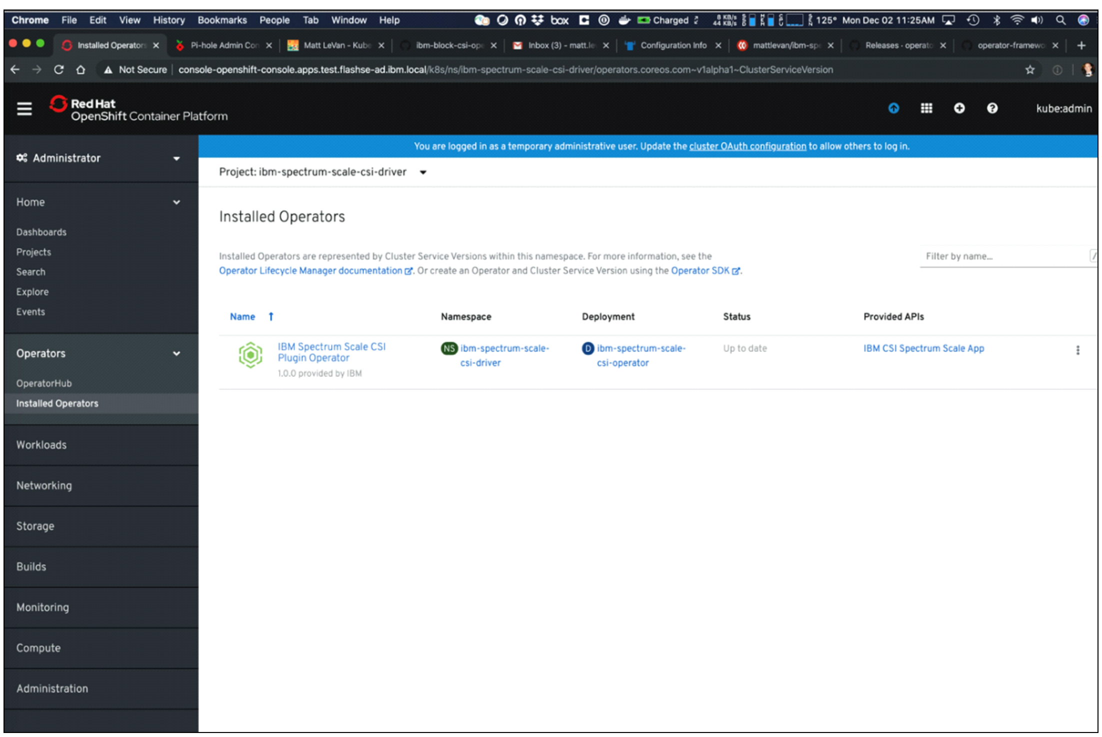

Figure 16 Operator subscription added

Wait until the Status is shown as “Up to date” and then, “Install Succeeded.”

Note: While you are waiting for the status to change, you can check the pod progress and readiness status by selecting **Workloads** → **Pods**.

7. When the operator installation progress completes, click the installed **IBM Spectrum Scale CSI Plug-in Operator** → **Create Instance** to create an IBM CSI Spectrum Scale application (see Figure 17).

