IBM Storage Enabler for Containers
Version 2.1.0

User Guide

SC27-9294-04

Note

Before using this document and the product it supports, read the information in "Notices" on page 53.

Edition notice

Publication number: SC27-9294-04. This publication applies to version 2.1.0 of IBM Storage Enabler for Containers and to all subsequent releases and modifications until otherwise indicated in a newer publication.

Contents

Figures

Tables

About this guide

This guide describes how to install, configure, and use IBM Storage Enabler for Containers.

Who should use this guide

This guide is intended for system administrators who are familiar with container-based application delivery, orchestration methods, and with the specific IBM storage system that is in use.

Conventions used in this guide

These notices are used in this guide to highlight key information.

Note: These notices provide important tips, guidance, or advice.

Important: These notices provide information or advice that might help you avoid inconvenient or difficult situations.

⚠ **Attention:** These notices indicate possible damage to programs, devices, or data. An attention notice appears before the instruction or situation in which damage can occur.

Related information and publications

You can find additional information and publications related to IBM Storage Enabler for Containers on the following information sources.

- IBM Knowledge Center (ibm.com/support/knowledgecenter)
- IBM DS8000® on IBM Knowledge Center (ibm.com®/support/knowledgecenter/STUVMB)
- IBM DS8800 on IBM Knowledge Center (ibm.com/support/knowledgecenter/STXN8P)
- IBM DS8870 on IBM Knowledge Center (ibm.com/support/knowledgecenter/ST8NCA)
- IBM FlashSystem 900 on IBM Knowledge Center (ibm.com/support/knowledgecenter/STKMQB)
- IBM SAN Volume Controller on IBM Knowledge Center (ibm.com/support/knowledgecenter/STPVGU)
- IBM Spectrum Scale on IBM Knowledge Center (ibm.com/support/knowledgecenter/STXKQY)
- IBM Storwize® V3500 on IBM Knowledge Center (ibm.com/support/knowledgecenter/STLM6B)
- IBM Storwize V3700 on IBM Knowledge Center (ibm.com/support/knowledgecenter/STLM5A)
- IBM Storwize V5000 on IBM Knowledge Center (ibm.com/support/knowledgecenter/STHGUJ)
- IBM Storwize V7000 on IBM Knowledge Center (ibm.com/support/knowledgecenter/ST3FR7)
- IBM Storwize V7000 Unified on IBM Knowledge Center (ibm.com/support/knowledgecenter/ST5Q4U)
- IBM XIV® Storage System on IBM Knowledge Center (ibm.com/support/knowledgecenter/STJTAG)
- IBM Spectrum Accelerate on IBM Knowledge Center (ibm.com/support/knowledgecenter/STZSWD)
- IBM FlashSystem® A9000 on IBM Knowledge Center (ibm.com/support/knowledgecenter/STJKMM)
- IBM FlashSystem A9000R on IBM Knowledge Center (ibm.com/support/knowledgecenter/STJKN5)
- Persistent volumes on Kubernetes (kubernetes.io/docs/concepts/storage/volumes)
- IBM Cloud Private (ibm.com/cloud/private)

Getting information, help, and service

If you need help, service, technical assistance, or want more information about IBM products, you can find various sources to assist you. You can view the following websites to get information about IBM products and services and to find the latest technical information and support.

- IBM website (ibm.com)
- IBM Support Portal website (ibm.com/support/entry/portal/support?brandind=Hardware~System_Storage)
- IBM Directory of Worldwide Contacts website (ibm.com/planetwide)

  Use the Directory of Worldwide Contacts to find the appropriate phone number for initiating voice call support. Select the Software option, when using voice response system.

  When asked, provide your Internal Customer Number (ICN) and/or the serial number of the storage system that requires support. Your call will then be routed to the relevant support team, to whom you can provide the specifics of your problem.

IBM Publications Center

The IBM Publications Center is a worldwide central repository for IBM product publications and marketing material.

The IBM Publications Center website (ibm.com/shop/publications/order) offers customized search functions to help you find the publications that you need. You can view or download publications at no charge.

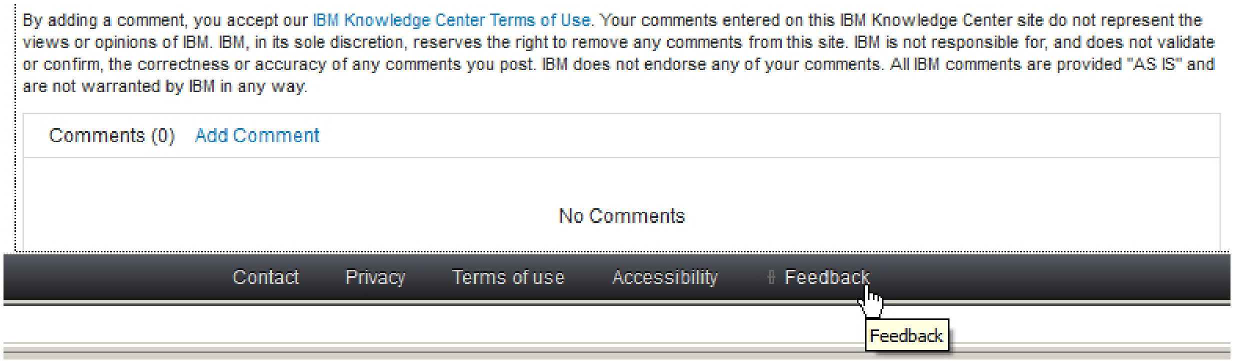
Sending or posting your comments

Your feedback is important in helping to provide the most accurate and highest quality information.

Procedure

To submit any comments about this guide:

- Go to IBM Spectrum Connect on IBM Knowledge Center (ibm.com/support/knowledgecenter/SS6JWS), drill down to the relevant page, and then click the **Feedback** link that is located at the bottom of the page.

  By adding a comment, you accept our IBM Knowledge Center Terms of Use. Your comments entered on this IBM Knowledge Center site do not represent the views or opinions of IBM. IBM, in its sole discretion, reserves the right to remove any comments from this site. IBM is not responsible for, and does not validate or confirm, the correctness or accuracy of any comments you post. IBM does not endorse any of your comments. All IBM comments are provided "AS IS" and are not warranted by IBM in any way.

  The feedback form is displayed and you can use it to enter and submit your comments privately.
- You can post a public comment on the Knowledge Center page that you are viewing, by clicking **Add Comment**. For this option, you must first log in to IBM Knowledge Center with your IBM ID.
- You can send your comments by email to starpubs@us.ibm.com. Be sure to include the following information:
  - Exact publication title and product version

- Publication form number (for example: SC01-0001-01)
- Page, table, or illustration numbers that you are commenting on
- A detailed description of any information that should be changed

Note: When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

Chapter 1. Introduction

IBM Storage Enabler for Containers allows IBM storage systems to be used as persistent volumes for stateful applications running in Kubernetes clusters.

IBM Storage Enabler for Containers is based on an open-source IBM project, Ubiquity. Through the IBM Storage Enabler for Containers, Kubernetes persistent volumes (PVs) can be provisioned from IBM storage. Thus, IBM storage can be accessed by containers and used with stateful microservices, such as database applications (MongoDB, PostgreSQL etc).

IBM Storage Enabler for Containers uses Kubernetes dynamic provisioning for creating and deleting volumes on IBM storage systems. For details about volume provisioning with Kubernetes, refer to Persistent volumes on Kubernetes (kubernetes.io/docs/concepts/storage/volumes). In addition, IBM Storage Enabler for Containers utilizes the full set of Kubernetes FlexVolume APIs for volume operations on a host. The operations include initiation, attachment/detachment, mounting/unmounting etc.

Note: For the user convenience, this guide might refer to IBM Storage Enabler for Containers as Enabler for Containers.

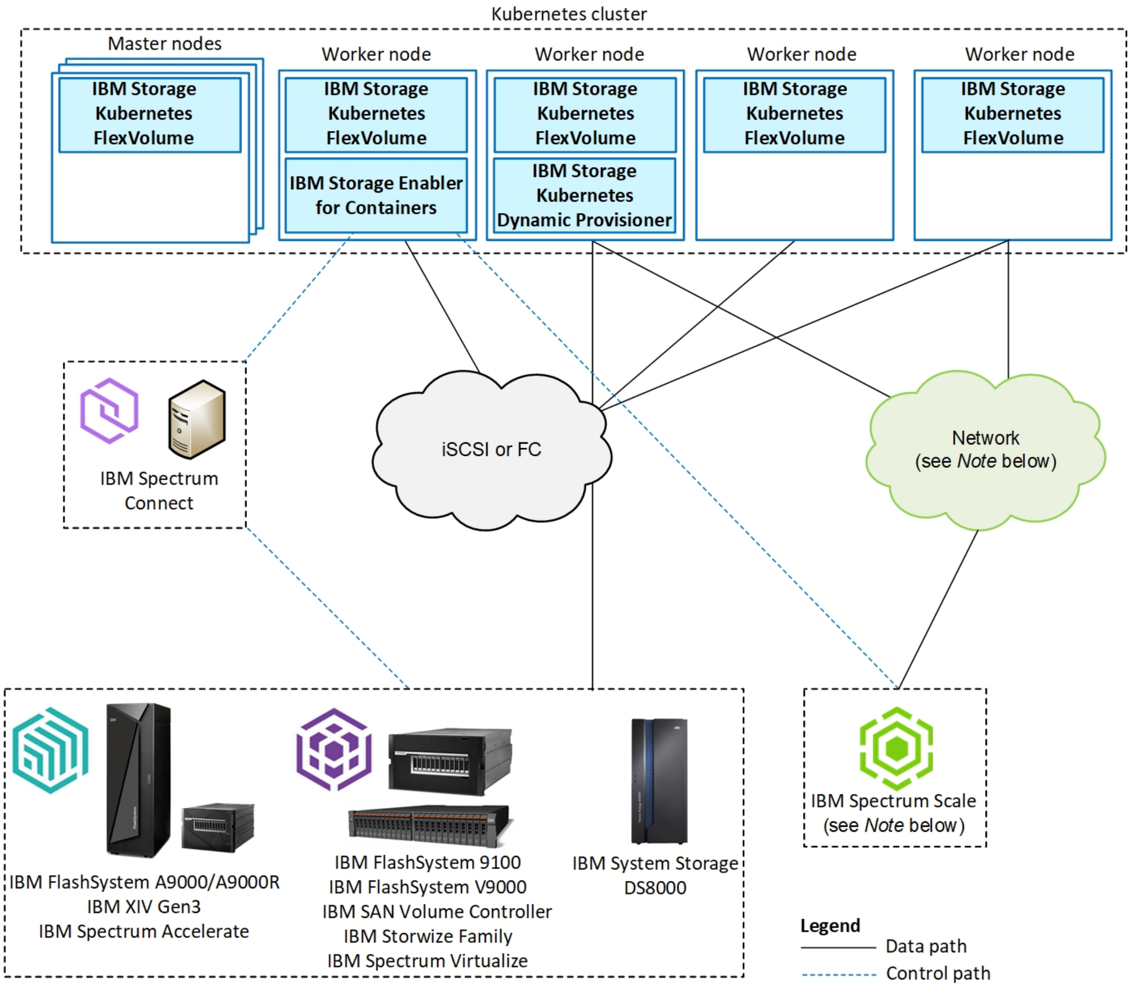

Kubernetes cluster
Master nodes
IBM Storage Kubernetes FlexVolume
Worker node
IBM Storage Kubernetes FlexVolume
IBM Storage Enabler for Containers
Worker node
IBM Storage Kubernetes FlexVolume
IBM Storage Kubernetes Dynamic Provisioner
Worker node
IBM Storage Kubernetes FlexVolume
Worker node
IBM Storage Kubernetes FlexVolume
IBM Spectrum Connect
iSCSI or FC
Network (see *Note* below)
IBM FlashSystem A9000/A9000R
IBM XIV Gen3
IBM Spectrum Accelerate
IBM FlashSystem 9100
IBM FlashSystem V9000
IBM SAN Volume Controller
IBM Storwize Family
IBM Spectrum Virtualize
IBM System Storage DS8000
IBM Spectrum Scale (see *Note* below)
Legend
Data path
Control path

Figure 1. Integration of IBM block storage systems and IBM Spectrum Scale in Kubernetes environment

Note:

- IBM Spectrum Scale uses the network for communication between nodes in the Spectrum Scale cluster, as well as for accessing IBM Spectrum Connect. For example, Spectrum Connect uses the RESTful API, running on the IBM Spectrum Scale GUI node.
- IBM Spectrum Scale must be installed on the worker nodes that access Spectrum Scale filesystems. Additional Spectrum Scale nodes might exist outside of the Kubernetes cluster. For example, the Spectrum Scale GUI and Spectrum Scale I/O servers might exist outside of the Kubernetes cluster. In this case, the storage from these nodes might be made available directly to worker nodes that run IBM Spectrum Scale.
- Currently, only one backend (block storage or IBM Spectrum Scale) can be configured on a single Kubernetes cluster via IBM Storage Enabler for Containers.

IBM block storage

IBM Storage Enabler for Containers allows IBM block storage systems to be used as persistent volumes for stateful application running in Kubernetes clusters.

Taking full advantage of the flexible service-based storage provisioning model in IBM Spectrum Connect, IBM Storage Enabler for Containers employs the storage profile (service) policy. This allows defining specific capabilities per storage service and creating a storage volume according to these requirements. This policy-driven approach helps the Kubernetes administrators easily define Kubernetes storage classes, such as gold, silver or bronze, by using the Spectrum Connect services. Only storage administrators deal with storage systems and manage IBM Spectrum Connect. These storage administrators create storage services, based on required storage attributes and capacities, according pre-defined SLAs. Then, these services are delegated to Kubernetes administrators. In their turn, the Kubernetes administrators easily consume the storage services by creating corresponding storage classes in Kubernetes without the need to know anything about underlying storage and without using IBM Spectrum Connect itself.

For details, see the IBM Spectrum Connect user guide and release notes on IBM Knowledge Center.

The IBM Storage Enabler for Containers ensures that the data persists (stays intact) even after the container is stopped or removed. The IBM Storage Enabler communicates with the IBM block storage systems through Spectrum Connect. Spectrum Connect creates a storage service (for example, gold, silver or bronze) and makes it available for Kubernetes Dynamic Provisioner and FlexVolume, automating IBM block storage provisioning for Kubernetes persistent volumes.

- The Dynamic Provisioner allows storage volumes to be created on-demand, using Kubernetes storage classes based on Spectrum Connect storage services. This provides abstraction for the underlying storage platform, eliminating the need for cluster administrators to pre-provision storage.
- The FlexVolume is deployed as a DaemonSet on all nodes of the cluster, enabling the users to attach and mount storage volumes into a pod within a Kubernetes node. The DaemonSet installs the FlexVolume CLI on every node in the cluster in the Kubernetes plug-in directory.

Note: The instances of IBM Storage Enabler for Containers (*ubiquity*), its database (*ubiquity-db*) and IBM Storage Kubernetes Dynamic Provisioner (*ubiquity-k8s-provisioner*) are protected, using standard Kubernetes methods for high-availability. They are deployed as Kubernetes Deployment objects with replica=1, so if a node fails, Kubernetes automatically reschedules them to run on another node. IBM Storage Kubernetes FlexVolume (*ubiquity-k8s-flex*) is deployed as a Kubernetes DaemonSet on all the worker and master nodes.

IBM Spectrum Scale

This section gives a brief introduction to IBM Spectrum Scale and IBM Storage Enabler for Containers.

IBM Spectrum Scale is a cluster file system that provides concurrent access to a single file system or set of file systems from multiple nodes. The nodes can be SAN-attached, network-attached, a mixture of SAN-attached and network-attached, or in a shared-nothing cluster configuration. This enables high performance access to this common set of data to support a scale-out solution or to provide a high-availability platform.

IBM Spectrum Scale has many features beyond common data access, including data replication, policy based storage management, and multi-site operations. You can create a cluster of AIX® nodes, Linux nodes, Windows server nodes, or a mix of all three.

