

IBM Storage Enabler for Containers
Version 2.0.0

User Guide



Note

Before using this document and the product it supports, read the information in “Notices” on page 55.

Edition notice

Publication number: SC27-9294-00. This publication applies to version 2.0.0 of IBM Storage Enabler for Containers and to all subsequent releases and modifications until otherwise indicated in a newer publication.

© **Copyright IBM Corporation 2018.**

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Figures	v
Tables	vii
About this guide.	ix
Who should use this guide	ix
Conventions used in this guide	ix
Related information and publications	ix
Getting information, help, and service	x
IBM Publications Center	x
Sending or posting your comments.	x
Chapter 1. Introduction	1
IBM block storage	2
IBM Spectrum Scale	3
Chapter 2. IBM block storage deployment	5
Installation	5
Compatibility and requirements	5
Managing SSL certificates	8
Downloading the software package.	9
Performing installation	10
Upgrading existing installation	14
Uninstalling the software package	16
Managing integration with IBM Spectrum Connect	17
Delegating storage services to the IBM Storage Enabler for Containers interface	18
Canceling service delegation to IBM Storage Enabler for Containers	19
Using IBM Storage Enabler for Containers with IBM block storage	20
Configuring storage classes, PVCs and pods	20
Sample configuration for running a stateful container	23
Recovering a crashed Kubernetes node	26
Updating configuration files.	27
Troubleshooting	30
Chapter 3. IBM Spectrum Scale deployment.	35
Installation	35
Performing pre-installation tasks	35
Managing SSL certificates.	37
Performing installation	38
Uninstalling the software package	41
Managing IBM Spectrum Scale when used with IBM Storage Enabler for Containers	42
Starting IBM Spectrum Scale in a system	42
Manually mounting IBM Spectrum Scale	42
Unmounting IBM Spectrum Scale	42
Shutting down IBM Spectrum Scale	43
IBM Spectrum Scale monitoring considerations	43
Using IBM Storage Enabler for Containers with IBM Spectrum Scale	43
Usage restrictions	43
Configuring storage classes, PVCs and pods	44
Updating the configuration files	48
Using an existing fileset for volume creation	51
Changing the IBM Spectrum Scale RESTful credentials	52
Troubleshooting	52
Resolving the terminating state of pods	52

Resolving the mounting, unmounting or remounting issues in IBM Spectrum Scale file systems 53

Notices 55

Trademarks 56

Figures

1.	Integration of IBM block storage systems and IBM Spectrum Scale in Kubernetes environment	2
2.	IBM Storage Enabler for Containers post-installation status	14
3.	Add New Enabler for Containers Interface dialog box	17
4.	IBM Storage Enabler for Containers interface on the Interfaces pane	18
5.	Enabler for Containers interface with a delegated service	19

Tables

1.	Configuration parameters in <code>ubiquity_installer.conf</code>	11
2.	Configuration parameters in <code>storage-class-template.yml</code>	20
3.	Configuration parameters in <code>pvc-template.yml</code>	21
4.	Configuration parameters in <code>sanity-pod.yml</code>	22
5.	Configuration parameters in <code>ubiquity-configmap.yml</code>	27
6.	Configuration parameters in <code>scbe-credentials-secret.yml</code>	30
7.	Configuration parameters in <code>ubiquity-db-credentials-secret.yml</code>	30
8.	Configuration parameters in <code>storage-class.yml</code> , <code>ubiquity-db-pvc.yml</code> , <code>sanity-pvc.yml</code>	30
9.	Troubleshooting for IBM Storage Enabler for Containers	32
10.	Configuration parameters in <code>ubiquity_installer_scale.conf</code>	39
11.	Configuration parameters in <code>storage-class-template.yml</code> for IBM Spectrum Scale	45
12.	Configuration parameters in <code>pvc-template.yml</code> for IBM Spectrum Scale	46
13.	Configuration parameters in <code>pod-template.yml</code> for IBM Spectrum Scale	48
14.	Configuration parameters in <code>ubiquity-configmap.yml</code>	48
15.	Configuration parameters in <code>spectrumscale-credentials-secret.yml</code> for IBM Spectrum Scale	50
16.	Configuration parameters in <code>ubiquity-db-credentials-secret.yml</code> for IBM Spectrum Scale	51

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/STHGJ)

- 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.

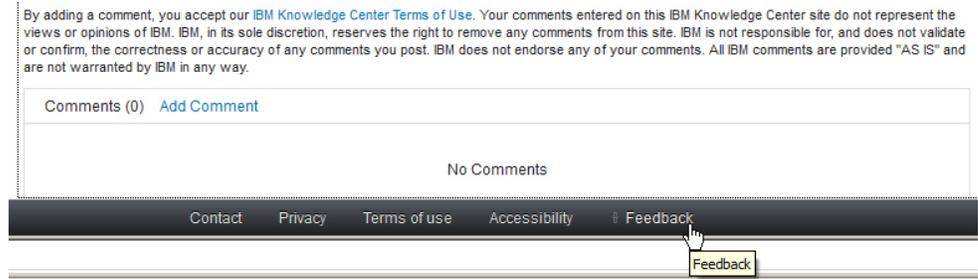
Sending or posting your comments

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.



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.

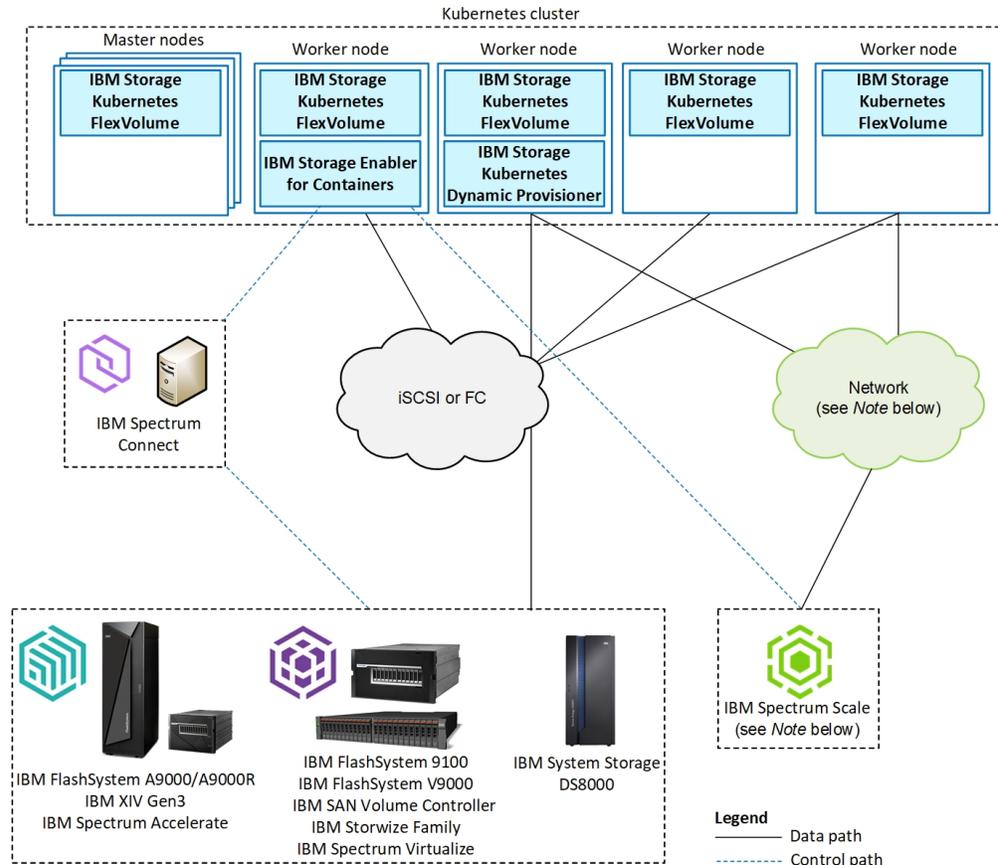


Figure 1. Integration of IBM block storage systems and IBM Spectrum Scale in Kubernetes environment

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 IBMStorage 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”
- “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 30

Installation

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

- “Compatibility and requirements”
- “Managing SSL certificates” on page 8
- “Downloading the software package” on page 9
- “Performing installation” on page 10
- “Upgrading existing installation” on page 14

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/SCKLT).

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).

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

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

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

- 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. 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. 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 **FLEX-LOG-DIR** parameter in the *ubiquity-configmap.yml* file, as detailed in “Updating configuration files” on page 27.