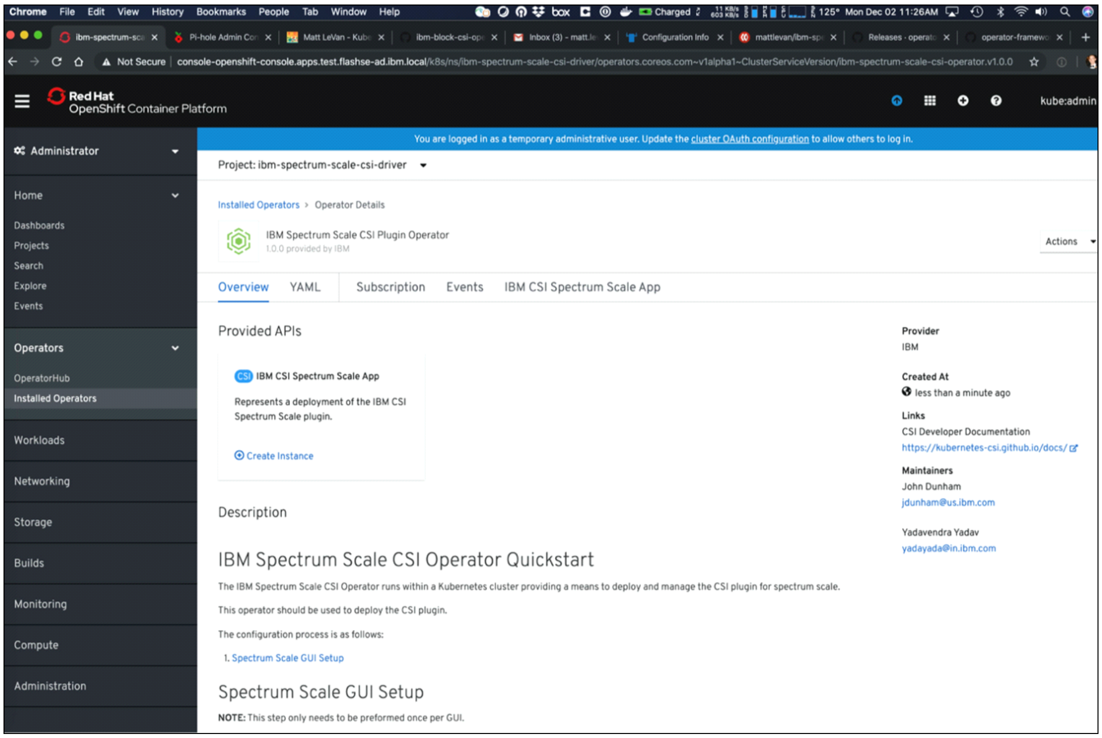

Figure 17 Creating a CR IBM Spectrum Scale Application

8. Edit the yaml file in the web console (see Figure 18).

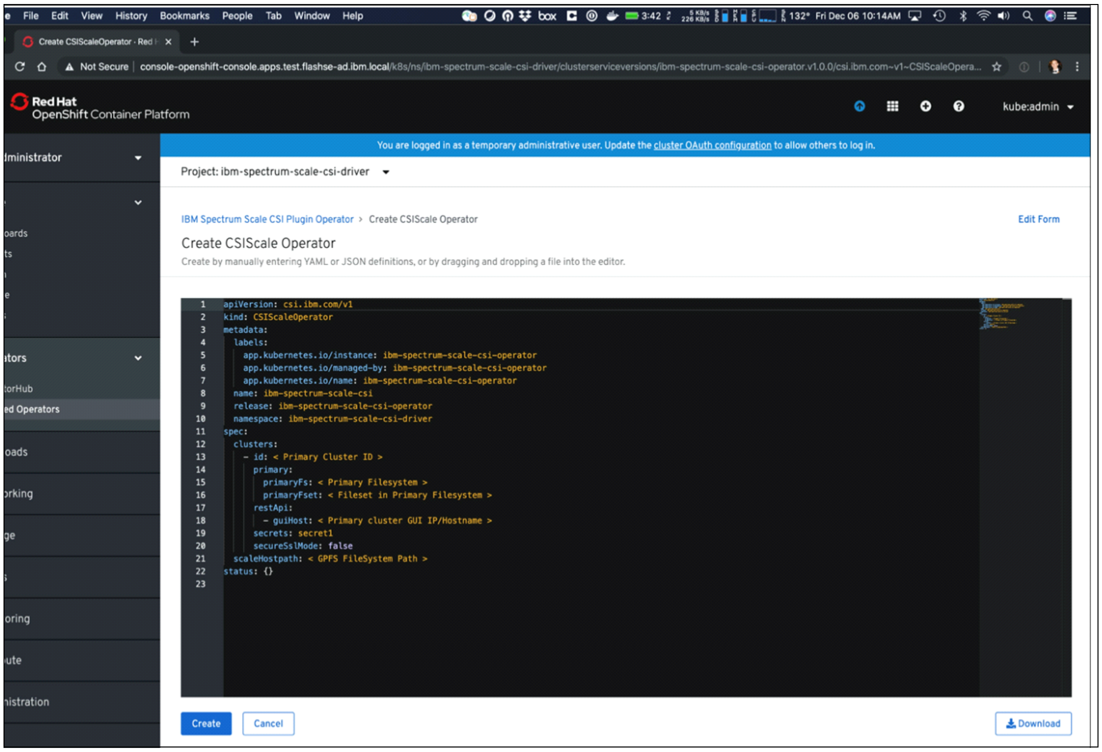

Figure 18   Editing CR

A sample yaml configuration is shown in Example 5.

Example 5   Sample yaml file configuration

```
apiVersion: csi.ibm.com/v1
kind: CSIScaleOperator
metadata:
  annotations:
  name: ibm-spectrum-scale-csi
  managedFields:
    - apiVersion: csi.ibm.com/v1
      fieldsType: FieldsV1
      fieldsV1:
        'f:metadata':
          'f:annotations':
            .: {}
            'f:kubectl.kubernetes.io/last-applied-configuration': {}
          'f:labels':
            .: {}
            'f:app.kubernetes.io/instance': {}
            'f:app.kubernetes.io/managed-by': {}
            'f:app.kubernetes.io/name': {}
        'f:spec':
          .: {}
          'f:attacherNodeSelector': {}
          'f:clusters': {}
          'f:nodeSelector': {}
          'f:pluginNodeSelector': {}
          'f:scaleHostpath': {}
```

```
      manager: oc
      operation: Update
      time: '2020-11-26T14:14:24Z'
    - apiVersion: csi.ibm.com/v1
      fieldsType: FieldsV1
      fieldsV1:
        'f:spec':
          'f:trigger': {}
      manager: Swagger-Codegen
      operation: Update
      time: '2020-12-04T10:21:12Z'
    - apiVersion: csi.ibm.com/v1
      fieldsType: FieldsV1
      fieldsV1:
        'f:metadata':
          'f:finalizers':
            .: {}
            'v:"finalizer.csiscaleoperators.csi.ibm.com"': {}
        'f:status':
          .: {}
          'f:conditions': {}
      manager: ansible-operator
      operation: Update
      time: '2020-12-04T10:21:38Z'
  namespace: ibm-spectrum-scale-csi-driver
  finalizers:
    - finalizer.csiscaleoperators.csi.ibm.com
  labels:
    app.kubernetes.io/instance: ibm-spectrum-scale-csi-operator
    app.kubernetes.io/managed-by: ibm-spectrum-scale-csi-operator
    app.kubernetes.io/name: ibm-spectrum-scale-csi-operator
spec:
  attacherNodeSelector:
    - key: scale
      value: 'true'
  clusters:
    - id: '18445892705419911881'
      primary:
        primaryFs: bm_filesystem_2
      restApi:
        - guiHost: isv35.storage-ocp.tuc.stglabs.ibm.com
      secrets: csisecret
      secureSslMode: false
  nodeSelector:
    - key: scale
      value: 'true'
  pluginNodeSelector:
    - key: scale
      value: 'true'
  scaleHostpath: /gpfs/bm_filesystem_2
  trigger: '18'
status: {}
```

Note: Consider the following points:

- If no deselector is needed to support a mixed RHEL/RHCOS cluster or to limit what nodes are attached to Spectrum Scale, see IBM Knowledge Center.
- If the Kubernetes node name differs from the IBM Spectrum Scale name, see IBM Knowledge Center.

9. Confirm that the Status is listed as running (see Figure 19).

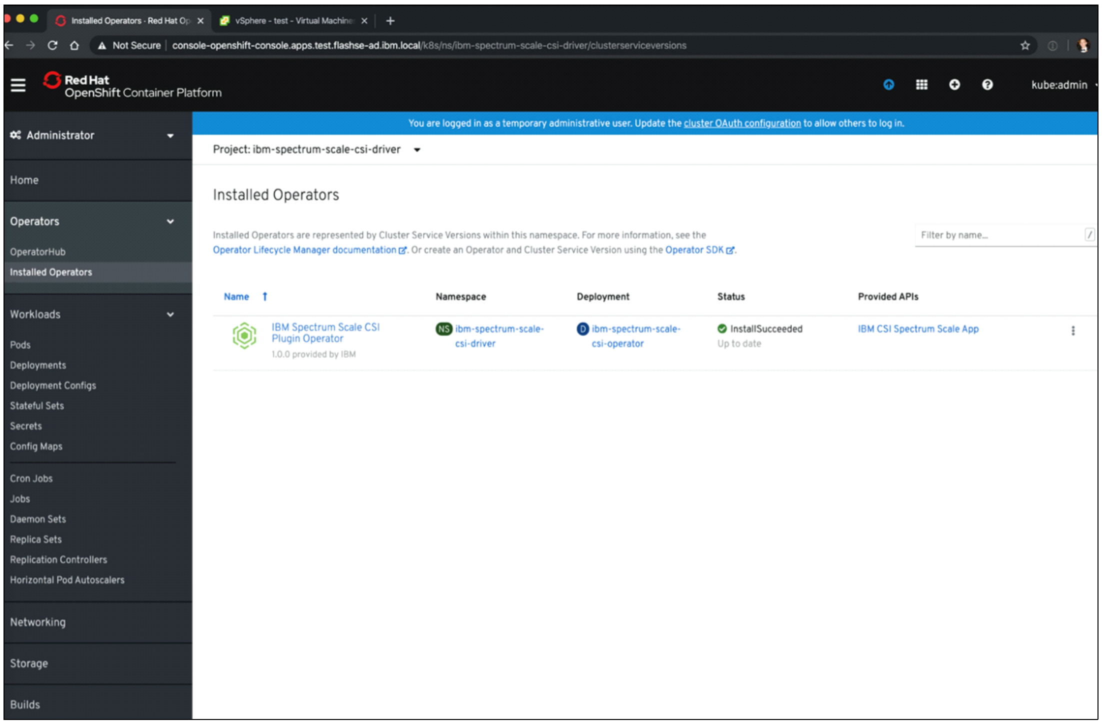

Figure 19 Installation complete

IBM Spectrum Scale storage class definitions

This section describes how to configure Kubernetes storage classes. Storage classes are used for creating lightweight volumes or file set-based volumes.