IBM® Storage Enabler for Containers allows IBM Spectrum Scale to be used as a source for persistent volumes intended for stateful application running in Kubernetes clusters.

Note: Currently, not all platform features are fully functional, when IBM Spectrum Scale is deployed with IBM Storage Enabler for Containers.

Chapter 2. IBM block storage deployment

This section explains how to install or upgrade IBM Storage Enabler for Containers and integrate it into IBM Spectrum Connect. In addition, it provides the usage and troubleshooting instructions for this software package.

- "Installation" on page 5
- "Managing integration with IBM Spectrum Connect" on page 17
- "Using IBM Storage Enabler for Containers with IBM block storage" on page 20
- "Troubleshooting" on page 24

Installation

Download and install the IBM Storage Enabler for Containers in Kubernetes cluster or IBM Cloud Private environment as described in the following sections.

- "Compatibility and requirements" on page 5
- "Managing SSL certificates" on page 10
- "Performing installation" on page 11
- "Upgrading existing installation" on page 14
- "Rolling back to a previous revision" on page 16

For information about uninstallation, see "Uninstalling the software package" on page 16.

Compatibility and requirements

For the complete and up-to-date information about the compatibility and requirements for using IBM Storage Enabler for Containers with Kubernetes, refer to its latest release notes. The release notes detail supported operating system and Kubernetes versions, as well as microcode versions of the supported storage systems. You can find the latest release notes on IBM Knowledge Center (www.ibm.com/support/knowledgecenter/SSCKLT).

About this task

Follow these steps to prepare your environment for installing the IBM Storage Enabler for Containers in the Kubernetes cluster that requires persistent volumes for stateful containers.

Procedure

1. Contact your storage administrator and make sure that IBM Spectrum Connect has been installed; IBM Storage Enabler for Containers interface has been added to active Spectrum Connect instance; at least one storage service has been delegated to it. See "Managing integration with IBM Spectrum Connect" on page 17 and "Delegating storage services to the IBM Storage Enabler for Containers interface" on page 18 for details.
2. Verify that there is a proper communication link between Spectrum Connect and Kubernetes cluster.
3. Perform these steps for each worker node in Kubernetes cluster:

   a. Install the following Linux packages to ensure Fibre Channel and iSCSI connectivity. Skip this step, if the packages are already installed.

      - RHEL:
        - `sg3_utils`.
        - `iscsi-initiator-utils` (if iSCSI connection is required).
        - `sysfsutils` (if Fibre Channel connection is required).

```
sudo yum -y install sg3_utils
sudo yum -y install iscsi-initiator-utils
sudo yum -y install sysfsutils
```

- Ubuntu:
  - scsitools.
  - open-iscsi (if iSCSI connection is required).
  - sysfsutils (if Fibre Channel connection is required).

```
sudo apt-get install scsitools
sudo apt-get install open-iscsi
sudo apt-get install sysfsutils
```

b. Configure Linux multipath devices on the host. Create and set the relevant storage system parameters in the /etc/multipath.conf file. You can also use the default multipath.conf file located in the /usr/share/doc/device-mapper-multipath-* directory.
Verify that the systemctl status multipathd output indicates that the multipath status is active and error-free.

- RHEL:

```
yum install device-mapper-multipath
sudo modprobe dm-multipath
systemctl start multipathd
systemctl status multipathd
multipath -ll
```

- Ubuntu:

```
apt-get install multipath-tools
sudo modprobe dm-multipath
systemctl start multipathd
systemctl status multipathd
multipath -ll
```

- SLES:

Note: For SLES, the multipath-tools package version must be 0.7.1 or above.

```
zypper install sg3_utils multipath-tools
systemctl start multipathd
systemctl status multipathd
multipath -ll
```

Important: When configuring Linux multipath devices, verify that the **find_multipaths** parameter in the multipath.conf file is disabled.

- RHEL: Remove the **find_multipaths yes** string from the multipath.conf file.
- Ubuntu: Add the **find_multipaths no** string to the multipath.conf file, see below:

```
defaults {
  find_multipaths no
}
```

c. Configure storage system connectivity.

- Define the hostname of each Kubernetes node on the relevant storage systems with the valid WWPN or IQN of the node. The hostname on the storage system must be the same as the hostname defined in the Kubernetes cluster. Use the **$> kubectl get nodes** command to display hostname, as illustrated below. In this example, the k8s-worker-node1 and the k8s-worker-node2 hostnames must be defined on a storage system.

Note: In most cases, the local hostname of the node is the same as the Kubernetes node hostname as displayed in the **kubectl get nodes** command output. However, if the names are different, make sure to use the Kubernetes node name, as it appears in the command output.

```
root@k8s-user-v18-master:~# kubectl get nodes
NAME               STATUS   ROLES      AGE       VERSION
k8s-master         Ready     master    34d       v1.8.4
k8s-worker-node1   Ready     <none>    34d       v1.8.4
k8s-worker-node2   Ready     <none>    34d       v1.8.4
```

- After the node hostnames are defined, log into Spectrum Connect UI and refresh the relevant storage systems in the **Storage System** pane.
- For iSCSI, perform these three steps.
  - Make sure that the login used to log in to the iSCSI targets is permanent and remains available after a reboot of the worker node. To do this, verify that the **node.startup** in the /etc/iscsi/iscsid.conf file is set to *automatic*. If not, set it as required and then restart the iscsid service (**$> service iscsid restart**).
  - Discover and log into at least two iSCSI targets on the relevant storage systems.

```
$> iscsiadm -m discoverydb -t st -p ${storage system iSCSI port IP}:3260
 --discover
$> iscsiadm -m node  -p ${storage system iSCSI port IP/hostname} --login
```

  - Verify that the login was successful and display all targets that you logged in. The *portal* value must be the iSCSI target IP address.

```
$> iscsiadm -m session --rescan
Rescanning session [sid: 1, target: {storage system IQN},
portal: {storage system iSCSI port IP},{port number}
```

d. If using Kubernetes version lower than 1.12, make sure that the node *kubelet* service has the attach/detach capability enabled, **enable-controller-attach-detach=true** (enabled by default). To verify the current status, run the following command and check that the Setting node annotation to enable volume controller attach/detach message is displayed:

```
$> journalctl -u kubelet | grep 'Setting node annotation to .
* volume controller attach/detach' | tail -1
Jan 03 17:55:05 k8s-19-master-shay kubelet[3627]: I0103 17:55:05.437720 3627
kubelet_node_status.go:273] Setting node annotation to enable volume controller
attach/detach
```

If the volume controller attach/detach functionality is disabled, enable it, as detailed in Kubernetes documentation.

4. Perform these steps for every master node in Kubernetes cluster:

   a. Enable the attach/detach capability for the *kubelet* service (**controller-attach-detach-enabled=true**). It is enabled by default.

   b. For Kubernetes version lower than 1.12, if the controller-manager is configured to run as a pod in your Kubernetes cluster, you must allow for event recording in controller-manager log file. To achieve this, add the default path to the log file (/var/log), as a host path. You can change this directory by configuring **ubiquityK8sFlex.flexLogDir** parameter in the values.yml file.

   - Stop the controller-manager pod by moving the kube-controller-manager.yml file to temporary directory: **mv /etc/kubernetes/manifests/kube-controller-manager.yml /tmp**.
   - Edit the kube-controller-manager.yml file: **vi /tmp/kube-controller-manager.yml**.

- Add the following lines under the **volumes** tag.

```
- hostPath:
    path: /var/log
    type: DirectoryOrCreate
  name: flexlog-dir
```

- Add the following lines under the **volumeMounts** tag:

```
- mountPath: /var/log
  name: flexlog-dir
```

- Restart the controller-manager pod by moving the `kube-controller-manager.yml` file to its original location: **`mv /tmp/kube-controller-manager.yml /etc/kubernetes/manifests/`**.
- Verify that the controller-manager pod is in the Running state: **`kubectl get pod -n kube-system | grep controller-manager`**.

c. `flexvolume-dir` must be available within `kube-controller-manager`.

- Verify that `flexvolume-dir` (`/usr/libexec/kubernetes/kubelet-plugins/volume/exec`) is mounted inside `kube-controller-manager`.

  Use the **`$ kubectl describe pod <kube-controller-manager-pod-id> -n kube-system`** command to show the details of the `kube-controller-manager`, which includes the `flexvolume-dir` mount.

  The output should look as follows:

```
flexvolume-dir:
Type: HostPath (bare host directory volume)
Path: /usr/libexec/kubernetes/kubelet-plugins/volume/exec
HostPathType: DirectoryOrCreate
```

  If `flexvolume-dir` is not present, continue with the following steps.

- Stop the controller-manager pod by moving the `kube-controller-manager.yml` file to temporary directory: **`mv /etc/kubernetes/manifests/kube-controller-manager.yaml /tmp/kube-controller-manager.yaml`**.
- Edit the `/tmp/kube-controller-manager.yaml` file.
  - Add the following lines under the **volumeMounts** tag:

```
mountPath: /usr/libexec/kubernetes/kubelet-plugins/volume/exec
name: flexvolume-dir
```

  - Add the following lines under the **Volumes** tag:

```
hostPath:
path: /usr/libexec/kubernetes/kubelet-plugins/volume/exec
type: DirectoryOrCreate
name: flexvolume-dir
```

  - Restart the controller-manager pod by moving the `kube-controller-manager.yml` file to its original location: **`mv /tmp/kube-controller-manager.yml /etc/kubernetes/manifests/`**.
  - Verify that the controller-manager pod is in the Running state: **`kubectl get pod -n kube-system | grep controller-manager`**.

5. Define a namespace to be used for creating secrets.

- Kubernetes:

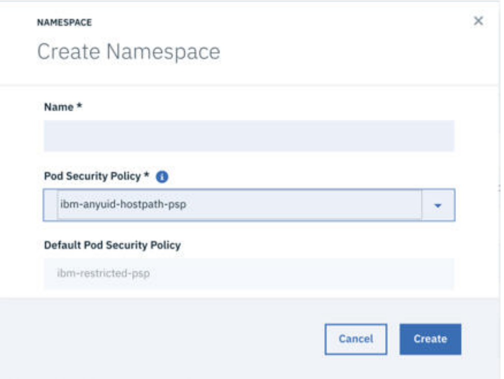
```
kubectl create ns <namespaces_name>
```

- ICP:

  a. In the ICP GUI, go to **Manage** > **Namespaces**.

  b. Click **Create Namespace**. In the **Create Namespace** dialog box, provide a namespace name and its pod security police.

     The recommended predefined pod security policy name is *ibm-anyuid-hostpath-psp*, and it has been verified for this Helm chart. If your target namespace is bound to this pod security policy, you can proceed with the chart installation. If you choose another pod security policy, you must enable the default pod security policy, and use the predefined cluster role: *ibm-anyuid-hostpath-clusterrole*.



Name *

Pod Security Policy *
ibm-anyuid-hostpath-psp

Default Pod Security Policy
ibm-restricted-psp

Figure 2. Create Namespace dialog box

6. Create two secrets: Enabler for Containers secret for its database and Enabler for Containers secret for the IBM Spectrum Connect (Verify that Spectrum Connect secret username and password are the same as Enabler for Containers interface username and password in Spectrum Connect UI.).

   - Kubernetes:

```
kubectl create secret generic <ubiquity_db_credentials_secret_name> --from-literal=dbname=ubiquity
 --from-literal=username=<username> --from-literal=password=<password>  -n <namespace>
kubectl create secret generic <ubiquity_scb_credentials_secret_name> --from-literal=username=<username>
 --from-literal=password=<password>  -n <namespace>
```

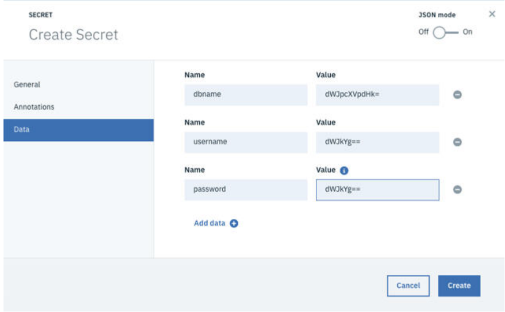
   - ICP:

     a. In the ICP GUI, go to **Configuration** > **Secrets**.

     b. Click **Create Secret**. In the **Create Secret** dialog box, provide the following values for the Enabler for Containers database:

        – In the **General** tab, select **Namespace**, and enter the namespace name, added in the previous step.

        – In the **Data** tab, add the Base64-encrypted **Name** values: *ubiquity*, *username* and *password*.



Name	Value
dbname	dWJpcXVpdHk=
username	dWJkYg==
password	dWJkYg==

Figure 3. Create Secret dialog box

c. Click **Create** to finish.

d. Repeat the secret creation procedure for the IBM Spectrum Connect secret:

   - In the **General** tab, select **Namespace**, and enter the namespace name, added in the previous step.
   - In the **Data** tab, add the Base64-encrypted **Name** values: *username* and *password*.

7. If dedicated SSL certificates are required, see the relevant section of the "Managing SSL certificates" on page 10 procedure. When no validation is required and you can use the self-signed certificates, generated by default by the IBM Storage Enabler for Containers server, skip this procedure.
8. When using IBM Cloud Private with the Spectrum Virtualize Family products, use only hostnames, and not IP addresses, for the Kubernetes cluster nodes. Then, in the `config.yaml` file, set the **kubelet_nodename** parameter to *hostname* to install the ICP nodes with hostnames as well.

Managing SSL certificates

IBM Storage Enabler for Containers uses SSL certificates for maintaining a secure communication link between the IBM Storage Enabler for Containers server, its database, the Dynamic Provisioner, the FlexVolume, and the Spectrum Connect server.

About this task

IBM Storage Enabler for Containers supports two SSL modes, when communicating with its components:

- *require*, when no validation is required. The IBM Storage Enabler for Containers server generates self-signed certificates on the fly. In this mode, you can skip the procedure detailed below and continue with the installation of the IBM Storage Enabler for Containers without any special SSL configuration.
- *verify-full*, expecting the user to provide relevant certificates. When enabled, this SSL mode requires additional configuration steps as listed below.

Procedure

1. When operating in the *verify-full* mode, you will need to generate the following three pairs of the public-private keys for:
   - Spectrum Connect server. You can upload these certificates to the server, as explained in the IBM Spectrum Connect user guide.
   - IBM Storage Enabler for Containers (*ubiquity*) service object.
   - IBM Storage Enabler for Containers database (*ubiquity-db*) service object.
2. Verify that:
   - The SSL certificates that you have generated are valid and signed by root CA.
   - The SSL certificates have valid common and alternative names. The alternative names list must contain valid DNS names and/or IP addresses of the Spectrum Connect server, *ubiquity* service object, and *ubiquity-db* service object.
     Run these commands on Kubernetes cluster master node to obtain the required network parameters for the *ubiquity* and *ubiquity-db* services (see example with the *ns1* namespace below):

     ```
     kubectl create service clusterip ubiquity --tcp=9999:9999 -n ns1
     kubectl set selector svc ubiquity app.kubernetes.io/name=ubiquity -n ns1

     kubectl create service clusterip ubiquity-db --tcp=5432:5432 -n ns1
     kubectl set selector svc ubiquity-db app.kubernetes.io/name=ubiquity-db -n ns1
     ```

     These commands generate two Kubernetes services that provide the required DNS/IP address combinations.
   - The private certificate and certificate key files have the following names:
     - `ubiquity.crt` and `ubiquity.key` for the *ubiquity* service object.
     - `ubiquity-db.crt` and `ubiquity-db.key` for the *ubiquity-db* service object.

- The trusted CA files contain the root CA certificate and have the following names:
  - `scbe-trusted-ca.crt` for the Spectrum Connect server.
  - `ubiquity-trusted-ca.crt` for the *ubiquity* service object.
  - `ubiquity-db-trusted-ca.crt` for the *ubiquity-db* service object.
- Copy all generated `*.crt` and `*.key` files to a dedicated directory.

