eSDK Cloud Storage COSI Plugins V1.2.0

User Guide

Issue	01
Date	2025-09-30

HUAWEI TECHNOLOGIES CO., LTD.

Trademarks and Permissions

Notice

The purchased products, services and features are stipulated by the contract made between Huawei and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, either express or implied.

The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.

Huawei Technologies Co., Ltd.

Address: Huawei Industrial Base
Bantian, Longgang
Shenzhen 518129
People's Republic of China

Website: https://e.huawei.com

Security Declaration

Product Lifecycle

Huawei's regulations on product lifecycle are subject to the *Product End of Life Policy.* For details about this policy, visit the following web page:
https://support.huawei.com/ecolumnsweb/en/warranty-policy

Vulnerability

Huawei's regulations on product vulnerability management are subject to the *Vul. Response Process.* For details about this process, visit the following web page:
https://www.huawei.com/en/psirt/vul-response-process
For vulnerability information, enterprise customers can visit the following web page:
https://securitybulletin.huawei.com/enterprise/en/security-advisory

Initial Digital Certificate

The Initial digital certificates on Huawei devices are subject to the *Rights and Responsibilities of Initial Digital Certificates on Huawei Devices.* For details about this document, visit the following web page:
https://support.huawei.com/enterprise/en/bulletins-service/ENEWS2000015789

Huawei Enterprise End User License Agreement

This agreement is the end user license agreement between you (an individual, company, or any other entity) and Huawei for the use of the Huawei Software. Your use of the Huawei Software will be deemed as your acceptance of the terms mentioned in this agreement. For details about this agreement, visit the following web page:
https://e.huawei.com/en/about/eula

Lifecycle of Product Documentation

Huawei after-sales user documentation is subject to the *Product Documentation Lifecycle Policy.* For details about this policy, visit the following web page:
https://support.huawei.com/enterprise/en/bulletins-website/ENEWS2000017761

About This Document

Symbol Conventions

The symbols that may be found in this document are defined as follows.

Symbol	Description
⚠ DANGER	Indicates a hazard with a high level of risk which, if not avoided, will result in death or serious injury.
⚠ WARNING	Indicates a hazard with a medium level of risk which, if not avoided, could result in death or serious injury.
⚠ CAUTION	Indicates a hazard with a low level of risk which, if not avoided, could result in minor or moderate injury.
NOTICE	Indicates a potentially hazardous situation which, if not avoided, could result in equipment damage, data loss, performance deterioration, or unanticipated results. NOTICE is used to address practices not related to personal injury.
NOTE	Supplements the important information in the main text. NOTE is used to address information not related to personal injury, equipment damage, and environment deterioration.

Change History

Issue	Date	Description
01	2025-09-30	This issue is the first official release.

Contents

1 Overview

Container Object Storage Interface (COSI) is a group of abstract standard interfaces used to configure and manage object storage in the Kubernetes ecosystem. It aims to become a common abstraction layer for multiple object storage vendors so that workloads can request and automatically configure object storage buckets.

2 Compatibility and Features

2.1 Kubernetes Compatibility

Table 2-1 Supported container management platforms

Container Management Platform	Version
Kubernetes	1.25 to 1.33
Red Hat OpenShift Container Platform	4.13 to 4.18

NOTICE

- In all commands in this document, the Kubernetes container management platform as an example. If the Huawei COSI service is installed and used on the OpenShift platform, replace **kubectl** commands with **oc** commands. For example, replace the **kubectl get pods -n default** command with the **oc get pods -n default** command.

2.2 Compatibility with Huawei Storage

Table 2-2 Storage compatibility

Storage Product	Version
OceanStor Pacific series	8.1.5, 8.2.0, 8.2.1, V800R001C10

2.3 Feature Matrix

Table 2-3 Supported features and Kubernetes versions

Feature	V1.25+
Static Bucket Provisioning	√
Dynamic Bucket Provisioning	√
Bucket Access Granting	√
Bucket Access Revoking	√

Table 2-4 Supported features and protocols

Feature	AWS S3	GCS	Azure Blob
Static Bucket Provisioning	√	x	x
Dynamic Bucket Provisioning	√	x	x
Bucket Access Granting	√	x	x
Bucket Access Revoking	√	x	x

3 Installation Preparations

3.1 Obtaining Tools

Table 3-1 lists the tools required for software installation, configuration, and commissioning.

Table 3-1 Required tools

Tool	Description	How to Obtain
PuTTY	Cross-platform remote access tool. It is used to access a node running a Windows OS during software installation.	You can visit the chiark homepage to download the PuTTY software. You are advised to use PuTTY of the latest version to ensure successful login to the storage system.
WinSCP	Cross-platform file transfer tool. Use version 5.7.5 or later and select SCP during file transfer. It is used to transfer files between Windows and Linux.	You can visit the WinSCP homepage to download the WinSCP software.

3.2 Obtaining the Huawei COSI Software Package

Step 1 Before deploying services, you need to prepare the COSI software installation packages listed in **Table 3-2**. The following uses the **eSDK_Cloud_Storage_COSI_V1.1.1_X86_64.zip** software package as an example.

Table 3-2 Required software packages

Software Package	Description	How to Obtain
eSDK_Cloud_Storage_COSI_V1.1.1_X86_64.zip eSDK_Cloud_Storage_COSI_V1.1.1_ARM_64.zip	COSI software installation package.	https://github.com/Huawei/cosi/releases

Step 2 Run the **unzip /opt/***Software package name* command to decompress the software package. *Software package name* indicates the software package name. **Table 3-3** lists the structure of the software packages generated upon decompression.

```
# unzip /opt/eSDK_Cloud_Storage_COSI_V1.1.1_X86_64.zip -d /opt/huawei-cosi
```

Table 3-3 Component description

Component	Description
image/	Image provided by Huawei COSI.
helm/	Helm project used to deploy Huawei COSI.
examples/	.yaml sample file used during the use of Huawei COSI.

----End

3.3 Uploading a Huawei COSI Image

To use the COSI image on the container management platform, you need to import the COSI image to the cluster in advance using either of the following methods:

- (Recommended) Use Docker to upload the COSI image to the image repository.
- Manually import the COSI image to all nodes where Huawei COSI needs to be deployed.

3.3.1 Uploading an Image to the Image Repository

Prerequisites

A Linux host with Docker installed is available, and the host can access the image repository.

Procedure

Step 1 Use a remote access tool, such as PuTTY, to log in to the Linux host where Docker is installed through the management IP address.

Step 2 Obtain the software package by following the instructions in **3.2 Obtaining the Huawei COSI Software Package** and go to the **image** working directory.

Step 3 Run the **docker load -i huawei-cosi-driver-1.1.1.tar** command to import the COSI Driver image to the current node.

```
# docker load -i huawei-cosi-driver-1.1.1.tar
Loaded image: huawei-cosi-driver:1.1.1
```

Step 4 Run the **docker tag huawei-cosi-driver:1.1.1 repo.huawei.com/huawei-cosi-driver:1.1.1** command to add the image repository address to the image tag. **repo.huawei.com** indicates the image repository address.

```
# docker tag huawei-cosi-driver:1.1.1 repo.huawei.com/huawei-cosi-driver:1.1.1
```

Step 5 Run the **docker push repo.huawei.com/huawei-cosi-driver:1.1.1** command to upload the COSI image to the image repository. **repo.huawei.com** indicates the image repository address.

```
# docker push repo.huawei.com/huawei-cosi-driver:1.1.1
```

Step 6 Run the **docker load -i huawei-cosi-liveness-probe-1.1.1.tar** command to import the COSI Driver image to the current node.

```
# docker load -i huawei-cosi-liveness-probe-1.1.1.tar
Loaded image: huawei-cosi-liveness-probe:1.1.1
```

Step 7 Run the **docker tag huawei-cosi-liveness-probe:1.1.1 repo.huawei.com/huawei-cosi-liveness-probe:1.1.1** command to add the image repository address to the image tag. **repo.huawei.com** indicates the image repository address.

```
# docker tag huawei-cosi-liveness-probe:1.1.1 repo.huawei.com/huawei-cosi-liveness-probe:1.1.1
```

Step 8 Run the **docker push repo.huawei.com/huawei-cosi-liveness-probe:1.1.1** command to upload the COSI image to the image repository. **repo.huawei.com** indicates the image repository address.

```
# docker push repo.huawei.com/huawei-cosi-liveness-probe:1.1.1
```

----End

3.3.2 Uploading an Image to a Local Node

If the image has been uploaded to the image repository, skip this section.

Prerequisites

- Docker or another container engine has been installed on the node.

Procedure

Step 1 Use a remote access tool, such as PuTTY, to log in to the node where the image is to be imported through the management IP address.

Step 2 Obtain the software package by following the instructions in **3.2 Obtaining the Huawei COSI Software Package** and go to the **image** working directory.

Step 3 Run the following commands in sequence to import all Huawei COSI images in the image directory to the local node. In the commands, *name* indicates the name of a .tar image package.

Run the following command using the Docker container engine:

```
# docker load -i <name>.tar
```

Run the following command using the containerd container engine:

```
# ctr -n k8s.io image import <name>.tar
```

Run the following command using the Podman container engine:

```
# podman load -i <name>.tar
```

NOTICE

If another container engine is installed on the node, use the image import command for the corresponding container engine.

----End

3.4 Checking the Images on Which COSI Depends

The installation of Huawei COSI depends on the images listed in the following table. If all worker nodes in the cluster have been connected to the Internet and can pull images online, you can skip this section. If nodes in the cluster cannot connect to the Internet, download the corresponding image file based on the Kubernetes version and upload it to the image repository or import it to the worker nodes in the Kubernetes cluster.

