eSDK Huawei Storage COSI Plugins V1.0.0

User Guide

Issue 01

Date 2024-10-09





Copyright © Huawei Technologies Co., Ltd. 2024. All rights reserved.

No part of this document may be reproduced or transmitted in any form or by any means without prior written consent of Huawei Technologies Co., Ltd.

Trademarks and Permissions

HUAWEI and other Huawei trademarks are trademarks of Huawei Technologies Co., Ltd.
All other trademarks and trade names mentioned in this document are the property of their respective holders.

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	2024-10-09	This issue is the first official release.

Contents

About This Document	iii
1 Overview	1
2 Compatibility and Features	2
2.1 Kubernetes Compatibility	
2.2 Compatibility with Huawei Storage	2
2.3 Feature Matrix	3
3 Installation Preparations	4
3.1 Obtaining Tools	
3.2 Obtaining the Huawei COSI Software Package	5
3.3 Uploading a Huawei COSI Image	5
3.3.1 Uploading an Image to the Image Repository	6
3.3.2 Uploading an Image to a Local Node	6
3.4 Checking the Images on Which COSI Depends	7
3.5 Installing Helm	8
3.6 Preparing the Configuration File	8
4 Installation and Deployment	14
4.1 Installing the Software	14
4.2 Uninstalling the Software	15
4.3 Updating the Software	16
4.4 Rolling Back the Update	17
4.5 Upgrading the Software	17
4.6 Rolling Back the Upgrade	18
5 Using Huawei COSI	20
5.1 Bucket Management	20
5.1.1 Dynamic Bucket Provisioning	20
5.1.1.1 Configuring a Secret for Storing Service Plane Account Information	20
5.1.1.2 Configuring a BucketClass	22
5.1.1.3 Configuring a BucketClaim	25
5.1.2 Static Bucket Provisioning	26
5.1.2.1 Configuring a Secret for Storing Service Plane Account Information	
5.1.2.2 Configuring a BucketClass	28

5.1.2.3 Configuring a Bucket	31
5.1.2.4 Configuring a BucketClaim	33
5.1.3 Bucket Reclamation	35
5.2 Bucket Access Management	35
5.2.1 Bucket Access Granting	35
5.2.1.1 Configuring a Secret for Storing Management Plane Account Information	
5.2.1.2 Configuring a BucketAccessClass	37
5.2.1.3 Configuring a BucketAccess	40
5.2.2 Bucket Access Revoking	43
6 Security Hardening	44
6.1 Parameter Configuration Guide for Huawei COSI Container with Minimum Running Permissions	44
7 FAQs	47
7.1 How Do I Download a Container Image to the Local Host?	47
7.2 How Do I View Huawei COSI Logs?	48
7.3 How Do I Obtain the COSI Version?	49
7.4 COSI Sidecar and Controller Community Issues	49

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
- 2.2 Compatibility with Huawei Storage
- 2.3 Feature Matrix

2.1 Kubernetes Compatibility

Table 2-1 Supported container management platforms

Container Management Platform	Version
Kubernetes	1.25 to 1.30
Red Hat OpenShift Container Platform	4.13, 4.14, 4.15

NOTICE

• In all commands in this document, the Kubernetes container management platform as an example. If the Huawei CDR 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

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	√	х	х
Dynamic Bucket Provisioning	√	х	х
Bucket Access Granting	√	х	х
Bucket Access Revoking	√	х	х

3 Installation Preparations

- 3.1 Obtaining Tools
- 3.2 Obtaining the Huawei COSI Software Package
- 3.3 Uploading a Huawei COSI Image
- 3.4 Checking the Images on Which COSI Depends
- 3.5 Installing Helm
- 3.6 Preparing the Configuration File

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_Huawei_Storage_COSI_V1.0.0_X86_64.zip** software package as an example.