3. Create two secrets and one configmap, as illustrated for the *ns1* namespace below:

```
 kubectl create secret -n ns1 generic ubiquity-db-private-certificate --from-file=ubiquity-
db.key
   --from-file=ubiquity-db.crt
 kubectl create secret -n ns1 generic ubiquity-private-certificate --from-file=ubiquity.key
   --from-file=ubiquity.crt
 kubectl create configmap -n ns1 ubiquity-public-certificates --from-file=ubiquity-db-trusted-
ca.crt
   --from-file=scbe-trusted-ca.crt --from-file=ubiquity-trusted-ca.crt
```

   - configmap `ubiquity-public-certificates` for all the trusted CA files.
   - The `ubiquity-private-certificate` secret for the private certificates used by the *ubiquity* service object.
   - The `ubiquity-db-private-certificate` secret for the private certificates used by the *ubiquity-db* service object.

4. Proceed with installation of the IBM Storage Enabler for Containers, as detailed in "Performing installation" on page 11.

Performing installation

You can install the IBM Storage Enabler for Containers software on a compatible version of Kubernetes. For more information, refer to the release notes of this software package.

Before you begin

Verify that you have completed the preliminary configuration steps, as detailed in "Compatibility and requirements" on page 5.

Important:

- During installation of the IBM Storage Enabler for Containers, the IBM Storage Kubernetes FlexVolume driver is automatically installed on all master and worker nodes in a Kubernetes cluster, using the `ubiquity-k8s-flex` DaemonSet.
- A single IBM Storage Enabler for Containers instance can be installed per one Kubernetes cluster.

Procedure

Follow these steps to install IBM Storage Enabler for Containers:

1. Add IBM Helm charts repository:

```
helm repo add ibm-stable https://raw.githubusercontent.com/IBM/charts/master/repo/stable
```

2. Download the Helm chart from the stable IBM repository. See Helm documentation for details.

```
helm fetch --untar ibm-stable/ibm-storage-enabler-for-containers
```

3. Configure parameters in the `values.yaml` file. The file is located in the `./ibm-storage-enabler-for-containers` folder. However, it is recommended to save the `values.yaml` file in a different location, rename it, and configure the new file, while preserving the original `values.yaml`.

Table 1. Configuration parameters in `values.yaml`

Parameter	Description
backend	Backend type for Provisioner and FlexVolume. Allowed values: *spectrumConnect* (default) or *spectrumScale*.
spectrumConnect. connectionInfo.fqdn	IP address or FQDN of the Spectrum Connect server.
spectrumConnect. connectionInfo.port	Communication port of the Spectrum Connect server. Default value is *8440*.
spectrumConnect. connectionInfo. existingSecret	Secret for Spectrum Connect interface. The value must be the same as configured in Spectrum Connect. Keys username and password are mandatory.
spectrumConnect. backendConfig. instanceName	A prefix for any new volume created on the storage system.
spectrumConnect. backendConfig. defaultStorageService	Default Spectrum Connect storage service to be used, if not specified by the storage class.
spectrumConnect. backendConfig. newVolumeDefaults. fsType	File system type of a new volume, if not specified by the user in the storage class. Allowed values: *ext4* (default) or *xfs*.
spectrumConnect. backendConfig. newVolumeDefaults.size	Default volume size (in GB), if not specified by the user when creating a new volume. Default value is *1*.
spectrumConnect. storageClass. storageService	Storage Class profile pointing to the Spectrum Connect storage service name.
spectrumConnect. storageClass. fsType	Storage class filesystem type. Allowed values: *ext4* (default) or *xfs*.
ubiquityDb.spectrumConnect. dbCredentials. existingSecret	Existing secret object if it is defined.
ubiquityDb.persistence. pvName	Name of the persistent volume to be used for the *ubiquity-db* database. For the Spectrum Virtualize and Spectrum Accelerate storage systems, use the default value (*ibm-ubiquity-db*). For the DS8000 storage system, use a shorter value, such as (*ibmdb*). This is necessary because the DS8000 volume name length cannot exceed 8 characters.
ubiquityDb.persistence. pvSize	Default size (in GB) of the persistent volume to be used for the *ubiquity-db* database. Default value is *20*.
ubiquityDb.persistence. useExistingPv	Enabling the usage of an existing PV as the *ubiquity-db* database PV. This parameter is in use only when upgrading Enabler for Containers from an older version, which has been installed via script. Allowed values: *True* or *False* (default).

Table 1. Configuration parameters in `values.yaml` (continued)

Parameter	Description
ubiquityDb.persistence.storageClass.storageClassName	Storage class name. **Note**: The storage class parameters are used for creating an initial storage class for the *ubiquity-db* PVC. You can use this storage class for other applications as well. It is recommended to set the storage class name to be the same as the Spectrum Connect storage service name.
ubiquityDb.persistence.storageClass.existingStorageClass	Enabling the usage of an existing storage class object if it exists.
ubiquityDb.persistence.storageClass.defaultClass	Setting StorageClass as the default storage class. Allowed values: *True* or *False* (default).
ubiquityK8sFlex.flexLogDir	Directory for storing the `ubiquity-k8s-flex.log` file. Set by default to `/var/log`.
globalConfig.logLevel	Log level. Allowed values: *debug*, *info* (default), *error*.
globalConfig.sslMode	SSL verification mode. Allowed values: *require* (No validation is required, the IBM Storage Enabler for Containers server generates self-signed certificates on the fly.) or *verify-full* (Certificates are provided by the user.). The *verify-full* mode requires additional configuration steps, as detailed in the "Managing SSL certificates" on page 10 section.
customPodSecurityPolicy.enabled	Custom pod security policy for ICP deployment. Allowed values: *True* or *False* (default). If set to *True*, the policy is applied to all pods in the chart. New policies cannot be defined. Configure a policy in advance or use existing ones. Then, attach one or more policies to a *role* or *clusterRole*, and provide the name for the *role* or *clusterRole*. Currently, only *clusterRole* is supported. It will be bound to all *serviceAccounts* under the current namespace.
customPodSecurityPolicy.clusterRole	In ICP deployment, the name of *clusterRole* that has the required policies attached. Default value is *ibm-anyuid-hostpath-clusterrole*.

4. If the `values.yaml` file was downloaded to a local machine, use the **`scp <path_to_values.yaml> user@master_ip:<path_to _master_node_store_values.yaml`** command to copy the updated `values.yaml` file to a master node.
5. Start the installation. Make sure to use the same namespace names that were used for secret generation during the preparation stage ("Compatibility and requirements" on page 5).
   - Kubernetes:
     - Run this command:

```
helm install --name <release_name> --namespace <namespace_name>
-f <path_to_values.yaml> ./ibm-storage-enabler-for-containers
```

When the installation is complete, the `notes.txt` file is displayed.

- ICP:

  a. In the ICP GUI, go to **Catalog**, then locate the `ibm-storage-enabler-for-containers` Helm chart.
  b. In the **Configuration** tab, set the values according to your environment requirements. See the `values.yml` table above.
  c. Click **Install** to start the installation. You can monitor the progress by going to **Workloads** > **Helm Releases**.

What to do next

1. Verify the post-installation status of the IBM Storage Enabler for Containers service and check that the status of all components is error-free:

```
$ helm status <release_name>
```

2. Perform the sanity test:

```
$ helm test <release_name>
```

   The following message must be displayed: `Sanity finished successfully (pvc1 and pod1 were successfully created and deleted).`

3. Implement the standard data protection methods and practices to safeguard the data in the Enabler for Containers database. This will ensure the proper operation of the IBM Storage Enabler for Containers.

Upgrading existing installation

If you are already using earlier releases of IBM Storage Enabler for Containers, you can upgrade to the newer version without having to uninstall the previous one.

Before you begin

Verify version of your current installation of IBM Storage Enabler for Containers. Only version 2.0.0 of the package can be upgraded to version 2.1.0. Unlike previous releases, version 2.1.0 of IBM Storage Enabler for Containers uses Helm chart for managing its installation process.

To preserve all existing configuration parameters, keep the `ubiquity-db` PVC, which contains the database.

1. Download version 2.1.0 of the IBM Storage Enabler for Containers installation package from IBM Fix Central (www.ibm.com/support/fixcentral).
2. Replace the `ubiquity_uninstall.sh` file from the version 2.0.0 installation package by the one from version 2.1.0, by running the following: **`cp ../installer-for-ibm-storage-enabler-for-containers-2.1.0/ubiquity_uninstall.sh ../installer-for-ibm-storage-enabler-for-containers-2.0.0/`**.
3. If you intend to continue using the *verify-full* SSL mode, write down DNS names and/or IP addresses of *ubiquity* and *ubiquity-db* service objects. These services must be recreated with the same settings after uninstallation.
4. Uninstall any previous installation of the IBM Storage Enabler for Containers, while keeping the `ubiquity-db` PVC. Use the following command: **`/ubiquity_uninstall.sh -k`** (The *-k* flag is used to keep the `ubiquity-db` PVC intact.).
5. Recreate two Kubernetes services with DNS/IP address combinations which were in use in the previous version.

```
kubectl create service clusterip ubiquity --clusterip='ubiquity_service_ip' --tcp=9999:9999 --
namespace ubiquity
kubectl set selector svc ubiquity app.kubernetes.io/name=ubiquity -n ubiquity
kubectl create service clusterip ubiquity-db --clusterip='ubiquity_db_service_ip' --
tcp=5432:5432 --namespace ubiquity
kubectl set selector svc ubiquity-db app.kubernetes.io/name=ubiquity-db -n ubiquity
```

6. Create two secrets: Enabler for Containers secret for its database and Enabler for Containers secret for the IBM Spectrum Connect (Verify that Spectrum Connect secret username and password are the same as Enabler for Containers interface username and password in Spectrum Connect UI.). See the "Compatibility and requirements" on page 5 for details.

   Note: When creating a secret for the Enabler for Containers database, the *dbname* must be set to *ubiquity*. Also, its username and password must be the same as in the previous version: *UBIQUITY_DB_USERNAME_VALUE* and *UBIQUITY_DB_PASSWORD_VALUE* in the `ubiquity_installer.conf` file.

Procedure

Perform the following procedure to upgrade IBM Storage Enabler for Containers:

1. Download the Helm chart from the stable IBM repository. See Helm documentation for details.

```
helm fetch --untar ibm-stable/ibm-storage-enabler-for-containers
```

2. Configure parameters in the `values.yaml` file. The file is located in the `./ibm-storage-enabler-for-containers` folder. However, it is recommended to save the `values.yaml` file in a different location, rename it, and configure the new file, while preserving the original `values.yaml`.

   Set the **ubiquityDb.useExistingPV** parameter to *True*. In addition, set **ubiquityDb.persistence.pvName** to be the same as *IBM_UBIQUITY_DB_PV_NAME_VALUE* in `ubiquity_installer.conf` of the installation package for Enabler for Containers (2.0.0).
3. Install the new version of Enabler for Containers:

```
helm install --name <release_name> --namespace <namespace_name> -f <path_to_values.yaml>
 ./ibm-storage-enabler-for-containers
```

4. You can change the current revision to a new one within the same release, and use updated parameters in the `values.yaml` file. Use one of the following methods:

   - Edit the `values.yaml`, and then run:

   ```
   helm upgrade -f values.yaml <release_name> ./ibm-storage-enabler-for-containers
   ```

   - Run:

   ```
   helm upgrade --set parameter_name=new_value <release_name> --reuse-values
    ./ibm-storage-enabler-for-containers
   ```

5. When upgrading Enabler for Containers via ICP, follow these instructions:

   a. Preserve the `ubiquity-db` PVC, which contains the database, as explained above.

   b. Create two secrets: Enabler for Containers secret for its database and Enabler for Containers secret for the backend (IBM Spectrum Connect or IBM Spectrum Scale), as detailed in "Compatibility and requirements" on page 5.

   c. Set the configuration parameters. Make sure to:

      - Set the **ubiquityDb.useExistingPV** parameter to *True*.
      - Set the **ubiquityDb.persistence.pvName** parameter to be the same as the existing `ubiquity-db` PVC.

Note: The predefined pod security policy name is *ibm-anyuid-hostpath-psp*, and it has been verified for this Helm chart. If your target namespace is bound to this pod security policy, you can proceed with the chart installation. If you choose another pod security policy, you must enable the default pod security policy, and use the predefined cluster role: *ibm-anyuid-hostpath-clusterrole*.

Rolling back to a previous revision

You can roll back to an earlier revision of the current IBM Storage Enabler for Containers release.

Procedure

Perform the following procedure to roll back to an earlier revision of the current IBM Storage Enabler for Containers release:

1. Display the current and all previous revisions of your release:

```
helm history <release_name>
```

For example,

```
$ helm history rel_1
REVISION  UPDATED                   STATUS       CHART                                      DESCRIPTION
1         Wed Oct 24 23:29:50 2018  SUPERSEDED   ibm-storage-enabler-for-containers-1.0.0   Install
complete
2         Fri Nov 2 03:57:32 2018   SUPERSEDED   ibm-storage-enabler-for-containers-1.0.0   Upgrade
complete
3         Fri Nov 2 04:01:24 2018   SUPERSEDED   ibm-storage-enabler-for-containers-1.0.0   Rollback to 1
4         Fri Nov 2 04:03:42 2018   DEPLOYED     ibm-storage-enabler-for-containers-1.0.0   Rollback to 1
```

2. Run:

```
helm rollback <release_name> <release_revision>
```

For example, to roll back release *rel_1* to revision 1:

```
helm rollback rel_1 1
```

Note: You can also roll back to a previous release using ICP GUI.

a. In the ICP GUI, go to **Workloads** > **Helm Releases**.

b. Select **Action** > **Rollback**, choose the release that you want to revert to, and then click **Rollback**.

Uninstalling the software package

If you want to completely remove the IBM Storage Enabler for Containers software, use the following procedure.

Before you begin

Verify that there are no persistent volumes (PVs) that have been created, using IBM Storage Enabler for Containers.

Important: The uninstallation process removes the IBM Storage Enabler for Containers components associated with the Helm chart, metadata, user credentials, and other elements.

Procedure

Run this command to completely uninstall IBM Storage Enabler for Containers:
`$ helm delete <release_name> --purge`.

Note: When the Helm chart is deleted, the first elements to be removed are the Enabler for Container database deployment and its PVC. If the **`helm delete`** command fails after several attempts, delete these entities manually before continuing. Then, verify that the Enabler for Container database deployment and its PVC are deleted, and complete the uninstall procedure by running the **`$ helm delete <release_name> --purge --no-hooks`** command.

Managing integration with IBM Spectrum Connect

The IBM Storage Enabler for Containers is used for provisioning of storage volumes from an external IBM storage system to Kubernetes containers.

About this task

The following procedure details how to add the IBM Storage Enabler for Containers interface to IBM Spectrum Connect. A single IBM Storage Enabler for Containers interface can be operated per one Kubernetes cluster.

Procedure

1. On the **Interfaces** pane, click **Add Interface**, and then select **Enabler for Containers**.
   The **Add New Enabler for Conatainers Interface** dialog box is displayed.

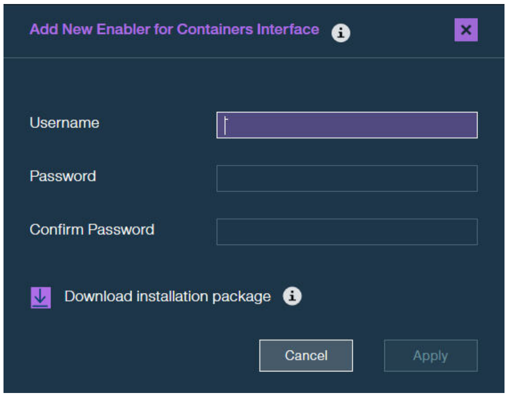

Add New Enabler for Containers Interface
Username
Password
Confirm Password
Download installation package
Cancel
Apply

Figure 4. Add New Enabler for Containers Interface dialog box

2. Enter credentials for the new IBM Storage Enabler for Containers user, and click **Apply**. The username and password must be the same as defined for the Spectrum Connect credentials secret, during installation of Enabler for Containers.

   Note: When entering a user name and password for an Enabler for Containers interface on the IBM Spectrum Connect with LDAP authentication, make sure that these credentials are the same as defined for LDAP. In addition, you can choose between using a single user or user group, if LDAP is enabled for this IBM Spectrum Connect instance.

