eSDK Huawei Storage Kubernetes CSM Plugins V2.1.0

User Guide

Issue 01

Date 2024-12-25





Copyright © Huawei Technologies Co., Ltd. 2025. 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

Intended Audience

This document is intended for:

- Technical support engineers
- O&M engineers
- Engineers with basic knowledge of storage, Kubernetes, and CSI

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	2024-12-25	This issue is the first official release.

Contents

About This Document	iii
1 Product Description	1
2 Compatibility and Features	2
2.1 Kubernetes Compatibility	2
2.2 Compatibility with Huawei Storage	2
2.3 csm-prometheus Feature Description	3
2.4 csm-storage Feature Description	5
2.5 Constraints	6
3 Installation Preparations	8
3.1 Obtaining Tools and Software Packages	8
3.2 Uploading Images to the Image Repository	g
3.3 Installing Helm	
3.4 Preparing the Configuration File	11
4 Installation and Deployment	20
4.1 Installation and Deployment Using Helm	20
4.1.1 Installing the Software	20
4.1.2 Installing the Software on CCE Agile	21
4.1.2.1 Creating a Helm Installation Package	21
4.1.2.2 Installing Huawei CSM	21
4.1.3 Upgrading the Software	23
4.1.4 Rolling Back the Software	
4.1.5 Uninstalling the Software	
4.1.6 Uninstalling the Software from CCE Agile	
4.2 Manual Installation and Deployment	
4.2.1 Manually Installing the Software	
4.2.2 Manually Updating the Software	
4.2.3 Manually Uninstalling the Software	29
5 Configuring Prometheus	
5.1 Installing Prometheus	
5.2 Configuring the Prometheus Service	
5.3 Configuring the Prometheus Dashboard	37

6 Configuring Grafana	38
6.1 Installing Grafana	
6.2 Using Grafana	
7 Common O&M Guide	44
7.1 Collecting Logs	44
7.1.1 Performing Check Before Collection	44
7.1.2 Collecting CSM Logs Using oceanctl	45
7.2 Viewing Version Information	45
8 Appendix	47
8.1 Accessing Kubernetes As a Non-root User	
8.2 Configuring the HTTPS Service for csm-prometheus	48
8.2.1 Configuring the HTTPS Service Certificate of csm-prometheus	
8.2.2 Deleting the HTTPS Service Certificate of csm-prometheus	49
8.2.3 Updating the HTTPS Service Certificate of csm-prometheus	
8.3 Managing Storage Certificates	50
8.4 Permission Matrix	
9 FAQs	52
9.1 Pod in OOMKilled State	
9.2 Manually Adjusting the Number of Concurrent Data Scraping Requests	
9.3 Residual Pod Labels on the Storage Side	

Product Description

Container Storage Monitor (CSM) is a tool used for visual display of Huawei storage resources and Kubernetes resources in Kubernetes container scenarios. This tool can notify storage of the relationship between a PV/Pod and a LUN/file system so that the relationship can be displayed on the storage for storage administrators to view. It can also upload the performance, capacity, IOPS, and other data of a LUN/file system to a third-party network management system for application administrators to view. In this way, O&M availability in container scenarios can be improved.

2 Compatibility and Features

- 2.1 Kubernetes Compatibility
- 2.2 Compatibility with Huawei Storage
- 2.3 csm-prometheus Feature Description
- 2.4 csm-storage Feature Description
- 2.5 Constraints

2.1 Kubernetes Compatibility

Table 2-1 Supported container management platforms

Container Management Platform	Version
Kubernetes	1.16 to 1.27, 1.31
CCE Agile	22.3.2

2.2 Compatibility with Huawei Storage

□ NOTE

- When CSM is used to interconnect with storage, only LUNs/file systems provisioned by CSI can be displayed. Dtrees are not supported.
- Only OceanStor 6.1.7 and later and OceanStor Dorado 6.1.7 and later support the CSM-Storage label function.

Table 2-2 CSM-Storage compatibility

Storage Version	Huawei CSI Version
OceanStor 6.1.7/6.1.8/ V700R001C00	4.6.0
OceanStor Dorado 6.1.7/6.1.8/ V700R001C00	

Table 2-3 CSM-Prometheus compatibility

Storage Version	Prometheus Version	Grafana Version	Huawei CSI Version
OceanStor 6.1.3/6.1.5/6.1.6/6.1.7/6 .1.8/V700R001C00	2.25.0 - 2.43.0	Grafana 7.0.4	4.6.0
OceanStor Dorado 6.1.0 ¹ /6.1.2/6.1.3/6.1.5/ 6.1.6/6.1.7/6.1.8/ V700R001C00			

• Note 1: If OceanStor Dorado 6.1.0 is used, the display of Prometheus performance monitoring metrics may be broken or discontinuous as the number of concurrent requests increases.

2.3 csm-prometheus Feature Description

csm-prometheus collects storage monitoring data and exposes the data to the Prometheus platform for collection. The following table lists the supported monitoring metrics.

Table 2-4 Object metrics supported by monitored objects of centralized storage

Object Type	Object Metric
Storage	Basic informationMonitoring statusRunning status
Controller	CPU usage Memory usage
	Health statusRunning status

Object Type	Object Metric
Storage pool	Total capacity
	Remaining capacity
	Used capacity
	Capacity usage
LUN	Total capacity
	Capacity usage
File system	Total capacity
	Capacity usage
PV	Total capacity
	Capacity usage

Table 2-5 Performance metrics supported by monitored objects of centralized storage

Object Type	Performance Metric	
Controller	• 21: bandwidth (MB/s)	
	23: read bandwidth (MB/s)	
	• 26: write bandwidth (MB/s)	
	• 22: IOPS	
	• 25: read IOPS	
	• 28: write IOPS	
	• 370 : average I/O response time (μs)	
Storage pool	• 21: bandwidth (MB/s)	
	• 22: IOPS	
	• 370 : average I/O response time (μs)	
LUN	• 21: bandwidth (MB/s)	
	• 22: IOPS	
	• 370 : average I/O response time (μs)	
File system	• 182: OPS	
	• 524 : average read OPS response time (μs)	
	• 525 : average write OPS response time (μs)	

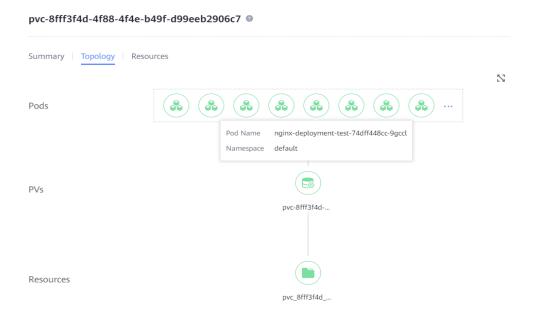
Object Type	Performance Metric
PV	LUN bandwidth: bandwidth (MB/s) of a PV of the LUN type on the storage
	• LUN IOPS: IOPS of a PV of the LUN type on the storage
	 Average LUN I/O response time: average I/O response time (μs) of a PV of the LUN type on the storage
	File system OPS: OPS of a PV of the file system type on the storage
	Average file system read OPS response time: average read OPS response time (μ s) of a PV of the file system type on the storage
	\bullet Average file system write OPS response time: average write OPS response time (µs) of a PV of the file system type on the storage

2.4 csm-storage Feature Description

NOTICE

The default memory quota of the CSM topology service is 512 Mi. The memory usage of the topology service increases linearly with the number of PVs and Pods in the cluster. If there are a large number of resources in the cluster, you can manually modify the memory quota of the topology service to ensure that the csm-storage feature can be used properly. For configuration details, see **Table 3-7**.

csm-storage reports the PVs and the topology relationships with associated Pods, file systems, and LUNs to the storage system. The following figure shows the resource display effect on the storage GUI.



2.5 Constraints

Performance Constraints

Table 2-6 CSM specifications management

Specifications Management		Recommended Value
Storage management	Number of connected storage devices	5
	Number of CSMs interconnected with a single storage device	≤ 3
Monitoring item management	Maximum number of monitoring items supported by a storage device	15000
	Maximum number of monitoring items of all storage devices	40000
Scraping interval management	Object data scraping interval (for metrics in Table 2-4)	300s
	Performance data scraping interval (for metrics in Table 2-5)	300s

NOTICE

- If the data scraping interval is too small, the storage backend pressure will increase, adversely impacting the running of the storage backend.
- It is recommended that the object data scraping interval be different from the performance data scraping interval to prevent query failures caused by heavy storage backend pressure.
- A single storage resource (LUN or file system) does not support repeated topology creation.
- Topology relationships cannot be reported for resources with the same name of different vStores on the same storage device.

