IBM Storage Scale

Container Storage Interface Driver Guide Version 2.11

SC28-3113-19

Note

Before using this information and the product it supports, read the information in "Notices" on page 71.

This edition applies to Version 5 release 2 modification 0 of the following products, and to all subsequent releases and modifications until otherwise indicated in new editions:

- IBM Storage Scale Data Management Edition ordered through Passport Advantage® (product number 5737-F34)
- IBM Storage Scale Data Access Edition ordered through Passport Advantage (product number 5737-I39)
- IBM Storage Scale Erasure Code Edition ordered through Passport Advantage (product number 5737-J34)
- IBM Storage Scale Data Management Edition ordered through AAS (product numbers 5641-DM1, DM3, DM5)
- IBM Storage Scale Data Access Edition ordered through AAS (product numbers 5641-DA1, DA3, DA5)
- IBM Storage Scale Data Management Edition for IBM® ESS (product number 5765-DME)
- IBM Storage Scale Data Access Edition for IBM ESS (product number 5765-DAE)
- IBM Storage Scale Backup ordered through Passport Advantage® (product number 5900-AXJ)
- IBM Storage Scale Backup ordered through AAS (product numbers 5641-BU1, BU3, BU5)
- IBM Storage Scale Backup for IBM® Storage Scale System (product number 5765-BU1)

Significant changes or additions to the text and illustrations are indicated by a vertical line (|) to the left of the change.

IBM welcomes your comments; see the topic "How to send your comments" on page xxv. When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

Contents

Tables

Figures

About this information

This edition applies to IBM Storage Scale version 5.2.0 for AIX®, Linux®, and Windows.

IBM Storage Scale is a file management infrastructure, based on IBM General Parallel File System (GPFS) technology, which provides unmatched performance and reliability with scalable access to critical file data.

To find out which version of IBM Storage Scale is running on a particular AIX node, enter:

```
lslpp -l gpfs\*
```

To find out which version of IBM Storage Scale is running on a particular Linux node, enter:

```
rpm -qa | grep gpfs     (for SLES and Red Hat Enterprise Linux)
```

```
dpkg -l | grep gpfs     (for Ubuntu Linux)
```

To find out which version of IBM Storage Scale is running on a particular Windows node, open **Programs and Features** in the control panel. The IBM Storage Scale installed program name includes the version number.

Which IBM Storage Scale information unit provides the information you need?

The IBM Storage Scale library consists of the information units listed in Table 1 on page x.

To use these information units effectively, you must be familiar with IBM Storage Scale and the AIX, Linux, or Windows operating system, or all of them, depending on which operating systems are in use at your installation. Where necessary, these information units provide some background information relating to AIX, Linux, or Windows. However, more commonly they refer to the appropriate operating system documentation.

Note: Throughout this documentation, the term "Linux" refers to all supported distributions of Linux, unless otherwise specified.

Table 1. IBM Storage Scale library information units		
Information unit	**Type of information**	**Intended users**
IBM Storage Scale: Concepts, Planning, and Installation Guide	This guide provides the following information: **Product overview** • Overview of IBM Storage Scale • GPFS architecture • Protocols support overview: Integration of protocol access methods with GPFS • Active File Management • AFM-based Asynchronous Disaster Recovery (AFM DR) • Introduction to AFM to cloud object storage • Introduction to system health and troubleshooting • Introduction to performance monitoring • Data protection and disaster recovery in IBM Storage Scale • Introduction to IBM Storage Scale GUI • IBM Storage Scale management API • Introduction to Cloud services • Introduction to file audit logging • Introduction to clustered watch folder • Understanding call home • IBM Storage Scale in an OpenStack cloud deployment • IBM Storage Scale product editions • IBM Storage Scale license designation • Capacity-based licensing • Dynamic pagepool	System administrators, analysts, installers, planners, and programmers of IBM Storage Scale clusters who are very experienced with the operating systems on which each IBM Storage Scale cluster is based
IBM Storage Scale: Concepts, Planning, and Installation Guide	• Firewall recommendations • Considerations for GPFS applications • Security-Enhanced Linux support • Space requirements for call home data upload	
IBM Storage Scale: Concepts, Planning, and Installation Guide	• Firewall recommendations • Considerations for GPFS applications • Security-Enhanced Linux support • Space requirements for call home data upload	

Table 1. IBM Storage Scale library information units (continued)		
Information unit	**Type of information**	**Intended users**
IBM Storage Scale: Concepts, Planning, and Installation Guide	**Installing** • Steps for establishing and starting your IBM Storage Scale cluster • Installing IBM Storage Scale on Linux nodes and deploying protocols • Installing IBM Storage Scale on public cloud by using cloudkit • Installing IBM Storage Scale on AIX nodes • Installing IBM Storage Scale on Windows nodes • Installing Cloud services on IBM Storage Scale nodes • Installing and configuring IBM Storage Scale management API • Installing GPUDirect Storage for IBM Storage Scale • Installation of Active File Management (AFM) • Installing AFM Disaster Recovery • Installing call home • Installing file audit logging • Installing clustered watch folder • Installing the signed kernel modules for UEFI secure boot • Steps to permanently uninstall IBM Storage Scale **Upgrading** • IBM Storage Scale supported upgrade paths • Online upgrade support for protocols and performance monitoring • Upgrading IBM Storage Scale nodes	System administrators, analysts, installers, planners, and programmers of IBM Storage Scale clusters who are very experienced with the operating systems on which each IBM Storage Scale cluster is based

Table 1. IBM Storage Scale library information units (continued)		
Information unit	**Type of information**	**Intended users**
IBM Storage Scale: Concepts, Planning, and Installation Guide	• Upgrading IBM Storage Scale non-protocol Linux nodes • Upgrading IBM Storage Scale protocol nodes • Upgrading IBM Storage Scale on cloud • Upgrading GPUDirect Storage • Upgrading AFM and AFM DR • Upgrading object packages • Upgrading SMB packages • Upgrading NFS packages • Upgrading call home • Upgrading the performance monitoring tool • Upgrading signed kernel modules for UEFI secure boot • Manually upgrading pmswift • Manually upgrading the IBM Storage Scale management GUI • Upgrading Cloud services • Upgrading to IBM Cloud Object Storage software level 3.7.2 and above • Upgrade paths and commands for file audit logging and clustered watch folder • Upgrading IBM Storage Scale components with the installation toolkit • Protocol authentication configuration changes during upgrade • Changing the IBM Storage Scale product edition • Completing the upgrade to a new level of IBM Storage Scale • Reverting to the previous level of IBM Storage Scale	System administrators, analysts, installers, planners, and programmers of IBM Storage Scale clusters who are very experienced with the operating systems on which each IBM Storage Scale cluster is based
IBM Storage Scale: Concepts, Planning, and Installation Guide	• Coexistence considerations • Compatibility considerations • Considerations for IBM Storage Protect for Space Management • Applying maintenance to your IBM Storage Scale system • Guidance for upgrading the operating system on IBM Storage Scale nodes • Considerations for upgrading from an operating system not supported in IBM Storage Scale 5.1.x.x • Servicing IBM Storage Scale protocol nodes • Offline upgrade with complete cluster shutdown	