- 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.**
5. If dedicated SSL certificates are required, see the relevant section of the “Managing SSL certificates” 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.
 6. 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.yml* 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.

Before you begin

Download and extract the IBM Storage Enabler for Containers installer to gain access to the installation script (*ubiquity_installer.sh*). See steps 1 to 3 of the “Performing installation” on page 10 section.

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 this command to obtain the required network parameters for the *ubiquity* and *ubiquity-db* services:

```
$> ./ubiquity_installer.sh -s create-services
```

The script generates 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. Run the **`$> ubiquity_installer.sh -s create-secrets-for-certificates -t <certificate directory>`** command to create the following ConfigMap and secrets:
 - 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 10.

Downloading the software package

IBM Storage Enabler for Containers is available as a free software solution for IBM storage system customers.

About this task

You can download the latest version of the Installer for IBM Storage Enabler for Containers at any time from the IBM Fix Central. Fix Central provides fixes and updates for your systems software, hardware, and operating system.

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. Download the installer. See “Downloading the software package” on page 9).
2. Use the `kubect1` command to copy the installer to a local folder on a host that can access the Kubernetes cluster. Usually, a master node has access to the cluster.
3. Extract the installer file (* ' represents the build number):

```
# tar -xzf installer-for-ibm-storage-enabler-for-containers-x.y.z-*.tar.gz
```

4. Update the `ubiquity_installer.conf` configuration file, according to your environment requirements. Replace the *VALUE* placeholders in the files with your values.

Important:

- Any change to be made after running the installation script must be performed manually in the corresponding `yaml` files themselves.
 - If you install Enabler for Containers offline:
 - Download the required images from the Docker Hub. Verify that the operating system and architecture of your host matches the operating system and architecture of the images being pulled. If you intend to support multiple architectures offline, download all the architecture images and generate a Docker manifest locally.
 - Copy the images to your internal Docker registry.
 - If you still want to support multiple architecture offline, then you need to download all the architecture images and generate an docker manifest inside your local site.
 - Update the `ubiquity_installer.conf` configuration file, according to the new image location.
-

Table 1. Configuration parameters in `ubiquity_installer.conf`

Parameter	Description
UBIQUITY_IMAGE	Docker image of the IBM Storage Enabler for Containers to be deployed as Kubernetes deployment/ubiquity. Set by default to <code>ibmcom/ibm-storage-enabler-for-containers:x.y.z</code> . The image is available in the Docker Hub.
UBIQUITY_DB_IMAGE	Docker image of the IBM Storage Enabler for Containers database to be deployed as Kubernetes deployment/ubiquity-db. Set by default to <code>ibmcom/ibm-storage-enabler-for-containers-db:x.y.z</code> . The image is available in the Docker Hub.
UBIQUITY_K8S_PROVISIONER_IMAGE	Docker image of the IBM Storage Dynamic Provisioner to be deployed as Kubernetes deployment//ubiquity-k8s-provisioner. Set by default to <code>ibmcom/ibm-storage-dynamic-provisioner-for-kubernetes:x.y.z</code> . The image is available in the Docker Hub.
UBIQUITY_K8S_FLEX_IMAGE	Docker image of the IBM Storage FlexVolume to be deployed as Kubernetes daemonset/ubiquity-k8s-flex. Set by default to <code>ibmcom/ibm-storage-flex-volume-for-kubernetes:x.y.z</code> . The image is available in the Docker Hub.
SCBE_MANAGEMENT_IP_VALUE	IP address or FQDN of the Spectrum Connect server.
SCBE_MANAGEMENT_PORT_VALUE	Communication port of the Spectrum Connect server. Default value is <code>8440</code> .
SCBE_DEFAULT_SERVICE_VALUE	Default Spectrum Connect storage service to be used, if not specified by the storage class.
UBIQUITY_INSTANCE_NAME_VALUE	A prefix for any new volume created on the storage system. For example, <code>u_<instance_name>_<PVC_ID></code> .
DEFAULT_FSTYPE_VALUE	File system type of a new volume, if not specified by the user in the storage class. Allowed values: <code>ext4</code> or <code>xfs</code> . Default value is <code>ext4</code> .
DEFAULT_VOLUME_SIZE_VALUE	Default volume size (in GB), if not specified by the user when creating a new volume. Default value is <code>1</code> .

Table 1. Configuration parameters in *ubiquity_installer.conf* (continued)

Parameter	Description
LOG_LEVEL_VALUE	Log level. Allowed values: <i>debug</i> , <i>info</i> , <i>error</i> . Default value is <i>info</i> .
SSL_MODE_VALUE	SSL verification mode. Allowed values: <i>require</i> (No validation is required, the IBM Storage Enabler for Containers server generates self-signed certificates on the fly.) or <i>verify-full</i> (Certificates are provided by the user). The <i>verify-full</i> mode requires additional configuration steps, as detailed in the “Managing SSL certificates” on page 8 section.
SCBE_USERNAME_VALUE	Username defined for the IBM Storage Enabler for Containers interface in Spectrum Connect. Note: The IBM Storage Enabler for Containers interface appears as Spectrum Connect GUI.
SCBE_PASSWORD_VALUE	Password defined for the IBM Storage Enabler for Containers interface in Spectrum Connect. Note: The IBM Storage Enabler for Containers interface appears as Enabler for Containers in Spectrum Connect GUI.
UBIQUITY_DB_PV_NAME_VALUE	Name of the persistent volume to be used for the <i>ubiquity-db</i> database. For the Spectrum Virtualize and Spectrum Accelerate storage systems, use the default value (<i>ibm-ubiquity-db</i>). For the DS8000 storage system, use a shorter value, such as (<i>ibmdb</i>). This is necessary because the DS8000 volume name length cannot exceed 16 characters. As a result, the UBIQUITY_INSTANCE_NAME and UBIQUITY_DB_PV_NAME combination (u_<instance_name>_<PV_name>) cannot exceed 16 characters.
UBIQUITY_DB_USERNAME_VALUE	Username and password for the deployment of <i>ubiquity-db</i> database. Do not use the <i>postgres</i> username, because it already exists.
UBIQUITY_DB_PASSWORD_VALUE	Username and password for the deployment of <i>ubiquity-db</i> database.
STORAGE_CLASS_NAME_VALUE	Storage class name. Note: The storage class parameters are used for creating an initial storage class for the <i>ubiquity-db</i> 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.
STORAGE_CLASS_PROFILE_VALUE	Storage class profile, directing to the Spectrum Connect storage service name.
STORAGE_CLASS_FSTYPE_VALUE	File system type for the storage class profile. Allowed values: <i>ext4</i> or <i>xfs</i> . Default value is <i>ext4</i> .
FLEX_LOG_DIR_VALUE	Directory for storing the <i>ubiquity-k8s-flex.log</i> file. Set by default to <i>/var/log</i> .

5. Apply the `ubiquity_installer.conf` settings to the relevant yml files of the installer. Run the **`$> ./ubiquity_installer.sh -s update-ymls -c ubiquity_installer.conf`** command to do this. The code example below illustrates a successful configuration of the yml files.

```
[root@k8s-18-master-ibm installer-for-ibm-storage-enabler-for-containers-x.y.z-*]#
./ubiquity_installer.sh -s update-ymls -c /var/tmp/ubiquity_installer.conf
Executing STEP [update-ymls]...
Updating yml files with placeholders from /var/tmp/ubiquity_installer.conf file. Are you sure (y/n): y
Update placeholder [UBIQUITY_IMAGE                ] in files : ./ymls/ubiquity-deployment.yml
Update placeholder [UBIQUITY_DB_IMAGE             ] in files : ./ymls/ubiquity-db-deployment.yml
Update placeholder [UBIQUITY_K8S_PROVISIONER_IMAGE] in files : ./ymls/ubiquity-k8s-provisioner-deployment.yml
...
...
...
Update placeholder [STORAGE_CLASS_NAME_VALUE     ] in files : ./ymls/storage-class.yml
./ymls/ubiquity-db-pvc.yml ./ymls/sanity_ymls/sanity-pvc.yml
Update placeholder [STORAGE_CLASS_PROFILE_VALUE  ] in files : ./ymls/storage-class.yml
Update placeholder [STORAGE_CLASS_FSTYPE_VALUE   ] in files : ./ymls/storage-class.yml
Finished updating yml files according to /var/tmp/ubiquity_installer.conf
```

6. Start the installation by running the **`$> ./ubiquity_installer.sh -s install`** command. Make sure that the name of the target configuration file is `config`. The installation is complete, illustrated by the following message "IBM Storage Enabler for Containers" installation finished successfully in the Kubernetes cluster".