   The IBM Storage Enabler for Containers interface is now added to IBM Spectrum Connect. The yellow frame and the exclamation mark indicate that the interface has no storage services delegated to it yet.

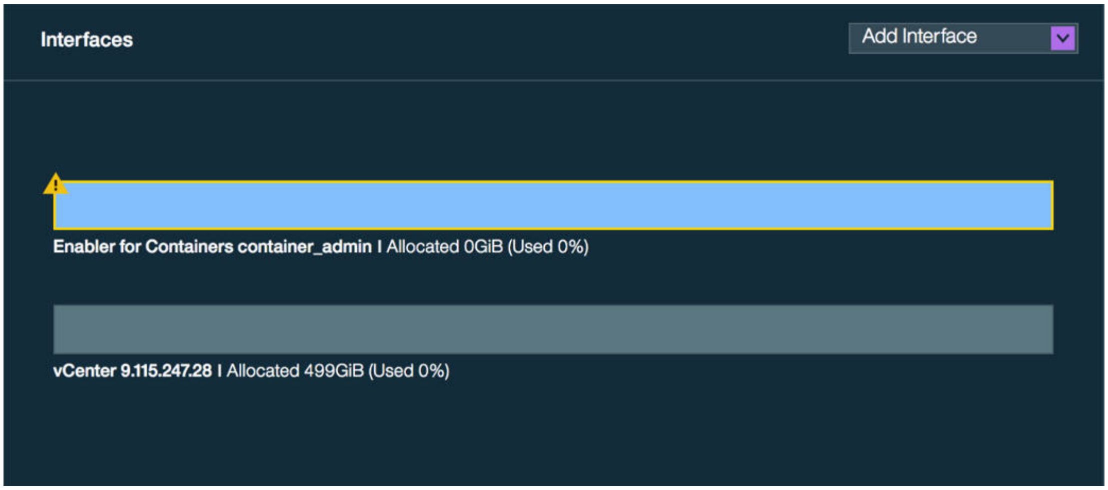

Interfaces
Add Interface
Enabler for Containers container_admin I Allocated 0GiB (Used 0%)
vCenter 9.115.247.28 I Allocated 499GiB (Used 0%)

Figure 5. IBM Storage Enabler for Containers interface on the Interfaces pane

What to do next

You can continue integrating the IBM Storage Enabler for Containers interface, as explained in the following sections:

- "Delegating storage services to the IBM Storage Enabler for Containers interface" on page 18.
- "Canceling service delegation to IBM Storage Enabler for Containers" on page 19.

Delegating storage services to the IBM Storage Enabler for Containers interface

Before you can use the IBM Storage Enabler for Containers to provision storage volumes from an external IBM storage system to Kubernetes containers, you must delegate the storage services that will be used by container plug-ins.

About this task

The services and their storage resources that you delegate on Spectrum Connect can be used in creating storage volumes in Kubernetes. Spectrum Connect storage services are translated into Kubernetes storage classes allowing for dynamic (on-demand) provisioning of storage for containers.

Service delegation is a prerequisite for deploying IBM Storage Enabler for Containers, IBM Storage Kubernetes Dynamic Provisioner and IBM Storage Kubernetes FlexVolume. For more information about deployment requirements, see "Compatibility and requirements" on page 5).

Procedure

To delegate storage services to IBM Storage Enabler for Containers:

1. On the **Interfaces** pane, click the Enabler for Containers interface to select it.
2. On the **Spaces/Storage Services** pane, select the storage space from which you want to choose storage services.
   The available services that reside on the selected storage space are immediately displayed.
3. Right-click on a service that you want to delegate to the Enabler for Containers interface, and then select **Delegate to <interface_name>**, or click the **Attach/Delegate** button on the service.
   The service and its frame colors change to indicate the successful delegation.

   You can continue the process by right-clicking available services under the current space.

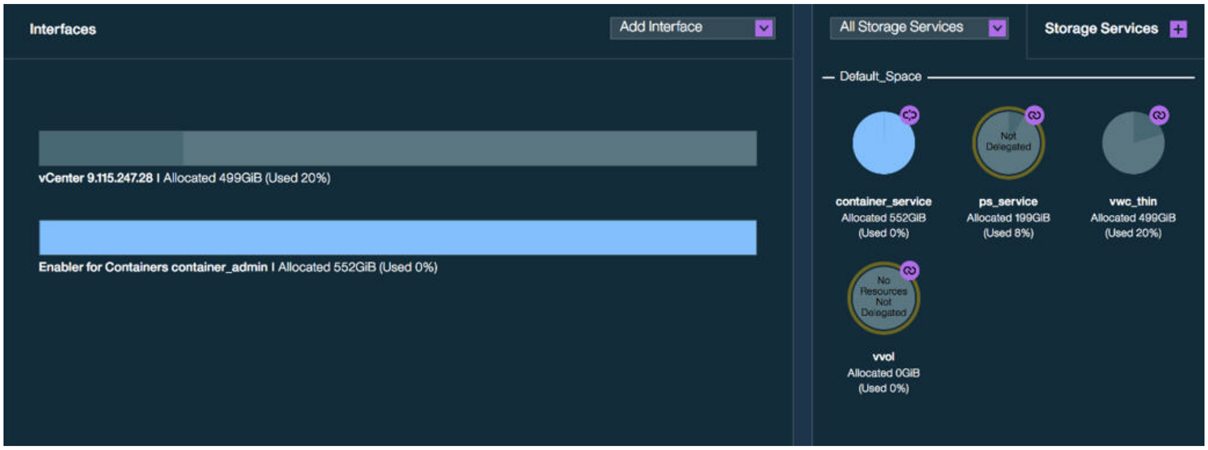


Figure 6. Enabler for Containers interface with a delegated service

The Enabler for Containers interface provides indication for the allocated and used storage space.

- Allocated – total amount of storage space available on all pools connected to the delegated services.
- Used – amount of storage space used by containers and snapshots on all pools connected to the delegated services.

What to do next

After service delegation, you can proceed with installation of the IBM Storage Enabler for Containers for further use of the allocated storage resources as persistent volumes for containers. See "Performing installation" on page 11.

If this is the first service defined before installation of the IBM Storage Enabler for Containers, a default storage class is created automatically during the installation. To link a Spectrum Connect storage service to a Kubernetes storage class, set the value of the `STORAGE_CLASS_PROFILE_VALUE` parameter in the `ubiqity.config` file to be is the same as the service name.

If you already installed the IBM Storage Enabler for Containers, add more services and delegate them to the Enabler for Containers interface. Then create Kubernetes storage classes and link them to the services. These storage classes can be used for creating new PVCs based on the Spectrum Connect services.

Canceling service delegation to IBM Storage Enabler for Containers

When required, you can cancel a storage service delegation to a IBM Storage Enabler for Containers interface.

Before you begin

Before canceling service delegation to IBM Storage Enabler for Containers, delete all Kubernetes storage classes linked to the services, which delegations are to be canceled.

About this task

Storage services, which delegation has been canceled, and their resources (pools) cannot be used as external storage for containers.

Procedure

1. On the **Interfaces** pane, click the IBM Storage Enabler for Containers interface.
   The services that are currently delegated to the interface are highlighted on the **Spaces/Storage Services** pane.
2. Right-click on a service which delegation you want cancel, and then select **Cancel delegation to <Interface_name>** , or click the **Detach/Cancel Delegation** button on the service.
   The service color changes to indicate the successful detachment.

You can continue the process by right-clicking delegated services under the current space.

Using IBM Storage Enabler for Containers with IBM block storage

This chapter covers the following topics:

- "Sample configuration for running a stateful container" on page 20.
- "Recovering a crashed Kubernetes node" on page 23.

Sample configuration for running a stateful container

You can use IBM Storage Enabler for Containers for running stateful containers with a storage volume provisioned from an external IBM block storage system.

About this task

This example illustrates a basic configuration required for running a stateful container with volume provisioned on a Spectrum Connect storage service.

- Creating a storage class `gold` that is linked to Spectrum Connect storage service `gold` with XFS file system.
- Creating a PersistentVolumeClaim (PVC) `pvc1` that uses the storage class `gold`.
- Creating a pod `pod1` with container `container1` that uses PVC `pvc1`.
- Starting I/Os into `/data/myDATA` in `pod1\container1`.
- Deleting the `pod1` and then creating a new `pod1` with the same PVC. Verifying that the file `/data/myDATA` still exists.
- Deleting all storage elements (pod, PVC, persistent volume and storage class).

Procedure

1. Open a command-line terminal.
2. Create a storage class, as shown below. The storage class `gold` is linked to a Spectrum Connect storage service on a pool from IBM FlashSystem A9000R with QoS capability and XFS file system. As a result, any volume with this storage class will be provisioned on the `gold` service and initialized with XFS file system.

```
$> cat storage_class_gold.yml
kind: StorageClass
apiVersion: storage.k8s.io/v1beta1
metadata:
  name: "gold"                 # Storage Class name
  annotations:
   storageclass.beta.kubernetes.io/is-default-class: "true"
provisioner: "ubiquity/flex"
parameters:
  profile: "gold"
  fstype: "xfs"
  backend: "scbe"

$> kubectl create -f storage_class_gold.yml
storageclass "gold" created
```

3. Display the newly created storage class to verify its successful creation.

```
$> kubectl get storageclass gold
NAME             TYPE
gold (default)   ubiquity/flex
```

4. Create a PVC `pvc1` with the size of 1 Gb that uses the storage class `gold`.

```
$> cat pvc1.yml
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: "pvc1"
```

```
spec:
  storageClassName: gold
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi

$> kubectl create -f pvc1.yml
persistentvolumeclaim "pvc1 created
```

The IBM Storage Enabler for Containers creates a persistent volume (PV) and binds it to the PVC. The PV name will be `PVC-ID`. The volume name on the storage will be `u_[ubiquity-instance]_[PVC-ID]`. Keep in mind that the `[ubiquity-instance]` value is set in the IBM Storage Enabler for Containers configuration file.

5. Display the existing PVC and persistent volume.

```
$> kubectl get pvc
NAME   STATUS    VOLUME                                     CAPACITY   ACCESSMODES   AGE
pvc1   Bound     pvc-254e4b5e-805d-11e7-a42b-005056a46c49   1Gi        RWO           1m

$> kubectl get pv
NAME                                       CAPACITY   ACCESSMODES   RECLAIMPOLICY
STATUS    CLAIM         REASON   AGE
pvc-254e4b5e-805d-11e7-a42b-005056a46c49   1Gi        RWO           Delete
Bound     default/pvc1
```

6. Display the additional persistent volume information, such as its WWN, location on a storage system, etc.

```
$> kubectl get -o json pv pvc-254e4b5e-805d-11e7-a42b-005056a46c49 |
grep -A15 flexVolume
        "flexVolume": {
            "driver": "ibm/ubiquity",
            "options": {
                "LogicalCapacity": "1000000000",
                "Name": "u_PROD_pvc-254e4b5e-805d-11e7-
a42b-005056a46c49",
                "PhysicalCapacity": "1023410176",
                "PoolName": "gold-pool",
                "Profile": "gold",
                "StorageName": "A9000 system1",
                "StorageType": "2810XIV",
                "UsedCapacity": "0",
                "Wwn": "36001738CFC9035EB0CCCCC5",
                "fstype": "xfs",
                "volumeName": "pvc-254e4b5e-805d-11e7-a42b-005056a46c49"
            }
        },
```

7. Create a pod pod1 with a persistent volume `vol1`.

```
$> cat pod1.yml
kind: Pod
apiVersion: v1
metadata:
  name: pod1
spec:
  containers:
  - name: container1
    image: alpine:latest
    command: [ "/bin/sh", "-c", "--" ]
    args: [ "while true; do sleep 30; done;" ]
    volumeMounts:
      - name: vol1
        mountPath: "/data"
  restartPolicy: "Never"
  volumes:
    - name: vol1
      persistentVolumeClaim:
        claimName: pvc1
```

```
$> kubectl create -f pod1.yml
pod "pod1" created
```

As a result, the IBM Storage Kubernetes FlexVolume performs the following:

- Attaches the volume to the host.

  Note: Volume attachment is triggered by the controller-manager which runs on the master node.

- Rescans and discover the multipath device of the new volume.
- Creates XFS or EXT4 file system on the device (if file system does not exist on the volume).
- Mounts the new multipath device on `/ubiquity/[WWN of the volume]`.
- Creates a symbolic link from `/var/lib/kubelet/pods/[pod ID]/volumes/ibm~ubiquity-k8s-flex/[PVC ID]` to `/ubiquity/[WWN of the volume]`.

8. Display the newly created pod1 and write data to its persistent volume. Make sure that the pod status is Running.

```
$> kubectl get pod pod1
NAME      READY     STATUS    RESTARTS   AGE
pod1      1/1       Running   0          16m


$> kubectl exec pod1 -c container1  -- bash -c "df -h /data"
Filesystem          Size  Used Avail Use% Mounted on
/dev/mapper/mpathi  951M   33M  919M   4% /data

$> kubectl exec pod1 -c container1  -- bash -c "mount | grep /data"
/dev/mapper/mpathi on /data type xfs (rw,relatime,seclabel,attr2,inode64,noquota)

$> kubectl exec pod1 touch /data/FILE
$> kubectl exec pod1 ls /data/FILE
File

$> kubectl describe pod pod1| grep "^Node:"
Node:        k8s-node1/hostname
```

9. Log in to the worker node that has the running pod and display the newly attached volume on the node.

```
> multipath -ll
mpathi (36001738cfc9035eb0ccccc5) dm-12 IBM     ,2810XIV
size=954M features='1 queue_if_no_path' hwhandler='0' wp=rw
`-+- policy='service-time 0' prio=1 status=active
  |- 3:0:0:1 sdb 8:16 active ready running
  `- 4:0:0:1 sdc 8:32 active ready running

$> df | egrep "ubiquity|^Filesystem"
Filesystem                       1K-blocks    Used Available Use% Mounted on
/dev/mapper/mpathi                  973148   32928    940220   4% /ubiquity/
6001738CFC9035EB0CCCCC5

$> mount |grep ubiquity
/dev/mapper/mpathi on /ubiquity/6001738CFC9035EB0CCCCC5 type xfs
(rw,relatime,seclabel,attr2,inode64,noquota)

$> ls -l /var/lib/kubelet/pods/*/volumes/ibm~ubiquity-k8s-flex/*
lrwxrwxrwx. 1 root root 42 Aug 13 22:41 pvc-254e4b5e-805d-11e7-a42b-005056a46c49 -> /
ubiquity/6001738CFC9035EB0CCCCC5
```

10. Delete the pod.

```
$> kubectl delete pod pod1
pod "pod1" deleted
```

As a result, the IBM Storage Kubernetes FlexVolume performs the following:

- Removes symbolic link from `/var/lib/kubelet/pods/[pod ID]/volumes/ibm~ubiquity-k8s-flex/[PVC ID]` to `/ubiquity/[WWN of the volume]`.

   - Unmounts the new multipath device on `/ubiquity/[WWN of the volume]`.
   - Removes the multipath device of the volume.
   - Detaches (unmap) the volume from the host.
   - Rescans in cleanup mode to remove the physical device files of the detached volume.

11. Remove the PVC and its PV (volume on the storage system).

    ```
    $> kubectl delete -f pvc1.yml
    persistentvolumeclaim "pvc1" deleted
    ```

12. Remove the storage class. This command removes the Kubernetes storage class only, the Spectrum Connect storage service remains intact.

    ```
    $> kubectl delete -f storage_class_gold.yml
    storageclass "gold" deleted
    ```

Recovering a crashed Kubernetes node

This section details a manual operation required to revive Kubernetes pods that reside on a crashed node due to an existing Kubernetes limitation.