Table 2-7 Concurrent data scraping requests recommended by CSM

Total Number of Storage File Systems and LUNs	Recommended Number of Concurrent Requests
< 2000	20
2000-5000	10
> 5000	5

NOTICE

The default number of concurrent CSM data scraping requests is 20. When the number of storage resources increases, you are advised to configure the number of concurrent data scraping requests based on the recommended values to reduce the storage query load. For details about how to configure the number of concurrent requests, see 9.2 Manually Adjusting the Number of Concurrent Data Scraping Requests.

3 Installation Preparations

- 3.1 Obtaining Tools and Software Packages
- 3.2 Uploading Images to the Image Repository
- 3.3 Installing Helm
- 3.4 Preparing the Configuration File

3.1 Obtaining Tools and Software Packages

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	You can visit the chiark homepage to download the PuTTY software.
	installation.	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.	You can visit the WinSCP homepage to download the WinSCP software.
	It is used to transfer files between Windows and Linux.	

Software Packages

Before deploying services, you need to prepare the CSM software installation packages listed in **Table 3-2**.

Table 3-2 Required software packages

Package	Description	How to Obtain
eSDK_Huawei_Storage_C SM_V2.1.0_X86_64.zip eSDK_Huawei_Storage_C SM_V2.1.0_ARM_64.zip	CSM software installation package.	https://github.com/ Huawei/csm/releases

□ NOTE

To prevent a software package from being maliciously tampered with during transmission or storage, download the corresponding digital signature file together with the software package for integrity verification.

After the software package is downloaded from Huawei Support website, verify its PGP digital signature by referring to *OpenPGP Signature Verification Guide*. If the verification fails, do not use the software package and contact Huawei technical support engineers.

Before a software package is used in installation or upgrade, its digital signature also needs to be verified according to *OpenPGP Signature Verification Guide* to ensure that the software package is not tampered with.

- For carrier users, visit https://support.huawei.com/carrier/digitalSignatureAction.
- For enterprise users, visit https://support.huawei.com/enterprise/en/tool/pgp-verify-TL1000000054.

3.2 Uploading Images to the Image Repository

Prerequisites

- The required CSM software package is available. For details about how to obtain it, see **3.1 Obtaining Tools and Software Packages**.
- An image repository has been prepared and can communicate with worker nodes. In addition, you have obtained the IP address, account, and password of the image repository.
- A project has been created in the image repository.
- 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** Use WinSCP to upload the software package to the **/opt** directory.
- **Step 3** Run the **unzip /opt/** *Software package name* command to decompress the software package.

Software package name indicates the CSM software package name. **eSDK_Huawei_Storage_CSM_V2.1.0_X86_64.zip** is used as an example. # unzip /opt/eSDK_Huawei_Storage_CSM_V2.1.0_X86_64.zip.zip -d /opt/huawei-csm

- **Step 4** Run the **docker login** *<IP* address of the image repository> command and enter the account and password to log in to the image repository.
- **Step 5** Run the following command to upload the CSM image.

chmod +x /opt/huawei-csm/helm/huawei-csm/upload-image.sh; ./opt/huawei-csm/helm/huawei-csm/upload-image.sh --imageRepo </mage repository project name>

chmod +x /opt/huawei-csm/helm/huawei-csm/upload-image.sh;sh /opt/huawei-csm/helm/huawei-csm/ upload-image.sh --imageRepo *<Image repository project name>*

----End

NOTICE

- The **upload-image.sh** script is interpreted and executed using the Bash (Bourne-Again Shell). Before executing the script, ensure that the current system supports the Unix shell of the Bash type.
- For details about how to import and upload images to the CCE Agile platform, see the user manual of the platform.

3.3 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.

3.4 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** Use WinSCP to upload the software package to the **/opt** directory.
- **Step 3** Run the **unzip /opt/** *Software package name* command to decompress the software package.
 - Software package name indicates the software package name. **eSDK_Huawei_Storage_CSM_V2.1.0_X86_64.zip** is used as an example. # unzip /opt/eSDK_Huawei_Storage_CSM_V2.1.0_X86_64.zip -d /opt/huawei-csm
- **Step 4** Run the **cd /opt/huawei-csm/helm/huawei-csm** command to go to the Helm working directory.
- Step 5 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 3-3, Table 3-4, Table 3-5, and Table 3-6 describe related parameters.

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

Table 3-3 global configuration items

Parameter	Description	Mandatory	Default Value
replicaCount	Number of Pod copies corresponding to the Deployment deployed using CSM. When the number of Pod copies is greater than 1, the system automatically enables the leaderElection function.	No	 It is recommended that the value be less than or equal to 2. If the number of Pod copies is 1: Ensure that the number of CPU cores is 4 or above and the memory is 4 GB or above in Kubernetes. If the number of Pod copies is 2: Ensure that the number of CPU cores is 8 or above and the memory is 8 GB or above in Kubernetes. NOTICE If the single-copy mode is used during deployment and you want to change it to the multi-copy mode, update the replicaCount parameter by following the instructions in 4.1.3 Upgrading the Software. Do not use other methods to change the number of Pod copies corresponding to the Deployment deployed using CSM. If you use other methods, for example, directly editing the Spec.replicas parameter in a Deployment deployed using CSM, the multicopy function will be abnormal.
imageRepo	Image repository name.	Yes	None. The value must be the same as the image repository name in Step 5 .
logging.mod ule	Log mode.	Yes	file The value can be file or console .

Parameter	Description	Mandatory	Default Value
logging.level	Log level.	Yes	info The value can be debug , info , warning , or error .
logging.fileS ize	Log file size.	Yes	20 MB
logging.max Backups	Maximum number of backup logs.	Yes	9
leaderElecti on.leaseDur ation	Leader duration. This parameter takes effect only in the multi-copy mode.	No	8s
leaderElecti on.renewDe adline	Time for the leader to be re-elected. This parameter takes effect only in the multi-copy mode.	No	6s
leaderElecti on.retryPeri od	Leader election retry time. This parameter takes effect only in the multi-copy mode.	No	2s
balancedDe ploy	If this parameter is set to true , Pods of different services will be scheduled to different nodes during CSM installation.	No	true If the schedule of balancedDeploy conflicts with that of nodeSelector, the schedule of balancedDeploy does not take effect.

The **features** configuration items are used to enable and disable features.

Table 3-4 features configuration items

Parameter	Description	Mandatory	Default Value	Remarks
prometheu sCollector.e nabled	Whether to enable the Prometheus collection service.	Yes	true	-

Parameter	Description	Mandatory	Default Value	Remarks
prometheu sCollector.n odePort	Prometheus nodePort.	No	30074	The default value is 30074 . This port is a host port. If a conflict occurs, change the value.
prometheu sCollector.c siDriverNa me	Registered CSI driver name.	Yes	csi.huaw ei.com	 Use the default value. For the CCE Agile platform, modify this field. For example, csi.oceanstor.co m.
prometheu sCollector.p rometheus CollectorSS L.enabled	Whether to enable the HTTPS service. After it is enabled, the Prometheus plug-in will provide the HTTPS service.	Yes	true	The certificate path must be specified after the service is enabled. For details about how to configure the certificate, see Configuring the HTTPS Service.
prometheu sCollector.p rometheus CollectorSS L.certPath	HTTPS certificate path after the Prometheus plug-in provides the HTTPS service.	This parameter is mandatory when prometheus Collector.pr ometheusC ollectorSSL. enabled is set to true.	-	The path must be a relative path in the Helm working path.
prometheu sCollector.p rometheus CollectorSS L.keyPath	HTTPS key path after the Prometheus plug-in provides the HTTPS service.	This parameter is mandatory when prometheus Collector.pr ometheus Collector SSL. enabled is set to true.	-	The path must be a relative path in the Helm working path.

Parameter	Description	Mandatory	Default Value	Remarks
prometheu sCollector.n odeSelector	Node selector of csm-prometheus-service. After this parameter is set, csm-prometheus-service will be scheduled only to a node with the label.	No	-	For details about the node selector, see Assign Pods to Nodes.
storageTop o.enabled	Whether to enable the storage topology display service.	Yes	true	Only OceanStor 6.1.7/OceanStor Dorado 6.1.7 and later versions are supported.
storageTop o.rtRetryMa xDelay	Maximum retry delay of a topo resource synchronization task.	No	5m	You are advised to use the default value. The minimum retry delay is 5 seconds.
storageTop o.pvRetryM axDelay	Maximum retry delay of PV resource creation and topo resource deletion tasks	No	1m	You are advised to use the default value. The minimum retry delay is 5 seconds.
storageTop o.podRetry MaxDelay	Maximum retry delay of the topo resource label update task for Pod resources.	No	1m	You are advised to use the default value. The minimum retry delay is 5 seconds.
storageTop o.resyncPeri od	Interval for refreshing topo resources.	No	15m	You are advised to use the default setting. The minimum refresh interval is 5 minutes.
storageTop o.nodeSelec tor	Node selector of csm-storage-service. After this parameter is set, csm-storage-service will be scheduled only to a node with the label.	No	-	For details about the node selector, see Assign Pods to Nodes.