The `fileset-based-storageclass.yaml` file is shown in Example 6.

Example 6 The fileset-based-storageclass.yaml file

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  annotations:
  labels:
    product: ibm-spectrum-scale-csi
  managedFields:
  name: spectrum-scale
parameters:
  clusterId: "18445892705419911881"
```

```
  type: fileset
  volBackendFs: bm_filesystem_2
# fileset-type: "<fileset type>" # Optional,
# Values: Independent[default] or dependent # dependantFileset: "<fileset>" #
Optional
# uid: "<uid number>" # Optional
# gid: "<gid number>" # Optional
# inode-limit: "<no of inodes to be preallocated>" # Optional
provisioner: spectrumscale.csi.ibm.com
reclaimPolicy: Delete
volumeBindingMode: Immediate
```

The `lightweight-storage-class.yaml` file is shown in Example 7.

Example 7 The lightweight-storage-class.yaml file

```
kind: StorageClass apiVersion: storage.k8s.io/v1 metadata:
  name: "<NAME>"
  labels:
product: ibm-spectrum-scale-csi # annotations:
# storageclass.beta.kubernetes.io/is-default-class: "true" #reclaimPolicy:
"Retain" # Optional,
# Values: Delete[default] or Retain provisioner: spectrumscale.csi.ibm.com
parameters:
  volBackendFs: "<filesystem name>"
  volDirBasePath: "<fileset name>"
  # uid: "<uid number>" # Optional
  # gid: "<gid number>" # Optional
```

The configuration parameters that are in the storage class template yaml file are listed in Table 5.

Table 5 Configuration parameters in storage class template yaml

Parameter	Description
Name	Storage class name.