Table 3-4 Images on which Huawei COSI depends

Container Name	Container Image	Feature Description
cosi-controller	gcr.io/k8s-staging-sig-storage/objectstorage-controller:v20250509-controllerv0.2.0-rc1-72-g945f40a	This image is provided by the Kubernetes community, used to manage the lifecycle of BucketClaim objects.
cosi-sidecar	gcr.io/k8s-staging-sig-storage/objectstorage-sidecar:v20250509-controllerv0.2.0-rc1-72-g945f40a	This image is provided by the Kubernetes community, used to manage the lifecycle of **Bucket** and **BucketAccess** objects.

Container Name	Container Image	Feature Description
huawei-cosi-driver	huawei-cosi-driver:1.1.1	This image is provided by Huawei COSI software package, used to provide all features supported by Huawei COSI.
livenessprobe	huawei-cosi-liveness-probe:1.1.1	This image is provided by Huawei COSI software package, used to provide the health check function of the Huawei COSI driver.

NOTE

The image file version v20250509-controllerv0.2.0-rc1-72-g945f40a used by cosi-controller and cosi-sidecar is a verified version. Replacing it with another version may cause unknown problems.

For details about how to download container images to the local host, see **7.1 How Do I Download a Container Image to the Local Host?**.

3.5 Installing Helm

NOTE

Currently, only Helm 3 is supported.

Helm is a software package management tool in the Kubernetes ecosystem. Similar to Advanced Packaging Tool (APT) of Ubuntu, Yellowdog Updater, Modified (YUM) of CentOS, or Package Installer for Python (PIP) of Python, Helm manages Kubernetes application resources. You can use Helm to package, distribute, install, upgrade, and roll back Kubernetes applications in a unified manner.

- For details about how to obtain and install Helm, **click here**.
- For other information about Helm, **click here**.
- For details about the version mapping between Helm and Kubernetes, **click here**.

3.6 Preparing the Configuration File

When using Helm, you need to prepare the **values.yaml** file based on the Huawei storage connected during deployment and the features to be used.

Procedure

Step 1 Use a remote access tool, such as PuTTY, to log in to any master node in the Kubernetes cluster through the management IP address.

Step 2 Run the **cd /opt/huawei-cosi/helm/** command to go to the Helm working directory.

Step 3 Run the **vi values.yaml** command to set parameters in the **values.yaml** file. After the modification is complete, press **Esc** and enter **:wq!** to save the modification. Table 3-5, Table 3-6, and Table 3-7 describe related parameters.

The **global** configuration items are used to configure the global information required by the system.

Table 3-5 global configuration items

Parameter	Description	Mandatory	Default Value
replicaCount	Number of Pod copies corresponding to the Deployment deployed using COSI.	No	1. It is recommended that the value be less than or equal to 2.
securityContext.runAsNonRoot	Whether the COSI container is run by a non-root user.	Yes	false **NOTICE** The **runAsUser** and **runAsGroup** parameters are available only when this parameter is set to **true**.
securityContext.runAsUser	ID of the user who runs the COSI container as a non-root user.	No	1000
securityContext.runAsGroup	ID of the user group that runs the COSI container as a non-root user.	No	1000
securityContext.enablePrivileged	Whether the COSI container runs as a privileged container.	Yes	true

Parameter	Description	Mandatory	Default Value
logging.module	Log recording module. The value can be: ● **file**: The run logs of the COSI container are persistently saved to the host where the container is running. ● **console**: COSI container logs are recorded in standard output mode.	Yes	file **NOTICE** The **fileSize** and **maxBackups** parameters are available only when this parameter is set to **file**.
logging.level	Log level.	No	info The value can be **debug**, **info**, **warning**, or **error**.
logging.fileSize	Log file size.	No	20 MB
logging.maxBackups	Maximum number of backup logs.	No	9

NOTICE

- According to the default parameter values of **securityContext** in the **global** configuration items, Huawei COSI container runs as the **root** user and privileged container by default. The purpose is to ensure that it can be properly installed and deployed on different container management platforms and that run logs can be persistently saved in the **/var/log/huawei-cosi** directory of the node host.
- If security requirements are posed for the running of Huawei COSI container, configure the container by following the instructions in **6.1 Parameter Configuration Guide for Huawei COSI Container with Minimum Running Permissions**.

The **deploy** configuration items are used to configure the deployment information required by COSI.

Table 3-6 deploy configuration items

Parameter	Description	Mandatory	Default Value	Remarks
cosiController.enabled	Whether to deploy the COSI Controller component.	Yes	true	-
cosiController.namespace	Namespace where the COSI Controller component is deployed.	Yes	huawei-cosi	-
cosiController.tolerations	Taint tolerations of the COSI Controller component. After this parameter is set, the Controller component can tolerate taints on a node.	No	None	For details about taints and tolerations, see **Taints and Tolerations**.
cosiController.nodeSelector	Node selector of the COSI Controller component. After this parameter is set, the Controller component will be scheduled only to a node with the label.	No	None	For details about the node selector, see **Assign Pods to Nodes**.
cosiController.affinity	Node affinity of the COSI Controller component. After this parameter is set, the Controller component will be preferentially scheduled to a node with the label.	No	None	For details about node affinity, see **Assigning Pods to Nodes**.
cosiProvisioner.namespace	Namespace where the COSI Provisioner component is deployed.	Yes	huawei-cosi	-
cosiProvisioner.driverName	Name of the driver corresponding to the COSI Provisioner component.	Yes	cosi.huawei.com	-
cosiProvisioner.tolerations	Taint tolerations of the COSI Provisioner component. After this parameter is set, the Provisioner component can tolerate taints on a node.	No	None	For details about taints and tolerations, see **Taints and Tolerations**.

Parameter	Description	Mandatory	Default Value	Remarks
cosiProvisioner.nodeSelector	Node selector of the COSI Provisioner component. After this parameter is set, the Provisioner component will be scheduled only to a node with the label.	No	None	For details about the node selector, see **Assign Pods to Nodes**.
cosiProvisioner.affinity	Node affinity of the COSI Provisioner component. After this parameter is set, the Provisioner component will be preferentially scheduled to a node with the label.	No	None	For details about node affinity, see **Assigning Pods to Nodes**.

The **images** configuration items are used to configure the image information required by COSI.

Table 3-7 images configuration items

Parameter	Description	Mandatory	Default Value
driver.cosiDriver	Image name of Huawei COSI Driver.	Yes	huawei-cosi-driver:1.1.1
driver.livenessProbe	Image name of Huawei COSI livenessProbe.	Yes	huawei-cosi-liveness-probe:1.1.1
controller.cosiController	Image name of COSI Controller.	Yes	gcr.io/k8s-staging-sig-storage/objectstorage-controller:v20250509-controllerv0.2.0-rc1-72-g945f40a
sidecar.cosiSidecar	Container monitoring interface image.	Yes	gcr.io/k8s-staging-sig-storage/objectstorage-sidecar:v20250509-controllerv0.2.0-rc1-72-g945f40a
images.imagePullPolicy.huaweiCosiDriverImagePullPolicy	Pull policy of Huawei COSI driver image.	Yes	IfNotPresent

Parameter	Description	Mandatory	Default Value
images.imagePullPolicy.huaweiCosiLivenessProbeImagePullPolicy	Pull policy of Huawei COSI driver health check image.	Yes	IfNotPresent
images.imagePullPolicy.cosiControllerImagePullPolicy	Pull policy of the COSI Controller image.	Yes	IfNotPresent
images.imagePullPolicy.cosiSidecarImagePullPolicy	Pull policy of the COSI sidecar image.	Yes	IfNotPresent
images.imagePullSecrets	Used by the Kubernetes cluster to pass the identity authentication of an image registry to pull private images.	No	For details, see **Pull an Image from a Private Registry**.

The **resources** configuration items are used to configure the resources used by COSI related containers.

Table 3-8 resources configuration items

Parameter	Description	Mandatory	Default Value
container.cosiDriver.requests.cpu	Minimum CPU resource of the cosiDriver container.	Yes	50m
container.cosiDriver.requests.memory	Minimum memory resource of the cosiDriver container.	Yes	128Mi
container.cosiDriver.limits.cpu	Maximum CPU resource of the cosiDriver container.	Yes	100m
container.cosiDriver.limits.memory	Maximum memory resource of the cosiDriver container.	Yes	256Mi
container.cosiLivenessProbe.requests.cpu	Minimum CPU resource of the cosiLivenessProbe container.	Yes	10m

Parameter	Description	Mandatory	Default Value
container.cosiLivenessProbe.requests.memory	Minimum memory resource of the cosiLivenessProbe container.	Yes	128Mi
container.cosiLivenessProbe.limits.cpu	Maximum CPU resource of the cosiLivenessProbe container.	Yes	100m
container.cosiLivenessProbe.limits.memory	Maximum memory resource of the cosiLivenessProbe container.	Yes	128Mi
container.cosiSidecar.requests.cpu	Minimum CPU resource of the cosiSidecar container.	Yes	50m
container.cosiSidecar.requests.memory	Minimum memory resource of the cosiSidecar container.	Yes	128Mi
container.cosiSidecar.limits.cpu	Maximum CPU resource of the cosiSidecar container.	Yes	100m
container.cosiSidecar.limits.memory	Maximum memory resource of the cosiSidecar container.	Yes	512Mi
container.cosiController.requests.cpu	Minimum CPU resource of the cosiController container.	Yes	50m
container.cosiController.requests.memory	Minimum memory resource of the cosiController container.	Yes	128Mi
container.cosiController.limits.cpu	Maximum CPU resource of the cosiController container.	Yes	100m
container.cosiController.limits.memory	Maximum memory resource of the cosiController container.	Yes	512Mi

----End

4 Installation and Deployment

4.1 Installing the Software

Prerequisites

- Helm 3 has been installed on the master node.
- The **values.yaml** file has been configured. For details, see **3.6 Preparing the Configuration File**.

Preparations

For the OpenShift platform, run the following commands to create the **SecurityContextConstraints** resource.