What to do next

1. Verify the post-installation status of the IBM Storage Enabler for Containers service. Run the following command: **`$> ./ubiquity_cli.sh -a status`**. Check that the database status is Bound, and all other elements are available and running, as illustrated in the following example.

```

NAME                                TYPE    DATA    AGE
secret/ubiquery-db-credentials      Opaque  3        6d4h

NAME                                DATA    AGE
configmap/ubiquery-configmap       18       6d4h

NAME                                CAPACITY ACCESS MODES RECLAIM POLICY STATUS CLAIM STORAGECLASS REASON AGE
persistentvolume/ibm-ubiquery-db    20Gi    RWO         Delete        Bound  ubiquery/ibm-ubiquery-db gold          6d4h

NAME                                STATUS VOLUME CAPACITY ACCESS MODES STORAGECLASS AGE
persistentvolumeclaim/ibm-ubiquery-db Bound  ibm-ubiquery-db 20Gi    RWO         gold          6d4h

NAME                                TYPE    CLUSTER-IP EXTERNAL-IP PORT(S) AGE
service/ubiquery                    ClusterIP 10.103.235.235 <none> 9999/TCP 6d4h
NAME                                TYPE    CLUSTER-IP EXTERNAL-IP PORT(S) AGE
service/ubiquery-db                  ClusterIP 10.96.119.189 <none> 5432/TCP 6d4h

NAME                                DESIRED CURRENT READY UP-TO-DATE AVAILABLE NODE SELECTOR AGE
daemonset.extensions/ubiquery-k8s-flex 3        3        3        3        3        <none>          6d4h

NAME                                DESIRED CURRENT UP-TO-DATE AVAILABLE AGE
deployment.extensions/ubiquery        1        1        1        1        6d4h
NAME                                DESIRED CURRENT UP-TO-DATE AVAILABLE AGE
deployment.extensions/ubiquery-db     1        1        1        1        6d4h
NAME                                DESIRED CURRENT UP-TO-DATE AVAILABLE AGE
deployment.extensions/ubiquery-k8s-provisioner 1        1        1        1        6d4h
NAME                                TYPE    DATA    AGE
scbe-credentials                      Opaque  2        6d4h

kubectrl get --namespace ubiquery pod | egrep "^ubiquery|^NAME"
-----
NAME                                READY STATUS RESTARTS AGE
ubiquery-7964d5ffdc-p7hpz           1/1   Running 3        6d4h
ubiquery-db-5555dd7c5d-966v8       1/1   Running 2        6d4h
ubiquery-k8s-flex-gzfg5             1/1   Running 4        6d4h
ubiquery-k8s-flex-v2sq9             1/1   Running 7        6d4h
ubiquery-k8s-flex-vks5z             1/1   Running 2        6d4h
ubiquery-k8s-provisioner-5b48d97577-k5tbq 1/1   Running 6        6d4h
[root@k8s-master-kas installer-for-ibm-storage-enabler-for-containers-1.2.0-118]#

```

Figure 2. IBM Storage Enabler for Containers post-installation status

Note: You can use the `$>./ubiquery_cli.sh -a status_wide` command to display the full system status. You can use it to verify that there is `ubiquery-k8s-flex` pod on each worker and master node.

2. Perform the sanity test to spin up and down the Kubernetes PVC and pod. Run this command: `$>./ubiquery_cli.sh -a sanity`. The following message must be displayed: Sanity finished successfully (pvcl and pod1 were successfully created and deleted).
3. In addition to the default storage class, define more Kubernetes storage classes, if needed. This default storage class, named as a value of the **STORAGE_CLASS_NAME_VALUE** parameter in the `ubiquery_installer.conf` file, was created during the installation. Template for setting storage classes is included in the `./ymls/templates/storage-class-template.yml` file. See “Configuring storage classes, PVCs and pods” on page 20 for details.
4. Use the IBM Storage Enabler for Containers for creating persistent volume claims (PVCs) on IBM storage systems. Template for PVC configuration is included in the `./ymls/templates/pvc-template.yml` file. See “Configuring storage classes, PVCs and pods” on page 20 for details.

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 1.2.0 of the package can be upgraded to version 2.0.0.

Procedure

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

1. Download the installer. See “Downloading the software package” on page 9.
2. Uncompress the installer *.tar file.
3. Create the service account files using the extracted yaml files.

Note: Enabler for Containers uses the `ubiquity-k8s-provisioner` service account for communication with the Kubernetes API server.

- **kubectl create -n ubiquity -f ubiquity-k8s-provisioner-serviceaccount.yml**
 - **kubectl create -n ubiquity -f ubiquity-k8s-provisioner-clusterrolebindings.yml**
 - **kubectl create -n ubiquity -f ubiquity-k8s-provisioner-clusterroles.yml**
4. Update and apply the `ubiquity-k8s-provisioner` deployment using the **kubectl edit deployment ubiquity-k8s-provisioner -n ubiquity** command.
 - Add the `serviceAccount: ubiquity-k8s-provisioner` line between `spec:` and `containers:.`
 - Change the `image: tag` of the Provisioner from version *<current version number>* to *<new version number>*.
 - Delete the `name: k8s-config` and `mountPath: /tmp/k8sconfig` lines in the **volumesMap** category.
 - Delete the `name: k8s-config`, `configMap:` and `name: k8s-config` lines in the **volumes** category.
 5. Save and exit the edit mode.
 6. Delete `k8s-config` using the **kubectl delete -n ubiquity configmap/k8s-config** command.
 7. Update the rest of the deployment images by changing the **image** tag from version *<current version number>* to *<new version number>*:
 - **kubectl edit deployment ubiquity-db -n ubiquity**
 - **kubectl edit deployment ubiquity -n ubiquity**
 - **kubectl edit daemonset ubiquity-k8s-flex -n ubiquity**
 8. Verify that the image update was successful by checking the status of the IBM Storage Enabler for Containers pods. Navigate to the directory where the IBM Storage Enabler for Containers was installed and run the following command **ubiquity_cli.sh -a status_wide** and check that all relevant pods are in the *Running* state.

Note: You can also use the **#> kubectl get -n ubiquity pod,daemonset,deployment -o wide** command to verify the update.

The image update is finished.

9. After the update, change the image version from <current version number> to <new version number> in all relevant yml files. Otherwise, you will not be able to start the images using **ubiquity_cli.sh -a start**. Use the following procedure:
 - a. Navigate to your installer directory and go the /yaml folder in it.
 - b. Locate the following yml files:
 - ubiquity-deployment.yml
 - ubiquity-k8s-provisioner-deployment.yml
 - ubiquity-k8s-flex-daemonset.yml
 - ubiquity-db-deployment.yml
 - c. Edit the files and manually change the image version from <current version number> to <new version number>. For example, image:
ibmcom/ibm-storage-enabler-for-containers-db:<current version number>
to image: ibmcom/ibm-storage-enabler-for-containers-db:<new version number>.

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, its metadata, user credentials, *ubiquity* namespace and other elements.

Procedure

To uninstall the IBM Storage Enabler for Containers software:

1. Log on to the host which contains the Enabler for Containers and navigate to the installation directory.
2. Run this script to completely uninstall IBM Storage Enabler for Containers:
\$> ./ubiquity_uninstall.sh. The installation is complete, illustrated by the following message
"IBM Storage Enabler for Containers" uninstall finished..
3. As this script does not uninstall the FlexVolume from all Kubernetes nodes, perform these additional steps for the complete FlexVolume removal:
 - a. If you use Kubernetes version 1.6 or 1.7 and have the /usr/libexec/kubernetes/kubelet-plugins/volume/exec directory on the controller-manager static pod, remove the Kubernetes FlexVolume from the controller-manager pod and restart the *kubelet* service.
 - b. Delete the /usr/libexec/kubernetes/kubelet-plugins/volume/exec/ibm~ubiquity-k8s-flex directory from all Kubernetes nodes.
 - c. Restart the *kubelet* service on all nodes.

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.

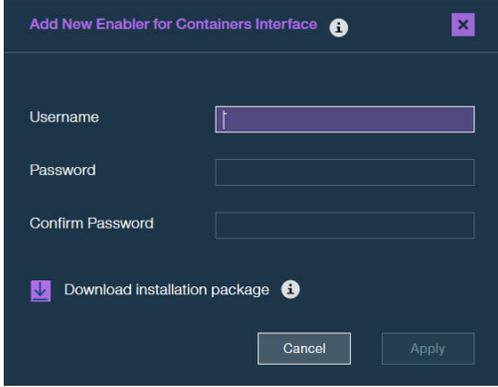


Figure 3. Add New Enabler for Containers Interface dialog box

2. Enter credentials for the new IBM Storage Enabler for Containers user, and click **Apply**.

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.