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

Table 3-5 images configuration items

Parameter	Description	Mandatory	Default Value
prometheusC ollector	Image of the Prometheus storage data collection plug- in.	Yes	csm-prometheus- collector:2.1.0
topoService	Resource topology service image.	Yes	csm-topo-service:2.1.0
containerMo nitorInterfac e	Container monitoring interface image.	Yes	csm-cmi:2.1.0
livenessProbe	Liveness probe service image.	Yes	csm-liveness- probe:2.1.0

The **cluster** configuration item mainly describes the Kubernetes cluster information.

Table 3-6 cluster configuration item

Parameter	Description	Mandatory	Default Value
cluster.name	Custom cluster name.	Yes	kubernetes

The **containerResoucesSet** configuration items are used to configure the container resources of each Pod.

Table 3-7 containerResoucesSet configuration items

Parameter	Description	Mandator y	Default Value
prometheusService.livenes sProbe.requests.memory	Minimum memory resource of the livenessProbe container in a Prometheus Pod.	Yes	128Mi

Parameter	Description	Mandator y	Default Value
prometheusService.livenes sProbe.limits.cpu	Maximum CPU resource of the livenessProbe container in a Prometheus Pod.	Yes	100m
prometheusService.livenes sProbe.limits.memory	Maximum memory resource of the livenessProbe container in a Prometheus Pod.	Yes	128Mi
prometheusService.prome theusCollector.requests.cp u	Minimum CPU resource of the prometheusCollector container in a Prometheus Pod.	Yes	50m
prometheusService.prome theusCollector.requests.m emory	Minimum memory resource of the prometheusCollector container in a Prometheus Pod.	Yes	128Mi
prometheusService.prome theusCollector.limits.cpu	Maximum CPU resource of the prometheusCollector container in a Prometheus Pod.	Yes	300m
prometheusService.prome theusCollector.limits.mem ory	Maximum memory resource of the prometheusCollector container in a Prometheus Pod.	Yes	512Mi
prometheusService.cmiCo ntroller.requests.cpu	Minimum CPU resource of the cmiController container in a Prometheus Pod.	Yes	50m
prometheusService.cmiCo ntroller.requests.memory	Minimum memory resource of the cmiController container in a Prometheus Pod.	Yes	128Mi
prometheusService.cmiCo ntroller.limits.cpu	Maximum CPU resource of the cmiController container in a Prometheus Pod.	Yes	300m

Parameter	Description	Mandator y	Default Value	
prometheusService.cmiCo ntroller.limits.memory	Maximum memory resource of the cmiController container in a Prometheus Pod.	Yes	512Mi	
storageService.livenessPro be.requests.cpu	Minimum CPU resource of the livenessProbe container in a storage Pod.	Yes	10m	
storageService.livenessPro be.requests.memory	Minimum memory resource of the livenessProbe container in a storage Pod.	Yes	128Mi	
storageService.livenessPro be.limits.cpu	Maximum CPU resource of the livenessProbe container in a storage Pod.	Yes	100m	
storageService.livenessPro be.limits.memory	Maximum memory resource of the livenessProbe container in a storage Pod.	Yes	128Mi	
storageService.cmiControl ler.requests.cpu	Minimum CPU resource of the cmiController container in a storage Pod.	Yes	50m	
storageService.cmiControl ler.requests.memory	Minimum memory resource of the cmiController container in a storage Pod.	Yes	128Mi	
storageService.cmiControl ler.limits.cpu	Maximum CPU resource of the cmiController container in a storage Pod.	Yes	300m	
storageService.cmiControl ler.limits.memory	Maximum memory resource of the cmiController container in a storage Pod.	Yes	512Mi	
storageService.topoServic e.requests.cpu	Minimum CPU resource of the topoService container in a storage Pod.	Yes	50m	

Parameter	Description	Mandator y	Default Value
storageService.topoServic e.requests.memory	Minimum memory resource of the topoService container in a storage Pod.	Yes	128Mi
storageService.topoServic e.limits.cpu	Maximum CPU resource of the topoService container in a storage Pod.	Yes	300m
storageService.topoServic e.limits.memory	Maximum memory resource of the topoService container in a storage Pod.	Yes	512Mi

----End

4 Installation and Deployment

- 4.1 Installation and Deployment Using Helm
- 4.2 Manual Installation and Deployment

4.1 Installation and Deployment Using Helm

4.1.1 Installing the Software

Prerequisites

- Helm 3 has been installed on the master node.
- The values.yaml file has been configured. For details, see 3.4 Preparing the Configuration File.

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-csm/helm/huawei-csm** command to go to the Helm working directory.
- **Step 3** Run the **helm install huawei-csm**./ -n huawei-csm --create-namespace command to install CSM services.

helm install huawei-csm ./ -n huawei-csm --create-namespace NAME: huawei-csm LAST DEPLOYED: Tue Aug 8 23:11:18 2023 NAMESPACE: huawei-csm STATUS: deployed REVISION: 1 TEST SUITE: None

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

kubectl get pod -n huawei-csm NAME READY STATUS RESTARTS AGE

csm-prometheus-service-86c795d68-b5xjg	2/2	Running	0	5s
csm-storage-service-85485fd75f-9wg8m	2/2	Running	0	4s

----End

4.1.2 Installing the Software on CCE Agile

4.1.2.1 Creating a Helm Installation Package

To install Huawei CSM on the CCE Agile platform, you need to create a Helm installation package. This section describes how to create a Helm installation package.

Prerequisites

- Helm 3 has been installed on a node server.
- The **values.yaml** file required for installing CSM has been prepared. For details, see **3.4 Preparing the Configuration File**.

Procedure

- **Step 1** Use a remote access tool, such as PuTTY, to log in to any node where Helm is deployed through the management IP address.
- **Step 2** Run the **cd /opt/huawei-csm/helm/** command to go to the Helm working directory.
- **Step 3** Run the **helm package huawei-csm/ -d ./** command to create a Helm installation package. This command will generate the installation package to the current path.

```
# helm package huawei-csm/ -d ./
Successfully packaged chart and saved it to: huawei-csm-2.1.0.tgz
```

----End

4.1.2.2 Installing Huawei CSM

Prerequisites

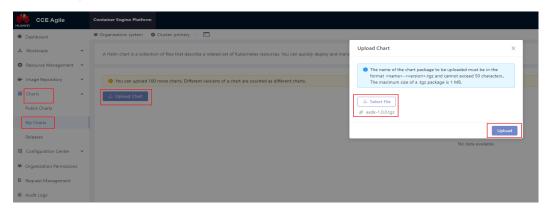
A Huawei CSM Helm installation package has been created. For details, see **4.1.2.1 Creating a Helm Installation Package**.

Procedure

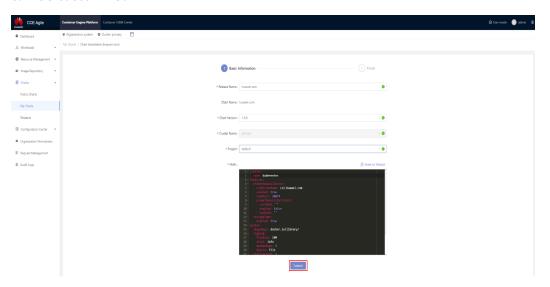
- **Step 1** Use a remote access tool, such as PuTTY, to log in to any master node where the CCE Agile platform is deployed through the management IP address.
- **Step 2** Run the **kubectl create namespace** *huawei-csm* command to create a namespace for deploying Huawei CSM. *huawei-csm* indicates the custom namespace.

 # kubectl create namespace huawei-csm
- **Step 3** Export the Helm installation package. For details, see **4.1.2.1 Creating a Helm Installation Package**.

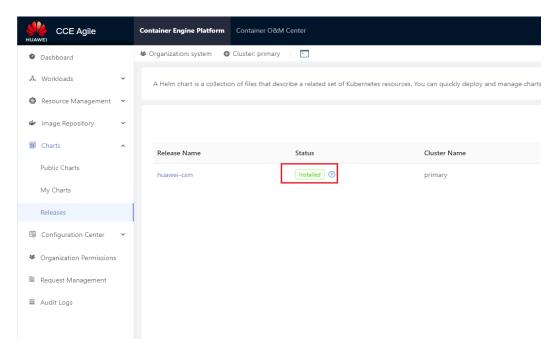
Step 4 On the home page, choose **Charts** > **My Charts** > **Upload Chart**. The **Upload Chart** dialog box is displayed. Import the exported Helm installation package to the CCE Agile platform.