1. Run the **vi huawei-cosi-scc.yaml** command to create a **SecurityContextConstraints** file.

```
# vi huawei-cosi-scc.yaml
allowHostDirVolumePlugin: true
allowHostIPC: false
allowHostNetwork: false
allowHostPID: false
allowHostPorts: false
allowPrivilegeEscalation: true
allowPrivilegedContainer: true

apiVersion: security.openshift.io/v1
kind: SecurityContextConstraints
metadata:
  name: huawei-cosi-scc
runAsUser:
  type: RunAsAny
seLinuxContext:
  type: RunAsAny
fsGroup:
  type: RunAsAny
```

```
users:
- system:serviceaccount:huawei-cosi:huawei-cosi-provisioner-sa
volumes:
- hostpath
- emptyDir
- persistentVolumeClaim
- secret
- configMap
```

NOTICE

If the namespace where the COSI Provisioner component is deployed is modified in **3.6 Preparing the Configuration File**, modify the namespace parameter (that is, *huawei-cosi* in the preceding example) under the **users** configuration item in the **huawei-cosi-scc.yaml** file too.

2. Run the **oc create -f huawei-cosi-scc.yaml** command to create **SecurityContextConstraints**.

```
# oc create -f huawei-cosi-scc.yaml
securitycontextconstraints.security.openshift.io/huawei-cosi-scc created
```

Procedure

Step 1 Use a remote access tool, such as PuTTY, to log in to any master node in the Kubernetes cluster through the management IP address.

Step 2 Run the **cd /opt/huawei-cosi/helm** command to go to the Helm working directory.

Step 3 Run the **helm install huawei-cosi ./ -n huawei-cosi --create-namespace** command to install COSI services.

```
# helm install huawei-cosi ./ -n huawei-cosi --create-namespace
NAME: huawei-cosi
LAST DEPLOYED: Thu Aug 15 10:33:54 2024
NAMESPACE: huawei-cosi
STATUS: deployed
REVISION: 1
TEST SUITE: None
```

Step 4 Run the **kubectl get pod -n huawei-cosi** command to check whether the services are started.

```
# kubectl get pod -n huawei-cosi
NAME                                        READY   STATUS    RESTARTS   AGE
cosi-controller-cffb8c678-2lgj8             1/1     Running   0          5s
huawei-cosi-provisioner-77f4655456-7v5tk    3/3     Running   0          4s
```

----End

4.2 Uninstalling the Software

Prerequisites

COSI has been deployed using Helm 3.

Procedure

Step 1 Use a remote access tool, such as PuTTY, to log in to any master node in the Kubernetes cluster through the management IP address.

Step 2 Run the **helm uninstall huawei-cosi -n huawei-cosi** command to uninstall COSI services.

```
# helm uninstall huawei-cosi -n huawei-cosi
release "huawei-cosi" uninstalled
```

Step 3 Run the **kubectl delete ns huawei-cosi** command to delete the namespace.

```
# kubectl delete ns huawei-cosi
namespace "huawei-cosi" deleted
```

NOTICE

- Deleting a namespace will clear all resources in the namespace. Exercise caution when performing this operation.
- If you do not delete a namespace and need to install the COSI software again, run the **kubectl delete lease --all -n huawei-cosi** command to clear all Lease objects in the namespace. Otherwise, you need to wait for the Lease objects to release the holder when installing the software. In this case, services cannot be received for 2 to 3 minutes.

----End

4.3 Updating/Rolling Back the Software

4.3.1 Updating the Software

Scenario

This section describes how to update Huawei COSI service deployment parameters.

Prerequisites

COSI has been deployed using Helm 3.

Procedure

Step 1 Use a remote access tool, such as PuTTY, to log in to any master node in the Kubernetes cluster through the management IP address.

Step 2 Run the **cd /opt/huawei-cosi/helm** command to go to the Helm working directory.

Step 3 Run the **helm get values huawei-cosi -n huawei-cosi -a > update-value.yaml** command to obtain the original service configuration file.

Step 4 Run the **vi update-value.yaml** command to open the file and update the parameter values as required. After the modification is complete, press **Esc** and enter **:wq!** to save the modification. For details, see **3.6 Preparing the Configuration File**.

Step 5 Run the **helm upgrade huawei-cosi ./ -n huawei-cosi -f update-value.yaml --wait --timeout 2m** command to update COSI services. If **Release "huawei-cosi" has been upgraded** is displayed in the command output, the COSI services are successfully updated.

```
# helm upgrade huawei-cosi ./ -n huawei-cosi -f update-value.yaml --wait  --timeout 2
Release "huawei-cosi" has been upgraded. Happy Helming!
NAME: huawei-cosi
LAST DEPLOYED: Fri Aug 30 17:07:33 2024
NAMESPACE: huawei-cosi
STATUS: deployed
REVISION: 2
TEST SUITE: None
```

----End

4.3.2 Rolling Back the Update

Scenario

This section describes how to roll back Huawei COSI services to the source version.

Prerequisites

- COSI has been deployed using Helm 3.
- Huawei COSI has been updated using Helm 3.

Procedure

Step 1 Use a remote access tool, such as PuTTY, to log in to any master node in the Kubernetes cluster through the management IP address.

Step 2 Run the **helm history huawei-cosi -n huawei-cosi** command to query the historical versions of the Huawei COSI services deployed using Helm.

```
# helm history huawei-cosi -n huawei-cosi
REVISION  UPDATED                   STATUS       CHART        APP VERSION   DESCRIPTION
1         Fri Aug 30 11:41:19 2024  superseded   cosi-1.1.1   1.1.1         Install complete
2         Fri Aug 30 17:07:33 2024  deployed     cosi-1.1.1   1.1.1         Upgrade complete
```

Step 3 Run the **helm rollback huawei-cosi** *revision-number* **-n huawei-cosi --wait --timeout 2m** command to roll back the Huawei COSI services to the specified version. If **Rollback was a success** is displayed in the command output, the Huawei COSI services are successfully rolled back to the specified version.

In the preceding command, *revision-number* indicates a version number queried in **Step 2**. For example, the version is **1**.

```
# helm rollback huawei-cosi 1 -n huawei-cosi  --wait  --timeout 2m
Rollback was a success! Happy Helming!
```

----End

4.4 Upgrading/Rolling Back the Software

4.4.1 Upgrading the Software

Scenario

When upgrading the Huawei COSI service version, perform the operations described in this section.

Prerequisites

COSI has been deployed using Helm 3.

Precautions

During the upgrade, if the **values.yaml** and **update-value.yaml** files contain the same parameter settings, the parameters in the **update-value.yaml** file are preferentially used.

Procedure

Step 1 Use a remote access tool, such as PuTTY, to log in to any master node in the Kubernetes cluster through the management IP address.

Step 2 Import the new images to the corresponding node. For details, see **3.3 Uploading a Huawei COSI Image**.

Step 3 Run the **cd /opt/huawei-cosi/helm** command to go to the Helm working directory in the new installation package.

Step 4 Run the **helm get values huawei-cosi -n huawei-cosi -a > update-value.yaml** command to obtain the original service configuration file.

Step 5 Run the **vi update-value.yaml** command to open the file and update the images to the specified new version. After the modification is complete, press **Esc** and enter **:wq!** to save the modification. For details, see **Table 3-7**.

Step 6 Run the **helm upgrade huawei-cosi ./ -n huawei-cosi -f ./values.yaml -f update-value.yaml --wait --timeout 2m** command to upgrade COSI services. If **Release "huawei-cosi" has been upgraded** is displayed in the command output, the COSI services are successfully upgraded.

```
# helm upgrade huawei-cosi ./ -n huawei-cosi -f ./values.yaml -f update-value.yaml --wait  --timeout 2m
Release "huawei-cosi" has been upgraded. Happy Helming!
NAME: huawei-cosi
LAST DEPLOYED: Fri Aug 30 17:22:30 2024
NAMESPACE: huawei-cosi
STATUS: deployed
REVISION: 4
TEST SUITE: None
```

----End

4.4.2 Rolling Back the Upgrade

Prerequisites

- COSI has been deployed using Helm 3.

- COSI has been upgraded using Helm 3.

Procedure

Step 1 Use a remote access tool, such as PuTTY, to log in to any master node in the Kubernetes cluster through the management IP address.

Step 2 Run the **helm history huawei-cosi -n huawei-cosi** command to query the historical versions of the Huawei COSI services deployed using Helm.

```
# helm history huawei-cosi -n huawei-cosi
REVISION  UPDATED                   STATUS      CHART       APP VERSION   DESCRIPTION
1         Fri Aug 30 11:41:19 2024  superseded  cosi-1.1.1  1.1.1         Install complete
2         Fri Aug 30 17:07:33 2024  deployed    cosi-1.1.1  1.1.1         Upgrade complete
```

Step 3 Run the **helm rollback huawei-cosi** *revision-number* **-n huawei-cosi --wait --timeout 2m** command to roll back the Huawei COSI services to the specified version. If **Rollback was a success** is displayed in the command output, the Huawei COSI services are successfully rolled back to the specified version.