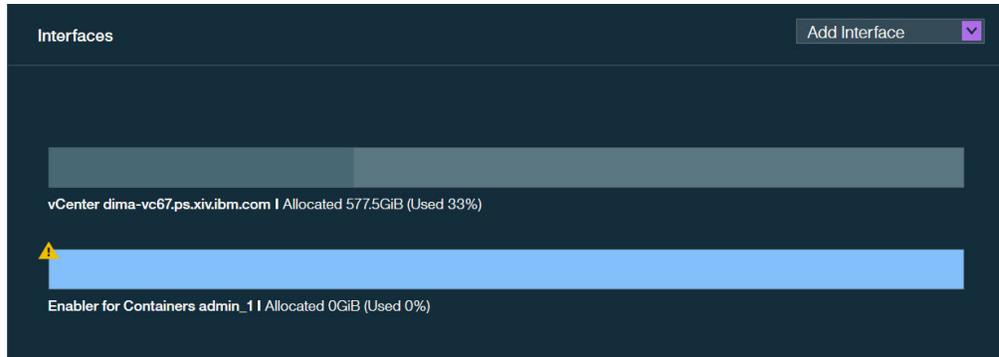


Figure 4. IBM Storage Enabler for Containers interface on the Interfaces pane

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.”
- “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 color changes to indicate the successful delegation.

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



Figure 5. 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 10.

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 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:

- “Configuring storage classes, PVCs and pods.”
- “Sample configuration for running a stateful container” on page 23.
- “Recovering a crashed Kubernetes node” on page 26.
- “Updating configuration files” on page 27.

Configuring storage classes, PVCs and pods

This section details how to configure Kubernetes storage classes, persistent volume claims and pods.

Defining additional Kubernetes storage classes

Define additional Kubernetes storage classes, if needed. Template for setting storage classes is illustrated below. The template is provided as the `./ymls/templates/storage-class-template.yml` file . As the only storage class, created during installation, is used for the database, you might need additional storage classes for volume provisioning on IBM storage. A separate storage class must be assigned for each storage service delegated to the IBM Storage Enabler for Containers interface in Spectrum Connect.

```
kind: StorageClass
apiVersion: storage.k8s.io/v1beta1
metadata:
  name: "<Storage class name>"
  labels:
    product: ibm-storage-enabler-for-containers
  annotations:
    # storageclass.beta.kubernetes.io/is-default-class: "boolean"
provisioner: "ubiquity/flex"
parameters:
  profile: "<SCBE service name>"
  fstype: "<Filesystem type>"
  backend: "sobe"
```

Table 2. Configuration parameters in storage-class-template.yml

Parameter	Description
name	Storage class name. It is recommended to use the storage class profile value as the storage class name.
profile	Spectrum Connect storage service name.

Table 2. Configuration parameters in *storage-class-template.yml* (continued)

Parameter	Description
fstype	File system type of a new volume Allowed values: <i>ext4</i> or <i>xfs</i> .
is-default-class	Configures the storage class to be the default <i>true</i> or not <i>false</i> . By default, it is set to <i>false</i> .
product	Permanently set to <i>ibm-storage-enabler-for-containers</i>
provisioner	Permanently set to <i>ubiquity/flex</i> .
backend	Permanently set to <i>scbe</i> .

Creating persistent volume claims (PVCs)

Use the IBM Storage Enabler for Containers for creating persistent volume claims (PVCs) on IBM storage systems. Template for PVC configuration is illustrated below. The template is provided as the *./ymls/templates/pvc-template.yml* file.

When a PVC is created, the IBM Storage Dynamic Provisioner generates a persistent volume (PV), according to Spectrum Connect storage service, defined for the PVC storage class, then it binds the PV to the PVC. By default, 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.

```
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
```

Table 3. Configuration parameters in *pvc-template.yml*

Parameter	Description
name	Persistent volume claim name
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.
accessModes	Permanently set to <i>ReadWriteOnce</i> . Other access modes, such as <i>ReadWriteMany</i> , are not supported.
storage	Volume size in Gb. Other volume size units are not supported.

Creating a pod to use the PVC for storage

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

- Volume attachment to the host, on which Kubernetes scheduled the pod to run.
- Rescanning and discovering the multipath device of the new volume.
- Creating an XFS or EXT4 file system on the device (if file system does not exist on the volume).
- Mounting the new multipath device on `/ubiquity/[WWN of the volume]`.
- Creating a symbolic link from `/var/lib/kubelet/pods/[pod ID]/volumes/ibm~ubiquity-k8s-flex/[PVC ID]` to `/ubiquity/[WWN of the volume]`.

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>
```

Table 4. Configuration parameters in `sanity-pod.yml`

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

When a pod is deleted, the following actions are performed automatically:

- Removing a symbolic link from `/var/lib/kubelet/pods/[pod ID]/volumes/ibm~ubiquity-k8s-flex/[PVC ID]` to `/ubiquity/[WWN of the volume]`.
- Unmounting the new multipath device on `/ubiquity/[WWN of the volume]`.
- Removing the multipath device of the volume.
- Detaching (unmapping) the volume from the host.
- Rescanning in the cleanup mode to remove the physical and multipath device files of the detached volume.

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 (36001738cfc9035eb0cccc5) 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/6001738CFC9035EB0CCCC5

$> mount |grep ubiquity
/dev/mapper/mpathi on /ubiquity/6001738CFC9035EB0CCCC5 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/6001738CFC9035EB0CCCC5
```

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 (**kubect1 delete node <node_name>**),
or
- Force delete the stateful pods, which are in *Unknown* state (**kubect1 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.

```
# kubect1 get pods -o wide
```

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

Updating configuration files

If required, you can adjust configuration parameters of the IBM Storage Enabler for Containers after its installation, using the *yml* files on the host:

- *./ubiquity-configmap.yml*
- *./scbe-credentials-secret.yml*
- *./ubiquity-db-credentials-secret.yml*
- *./ubiquity-db-credentials-secret.yml*
- *./ymls/storage-class.yml*
- *./ymls/ubiquity-db-pvc.yml*
- *./ymls/sanity_ymls/sanity-pvc.yml*

Table 5. Configuration parameters in *ubiquity-configmap.yml*

Parameter	Description
SCBE-MANAGEMENT-IP	<p>IP address or FQDN of the Spectrum Connect server.</p> <p>To update this parameter after installation:</p> <ol style="list-style-type: none"> 1. Update the parameter in the <i>ubiquity-configmap configMap</i> (\$> kubect1 edit -n ubiquity configmap ubiquity-configmap). 2. Delete the <i>ubiquity</i> pod of the <i>ubiquity</i> deployment. Then, it is automatically recreated by the deployment with the new parameters.

Table 5. Configuration parameters in *ubiquity-configmap.yml* (continued)

Parameter	Description
SCBE-MANAGEMENT-PORT	<p>Communication port of the Spectrum Connect server.</p> <p>To update this parameter after installation, use the same procedure as the one for SCBE-MANAGEMENT-IP.</p>
SCBE-DEFAULT-SERVICE	<p>Default Spectrum Connect storage service to be used, if not specified by the storage class.</p> <p>To update this parameter after installation, use the same procedure as the one for SCBE-MANAGEMENT-IP.</p>
UBIQUITY-INSTANCE-NAME	<p>A prefix for any new volume created on the storage system. This parameter cannot be changed.</p>
UBIQUITY-IP-ADDRESS	<p>IP address of the <i>ubiquity</i> service object. The <i>ubiquity_installer.sh</i> automatically updates this parameter during the initial installation.</p> <p>If the <i>ubiquity</i> service is recreated, it receives a new IP address. As a result, this parameter must be updated to allow the FlexVolume pods to access it.</p> <p>To update the IP address after installation:</p> <ol style="list-style-type: none"> 1. Update the IP address in the <i>ubiquity_configmap</i> configMap (<code>\$> kubectl edit -n ubiquity configmap ubiquity-configmap</code>). 2. Delete all FlexVolume pods (<code>\$> kubectl get pod -n ubiquity grep ubiquity-k8s-flex awk '{print \$1}' xargs kubectl delete pod -n ubiquity</code>).
DEFAULT-FSTYPE	<p>File system type of a new volume, if not specified by the user in the storage class.</p> <p>Allowed values: <i>ext4</i> or <i>xfs</i>.</p> <p>To update this parameter after installation, use the same procedure as the one for SCBE-MANAGEMENT-IP.</p>
DEFAULT-VOLUME-SIZE	<p>Default volume size (in GB), if not specified by the user when creating a new volume.</p> <p>To update this parameter after installation, use the same procedure as the one for SCBE-MANAGEMENT-IP.</p>
LOG-LEVEL	<p>Log level.</p> <p>Allowed values: <i>debug</i>, <i>info</i>, <i>error</i>.</p> <p>To receive more detailed events about the IBM Storage Enabler for Containers operation, you can adjust the log level after installation. Use this information your troubleshooting and debug processes.</p> <p>To change the log level after installation:</p> <ol style="list-style-type: none"> 1. Change the value in the <i>ubiquity_configmap</i> configMap (<code>\$> kubectl edit -n ubiquity configmap ubiquity-configmap</code>). 2. Stop the IBM Storage Enabler for Containers (<code>\$> ubiquity_cli.sh -a stop</code>). 3. Restart the IBM Storage Enabler for Containers (<code>\$> ubiquity_cli.sh -a start</code>).