Table 1. IBM Storage Scale library information units (continued)		
Information unit	**Type of information**	**Intended users**
IBM Storage Scale: Administration Guide	This guide provides the following information: **Configuring** • Configuring the GPFS cluster • Configuring GPUDirect Storage for IBM Storage Scale • Configuring the CES and protocol configuration • Configuring and tuning your system for GPFS • Parameters for performance tuning and optimization • Ensuring high availability of the GUI service • Configuring and tuning your system for Cloud services • Configuring IBM Power Systems for IBM Storage Scale • Configuring file audit logging • Configuring clustered watch folder • Configuring the cloudkit • Configuring Active File Management • Configuring AFM-based DR • Configuring AFM to cloud object storage • Tuning for Kernel NFS backend on AFM and AFM DR • Configuring call home • Integrating IBM Storage Scale Cinder driver with Red Hat OpenStack Platform 16.1 • Configuring Multi-Rail over TCP (MROT) • Dynamic pagepool configuration	System administrators or programmers of IBM Storage Scale systems
IBM Storage Scale: Administration Guide	**Administering** • Performing GPFS administration tasks • Performing parallel copy with mmxcp command • Protecting file data: IBM Storage Scale safeguarded copy • Verifying network operation with the mmnetverify command • Managing file systems • File system format changes between versions of IBM Storage Scale • Managing disks	System administrators or programmers of IBM Storage Scale systems

Table 1. IBM Storage Scale library information units (continued)		
Information unit	**Type of information**	**Intended users**
IBM Storage Scale: Administration Guide	• Managing protocol services • Managing protocol user authentication • Managing protocol data exports • Managing object storage • Managing GPFS quotas • Managing GUI users • Managing GPFS access control lists • Native NFS and GPFS • Accessing a remote GPFS file system • Information lifecycle management for IBM Storage Scale • Creating and maintaining snapshots of file systems • Creating and managing file clones • Scale Out Backup and Restore (SOBAR) • Data Mirroring and Replication • Implementing a clustered NFS environment on Linux • Implementing Cluster Export Services • Identity management on Windows / RFC 2307 Attributes • Protocols cluster disaster recovery • File Placement Optimizer • Encryption • Managing certificates to secure communications between GUI web server and web browsers • Securing protocol data • Managing file audit logging • RDMA tuning • Configuring Mellanox Memory Translation Table (MTT) for GPFS RDMA VERBS Operation • Administering cloudkit • Administering AFM • Administering AFM DR	System administrators or programmers of IBM Storage Scale systems
IBM Storage Scale: Administration Guide	• Administering AFM to cloud object storage • Highly available write cache (HAWC) • Local read-only cache • Miscellaneous advanced administration topics • GUI limitations	System administrators or programmers of IBM Storage Scale systems

Table 1. IBM Storage Scale library information units (continued)		
Information unit	**Type of information**	**Intended users**
IBM Storage Scale: Problem Determination Guide	This guide provides the following information: **Monitoring** • Monitoring system health by using IBM Storage Scale GUI • Monitoring system health by using the mmhealth command • Dynamic pagepool monitoring • Performance monitoring • Monitoring GPUDirect storage • Monitoring events through callbacks • Monitoring capacity through GUI • Monitoring AFM and AFM DR • Monitoring AFM to cloud object storage • GPFS SNMP support • Monitoring the IBM Storage Scale system by using call home • Monitoring remote cluster through GUI • Monitoring file audit logging • Monitoring clustered watch folder • Monitoring local read-only cache **Troubleshooting** • Best practices for troubleshooting • Understanding the system limitations • Collecting details of the issues • Managing deadlocks • Installation and configuration issues • Upgrade issues • CCR issues • Network issues • File system issues • Disk issues • GPUDirect Storage troubleshooting • Security issues • Protocol issues • Disaster recovery issues • Performance issues	System administrators of GPFS systems who are experienced with the subsystems used to manage disks and who are familiar with the concepts presented in the *IBM Storage Scale: Concepts, Planning, and Installation Guide*

Table 1. IBM Storage Scale library information units (continued)		
Information unit	**Type of information**	**Intended users**
IBM Storage Scale: Problem Determination Guide	• GUI and monitoring issues • AFM issues • AFM DR issues • AFM to cloud object storage issues • Transparent cloud tiering issues • File audit logging issues • Cloudkit issues • Troubleshooting mmwatch • Maintenance procedures • Recovery procedures • Support for troubleshooting • References	

Table 1. IBM Storage Scale library information units (continued)		
Information unit	**Type of information**	**Intended users**
IBM Storage Scale: Command and Programming Reference Guide	This guide provides the following information: **Command reference** • cloudkit command • gpfs.snap command • mmaddcallback command • mmadddisk command • mmaddnode command • mmadquery command • mmafmconfig command • mmafmcosaccess command • mmafmcosconfig command • mmafmcosctl command • mmafmcoskeys command • mmafmctl command • mmafmlocal command • mmapplypolicy command • mmaudit command • mmauth command • mmbackup command • mmbackupconfig command • mmbuildgpl command • mmcachectl command • mmcallhome command • mmces command • mmchattr command • mmchcluster command • mmchconfig command • mmchdisk command • mmcheckquota command • mmchfileset command • mmchfs command • mmchlicense command • mmchmgr command • mmchnode command • mmchnodeclass command • mmchnsd command • mmchpolicy command • mmchpool command • mmchqos command • mmclidecode command	• System administrators of IBM Storage Scale systems • Application programmers who are experienced with IBM Storage Scale systems and familiar with the terminology and concepts in the XDSM standard

Table 1. IBM Storage Scale library information units (continued)		
Information unit	**Type of information**	**Intended users**
IBM Storage Scale: Command and Programming Reference Guide	• mmclone command • mmcloudgateway command • mmcrcluster command • mmcrfileset command • mmcrfs command • mmcrnodeclass command • mmcrnsd command • mmcrsnapshot command • mmdefedquota command • mmdefquotaoff command • mmdefquotaon command • mmdefragfs command • mmdelacl command • mmdelcallback command • mmdeldisk command • mmdelfileset command • mmdelfs command • mmdelnode command • mmdelnodeclass command • mmdelnsd command • mmdelsnapshot command • mmdf command • mmdiag command • mmdsh command • mmeditacl command • mmedquota command • mmexportfs command • mmfsck command • mmfsckx command • mmfsctl command • mmgetacl command • mmgetstate command • mmhadoopctl command • mmhdfs command • mmhealth command • mmimgbackup command • mmimgrestore command • mmimportfs command • mmkeyserv command	• System administrators of IBM Storage Scale systems • Application programmers who are experienced with IBM Storage Scale systems and familiar with the terminology and concepts in the XDSM standard