In the preceding command, *revision-number* indicates a version number queried in **Step 2**. For example, the version is **1**.

```
# helm rollback huawei-cosi 1 -n huawei-cosi  --wait  --timeout 2m
Rollback was a success! Happy Helming!
```

----End

5 Using Huawei COSI

5.1 Bucket Management

5.1.1 Dynamic Bucket Provisioning

To implement dynamic bucket provisioning, perform the following steps:

- Configuring a Secret for storing service plane account information
- Configuring a BucketClass
- Configuring a BucketClaim

5.1.1.1 Configuring a Secret for Storing Service Plane Account Information

The following is an example of configuration file **/opt/huawei-cosi/examples/accountsecret-service.yaml**:

```
kind: Secret
apiVersion: v1
metadata:
  name: sample-account-service-secret
  namespace: huawei-cosi
stringData:
  accessKey: <ak-value>
  secretKey: <sk-value>
  endpoint: <point-value>
```

Table 5-1 Secret configuration parameters

Parameter	Description	Mandatory	Default Value	Remarks
metadata.name	Name of the Secret object.	Yes	-	The value can contain digits, lowercase letters, hyphens (-), and periods (.), and must start and end with a letter or digit. A hyphen (-) cannot be adjacent to a period (.), and periods (.) cannot be adjacent to each other. The value can contain a maximum of 63 characters.
metadata.namespace	Namespace of the Secret object.	No	default	The name must consist of lowercase letters, digits, and hyphens (-), for example, **my-name** and **123-abc**.
stringData.accessKey	AK of the corresponding account on the storage side.	Yes	-	-
stringData.secretKey	SK of the corresponding account on the storage side.	Yes	-	-

Parameter	Description	Mandatory	Default Value	Remarks
stringData.endpoint	Endpoint of the service plane on the storage side.	Yes	-	The value can be a domain name + port number or an IP address + port number. Example: **https://*xx.xx.xx.xx*:5443**. If HTTPS is used, the port number must be set to **5443**. **NOTICE** If HTTP is used, use port **5080** and ensure that the object service supports HTTP requests on the storage device.
data.rootCA	Root certificate information, which is used to verify the certificate of the storage server.	No	-	Enter the certificate data encoded using Base64.

Step 1 Run the **cd /opt/huawei-cosi/examples/** command to go to the example file directory.

Step 2 Run the **vi accountsecret-service.yaml** command and configure the example configuration file according to **Table 5-1**.

Step 3 Run the **kubectl create -f accountsecret-service.yaml** command to create a Secret based on the prepared .yaml file.

```
# kubectl create -f accountsecret-service.yaml
secret/sample-account-service-secret created
```

Step 4 Run the **kubectl get secret sample-account-service-secret -n huawei-cosi** command to view information about the created Secret.

```
# kubectl get secret sample-account-service-secret -n huawei-cosi
NAME                             TYPE      DATA   AGE
sample-account-service-secret        Opaque       3      10s
```

----End

5.1.1.2 Configuring a BucketClass

The following is an example of configuration file **/opt/huawei-cosi/examples/bucketclass.yaml**:

```
kind: BucketClass
apiVersion: objectstorage.k8s.io/v1alpha1
```

```
metadata:
  name: sample-bucket-class
driverName: cosi.huawei.com
deletionPolicy: Delete
parameters:
  accountSecretName: sample-account-service-secret
  accountSecretNamespace: huawei-cosi
  bucketACL: <bucket-acl>
  bucketLocation: <bucket-location>
```

Table 5-2 BucketClass configuration parameters

Parameter	Description	Mandatory	Default Value	Remarks
metadata.name	User-defined name of a BucketClass object.	Yes	-	The value can contain digits, lowercase letters, hyphens (-), and periods (.), and must start and end with a letter or digit. A hyphen (-) cannot be adjacent to a period (.), and periods (.) cannot be adjacent to each other. **NOTICE** It is recommended that the value contain a maximum of 27 characters. Otherwise, the functions of the bucket resources provisioned on the storage side using the BucketClass may be restricted because the name contains more than 63 characters.
driverName	Name of the used driver.	Yes	-	Set this parameter to the driver name set during Huawei COSI installation. The value is the same as that of **driverName** in the **values.yaml** configuration file.

Parameter	Description	Mandatory	Default Value	Remarks
deletionPolicy	Bucket resource reclamation policy on the storage side. The value can be: ● Delete ● Retain	No	Retain	**Delete**: When a BucketClaim is deleted, the bucket resource on the storage side is also deleted. **Retain**: When a BucketClaim is deleted, the bucket resource on the storage side is retained.
parameters.accountSecretName	Name of the Secret object.	Yes	-	-
parameters.accountSecretNamespace	Namespace of the Secret object.	Yes	-	-

Parameter	Description	Mandatory	Default Value	Remarks
parameters.bucketACL	Bucket permission. The value can be: ● private ● public-read ● public-read-write ● authenticated-read	No	private	**private**: The owner of a bucket has the full control permission on the bucket. Other users have no permission to access the bucket. **public-read**: The owner of a bucket has the full control permission on the bucket. Other users, including anonymous users, have the read permission. **public-read-write**: The owner of a bucket has the full control permission on the bucket. Other users, including anonymous users, have the read and write permissions. **authenticated-read**: The owner of a bucket has the full control permission on the bucket. Other object service grantees have the read permission.
parameters.bucketLocation	Bucket storage region.	No	-	-

Step 1 Run the **cd /opt/huawei-cosi/examples/** command to go to the example file directory.

Step 2 Run the **vi bucketclass.yaml** command and configure the example configuration file according to **Table 5-2**.

Step 3 Run the **kubectl create -f bucketclass.yaml** command to create a BucketClass based on the prepared .yaml file.

```
# kubectl create -f bucketclass.yaml
bucketclass.objectstorage.k8s.io/sample-bucket-class created
```

Step 4 Run the **kubectl get bucketclass sample-bucket-class** command to view information about the created BucketClass.

```
# kubectl get bucketclass sample-bucket-class
NAME                    AGE
sample-bucket-class     10s
```

----End

5.1.1.3 Configuring a BucketClaim

The following is an example of configuration file **/opt/huawei-cosi/examples/bucketclaim.yaml**:

```
kind: BucketClaim
apiVersion: objectstorage.k8s.io/v1alpha1
metadata:
  name: sample-bucket-claim
  namespace: huawei-cosi
spec:
  bucketClassName: sample-bucket-class
  protocols:
    - s3
```

Table 5-3 BucketClaim configuration parameters

Parameter	Description	Mandatory	Default Value	Remarks
metadata.name	User-defined name of a BucketClaim object.	Yes	-	The value can contain digits, lowercase letters, hyphens (-), and periods (.), and must start and end with a letter or digit. A hyphen (-) cannot be adjacent to a period (.), and periods (.) cannot be adjacent to each other. The value can contain a maximum of 63 characters.
metadata.namespace	Namespace of the user-defined BucketClaim object.	No	default	Kubernetes namespace of the user-defined BucketClaim object. The name must consist of lowercase letters, digits, and hyphens (-), for example, **my-name** and **123-abc**.
spec.bucketClassName	Name of a BucketClass object.	Yes	-	-
spec.protocols	Protocol. The value can be: ● s3	Yes	-	-

Step 1 Run the **cd /opt/huawei-cosi/examples/** command to go to the example file directory.

Step 2 Run the **vi bucketclaim.yaml** command and configure the example configuration file according to **Table 5-3**.

Step 3 Run the **kubectl create -f bucketclaim.yaml** command to create a BucketClaim based on the prepared .yaml file.

```
# kubectl create -f bucketclaim.yaml
bucketclaim.objectstorage.k8s.io/sample-bucket-claim created
```

Step 4 Run the **kubectl get bucketclaim sample-bucket-claim -n huawei-cosi -o yaml** command to view information about the created BucketClaim. If the value of **status.bucketReady** in the BucketClaim is **true**, the BucketClaim is successfully created.

```
# kubectl get bucketclaim sample-bucket-claim -n huawei-cosi -o yaml
apiVersion: objectstorage.k8s.io/v1alpha1
kind: BucketClaim
metadata:
  creationTimestamp: "2024-09-25T07:10:37Z"
  finalizers:
  - cosi.objectstorage.k8s.io/bucketclaim-protection
  generation: 1
  name: sample-bucket-claim
  namespace: huawei-cosi
  resourceVersion: "166751963"
  uid: 53facdb1-9e9e-46eb-b59d-046b9982e78d
spec:
  bucketClassName: sample-bucket-class
  protocols:
  - s3
status:
  bucketName: sample-bucket-class53facdb1-9e9e-46eb-b59d-046b9982e78d
  bucketReady: true
```

----End

5.1.2 Static Bucket Provisioning

To implement static bucket provisioning, perform the following steps:

- Configuring a Secret for storing service plane account information
- Configuring a BucketClass
- Configuring a Bucket
- Configuring a BucketClaim

5.1.2.1 Configuring a Secret for Storing Service Plane Account Information

The following is an example of configuration file **/opt/huawei-cosi/examples/accountsecret-service.yaml**:

```
kind: Secret
apiVersion: v1
metadata:
  name: sample-account-service-secret
  namespace: huawei-cosi
stringData:
  accessKey: <ak-value>
  secretKey: <sk-value>
  endpoint: <point-value>
```

Table 5-4 Secret configuration parameters

Parameter	Description	Mandatory	Default Value	Remarks
metadata.name	Name of the Secret object.	Yes	-	The value can contain digits, lowercase letters, hyphens (-), and periods (.), and must start and end with a letter or digit. A hyphen (-) cannot be adjacent to a period (.), and periods (.) cannot be adjacent to each other. The value can contain a maximum of 63 characters.
metadata.namespace	Namespace of the Secret object.	No	default	The name must consist of lowercase letters, digits, and hyphens (-), for example, **my-name** and **123-abc**.
stringData.accessKey	AK of the corresponding account on the storage side.	Yes	-	-
stringData.secretKey	SK of the corresponding account on the storage side.	Yes	-	-

Parameter	Description	Mandatory	Default Value	Remarks
stringData.endpoint	Endpoint of the service plane on the storage side.	Yes	-	The value can be a domain name + port number or an IP address + port number. Example: **https://***xx.xx.xx.xx***:5443**. If HTTPS is used, the port number must be set to **5443**. **NOTICE** If HTTP is used, use port **5080** and ensure that the object service supports HTTP requests on the storage device.
data.rootCA	Root certificate information, which is used to verify the certificate of the storage server.	No	-	Enter the certificate data encoded using Base64.