Table 3-2 Required software packages

Software Package	Description	How to Obtain
eSDK_Huawei_Storage_COSI_V1. 0.0_X86_64.zip eSDK_Huawei_Storage_COSI_V1. 0.0_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_Huawei_Storage_COSI_V1.0.0_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-1.0.0.tar** command to import the COSI Driver image to the current node.

docker load -i huawei-cosi-driver-1.0.0.tar Loaded image: huawei-cosi-driver:1.0.0

Step 4 Run the **docker tag huawei-cosi-driver:1.0.0 repo.huawei.com/huawei-cosi-driver:1.0.0** command to add the image repository address to the image tag. **repo.huawei.com** indicates the image repository address.

docker tag huawei-cosi-driver:1.0.0 repo.huawei.com/huawei-cosi-driver:1.0.0

Step 5 Run the **docker push repo.huawei.com/huawei-cosi-driver:1.0.0** 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:1.0.0

Step 6 Run the **docker load -i huawei-cosi-liveness-probe-1.0.0.tar** command to import the COSI Driver image to the current node.

docker load -i huawei-cosi-liveness-probe-1.0.0.tar Loaded image: huawei-cosi-liveness-probe:1.0.0

- Step 7 Run the docker tag huawei-cosi-liveness-probe-1.0.0 repo.huawei.com/ huawei-cosi-liveness-probe:1.0.0 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:1.0.0 repo.huawei.com/huawei-cosi-liveness-probe:1.0.0
- Step 8 Run the docker push repo.huawei.com/huawei-cosi-liveness-probe:1.0.0 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:1.0.0

----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:latest	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:latest	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:1.0.0	This image is provided by Huawei COSI software package, used to provide all features supported by Huawei COSI.
livenessprobe	huawei-cosi-liveness- probe:1.0.0	This image is provided by Huawei COSI software package, used to provide the health check function of the Huawei COSI driver.

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

MOTE

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.

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	Man dato ry	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.	No	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. enablePrivilege d	Whether the COSI container runs as a privileged container.	No	true
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.	Yes	info The value can be debug , info, warning, or error.
logging.fileSize	Log file size.	Yes	20 MB
logging.maxBac kups	Maximum number of backup logs.	Yes	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
cosiController.e nabled	Whether to deploy the COSI Controller component.	No	true
cosiController.n amespace	Namespace where the COSI Controller component is deployed.	No	huawei-cosi
cosiProvisioner. namespace	Namespace where the COSI Provisioner component is deployed.	No	huawei-cosi
cosiProvisioner. driverName	Name of the driver corresponding to the COSI Provisioner component.	No	cosi.huawei.com

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.cosiDrive	Image name of	Yes	huawei-cosi-
r	Huawei COSI Driver.		driver:1.0.0.

Parameter	Description	Mandatory	Default Value
driver.livenessP robe	Image name of Huawei COSI livenessProbe.	Yes	huawei-cosi- liveness-probe:1.0.0
controller.cosiC ontroller	Image name of COSI Controller.	Yes	gcr.io/k8s-staging- sig-storage/ objectstorage- controller:latest
sidecar.cosiSide car	Container monitoring interface image.	Yes	gcr.io/k8s-staging- sig-storage/ objectstorage- sidecar:latest
images.imageP ullPolicy.huaw eiCosiDriverIm agePullPolicy	Pull policy of Huawei COSI driver image.	Yes	IfNotPresent
images.imageP ullPolicy.huaw eiCosiLiveness ProbeImagePul lPolicy	Pull policy of Huawei COSI driver health check image.	Yes	IfNotPresent
images.imageP ullPolicy.cosiCo ntrollerImageP ullPolicy	Pull policy of the COSI Controller image.	Yes	IfNotPresent
images.imageP ullPolicy.cosiSi decarImagePul lPolicy	Pull policy of the COSI sidecar image.	Yes	IfNotPresent

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.cosiD river.requests.c pu	Minimum CPU resource of the cosiDriver container.	Yes	50m
container.cosiD river.requests. memory	Minimum memory resource of the cosiDriver container.	Yes	128Mi