Table 1. IBM Storage Scale library information units (continued)		
Information unit	**Type of information**	**Intended users**
IBM Storage Scale: Command and Programming Reference Guide	• mmlinkfileset command • mmlsattr command • mmlscallback command • mmlscluster command • mmlsconfig command • mmlsdisk command • mmlsfileset command • mmlsfs command • mmlslicense command • mmlsmgr command • mmlsmount command • mmlsnodeclass command • mmlsnsd command • mmlspolicy command • mmlspool command • mmlsqos command • mmlsquota command • mmlssnapshot command • mmmigratefs command • mmmount command • mmnetverify command • mmnfs command • mmnsddiscover command • mmobj command • mmperfmon command • mmpmon command • mmprotocoltrace command • mmpsnap command • mmputacl command • mmqos command • mmquotaoff command • mmquotaon command • mmreclaimspace command • mmremotecluster command • mmremotefs command • mmrepquota command • mmrestoreconfig command • mmrestorefs command • mmrestrictedctl command • mmrestripefile command	• System administrators of IBM Storage Scale systems • Application programmers who are experienced with IBM Storage Scale systems and familiar with the terminology and concepts in the XDSM standard

Table 1. IBM Storage Scale library information units (continued)		
Information unit	**Type of information**	**Intended users**
IBM Storage Scale: Command and Programming Reference Guide	• mmrestripefs command • mmrpldisk command • mmsdrrestore command • mmsetquota command • mmshutdown command • mmsmb command • mmsnapdir command • mmstartup command • mmstartpolicy command • mmtracectl command • mmumount command • mmunlinkfileset command • mmuserauth command • mmwatch command • mmwinservctl command • mmxcp command • spectrumscale command **Programming reference** • IBM Storage Scale Data Management API for GPFS information • GPFS programming interfaces • GPFS user exits • IBM Storage Scale management API endpoints • Considerations for GPFS applications	• System administrators of IBM Storage Scale systems • Application programmers who are experienced with IBM Storage Scale systems and familiar with the terminology and concepts in the XDSM standard

Table 1. IBM Storage Scale library information units (continued)		
Information unit	**Type of information**	**Intended users**
IBM Storage Scale: Big Data and Analytics Guide	This guide provides the following information: Summary of changes Big data and analytics support Hadoop Scale Storage Architecture • Elastic Storage Server • Erasure Code Edition • Share Storage (SAN-based storage) • File Placement Optimizer (FPO) • Deployment model • Additional supported storage features IBM Spectrum® Scale support for Hadoop • HDFS transparency overview • Supported IBM Storage Scale storage modes • Hadoop cluster planning • CES HDFS • Non-CES HDFS • Security • Advanced features • Hadoop distribution support • Limitations and differences from native HDFS • Problem determination IBM Storage Scale Hadoop performance tuning guide • Overview • Performance overview • Hadoop Performance Planning over IBM Storage Scale • Performance guide	• System administrators of IBM Storage Scale systems • Application programmers who are experienced with IBM Storage Scale systems and familiar with the terminology and concepts in the XDSM standard
IBM Storage Scale: Big Data and Analytics Guide	Cloudera Data Platform (CDP) Private Cloud Base • Overview • Planning • Installing • Configuring • Administering • Monitoring • Upgrading • Limitations • Problem determination	• System administrators of IBM Storage Scale systems • Application programmers who are experienced with IBM Storage Scale systems and familiar with the terminology and concepts in the XDSM standard

Table 1. IBM Storage Scale library information units (continued)		
Information unit	**Type of information**	**Intended users**
IBM Storage Scale: Big Data and Analytics Guide	Cloudera HDP 3.X • Planning • Installation • Upgrading and uninstallation • Configuration • Administration • Limitations • Problem determination Open Source Apache Hadoop • Open Source Apache Hadoop without CES HDFS • Open Source Apache Hadoop with CES HDFS	• System administrators of IBM Storage Scale systems • Application programmers who are experienced with IBM Storage Scale systems and familiar with the terminology and concepts in the XDSM standard
IBM Storage Scale Erasure Code Edition Guide	IBM Storage Scale Erasure Code Edition • Summary of changes • Introduction to IBM Storage Scale Erasure Code Edition • Planning for IBM Storage Scale Erasure Code Edition • Installing IBM Storage Scale Erasure Code Edition • Uninstalling IBM Storage Scale Erasure Code Edition • Creating an IBM Storage Scale Erasure Code Edition storage environment • Using IBM Storage Scale Erasure Code Edition for data mirroring and replication • Deploying IBM Storage Scale Erasure Code Edition on VMware infrastructure • Upgrading IBM Storage Scale Erasure Code Edition • Incorporating IBM Storage Scale Erasure Code Edition in an Elastic Storage Server (ESS) cluster • Incorporating IBM Elastic Storage Server (ESS) building block in an IBM Storage Scale Erasure Code Edition cluster • Administering IBM Storage Scale Erasure Code Edition • Troubleshooting • IBM Storage Scale RAID Administration	• System administrators of IBM Storage Scale systems • Application programmers who are experienced with IBM Storage Scale systems and familiar with the terminology and concepts in the XDSM standard

Table 1. IBM Storage Scale library information units (continued)		
Information unit	**Type of information**	**Intended users**
IBM Storage Scale Container Native Storage Access	This guide provides the following information: • Overview • Planning • Installation prerequisites • Installing the IBM Storage Scale container native operator and cluster • Upgrading • Configuring IBM Storage Scale Container Storage Interface (CSI) driver • Using IBM Storage Scale GUI • Maintenance of a deployed cluster • Cleaning up the container native cluster • Monitoring • Troubleshooting • References	• System administrators of IBM Storage Scale systems • Application programmers who are experienced with IBM Storage Scale systems and familiar with the terminology and concepts in the XDSM standard
IBM Storage Scale Data Access Service	This guide provides the following information: • Overview • Architecture • Security • Planning • Installing and configuring • Upgrading • Administering • Monitoring • Collecting data for support • Troubleshooting • The **mmdas** command • REST APIs	• System administrators of IBM Storage Scale systems • Application programmers who are experienced with IBM Storage Scale systems and familiar with the terminology and concepts in the XDSM standard

Table 1. IBM Storage Scale library information units (continued)		
Information unit	**Type of information**	**Intended users**
IBM Storage Scale Container Storage Interface Driver Guide	This guide provides the following information: • Summary of changes • Introduction • Planning • Installation • Upgrading • Configurations • Using IBM Storage Scale Container Storage Interface Driver • Managing IBM Storage Scale when used with IBM Storage Scale Container Storage Interface driver • Cleanup • Limitations • Troubleshooting	• System administrators of IBM Storage Scale systems • Application programmers who are experienced with IBM Storage Scale systems and familiar with the terminology and concepts in the XDSM standard

Prerequisite and related information

For updates to this information, see IBM Storage Scale in IBM Documentation.