Step 1 Run the **cd /opt/huawei-cosi/examples/** command to go to the example file directory.

Step 2 Run the **vi accountsecret-service.yaml** command and configure the example configuration file according to **Table 5-4**.

Step 3 Run the **kubectl create -f accountsecret-service.yaml** command to create a Secret based on the prepared .yaml file.

```
# kubectl create -f accountsecret-service.yaml
secret/sample-account-service-secret created
```

Step 4 Run the **kubectl get secret sample-account-service-secret -n huawei-cosi** command to view information about the created Secret.

```
# kubectl get secret sample-account-service-secret -n huawei-cosi
NAME                              TYPE          DATA   AGE
sample-account-service-secret          Opaque          3      10s
```

----End

5.1.2.2 Configuring a BucketClass

The following is an example of configuration file **/opt/huawei-cosi/examples/bucketclass.yaml**:

```
kind: BucketClass
apiVersion: objectstorage.k8s.io/v1alpha1
```

```
metadata:
  name: sample-bucket-class
driverName: cosi.huawei.com
deletionPolicy: Delete
parameters:
  accountSecretName: sample-account-service-secret
  accountSecretNamespace: huawei-cosi
  bucketACL: <bucket-acl>
  bucketLocation: <bucket-location>
```

Table 5-5 BucketClass configuration parameters

Parameter	Description	Mandatory	Default Value	Remarks
metadata.name	User-defined name of a BucketClass object.	Yes	-	The value can contain digits, lowercase letters, hyphens (-), and periods (.), and must start and end with a letter or digit. A hyphen (-) cannot be adjacent to a period (.), and periods (.) cannot be adjacent to each other. **NOTICE** It is recommended that the value contain a maximum of 27 characters. Otherwise, the functions of the bucket resources provisioned on the storage side using the BucketClass may be restricted because the name contains more than 63 characters.
driverName	Name of the used driver.	Yes	-	Set this parameter to the driver name set during Huawei COSI installation. The value is the same as that of **driverName** in the **values.yaml** configuration file.

Parameter	Description	Mandatory	Default Value	Remarks
deletionPolicy	Bucket resource reclamation policy on the storage side. The value can be: ● Delete ● Retain	No	Retain	**Delete**: When a BucketClaim is deleted, the bucket resource on the storage side is also deleted. **Retain**: When a BucketClaim is deleted, the bucket resource on the storage side is retained.
parameters.accountSecretName	Name of the Secret object.	Yes	-	-
parameters.accountSecretNamespace	Namespace of the Secret object.	Yes	-	-

Parameter	Description	Mandatory	Default Value	Remarks
parameters.bucketACL	Bucket permission. The value can be: ● private ● public-read ● public-read-write ● authenticated-read	No	private	**private**: The owner of a bucket has the full control permission on the bucket. Other users have no permission to access the bucket. **public-read**: The owner of a bucket has the full control permission on the bucket. Other users, including anonymous users, have the read permission. **public-read-write**: The owner of a bucket has the full control permission on the bucket. Other users, including anonymous users, have the read and write permissions. **authenticated-read**: The owner of a bucket has the full control permission on the bucket. Other object service grantees have the read permission.
parameters.bucketLocation	Bucket storage region.	No	-	-

Step 1 Run the **cd /opt/huawei-cosi/examples/** command to go to the example file directory.

Step 2 Run the **vi bucketclass.yaml** command and configure the example configuration file according to **Table 5-5**.

Step 3 Run the **kubectl create -f bucketclass.yaml** command to create a BucketClass based on the prepared .yaml file.

```
# kubectl create -f bucketclass.yaml
bucketclass.objectstorage.k8s.io/sample-bucket-class created
```

Step 4 Run the **kubectl get bucketclass sample-bucket-class** command to view information about the created BucketClass.

```
# kubectl get bucketclass sample-bucket-class
NAME                    AGE
sample-bucket-class     10s
```

----End

5.1.2.3 Configuring a Bucket

The following is an example of configuration file **/opt/huawei-cosi/examples/static-bucket.yaml**:

```
kind: Bucket
apiVersion: objectstorage.k8s.io/v1alpha1
metadata:
  name: sample-static-bucket
spec:
  bucketClaim: {}
  driverName: cosi.huawei.com
  bucketClassName: sample-bucket-class
  existingBucketID: <account-service-secret-namespace>/<account-service-secret-name>/<storage-existing-
bucket-name>
  deletionPolicy: Retain
  protocols:
  - s3
```

Table 5-6 Bucket configuration parameters

Parameter	Description	Mandatory	Default Value	Remarks
metadata.name	User-defined name of a static Bucket object.	Yes	-	The value can contain digits, lowercase letters, hyphens (-), and periods (.), and must start and end with a letter or digit. A hyphen (-) cannot be adjacent to a period (.), and periods (.) cannot be adjacent to each other. The value can contain a maximum of 63 characters.
spec.bucketClaim	Name of a BucketClaim object.	Yes	-	Set this parameter to **{}**.
spec.driverName	Driver name.	Yes	-	Set this parameter to the driver name set during Huawei COSI installation. The value is the same as that of **driverName** in the **values.yaml** file.

Parameter	Description	Mandatory	Default Value	Remarks
spec.bucketClassName	Name of a BucketClass object.	Yes	-	-
spec.existingBucketID	Existing bucket information. It consists of the namespace of the Secret object on the service plane in the cluster, the name of the Secret object on the service plane, and the existing bucket name on the storage side.	Yes	-	Format: <account-service-secret-namespace>/<account-service-secret-name>/<storage-existing-bucket-name> Example: secret-ns/secret-name/exist-bucket
spec.deletionPolicy	Bucket resource reclamation policy on the storage side. The value can be: • Delete • Retain	No	Retain	**Delete**: When a Bucket is deleted, the bucket resource on the storage side is also deleted. **Retain**: When a Bucket is deleted, the bucket resource on the storage side is retained.
spec.protocols	Protocol. The value can be: • s3	Yes	-	-

Step 1 Run the **cd /opt/huawei-cosi/examples/** command to go to the example file directory.

Step 2 Run the **vi static-bucket.yaml** command and configure the example configuration file according to **Table 5-6**.

Step 3 Run the **kubectl create -f static-bucket.yaml** command to create a Bucket based on the prepared .yaml file.

```
# kubectl create -f static-bucket.yaml
bucket.objectstorage.k8s.io/sample-static-bucket created
```

Step 4 Run the **kubectl get bucket sample-static-bucket -o yaml** command to view information about the created Bucket. If the value of **status.bucketReady** in the Bucket is **true**, the Bucket is successfully created.

```
# kubectl get bucket sample-static-bucket -o yaml
apiVersion: objectstorage.k8s.io/v1alpha1
```

```
kind: Bucket
metadata:
  creationTimestamp: "2024-09-25T07:34:26Z"
  finalizers:
  - cosi.objectstorage.k8s.io/bucket-protection
  generation: 2
  name: sample-static-bucket
  resourceVersion: "166754807"
  uid: ffc81c82-c8d1-4d48-946a-7191e52fda1a
spec:
  bucketClaim: {}
  bucketClassName: sample-bucket-class
  deletionPolicy: Retain
  driverName: cosi.huawei.com
  existingBucketID: huawei-cosi/sample-account-service-secret/bucket-xxx
  parameters:
    accountSecretName: sample-account-service-secret
    accountSecretNamespace: huawei-cosi
    bucketACL: private
  protocols:
  - s3
status:
  bucketID: huawei-cosi/sample-account-service-secret/bucket-xxx
  bucketReady: true
```

----End

5.1.2.4 Configuring a BucketClaim

The following is an example of configuration file **/opt/huawei-cosi/examples/static-bucketclaim.yaml**:

```
kind: BucketClaim
apiVersion: objectstorage.k8s.io/v1alpha1
metadata:
  name: sample-static-bucket-claim
  namespace: huawei-cosi
spec:
  bucketClassName: sample-bucket-class
  existingBucketName: sample-static-bucket
  protocols:
  - s3
```

Table 5-7 BucketClaim configuration parameters

Parameter	Description	Mandatory	Default Value	Remarks
metadata.name	User-defined name of a static BucketClaim object.	Yes	-	The value can contain digits, lowercase letters, hyphens (-), and periods (.), and must start and end with a letter or digit. A hyphen (-) cannot be adjacent to a period (.), and periods (.) cannot be adjacent to each other. The value can contain a maximum of 63 characters.

Parameter	Description	Mandatory	Default Value	Remarks
metadata.namespace	Namespace of the user-defined static BucketClaim object.	No	default	Kubernetes namespace of the user-defined BucketClaim object. The name must consist of lowercase letters, digits, and hyphens (-), for example, **my-name** and **123-abc**.
spec.bucketClassName	BucketClass name.	Yes	-	-
spec.existingBucketName	Name of a static Bucket.	Yes	-	**NOTICE** When creating multiple BucketClaim objects, do not bind them to the same static Bucket object.
spec.protocols	Protocol. The value can be: ● s3	Yes	-	-

Step 1 Run the **cd /opt/huawei-cosi/examples/** command to go to the example file directory.

Step 2 Run the **vi static-bucketclaim.yaml** command and configure the example configuration file according to **Table 5-7**.