Table 5. Configuration parameters in `ubiquity-configmap.yml` (continued)

Parameter	Description
SSL-MODE	<p>SSL verification mode.</p> <p>Allowed values: <i>require</i> (No validation is required, the IBM Storage Enabler for Containers server generates self-signed certificates on the fly.) or <i>verify-full</i> (Certificates are provided by the user.)</p> <p>To change the SSL mode to <i>verify-full</i> after installation:</p> <ol style="list-style-type: none"> 1. Stop the IBM Storage Enabler for Containers (<code>\$> ubiquity_cli.sh -a stop</code>). 2. Edit the <code>ubiquity-configmap.yml</code> file, setting the SSL_MODE to <i>verify-full</i>. 3. Edit the <code>ubiquity_installer.conf</code> file, setting the SSL_MODE to <i>verify-full</i>. 4. Run <code>\$> ubiquity_installer.sh -a update-ymls -c ubiquity_installer.conf</code>. 5. Create SSL certificates, as detailed in the “Managing SSL certificates” on page 8 section. 6. Restart the IBM Storage Enabler for Containers (<code>\$> ubiquity_cli.sh -a start</code>).
FLEX-LOG-DIR	<p>Directory for the <code>ubiquity-k8s-flex.log</code> file. The default location is the <code>/var/log</code> directory.</p> <p>To change the <code>ubiquity-k8s-flex.log</code> file directory after installation:</p> <ol style="list-style-type: none"> 1. Change the default folder in the <code>ubiquity_configmap configMap</code> (<code>\$> kubectl edit -n ubiquity FLEX-LOG-DIR <new_directory></code>). 2. Edit the <code>ubiquity-k8s-flex-daemonset.yml</code> file, setting the hostPath and mountPath to the new directory value: <ul style="list-style-type: none"> • <code>kubectl delete -f yamls/ubiquity-k8s-flex-daemonset.yml -n ubiquity</code> • <code>kubectl create -f yamls/ubiquity-k8s-flex-daemonset.yml -n ubiquity</code> 3. If you change the default log file location, make sure to set it to the same value in the relevant yml files on the controller-manager pod.
FLEX-LOG-ROTATE-MAXSIZE	<p>Maximum size of the <code>ubiquity-k8s-flex.log</code> file.</p> <p>To change the default log file size after installation:</p> <ul style="list-style-type: none"> • Change the default log file size in the <code>ubiquity_configmap configMap</code> (<code>\$> kubectl edit -n ubiquity FLEX-LOG-ROTATE-MAXSIZE <new_size></code>).

Table 6. Configuration parameters in *scbe-credentials-secret.yml*

Parameter	Description
SCBE-USERNAME	<p>Username defined for the IBM Storage Enabler for Containers interface in Spectrum Connect.</p> <p>To change the Enabler for Containers interface credentials after installation:</p> <ol style="list-style-type: none"> 1. Change the credentials for the Enabler for Containers interface, using the Spectrum Connect GUI. 2. Enter the new Base-64-encoded username and password (<code>\$> kubectl edit -n ubiquity scbe-credentials</code>). 3. Delete the <code>ubiquity-k8s-flex</code> pods, <code>ubiquity-k8s-provisioner</code> pod and <code>ubiquity</code> pod. Do not delete the <code>ubiquity-db</code> pod After the deletion, Kubernetes restarts the pods with new credentials.
SCBE-PASSWORD	<p>Password defined for the IBM Storage Enabler for Containers interface in Spectrum Connect. Instructions on how to change the interface credentials after installation are detailed in the SCBE-USERNAME description above.</p>

Table 7. Configuration parameters in *ubiquity-db-credentials-secret.yml*

Parameter	Description
UBIQUITY-DB-USERNAME	<p>Username for the deployment of <i>ubiquity-db</i> database.</p> <p>Do not use the <i>postgres</i> username, because it already exists.</p>
UBIQUITY-DB-PASSWORD	<p>Password for the deployment of <i>ubiquity-db</i> database.</p>
<p>Note: : If you change the <i>ubiquity-db</i> database password, make sure to set the new value for the PostgreSQL as well. Run the psql command from the Enabler for Containers container:</p> <pre>#>> psql " postgresql://ubiquity@ubiquity-db:5432/ubiquity?sslmode=require" ==> ALTER ROLE ubiquity password 'new password';</pre>	

Table 8. Configuration parameters in *storage-class.yml*, *ubiquity-db-pvc.yml*, *sanity-pvc.yml*

Parameter	Description
STORAGE-CLASS-NAME	Storage class name.
STORAGE-CLASS-PROFILE	Storage class profile, directing to the Spectrum Connect storage service.
STORAGE-CLASS-FSTYPE	<p>File system type for the storage class profile.</p> <p>Allowed values: <i>ext4</i> or <i>xfs</i>.</p>
<p>Note: The storage class parameters are used for creating an initial storage class for the <i>ubiquity-db</i> PVC.</p>	

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 a folder, named as `./ubiquity_collect_logs_MM-DD-YYYY-h:m:s`. The folder is placed in the directory, from which the log collection command was run.
- 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`. You can change this directory by configuring **FLEX-LOG-DIR** parameter in the `ubiquity-configmap.yml` file, as detailed in “Updating configuration files” on page 27.
- 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 **ubiquity_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.

Note: The output of the **ubiquity_cli.sh -a status_wide** is similar to the **./ubiquity_cli.sh -a status** output, illustrated in the *What to do next* section of “Performing installation” on page 10.

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 9. 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: <ul style="list-style-type: none"> • Controller-manager 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 <i>ubiquitytest</i> (Kubernetes 1.6 or 1.7 only)	Restart the <i>kubelet</i> on Kubernetes worker and master nodes. See “Performing installation” on page 10.
IBM Storage Kubernetes FlexVolume log for the worker node that runs the pod contains errors, related to WWN identification in the <i>multipath -ll</i> output	Check that: <ul style="list-style-type: none"> • 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.

Table 9. Troubleshooting for IBM Storage Enabler for Containers (continued)

Description	Corrective action
No connectivity between the FlexVolume pod and the IBM Storage Enabler for Containers server	<p>Log into the node and run the FlexVolume in a test mode (<code>\$> /usr/libexec/kubernetes/kubelet-plugins/volume/exec/ibm~ubiquity-k8s-flex/ubiquity-k8s-flex testubiquity</code>).</p> <p>If there is an error, make sure the IP of <i>ubiquity</i> service is the same as configured in the <i>ubiquity-configmap.yml</i> file. If not, configure the IP properly, then delete the FlexVolume DaemonSet and re-create it to apply the new address value.</p>
Failure to mount a storage volume to a Kubernetes node	<p>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.</p>
IBM Storage Enabler for Containers database fails to achieve the <i>Running</i> status after the configured timeout expires	<ul style="list-style-type: none"> • Check the <i>kubectl</i> 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 <i>kubectl</i> logs for the FlexVolume pod that runs on the master node. Check any error related to attachment of the <i>ibm-ubiquity-db</i> 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 <i>ibm-ubiquity-db</i> status is <i>Running</i>.
IBM Storage Enabler for Containers database persists in the <i>Creating</i> status. In addition, the Volume has not been added to the list of VolumesInUse in the node's volume status message is stored in <code>/var/log/message</code> file on the node, where the database is deployed.	<p>To resolve this, move <code>kube-controller-manager.yaml</code> out and into <code>/etc/kubernetes/manifests/</code> to be recreated the control-manager pod:</p> <pre> 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. </pre>
Persistent volume remains in the Delete state, failing to release	<p>Review the Provisioner log (<code>\$> kubectl logs -n ubiquity deploy/ubiquity-k8s-provisioner</code>) to identify the reason for deletion failure. Use the <code>\$ kubectl delete</code> command to delete the volume. Then, contact the storage administrator to remove the persistent volume on the storage system itself.</p>
Communication link between IBM Storage Dynamic Provisioner and other solution elements fails due to Provisioner token expiration	<p>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 <i>service-account</i> token without expiration time. You can replace the current token with the <i>service-account</i> token, as follows:</p> <pre> \$> 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} </pre>