Identifying a crashed node

When a worker node shuts down or crashes, all stateful pods that reside on it become unavailable, and the node status appears as *NotReady*.

```
# kubectl get
nodes


NAME          STATUS     AGE       VERSION
kuber-node1   Ready      2h        v1.7.5
kuber-node2   NotReady   2h        v1.7.5
kuber-serv1   Ready      2h        v1.7.5
```

When this node status persists for more than five minutes (default setting, see note below for instructions on how to change this value), the following occurs:

- Status of a pod scheduled on the pod becomes *Unknown*.
- The new pod is scheduled on another node in the cluster with status *ContainerCreating*, denoting that the pod is scheduled on a crashed node.

  As a result, the pod scheduled on a crashed node appears twice on two nodes with two statuses, as illustrated below.

  ```
  # kubectl get pods -o wide


  NAME                          READY  STATUS             RESTARTS  AGE  IP          NODE
  sanity-deployment-2414-538d2  1/1    Unknown            0         15m  IP_address  kuber-node2
  sanity-deployment-2414-n8cfv  0/1    ContainerCreating  0         34s  <none>      kuber-node1
  ```

Note: The time period between the node failure and creation of a new pod on another node is user-configurable. Use the following procedure to change the `pod-eviction-timeout` value:

1. Move the `kube-controller-manager.yml` file to `/tmp` folder (**`mv /etc/kubernetes/manifests/kube-controller-manager.yml /tmp`**).
2. Edit the controller-manager file (**`vim /tmp/kube-controller-manager.yml`**).
3. Add the `--pod-eviction-timeout=60s` line to the **`kube-controller-manager`** command.
4. Move the `kube-controller-manager.yml` file to its original location (**`mv /tmp/kube-controller-manager.yml /etc/kubernetes/manifests/kube-controller-manager.yml`**).

Recovering a crashed node

To allow Kubernetes to recover the stateful pods from a crashed node and schedule them on a functional node in the cluster:

- Remove the crashed node from the cluster to free up all its pods (**kubectl delete node <node_name>**),

  or

- Force delete the stateful pods, which are in *Unknown* state (**kubectl delete pods <pod_name> --grace-period=0 --force -n <namespace>**).

  After the mandatory five-minute timeout, as set by Kubernetes itself, the pod runs on a scheduled node. The pod status changes from *ContainerCreating* to *Running*. See example below for the sanity-deployment-2414-n8cfv pod.

If the crashed node recovers by itself or the user reboots the node, no additional actions are required to release its pods. The pods recover automatically after the node restores itself and joins the cluster. When a crashed node is recovered, the following occurs:

1. The pod with the *Unknown* status is deleted.
2. The volume(s) is detached from the crashed node.
3. The volume(s) is attached to node, on which the new pod is scheduled.
4. After the mandatory five-minute timeout, as set by Kubernetes itself, the pod runs on a scheduled node. The pod status changes from *ContainerCreating* to *Running*. See example below for the sanity-deployment-2414-n8cfv pod.

```
# kubectl get pods -o wide

NAME                               READY  STATUS    RESTARTS   AGE   IP           NODE
sanity-deployment-2414-n8cfv       1/1    Running   0          8m    IP_address   kuber-node1
```

Troubleshooting

This section can help you detect and solve problems that you might encounter when using the IBM Storage Enabler for Containers.

Checking logs

You can use the IBM Storage Enabler for Containers logs for problem identification. To collect and display logs, related to the different components of IBM Storage Enabler for Containers, use the following Kubernetes commands:

- Log collection – **./ubiquity_cli.sh -a collect_logs**. The logs are kept in the ./ubiquity_collect_logs_MM-DD-YYYY-h:m:s folder. The folder is placed in the directory, from which the log collection command was run.

  Note: The **ubiquity_cli.sh** and **ubiquity_lib.sh** script needed for log collection are available on GitHub or IBM Fix Central.

- IBM Storage Enabler for Containers – **$> kubectl logs -n ubiquity deploy/ubiquity**.
- IBM Storage Enabler for Containers database – **$> kubectl logs -n ubiquity deploy/ubiquity-db**.
- IBM Storage Kubernetes Dynamic Provisioner – **$> kubectl logs -n ubiquity deploy/ubiquity-k8s-provisioner**.
- IBM Storage Kubernetes FlexVolume for a pod – **$> kubectl logs -n ubiquity pod ubiquity-k8s-flex<pod_ID>**. In addition, events for all pods on a specific Kubernetes node are recorded in the ubiquity-k8s-flex.log file. You can view this file in the following default directory: /var/log.

YYou can change this directory by configuring **ubiquityK8sFlex.flexLogDir** parameter in the `values.yml` file.

- Controller-manager:
  - Static pod – **`kubectl get pods -n kube-system`** to display the master pod name. Then, **`kubectl logs -n kube-system pod_name`** to check the logs.
  - Non-static pod – **`journalctl`** to display the system journal. Then, search for the lines that have controller-manager entries.

Detecting errors

This is an overview of actions that you can take to pinpoint a potential cause for a stateful pod failure. The table at the end of the procedure describes the problems and provides possible corrective actions.

1. Run the **`ubiquitu_cli.sh -a status_wide`** command to check if:
   - All Kubernetes pods are in Running state.
   - All PVCs are in Bound state.
   - *ubiquity-k8s-flex* pod exists on each master node in the cluster. If you have three master nodes and five worker nodes, you must see a eight *ubiquity-k8s-flex* pods.
2. If you find no errors, but still unable to create or delete pods with PVCs, continue to the next step.
3. Display the malfunctioned stateful pod (**`$> kubectl describe pod pod_ ID`**). Usually, pod description contains information about possible cause of the failure. Then, proceed with reviewing the IBM Storage Enabler for Containers logs.
4. Display the IBM Storage Kubernetes FlexVolume log for the active master node (the node that the controller-manager is running on). Use the **`$> kubectl logs -n ubiquity ubiquity-k8s-flex-<pod_ID_running_on_master_node>`** command. As the controller-manager triggers the storage system volume mapping, the log displays details of the FlexVolume attach or detach operations.
   Additional information can be obtained from the controller-manager log as well.
5. Review the IBM Storage Kubernetes FlexVolume log for the worker node, on which the container pod is scheduled. Use the **`$> kubectl logs -n ubiquity ubiquity-k8s-flex-<pod_ID_running_on_worker_node>`** command. As the *kubelet* service on the worker node triggers the FlexVolume mount and unmount operations, the log is expected to display the complete volume mounting flow.
   Additional information can be obtained from the *kubelet* service as well, using the **`$> journalctl -u kubelet`** command.
6. Display the IBM Storage Enabler for Containers server log (**`$> kubectl logs -n ubiquity deploy/ubiquity`** command) or its database log (**`$> kubectl logs -n ubiquity deploy/ubiquity-db`** command) to check for possible failures.
7. Display the IBM Storage Dynamic Provisioner log (**`$> kubectl logs -n ubiquity ubiquity-k8s-provisioner`**) to identify any problem related to volume provisioning.

   Note: In general, you can use a request ID of a log entry to identify a particular event in the IBM Storage Enabler for Containers log. This will help you understand if the event is related to FlexVolume or Dynamic Provisioner.
   For example, if you have this event stored in the Provisioner log `INFO provision.go:141 volume::Provision [9a48c08c-6e3e-11e8-b510-a2547d4dae22-Provision] PVC with capacity 1073741824`, the string section 9a48c08c-6e3e-11e8-b510-a2547d4dae22 serves as a request ID. It identifies the event as related to the volume provisioning. Then, in the Enabler for Containers log, you can easily detect all entries with the same request ID, identifying them as relevant to volume provisioning.
   The same identification method can be applied to events, originating from the FlexVolume log.

8. View the Spectrum Connect log (`hsgsrv.log`) for list of additional events related to the storage system and volume operations.

Table 2. Troubleshooting for IBM Storage Enabler for Containers

Description	**Corrective action**
IBM Storage Kubernetes FlexVolume log for the active master node has no attach operations	Verify that: - Controller-manger pod can access the Kubernetes plug-in directory. See “Compatibility and requirements” on page 5 for instructions on configuring the access. - The correct hostname of the node is defined on the storage systems with the valid WWPN or IQN of the node, as described in “Compatibility and requirements” on page 5. This information appears in the controller-manager log.
IBM Storage Kubernetes FlexVolume log for the worker node that runs the pod has no new entries, except for *ubiquitytest* (Kubernetes 1.6 or 1.7 only)	Restart the *kubelet* on Kubernetes worker and master nodes. See “Performing installation” on page 11.
IBM Storage Kubernetes FlexVolume log for the worker node that runs the pod contains errors, related to WWN identification in the *multipath -ll* output	Check that: - Fibre Channel zoning configuration of the host is correct. - The Kubernetes node name is defined properly on the storage system. - Node rescan process was successful.
No connectivity between the FlexVolume pod and the IBM Storage Enabler for Containers server	Log into the node and run the FlexVolume in a test mode (**`$> /usr/libexec/kubernetes/kubelet-plugins/volume/exec/ibm~ubiquity-k8s-flex/ubiquity-k8s-flex testubiquity`**). If there is an error, make sure the IP of *ubiquity* service is the same as configured in the `ubiquity-configmap.yml` file. If not, configure the IP properly, then delete the FlexVolume DeamonSet and re-create it to apply the new address value.
Failure to mount a storage volume to a Kubernetes node	If the FlexVolume fails to locate a WWPN within multipath devices, verify your multipathing configuration and connectivity to a storage system. See “Compatibility and requirements” on page 5.
IBM Storage Enabler for Containers database fails to achieve the *Running* status after the configured timeout expires	- Check the *kubectl* logs for the FlexVolume pod on a node where the database was scheduled to run. Verify that the mount and rescan operations were successful. Another reason might be that the Docker image pulling is taking too much time, preventing the deployment to become active. - Check the *kubectl* logs for the FlexVolume pod that runs on the master node. Check any error related to attachment of the *ibm-ubiquity-db* volume. - Check the Kubernetes scheduling. Verify that it does not exceed the timeout configured in the installation script. - After you resolve the issue, verify that the *ibm-ubiquity-db* status is *Running*.

IBM Storage Enabler for Containers database persists in the *Creating* status. In addition, the `Volume has not been added to the list of VolumesInUse in the node's volume status` message is stored in `/var/log/message` file on the node, where the database is deployed.	To resolve this, move `kube-controller-manager.yaml` out and into `/etc/kubernetes/manifests/` to be recreated the control-manager pod:
	`mv /etc/kubernetes/manifests/kube-controller-manager.yaml  /tmp`
	`sleep 5`
	`mv /tmp/kube-controller-manager.yaml /etc/kubernetes/manifests/`
	`sleep 15`
	`#check the control-manager pod is running.`
	`$> kubectl get pod -n kube-system  | grep controller-manager`
	`# Verify it is in Running state.`

Persistent volume remains in the Delete state, failing to release	Review the Provisioner log (**`$> kubectl logs -n ubiquity deploy/ubiquity-k8s-provisioner`**) to identify the reason for deletion failure. Use the **`$ kubectl delete`** command to delete the volume. Then, contact the storage administrator to remove the persistent volume on the storage system itself.

Table 2. Troubleshooting for IBM Storage Enabler for Containers (continued)

Description	Corrective action

Communication link between IBM Storage Dynamic Provisioner and other solution elements fails due to Provisioner token expiration	IBM Storage Dynamic Provisioner uses a token that in some environments has an expiration time, for example twelve hours. To keep the link alive for an unlimited time, you can use a *service-account* token without expiration time. You can replace the current token with the *service-account* token, as follows:
	`$> TOKEN=$(kubectl get secret --namespace default $(kubectl get secret`
	`--namespace default | grep service-account | awk '{print $1}') -o yaml |`
	`grep token: | awk '{print $2}' | base64 -d)`
	`$> kubectl config set-credentials <mycluster.user> --token=${TOKEN}`

A pod creation fails and the following error is stored in the FlexVolume log of the node intended for the pod: `DEBUG 4908 executor.go:63 utils::Execute Command executed with args and error and output. [[{command=iscsiadm} {args=[-m session --rescan]} {error=iscsiadm: No session found.} {output=}]]"`	Verify that the node has iSCSI connectivity to the storage system. If the node has none, see the "Compatibility and requirements" on page 5 section for instructions on how to discover and log into iSCSI targets on the storage system.
Status of a stateful pod on a malfunctioned (crashed) node is *Unknown*	Manually recover the crashed node, as described in the "Recovering a crashed Kubernetes node" on page 23 section.
A pod becomes unresponsive, persisting in the *ContainerCreating* status. The `"error=command [mount] execution failure [exit status 32]"` error is stored in the FlexVolume log of the node, where the pod was scheduled. The failure occurs because the mountPoint already exists on this node. This might happen due to earlier invalid pod deletion.	Manually recover the pod, using the following procedure: 1. Check if there is a symbolic link to the mountPoint by running **`$> ls -l /var/lib/kubelet/pods/<pod_ID>/volumes/ibm~ubiquity-k8s-flex/<PVC_ID>`**. 2. If the file exists and there is a symbolic link to the `/ubiquity/<PVC_WWN>`, remove it by running **`rm /var/lib/kubelet/pods/<pod_ID>/volumes/ibm~ubiquity-k8s-flex/<PVC_ID>`**. 3. Unmount the PV by running **`umount /ubiquity/<PVC_WWN>`**. 4. Wait for several minutes for Kubernetes to rerun the mountFlow. Then, at the end of the process, display the FlexVolume log by running **`kubectl logs -n ubiquity ubiquity-k8s-flex-<pod_ID_on_the_node>`** to verify the *Running* status of the pod.
A pod becomes unresponsive, persisting in the *ContainerCreating* status. An error indicating a failure to discover a new volume WWN, while running the **`multipath -ll`** command, is stored in the FlexVolume log. This log belongs to the node, where the pod was scheduled.	Restart the *multipathd* service by running the **`service multipathd restart`** command on the worker node, where the pod was scheduled.

Chapter 3. IBM Spectrum Scale deployment

This section explains how to install IBM Storage Enabler for Containers and integrate it into Spectrum Scale.

- "Installation" on page 29
- "Managing IBM Spectrum Scale when used with IBM Storage Enabler for Containers" on page 42
- "Using IBM Storage Enabler for Containers with IBM Spectrum Scale" on page 43
- "Troubleshooting" on page 49

Installation

Download and install the IBM Storage Enabler for Containers with IBM Spectrum Scale in Kubernetes cluster as described in the following sections.

- "Performing pre-installation tasks" on page 29
- "Managing SSL certificates" on page 34
- "Performing installation" on page 36
- "Performing installation for a remote IBM Spectrum Scale cluster mount" on page 40

The steps to upgrade, uninstall, and roll back to a previous version of IBM Storage Enabler for Containers for IBM Spectrum Scale is same as those for IBM file storage.

- "Upgrading existing installation" on page 14
- "Rolling back to a previous revision" on page 16
- "Uninstalling the software package" on page 16.

Performing pre-installation tasks

The following conditions must be met before the installation:

Important: You can either choose to use IBM Block Storage or IBM Spectrum Scale, but not both the IBM Storage Enabler for Containers on the same Kubernetes or ICP cluster. Ensure that only one type of IBM storage back end is configured on the same Kubernetes or ICP cluster.

- Ensure that the IBM Spectrum Scale version 5.x.x or above is installed along with the IBM Spectrum Scale management API (GUI).
- Verify that there is a proper communication link between the IBM Spectrum Scale Management API Server (GUI) and the Kubernetes cluster.
- Ensure that all Kubernetes worker nodes have the IBM Spectrum Scale client installed on them.
- Verify that quota is enabled for all the file systems being used for creating persistent volumes.
- The file system used for the persistent volume must be mounted on all the worker nodes at all times.
- Ensure that all the worker nodes are running RedHat Enterprise Linux (RHEL) x86_64, ppc64le, s390x or SLES 12 SP3 s390x. For more information on supported RHEL versions, check the IBM Spectrum Scale support matrix at IBM Spectrum Scale FAQs.
- All worker nodes must be running the same platform (hardware and Linux distribution).
- Kubernetes controller-manager process must be run as **root**.
- Ensure that IBM Cloud Private (ICP) or Kubernetes is installed. For supported version, see the release notes of IBM Storage Enabler for Containers.
- Ensure that SELinux is in disabled mode.