For the latest support information, see the IBM Storage Scale FAQ in IBM Documentation.

Conventions used in this information

Table 2 on page xxiv describes the typographic conventions used in this information. UNIX file name conventions are used throughout this information.

Note: Users of IBM Storage Scale for Windows must be aware that on Windows, UNIX-style file names need to be converted appropriately. For example, the GPFS cluster configuration data is stored in the `/var/mmfs/gen/mmsdrfs` file. On Windows, the UNIX namespace starts under the `%SystemDrive%\cygwin64` directory, so the GPFS cluster configuration data is stored in the `C:\cygwin64\var\mmfs\gen\mmsdrfs` file.

Table 2. Conventions

Convention	Usage
bold	Bold words or characters represent system elements that you must use literally, such as commands, flags, values, and selected menu options. Depending on the context, **bold** typeface sometimes represents path names, directories, or file names.
bold underlined	bold underlined keywords are defaults. These take effect if you do not specify a different keyword.
constant width	Examples and information that the system displays appear in `constant-width` typeface. Depending on the context, `constant-width` typeface sometimes represents path names, directories, or file names.

Table 2. Conventions (continued)

Convention	Usage
italic	*Italic* words or characters represent variable values that you must supply. *Italics* are also used for information unit titles, for the first use of a glossary term, and for general emphasis in text.
<*key*>	Angle brackets (less-than and greater-than) enclose the name of a key on the keyboard. For example, <Enter> refers to the key on your terminal or workstation that is labeled with the word *Enter*.

****	In command examples, a backslash indicates that the command or coding example continues on the next line. For example:
	`mkcondition -r IBM.FileSystem -e "PercentTotUsed > 90" \`
	`-E "PercentTotUsed < 85" -m p "FileSystem space used"`

{*item*}	Braces enclose a list from which you must choose an item in format and syntax descriptions.
[*item*]	Brackets enclose optional items in format and syntax descriptions.
<Ctrl-*x*>	The notation <Ctrl-*x*> indicates a control character sequence. For example, <Ctrl-c> means that you hold down the control key while pressing <c>.
***item*...**	Ellipses indicate that you can repeat the preceding item one or more times.
|	In *synopsis* statements, vertical lines separate a list of choices. In other words, a vertical line means *Or*. In the left margin of the document, vertical lines indicate technical changes to the information.

Note: CLI options that accept a list of option values delimit with a comma and no space between values. As an example, to display the state on three nodes use `mmgetstate -N` *NodeA*,*NodeB*,*NodeC*. Exceptions to this syntax are listed specifically within the command.

How to send your comments

Your feedback is important in helping us to produce accurate, high-quality information. If you have any comments about this information or any other IBM Storage Scale documentation, send your comments to the following e-mail address:

`mhvrcfs@us.ibm.com`

Include the publication title and order number, and, if applicable, the specific location of the information about which you have comments (for example, a page number or a table number).

To contact the IBM Storage Scale development organization, send your comments to the following e-mail address:

`scale@us.ibm.com`

Chapter 1. Summary of changes

Summary of changes for IBM Storage Scale Container Storage Interface driver.

The following enhancements are made in this release:

- Support for shallow copy volume
- Improvements in the script for debug data collection
- Upgrade of the containers used by the Kubernetes CSI sidecar
- Support for Kubernetes 1.28 and 1.29 and Red Hat® OpenShift® 4.15
- Support to configure resource limits of IBM Storage Scale Container Storage Interface driver

Chapter 2. Introduction

IBM Storage Scale is a clustered file system that provides concurrent access to a single file system or set of file systems from multiple nodes. The nodes can be SAN-attached, network attached, a mixture of SAN-attached and network attached, or in a shared nothing cluster configuration. This enables high-performance access to this common set of data to support a scale-out solution or to provide a high availability platform. For more information on IBM Storage Scale features, see the *Product overview* section in the *IBM Storage Scale: Concepts, Planning, and Installation Guide*.

Container Storage Interface (CSI) is a standard for exposing arbitrary block and file storage systems to containerized workloads on Container Orchestration Systems like Kubernetes. The IBM Storage Scale Container Storage Interface driver specification is defined in the CSI specification repository in the Container Storage Interface project in the GitHub.

IBM Storage Scale Container Storage Interface driver allows IBM Storage Scale to be used as a persistent storage for stateful application running in Kubernetes clusters. Through the IBM Storage Scale Container Storage Interface driver, Kubernetes persistent volumes (PVs) can be provisioned from IBM Storage Scale. Containers can essentially be used with stateful microservices such as database applications (MongoDB, PostgreSQL, and so on).