volBackendFS	IBM Spectrum Scale file system name for creating volumes.
Fileset-types	Optional parameter. Type of file set to be created for a volume. Permitted values: • Independent (default) • Dependent
dependentFileset	Parent file set name in the case of dependent fileset-type.
volDirBasePath	Base path under in which all volumes with this storage class are created. This path must exist.
Uid	Optional parameter. Owner to be set on the file set for newly created volume. User with specified Uid or name must exist on IBM Spectrum Scale.

Parameter	Description
Gid	Optional parameter. Group owner to be set on the file set for newly created volume. Must be specified along with UID. Group with specified Gid or name must exist on IBM Spectrum Scale.
Inode-limit	Optional parameter Number of inodes to be preallocated for newly created file set.
isPreexisting	Optional parameter Used to indicate whether to use existing file set or to create a file set for volume. Permitted values: • False (defaut) • True If true is specified, user must set pv-name parameter while creating PVC.
Type	Permanently set to file set.
Product	Permanently set to ibm-spectrum-scale-csi.
Provisioner	Permanently set to spectrumscale.csi.ibm.com.

Deploying CockroachDB Operator and Database instance

This section provides detailed steps for deploying the CockroachDB Operator by using the Red Hat OpenShift Container Platform OperatorHub, and for properly provisioning persistent storage for the CockroachDB StatefulSet.

Creating a subscription to the CockroachDB Operator