Table 9. Troubleshooting for IBM Storage Enabler for Containers (continued)

Description	Corrective action
<p>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=}]]"</p>	<p>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.</p>
<p>Status of a stateful pod on a malfunctioned (crashed) node is <i>Unknown</i></p>	<p>Manually recover the crashed node, as described in the "Recovering a crashed Kubernetes node" on page 26 section.</p>
<p>A pod becomes unresponsive, persisting in the <i>ContainerCreating</i> 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.</p> <p>The failure occurs because the mountPoint already exists on this node. This might happen due to earlier invalid pod deletion.</p>	<p>Manually recover the pod, using the following procedure:</p> <ol style="list-style-type: none"> 1. Check if there is a symbolic link to the mountPoint by running <code>\$> ls -l /var/lib/kubelet/pods/<pod_ID>/volumes/ibm~ubiquity-k8s-flex/<PVC_ID></code>. 2. If the file exists and there is a symbolic link to the /ubiquity/<PVC_WWN>, remove it by running <code>rm /var/lib/kubelet/pods/<pod_ID>/volumes/ibm~ubiquity-k8s-flex/<PVC_ID></code>. 3. Unmount the PV by running <code>umount /ubiquity/<PVC_WWN></code>. 4. Wait for several minutes for Kubernetes to rerun the mountFlow. Then, at the end of the process, display the FlexVolume log by running <code>kubect1 logs -n ubiquity ubiquity-k8s-flex-<pod_ID_on_the_node></code> to verify the <i>Running</i> status of the pod.
<p>A pod becomes unresponsive, persisting in the <i>ContainerCreating</i> 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.</p>	<p>Restart the <i>multipathd</i> service by running the service multipathd restart command on the worker node, where the pod was scheduled.</p>

Chapter 3. IBM Spectrum Scale deployment

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

- “Installation”
- “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 52

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”
- “Managing SSL certificates” on page 37
- “Performing installation” on page 38

For information about uninstallation, see “Uninstalling the software package” on page 41.

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 of architecture x86 or ppc64le.
- 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 all the worker nodes are running RedHat Enterprise Linux (RHEL). For supported RHEL version information, check the IBM Spectrum Scale support matrix at IBM Spectrum Scale FAQs.
- Ensure 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 following command to ensure that the GPFS file systems are mounted before starting Kubernetes on the nodes:

```
# mmlsmount all -L
```

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 following command to ensure that the quota is enabled on the file systems:

```
# mmlsfs gpfs0 -Q
```

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 don't see this then run

If you do not get this output, then run the following command:

```
# mmchfs gpfs0 -Q yes
```

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

```
curl -k -u gui_admin_user:gui_adm_user_password -X GET --header 'accept:application/json' 'https://gui_node:443/scalegmt/v2/filesystems'
```

The command gives an output similar to the following:

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

- Run the following command to set the **enforceFilesetQuotaOnRoot** value to yes to ensure that quotas are enforced for the PVC created with root user ID:

```
mmchconfig enforceFilesetQuotaOnRoot=yes
```

- Run the following command to ensure that the Kubernetes are not installed or configured on the nodes running the IBM Spectrum Scale GUI . This is done to avoid port number conflicts and memory usage concerns:

```
# mmlsnodeclass
```

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 closely tied to memory usage, such as the **mmchconfig** options `pagepool` and `maxFilesToCache`.

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

Managing SSL certificates

The IBM Storage Enabler for Containers supports two SSL modes, when communicating with its components: the `require` mode and the `verify-full` mode.

You can choose one of the following SSL modes:

require

This SSL mode is activated when no validation is required. The IBM Storage Enabler for Containers server generates self-signed certificates as required during run-time. In this mode, you can skip the procedure detailed below and continue from “Performing installation” on page 38.

verify-full

This SSL mode is activated when the user is expected to provide the relevant certificates. When enabled, this SSL mode requires additional configuration steps.

Follow these steps to manage the SSL certificates when using the `verify full` mode:

1. When operating in the `verify-full` mode, you need to generate the following three pairs of the public-private keys:
 - IBM Storage Enabler for Containers (`ubiquity`) service object.
 - IBM Storage Enabler for Containers database (`ubiquity-db`) service object.
 - IBM Spectrum Scale Management API server (GUI) – This might already exist.
2. Verify the following:
 - 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 server (GUI), `ubiquity` service object, and `ubiquity-db` service object.

Run the following script to obtain the required network parameters for the ubiquity and ubiquity-db services:

```
$> ./ubiquity_installer.sh -s create-services
```

The script generates two Kubernetes services, which provide the required DNS/IP address combinations.

Note: Ensure that the following conditions are met before running the installation script:

- The private certificate and certificate key files must 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:
 - The file name spectrumscale-trusted-ca.crt for the IBM Spectrum Scale Management API server (GUI).
 - The file name ubiquity-trusted-ca.crt for the ubiquity service object.
 - The file name ubiquity-db-trusted-ca.crt for the ubiquity-db service object.
- All the *.crt and *.key files generated are copied into a dedicated directory.

-
3. Run the `$> ubiquity_installer.sh -s create-secrets-for-certificates -t <certificatedirectory>` command to create the following ConfigMap and secrets:
 - The 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.

Performing installation

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.

About this task

A single IBM Storage Enabler for Containers instance can be installed per one Kubernetes cluster.

Procedure

1. Download the installer.

IBM Storage Enabler for Containers is available as a free software solution for IBM Spectrum Scale customers. You can download the latest version of the installer for IBM Storage Enabler for Container from the IBM Fix Central.
2. Copy the installer to a local folder on a host that can manage the Kubernetes cluster. The `kubect1` command can be used to deploy and manage applications on Kubernetes.

Note: Usually, a master node has access to the Kubernetes cluster.

3. Extract the installer file `# tar -xzvf installer-for-ibm-storage-enabler-for-containers-x.y.z-*.tar.gz`.

Note: * represents the build number.

4. Update the `ubiquity_installer_scale.conf` configuration file according to your environment requirements. Replace the VALUE placeholders in the files with your values. The following parameters are applicable to this configuration file:

Table 10. Configuration parameters in `ubiquity_installer_scale.conf`

Parameter	Description
UBIQUITY_IMAGE	This parameter defines the name of the Docker image file for the Storage Enabler for Containers code to be deployed as Kubernetes deployment/ubiquity. It is set by default to <code>ibmcom/ibm-storage-enabler-for-containers:x.y.z</code> . The image is available in the Docker Hub.
UBIQUITY_DB_IMAGE	This parameter defines the name of the Docker image file for the Storage Enabler for Containers code to be deployed as Kubernetes deployment/ubiquity-db. It is set by default to <code>ibmcom/ibm-storage-enabler-for-containers-db:x.y.z</code> . The image is available in the Docker Hub.
UBIQUITY_K8S_PROVISIONER_IMAGE	This parameter defines the name of the Docker image file for the Storage Enabler for Containers code to be deployed as Kubernetes deployment/ubiquity-k8s-provisioner. It is set by default to <code>ibmcom/ibm-storage-dynamic-provisioner-for-kubernetes:x.y.z</code> . The image is available in the Docker Hub.
UBIQUITY_K8S_FLEX_IMAGE	This parameter defines the name of the Docker image file for the IBM Storage FlexVolume to be deployed as Kubernetes daemonset/ubiquity-k8s-flex. It is set by default to <code>ibmcom/ibm-storage-flex-volume-for-kubernetes:x.y.z</code> . The image is available in the Docker Hub.
SPECTRUMSCALE_MANAGEMENT_IP_VALUE	IP address or FQDN of the IBM Spectrum Scale Management API(GUI) server.
SPECTRUMSCALE_MANAGEMENT_PORT_VALUE	Communication port of the IBM Spectrum Scale Management API(GUO) server. Default value is 443.
SPECTRUMSCALE_DEFAULT_FILESYSTEM_NAME_VALUE	Default IBM Spectrum Scale file system to be used for creating persistent volumes, if not specified by the storage class. This file system is used to create persistent volume for <code>ubiquity-db</code> .
LOG_LEVEL_VALUE	Log level. Allowed values: <code>debug</code> , <code>info</code> , <code>error</code> . Default value is <code>info</code> .