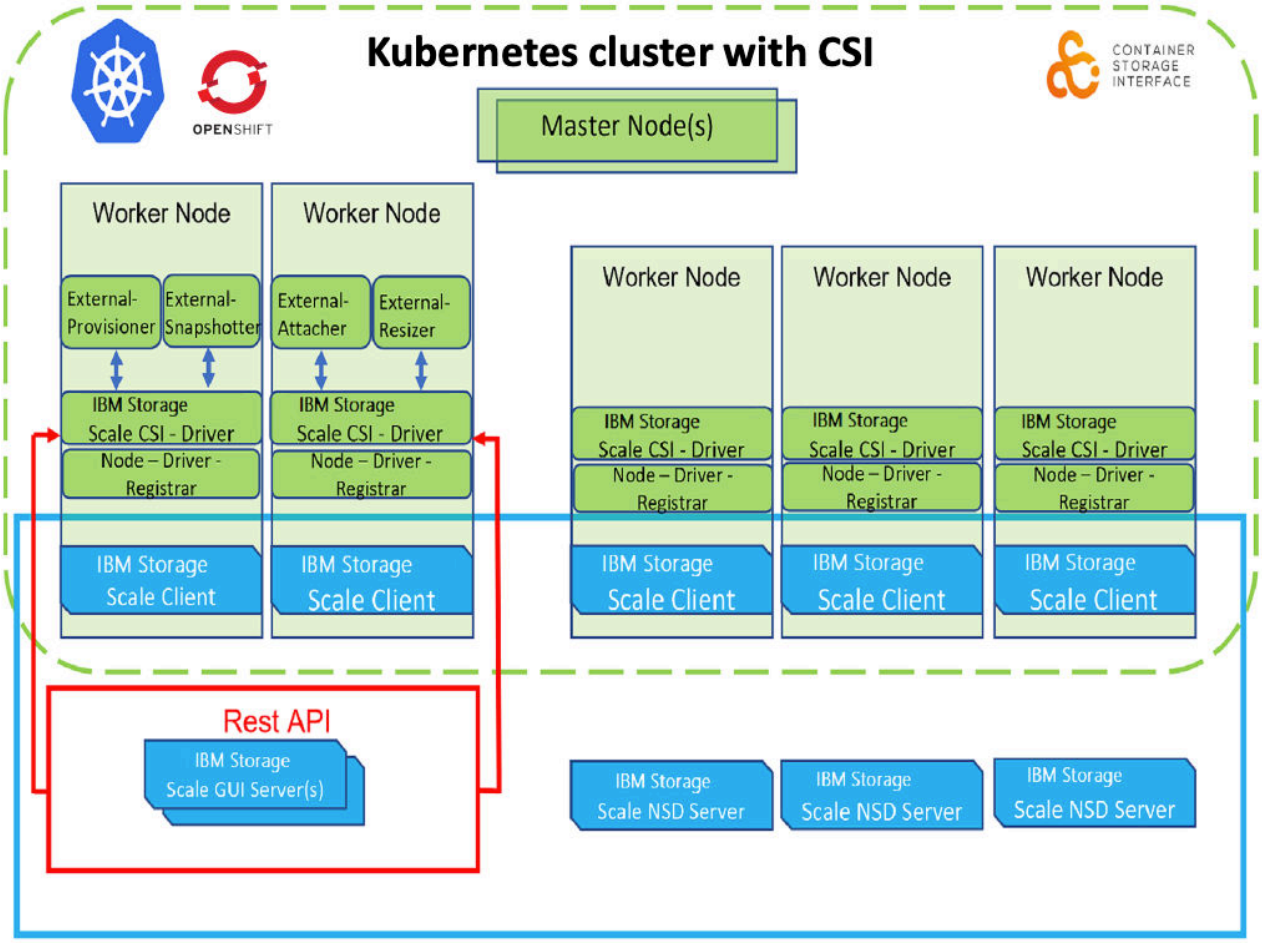

Figure 1. IBM Storage Scale Container Storage Interface Driver

IBM implements the CSI specification of storage plug-in in the following manner:

The external-provisioner, external-snapshotter, external-attacher, and external-resizer Sidecar Containers are deployments, which might be deployed on separate infrastructure hosts for resiliency.

- The external-provisioner watches for create and delete volume API calls.
- The external-snapshotter watches for create and delete volume snapshots calls.
- The external-attacher watches for mount and unmount API calls.
- The external-resizer watches the volume expansion calls.

The node-driver-registrar is a kubelet service, which runs alongside the node plug-in at the time of initialization. The IBM Storage Scale Container Storage Interface driver provides the interconnect for persistent volume mount from the container worker node to the underlying storage system and is deployed as a DaemonSet.

The IBM Storage Scale Container Storage Interface driver makes the REST API calls to the IBM Storage Scale storage system to perform storage management functions. For more information, see Deployments and DaemonSet in the Kubernetes Documentation.

Support for consistency group is available from IBM Storage Scale Container Storage Interface driver 2.5.0. For more information, see “Consistency Group (CG)” on page 39.

Features covered

The following features are available with IBM Storage Scale Container Storage Interface driver:

- Static provisioning: Ability to use existing directories and filesets as persistent volumes.
- Lightweight dynamic provisioning: Ability to create directory-based volumes dynamically.
- Fileset-based dynamic provisioning: Ability to create fileset-based volumes dynamically.
- Multiple file systems support: Ability to create volume across multiple file systems.
- Remote mount support: Ability to create volume on a remotely mounted file system.
- Operator support for easier deployment, upgrade, and cleanup.
- Supported volume access modes: RWX (ReadWriteMany) and RWO (ReadWriteOnce)
- Snapshot feature support: Ability to create a volume snapshot and to restore a snapshot into a new volume.
- Volume Expansion support: Ability to expand a dynamically provisioned volume.
- Shallow copy volume support: Ability to create a new persistent volume (ROX) from a volume snapshot.
- Support to configure resource limits of IBM Storage Scale Container Storage Interface driver.
- Volume Cloning support: Ability to create a clone of an existing volume.
- Compression support: Ability to enable compression for dynamically provisioned volumes.
- Tiering support: Ability to enable tiering for dynamically provisioned volumes.
- Consistency Group support: Ability to have group of volumes for application groups.
- fsGroup support for RWO volumes: When used, Kubernetes recursively changes the ownership and permission of volumes content to match the fsGroup specified in a pod’s `securityContext`.
- Volume stat support for fileset based volumes: Ability to show available and used capacity of fileset based volumes.
- Support for the multiple GUIs configuration (GUI HA) for a CSI driver on Vanilla Kubernetes: Ability to configure multiple GUIs when the GUI is installed on multiple nodes of a storage cluster.

Chapter 3. Planning

Describes the planning information for using IBM Storage Scale Container Storage Interface driver.

Note: If you are using a Red Hat OpenShift cluster, ensure to replace "kubectl" with "oc" in all commands.

Hardware and software requirements

Hardware and software requirements for IBM Storage Scale Container Storage Interface driver.

The following hardware and software requirements must be met for using IBM Storage Scale Container Storage Interface driver at your site:

- Red Hat Enterprise Linux CoreOS (RHCOS) support (x86_64, ppc64le, s390x architectures) when installed in combination with IBM Storage Scale Container Native storage access. For information about the instruction sets, see IBM Storage Scale Container Native Storage Access documentation.
- Red Hat OpenShift 4.13, 4.14, and 4.15 (x86_64 architecture) through Red Hat Enterprise Linux (RHEL) 8 worker nodes.
- Vanilla Kubernetes 1.27, 1.28, and 1.29 (x86_64 architecture) through RHEL 7.9 worker nodes.
- Vanilla Kubernetes 1.28 and 1.29 (x86_64 and ppc64le architectures) through RHEL 8 or RHEL 9 worker nodes with a limited support. For more information, see Appendix A, “Installing IBM Storage Scale CSI on a Kubernetes cluster with RHEL 8 or RHEL 9 nodes,” on page 67.
- IBM Storage Scale 5.1.2.1 or later.
- Ubuntu 20.04 and Ubuntu 22.04 (x86_64 architecture) with Vanilla Kubernetes.
- If you want to use multiple GUIs on a storage cluster, use IBM Storage Scale 5.1.6.1+.

Note: For IBM Storage Scale Container Storage Interface driver version 2.10 onward, the minimum supported `ppc64le` architecture is Power9®.