Parameter	Description	Mandatory	Default Value
container.cosiD river.limits.cpu	Maximum CPU resource of the cosiDriver container.	Yes	100m
container.cosiD river.limits.me mory	Maximum memory resource of the cosiDriver container.	Yes	256Mi
container.cosiLi venessProbe.re quests.cpu	Minimum CPU resource of the cosiLivenessProbe container.	Yes	10m
container.cosiLi venessProbe.re quests.memor y	Minimum memory resource of the cosiLivenessProbe container.	Yes	128Mi
container.cosiLi venessProbe.li mits.cpu	venessProbe.li resource of the		100m
container.cosiLi venessProbe.li mits.memory	nessProbe.li resource of the		128Mi
container.cosiSi decar.requests. cpu	ar.requests. resource of the		50m
container.cosiSi decar.requests. memory	ecar.requests. resource of the		128Mi
container.cosiSi decar.limits.cp u	decar.limits.cp resource of the		100m
container.cosiSi decar.limits.me mory	ecar.limits.me resource of the		512Mi
container.cosiC ontroller.reque sts.cpu	roller.reque resource of the		50m
container.cosiC ontroller.reque sts.memory cosiController container.		Yes	128Mi

Parameter	Description	Mandatory	Default Value
container.cosiC ontroller.limits. cpu	Maximum CPU resource of the cosiController container.	Yes	100m
container.cosiC ontroller.limits. memory	Maximum memory resource of the cosiController container.	Yes	512Mi

----End

Installation and Deployment

- 4.1 Installing the Software
- 4.2 Uninstalling the Software
- 4.3 Updating the Software
- 4.4 Rolling Back the Update
- 4.5 Upgrading the Software
- 4.6 Rolling Back the Upgrade

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.