Table 10. Configuration parameters in `ubiquity_installer_scale.conf` (continued)

Parameter	Description
SSL_MODE_VALUE	<p>SSL verification mode.</p> <p>Allowed values:</p> <p>require No validation is required, the IBM Storage Enabler for containers server generates self-signed certificates on the fly.</p> <p>verify-full Certificates are provided by the user.</p> <p>Note: The verify-full mode requires additional configuration steps, as detailed in “Managing SSL certificates” on page 37.</p>
SPECTRUMSCALE_USERNAME_VALUE	Username defined for the IBM Storage Enabler for Containers interface in IBM Spectrum Scale Management API server(GUI).
SPECTRUMSCALE_PASSWORD_VALUE	Password defined for the IBM Storage Enabler for Containers interface in IBM Spectrum Scale Management API server(GUI).
UBIQUITY_DB_PV_NAME_VALUE	Name of the persistent volume to be used for the ubiquity-db database.
UBIQUITY_DB_USERNAME_VALUE	Username for the deployment of ubiquity-db database.
UBIQUITY_DB_PASSWORD_VALUE	Password for the deployment of ubiquity-db database.
FLEX_LOG_DIR_VALUE	Directory for storing the <code>ubiquity-k8s-flex.log</code> file. Set by default to <code>/var/log</code> .
STORAGE_CLASS_NAME_VALUE	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.

Important: Any change to be made after running the installation script must be performed manually in the corresponding yml files themselves.

- Apply the `ubiquity_installer_scale.conf` settings to the relevant yml files of the installer. Run the `$> ./ubiquity_installer.sh -s update-ymls -c ubiquity_installer_scale.conf` command to do this. The following code displays an example of successfully configured yml files:

```
./ubiquity_installer.sh -s update-ymls -c ubiquity_installer_scale.conf
Executing STEP [update-ymls]...
Updating yml files with placeholders from ubiquity_installer_scale.conf file. Are you sure (y/n): y
Update placeholder [UBIQUITY_IMAGE] in files : ./ymls/ubiquity-deployment.yml
Update placeholder [UBIQUITY_DB_IMAGE] in files : ./ymls/ubiquity-db-deployment.yml
Update placeholder [UBIQUITY_K8S_PROVISIONER_IMAGE] in files : ./ymls/ubiquity-k8s-provisioner-deployment.yml
```

```

Update placeholder [UBIQUITY_K8S_FLEX_IMAGE      ] in files : ./ymls/ubiquity-k8s-flex-daemonset.yml
Update placeholder [SPECTRUMSCALE_MANAGEMENT_IP_VALUE] in files : ./ubiquity-configmap.yml
Update placeholder [SPECTRUMSCALE_MANAGEMENT_PORT_VALUE] in files : ./ubiquity-configmap.yml
Update placeholder [SPECTRUMSCALE_DEFAULT_FILESYSTEM_NAME_VALUE] in files :
./ymls/storage-class-spectrumScale.yml ./ubiquity-configmap.yml
.....
Update placeholder [IBM_UBIQUITY_DB_PV_NAME_VALUE ] in files : ./ymls/ubiquity-db-pvc.yml ./ubiquity-configmap.yml
Update placeholder [FLEX_LOG_DIR_VALUE          ] in files : ./ymls/ubiquity-k8s-flex-daemonset.yml
./ymls/ubiquity-k8s-flex-daemonset.yml ./ubiquity-configmap.yml
Update placeholder [LOG_LEVEL_VALUE            ] in files : ./ubiquity-configmap.yml
Update placeholder [SSL_MODE_VALUE            ] in files : ./ubiquity-configmap.yml
Update (base64) placeholder [SPECTRUMSCALE_USERNAME_VALUE ] in files : ./spectrumScale-credentials-secret.yml
Update (base64) placeholder [SPECTRUMSCALE_PASSWORD_VALUE ] in files : ./spectrumScale-credentials-secret.yml
Update (base64) placeholder [UBIQUITY_DB_USERNAME_VALUE ] in files : ./ubiquity-db-credentials-secret.yml
Update (base64) placeholder [UBIQUITY_DB_PASSWORD_VALUE ] in files : ./ubiquity-db-credentials-secret.yml
Update placeholder [STORAGE_CLASS_NAME_VALUE ] in files : ./ymls/storage-class-spectrumScale.yml
./ymls/storage-class.yml ./ymls/ubiquity-db-pvc.yml ./ymls/sanity_ymls/sanity-pvc.yml
Finished updating yml files according to ubiquity_installer_scale.conf

```

6. Run the `$> ./ubiquity_installer.sh -s install` command to start the installation.

It is recommended to remove the plain text password specified for **SPECTRUMSCALE_PASSWORD_VALUE** and **UBIQUITY_DB_PASSWORD_VALUE** in the `ubiquity_installer_scale.conf` file immediately after the installation.

What to do next

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

1. Run the `$> ./ubiquity_cli.sh -a status` command to verify the post-installation status of the IBM Storage Enabler for containers service. Check that the database status is set to Bound, and all other elements are available and running.

Note: You can use the `$> ./ubiquity_cli.sh -a status_wide` command to display the full system status. You can use it to verify that there is an `ubiquity-k8s-flexpod` on each worker and master node.

2. Run the `$> ./ubiquity_cli.sh -a sanity` command to perform a test that verifies the expected behavior of a pod creating a PVC. When running this verification test, the following output is expected:


```
Sanity finished successfully (pvc1 and pod1 were successfully created and deleted).
```
3. In addition to the default storage class, define more Kubernetes storage classes, if needed. This default storage class, named as a value of the **STORAGE_CLASS_NAME_VALUE** parameter in the `ubiquity_installer_scale.conf` file, was created during the installation. The template for the setting storage classes is included in the `./ymls/templates/storage-class-spectrumScale-template.yml` file.

Uninstalling the software package

The uninstallation process removes the IBM Storage Enabler for Containers components, its metadata, user credentials, IBM Storage Enabler for Containers namespace and other elements.

Before you begin

Verify that there are no persistent volumes (PVs) that have been created using the IBM Storage Enabler for Containers. If you want to completely remove the IBM Storage Enabler for Containers software, use the following procedure:

Procedure

1. Logon to the host which contains the IBM Storage Enabler for Containers and navigate to the installation directory.

2. Run the `$. /ubiquity_uninstall.sh` script to completely uninstall IBM Storage Enabler for Containers. Once the uninstallation is complete, the following message is displayed:
"IBM Storage Enabler for Containers" uninstall finished.

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 autoloading is enabled.

Note: Use the `mmchconfig` command to enable autoloading:

```
mmchconfig autoloading=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 `mmmout` command to mount the IBM Spectrum Scale file systems again.

For more information on the `mmmout` 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 53.

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 53.

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 `mmsshutdown` command.

For more information on the `mmsshutdown` 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.”
- “Configuring storage classes, PVCs and pods” on page 44.
- “Updating the configuration files” on page 48.
- “Using an existing fileset for volume creation” on page 51.
- “Changing the IBM Spectrum Scale RESTful credentials” on page 52.

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, Remote Mount, 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.
- 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.

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.

The template for creating storage classes is provided in the `./yamls/templates/storage-class-spectrumscale-template.yml` file as shown:

```
# cat yamls/templates/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 11. 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 <i>fileset</i> .
product	Permanently set to <i>ibm-storage-enabler-for-containers</i> .
provisioner	Permanently set to <i>ubiquity/flex</i> .
backend	Permanently set to <i>spectrum-scale</i> .

Table 11. Configuration parameters in *storage-class-template.yml* for IBM Spectrum Scale (continued)

Parameter	Description
reclaimPolicy	<p>Optional parameter.</p> <p>The reclaim policy tells the cluster what to do with the volume after it has been released of its claim.</p> <p>Persistent Volume inherit the reclaim policy from Storage Class.</p> <p>Permitted values :</p> <p>Delete If PVC is deleted then the PVC, PV and fileset that were created by IBM Storage Enabler for Containers are deleted.</p> <p>Note: This is the default value.</p> <p>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.</p>

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

The template for creating persistent volume claims is provided in the `./ymls/templates/pvc-template.yml` file. 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 12. Configuration parameters in *pvc-template.yml* for IBM Spectrum Scale

Parameter	Description
name	Persistent volume claim name.
storageClassName	Storage class name used for the PVC provisioning.