Table 3. CSI Features, OCP, Kubernetes, and IBM Storage Scale Compatibility Matrix

IBM Storage Scale CSI feature or parameter	IBM Storage Scale CSI level	OCP level	Kubernetes level	IBM Storage Scale level	IBM Storage Scale file system version
Volume snapshot	2.2.0+	4.7+	1.20+	5.1.1.0+	N/A
Permissions parameter in `storageClass`	2.3.0+	N/A	N/A	5.1.1.2+ **Recommended :** 5.1.2.1 or later	N/A
Volume cloning	2.4.0+	4.8+	1.22+	5.1.2.1+	N/A
Consistency group	2.5.0+	4.8+	1.21+	5.1.3.0+	N/A
Compression	2.5.0+	4.8+	1.21+	5.1.3.0+	N/A
Tiering	2.5.0+	4.8+	1.21+	5.1.3.0+	27.00
`fsGroup`	2.6.0+	N/A	N/A	5.1.1.2+	N/A
GUI HA	2.8.0+	N/A	N/A	5.1.6.1+	N/A

Table 3. CSI Features, OCP, Kubernetes, and IBM Storage Scale Compatibility Matrix (continued)

IBM Storage Scale CSI feature or parameter	IBM Storage Scale CSI level	OCP level	Kubernetes level	IBM Storage Scale level	IBM Storage Scale file system version
Shallow copy volume	2.11.0+	N/A	N/A	5.1.2.1+	N/A

Table 4. IBM Storage Scale CSI compatibility matrix

IBM Storage Scale Container Storage Interface driver version	Architecture	Noncontainerized IBM Storage Scale level for worker nodes	IBM Storage Scale level if remote cluster is used	OCP levels
2.9.0	x86, ppc64le	5.1.2.1 or later	5.1.2.1 or later	4.10, 4.11, 4.12
2.10.x	x86, ppc64le	5.1.2.1 or later	5.1.2.1 or later	4.12, 4.13, 4.14
2.11.x[1]	x86, ppc64le[2]	5.1.2.1 or later	5.1.2.1 or later	4.13, 4.14, 4.15

[1] For IBM Storage Scale Container Storage Interface driver 2.11.x, refer to Table 3 on page 5 to check features supported for various IBM Storage Scale versions.

[2] The minimum supported `ppc64le` architecture is Power9 for IBM Storage Scale Container Storage Interface driver 2.10 onward.

Table 5. Hardware requirements of IBM Storage Scale Container Storage Interface

Pods	Where deployed	Container name	CPU request	CPU limit	Memory request	Memory limit	Ephemeral storage request	Ephemeral storage limit	Description
Driver (`ibm-spectrum-scale-csi-driver-xxxxx`)	All worker nodes with scale=true label	`ibm-spectrum-scale-csi`	20mCPU	600mCPU	20Mi	600Mi	1GiB	10GiB	Driver pod allows IBM Storage Scale to be used as a persistent storage for stateful application running in Kubernetes clusters.
		`driver-registrar`	20mCPU	300mCPU	20Mi	800Mi	1GiB	5GiB	
		`livenes-probe`	20mCPU	300mCPU	20Mi	800Mi	1GiB	5GiB	

Table 5. Hardware requirements of IBM Storage Scale Container Storage Interface (continued)

Pods	Where deployed	Container name	CPU request	CPU limit	Memory request	Memory limit	Ephemeral storage request	Ephemeral storage limit	Description
Operator (ibm-spectrum-scale-csi-operator-xxxxxxxxxx-xxxxx)	Single worker node	operator	50mCPU	600mCPU	50Mi	600Mi	1GiB	5GiB	The controller runtime that manages CSI custom resources.
Attacher sidecar (ibm-spectrum-scale-csi-attacher-xxxxxxxxxx-xxxxx)	Two worker nodes with scale=true label	ibm-spectrum-scale-csi-attacher	20mCPU	300mCPU	20Mi	800Mi	1GiB	5GiB	Attacher Sidecar is the pod that runs along with the main CSI driver container responsible for attach or detach of Persistent Volume.

Table 5. Hardware requirements of IBM Storage Scale Container Storage Interface (continued)

Pods	Where deployed	Container name	CPU request	CPU limit	Memory request	Memory limit	Ephemeral storage request	Ephemeral storage limit	Description
Provisioner sidecar (ibm-spectrum-scale-csi-provisioner-xxxxxxxxxx-xxxxx)	Single worker node with scale=true label	ibm-spectrum-scale-csi-provisioner	20mCPU	300mCPU	20Mi	800Mi	1GiB	5GiB	Provisioner Sidecar is the pod that runs along with the main CSI driver container responsible for creation, deletion, or cloning of Persistent Volume.
Snapshotter sidecar (ibm-spectrum-scale-csi-snapshotter-xxxxxxxxxx-xxxxx)	Single worker node with scale=true label	ibm-spectrum-scale-csi-snapshotter	20mCPU	300mCPU	20Mi	800Mi	1GiB	5GiB	Snapshotter Sidecar is the pod that runs along with the main CSI driver container responsible for creation or deletion of Persistent Volume Snapshots.

Table 5. Hardware requirements of IBM Storage Scale Container Storage Interface (continued)									
Pods	**Where deployed**	**Container name**	**CPU request**	**CPU limit**	**Memory request**	**Memory limit**	**Ephemeral storage request**	**Ephemeral storage limit**	**Description**
Resizer sidecar (`ibm-spectrum-scale-csi-resizer-xxxxxxxxxx-xxxxx`)	Single worker node with scale=true label	`ibm-spectrum-scale-csi-resizer`	20mCPU	300mCPU	20Mi	800Mi	1GiB	5GiB	Resizer Sidecar is the pod that runs along with the main CSI driver container responsible for Expansion of Persistent Volume.

Note: For more information about resource requests and limits, see Kubernetes resource management in Kubernetes documentation.

Deployment considerations

Ensure that the following steps are completed before you deploy IBM Storage Scale Container Storage Interface driver in your cluster.

- **Worker nodes selection**: By default, Kubernetes or Red Hat OpenShift schedules the IBM Storage Scale Container Storage Interface driver pods on all worker nodes. It is essential to have IBM Storage Scale client installed on all these nodes. If you want to schedule the IBM Storage Scale Container Storage Interface driver pods only on selected worker nodes, you must label the selected nodes and use this label in node selector. For more information, see "Using the node selector" on page 30.
- **Node selection for sidecar pods**: The CSI sidecar pods can be scheduled on specific nodes by adding labels to nodes, and adding the nodeSelectors for the sidecar pods. For more information, see "Using the node selector" on page 30. IBM Storage Scale Container Storage Interface driver pod must also be scheduled on the nodes that run sidecar pods. On the Red Hat OpenShift, if the infrastructure nodes are worker nodes, schedule the sidecar pods to run on the infrastructure nodes.
- **Local file system**: If you plan to use a local file system for PVC provisioning, ensure that the IBM Storage Scale GUI is initialized and running on your IBM Storage Scale cluster.
- **Remote cluster setup**: If you plan to use a remotely mounted file system for PVC provisioning, ensure that the following setup is completed:
  - The IBM Storage Scale GUI is initialized and running on both clusters (owning cluster and accessing cluster)
  - Remote cluster details are added to the Operator configuration. For more information, see "Remote cluster support" on page 26.
- **SELinux considerations**: Different Kubernetes distributions handle the SELinux enforcing mode differently. There might be differences in terms of SELinux context that is set on files, relabeling of volumes and the process context of containers. As a prerequisite, appropriate SELinux rules must be

set up to allow IBM Storage Scale Container Storage Interface driver containers to access the required resources on host. For example, the “container_t” context needs to have access to `csi.sock` and the IBM Storage Scale file system, or the files that need access from containers need to have the “`container_file_t`” context set. Refer to audit logs for any SELinux failures and set up appropriate rules as required.