- Ensure that the node `kubelet service` has the attach/detach capability enabled. The enable-controller-attach-detach is set to **True** by default. However, confirm that this option is set to **True** if you are debugging a functional problem.
- If the controller-manager is configured to run as a pod in your Kubernetes cluster, you must allow the events to be recorded in the controller-manager log file. To enable this, add the default path `/var/log` to the log file as a host path. You can change this directory by configuring the **FLEX-LOG-DIR** parameter in the `ubiquity-configmap.yml` file.
- Run the **`# mmlsmount all -L`** command to ensure that the GPFS file systems are mounted before starting Kubernetes on the nodes.

  The command gives an output similar to the following:

```
File system gpfs0 is mounted on 6 nodes:
  192.168.138.94     borg45
  192.168.138.62     borg48
  172.16.7.41        borg44
  192.168.138.59     borg50
  192.168.138.95     borg47
  192.168.138.92     borg43
```

- Run the **`# mmlsfs gpfs0 -Q`** command to ensure that the quota is enabled on the file systems.

  The command gives an output similar to the following:

```
flag                value                    description
------------------- ------------------------ -----------------------------------
 -Q                 user;group;fileset       Quotas accounting enabled
                    user;group;fileset       Quotas enforced
                    none                     Default quotas enabled
```

  If you fail to obtain this output, run the **`# mmchfs gpfs0 -Q yes`** command.

- Run the following command to ensure that the GUI server is running and can communicate with the Kubernetes nodes:

```
curl -u “admin:admin001”
-X GET https://9.11.213.85:443/scalemgmt/v2/cluster
--cacert < cert name>
```

  .

  The command gives an output similar to the following:

```
{
  "filesystems" : [ {
    "name" : "gpfs0"
  } ],
  "status" : {
    "code" : 200,
    "message" : "The request finished successfully."
  }
```

- Run the **`mmchconfig enforceFilesetQuotaOnRoot=yes`** command to set the **enforceFilesetQuotaOnRoot** value to `yes`. This ensures that quotas are enforced for the PVC created with root user ID.
- Run the # **`mmlsnodeclass`** command to ensure that the Kubernetes are not installed or configured on the nodes running the IBM Spectrum Scale GUI. This prevents port number conflicts and memory usage concerns.

  The command gives an output similar to the following. In this example, the name of the node running the IBM Spectrum Scale GUI is defined by the **Node Class Name** with value *GUI_MGMT_SERVERS*. In this example, the hostname is `borg43.<domain>`.

```
Node Class Name       Members
--------------------- -----------------------------------------------------------
GUI_MGMT_SERVERS      borg43.<domain>
GUI_SERVERS           borg45.<domain>,borg47.<domain>
                      borg48.<domain>,borg43.<domain>
```

- Ensure that IBM Spectrum Scale is tuned for the Kubernetes pod workload and the memory requirement of pods.

Note:

The Kubernetes and the running of workloads in pods consumes additional memory, beyond what similar workloads might consume running outside of Kubernetes.

The additional memory overheads of the Kubernetes environment must be considered when making choices for IBM Spectrum Scale configuration parameters that are related to memory usage. For example, the **`mmchconfig`** options pagepool and maxFilesToCache.

- Perform these steps for every master node in Kubernetes cluster:

  1. Enable the attach/detach capability for the *kubelet* service (**`controller-attach-detach-enabled=true`**). It is enabled by default.
  2. For Kubernetes version lower than 1.12, if the controller-manager is configured to run as a pod in your Kubernetes cluster, you must allow for event recording in controller-manager log file. To achieve this, add the default path to the log file (`/var/log`), as a host path. You can change this directory by configuring **ubiquityK8sFlex.flexLogDir** parameter in the `values.yml` file.
     - Stop the controller-manager pod by moving the `kube-controller-manager.yml` file to temporary directory: **`mv /etc/kubernetes/manifests/kube-controller-manager.yml /tmp`**.
     - Edit the `kube-controller-manager.yml` file: **`vi /tmp/kube-controller-manager.yml`**.
       - Add the following lines under the **volumes** tag.

         ```
         - hostPath:
             path: /var/log
             type: DirectoryOrCreate
           name: flexlog-dir
         ```

       - Add the following lines under the **volumeMounts** tag:

         ```
         - mountPath: /var/log
           name: flexlog-dir
         ```

       - Restart the controller-manager pod by moving the `kube-controller-manager.yml` file to its original location:
         `mv /tmp/kube-controller-manager.yml /etc/kubernetes/manifests/`.
       - Verify that the controller-manager pod is in the Running state: **`kubectl get pod -n kube-system | grep controller-manager`**.
  3. `flexvolume-dir` must be available within `kube-controller-manager`.
     - Verify that `flexvolume-dir` (`/usr/libexec/kubernetes/kubelet-plugins/volume/exec`) is mounted inside `kube-controller-manager`.

       Use the **`$ kubectl describe pod <kube-controller-manager-pod-id> -n kube-system`** command to show the details of the `kube-controller-manager`, which includes the `flexvolume-dir` mount.

       The output should look as follows:

```
flexvolume-dir:
Type: HostPath (bare host directory volume)
Path: /usr/libexec/kubernetes/kubelet-plugins/volume/exec
HostPathType: DirectoryOrCreate
```

If `flexvolume-dir` is not present, continue with the following steps.

- Stop the controller-manager pod by moving the `kube-controller-manager.yml` file to temporary directory: **`mv /etc/kubernetes/manifests/kube-controller-manager.yaml /tmp/kube-controller-manager.yaml`**.
- Edit the `/tmp/kube-controller-manager.yaml` file.
  - Add the following lines under the **volumeMounts** tag:

    ```
    mountPath: /usr/libexec/kubernetes/kubelet-plugins/volume/exec
    name: flexvolume-dir
    ```

  - Add the following lines under the **Volumes** tag:

    ```
    hostPath:
    path: /usr/libexec/kubernetes/kubelet-plugins/volume/exec
    type: DirectoryOrCreate
    name: flexvolume-dir
    ```

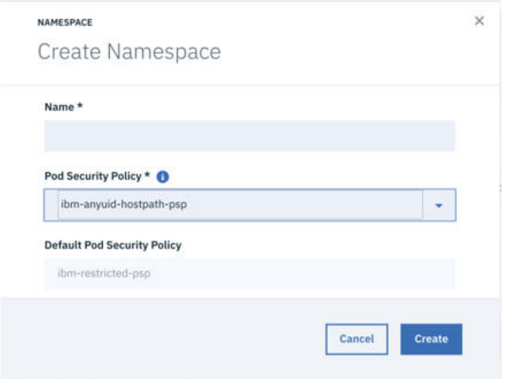
  - Restart the controller-manager pod by moving the `kube-controller-manager.yml` file to its original location:
    `mv /tmp/kube-controller-manager.yml /etc/kubernetes/manifests/`.
  - Verify that the controller-manager pod is in the Running state: **`kubectl get pod -n kube-system | grep controller-manager`**.

- Define a namespace to be used for creating secrets.
  - Kubernetes:

    ```
    kubectl create ns <namespaces_name>
    ```

  - ICP:
    1. In the ICP GUI, go to **Manage** > **Namespaces**.
    2. Click **Create Namespace**. In the **Create Namespace** dialog box, provide a namespace name and its pod security police.

       The predefined pod security policy name is *ibm-anyuid-hostpath-psp*, and it has been verified for this Helm chart. If your target namespace is bound to this pod security policy, you can proceed with the chart installation. If you choose another pod security policy, you must enable the default pod security policy, and use the predefined cluster role: *ibm-anyuid-hostpath-clusterrole*.



NAMESPACE

Create Namespace

Name *

Pod Security Policy *: ibm-anyuid-hostpath-psp

Default Pod Security Policy: ibm-restricted-psp

Figure 7. Create Namespace dialog box

- Create two secrets: Enabler for Containers secret for its database and Enabler for Containers secret for the IBM Spectrum Scale. Verify that IBM Spectrum Scale secret username and password are the same as Enabler for Containers interface username and password in IBM Spectrum Scale UI.

  – Kubernetes:

    1. Create secret for database using the following command:

       ```
       kubectl create secret generic <ubiquity_db_credentials_secret_name> --from-
       literal=dbname=<db_name>
        --from-literal=username=<username> --from-literal=password=<password>  -n <namespace>
       ```

       where

       ubiquity_db_credentials_secret_name
       Secret name of your choice. The same secret name should be used in **ubiquityDb.dbCredentials.existingSecret** parameter of `values.yaml`.

       db_name
       Specify the value as `ubiquity`.

       Username
       Username of your choice.

       Password
       Password of your choice.

       namespace
       Namespace under which the secret is to be created.

    2. Create secret for IBM Spectrum Scale Management API (GUI) using the following command:

       ```
       kubectl create secret generic <ubiquity_spectrumscale_credentials_secret_name> --from-
       literal=username=<username>
        --from-literal=password=<password>  -n <namespace>
       ```

       where

       ubiquity_spectrumscale_credentials_secret_name
       Secret name of your choice. The same secret name should be used in **ubiquity.spectrumScale.connectionInfo.existingSecret** parameter of `values.yaml`.

       Username
       IBM Spectrum Scale Management API (GUI) username. In case of remote mount, provide the remote IBM Spectrum Scale cluster management API(GUI) username.

       Password
       IBM Spectrum Scale Management API (GUI) password. In case of remote mount, provide the remote IBM Spectrum Scale cluster management API(GUI) password.

       namespace
       Namespace under which the secret is to be created.

  – ICP:

    1. In the ICP GUI, go to **Configuration** > **Secrets**.
    2. Click **Create Secret**. In the **Create Secret** dialog box, provide the following values for the Enabler for Containers database:

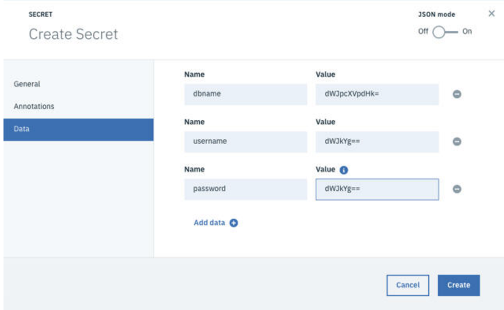
       - In the **General** tab, select **Namespace**, and enter the namespace name, added in the previous step.
       - In the **Data** tab, specify the following values:

         dbname
         The name of the first entry should be set to `dbname`, and the value should be set to `dWJpcXVpdHk=`.

Note: `dWJpcXVpdHk=` is the Base64-encrypted formatted representation for the value `ubiquity`.

username
The name of the second entry should be set to `username`, and the value should be a username of your choice in Base64-encrypted format.

password
The name of the third entry should be set to `password`, and the value should be a password of your choice in Base64-encrypted format.



Name: dbname Value: dWJpcXVpdHk=

Name: username Value: dWJkYg==

Name: password Value: dWJkYg==

Figure 8. Create Secret dialog box

3. Click **Create** to finish.
4. Repeat the secret creation procedure for the IBM Spectrum Scale management API (GUI) server secret.

   - In the **General** tab, select **Namespace**, and enter the namespace name, added in the previous step.
   - In the **Data** tab, specify the following values:

   username
   The name of the first entry should be set to `username`, and the value should be an IBM Spectrum Scale Management API (GUI) username in Base64-encrypted format. In case of remote mount, provide the remote IBM Spectrum Scale cluster management API(GUI) username in Base64-encrypted format.

   password
   The name of the second entry should be set to `password`, and the value should be an IBM Spectrum Scale Management API (GUI) password in Base64-encrypted format. In case of remote mount, provide the remote IBM Spectrum Scale cluster management API(GUI) password in Base64-encrypted format.

- If dedicated SSL certificates are required, see the relevant section of the "Managing SSL certificates" on page 34 procedure. When no validation is required and you can use the self-signed certificates, generated by default by the IBM Storage Enabler for Containers server, skip this procedure.

Note: IBM Cloud Private (ICP) and IBM Spectrum Scale GUI uses the port 443.

Managing SSL certificates

IBM Storage Enabler for Containers uses SSL certificates for maintaining a secure communication link between the IBM Storage Enabler for Containers server, its database, the Dynamic Provisioner, the FlexVolume, and the IBM Spectrum Scale Management API (GUI) server.

About this task

IBM Storage Enabler for Containers supports two SSL modes, when communicating with its components:

- *require*, when no validation is required. The IBM Storage Enabler for Containers server generates self-signed certificates on the fly. In this mode, you can skip the procedure detailed below and continue with the installation of the IBM Storage Enabler for Containers without any special SSL configuration.
- *verify-full*, expecting the user to provide relevant certificates. When enabled, this SSL mode requires additional configuration steps as listed below.

Procedure

1. When operating in the *verify-full* mode, you will need to generate the following three pairs of the public-private keys for:
   - IBM Spectrum Scale Management API (GUI) server. You can upload these certificates to the server, as explained in the IIBM Spectrum Scale Management API (GUI) user guide.
   - IBM Storage Enabler for Containers (*ubiquity*) service object.
   - IBM Storage Enabler for Containers database (*ubiquity-db*) service object.
2. Verify that:
   - The SSL certificates that you have generated are valid and signed by root CA.
   - The SSL certificates have valid common and alternative names. The alternative names list must contain valid DNS names and/or IP addresses of the IBM Spectrum Scale Management API (GUI) server, *ubiquity* service object, and *ubiquity-db* service object.
     Run these commands on Kubernetes cluster master node to obtain the required network parameters for the *ubiquity* and *ubiquity-db* services (see example with the *ubiquity* namespace below):

     ```
     kubectl create service clusterip ubiquity --tcp=9999:9999 -n ubiquity
     kubectl label service ubiquity -n ubiquity product=ibm-storage-enabler-for-containers

     kubectl create service clusterip ubiquity-db --tcp=5432:5432 -n ubiquity
     kubectl label service ubiquity-db -n ubiquity product=ibm-storage-enabler-for-containers
     ```

     These commands generate two Kubernetes services that provide the required DNS/IP address combinations.
   - The private certificate and certificate key files have the following names:
     - `ubiquity.crt` and `ubiquity.key` for the *ubiquity* service object.
     - `ubiquity-db.crt` and `ubiquity-db.key` for the *ubiquity-db* service object.
   - The trusted CA files contain the root CA certificate and have the following names:
     - `spectrumscale-trusted-ca.crt` for the IBM Spectrum Scale Management API server (GUI).
     - `ubiquity-trusted-ca.crt` for the *ubiquity* service object.
     - `ubiquity-db-trusted-ca.crt` for the *ubiquity-db* service object.
   - Copy all generated `*.crt` and `*.key` files to a dedicated directory.
3. Create two secrets and one configmap, as illustrated for the *ubiquity* namespace below:

   ```
    kubectl create secret -n ubiquity generic ubiquity-db-private-certificate --from-
   file=ubiquity-db.key
      --from-file=ubiquity-db.crt
    kubectl create secret -n ubiquity generic ubiquity-private-certificate --from-
   file=ubiquity.key
      --from-file=ubiquity.crt
    kubectl create configmap -n ubiquity ubiquity-public-certificates --from-file=ubiquity-db-
   trusted-ca.crt
      --from-file=spectrumscale-trusted-ca.crt --from-file=ubiquity-trusted-ca.crt
   ```

   - configmap `ubiquity-public-certificates` for all the trusted CA files.
   - The `ubiquity-private-certificate` secret for the private certificates used by the *ubiquity* service object.

- The `ubiquity-db-private-certificate` secret for the private certificates used by the *ubiquity-db* service object.

4. Proceed with installation of the IBM Storage Enabler for Containers, as detailed in "Performing installation" on page 11.

Performing installation

You can install the IBM® Storage Enabler for Containers software on a compatible version of Kubernetes. For more information, refer to the release notes of this software package.

Before you begin

Verify that you have completed the preliminary configuration steps for accessing a cluster described in the Performing pre-installation tasks section.