Table 12. Configuration parameters in *pvc-template.yml* for IBM Spectrum Scale (continued)

Parameter	Description
pv-name	<p>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.</p> <p>Optional parameter.</p> <p>This must be set for using existing fileset for creating persistent volumes along with the fileset parameter from storage-class.</p> <p>Note:</p> <p>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.</p>
accessModes	<p>Permitted values : <i>ReadWriteOnce</i> and <i>ReadWriteMany</i>. Other access modes are not supported.</p>
storage	<p>Volume size in Gb. Other volume size units are not supported.</p>

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: <yml volume name>
mountPath: <Mount point>
volumes:
- name: <yml volume name>
persistentVolumeClaim:
claimName: <PVC name>
```

You can set the following configuration parameters in the file:

Table 13. 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 files

If required, you can adjust configuration parameters of the IBM Storage Enabler for Containers after its installation, using the *yml* files on the host:

- `./ubiquity-configmap.yml`
- `./spectrum-scale-credentials-secret.yml`
- `./ubiquity-db-credentials-secret.yml`

Table 14. Configuration parameters in *ubiquity-configmap.yml*

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 <code>ubiquity-configmap</code> configMap (<code>\$> kubectl edit -n ubiquity configmap ubiquity-configmap</code>). 2. Delete the <code>ubiquity</code> pod of the <code>ubiquity</code> deployment. Then, it is automatically recreated by the deployment with the new parameters. Note: Ensure that PVC creation or deletion using IBM Storage Enabler for Containers is not in progress while deleting the <code>ubiquity</code> pod.
SPECTRUMSCALE-MANAGEMENT-PORT	Communication port of IBM Spectrum Scale management API (GUI) server. To update this parameter after installation, use the same procedure as the one for SPECTRUMSCALE-MANAGEMENT-IP .
SPECTRUMSCALE-DEFAULT-FILESYSTEM-NAME	Default IBM Spectrum Scale file system to be used for creating persistent volumes. Note: You cannot update this parameter.

Table 14. Configuration parameters in *ubiquity-configmap.yml* (continued)

Parameter	Description
UBIQUITY-IP-ADDRESS	<p>IP address of the <i>ubiquity</i> service object. The <i>ubiquity_installer.sh</i> automatically updates this parameter during the initial installation.</p> <p>If the <i>ubiquity</i> service is recreated, it receives a new IP address. As a result, this parameter must be updated to allow the FlexVolume pods to access it.</p> <p>To update the IP address after installation:</p> <ol style="list-style-type: none"> 1. Update the IP address in the <i>ubiquity_configmap configMap</i> (\$> kubectl edit -n ubiquity configmap ubiquity-configmap). 2. Delete all FlexVolume pods (\$> kubectl get pod -n ubiquity grep ubiquity-k8s-flex awk '{print \$1}' xargs kubectl delete pod -n ubiquity).
LOG-LEVEL	<p>Log level.</p> <p>Allowed values: <i>debug</i>, <i>info</i>, <i>error</i>.</p> <p>To receive more detailed events about the IBM Storage Enabler for Containers operation, you can adjust the log level after installation. Use this information your troubleshooting and debug processes.</p> <p>To change the log level after installation:</p> <ol style="list-style-type: none"> 1. Change the value in the <i>ubiquity_configmap configMap</i> (\$> kubectl edit -n ubiquity configmap ubiquity-configmap). 2. Stop the IBM Storage Enabler for Containers (\$> ubiquity_cli.sh -a stop). 3. Restart the IBM Storage Enabler for Containers (\$> ubiquity_cli.sh -a start).
SSL-MODE	<p>SSL verification mode.</p> <p>Allowed values: <i>require</i> (No validation is required, the IBM Storage Enabler for Containers server generates self-signed certificates on the fly.) or <i>verify-full</i> (Certificates are provided by the user.)</p> <p>To change the SSL mode to <i>verify-full</i> after installation:</p> <ol style="list-style-type: none"> 1. Stop the IBM Storage Enabler for Containers (\$> ubiquity_cli.sh -a stop). 2. Edit the <i>ubiquity-configmap.yml</i> file, setting the SSL_MODE to <i>verify-full</i>. 3. Edit the <i>ubiquity_installer_scale.conf</i> file, setting the SSL_MODE to <i>verify-full</i>. 4. Run \$> ubiquity_installer.sh -a update-ymls -c ubiquity_installer_scale.conf. 5. Create SSL certificates, as detailed in “Managing SSL certificates” on page 37. 6. Restart the IBM Storage Enabler for Containers (\$> ubiquity_cli.sh -a start).

Table 14. Configuration parameters in *ubiquity-configmap.yml* (continued)

Parameter	Description
FLEX-LOG-DIR	<p>Directory for the <code>ubiquity-k8s-flex.log</code> file. The default location is the <code>/var/log</code> directory.</p> <p>To change the <code>ubiquity-k8s-flex.log</code> file directory after installation:</p> <ol style="list-style-type: none"> 1. Change the default folder in the <code>ubiquity_configmap configMap</code> (<code>\$> kubectl edit -n ubiquity FLEX-LOG-DIR <new_directory></code>). 2. Edit the <code>ubiquity-k8s-flex-daemonset.yml</code> file, setting the <code>hostPath</code> and <code>mountPath</code> to the new directory value: <ul style="list-style-type: none"> • <code>kubectl delete -f yamls/ubiquity-k8s-flex-daemonset.yml -n ubiquity</code> • <code>kubectl create -f yamls/ubiquity-k8s-flex-daemonset.yml -n ubiquity</code> 3. If you change the default log file location, make sure to set it to the same value in the relevant yml files on the controller-manager pod.
FLEX-LOG-ROTATE-MAXSIZE	<p>Maximum size of the <code>ubiquity-k8s-flex.log</code> file.</p> <p>To change the default log file size after installation:</p> <ul style="list-style-type: none"> • Change the default log file size in the <code>ubiquity_configmap configMap</code> (<code>\$> kubectl edit -n ubiquity FLEX-LOG-ROTATE-MAXSIZE <new_size></code>).

Table 15. Configuration parameters in *spectrumscale-credentials-secret.yml* for IBM Spectrum Scale

Parameter	Description
username	<p>Username defined for the IBM Storage Enabler for Containers interface in IBM Spectrum Scale.</p> <p>To change the IBM Storage Enabler for Containers interface credentials after installation:</p> <ol style="list-style-type: none"> 1. Change the credentials for the IBM Storage Enabler for Containers interface, using the IBM Spectrum Scale GUI. 2. Enter the new Base-64-encoded username and password (<code>\$> kubectl edit -n ubiquity spectrumscale-credentials</code>). 3. Delete the <code>ubiquity-k8s-flex</code> pods, <code>ubiquity-k8s-provisioner</code> pod and <code>ubiquity</code> pod. Do not delete the <code>ubiquity-db</code> pod After the deletion, Kubernetes restarts the pods with new credentials. <p>Note: Ensure that PVC creation or deletion using IBM Storage Enabler for Containers is not in progress while deleting the <code>ubiquity</code>, <code>ubiquity-k8s-flex</code> or <code>ubiquity-k8s-provisioner</code> pods</p>
password	<p>Password defined for the IBM Storage Enabler for Containers interface in IBM Spectrum Scale. Instructions on how to change the interface credentials after installation are detailed in the username description above.</p>

Table 16. Configuration parameters in *ubiquity-db-credentials-secret.yml* for IBM Spectrum Scale

Parameter	Description
username	Username for the deployment of <i>ubiquity-db</i> database. Do not use the <i>postgres</i> username, because it already exists.
password	Password for the deployment of <i>ubiquity-db</i> database.
<p>Note: : If you change the <i>ubiquity-db</i> database password, make sure to set the new value for the PostgreSQL as well. Run the psql command from the IBM Storage Enabler for Containers container:</p> <pre>#>> psql " postgresql://ubiquity@ubiquity-db:5432/ubiquity?sslmode=require" ==> ALTER ROLE ubiquity password 'new password';</pre>	

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, 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 become 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:

- ```
kubect1 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:  

```
kubect1 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 **mmunmount** 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:  

```
kubect1 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:  

```
kubect1 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 **mmmoun**t 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.

---

## 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](http://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](http://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

SC27-9294-00