To create a subscription, complete the following steps:

1. In the OpenShift Container Platform console select **Console** → **OperatorHub**.

2. Select **CockroachDB** from the available Database Operators (see Figure 20).

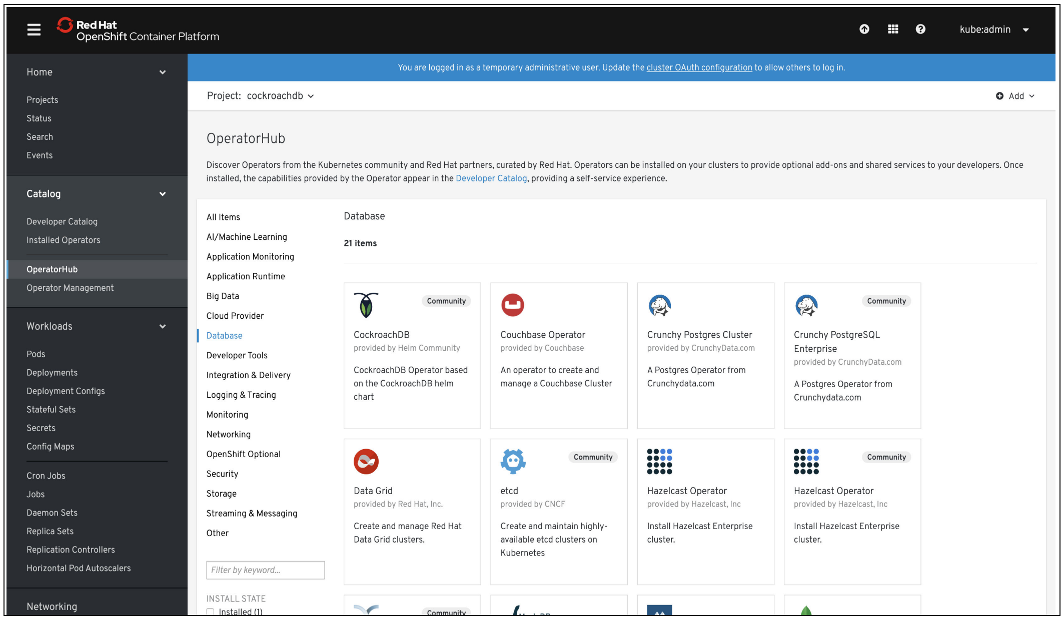

Figure 20 CockroachDB OperatorHub

3. Accept the warning about Community provided Operators.

4. Select **Install** (see Figure 21).

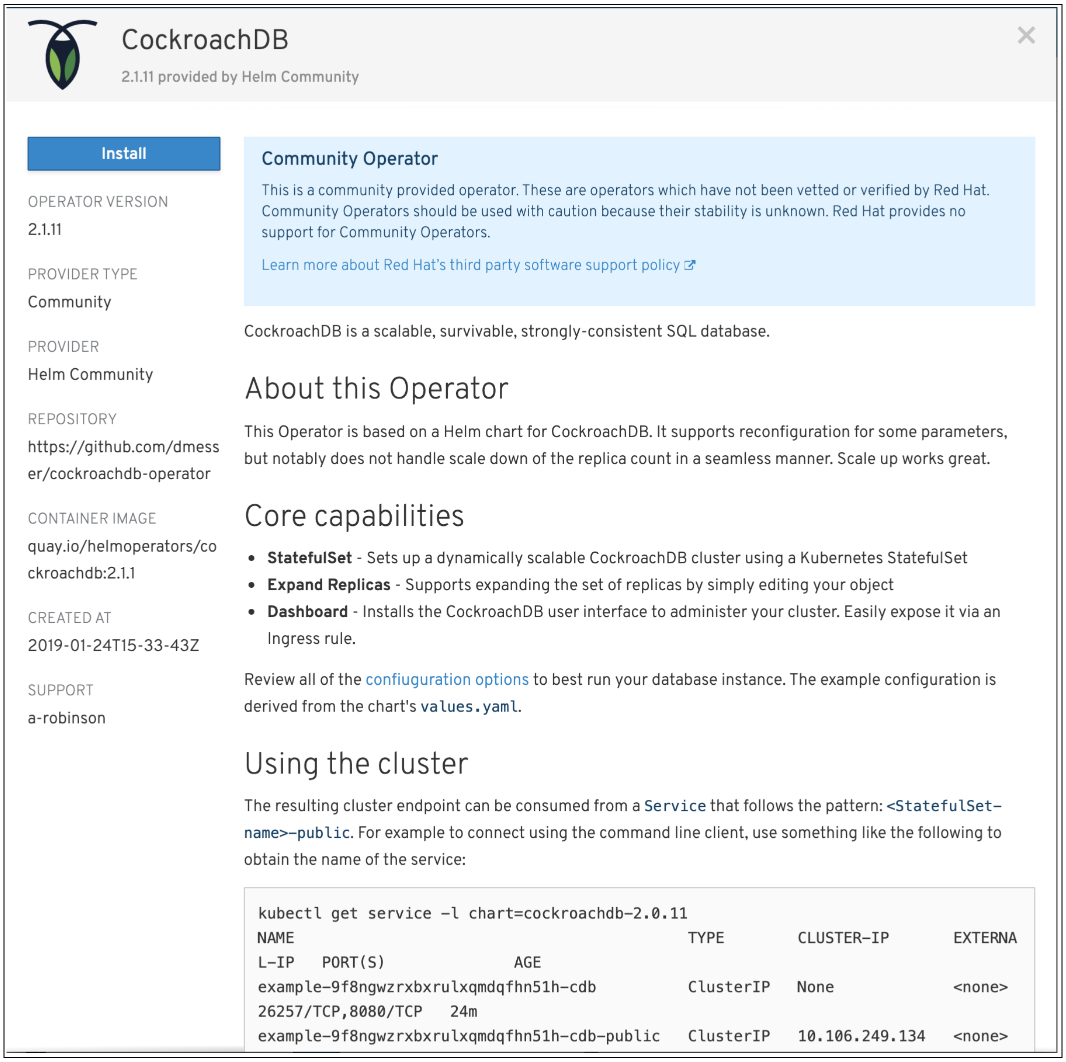

Figure 21 Installing CockroachDB

5. Select **Subscribe** to create a subscription to the CockroachDB Operator (see Figure 22).

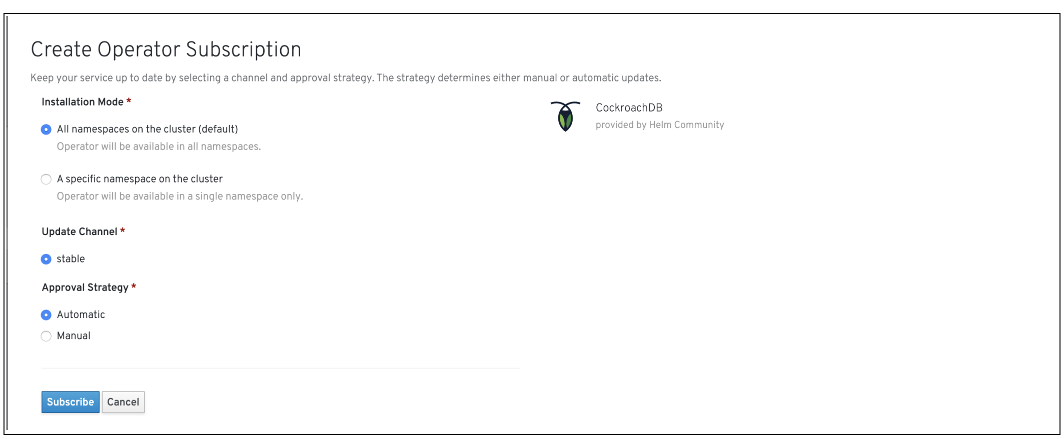

Figure 22 Creating a subscription for CockroachDB Operator

6. Wait for Upgrade status to show 1 installed (see Figure 23).

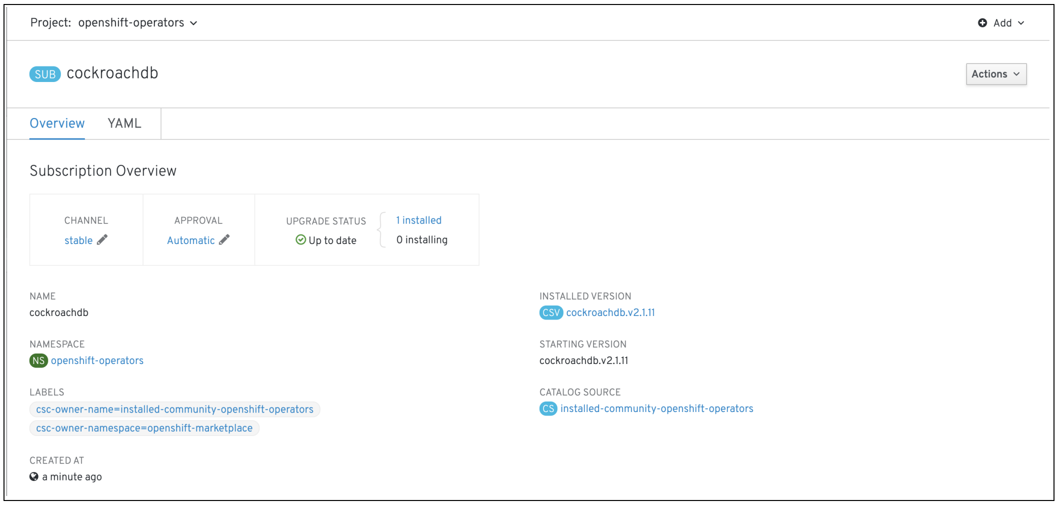

Figure 23 Installing Operator complete

Using the CockroachDB operator to deploy a StatefulSet

To deploy a StatefulSet, complete the following steps:

1. Select **Installed Operators** from the side bar.
2. From the drop-down menu, select the namespace to which you want to deploy the application. In the lab environment, we used the `cockroachdb` namespace.
3. Select **CockroachDB** to start (see Figure 24).

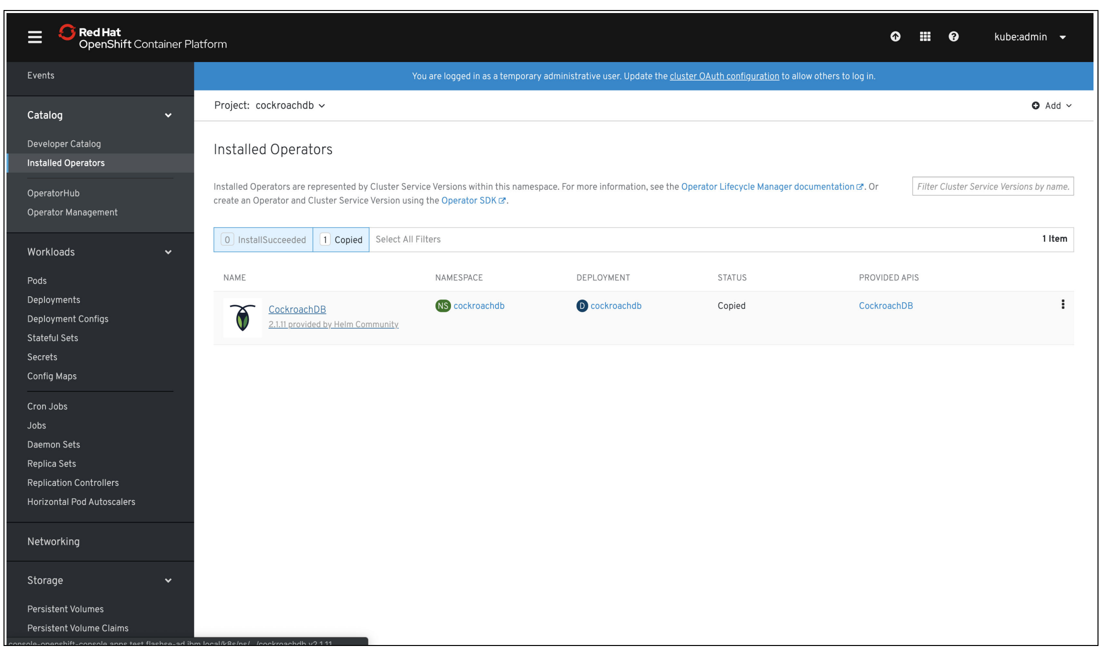

Figure 24 Installed Operators

4. Select **Create New** to enter options for the application (see Figure 25).

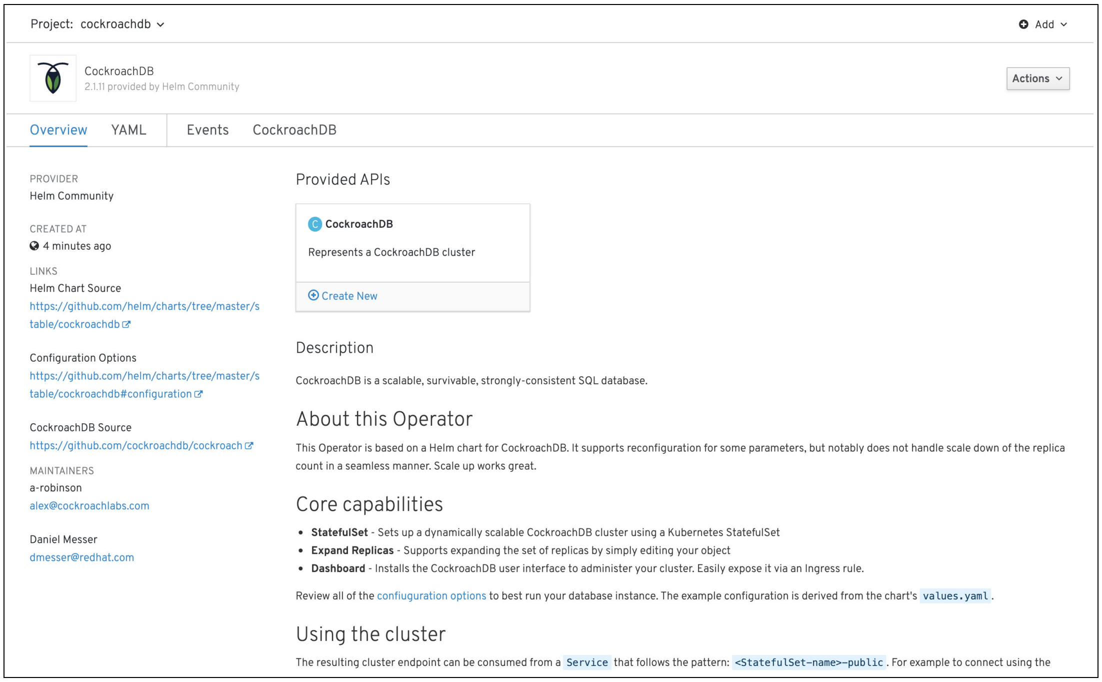

Figure 25 CockroachDB Operator

In our lab, we updated the `spec.StorageClass` from null to `ibmc-block-flashsystem`.

5. Select **Create** to deploy the application.