About this task

During installation of the IBM Storage Enabler for Containers, the IBM Storage Kubernetes FlexVolume driver is automatically installed on all master and worker nodes in a Kubernetes cluster, using the `ubiquity-k8s-flex` DaemonSet. A single IBM Storage Enabler for Containers instance can be installed per one Kubernetes cluster. Follow these steps to install IBM Storage Enabler for Containers:

Procedure

1. Add the IBM Helm charts repository, using the following command:

```
helm repo add ibm-stable https://raw.githubusercontent.com/IBM/charts/master/repo/stable
```

2. Use the following command to download the Helm chart from the stable IBM repository:

```
helm fetch --untar ibm-stable/ibm-storage-enabler-for-containers
```

   For more information on Helm charts, see Helm documentation.

3. Configure the parameters in the `values.yaml` file. The file is located in the `ibm-storage-enabler-for-containers` folder. However, it is recommended to save the `values.yaml` file in a different location, rename it, and configure the new file, while preserving the original `values.yaml` file.

Table 3. Configuration parameters in `values.yaml` file

Parameter	Description
ubiquity.spectrumScale.connectionInfo.fqdn	Used to specify the IP address or FQDN of remote IBM Spectrum Scale cluster management API(GUI) server. In case of remote mount refer section <remote mount setup> before proceeding with installation.
ubiquity.spectrumScale.connectionInfo.port	Used to specify the communication port of remote IBM Spectrum Scale cluster management API(GUI) server. Default value is 443.

Table 3. Configuration parameters in `values.yaml` file (continued)

Parameter	Description
ubiquity.spectrumScale.connectionInfo.existingSecret	Used to specify the secret for IBM Spectrum Scale Management API (GUI) server user credentials.
ubiquity.spectrumScale.backendConfig.defaultFilesystemName	Used to specify the remote IBM Spectrum Scale cluster filesystem to be used for creating persistent volumes, if not specified by the storage class. This filesystem is used to create persistent volume for `ubiquity-db`.
ubiquityDb.persistence.pvName	Used to specify the name of the persistent volume to be used for the `ubiquity-db` database.
ubiquityDb.persistence.pvSize	Used to specify the default size of the persistent volume to be used for the `ubiquity-db` database in GB. The default value is 20.
ubiquityDb.persistence.useExistingPv	Used to enable the usage of an existing PV as the `ubiquity-db` database PV. The allowed values are `True` and `False`. The parameter is set to`True` by default. Set this parameter to `True` if you want to use an existing PVC as Enabler for Containers database PVC. Use it only when you want to upgrade the IBM Storage Enabler for Containers from an old version installed by script to the latest version.
ubiquityDb.persistence.storageClass.storageClassName	Used to specify the storage class name. Used for creating an initial storage class for the `ubiquity-db` PVC. You can use this storage class for other applications as well.

Table 3. Configuration parameters in `values.yaml` file (continued)

Parameter	Description
ubiquityDb.persistence.storageClass.existingStorageClass	Used to enabling the usage of an existing storage class object if it exists.
ubiquityDb.persistence.storageClass.defaultClass	Used for setting `StorageClass` as the default storage class. The allowed values are `True` and `False`. The parameter is set to`False` by default.
ubiquityDb.dbCredentials.existingSecret	Used to specify the secret for `ubiquity-db` database.
ubiquityK8sFlex.flexLogDir	Used to specify the directory for storing the `ubiquity-k8s-flex.log` file. Set by default to `/var/log`.
ubiquityK8sFlex.ubiquityIPaddress	Used to specify the IP address of the `ubiquity` service object.
globalConfig.logLevel	Used to specify the log level. The allowed values are `debug`,`info` and `error`. The parameter is set to`error` by default.

Table 3. Configuration parameters in `values.yaml` file (continued)

Parameter	Description
globalConfig.sslMode	Used to specify the SSL verification mode. The allowed values are `debug`,`info` and `error`. The parameter is set to`error` by default. Allowed values: **require** No validation is required, the IBM Storage Enabler for containers server generates self-signed certificates on the fly. **verify-full** Certificates are provided by the user. The `verify-full` mode requires additional configuration steps, as detailed in the Managing SSL certificates section.
defaultPodSecurityPolicy. enabled	Default pod security policy for ICP deployment. Allowed values: *True* or *False* (default). If set to *True*, the policy is applied to all pods in the chart. New policies cannot be defined. Configure a policy in advance or use existing ones. Then, attach one or more policies to a *role* or *clusterRole*, and provide the name for the *role* or *clusterRole*. Currently, only *clusterRole* is supported. It will be bound to all *serviceAccounts* under the current namespace.
defaultPodSecurityPolicy. clusterRole	In ICP deployment, the name of *clusterRole* that has the required policies attached. Default value is *ibm-anyuid-hostpath-clusterrole*.

4. If the `values.yaml` file was downloaded to a local machine, use the following command to copy the updated `values.yaml` file to a master node:

   ```
   scp <path_to_values.yaml> user@master_ip:<path_to _master_node_store_values.yaml
   ```

5. Start the installation. Make sure to use the same namespace names that were used for secret generation during the pre-installation stage.

   - Kubernetes:

     – Run the following command:

     ```
     helm install --name <release_name> --namespace <namespace_name>
     -f <path_to_values.yaml> ./ibm-storage-enabler-for-containers
     ```

     When the installation is complete, the `notes.txt` file is displayed.

   - ICP:

     a. In the ICP GUI, go to **Catalog**, then locate the `ibm-storage-enabler-for-containers` Helm chart.

     b. In the **Configuration** tab, set the values according to your environment requirements. See the `values.yml` table above.

     c. Click **Install** to start the installation. You can monitor the progress by going to **Workloads** > **Helm Releases**.

What to do next

Once the installation is complete, ensure that the following steps are taken:

1. Verify the post-installation status of the IBM Storage Enabler for Containers service. Run the following command:

   ```
   $ helm status <release_name>
   ```

   Check that the status of all components is error-free.

2. Perform the sanity test by running the following command:

   ```
   $ helm test <release_name>
   ```

   The following message is displayed:

   ```
   Sanity finished successfully (pvc1 and pod1 were successfully created and deleted)
   ```

3. Implement the standard data protection methods and practices to safeguard the data in the Enabler for Containers database. This will ensure the proper operation of the IBM Storage Enabler for Containers.

Note: If there is a need to upgrade, roll back to a previous version or uninstall the Enabler for Containers package completely, see instructions in the following sections for the block-based storage systems.

- "Upgrading existing installation" on page 14
- "Rolling back to a previous revision" on page 16
- "Uninstalling the software package" on page 16.

Performing installation for a remote IBM Spectrum Scale cluster mount

IBM Spectrum Scale allows users shared access to files in either the cluster where the file system was created, or other IBM Spectrum Scale clusters. The ability to access and mount IBM Spectrum Scale filesystems owned by other clusters in a network can be accomplished using the **`mmauth`**,

mmremotecluster and **mmremotefs** commands. Each site in the network is managed as a separate cluster, while allowing shared file system access.

The cluster owning the filesystem is responsible for administering the filesystem, and granting access to other clusters on a per-cluster basis. After access to a filesystem has been granted to nodes in another IBM Spectrum Scale cluster, the nodes can mount the file system and perform data operations as if the filesystem were locally owned.

IBM Spectrum Scale Remote Cluster Mount functionality

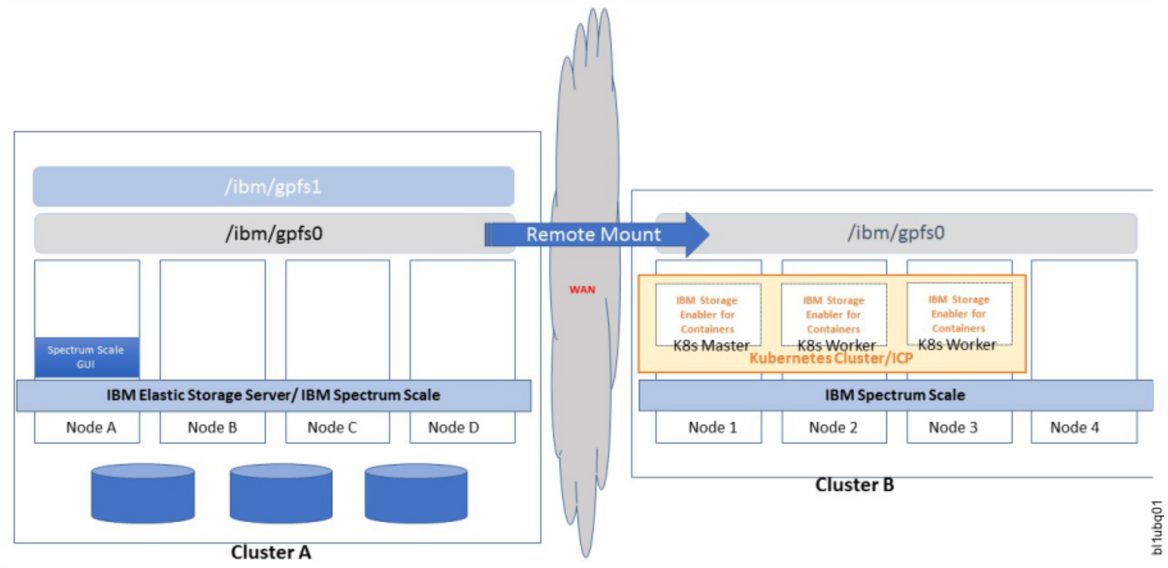


Figure 9. Sample configuration supported by IBM Storage Enabler for Containers

The diagram depict a sample configuration supported by IBM Storage Enabler for Containers. Cluster A is the ESS or IBM Spectrum Scale cluster having the local filesystems `gpfs0` and `gpfs1`. Cluster B is the IBM Spectrum Scale cluster that is configured to use IBM Spectrum Scale remote cluster mount functionality. `gpfs0` filesystem from Cluster A is available on Cluster B via remote cluster mount functionality. The mount point of the remote filesystem to be used for IBM Storage Enabler for Container must be same on both clusters. For example, in this sample setup it is `/ibm/gpfs0` on both clusters.

The file system management is done by Cluster A using the IBM Spectrum Scale GUI running on Cluster A. IBM Spectrum Scale GUI must not be running on the kubernetes nodes of Cluster B .

Note: It is not possible to use the local filesystems for IBM Storage Enabler for Container in this setup.

Important: Ensure that your remote mount is set up. For more information on accessing and mounting filesystems, see Accessing & mounting a remote GPFS file system. To install the IBM Storage Enabler for Containers for a remote mount, follow the steps given in Performing Installation. In step 3, ensure that for the **ubiquity.spectrumScale.connectionInfo** parameter you provide the following remote IBM Spectrum Scale cluster management API(GUI) server details and remote mounted file system information:

ubiquity.spectrumScale.connectionInfo.fqdn
IP/FQDN of IBM Spectrum Scale Management API server (GUI) on Cluster A.

ubiquity.spectrumScale.connectionInfo.port
Communication port of IBM Spectrum Scale Management API server (GUI) on Cluster A.

ubiquity.spectrumScale.connectionInfo.existingSecret
Set this parameter with an existing secret created for IBM Spectrum Scale Management API (GUI) server credentials of Cluster A.

ubiquity.spectrumScale.backendConfig.defaultFilesystemName
Filesystem name from Cluster A. For example, in this sample setup it is `gpfs0`.

Managing IBM Spectrum Scale when used with IBM Storage Enabler for Containers

When IBM Spectrum Scale is used for providing persistent volumes for containers, then the following must be considered:

Starting IBM Spectrum Scale in a system

The following startup sequence is required if PVCs are provisioned on IBM Spectrum Scale:

1. Start IBM Spectrum Scale, and ensure that autoload is enabled.

   Note: Use the **mmchconfig** command to enable autoload:

   ```
   mmchconfig autoload=yes
   ```

2. Mount the IBM Spectrum Scale file systems, and ensure that the automount is enabled for the file system.

   Note: Use the **mmchfs** command to enable automount:

   ```
   mmchfs <filesystem name> -A yes
   ```

3. Start Docker. Ensure that Docker starts after the file systems are mounted.
4. Start Kubernetes. Ensure that Kubernetes starts after the file systems are mounted.

Manually mounting IBM Spectrum Scale

Generally IBM Spectrum Scale automatically mounts the file systems to the host when **mmchconfig automount** is set to `yes`.

In some cases, after IBM Spectrum Scale is unmounted, it might be necessary to run the **mmmount** command to mount the IBM Spectrum Scale file systems again.

For more information on the **mmmount** command, see the *IBM Spectrum Scale: Command and Programming Reference* guide in the IBM Spectrum Scale Knowledge Center.

If you face an issue while mounting the IBM Spectrum Scale file systems, see "Resolving the mounting, unmounting or remounting issues in IBM Spectrum Scale file systems" on page 50.

Unmounting IBM Spectrum Scale

Follow these steps to unmount IBM Spectrum Scale file systems from a node:

1. Ensure that all the containers that are using the IBM Spectrum Scale file systems being unmounted are moved to other nodes.
2. Ensure that the new pods that are using the IBM Spectrum Scale file systems being unmounted are not scheduled on the node.
3. Unmount the IBM Spectrum Scale file system using the **mmunmount** command.

For more information on the **mmunmount** command, see the *IBM Spectrum Scale: Command and Programming Reference* guide in the IBM Spectrum Scale Knowledge Center.

If you face an issue in unmounting the IBM Spectrum Scale file systems, see "Resolving the mounting, unmounting or remounting issues in IBM Spectrum Scale file systems" on page 50.

Shutting down IBM Spectrum Scale

Follow these steps to shut down IBM Spectrum Scale when using IBM Storage Enabler for Containers:

1. Ensure that all the containers that are using the IBM Spectrum Scale file systems that are being unmounted are moved to other nodes.
2. Ensure that the new pods that are using the IBM Spectrum Scale file systems that are being unmounted are not scheduled on the node.
3. Stop Kubernetes and Docker.
4. Shutdown the IBM Spectrum Scale file system using the **`mmshutdown`** command.

   Note: Stop all PODs manually before running the **`mmshutdown`** command. Otherwise, a worker node might crash. If a crash occurs, its recovery involves recovery of the node, followed by manually stopping all PODs before resuming any prior shutdown.

   For more information on the **`mmshutdown`** command, see the *IBM Spectrum Scale: Command and Programming Reference* guide in the IBM Spectrum Scale Knowledge Center.

IBM Spectrum Scale monitoring considerations

Consider the following information for IBM Spectrum Scale when using IBM Storage Enabler for Containers:

If an IBM Spectrum Scale file system that is being used by Kubernetes or ICP get unmounted, or if there is an issue with the IBM Spectrum Scale file system mounted on a particular node, then the applications in the containers that are using the PVC from IBM Spectrum Scale throw an I/O error.

IBM Storage Enabler for Containers 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 that users directly monitor IBM Spectrum Scale for any IBM Spectrum Scale specific issues, since such monitoring is not done by Kubernetes, ICP or IBM Storage Enabler for Containers.

Using IBM Storage Enabler for Containers with IBM Spectrum Scale

This chapter covers the following topics:

Usage restrictions

Take note of the following restrictions before using IBM Storage Enabler for Containers with IBM Spectrum Scale.

- If a single PVC is used by multiple pods then it is the application's responsibility to maintain data consistency.
- It is recommended to create the PVCs one at a time, serially whenever possible. You can create a new PVC after all the earlier PVCs created using the IBM Storage Enabler for Containers are in bound state.
- 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 filesets on IBM Spectrum Scale.
- The **uid**, **gid**, **inode-limit**, and **fileset-type** parameters from the storage-classes are only allowed for new fileset creation.

- For each **uid-gid** combination a new storage class needs to be defined.
- Advanced IBM Spectrum Scale functionalities like AFM, Encryption, Compression, TCT etc. are not supported by IBM Storage Enabler for Containers.

  Note: There is no interface to these features through IBM Storage Enabler for Containers, and these feature are not tested with IBM Storage Enabler for Containers.

- IBM Storage Enabler for Containers does not check the storage space 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.
- Installing IBM Storage Enabler for Containers on Elastic Storage Server (ESS) I/O node and ESS EMS node are not supported.
- IBM Storage Enabler for Containers is supported on zLinux platforms.

  Note: SLES is also supported on zLinux. IBM Storage Enabler for Containers supporting SLES is only valid on a zLinux platform.