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
```

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 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

4.4 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-1.0.0 1.0.0 Install complete

2 Fri Aug 30 17:07:33 2024 deployed cosi-1.0.0 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

4.5 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

4.6 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-1.0.0 1.0.0 Install complete
2 Fri Aug 30 17:07:33 2024 deployed cosi-1.0.0 1.0.0 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
- 5.2 Bucket Access Management

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

Paramet er	Description	Mandator y	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 .namespa ce	Namespace of the Secret object.	Yes	-	The name must consist of lowercase letters, digits, and hyphens (-), for example, my-name and 123-abc.
stringDat a.accessK ey	AK of the corresponding account on the storage side.	Yes	-	-
stringDat a.secretKe y	SK of the corresponding account on the storage side.	Yes	-	-
stringDat a.endpoin t	Endpoint of the service plane on the storage side.	Yes	-	The value can be a domain name or an IP address + port number. Example: https://xx.xx.xx.5443. The port number must be set to 5443.
data.root CA	Root certificate information, which is used to verify the certificate of the storage server.	No	1	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

Paramet er	Description	Mandato ry	Default Value	Remarks
metadat a.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.
driverNa me	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.
deletionP olicy	Bucket resource reclamation policy on the storage side. The value can be: Delete Retain	Yes	-	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.
paramet ers.accou ntSecret Name	Name of the Secret object.	Yes	-	-

Paramet er	Description	Mandato ry	Default Value	Remarks
paramet ers.accou ntSecret Namesp ace	Namespace of the Secret object.	Yes	-	-
paramet ers.bucke tACL	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 users, including anonymous users, have the read and write permissions.
				service grantees have the read permission.
paramet ers.bucke tLocation	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

Paramet er	Description	Mandator y	Defau lt 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 .namespa ce	Namespace of the user-defined BucketClaim object.	Yes	-	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.buck etClassNa me	Name of a BucketClass object.	Yes	-	-

Paramet er	Description	Mandator y	Defau lt Value	Remarks
spec.prot ocols	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

Paramet er	Description	Mandator y	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 .namespa ce	Namespace of the Secret object.	Yes	-	The name must consist of lowercase letters, digits, and hyphens (-), for example, my-name and 123-abc.
stringDat a.accessK ey	AK of the corresponding account on the storage side.	Yes	-	-
stringDat a.secretKe y	SK of the corresponding account on the storage side.	Yes	-	-

Paramet er	Description	Mandator y	Default Value	Remarks
stringDat a.endpoin t	Endpoint of the service plane on the storage side.	Yes	-	The value can be a domain name or an IP address + port number. Example: https://xx.xx.xx.5443. The port number must be set to 5443.
data.root CA	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 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

deletionPolicy: Delete

driverName: cosi.huawei.com

parameters: accountSecretName: sample-account-service-secret

accountSecretNamespace: huawei-cosi

bucketACL: <bucket-acl> bucketLocation: <bucket-location>

Table 5-5 BucketClass configuration parameters

Paramet er	Description	Mandato ry	Default Value	Remarks
metadat a.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.
driverNa me	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.
deletionP olicy	Bucket resource reclamation policy on the storage side. The value can be: Delete Retain	Yes	-	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.
paramet ers.accou ntSecret Name	Name of the Secret object.	Yes	-	-

Paramet er	Description	Mandato ry	Default Value	Remarks
paramet ers.accou ntSecret Namesp ace	Namespace of the Secret object.	Yes	-	-
paramet ers.bucke tACL	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.
paramet ers.bucke tLocation	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

Paramete r	Description	Mandato ry	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.buck etClaim	Name of a BucketClaim object.	Yes	-	Set this parameter to {}.
spec.drive rName	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.

Paramete r	Description	Mandato ry	Default Value	Remarks
spec.buck etClassNa me	Name of a BucketClass object.	Yes	-	-
spec.existi ngBucketI D	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</storage-existing-></account-></account-service-secret->
spec.delet ionPolicy	Bucket resource reclamation policy on the storage side. The value can be: Delete Retain	Yes	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.prot ocols	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

Paramet er	Description	Mandat ory	Default Value	Remarks
metadat a.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.

Paramet er	Description	Mandat ory	Default Value	Remarks
metadat a.names pace	Namespace of the user-defined static BucketClaim object.	Yes	-	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.buc ketClass Name	BucketClass name.	Yes	-	-
spec.exist ingBucke tName	Name of a static Bucket.	Yes	-	NOTICE When creating multiple BucketClaim objects, do not bind them to the same static Bucket object.