Verifying successful deployment of CockroachDB

To verify deployment, complete the following steps:

1. Select the newly deployed Cockroachdb. The default name is **example**, as shown in Figure 26.

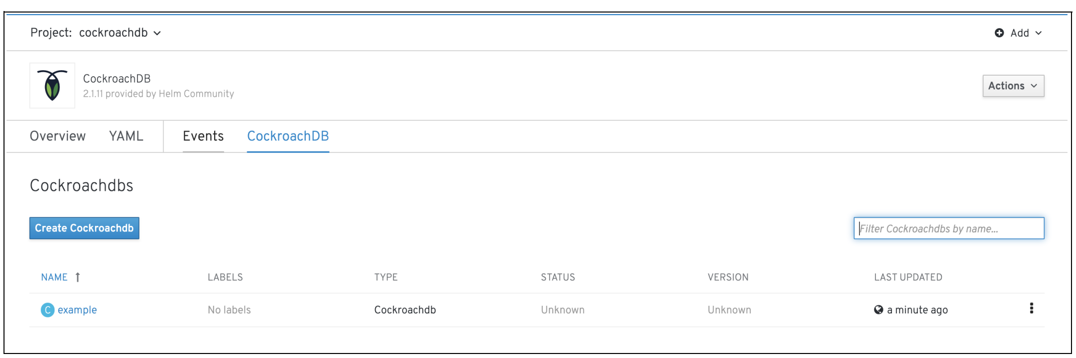

Figure 26 Installed CockroachDB example

2. Select **Resources** to see that two Services and a StatefulSet were created (see Figure 27).

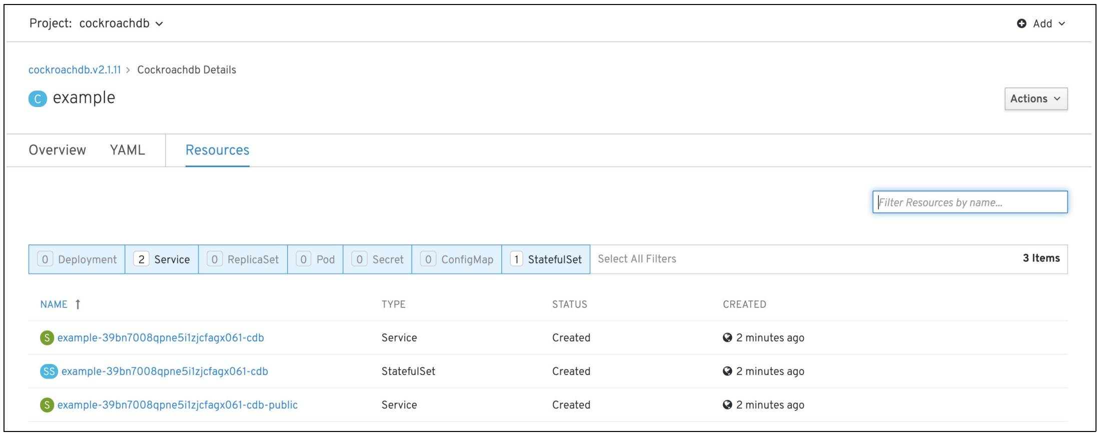

Figure 27 CockroachDB Resources

3. Select the Statefulset and then, navigate to the pods (see Figure 28).

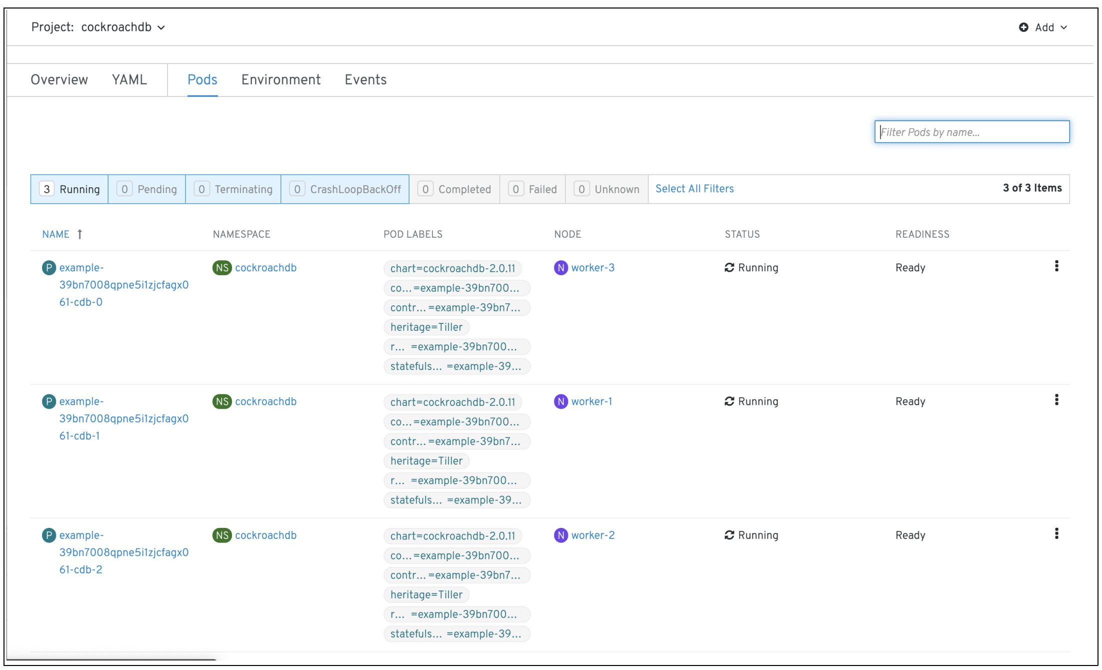

Figure 28 CockroachDB Statefulset Pods

4. Wait for the three pods to become Ready.
5. Select **Storage** → **Persistent Volume Claims**.

6. Select one of the PVCs and see that it is bound to a PersistentVolume (see Figure 29).

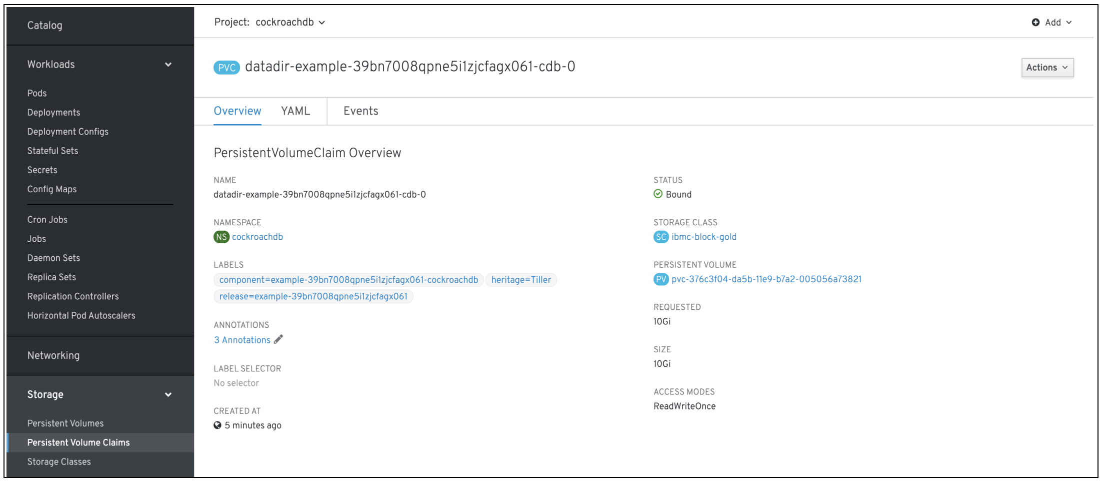

Figure 29 PVC information

Creating volume snapshots with IBM Block CSI Driver

Volume snapshots allow for the IBM Block CSI Driver to provision FlashCopy® Snapshots and use them for backup, testing, and other purposes. In this example, we use the database that we provisioned in the Deploying CockroachDB Operator and database instance.

> **Note:** IBM® Block CSI Driver requires OpenShift Container Platform 4.4 or Kubernetes 1.17 to support volume snapshots. Volume snapshot capabilities use the beta API and do not support the alpha API of the Snapshot CSI Specification.

Complete the following steps:

1. Create a `VolumeSnapshotClass`, as shown in Example 8.

Example 8 ibm-block-flashsystem-snapshot.yml file