Step 3 Run the **kubectl create -f static-bucketclaim.yaml** command to create a BucketClaim based on the prepared .yaml file.

```
# kubectl create -f static-bucketclaim.yaml
bucketclaim.objectstorage.k8s.io/sample-static-bucket-claim created
```

Step 4 Run the **kubectl get bucketclaim sample-static-bucket-claim -n huawei-cosi -o yaml** command to view information about the created BucketClaim. If **status.bucketName** in the BucketClaim is the name of the Bucket created in **5.1.2.3 Configuring a Bucket** and **status.bucketReady** is **true**, the BucketClaim is successfully created.

```
# kubectl get bucketclaim sample-static-bucket-claim -n huawei-cosi  -o yaml
apiVersion: objectstorage.k8s.io/v1alpha1
kind: BucketClaim
metadata:
  creationTimestamp: "2024-09-25T07:37:45Z"
  finalizers:
  - cosi.objectstorage.k8s.io/bucketclaim-protection
  generation: 1
  name: sample-static-bucket-claim
  namespace: huawei-cosi
  resourceVersion: "166755203"
  uid: 3e6dd528-074d-4194-9b20-46ddb409e757
spec:
  bucketClassName: sample-bucket-class
  existingBucketName: sample-static-bucket
  protocols:
  - s3
status:
  bucketName: sample-static-bucket
  bucketReady: true
```

----End

5.1.3 Bucket Reclamation

Prerequisites

A static or dynamic Bucket has been created, and the corresponding BucketClaim has been created.

Procedure

Step 1 Take the BucketClaim named **sample-bucket-claim** as an example. Run the **kubectl delete bucketclaim sample-bucket-claim -n huawei-cosi** command to reclaim the objects in the bucket.

```
# kubectl delete bucketclaim sample-bucket-claim -n huawei-cosi
bucketclaim.objectstorage.k8s.io "sample-bucket-claim" deleted
```

----End

NOTICE

When a static bucket is reclaimed, the value of **deletionPolicy** in the Bucket may be different from that in the BucketClass. When a bucket is reclaimed, the value of **deletionPolicy** in the Bucket is used.

5.2 Bucket Access Management

Prerequisites

A bucket has been provisioned.

5.2.1 Bucket Access Granting

To grant bucket access, perform the following steps:

- Configuring a Secret for storing management plane account information
- Configuring a BucketAccessClass
- Configuring a BucketAccess

5.2.1.1 Configuring a Secret for Storing Management Plane Account Information

The following is an example of configuration file **/opt/huawei-cosi/examples/accountsecret-management.yaml**:

```
kind: Secret
apiVersion: v1
```

```
metadata:
  name: sample-account-management-secret
  namespace: huawei-cosi
stringData:
  accessKey: <ak-value>
  secretKey: <sk-value>
  endpoint: <point-value>
```

Table 5-8 Secret configuration parameters

Parameter	Description	Mandatory	Default Value	Remarks
metadata.name	Name of the Secret object.	Yes	-	The value can contain digits, lowercase letters, hyphens (-), and periods (.), and must start and end with a letter or digit. A hyphen (-) cannot be adjacent to a period (.), and periods (.) cannot be adjacent to each other. The value can contain a maximum of 63 characters.
metadata.namespace	Namespace of the Secret object.	No	default	The name must consist of lowercase letters, digits, and hyphens (-), for example, **my-name** and **123-abc**.
stringData.accessKey	AK of the corresponding account on the storage side.	Yes	-	-
stringData.secretKey	SK of the corresponding account on the storage side.	Yes	-	-

Parameter	Description	Mandatory	Default Value	Remarks
stringData.endpoint	Endpoint of the management plane on the storage side.	Yes	-	The value can be an IP address + port number. Example: **https://***xx.xx.xx.xx***:9443**. The port number must be set to **9443**.
data.rootCA	Root certificate information, which is used to verify the certificate of the storage server.	No	-	Enter the certificate data encoded using Base64.

Step 1 Run the **cd /opt/huawei-cosi/examples/** command to go to the example file directory.

Step 2 Run the **vi accountsecret-management.yaml** command and configure the example configuration file according to **Table 5-8**.

Step 3 Run the **kubectl create -f accountsecret-management.yaml** command to create a Secret based on the prepared .yaml file.

```
# kubectl create -f accountsecret-management.yaml
secret/sample-account-management-secret created
```

Step 4 Run the **kubectl get secret sample-account-management-secret -n huawei-cosi** command to view information about the created Secret.

```
# kubectl get secret sample-account-management-secret -n huawei-cosi
NAME                                    TYPE                DATA   AGE
sample-account-management-secret             Opaque              3      10s
```

----End

5.2.1.2 Configuring a BucketAccessClass

The following is an example of configuration file **/opt/huawei-cosi/examples/bucketaccessclass.yaml**:

```
kind: BucketAccessClass
apiVersion: objectstorage.k8s.io/v1alpha1
metadata:
  name: sample-bucket-access-class
driverName: cosi.huawei.com
authenticationType: Key
parameters:
  accountSecretName: sample-account-management-secret
  accountSecretNamespace: huawei-cosi
  bucketPolicyModel: rw
```

Table 5-9 BucketAccessClass configuration parameters

Parameter	Description	Mandatory	Default Value	Remarks
metadata.name	User-defined name of a BucketAccessClass object.	Yes	-	The value can contain digits, lowercase letters, hyphens (-), and periods (.), and must start and end with a letter or digit. A hyphen (-) cannot be adjacent to a period (.), and periods (.) cannot be adjacent to each other. The value can contain a maximum of 63 characters.
driverName	Name of the used driver.	Yes	-	Set this parameter to the driver name set during Huawei COSI installation. The value is the same as that of **driverName** in the **values.yaml** file.
authenticationType	Authorization type. The value can be: ● Key	Yes	-	-
parameters.accountSecretName	Name of the Secret object.	Yes	-	-
parameters.accountSecretNamespace	Namespace of the Secret object.	Yes	-	-

Parameter	Description	Mandatory	Default Value	Remarks
parameters.bucketPolicyModel	Bucket policy. The value can be: ● ro ● rw	No	rw	**ro**: bucket policy in read mode, including the following s3 operations: s3:GetObject, s3:GetObjectVersion, s3:ListMultipartUploadParts, s3:GetObjectAcl, s3:GetObjectVersionAcl, s3:ListBucketVersions, s3:ListBucket, s3:ListBucketMultipartUploads **rw**: bucket policy in read/write mode, including the following s3 operations: s3:GetObject, s3:GetObjectVersion, s3:ListMultipartUploadParts, s3:GetObjectAcl, s3:GetObjectVersionAcl, s3:ListBucketVersions, s3:ListBucket, s3:ListBucketMultipartUploads, s3:AbortMultipartUpload, s3:PutObjectAcl, s3:DeleteObjectVersion, s3:PutObjectVersionAcl, s3:PutObject,s3:DeleteObject

Step 1 Run the **cd /opt/huawei-cosi/examples/** command to go to the example file directory.

Step 2 Run the **vi bucketaccessclass.yaml** command and configure the example configuration file according to **Table 5-9**.

Step 3 Run the **kubectl create -f bucketaccessclass.yaml** command to create a BucketAccessClass based on the prepared .yaml file.

```
# kubectl create -f bucketaccessclass.yaml
bucketclass.objectstorage.k8s.io/sample-bucket-access-class created
```

Step 4 Run the **kubectl get bucketaccessclass sample-bucket-access-class** command to view information about the created BucketAccessClass.

```
# kubectl get bucketaccessclass sample-bucket-access-class
NAME                                AGE
sample-bucket-access-class          10s
```

----End

5.2.1.3 Configuring a BucketAccess

The following is an example of configuration file **/opt/huawei-cosi/examples/bucketaccess.yaml**:

```
kind: BucketAccess
apiVersion: objectstorage.k8s.io/v1alpha1
metadata:
  name: sample-bucket-access
  namespace: huawei-cosi
spec:
  bucketClaimName: sample-bucket-claim
  bucketAccessClassName: sample-bucket-access-class
  credentialsSecretName: sample-cred-secret
  protocol: s3
```

Table 5-10 BucketAccess configuration parameters

Parameter	Description	Mandatory	Default Value	Remarks
metadata.name	User-defined name of a BucketAccess object.	Yes	-	The value can contain digits, lowercase letters, hyphens (-), and periods (.), and must start and end with a letter or digit. A hyphen (-) cannot be adjacent to a period (.), and periods (.) cannot be adjacent to each other. The value can contain a maximum of 63 characters.
metadata.namespace	Namespace where the BucketAccess object is located.	No	default	The name must consist of lowercase letters, digits, and hyphens (-), for example, **my-name** and **123-abc**.
spec.bucketClaimName	Name of the BucketClaim object to which access needs to be granted.	Yes	-	-

Parameter	Description	Mandatory	Default Value	Remarks
spec.bucketAccessClassName	Name of the BucketAccessClass object that needs to be used.	Yes	-	-
spec.credentialsSecretName	Name of the Secret object that stores the provisioned access credential information.	Yes	-	The value can contain digits, lowercase letters, hyphens (-), and periods (.), and must start and end with a letter or digit. A hyphen (-) cannot be adjacent to a period (.), and periods (.) cannot be adjacent to each other. The value can contain a maximum of 63 characters. **NOTICE** ● Enter the name of a Secret object that does not exist in the BucketAccess object namespace. ● If the configured Secret object already exists, the Secret object will be reused.
spec.protocol	Protocol. The value can be: ● s3	Yes	-	-

Step 1 Run the **cd /opt/huawei-cosi/examples/** command to go to the example file directory.

Step 2 Run the **vi bucketaccess.yaml** command and configure the example configuration file according to **Table 5-10**.