Step 5 After the installation package is uploaded, choose **Charts > My Charts**. On the **My Charts** page that is displayed, choose **Install > Submit**. The chart release name can be customized.



Step 6 On the home page, choose **Charts** > **Releases** and select the project specified during installation (for example, **default** in the following figure). After the installation is successful, **Installed** is displayed in the **Status** column.



----End

4.1.3 Upgrading the Software

Scenario

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

Prerequisites

CSM 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 **3.2 Uploading** Images to the Image Repository.
- Step 3 (Optional) Run the kubectl delete validatingWebhookConfiguration toposervice.xuanwu.huawei.io command to delete the webhook resource. If the toposervice.xuanwu.huawei.io resource does not exist, skip this step.
 - # kubectl delete validatingWebhookConfiguration topo-service.xuanwu.huawei.io
- **Step 4** Run the **cd /opt/huawei-csm/helm/huawei-csm/** command to go to the Helm working directory.

- **Step 5** Run the **kubectl apply -f crds/** command to update crd resources.
 - # kubectl apply -f crds/ customresourcedefinition.apiextensions.k8s.io/resourcetopologies.xuanwu.huawei.io configured
- **Step 6** Run the **helm get values huawei-csm -n huawei-csm -a > update-value.yaml** command to obtain the original service configuration file.
- **Step 7** 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 **3.4 Preparing the Configuration File**.
- Step 8 Run the helm upgrade huawei-csm ./ -n huawei-csm -f ./values.yaml -f update-value.yaml --wait --timeout 2m command to upgrade CSM services. If Release "huawei-csm" has been upgraded is displayed in the command output, the CSM services are successfully upgraded.

helm upgrade huawei-csm ./ -n huawei-csm -f ./values.yaml -f update-value.yaml --wait --timeout 2m Release "huawei-csm" has been upgraded. Happy Helming!

NAME: huawei-csm

LAST DEPLOYED: Wed Aug 9 04:19:10 2023

NAMESPACE: huawei-csm

STATUS: deployed

REVISION: 3

TEST SUITE: None

Step 9 (Optional) Run the kubectl get rt | grep topo- | awk '{print "kubectl delete rt "\$1}' | sh command to delete the residual topo resources of the source version. If there is no residual topo resource of the source version in the cluster, skip this step.

----End

4.1.4 Rolling Back the Software

Prerequisites

- CSM has been deployed using Helm 3.
- CSM 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 **cd /opt/huawei-csm/helm/huawei-csm/** command to go to the Helm working directory.
- **Step 3** Run the **helm history huawei-csm -n huawei-csm** command to query the historical versions of the services deployed using Helm.

```
# helm history huawei-csm -n huawei-csm
REVISION UPDATED STATUS CHART APP VERSION
DESCRIPTION

1  Tue Aug 8 23:11:18 2023 superseded huawei-csm-2.1.0 1.0.0 Install complete
2  Wed Aug 9 04:19:10 2023 deployed huawei-csm-2.1.0 1.0.0 Upgrade complete
```

Step 4 Run the **helm rollback huawei-csm** *revision-number* **-n huawei-csm --wait -- timeout 2m** command to roll back the CSM services to the specified version. If

Rollback was a success is displayed in the command output, the CSM services are successfully rolled back to the specified version.

In the preceding command, *revision-number* indicates a version number queried in **Step 3**. For example, the version is **1**.

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

----End

4.1.5 Uninstalling the Software

Prerequisites

- CSM has been deployed using Helm 3.
- Resources created using CSM are no longer needed and have been deleted.

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-csm/helm/huawei-csm/** command to go to the Helm working directory.
- **Step 3** Run the **helm uninstall huawei-csm -n huawei-csm** command to uninstall CSM services. If **release "huawei-csm" uninstalled** is displayed in the command output, the services are successfully uninstalled.

helm uninstall huawei-csm -n huawei-csm release "huawei-csm" uninstalled

Step 4 (Optional) Run the **kubectl delete validatingWebhookConfiguration topo- service.xuanwu.huawei.io** command to delete the webhook resource. If the **topo- service.xuanwu.huawei.io** resource does not exist, skip this step.

kubectl delete validatingWebhookConfiguration topo-service.xuanwu.huawei.io

Step 5 (Optional) Run the **kubectl delete -f crds/** command to clear crd resources.

NOTICE

- If Huawei CSM is no longer used and related resource objects of Huawei CSM have been cleared from the environment, perform this step. Otherwise, skip this step.
- Before deleting crd resources, ensure that the resources created using Huawei CSM have been cleared. For details about the crd resource types, see the files in the /opt/huawei-csm/helm/huawei-csm/crds directory.
- Deleting a crd resource will clear all resources associated with the crd. Exercise caution when performing this operation.
- If a message similar to <Error from server (NotFound): error when deleting "crds/xuanwu.huawei.io_resourcetopologies.yaml": customresourcedefinitions.apiextensions.k8s.io "resourcetopologies.xuanwu.huawei.io" not found> is displayed, the crd resource has been uninstalled. In this case, ignore the message.

kubectl delete -f crds/ customresourcedefinition.apiextensions.k8s.io "resourcetopologies.xuanwu.huawei.io" deleted

Step 6 Run the **kubectl delete ns huawei-csm** command to delete the namespace.

NOTICE

Deleting a namespace will clear all resources in the namespace. Exercise caution when performing this operation.

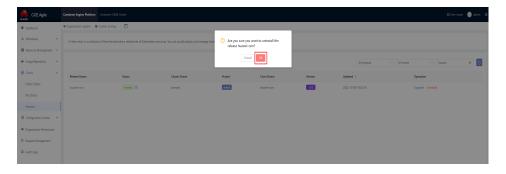
----End

4.1.6 Uninstalling the Software from CCE Agile

This section describes how to uninstall Huawei CSM from the CCE Agile platform. The following uses CCE Agile v22.3.2 as an example.

Procedure

- **Step 1** Log in to the CCE Agile platform.
- **Step 2** On the home page, choose **Charts** > **Releases**. The **Releases** page is displayed.
- **Step 3** Select a Huawei CSM release and click **Uninstall**. In the displayed dialog box, click **OK**.



----End

4.2 Manual Installation and Deployment

4.2.1 Manually Installing the Software

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 create namespace huawei-csm** command to create a namespace named **huawei-csm**.

kubectl create namespace huawei-csm namespace/huawei-csm created

- **Step 3** Run the **cd /opt/huawei-csm/manual/huawei-csm** command to go to the Huawei CSM installation directory.
- **Step 4** Run the **kubectl apply -f crds/** command to install crd resources.

```
# kubectl apply -f crds/
customresourcedefinition.apiextensions.k8s.io/resourcetopologies.xuanwu.huawei.io created
```

- **Step 5** Run the **cd /opt/huawei-csm/manual/huawei-csm/templates** command to go to the Huawei CSM working directory.
- **Step 6** (Optional) If the single-copy deployment mode is used, skip this step. If the multicopy deployment mode is used, you need to manually change the number of Deployment resource copies in all files in the **templates** directory and set all **enable-leader-election** parameters to **true**.

```
apiVersion: apps/v1
kind: Deployment
metadata:
labels:
app: csm-storage-service
name: csm-storage-service
namespace: huawei-csm
spec:
progressDeadlineSeconds: 600
replicas: 2
...
args:
---enable-leader-election=true
...
```

Step 7 (Optional) To configure the node selector of the CSM service, manually uncomment the **nodeSelector** field of the Deployment resources in all files in the **templates** directory and configure the corresponding node matching tag.

```
apiVersion: apps/v1
kind: Deployment
metadata:
name: csm-prometheus-service
...
spec:
# uncomment if you wish to configure selection constraints for csm-prometheus-service pods
nodeSelector:
kubernetes.io/hostname: <host-name>
...
```

Step 8 (Optional) To enable the Prometheus plug-in to provide the HTTPS service, manually configure the **prometheus-ssl.yaml** certificate file by following the instructions in **Table 4-1**. Run the **kubectl apply -f prometheus-ssl.yaml** command to create a certificate file. Manually uncomment the certificate-related fields in the **csm-prometheus.yaml** file and change the value of **use-https** to **true**. If the default HTTP service is used, skip this step.

```
# kubectl apply -f prometheus-ssl.yaml secret/prometheus-ssl created
```