```
apiVersion: snapshot.storage.k8s.io/v1beta1
kind: VolumeSnapshotClass
  metadata:
    name: ibm-block-flashsystem-snapshot
driver: block.csi.ibm.com
deletionPolicy: Delete # Retain or Delete
parameters:
  csi.storage.k8s.io/snapshotter-secret-name: flashsystem
  csi.storage.k8s.io/snapshotter-secret-namespace: ibm-block-csi
#snapshot_name_prefix: <NAME_PREFIX> # Optional.
```

2. Create a `VolumeSnapShot` of one of the database `persistentVolumeClaims`, as shown in Example 9.

Example 9 datadir-example-cockroachdb-0.snapshot.yaml file

```
apiVersion: snapshot.storage.k8s.io/v1beta1
kind: VolumeSnapshot
metadata:
  name: datadir-example-cockroachdb-0-snapshot
driver: block.csi.ibm.com
deletionPolicy: Delete # Retain or Delete
spec:
  volumeSnapshotClassName: ibm-block-flashsystem-snapshot
  source:
    persistentVolumeClaimName: datadir-example-cockroachdb-0
```

3. Verify that `volumeSnapshot` and `volueSnapShotContents` were created, as shown in the following example:

```
oc get volumeSnapshots datadir-example-cockroachdb-0-snapshot
oc get volumeSnapshotContents <snapcontent listed from above command>
```

4. Create a Volume Clone of one of the `volumeSnapshots` (see Figure 30).

   Volume Clones of `SnapshotContents` allow for the IBM Block CSI Driver to provision FlashCopy Clones and use them for backing up, testing, and other purposes. As shown in Example 10 on page 37, we use the snapshot that we created earlier.

Example 10 datadir-example-cockroachdb-0-snapshot-pvc.yml file

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: datadir-example-cockroachdb-0-snapshot-pvc
spec:
  storageClassName: ibm-block-flashsystem
  dataSource:
    name: datadir-example-cockroachdb-0-snapshot
    kind: VolumeSnapshot
    apiGroup: snapshot.storage.k8s.io
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 10Gi
Apply the new volumeSnapshot :
oc create -f datadir-example-cockroachdb-0-snapshot-pvc.yml
```

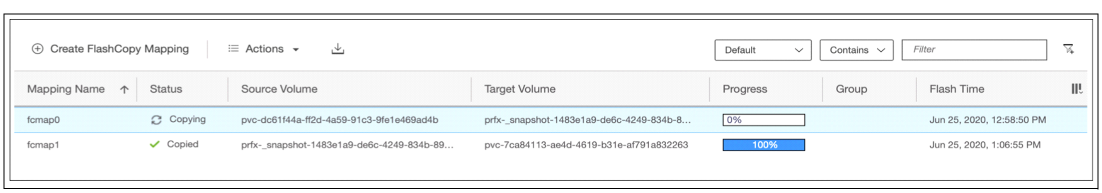

Figure 30 volumeSnapshot and associated persistentVolumeClaim

Appendix A: DNS configuration

```
[root@dns named]# cat /etc/named.conf
options{
  directory "/var/named";
  forwarders {
    1.1.1.1;
    2.2.2.2;
    };
};

zone "3.168.192.IN-ADDR.ARPA." IN {
  type master;
  file "192.168.3.db";
};

zone "isvcluster.net." {
  type master;
  file "db.isvcluster.net";
};


---------------------------------------------------------

[root@dns named]# cat /var/named/db.isvcluster.net

$ORIGIN .
$TTL 3600; 1 hour
isvcluster.netIN SOA isvcluster.net. root.isvcluster.net. (
            2020082574 ; serial
            10800      ; refresh (3 hours)
            3600       ; retry (1 hour)
            604800     ; expire (1 week)
            3600       ; minimum (1 hour)
            )
         NS dns.isvcluster.net.

$ORIGIN isvcluster.net.
rhel-host     IN  1H  A 192.168.3.161

$ORIGIN openshift.isvcluster.net
rhel-host     IN  1H  A 192.168.3.161

$ORIGIN isvsol.openshift.isvcluster.net
rhel-host     IN  1H  A 192.168.3.161
api           IN  1H  A 192.168.3.162

$ORIGIN apps.isvsol.openshift.isvcluster.net
*             IN  1H  A 192.168.3.163

-----------------------
[root@dns named]# cat 192.168.3.db

$TTL 3600; 1 hour
@ SOA isvcluster.net. root.isvcluster.net.(
```

```
        2020082574 ; serial
        10800      ; refresh (3 hours)
        3600       ; retry (1 hour)
        604800     ; expire (1 week)
        3600       ; minimum (1 hour)
        )
        NS dns.isvcluster.net.

161     PTR rhel-host.openshift.isvcluster.net.
162     PTR api.isvsol.openshift.isvcluster.net.
163     PTR ingress.isvsol.openshift.isvcluster.net.

-------------------------------------

[root@dns named]# cat /etc/dhcp/dhcpd.conf
option domain-name "isvcluster.net";
option domain-name-servers dns.isvcluster.net;
default-lease-time 86400;
max-lease-time 604800;
authoritative;
log-facility local7;
subnet 192.168.3.0 netmask 255.255.255.0 {
    option routers 192.168.3.10;
    option subnet-mask 255.255.255.0;
    option domain-search "openshift.isvcluster.net";
    option domain-name-servers 192.168.3.80;
    option time-offset -18000; # Eastern Standard Time
    range 192.168.3.164 192.168.3.180;
}

host rhel-host {
  option host-name "rhel-host.openshift.isvcluster.net";
  hardware ethernet 00:50:56:AF:DA:CC;
  fixed-address 192.168.3.161;
}
```

Appendix B: DHCP configuration

```
[root@dns named]# cat /etc/dhcp/dhcpd.conf
option domain-name "isvcluster.net";
option domain-name-servers dns.isvcluster.net;
default-lease-time 86400;
max-lease-time 604800;
authoritative;
log-facility local7;
subnet 192.168.3.0 netmask 255.255.255.0 {
    option routers 192.168.3.10;
    option subnet-mask 255.255.255.0;
    option domain-search "openshift.isvcluster.net";
    option domain-name-servers 192.168.3.80;
    option time-offset -18000; # Eastern Standard Time
    range 192.168.3.164 192.168.3.180;
```

```
}

host rhel-host {
  option host-name "rhel-host.openshift.isvcluster.net";
  hardware ethernet 00:50:56:AF:DA:CC;
  fixed-address 192.168.3.161;
}
```

Appendix C: Firewall rules

Allow communication between boot and master nodes haproxy, HTTP, DNS, and DHCP systems. In the lab setup the three daemons (HTTP, DNS, and DHCP) were running on the same system. The following firewall rules allowed connections to each of these daemons:

- RH-DNS-410

  ```
  firewall-cmd --permanent --add-service=dns
  firewall-cmd --permanent --add-service=http
  firewall-cmd --permanent --add-service=dhcp
  ```

- RH-HOST-410

  ```
  firewall-cmd --permanent --add-port=80/tcp
  firewall-cmd --permanent --add-port=443/tcp
  firewall-cmd --permanent --add-port=6443/tcp
  firewall-cmd --permanent --add-port=10256/tcp
  firewall-cmd --permanent --add-port=22623/tcp
  firewall-cmd --permanent --add-port=2379-2380/tcp
  firewall-cmd --permanent --add-port=9000-9999/tcp
  firewall-cmd --permanent --add-port=10249-10259/tcp
  firewall-cmd --permanent --add-port=30000-32767/tcp
  ```

Appendix D: IBM Spectrum Scale usage restrictions

Consider the following conditions before using IBM Spectrum Scale CSI Driver with IBM Spectrum Scale:

- IBM Spectrum Scale storage tiering/pooling is not surfaced to the PVC/POD level. To gain a similar look and feel as the IBM Spectrum Scale storage tiering, it is advised to create a separate IBM Spectrum Scale file system on each type of wanted storage (such as flash, SSD, and SAS). After that file system is created, associate file system, fileset-type, and storage classes in the `storage-class-template.yml` and `pvc-template.yml` files. This association enables multiple types of storage to be presented to the pods by way of multiple PVCs.
- IBM Spectrum Scale must be pre-installed with the IBM Spectrum Scale GUI.
- IBM Spectrum Scale CSI Driver is supported on IBM Spectrum Scale V5.0.4.1 and later only.
- At least one file system must exist and be mounted on all of the worker nodes.
- IBM Spectrum Scale Container Storage Interface Driver does not use the size that is specified in PersistentVolumeClaim for lightweight volume and dependent fileset volume.
- Quota must be enabled for all of the file systems that are used for creating persistent volumes.