Step 3 Run the **kubectl create -f bucketaccess.yaml** command to create a BucketAccess based on the prepared .yaml file.

```
# kubectl create -f bucketaccess.yaml
bucketclass.objectstorage.k8s.io/sample-bucket-access created
```

Step 4 Run the **kubectl get bucketaccess sample-bucket-access -n huawei-cosi -o yaml** command to view information about the created BucketAccess. If the value of **status.accessGranted** in the BucketAccess is **true**, the BucketAccess is successfully created.

```
# kubectl get bucketaccess sample-bucket-access -n huawei-cosi -o yaml
apiVersion: objectstorage.k8s.io/v1alpha1
kind: BucketAccess
metadata:
  creationTimestamp: "2024-09-25T07:11:01Z"
  finalizers:
  - cosi.objectstorage.k8s.io/bucketaccess-protection
  generation: 1
  name: sample-bucket-access
  namespace: huawei-cosi
  resourceVersion: "166752017"
  uid: 64dd7898-5db3-4969-afce-0aee0c2cdfee
spec:
  bucketAccessClassName: sample-bucket-access-class
  bucketClaimName: sample-bucket-claim
  credentialsSecretName: sample-cred-secret
  protocol: s3
status:
  accessGranted: true
  accountID: huawei-cosi/sample-account-management-secret/ba-64dd7898-5db3-4969-afce-0aee0c2cdfee
```

Step 5 Run the **kubectl get secret sample-cred-secret -n huawei-cosi -o yaml** command to view details about the generated Secret object. For the BucketClaim named **sample-bucket-claim**, the provisioned bucket access credential information is stored in the **data.BucketInfo** field in Base64 encoding format.

```
# kubectl get secret sample-cred-secret -n huawei-cosi -o yaml
apiVersion: v1
data:
  BucketInfo:
eyJtZXRhZGF0YSI6eyJuYW1lIjoiYmMtZGJjZWJlN2ItMDMzMy00MTYwLThkMTYtMGMyNzcyZmQyMTk5IiwiY3J
lYXRpb25UaW1lc3RhbXAiOm51bGx9LCJzcGVjIjp7ImJ1Y2tldE5hbWUiOiJzYW1wbGUtYnVja2V0LWNsYXNzMD
g4OGNiOWYtYzMyYi00YjRiLWEwYmItYjA1MzNlNDg0ZjQyIiwiYXV0aGVudGljYXRpb25UeXBlIjoiS2V5Iiwic2Vjc
mV0UzMiOnsiZW5kcG9pbnQiOiJodHRwczovL3gueHgueHh4Lnh4eDo1NDQzIiwicmVnaW9uIjoiIiwiYWNjZXNz
S2V5SUQiOiIxMjM0NTY3OTg5IiwiYWNjZXNzU2VjcmV0S2V5IjoiMTIzNDU2Nzk4OSJ9LCJzZWNyZXRBenVyZSI
6bnVsbCwicHJvdG9jb2xzIjpbInMzIl19fQ==
kind: Secret
metadata:
  creationTimestamp: "2024-09-25T06:54:44Z"
  finalizers:
  - cosi.objectstorage.k8s.io/secret-protection
  name: sample-cred-secret
  namespace: huawei-cosi
  resourceVersion: "165711865"
  uid: 7d384522-aba9-4e87-b2c6-24f88d820fcd
type: Opaque
```

Step 6 Run the **echo "<bucketInfo>" | base64 -d** command to decode the BucketInfo information encoded using Base64.

```
# echo
"eyJtZXRhZGF0YSI6eyJuYW1lIjoiYmMtZGJjZWJlN2ItMDMzMy00MTYwLThkMTYtMGMyNzcyZmQyMTk5IiwiY3
JlYXRpb25UaW1lc3RhbXAiOm51bGx9LCJzcGVjIjp7ImJ1Y2tldE5hbWUiOiJzYW1wbGUtYnVja2V0LWNsYXNzMD
g4OGNiOWYtYzMyYi00YjRiLWEwYmItYjA1MzNlNDg0ZjQyIiwiYXV0aGVudGljYXRpb25UeXBlIjoiS2V5Iiwic2Vjc
mV0UzMiOnsiZW5kcG9pbnQiOiJodHRwczovL3gueHgueHh4Lnh4eDo1NDQzIiwicmVnaW9uIjoiIiwiYWNjZXNz
S2V5SUQiOiIxMjM0NTY3OTg5IiwiYWNjZXNzU2VjcmV0S2V5IjoiMTIzNDU2Nzk4OSJ9LCJzZWNyZXRBenVyZSI
6bnVsbCwicHJvdG9jb2xzIjpbInMzIl19fQ==" | base64 -d

{"metadata":{"name":"bc-dbcebe7b-0333-4160-8d16-0c2772fd2199","creationTimestamp":null},"spec":
{"bucketName":"sample-bucket-class0888cb9f-c32b-4b4b-a0bb-
b0533e484f42","authenticationType":"Key","secretS3":{"endpoint":"https://
x.xx.xxx.xxx:5443","region":"","accessKeyID":"1234567989","accessSecretKey":"1234567989"},"secretAzure":n
ull,"protocols":["s3"]}}
```

NOTICE

The encoding information in this step is simulated data. The actual data contains sensitive information. Exercise caution when performing this operation to avoid data security problems.

----End

5.2.2 Bucket Access Revoking

Prerequisites

Bucket access has been granted.

Procedure

Step 1 Take the BucketAccess named **sample-bucket-access** as an example. Run the **kubectl delete bucketaccess sample-bucket-access -n huawei-cosi** command to reclaim the access credentials of the bucket.

```
# kubectl delete bucketaccess sample-bucket-access -n huawei-cosi
bucketaccess.objectstorage.k8s.io "sample-bucket-access" deleted
```

----End

6 Security Hardening

6.1 Parameter Configuration Guide for Huawei COSI Container with Minimum Running Permissions

Context

According to the default parameter values of **securityContext** in the **global** configuration items in the **values.yaml** file, Huawei COSI container runs as the **root** user and privileged container by default. If security requirements are posed for the running of Huawei COSI container, you can configure Huawei COSI container to run with the minimum permissions by following the instructions in this section.

There are two scenarios:

- Scenario 1: The **/var/log/huawei-cosi** log directory is not planned on the host where the COSI container is running in advance. In this case, the **/var/log/huawei-cosi** log directory will be created when the COSI container is started.
- Scenario 2: The **/var/log/huawei-cosi** log directory is planned on the host where the COSI container is running in advance. In this case, the **/var/log/huawei-cosi** log directory is used when the COSI container is started.

Procedure for Scenario 1

Step 1 Configure the permissions and log recording module for Huawei COSI container running by following the instructions in **Table 3-5** and **Table 6-1**.

Table 6-1 Mapping between container running permissions and supported log recording modules

Container Management Platform	Whether the Container Runs as User root	Privileged Container Enabled or Not	Supported Log Recording Module
Kubernetes	√	√	file, console
Kubernetes	√	×	file, console
Kubernetes	×	√	console
Kubernetes	×	×	console
Red Hat OpenShift Container Platform	√	√	file, console
Red Hat OpenShift Container Platform	√	×	console
Red Hat OpenShift Container Platform	×	√	console
Red Hat OpenShift Container Platform	×	×	console

NOTICE

If the configured container running permission parameter does not match the supported log recording module, the container cannot be started due to insufficient permission.

----End

Procedure for Scenario 2

In this case, Huawei COSI container can run with the minimum permissions (non-root user/non-privileged container) and supports both file and console log recording modules.

Step 1 Use a remote access tool, such as PuTTY, to log in to a node in the Kubernetes cluster through the management IP address.

Step 2 If the container platform is Kubernetes, run the **mkdir -p /var/log/huawei-cosi && chmod 757 /var/log/huawei-cosi** command to create a log directory and set the DAC permission of the log directory to **757**.

```
# mkdir -p /var/log/huawei-cosi && chmod 757 /var/log/huawei-cosi
```

If the container platform is OpenShift, run the **mkdir -p /var/log/huawei-cosi && chmod 757 /var/log/huawei-cosi && chcon -t svirt_sandbox_file_t /var/log/**

huawei-cosi command to create a log directory, and set the DAC permission of the log directory to **757** and the SELinux permission to **svirt_sandbox_file_t**.

```
# mkdir -p /var/log/huawei-cosi && chmod 757 /var/log/huawei-cosi && chcon -t
svirt_sandbox_file_t /var/log/huawei-cosi
```

Step 3 Repeat the preceding steps to plan the **/var/log/huawei-cosi** log directory on the nodes where Huawei COSI container runs.

NOTICE

Ensure that the **/var/log/huawei-cosi** log directory has been planned for all nodes that may be scheduled by Huawei COSI container. If node failover occurs during the running of Huawei COSI container and the log directory is not planned for the new node where the container runs in advance, the container cannot be started due to insufficient permission.

----End

7 FAQs

7.1 How Do I Download a Container Image to the Local Host?

The following uses the **k8s.gcr.io/sig-storage/livenessprobe:v2.5.0** image as an example.

Downloading a Container Image Using containerd

Step 1 Run the **ctr image pull** *image:tag* command to download an image to the local host. In the command, *image:tag* indicates the image to be pulled and its tag.

```
# ctr image pull k8s.gcr.io/sig-storage/livenessprobe:v2.5.0
```

Step 2 Run the **ctr image export** *image***.tar** *image:tag* command to export the image to a file. In the command, *image:tag* indicates the image to be exported, and *image.tar* indicates the name of the exported image file.

```
# ctr image export livenessprobe.tar k8s.gcr.io/sig-storage/livenessprobe:v2.5.0
```

----End

Downloading a Container Image Using Docker

Step 1 Run the **docker pull** *image:tag* command to download an image to the local host. In the command, *image:tag* indicates the image to be pulled.

```
# docker pull k8s.gcr.io/sig-storage/livenessprobe:v2.5.0
```

Step 2 Run the **docker save** *image:tag* **-o** *image.tar* command to export the image to a file. In the command, *image:tag* indicates the image to be exported, and *image.tar* indicates the name of the exported image file.

```
# docker save k8s.gcr.io/sig-storage/livenessprobe:v2.5.0 -o livenessprobe.tar
```

----End

Downloading a Container Image Using Podman

Step 1 Run the **podman pull** *image:tag* command to download an image to the local host. In the command, *image:tag* indicates the image to be pulled.

```
# podman pull k8s.gcr.io/sig-storage/livenessprobe:v2.5.0
```

Step 2 Run the **podman save** *image:tag* **-o** *image.tar* command to export the image to a file. In the command, *image:tag* indicates the image to be exported, and *image.tar* indicates the name of the exported image file.

```
# podman save k8s.gcr.io/sig-storage/livenessprobe:v2.5.0 -o livenessprobe.tar
```

----End

7.2 How Do I View Huawei COSI Logs?

Viewing the Persistent Logs of the huawei-cosi-provisioner Service

Step 1 Run the **kubectl get pods -n** *namespace* **-o wide** command. In the command, *namespace* indicates the namespace where the huawei-cosi-provisioner service is deployed. Locate the node where the huawei-cosi-provisioner service is deployed based on the command output.

```
# kubectl get pods -n huawei-cosi -o wide
NAME                                        ···          NODE
huawei-cosi-provisioner-66f5747d8c-f8kxv     ···          <node-name>
```

Step 2 Use a remote access tool, such as PuTTY, to log in to the node where the huawei-cosi-provisioner service resides in the Kubernetes cluster through the management IP address.