Table 4-1 prometheus-ssl parameters

Parameter	Description	Mandatory	Default Value
data.tls.crt	Certificate information encoded using Base64.	Yes	-
data.tls.key	Private key information encoded using Base64.	Yes	-

```
prometheus-ssl.yaml
   containers:
     - name: prometheus-collector
        ---use-https=true # modify the value to "true" if configured the SSL cert
      volumeMounts:
# uncomment if configured the SSL cert
        - name: secret-volume
         mountPath: /etc/secret-volume
         readOnly: true
      livenessProbe:
       failureThreshold: 5
       httpGet:
# uncomment if configured the SSL cert
         scheme: HTTPS
         path: /healthz
         port: 8887
   volumes:
# uncomment if configured the SSL cert
     - name: secret-volume
      secret:
       secretName: prometheus-ssl
       defaultMode: 0400
```

Step 9 Run the **kubectl apply -f csm-prometheus.yaml** command to deploy the csm-prometheus service.

```
# kubectl apply -f csm-prometheus.yaml
serviceaccount/csm-prometheus-sa created
clusterrole.rbac.authorization.k8s.io/prometheus-collector-role created
clusterrolebinding.rbac.authorization.k8s.io/prometheus-collector-binding created
clusterrole.rbac.authorization.k8s.io/cmi-collector-role created
clusterrolebinding.rbac.authorization.k8s.io/cmi-collector-binding created
deployment.apps/csm-prometheus-service created
service/csm-prometheus-service created
```

Step 10 Run the **kubectl apply -f csm-storage.yaml** command to deploy the csm-storage service.

```
# kubectl apply -f csm-storage.yaml
serviceaccount/csm-storage-sa created
clusterrole.rbac.authorization.k8s.io/topo-service-role created
clusterrole.rbac.authorization.k8s.io/cmi-controller-role created
clusterrolebinding.rbac.authorization.k8s.io/topo-service-binding created
clusterrolebinding.rbac.authorization.k8s.io/cmi-controller-binding created
deployment.apps/csm-storage-service created
service/csm-storage-service created
```

Step 11 Run the **kubectl get pod -n huawei-csm** command to check the Pod creation result. If the following information is displayed, the Huawei CSM services are successfully created.

```
# kubectl get pod -n huawei-csm
NAME READY STATUS RESTARTS AGE
csm-prometheus-service-7c8bd8fd89-lbfvn 3/3 Running 0 3m21s
csm-storage-service-747dbc5cbd-n24j8 3/3 Running 0 2m19s
```

----End

4.2.2 Manually Updating the Software

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 1** Manually uninstall Huawei CSM of the earlier version by following the instructions in **4.2.3 Manually Uninstalling the Software**.
- **Step 2** Go to the **/opt/huawei-csm/manual/huawei-csm** directory where the new installation package is stored and run the **kubectl apply -f** *crds/* command to update crd resources.

kubectl apply -f crds/customresourcedefinition.apiextensions.k8s.io/resourcetopologies.xuanwu.huawei.io configured

Step 3 Install Huawei CSM of the current version. For details, see **4.2.1 Manually Installing the Software**.

----End

4.2.3 Manually Uninstalling the Software

Prerequisites

Huawei CSM has been installed, and Huawei CSM services are running properly.

Uninstalling the csm-prometheus Service

- **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-csm/manual/huawei-csm/templates** command to go to the Huawei CSM working directory.
- **Step 3** Run the **kubectl delete -f csm-prometheus.yaml** command to uninstall the csm-prometheus service.

kubectl delete -f csm-prometheus.yaml serviceaccount "csm-prometheus-sa" deleted clusterrole.rbac.authorization.k8s.io "prometheus-collector-role" deleted clusterrolebinding.rbac.authorization.k8s.io "prometheus-collector-binding" deleted clusterrole.rbac.authorization.k8s.io "cmi-collector-role" deleted clusterrolebinding.rbac.authorization.k8s.io "cmi-collector-binding" deleted deployment.apps "csm-prometheus-service" deleted service "csm-prometheus-service" deleted

Step 4 (Optional) If no certificate file is configured, skip this step. Run the **kubectl delete -f prometheus-ssl.yaml** command to delete the certificate secret file.

kubectl delete -f prometheus-ssl.yaml secret "prometheus-ssl" deleted

----End

Uninstalling the csm-storage Service

- **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-csm/manual/huawei-csm/templates** command to go to the Huawei CSM working directory.

Step 3 Run the **kubectl delete -f csm-storage.yaml** command to uninstall the csm-storage service.

kubectl delete -f csm-storage.yaml serviceaccount "csm-storage-sa" deleted clusterrole.rbac.authorization.k8s.io "topo-service-role" deleted clusterrole.rbac.authorization.k8s.io "cmi-controller-role" deleted clusterrolebinding.rbac.authorization.k8s.io "topo-service-binding" deleted clusterrolebinding.rbac.authorization.k8s.io "cmi-controller-binding" deleted deployment.apps "csm-storage-service" deleted service "csm-storage-service" deleted

----End

Uninstalling Other Resources

- **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-csm/manual/huawei-csm** command to go to the Huawei CSM installation directory.
- Step 3 Run the kubectl delete configmap -n huawei-csm huawei-csm-version command to delete the configmap resources.

kubectl delete configmap -n huawei-csm huawei-csm-version configmap "huawei-csm-version" deleted

- **Step 4** (Optional) If Huawei CSM is deployed in single-copy mode, skip this step. Run the **kubectl delete lease -n huawei-csm resource-topology** command to delete the **lease** resources.
- **Step 5** (Optional) Run the **kubectl delete -f crds/** command to clear crd resources.

NOTICE

- If Huawei CSM is no longer used and related resource objects of Huawei CSM have been cleared from the environment, perform this step. Otherwise, skip this step.
- Before deleting crd resources, ensure that the resources created using Huawei CSM have been cleared. For details about the crd resource types, see the files in the /opt/huawei-csm/manual/huawei-csm/crds directory.
- Deleting a crd resource will clear all resources associated with the crd. Exercise caution when performing this operation.
- If a message similar to <Error from server (NotFound): error when deleting "crds/xuanwu.huawei.io_resourcetopologies.yaml": customresourcedefinitions.apiextensions.k8s.io "resourcetopologies.xuanwu.huawei.io" not found> is displayed, the crd resource has been uninstalled. In this case, ignore the message.

kubectl delete -f crds/ customresourcedefinition.apiextensions.k8s.io "resourcetopologies.xuanwu.huawei.io" deleted

Step 6 Run the **kubectl delete ns huawei-csm** command to delete the namespace.

NOTICE

Deleting a namespace will clear all resources in the namespace. Exercise caution when performing this operation.

5 Configuring Prometheus

- 5.1 Installing Prometheus
- 5.2 Configuring the Prometheus Service
- 5.3 Configuring the Prometheus Dashboard

5.1 Installing Prometheus

Prometheus is an open-source systems monitoring and alerting toolkit. Prometheus joined the Cloud Native Computing Foundation (CNCF) in 2016 as the second hosted project, after Kubernetes.

For details about how to obtain and install Prometheus, click here.

For other information about Prometheus, click here.

5.2 Configuring the Prometheus Service

Configuration Description

Prometheus invokes the interfaces provided by the plug-in to monitor storage data. No matter Prometheus is deployed in Kubernetes or on an independent server, a configuration file is required to start Prometheus and monitor the plug-in.

This section only describes how to add configurations related to the plug-in to the Prometheus configuration file. For details about other configurations, **click here** to see the Prometheus official document.

Step 1 Modify the Prometheus configuration file. Add the storage backend monitoring items created using CSM to the **scrape_configs** section.

□ NOTE

This configuration file is the configuration file of the Prometheus platform. Its location varies depending on the deployment mode. For details, **click here** to see the Prometheus official document.

Step 2 Add resource monitoring items. The configuration template is as follows:

```
scrape_configs:
- job_name: *****
 scrape_interval: ***
 scrape_timeout: ***
 metrics_path: /object/***
 scheme: http/https
 params:
   controller: ["]
   storagepool: ["]
   filesystem: ["]
   lun: ["]
   array: ["]
pv: ["]
 tls_config:
  ca_file: ******
cert_file: ******
   key_file: *****
 static_configs:
- targets: ['*.*.*:<Port number>']
```

Table 5-1 describes the parameters in the **scrape_configs** section.