- The fileset created using IBM Storage Enabler for Containers must not be unlinked or deleted from any other interface.
- IBM Storage Enabler for Containers does not support volume expansion for storage class.
- The **df** command inside the container shows the full size of the IBM Spectrum Scale file systems.
- IBM Storage Enabler for Containers supports only up to 1000 PVCs with IBM Spectrum Scale.
- Stop all PODs manually before running the **mmshutdown** command. Otherwise, a worker node might crash. If a crash occurs, its recovery involves recovery of the node, followed by manually stopping all PODs before resuming any prior shutdown.

Configuring storage classes, PVCs and pods

As the only storage class created during installation is used for the database, you might need additional storage classes for volume provisioning on IBM Spectrum Scale.

A separate storage class must be created for each IBM Spectrum Scale file system to be used for creating persistent volumes. This section details how to configure Kubernetes storage classes, persistent volume claims, and pods. Follow these steps to configure the storage classes, PVCs and pods:

1. Define additional Kubernetes storage classes, if needed.

   Use the following template for creating additional storage classes for IBM Spectrum Scale filesystem:

```
# cat storage-class-spectrumscale-template.yml
# This is an IBM Storage Enabler for Containers Storage Class template.
kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
  name: "<NAME>"
  labels:
    product: ibm-storage-enabler-for-containers
#  annotations:
#   storageclass.beta.kubernetes.io/is-default-class: "true"
#reclaimPolicy: "Retain"                            # Optional, Values: Delete[default] or Retain
provisioner: "ubiquity/flex"
parameters:
  backend: "spectrum-scale"
  filesystem: "<filesystem name>"
  type: "fileset"
#  fileset-type: "<fileset type>"                   # Optional, Values: Independent[default] or dependent
#  uid: "<uid number>"                              # Optional
#  gid: "<gid number>"                              # Optional
#  inode-limit: "<no of inodes to be preallocated>" # Optional
#  isPreexisting: "<false|true>"                    # Optional, Values: false[default] or true
```

   You can set the following configuration parameters in the file:

Table 4. Configuration parameters in `storage-class-template.yml` for IBM Spectrum Scale

Parameter	**Description**
name	Storage class name.
filesystem	IBM Spectrum Scale file system name for creating new volumes.
fileset-type	Optional parameter. Type of fileset to be created for volume. Permitted values: independent[default], dependent.
uid	Optional parameter. Owner to be set on the fileset for newly created volume. User with specified uid/name must exist on IBM Spectrum Scale.
gid	Optional parameter. Group owner to be set on the fileset for newly created volume. Must be specified along with uid. Group with specified gid/group must exist on Spectrum Scale.
inode-limit	Optional parameter. Number of inode to be pre-allocated for newly created fileset
isPreexisting	Optional parameter. Used to indicate whether to use existing fileset or create new fileset for volume Permitted values: false[default], true. If true is specified, user must set **pv-name** parameter while creating PVC.
type	Permanently set to *fileset*.
product	Permanently set to *ibm-storage-enabler-for-containers*.
provisioner	Permanently set to *ubiquity/flex*.
backend	Permanently set to *spectrum-scale*.

Table 4. Configuration parameters in `storage-class-template.yml` for IBM Spectrum Scale (continued)

Parameter	Description
reclaimPolicy	Optional parameter. The reclaim policy tells the cluster what to do with the volume after it has been released of its claim. Persistent Volume inherit the reclaim policy from Storage Class. Permitted values : **Delete** If PVC is deleted then the PVC, PV and fileset that were created by IBM Storage Enabler for Containers are deleted. **Note:** This is the default value. **Retain** If a PVC is deleted, PV is not deleted and remains in `Released` state. The fileset is not deleted either. Deleting the PV which is in `Released` state does not delete the fileset associated with it. The user must manually delete the fileset from IBM Spectrum Scale.

Note: To use a PV within a non-root id container we can set the **uid** or **gid** fields in the storage class to the **uid** or **gid** which is used by the container process. This changes the ownership of the volume to the **uid** or **gid** specified in the storage class. By default, the owner of the volume is root.

2. Use the IBM Storage Enabler for Containers for creating persistent volume claims (PVCs) on IBM Spectrum Scale.

   Use the following template for creating persistent volume claim. When a PVC is created, the IBM Storage Dynamic Provisioner generates a persistent volume (PV), according to the IBM Spectrum Scale configuration defined for the PVC storage class, and then binds the PV to the PVC. By default, the PV name is `PVC-ID`. The fileset name on the storage is `[PVC-ID]`.

```
kind: PersistentVolumeClaim apiVersion: v1
metadata:
  name: "<PVC name>"
  labels:
    product: ibm-storage-enabler-for-containers
    #pv-name: "<PV name>"
spec:
  storageClassName: <Storage Class Name>
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: <Number>Gi
```

   You can set the following configuration parameters in the file:

Table 5. Configuration parameters in `pvc-template.yml` for IBM Spectrum Scale

Parameter	Description
name	Persistent volume claim name.

Table 5. Configuration parameters in `pvc-template.yml` for IBM Spectrum Scale (continued)

Parameter	Description
storageClassName	Storage class name used for the PVC provisioning.
pv-name	Persistent volume name. This name is used for creating a PV with a specific name, which is different from the default PV. The default PV name is its PVC ID. However, this dedicated PV name must be unique. No other PV with the same name is allowed within the Kubernetes cluster. Optional parameter. This must be set for using existing fileset for creating persistent volumes along with the **fileset** parameter from storage-class. **Note:** If a PVC with **pv-name**= `xyz` is created with **reclaimPolicy**= Retain, then another PVC cannot be created with same pv-name, that is with **pv-name**= `xyz`, even if the previous PVC with **pv-name**= `xyz` is deleted along with the PV.
accessModes	Permitted values : *ReadWriteOnce* and *ReadWriteMany*. Other access modes are not supported.
storage	Volume size in Gb. Other volume size units are not supported.

3. Create a pod to use the Kubernetes for storage.

   The PVCs can be used by Kubernetes pods for running stateful applications. The following example displays the template for using PVC in the pod yml file. When a pod is created, The IBM Storage FlexVolume performs the following actions:

   - Creates a link from `/var/lib/kubelet/pods/[pod ID]/volumes/ibm~ubiquity-k8s-flex/[PVCID]` to the *fileset link path* automatically. As a result, the pod goes up with the mounted PV on the container pod in the `mountPath` defined in the yml file.

```
kind: Pod
apiVersion: v1
metadata:
  name: <Pod name>
spec:
  containers:
  - name: <Container name>
    image: <Image name>
    volumeMounts:
      - name: <yaml volume name>
        mountPath: <Mount point>
  volumes:
    - name: <yaml volume name>
      persistentVolumeClaim:
        claimName: <PVC name>
```

   You can set the following configuration parameters in the file:

Table 6. Configuration parameters in `pod-template.yml` for IBM Spectrum Scale

Parameter	Description
name	Pod name
containers.name	Container name
containers.image	Container image
volumeMounts.name	Internal volume name
volumeMounts.mountPath	Mounting point for the PVC in the container
volumes.name	Internal volume name
volumes.persistentVolumeClaim	Name of the persistent volume claim

- When a pod is deleted, the link is removed from `/var/lib/kubelet/pods/[pod ID]/volumes/ibm~ubiquity-k8s-flex/[PVCID]` to the *fileset link path* automatically.

Updating the configuration parameters

If required, you can adjust configuration parameters of the IBM Storage Enabler for Containers after its installation.

Table 7. Configuration parameters in ubiquity-configmap

Parameter	Description
SPECTRUMSCALE-MANAGEMENT-IP	IP address or FQDN of IBM Spectrum Scale management API (GUI) server. To update this parameter after installation: 1. Update the parameter in the `ubiquity-configmap` configMap (**`$> kubectl edit -n ubiquity configmap ubiquity-configmap`**). 2. Delete the *ubiquity* pod of the *ubiquity* deployment. Then, it is automatically recreated by the deployment with the new parameters. **Note:** Ensure that PVC creation or deletion using Enabler for Containers is not in progress while deleting the Enabler for Containers pod.
SPECTRUMSCALE-DEFAULT-FILESYSTEM-NAME	Default IBM Spectrum Scale file system to be used for creating persistent volumes. **Note:** You cannot update this parameter.

Using an existing fileset for volume creation

You might want to use the existing fileset for creating persistent volume.

The following section describes the limitation for using existing fileset as a Kubernetes volume.

- The existing fileset to be used as a Kubernetes volume must exist in the file system specified in storage class and it must be linked.
- Quota must be enabled for the file system specified in the StorageClass.
- Creating PVC using a root fileset is not supported.
- StorageClass parameters **uid**, **gid**, **inode-limit**, **fileset-type** are not valid for this functionality and must not be specified in the StorageClass.

- Quota on the fileset must be equal or greater than storage requested in the PVC.
- If the **pv-name** is not specified in the PVC yaml then a random PV name is generated, and the IBM Storage Enabler for Containers tries to lookup the fileset with that random PV name. However, the random PV name most likely does not exist, and hence the PVC wont become available for use.
- If an existing fileset is used with the **reclaimPolicy** set to `retain`, then deleting the PVC does not delete the PV. The PV remains in a released state. If the released PV is deleted manually, and then if you tried to create a PVC with the same fileset name, the process fails.

Use the following steps to create volume using an existing fileset:

1. Create a new StorageClass, and set parameter **isPreexisting** to `true`.

```
kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
  name: "<NAME>"
  labels:
    product: ibm-storage-enabler-for-containers
#  annotations:
#   storageclass.beta.kubernetes.io/is-default-class: "true"
provisioner: "ubiquity/flex"
parameters:
  backend: "spectrum-scale"
  filesystem: "<filesystem name>"
  type: "fileset"
  isPreexisting: "true"
```

2. Create a PVC using the StorageClass created in step "1" on page 49, and set the parameter **pv-name** to `<name of existing fileset>`.

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: "<PVC name>"
  labels:
     product: ibm-storage-enabler-for-containers
     pv-name: "<name of existing fileset>"
spec:
  storageClassName: <StorageClass Name>
  accessModes:
    -ReadWriteOnce #ReadWriteOnce and ReadWriteMany
  resources:
    requests:
       storage: <Number>Gi
```

Changing the IBM Spectrum Scale RESTful credentials

To change the IBM Spectrum Scale RESTful credentials after installation, take the following steps:

1. Obtain the new credentials from IBM Spectrum Scale GUI.
2. Enter the new Base-64-encoded username and password (**`$>kubectl edit -n ubiquity secret spectrumscale-credentials`**).
3. Delete the `ubiquity` pod. Do not delete the `ubiquity-db` pod. After deleting Kubernetes restarts the pod with new the credentials.

Attention: Changing the default file system name is not supported.

Troubleshooting

The following issues have been encountered when using IBM Enabler for Containers with IBM Spectrum Scale:

Resolving the terminating state of pods

Sometimes if the volume path is not available, the pod might go into a `Terminating` state. The volume path could becomes unavailable due to one of the following reasons:

1. IBM Spectrum Scale is not running.
2. The file system is unmounted.
3. The fileset is unlinked.

When a PVC is created using IBM Storage Enabler for Containers, a soft link is created between `/var/lib/kubelet/pods/<pod id>/volumes/ibm~ubiquity-k8s-flex/<pv id>` and the *fileset link path* as part of the volume mount operation.

If the *fileset link path* is unavailable, then the soft link becomes broken. At this stage, if a pod gets deleted, then it might go into a `Terminating` state for infinite period of time. Follow these step to clean up the pod which is in a `Terminating` state:

Note: These steps are only applicable if the PVC used in a pod is created using IBM Storage Enabler for Containers.

1. Find the id of the pod which is in `Terminating` state using the following command:

   ```
   kubectl get pod --no-headers -o custom-columns=wwn:metadata.uid <pod name>
   ```

2. Find the node on which the pod was scheduled using the following command:

   ```
    kubectl get pod --no-headers -o custom-columns=wwn:spec.nodeName <pod name>
   ```

3. Check if a broken soft link exists on the node found in step 2 using the following command:

   ```
   ls /var/lib/kubelet/pods/<pod id>/volumes/ibm~ubiquity-k8s-flex/
   ```

4. If the soft link is broken, then delete the soft link using the following command:

   ```
   rm /var/lib/kubelet/pods/<pod id>/volumes/ibm~ubiquity-k8s-flex/<pv id>
   ```

   Note: Trying to mount the above IBM Spectrum Scale file system might fail. You might need to stop the Docker service to mount the IBM Spectrum Scale file system.

Resolving the mounting, unmounting or remounting issues in IBM Spectrum Scale file systems

The following issues have been observed if the IBM Spectrum Scale filesystem is not properly mounted, remounted, or unmounted.

- The **`mmunmount`** command fails to raise an error when I/O functions continue inside the container even after the user has unmounted the file system.

  Note: In this case, a container might still be accessing an IBM Spectrum Scale file system through a link, although the **`mmumount`** command has reported that the file system has been unmounted.

- In some cases, remounting the file system fails, and the system might display one of the following EBUSY errors:

  ```
  mmmount: Mounting file systems ...
  ```

  ```

  mount: gpfs is already mounted or /gpfs busy
  ```

  ```
  mmmount: Command failed. Examine previous error messages to determine cause.
  ```

These errors might occur after the unmount of an IBM Spectrum Scale file system has occurred while a pod is accessing the file system. Follow these steps to check for such issues and resolve them.

1. Find the ID of pod which is in terminating state using the following command:

   ```
   kubectl get pod --no-headers -o custom-columns=wwn:metadata.uid <pod name>
   ```

2. Find the node on which pod was scheduled using the following command:

   ```
   kubectl get pod --no-headers -o custom-columns=wwn:spec.nodeName <pod name>
   ```

3. Check if the broken soft link exists on the node. If the soft link is broken then delete the soft link.

   Note:

   – When a PVC is created as part of volume mount a soft link is created between `/var/lib/kubelet/pods/<pod id>/volumes/ibm~ubiquity-k8s-flex/<pv id>` and the *fileset link path*.

   – If the *fileset link path* is unavailable, then the soft link become broken. When a pod is deleted, the link should automatically be moved from `/var/lib/kubelet/pods/[pod ID]/volumes/ibm~ubiquity-k8s-flex/[PVCID]` to the *fileset link path*.

You can now remount the IBM Spectrum Scale file system using the **`mmmount`** command. For more information on the **`mmunmount command`**, see the *IBM Spectrum Scale: Command and Programming Reference* guide in the IBM Spectrum Scale Knowledge Center.

Note: You might need to restart Docker on the node with the terminated pod. Use the **`systemctl restart docker`** command to restart Docker.

Resolving the failure to create ClusterRoleBinding for ubiquity-k8s-provisioner

The plugin installation might fail with the following error:

```
Error from server (Forbidden): error when creating
"./yamls/ubiquity-k8s-provisioner-clusterroles.yml":
clusterroles.rbac.authorization.k8s.io "ubiquity-k8s-provisioner"
is forbidden: attempt to grant extra privileges:
```

To resolve this, create a `clusterrolebinding` for the admin user with the **cluster-role** value set to *cluster-admin*:

```
kubectl create clusterrolebinding myname-cluster-admin-binding
 --clusterrole=cluster-admin --user=admin
```

Notices

These legal notices pertain to the information in this IBM Storage product documentation.

This information was developed for products and services offered in the US. This material may 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
USA

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

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 states 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 Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

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

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119

Armonk, NY 10504-1785
USA

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

The performance data discussed herein is presented as derived under specific operating conditions. Actual results may vary.

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.

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

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of the International Business Machines Corp., 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 Copyright and trademark information website (www.ibm.com/legal/us/en/copytrade.shtml).

VMware, the VMware logo, ESX, ESXi, vSphere, vCenter, and vCloud are trademarks or registered trademarks of VMware Corporation in the United States, other countries, or both.

Microsoft, Windows Server, Windows, and the Windows logo are trademarks or registered trademarks of Microsoft Corporation in the United States, other countries, or both.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Other product and service names might be trademarks of IBM or other companies.

Printed in USA

SC27-9294-04