Step 3 Run the **cd /var/log/huawei-cosi/cosi-driver/** command to go to the log directory.

```
# cd /var/log/huawei-cosi/cosi-driver/
```

Step 4 Run the **vi cosi-driver** command to view the persistent logs of the cosi-driver container.

```
# vi cosi-driver
```

Step 5 Run the **vi liveness-probe** command to view the persistent logs of the liveness-probe container.

```
# vi liveness-probe
```

----End

Viewing the Standard Output Logs of the huawei-cosi-provisioner Service Container

Step 1 Run the **kubectl get pods -n** *namespace* **-o wide** command. In the command, *namespace* indicates the namespace where the huawei-cosi-provisioner service is deployed. Locate the node where the huawei-cosi-provisioner service is deployed based on the command output.

```
# kubectl get pods -n huawei-cosi -o wide
NAME                                        ...       NODE
huawei-cosi-provisioner-66f5747d8c-f8kxv    ...          <node-name>
```

Step 2 Use a remote access tool, such as PuTTY, to log in to the node where the huawei-cosi-provisioner service resides in the Kubernetes cluster through the management IP address.

Step 3 Run the **cd /var/log/containers** command to go to the container log directory.

```
# cd /var/log/containers
```

Step 4 Run the **vi huawei-cosi-provisioner-<name>_huawei-cosi_huawei-cosi-driver-<container-id>.log** command to view the standard output logs of the huawei-cosi-driver container.

```
# vi huawei-cosi-provisioner-<name>_huawei-cosi_huawei-cosi-driver-<container-id>.log
```

NOTICE

You can use the same method to view the standard output logs of the cosi-controller, cosi-sidecar, and liveness-probe containers.

----End

7.3 How Do I Obtain the COSI Version?

Procedure

Step 1 Use a remote access tool, such as PuTTY, to log in to any master node in the Kubernetes cluster through the management IP address.

Step 2 Run the **kubectl get cm huawei-cosi-version -n** *namespace* **-o yaml** command. In the command, *namespace* indicates the namespace where the COSI Provisioner service is deployed.

```
# kubectl get cm huawei-cosi-version -n huawei-cosi -o yaml
apiVersion: v1
data:
  cosi-driver: 1.1.1
  liveness-probe: 1.1.1
kind: ConfigMap
metadata:
  creationTimestamp: "2024-08-16T08:18:30Z"
  name: huawei-cosi-version
  namespace: huawei-cosi
  resourceVersion: "159241105"
  uid: 689feb62-e327-4651-8db3-34417a219271
```

----End

7.4 COSI Sidecar and Controller Community Issues

Context

Currently, the Sidecar and Controller components provided by the COSI open-source community are in the alpha phase. For details about the problems encountered during the use, see the community issues and **8 Troubleshooting**.

Links for Community Issues

https://github.com/kubernetes-sigs/container-object-storage-interface/issues

8 Troubleshooting

8.1 After a BucketClaim Is Deleted from a Static Bucket Bound to Multiple BucketClaim Resources, Other BucketClaim Resources Become Abnormal

Symptom

For the same static bucket, **bucket-claim-1** and **bucket-claim-2** are created and bound to the bucket in sequence in the **huawei-cosi** namespace. The **bucket-claim-1** resource is deleted and the **kubectl get bucketclaim bucket-claim-2 -n huawei-cosi -o yaml** command is executed to check **bucket-claim-2**. It is found that the information about the bound static bucket still exists. However, the static bucket cannot be obtained by running the **kubectl get bucket** command.

Root Cause Analysis

For details, see the COSI community issue link at **https://github.com/kubernetes-sigs/container-object-storage-interface/issues/76**.

Solution or Workaround

Step 1 Use a remote access tool, such as PuTTY, to log in to any master node in the Kubernetes cluster through the management IP address.

Step 2 Run the following command to clear BucketClaim resources.

```
kubectl delete bucketClaim bucket-claim-2 -n huawei-cosi
```

Step 3 If a BucketClaim resource cannot be directly deleted, run the following command to remove finalizers from the BucketClaim.

```
kubectl patch bucketClaim bucket-claim-2 --type json --patch='[{"op":"remove","path": "/metadata/
finalizers"}]'
```

----End

NOTICE

When creating multiple BucketClaim objects, do not bind them to the same static bucket object. For details, see **Table 5-7**.

8.2 "Delete BucketAccess" Is Displayed Multiple Times in Logs After a BucketAccess Is Deleted in an Environment with Multiple Sidecar Components

Symptom

When the sidecar component is deployed twice in different ways in the cluster environment, two deletion operations are performed when a BucketAccess is deleted. In this case, the log information of cosi-sidecar in the **/var/log/containers** directory shows two "Delete BucketAccess" records, but only one BucketAccess is deleted.

Root Cause Analysis

For details, see the COSI community issue link at **https://github.com/kubernetes-sigs/container-object-storage-interface/issues/80**.

Solution or Workaround

NOTICE

Do not deploy multiple sidecar components in the cluster environment. For details, see **Table 3-6**.

8.3 When the BucketAccess Resources Reused by credentialsSecret Is Deleted, "poe client http call not success" Is Displayed in Logs

Symptom

In the **huawei-cosi** namespace, two BucketAccess resources (**bucket-access-1** and **bucket-access-2**) are created in sequence, and the same credentialsSecretName is set for them. Then the BucketAccess resources are deleted in sequence. However, the deletion of the second BucketAccess (**bucket-access-2**) is suspended. Error message "poe client http call not success" is displayed in the logs in **/var/log/huawei-cosi/cosi-driver**.

Root Cause Analysis

For details, see the COSI community issue link at **https://github.com/kubernetes-sigs/container-object-storage-interface/issues/83**.

Solution or Workaround

Step 1 Use a remote access tool, such as PuTTY, to log in to any master node in the Kubernetes cluster through the management IP address.

Step 2 Run the following command to clear BucketAccess resources.

```
kubectl delete bucketAccess bucket-access-2 -n huawei-cosi
```

Step 3 If a BucketAccess resource cannot be directly deleted, run the following command to remove finalizers from the BucketAccess.

```
kubectl patch bucketAccess bucket-access-2 --type json --patch='[{"op":"remove","path": "/metadata/
finalizers"}]'
```

Step 4 Use a browser to log in to the OceanStor Pacific storage GUI and manually delete related resources from the storage system.

----End

NOTICE

Do not reuse the credentialsSecret object. For details, see **Table 5-10**.

8.4 A Deployed Community Sidecar Occasionally Fails to Receive BucketAccess Creation Events

Symptom

The configuration of a BucketAccess created in the cluster environment needs to be modified. After the BucketAccess is deleted, a BucketAccess with the same name is created immediately. In this case, the cosi-sidecar log in the **/var/log/containers** directory does not contain "Add BucketAccess" and the BucketAccess fails to be created.

Root Cause Analysis

For details, see the COSI community issue link at **https://github.com/kubernetes-sigs/container-object-storage-interface/issues/82**.

Solution or Workaround

NOTICE

Change the name of the BucketAccess to be created.

8.5 Commands Cannot Be Received During Uninstallation and Reinstallation of Community Sidecar Applications

Symptom

After Huawei COSI is uninstalled and reinstalled, an existing BucketClaim resource needs to be deleted immediately. However, the BucketClaim cannot be deleted. In addition, error message "lock is held by xxx and has not yet expired" and "failed to acquire lease huawei-cosi/cosi-huawei-com-cosi" are displayed in the **cosi-sidecar** log in the **/var/log/containers** directory. In this case, the sidecar application cannot receive any command.

Root Cause Analysis

For details, see the COSI community issue link at **https://github.com/kubernetes-sigs/container-object-storage-interface/issues/77**.

Solution or Workaround

Step 1 Run the following commands to uninstall the Huawei COSI plug-in and delete the lease resources in the **huawei-cosi** space. For details, see **Procedure**.

```
helm uninstall huawei-cosi -n huawei-cosi
kubectl delete lease --all -n huawei-cosi
```

Step 2 Reinstall the Huawei COSI plug-in. For details, see **4.1 Installing the Software**.

----End

8.6 After a BucketClaim with Incorrect Configurations Is Created and Deleted, Creation Events Are Continuously Recorded in Sidecar Logs

Symptom

After a BucketClaim with incorrect configurations is created in the cluster, resources fail to be provisioned on the storage side, but BucketClaim resources are successfully created in the cluster. After the BucketClaim is deleted, the COSI sidecar application logs continuously record creation events. Error message "Unable to write event" is constantly recorded in the cosi-controller log in the **/var/log/containers** directory.

Root Cause Analysis

For details, see the COSI community issue link at **https://github.com/kubernetes-sigs/container-object-storage-interface/issues/81**.

Solution or Workaround

Run the following command to restart the Huawei COSI application.

```
kubectl delete pod -n huawei-cosi --all
```