Table 5-1 Parameter description

Parameter	Description	Example	
job_name	Monitoring job name.	job_name: OceanStor- Monitor	
scheme	Web request mode. If this parameter is not set, the default value http is used. After tls_config is configured in the plug-in, set this parameter to https.	scheme: https	
scrape_interv al	Data scraping interval.	scrape_interval: 15m	
scrape_timeo ut	Data scraping timeout. You are advised to set this parameter to be the same as the value of scrape_interval.	scrape_timeout: 15m	
metrics_path	Data scraping path, in the format of /object/***. **** indicates the storage backend name in the plug-in configuration file. NOTE Only users with the corresponding roles in the storage system group can query the metric data of the	metrics_path: /object/ backend1	
	backend. Such roles include the super administrator, administrator, and monitoring administrator.		

Parameter	Description	Example
params	Monitored object of centralized storage. Currently, the following types are supported: • controller: controller • storagepool: storage pool • filesystem: file system • lun: LUN • array: array • pv: PV data of Kubernetes NOTE • If you do not need a certain object, you can delete it. • The value of an object parameter must be in the format of ["]. • For details about object metric data, see Table 2-4.	controller: ["] storagepool: ["] filesystem: ["] lun: ["] array: ["] pv: ["]
tls_config	 TLS configuration when scheme is set to https. This parameter is mandatory when scheme is set to https. ca_file: path of the CA certificate used to verify the API server certificate. cert_file: path of the certificate file used for client certificate authentication on the server. key_file: path of the key file used for client certificate authenticate authenticate authentication on the server. 	ca_file: /opt/huawei/ca.crt cert_file: /opt/huawei/ client.crt key_file: /opt/huawei/ client.key NOTE The paths can be customized.
targets	Listening address and port of the plug-in. For Kubernetes deployment, the default port is 30074. The specific port is the one specified during plug-in deployment.	['192.168.1.1:30074']

₩ NOTE

For details about **scrape_configs** parameters, **click here** to visit the Prometheus official website.

Step 3 Add performance monitoring items. The configuration template is as follows:

scrape_configs:
- job_name: ******

```
scrape_interval: ***
scrape_timeout: ***
metrics_path: /performance/***
params:
controller: ['21,22,370']
storagepool: ['21,22,370']
lun: ['21,22,370']
filesystem: ['182,524,525']
pv: ['filesystem,lun'] #
tls_config:
ca_file: ******
cert_file: ******
key_file: ******
static_configs:
- targets: ['*:****:<Port number>']
```

Table 5-2 describes the parameters in the **scrape_configs** section.

Table 5-2 Parameter description

Parameter	Description	Example	
job_name	Monitoring job name.	job_name: OceanStor- Monitor	
scheme	Web request mode. If this parameter is not set, the default value http is used. You can set this parameter to https .	scheme: https	
scrape_interv al	Data scraping interval.	scrape_interval: 15m	
scrape_timeo ut	Data scraping timeout. You are advised to set this parameter to be the same as the value of scrape_interval.	scrape_timeout: 15m	
metrics_path	Data scraping path, in the format of /performance/***. **** indicates the storage backend name in the plug-in configuration file. NOTE Only users with the corresponding roles in the storage system group can query the metric data of the backend. Such roles include the super administrator, administrator, and monitoring administrator.	metrics_path: /performance/ backend1	

Parameter	Description	Example
params	Currently, the following types are supported: controller: controller storagepool: storage pool filesystem: file system lun: LUN pv: PV data of Kubernetes NOTE If you do not need a certain object, you can delete it. The object parameter value is used to specify the performance metric of the object. The format is 'Metric 1,Metric 2'. If you do not need a certain metric, you can delete it. For details about performance metric data and mappings, see Table 2-5.	controller: ['21,22,23,25,26,28,370'] storagepool: ['21,22,370'] lun: ['21,22,370'] filesystem: ['182,524,525'] pv: ['filesystem,lun']
tls_config	TLS configuration when scheme is set to https. This parameter is mandatory when scheme is set to https. • ca_file: path of the CA certificate used to verify the API server certificate. • cert_file: path of the certificate file used for client certificate authentication on the server. • key_file: path of the key file used for client certificate authenticate authenticate authentication on the server.	ca_file: /opt/huawei/ca.crt cert_file: /opt/huawei/ client.crt key_file: /opt/huawei/ client.key NOTE The paths can be customized.
targets	Listening address and port of the plug-in. For Kubernetes deployment, the default port is 30074. The specific port is the one specified during plug-in deployment. For non-Kubernetes deployment, no default port is available. The specific port is the one specified during plug-in deployment.	['192.168.1.1:30074']

■ NOTE

For details about **scrape_configs** parameters, **click here** to visit the Prometheus official website

- **Step 4** To monitor multiple storage backends, repeat **Step 2** and **Step 3**.
- **Step 5** Restart the Prometheus service.

□ NOTE

Prometheus is an open-source component. The startup mode varies depending on the deployment mode. For details, **click here** to see the Prometheus official document.

----End

5.3 Configuring the Prometheus Dashboard

- **Step 1** In the address box of a browser, enter the IP address and port of the Prometheus service to log in to the Prometheus monitoring UI. The default port is **9090**.
- **Step 2** Click the **Graph** tab, enter **huawei** in the search box, and select the desired monitoring item.

6 Configuring Grafana

- 6.1 Installing Grafana
- 6.2 Using Grafana

6.1 Installing Grafana

Grafana is an open-source visualization platform that provides full support for Prometheus.

- Grafana 2.5.0 (released on October 28, 2015) and later versions allow Prometheus to function as a Grafana data source. Currently, Huawei storage can interconnect with only Grafana 7.0.4
- CCE Agile provides the Grafana component and has the default Prometheus data source. You can go to **Using Grafana to Monitor Storage** to perform the next step.

For details about how to install Grafana, see **the Grafana official documentation**.

6.2 Using Grafana

□ NOTE

By default, Grafana listens on port 3000.

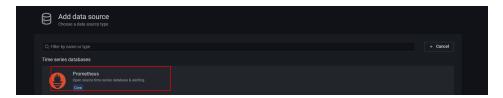
Adding Prometheus and Configuring a Data Source

Step 1 In the navigation pane on the left, choose **Configuration** > **Data sources** and click **Add data source** to add a data source.



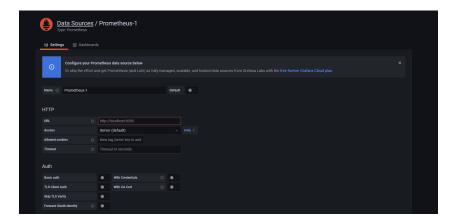


Step 2 Select Prometheus and add a data source.



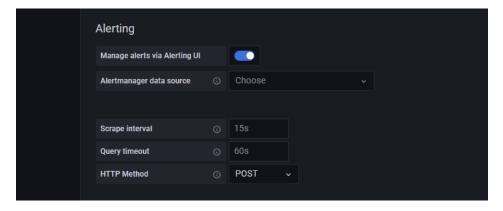
Step 3 Configure data source information.

- Name: name of the data source.
- URL: IP address of the Prometheus server.
- Access: Set this parameter based on site requirements.
 - If this parameter is set to Server, a browser sends a request to the Grafana server and then the Grafana server sends a request to the Prometheus server to obtain data.
 - If this parameter is set to **Browser**, a browser directly sends a request to the Prometheus server to obtain data. (The cross-domain problem needs to be solved.)

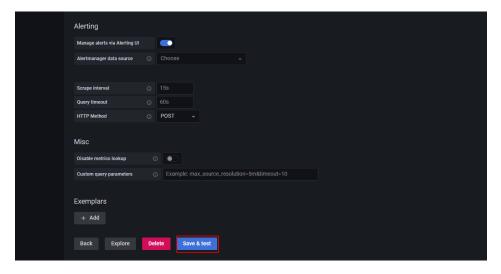


Step 4 Set the HTTP method.

• **POST**: recommended when a large amount of data is queried.



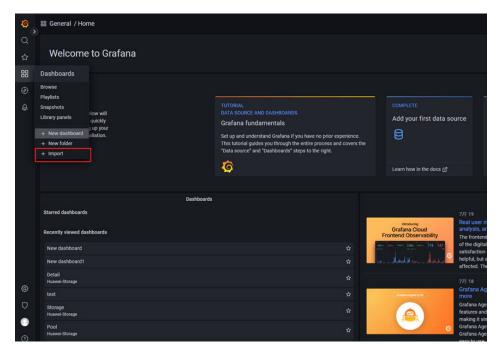
Step 5 Click **Save & Test** to save the new data source.