Note: If you are running IBM Storage Scale Container Storage Interface driver with IBM Storage Scale Container Native, refer the SELinux limitations.

- **Node names**: At times, it is possible that IBM Storage Scale cluster and Kubernetes or Red Hat OpenShift cluster are configured with different node names for the same host.

  1. Issue the following command to check the nodes used by the Kubernetes. Do not consider the nodes name where the IBM Storage Scale is not expected to run. For example, the master nodes.

     ```
     kubectl get nodes
     ```

  2. Check the node name used by IBM Storage Scale by issuing the following curl command against the IBM Storage Scale GUI host of primary cluster.

     ```
     curl --insecure -u '<gui_username>:<gui_username_password>' -X GET https://<gui host IP/
     name>:443/scalemgmt/v2/filesystems/<filesystemname>?fields=mount
     ```

  Note:

  – The preceding command lists the node names where the file system is mounted in the field `nodesMountedReadWrite`. The preceding command may return the long list of the nodes where the specified file system mounts. Consider the node names listed only for the Kubernetes nodes.

  – If the node names listed in step 1 are not present as is (exact string) in the node names that are listed in step 2, then configure node mapping in the Operator configuration. For more information, see “Kubernetes to IBM Storage Scale node mapping” on page 31.

- **Internet connectivity**: If your worker nodes do not have internet connectivity and access to the quay.io registry, you need to manually download the following images and upload to the local image registry.

Table 6. Image Links for IBM Storage Scale Container Storage Interface driver 2.11.1

Name	Version	Image
`ibm-spectrum-scale-csi-operator`	2.11.1	quay.io/ibm-spectrum-scale/ibm-spectrum-scale-csi-operator@sha256:c5ab8375e746233fe3370af25c4b6431742e95d04d042b4b2587002c8c3e71a6
`ibm-spectrum-scale-csi-driver`	2.11.1	quay.io/ibm-spectrum-scale/ibm-spectrum-scale-csi-driver@sha256:fb25463d85c1a81555e481118b24c30d337397a9719547a02d3a408bb645ae0f
`csi-node-driver-registrar`	2.10.0	registry.k8s.io/sig-storage/csi-node-driver-registrar@sha256:c53535af8a7f7e3164609838c4b191b42b2d81238d75c1b2a2b582ada62a9780

Table 6. Image Links for IBM Storage Scale Container Storage Interface driver 2.11.1 (continued)		
Name	**Version**	**Image**
livenessprobe	2.12.0	registry.k8s.io/sig-storage/livenessprobe@sha256:5baeb4a6d7d517434292758928bb33efc6397368cbb48c8a4cf29496abf4e987
csi-attacher	4.5.0	registry.k8s.io/sig-storage/csi-attacher@sha256:d69cc72025f7c40dae112ff989e920a3331583497c8dfb1600c5ae0e37184a29
csi-provisioner	4.0.0	registry.k8s.io/sig-storage/csi-provisioner@sha256:de79c8bbc271622eb94d2ee8689f189ea7c1cb6adac260a421980fe5eed666708
csi-snapshotter	7.0.1	registry.k8s.io/sig-storage/csi-snapshotter@sha256:1a29ab1e4ecdc33a84062cec757620d9787c28b28793202c5b78ae097c3dee27
csi-resizer	1.10.0	registry.k8s.io/sig-storage/csi-resizer@sha256:4c148bbdf883153bc72d321be4dc55c33774a6d98b2b3e0c2da6ae389149a9b7

- **Parallel volume clones and volume restore**: To increase the limit of copy jobs in parallel for the volume clones and the volume restore use the following command:

  ```
  mmxcp config --set-max-value <no. of copy jobs in parallel>
  ```

  The default limit is 10, the maximum limit is 100.
- **IBM Storage Scale services**: GUI nodes, protocol nodes, and NSD nodes are not part of the Kubernetes cluster.
- **IBM Storage Scale Container Storage Interface driver version support**: Rollback to the old versions of IBM Storage Scale Container Storage Interface driver is not supported.

Roles and personas for IBM Storage Scale Container Storage Interface driver

Describes the use of different roles, cluster roles, and levels of access that are needed to deploy a fully functional IBM Storage Scale Container Storage Interface driver in a Kubernetes or Red Hat OpenShift cluster.

Personas

A Kubernetes or Red Hat OpenShift cluster administrator is required to deploy the IBM Storage Scale Container Storage Interface driver cluster.

Operator permissions

The IBM Storage Scale Container Storage Interface driver operator is a namespace scoped operator. The operator keeps a watch on whatever the namespace that it is deployed into. As part of the operator installation process, the user deploys various role-based access control (RBAC) related YAML files. These RBAC YAML files control the operator's access to the resources within the namespace it is watching.

While the operator is running with a namespace scope, it requires access to the cluster level resources to successfully deploy. Access to the cluster level resources is handled through a cluster role that is deployed during the previously mentioned deployment of RBAC YAML files. The role and cluster role are bound to the custom `ibm-spectrum-scale-csi-operator` ServiceAccount, which the operator uses to create the IBM Storage Scale Container Storage Interface driver cluster.

Table 7. IBM-Spectrum-Scale-CSI-operator role

Resources	**Verbs**	**API Groups**
pods,persistentvolumeclaims,services,endpoints,events,configmaps,secrets,secrets/status,services/finalizers,serviceaccounts	*	-
clusterroles,clusterrolebindings	*	rbac.authorization.k8s.io
deployments,daemonsets, replicasets,statefulsets	CREATE, DELETE, GET, LIST, UPDATE, and WATCH	apps
deployments/finalizers	GET, UPDATE	apps
volumeattachments,storageclasses,csidrivers	CREATE, DELETE, GET, LIST, PATCH, UPDATE, and WATCH	storage.k8s.io
servicemonitors	GET, CREATE	monitoring.coreos.com
securitycontextconstraints	*	security.open shift.io
clusterversions	GET, LIST, and WATCH	config.openshift.io
leases	CREATE, DELETE, GET, LIST, UPDATE, and WATCH	coordination.k8s.io

Chapter 4. Installation

Install or clean up the IBM Storage Scale Container Storage Interface driver. CSI Operators are used for performing these activities.

Note: To install IBM Storage Scale Container Storage Interface driver with CNSA, see IBM Storage Scale Container Native documentation.