- All Red Hat OpenShift worker nodes must have the IBM Spectrum Scale client installed on them.
- The maximum number of supported volumes that can be created by using the independent fileset storage class is 998 (excluding the root fileset and primary fileset reserved for the IBM Spectrum Scale Container Storage Interface Driver). For more information, see IBM Knowledge Center.
- IBM Spectrum Scale Container Storage Interface Driver relies on the GUI server for performing IBM Spectrum Scale operations. If the GUI password or CA certificate expires, manual intervention is needed by the administrator to reset the password on the GUI or generate a certificate and update the configuration in IBM Spectrum Scale Container Storage Interface Driver.
- IBM Spectrum Scale Container Storage Interface Driver does not support mounting volumes in read-only mode.
- Red Hat OpenShift nodes should be configured to schedule pods after the IBM Spectrum Scale file system is mounted on worker nodes. This process can be monitored by **`<mmlsmount all>`** and it is recommended to script Red Hat OpenShift startup based on the return code and `systemd` results. If scripting off of this command, use the **`-Y`** parameter because it can be parsed and the formatting is consistent from release to release.
- If the IBM Spectrum Scale file system is unmounted (or if an issue exists with IBM Spectrum Scale mounted on a particular node), the applications in the containers that are using the PVC from IBM Spectrum Scale return an I/O error. IBM Spectrum Scale CSI Driver does not monitor IBM Spectrum Scale and is unaware of any failure in the I/O path.

  Kubernetes also does not monitor IBM Spectrum Scale and is unaware of any failure in the I/O path. It is recommended to monitor IBM Spectrum Scale to avoid any issues. Monitoring can be accomplished by way of scripting, such as an IBM General Parallel File System (GPFS) callback set to take action if the GPFS file system is unmounted or shut down (such as through script-based activation of cordon or drain node).
- If a single PVC is used by multiple pods, the application must maintain data consistency.
- Creating a large number of PVCs in a single batch, or deleting all of them simultaneously, is not recommended. Such actions might result in overloading the IBM Spectrum Scale GUI node, which in turn might lead to the failure of creation and deletion of file sets on IBM Spectrum Scale.
- The **`uid`**, **`gid`**, **`inode-limit`**, and **`fileset-type`** parameters from the storage-classes are allowed only for new fileset creation.
- For each **`uid-gid`** combination, a new storage class must be defined.
- Advanced IBM Spectrum Scale functionality, such as active file management, remote mount, encryption, and compression, are not supported by IBM Spectrum Scale CSI Driver.
- The persistent volumes that are created by using IBM Spectrum Scale CSI Driver with IBM Spectrum Scale as the back-end use the IBM Spectrum Scale quota to ensure that the users cannot use more storage space than the amount that is specified in the PVC. However, this does not guarantee that the storage specified in the PVC is available. The storage administrator must ensure that the required storage is available on the IBM Spectrum Scale file system.
- IBM Spectrum Scale CSI Driver does not check the storage space that is available on the IBM Spectrum Scale file system before creating the PVC. You can use the Kubernetes storage resource quota to limit the number of PVCs or storage space.
- The file set that is created by the storage IBM Spectrum Scale CSI Driver should not be unlinked or deleted from any other interface.

- The file system that is used for the persistent volume must be mounted on all worker nodes at all times.
- Red Hat OpenShift and IBM Spectrum Scale GUI use port 443.
- IBM Spectrum Scale CSI Driver does not support volume expansion for storage class.
- The **df** command inside the container shows the full size of the IBM Spectrum Scale file system.

Notices

This information was developed for products and services offered in the US. This material might be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:
IBM Director of Licensing, IBM Corporation, North Castle Drive, MD-NC119, Armonk, NY 10504-1785, US

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you provide in any way it believes appropriate without incurring any obligation to you.

The performance data and client examples cited are presented for illustrative purposes only. Actual performance results may vary depending on specific configurations and operating conditions.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to actual people or business enterprises is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at "Copyright and trademark information" at http://www.ibm.com/legal/copytrade.shtml

The following terms are trademarks or registered trademarks of International Business Machines Corporation, and might also be trademarks or registered trademarks in other countries.

Enterprise Storage Server®
FlashCopy®
IBM®
IBM Cloud®
IBM Elastic Storage®
IBM FlashSystem®
IBM Spectrum®
Redbooks (logo) ®
Storwize®

The following terms are trademarks of other companies:

The registered trademark Linux® is used pursuant to a sublicense from the Linux Foundation, the exclusive licensee of Linus Torvalds, owner of the mark on a worldwide basis.

Ansible, Ceph, OpenShift, Red Hat, are trademarks or registered trademarks of Red Hat, Inc. or its subsidiaries in the United States and other countries.

VMware, VMware vSphere, and the VMware logo are registered trademarks or trademarks of VMware, Inc. or its subsidiaries in the United States and/or other jurisdictions.

Other company, product, or service names may be trademarks or service marks of others.

Terms and conditions for product documentation

Permissions for the use of these publications are granted subject to the following terms and conditions.

Applicability

These terms and conditions are in addition to any terms of use for the IBM website.

Commercial use

You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of IBM.

Rights

Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.

Privacy policy considerations

IBM Software products, including software as a service solutions, ("Software Offerings") may use cookies or other technologies to collect product usage information, to help improve the end user experience, to tailor interactions with the end user, or for other purposes. In many cases no personally identifiable information is collected by the Software Offerings. Some of our Software Offerings can help enable you to collect personally identifiable information. If this Software Offering uses cookies to collect personally identifiable information, specific information about this offering's use of cookies is set forth below.

This Software Offering does not use cookies or other technologies to collect personally identifiable information.

If the configurations deployed for this Software Offering provide you as customer the ability to collect personally identifiable information from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent.

For more information about the use of various technologies, including cookies, for these purposes, see IBM's Privacy Policy at http://www.ibm.com/privacy and IBM's Online Privacy Statement at http://www.ibm.com/privacy/details in the section entitled “Cookies, Web Beacons and Other Technologies,” and the “IBM Software Products and Software-as-a-Service Privacy Statement” at http://www.ibm.com/software/info/product-privacy.

December 2020

ISBN 0738459275

REDP-5565-05