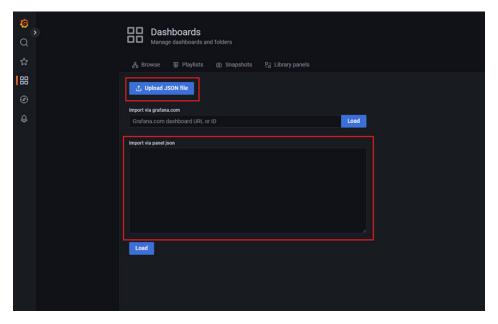
----End

Using Grafana to Monitor Storage

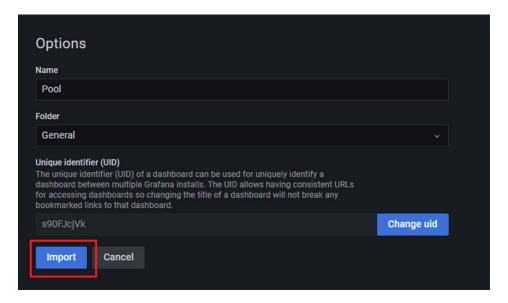
Step 1 In the navigation pane on the left, choose **Dashboards** > **Import**.



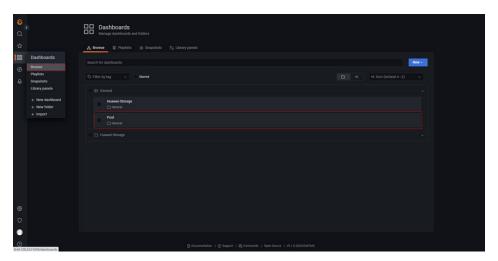
Step 2 Click Upload JSON file to upload a JSON file or directly paste JSON data to load the existing dashboard. After the loading is successful, you can view the JSON file in the helm/huawei-csm/grafana/OceanStor.json directory in the path where the CSM software package is decompressed on the Browse tab page. The JSON file is a sample template.



Step 3 After setting the dashboard name and folder, click **Import**.



Step 4 In the navigation pane on the left, choose **Dashboards** > **Browse**. On the **Browse** tab page, find the imported dashboard and click it.



Step 5 In the **DataSource** drop-down list on the dashboard, select the configured data source.



□ NOTE

The Huawei logo in the upper left corner of the template can be loaded only in a public network environment.

7 Common O&M Guide

7.1 Collecting Logs

7.2 Viewing Version Information

7.1 Collecting Logs

NOTICE

If you use the oceanctl tool to collect logs, the CPU usage on the master node may increase sharply in a short period of time. Therefore, determine whether to use this function based on the workload and CPU usage on the master node. You can also manually collect CSM logs in the log archive directory (/var/log/huawei-csm/) on each node.

7.1.1 Performing Check Before Collection

- **Step 1** Use a remote access tool, such as PuTTY, to log in to the node where the oceanctl tool is installed in the Kubernetes cluster through the management IP address.
- **Step 2** Run the **oceanctl version** command. The displayed version is **v4.3.0**.

Step 3 Run the **oceanctl --help** command. The following information is displayed.

```
$ oceanctl --help
A CLI tool for Ocean Storage in Kubernetes

Usage:
    oceanctl [command]

Available Commands:
    collect collect messages in Kubernetes
    create Create a resource to Ocean Storage in Kubernetes
    delete Delete one or more resources from Ocean Storage in Kubernetes
    get Ocean Ocean Storage in Kubernetes
```

```
help Help about any command
update Update a resource for Ocean Storage in Kubernetes
version Print the version of oceanctl

Flags:
-h, --help help for oceanctl

Use "oceanctl [command] --help" for more information about a command.
```

Step 4 Run the **kubectl get deploy -n** *\${NAMESPACE}* command to check whether a Pod is started properly. In the preceding command, *\${NAMESPACE}* indicates the namespace for installing CSM. **huawei-csm** is used as an example.

```
$ kubectl get deploy -n huawei-csm
NAME READY UP-TO-DATE AVAILABLE AGE
csm-prometheus-service 1/1 1 1 3h23m
csm-storage-service 1/1 1 1 3h23m
```

----End

7.1.2 Collecting CSM Logs Using oceanctl

- **Step 1** Use a remote access tool, such as PuTTY, to log in to the node checked in **7.1.1 Performing Check Before Collection** through the management IP address.
- **Step 2** Run the **oceanctl collect logs -n huawei-csm -a** command to collect CSM logs of all nodes where CSM containers reside in the cluster.

Step 3 Check the log package generated in the **/tmp** directory. You can run the **unzip** \$ {zip_name} -d **collect_logs** command to decompress the log package. In the preceding command, \${zip_name} indicates the package name.

```
$ date
Fri Sep 8 22:14:57 CST 2023
$ ls /tmp
huawei-csm-2023-09-08-22:13:01-all.zip
```

----End

7.2 Viewing Version Information

- **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 -n** *\${NAMESPACE}* **huawei-csm-version -o yaml** command. In the preceding command, *\${NAMESPACE}* indicates the namespace name, for example, **huawei-csm**.

```
$ kubectl get cm -n huawei-csm huawei-csm-version -o yaml
apiVersion: v1
data:
cmi-controller: 2.1.0
liveness-probe: 2.1.0
prometheus-collector: 2.1.0
topo-service: 2.1.0
kind: ConfigMap
metadata:
```

name: huawei-csm-version namespace: huawei-csm

8 Appendix

- 8.1 Accessing Kubernetes As a Non-root User
- 8.2 Configuring the HTTPS Service for csm-prometheus
- 8.3 Managing Storage Certificates
- 8.4 Permission Matrix

8.1 Accessing Kubernetes As a Non-root User

Prerequisites

The non-root user has the sudo permission to use /bin/cp and /bin/chown.

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 **mkdir -p \$HOME/.kube** command to create a directory for storing the authentication file of the Kubernetes cluster.

\$ mkdir -p \$HOME/.kube

Step 3 Run the **sudo cp -i /etc/kubernetes/admin.conf \$HOME/.kube/config** command to copy the authentication file of the Kubernetes cluster.

Replace /etc/kubernetes/admin.conf with the actual authentication file. \$ sudo cp -i /etc/kubernetes/admin.conf \$HOME/.kube/config

- Step 4 Run the sudo chown \$(id -u):\$(id -g) \$HOME/.kube/config command to change the user and user group of the authentication file.

 \$ sudo chown \$(id -u):\$(id -q) \$HOME/.kube/config
- **Step 5** Run the following commands to configure the **KUBECONFIG** environment variable for the current user (Ubuntu 20.04 is used as an example).

\$ echo "export KUBECONFIG=\$HOME/.kube/config" >> ~/.bashrc
\$ source ~/.bashrc

8.2 Configuring the HTTPS Service for csm-prometheus

8.2.1 Configuring the HTTPS Service Certificate of csmprometheus

Prerequisites

- Kubernetes is running properly.
- You have obtained the certificate and key.

Procedure

- **Step 1** Use a remote access tool, such as PuTTY, to log in to the node where the **helm** folder of CSM is stored in the Kubernetes cluster through the management IP address.
- **Step 2** Run the **cd /opt/huawei-csm/helm/huawei-csm** command to go to the Helm working directory.
- **Step 3** Run the **mkdir cert** command to create a folder for storing the certificate.
- **Step 4** Save the certificate and key files to the **/opt/huawei-csm/helm/huawei-csm/cert** directory.

```
-rw-r--r-- 1 root root 4.5K Sep 8 23:00 server.crt
-rw-r--r-- 1 root root 1.7K Sep 8 23:00 server.key
```

- Step 5 Run the vim /opt/huawei-csm/helm/huawei-csm/values.yaml command. Then:
 - Set features.prometheusCollector.prometheusCollectorSSL.enabled to true.
 - Set features.prometheusCollector.prometheusCollectorSSL.certPath to "cert/server.crt".
 - Set **features.prometheusCollector.prometheusCollectorSSL.keyPath** to "**cert/server.key**".

```
# all supported features
features:
 # prometheusCollector: allowed prometheus use the storage to collect metrics
 prometheusCollector:
  # Allowed values:
  # true: enable prometheus collect feature
  # false: disable prometheus collect feature
  # Default value: false
  enabled: true
  # nodePort: port the containers are provided to the prometheus
  # Default value: 30074
  nodePort: 30074
  # prometheusCollectorSSL: parameters required to start https
  # Default value: 30074
  prometheusCollectorSSL:
    # Allowed values:
      true: enable https, when set it certPath and keyPath must set
      false: disable https, use http
    # Default value: true
    enabled: true
    # The Path of cert, need to be placed in the huawei-csm directory
    certPath: "cert/server.crt"
```

```
# The Path of key, need to be placed in the huawei-csm directory keyPath: "cert/server.key"
```

Step 6 Complete the installation by referring to **4.1.1 Installing the Software**.