spec.prot ocols	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

Paramete r	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. namespac e	Namespace of the Secret object.	Yes	-	The name must consist of lowercase letters, digits, and hyphens (-), for example, my-name and 123-abc.
stringDat a.accessK ey	AK of the corresponding account on the storage side.	Yes	-	-
stringDat a.secretKe y	SK of the corresponding account on the storage side.	Yes	-	-
stringDat a.endpoin t	Endpoint of the management plane on the storage side.	Yes	-	The value can be a domain name or an IP address + port number. Example: https://xx.xx.xx.y443. The port number must be set to 9443.

Paramete r	Description	Mandatory	Default Value	Remarks
data.root CA	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

accountSecretName: sample-account-management-secret

accountSecretNamespace: huawei-cosi

bucketPolicyModel: rw

parameters:

Table 5-9 BucketAccessClass configuration parameters

Paramete r	Description	Mandator y	Defaul t Value	Remarks
metadata. name	User-defined name of a BucketAccessCl ass 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.
driverNa me	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.
authentic ationType	Authorization type. The value can be: • Key	Yes	-	-
paramete rs.account SecretNa me	Name of the Secret object.	Yes	-	-
paramete rs.account SecretNa mespace	Namespace of the Secret object.	Yes	-	-

Paramete r	Description	Mandator y	Defaul t Value	Remarks
paramete rs.bucketP olicyMod el	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:ListMultipartUpload-Parts, s3:GetObjectVersionAcl, s3:ListBucketVersions, s3:ListBucket, s3:ListBucketMultipartUploads rw: bucket policy in read/write mode, including the following s3 operations: s3:GetObject, s3:GetObject, s3:GetObjectVersion, s3:ListMultipartUpload-Parts, s3:GetObjectVersionAcl, s3:ListBucket, s3:ListBucket, s3:ListBucket, s3:ListBucket, s3:ListBucket, s3:ListBucket, s3:ListBucket, s3:ListBucket, s3:ListBucket, s3:ListBucketMultipartUpload, s3:PutObjectAcl, s3:DeleteObjectVersionAcl, s3:PutObject, s3:DeleteObject, s3:

- **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

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

Paramet er	Description	Mandato ry	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 .namespa ce	Namespace where the BucketAccess object is located.	Yes	-	The name must consist of lowercase letters, digits, and hyphens (-), for example, my-name and 123-abc.
spec.buck etClaimN ame	Name of the BucketClaim object to which access needs to be granted.	Yes	-	-
spec.buck etAccessC lassName	Name of the BucketAccessClas s object that needs to be used.	Yes	-	-

Paramet er	Description	Mandato ry	Default Value	Remarks
spec.cred entialsSec retName	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.prot ocol	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:

eyJtZXRhZGF0YSl6eyJuYW1lljoiYmMtZGJjZWJlN2ltMDMzMy00MTYwLThkMTYtMGMyNzcyZmQyMTk5liwiY3JlYXRpb25UaW1lc3RhbXAiOm51bGx9LCJzcGVjljp7lmJ1Y2tldE5hbWUiOiJzYW1wbGUtYnVja2V0LWNsYXNzMDg4OGNiOWYtYzMyYi00YjRiLWEwYmItYjA1MzNlNDg0ZjQyliwiYXV0aGVudGljYXRpb25UeXBlljoiS2V5liwic2VjcmV0UzMiOnsiZW5kcG9pbnQiOiJodHRwczovL3gueHgueHh4Lnh4eDo1NDQzliwicmVnaW9uljoiliwiYWNjZXNzS2V5SUQiOilxMjM0NTY3OTg5liwiYWNjZXNzU2VjcmV0S2V5ljoiMTlzNDU2Nzk4OSJ9LCJzZWNyZXRBenVyZSl6bnVsbCwicHJvdG9jb2xzljpblnMzll19fQ==

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

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

echo

type: Opaque

"eyJtZXRhZGF0YSl6eyJuYW1lljoiYmMtZGJjZWJlN2ItMDMzMy00MTYwLThkMTYtMGMyNzcyZmQyMTk5liwiY3 JlYXRpb25UaW1lc3RhbXAiOm51bGx9LCJzcGVjljp7lmJ1Y2tldE5hbWUiOiJzYW1wbGUtYnVja2V0LWNsYXNzMD g4OGNiOWYtYzMyYi00YjRiLWEwYmItYjA1MzNlNDg0ZjQyliwiYXV0aGVudGljYXRpb25UeXBlljoiS2V5liwic2Vjc mV0UzMiOnsiZW5kcG9pbnQiOiJodHRwczovL3gueHgueHh4Lnh4eDo1NDQzliwicmVnaW9uljoiliwiYWNjZXNz S2V5SUQiOilxMjM0NTY3OTg5liwiYWNjZXNzU2VjcmV0S2V5IjoiMTlzNDU2Nzk4OSJ9LCJzZWNyZXRBenVyZSI 6bnVsbCwicHJvdG9jb2xzljpblnMzll19fQ==" | base64 -d

```
\label{thm:protocols} $$ {\rm metadata":{\rm mame":"bc-dbcebe7b-0333-4160-8d16-0c2772fd2199","creationTimestamp":null},"spec": $$ {\rm bucketName":"sample-bucket-class0888cb9f-c32b-4b4b-a0bb-b0533e484f42","authenticationType":"Key","secretS3":{\rm endpoint":"https://x.xx.xxx.xxx:5443","region":"","accessKeyID":"1234567989","accessSecretKey":"1234567989"},"secretAzure":null,"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.

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

6 Security Hardening

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

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**.

recording modules	ecording 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				

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

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.

7 FAQs

- 7.1 How Do I Download a Container Image to the Local Host?
- 7.2 How Do I View Huawei COSI Logs?
- 7.3 How Do I Obtain the COSI Version?
- 7.4 COSI Sidecar and Controller Community Issues

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.qcr.io/siq-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 huaweicosi-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 huaweicosi-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 cosicontroller, 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: 1.0.0
liveness-probe: 1.0.0
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 opensource community are in the alpha phase and may have some issues. For details about the issues, see the community issues.

Links for Community Issues

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

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