----End

8.2.2 Deleting the HTTPS Service Certificate of csmprometheus

Prerequisites

- Kubernetes is running properly.
- You have obtained the certificate and key.

Procedure

- **Step 1** Use a remote access tool, such as PuTTY, to log in to the node where the **helm** folder of CSM is stored in the Kubernetes cluster through the management IP address.
- **Step 2** Run the **cd /opt/huawei-csm/helm/huawei-csm** command to go to the Helm working directory.
- Step 3 Run the helm get values huawei-csm -n huawei-csm -a > update-value.yaml command to obtain the original service configuration file.
- **Step 4** Run the **vim /opt/huawei-csm/helm/huawei-csm/update-value.yaml** command. Then:
 - Set features.prometheusCollector.prometheusCollectorSSL.enabled to false.
 - Set features.prometheusCollector.prometheusCollectorSSL.certPath to "".
 - Set features.prometheusCollector.prometheusCollectorSSL.keyPath to "".

```
features:
prometheusCollector:
csiDriverName: csi.huawei.com
enabled: true
nodePort: 30074
prometheusCollectorSSL:
enabled: false
certPath: ""
keyPath: ""
```

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

----End

8.2.3 Updating the HTTPS Service Certificate of csmprometheus

Step 1 Use a remote access tool, such as PuTTY, to log in to the node where the **helm** folder of CSM is stored in the Kubernetes cluster through the management IP address.

- **Step 2** Run the **cd /opt/huawei-csm/helm/huawei-csm** command to go to the Helm working directory.
- **Step 3** Run the **mkdir cert** command to create a folder for storing the certificate.
- **Step 4** Save the new certificate and key files to the **/opt/huawei-csm/helm/huawei-csm/cert** directory.

```
-rw-r--r- 1 root root 4.5K Sep 8 23:00 server.crt
-rw-r--r- 1 root root 1.7K Sep 8 23:00 server.key
```

- Step 5 Run the helm get values huawei-csm -n huawei-csm -a > update-value.yaml command to obtain the original service configuration file.
- **Step 6** Run the **vim /opt/huawei-csm/helm/huawei-csm/update-value.yaml** command. Then:
 - Set **features.prometheusCollector.prometheusCollectorSSL.enabled** to **true**
 - Set features.prometheusCollector.prometheusCollectorSSL.certPath to "cert/server.crt".
 - Set features.prometheusCollector.prometheusCollectorSSL.keyPath to "cert/server.key".

```
features:
prometheusCollector:
csiDriverName: csi.huawei.com
enabled: true
nodePort: 30074
prometheusCollectorSSL:
enabled: true
certPath: "cert/server.crt"
keyPath: "cert/server.key"
```

Step 7 Run the helm upgrade huawei-csm ./ -n huawei-csm -f ./values.yaml -f update-value.yaml --wait --timeout 2m command to upgrade CSM services. If Release "huawei-csm" has been upgraded is displayed in the command output, the CSM services are successfully upgraded.

----End

8.3 Managing Storage Certificates

For details, see section "Storage Backend Management" in the user guide of Huawei Storage Kubernetes CSI.

8.4 Permission Matrix

Table 8-1 Permission matrix

ClusterRole	ApiGroups	Resources	Verbs
prometheus- collector-role	-	"persistentvolumes"," persistentvolume- claims","pods"	"get","list"

ClusterRole	ApiGroups	Resources	Verbs
	"xuanwu.huaw ei.io"	"storagebackendclai ms"	"get","list"
cmi-collector- role	"xuanwu.huaw ei.io"	"storagebackendclai ms"	"get"
	-	"secrets"	"get"
	-	"configmaps"	"create", "get", "update"
topo-service- role	-	"secrets", "events", "configmaps"	"create", "get", "update", "delete"
	"coordination. k8s.io"	"leases"	"create", "get", "update", "delete"
	"xuanwu.huaw ei.io"	"resourcetopologies", "resourcetopologies/ status"	"create", "get", "list", "watch", "update", "delete"
	"*"	11*11	"get", "list", "watch"
cmi-controller- role	"xuanwu.huaw ei.io"	"storagebackendclai ms"	"get"
	-	"secrets"	"get"
	-	"configmaps"	"create", "get", "update"

9 FAQS

- 9.1 Pod in OOMKilled State
- 9.2 Manually Adjusting the Number of Concurrent Data Scraping Requests
- 9.3 Residual Pod Labels on the Storage Side

9.1 Pod in OOMKilled State

Symptom

Check the Pod status. The Pod state is **OOMKilled**, and alarm "unable to set memory limit to xxx (current usage: xxx, peak usage: xxx): unknown" is generated in the Pod.

```
[root@k8s-master1-yqz CSMImage]# kubectl get pod -n huawei-csm csm-prometheus-service-65b446d597-ljx7q
NAME READY STATUS RESTARTS AGE
csm-prometheus-service-65b446d597-ljx7q 2/3 00MKilled 3 (<invalid> ago) 43s
```

Root Cause Analysis

The container memory usage in the Pod reaches the upper limit. You need to increase the upper limit of the available memory.

Solution

Modify the container resource configuration and update CSM. For details, see **4.1.3 Upgrading the Software**.

9.2 Manually Adjusting the Number of Concurrent Data Scraping Requests

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 edit deployments.apps -n huawei-csm csm-prometheus-service command to edit the deployments configuration item.
- **Step 3** For example, to set the number of concurrent requests to 10, add **args** list item **-- client-max-threads=10** to the container whose **name** is **cmi-controller** as follows

```
apiVersion: apps/v1
kind: Deployment
metadata:
 name: csm-prometheus-service
 namespace: huawei-csm
spec:
 template:
  metadata:
   creationTimestamp: null
   labels:
    app: csm-prometheus-service
  spec:
   containers:
    - args:
     - --cmi-address=$(ENDPOINT)
     - --cmi-name=cmi.huawei.com
     - --page-size=100
     - --backend-namespace=huawei-csi
     - --log-file-dir=/var/log/huawei-csm/csm-prometheus-service
     - --log-file=cmi-service
     - --logging-module=file
     - --log-level=info
     - --log-file-size=20M
     - --max-backups=9
     - --client-max-threads=10
     name: cmi-controller
```

- **Step 4** Run the :wq command to save the configuration and exit.
- **Step 5** Run the **kubectl get pod -n huawei-csm** command and wait until the container is restarted.
- Step 6 Run the kubectl get deployments.apps -n huawei-csm csm-prometheus-service -o yaml | grep client-max-threads command to check whether the configuration is successful.

----End

9.3 Residual Pod Labels on the Storage Side

Symptom

After CSM is upgraded from 1.x.x to 2.x.x, residual Pod topo relationships exist on storage resources and cannot be cleared.

Root Cause Analysis

In 1.x.x, the name of a topo resource is the name of a storage resource. In 2.x.x, the name of a topo resource is changed to a PV name in the cluster, and the prefix is changed from **topo**- to **rt**-.

1.*x.x*.

NAME PROVISIONER VOLUMEHANDLE

STATUS AGE
topo-pvc-0a8f7871-f26e-4665-b4be-53dfa9c878cb cmi.huawei.com 181-iscsi.pvc-0a8f7871-f26e-4665-b4be-53dfa9c878cb Normal 353d
topo-pvc-3b6b1dd6-b3dd-4394-8251-4bd6009411cf cmi.huawei.com 181-iscsi.pvc-3b6b1dd6-b3dd-4394-8251-4bd6009411cf Normal 354d

2.x.x and later:

NAME PROVISIONER VOLUMEHANDLE
STATUS AGE
rt-pvc-0a8f7871-f26e-4665-b4be-53dfa9c878cb cmi.huawei.com 181-iscsi.pvc-0a8f7871-f26e-4665-b4be-53dfa9c878cb Normal 353d
rt-pvc-3b6b1dd6-b3dd-4394-8251-4bd6009411cf cmi.huawei.com 181-iscsi.pvc-3b6b1dd6-b3dd-4394-8251-4bd6009411cf Normal 354d

After CSM is upgraded to 2.x.x, the topology service creates new topo resources based on the resource information in the cluster and binds the new topo resources to the topology relationships of the source version on the storage side.

If a Pod whose topology relationship has been to the storage device is deleted when the CSM topology service is disabled, the old topology relationship bound to the topo resource cannot be created based on the existing resources in the cluster after the CSM topology service is enabled again.

In this case, the old topology relationship remains on the storage device and cannot be cleared.

Solution or Workaround

If the preceding storage topology relationship remains, you can only delete the storage resources to clear the residual topology relationship.

During the upgrade from CSM 1.x.x to 2.x.x, if the CSM topology service is enabled in the source version, enabling the CSM topology service during the upgrade can prevent the preceding problem.