Performing pre-installation tasks

Complete the following tasks before you start installing the IBM Storage Scale Container Storage Interface driver:

- Install IBM Storage Scale along with the IBM Storage Scale management API (GUI) and make sure that the GUI is running on both the owning and accessing clusters that are used for IBM Storage Scale Container Storage Interface driver configuration.
  - For the supported versions, see IBM Storage Scale FAQ.
  - For more information about installing IBM Storage Scale, see the topic *Installing IBM Storage Scale on Linux nodes with the installation toolkit* in *IBM Storage Scale: Concepts, Planning, and Installation Guide*.
  - IBM Storage Scale management API (GUI) must be running all the time and must be in healthy state for proper functioning of IBM Storage Scale Container Storage Interface driver.
- Install and set up either Kubernetes or Red Hat OpenShift.
  - For supported versions, see IBM Storage Scale FAQ in the IBM Storage Scale documentation.
  - For more information about installing Kubernetes, see Install and Set Up kubectl topic in the Kubernetes documentation.
  - For more information about installing Red Hat OpenShift, see Installing and configuring OpenShift Container Platform clusters topic in the Red Hat OpenShift documentation.
- Install the IBM Storage Scale client on the required Kubernetes worker nodes and add these nodes to the IBM Storage Scale cluster.
- Mount the primary file system that is used for IBM Storage Scale configuration on all Kubernetes worker nodes and on the IBM Storage Scale GUI nodes.
- Initialize the IBM Storage Scale GUI (if not already done) either by logging in to the GUI console once or by issuing the following command on the GUI node:

  ```
  /usr/lpp/mmfs/gui/cli/initgui
  ```

- Create an IBM Storage Scale user group "CsiAdmin" if it does not exist. Issue the following command to create the CsiAdmin user:

  ```
  /usr/lpp/mmfs/gui/cli/mkusergrp CsiAdmin --role csiadmin
  ```

- Create an IBM Storage Scale user in the "CsiAdmin" group. This user must be used on IBM Storage Scale Container Storage Interface driver configuration. Issue this command on the GUI node to create the user:

  ```
  /usr/lpp/mmfs/gui/cli/mkuser <username> -p <password> -g CsiAdmin
  ```

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

  ```
  curl --insecure -u 'gui_username:gui_username_password' -X GET https://guihostname:443/
  scalemgmt/v2/cluster
  ```

  The command gives an output similar to the following text:

```
{
  "cluster" : {
    "clusterSummary" : {
      "clusterId" : 17258972170939727157,
      "clusterName" : "node10.node10",
      "primaryServer" : "node10",
      "rcpPath" : "/usr/bin/scp",
      "rcpSudoWrapper" : false,
      "repositoryType" : "CCR",
      "rshPath" : "/usr/bin/ssh",
      "rshSudoWrapper" : false,
      "uidDomain" : "node10.node10"
    },
    "capacityLicensing" : {
      "liableCapacity" : 96636764160,
      "liableNsdCount" : 2,
      "liableNsds" : [ {
        "nsdName" : "nsd1",
        "liableCapacity" : 53687091200
      }, {
        "nsdName" : "nsd2",
        "liableCapacity" : 42949672960
      } ]
    }
  },
  "status" : {
    "code" : 200,
    "message" : "The request finished successfully."
  }
}
```

- Issue the following command to set the quota value:

```
/usr/lpp/mmfs/bin/mmchfs gpfs0 -Q yes
```

- Issue the following command to verify the file system configuration.

```
/usr/lpp/mmfs/bin/mmlsfs gpfs0 --filesetdf -Q --perfileset-quota
```

```
flag                value                    description
------------------- ------------------------ -----------------------------------
 --filesetdf        yes                      Fileset df enabled?
 -Q                 user;group;fileset       Quotas accounting enabled
                    user;group;fileset       Quotas enforced
                    none                     Default quotas enabled
 --perfileset-quota no                       Per-fileset quota enforcement
```

Note: The `perfileset-quota` must be disabled with value set to `"no"`.

- Enable quota for `root` user by issuing the following command:

```
/usr/lpp/mmfs/bin/mmchconfig enforceFilesetQuotaOnRoot=yes
```

- For Red Hat OpenShift, ensure that the `controlSetxattrImmutableSELinux` parameter is set to "yes" by issuing the following command:

```
/usr/lpp/mmfs/bin/mmchconfig controlSetxattrImmutableSELinux=yes
```

- To display the correct volume size in the container, enable `filesetdf` of the file system by issuing the following command:

```
/usr/lpp/mmfs/bin/mmchfs <filesystem name> --filesetdf
```

- With the IBM Storage Scale 5.1.4 release, `inode` expansion can happen automatically.

  After this setting is enabled, the inode-limit setting that is specified on the fileset gets ignored. For more information, see mmchfs command.

  To enable the auto inode expansion, issue the following command:

```
/usr/lpp/mmfs/bin/mmchfs gpfs0 --auto-inode-limit
```

- Mount the file system that is used for IBM Storage Scale Container Storage Interface driver on the same mount point on worker nodes.
- Issue the following command to label the Kubernetes worker nodes where IBM Storage Scale client is installed and where IBM Storage Scale Container Storage Interface driver runs:

```
kubectl label node node1 scale=true --overwrite=true
```

  For more information, see “Using the node selector” on page 30.
- For Vanilla Kubernetes cluster, perform the following steps for the snapshot functions to work:

  Note: You need to perform the following steps only if the snapshot controller is not available. These steps are not required for OpenShift Container Platform 4.7 and later clusters.

  1. Install the external snapshotter CRDs:

     - ```
       kubectl apply -f https://raw.githubusercontent.com/kubernetes-csi/external-snapshotter/
       release-7.0/client/config/crd/snapshot.storage.k8s.io_volumesnapshotclasses.yaml
       ```
     - ```
       kubectl apply -f https://raw.githubusercontent.com/kubernetes-csi/external-snapshotter/
       release-7.0/client/config/crd/snapshot.storage.k8s.io_volumesnapshotcontents.yaml
       ```
     - ```
       kubectl apply -f https://raw.githubusercontent.com/kubernetes-csi/external-snapshotter/
       release-7.0/client/config/crd/snapshot.storage.k8s.io_volumesnapshots.yaml
       ```

  2. Install the snapshot controller.

     - ```
       kubectl apply -f https://raw.githubusercontent.com/kubernetes-csi/external-snapshotter/
       release-7.0/deploy/kubernetes/snapshot-controller/rbac-snapshot-controller.yaml
       ```
     - ```
       kubectl apply -f https://raw.githubusercontent.com/kubernetes-csi/external-snapshotter/
       release-7.0/deploy/kubernetes/snapshot-controller/setup-snapshot-controller.yaml
       ```

Installing IBM Storage Scale Container Storage Interface driver by using CLIs

Before you install the IBM Storage Scale Container Storage Interface driver, make sure that the prerequisites are met. For more information, see “Performing pre-installation tasks” on page 13.

Note: This procedure is applicable for both Kubernetes and Red Hat OpenShift with RHEL worker nodes. For Red Hat OpenShift, replace "kubectl" with "oc" in all the commands.

Installing IBM Storage Scale Container Storage Interface driver by using Operator involves the following phases:

1. Deploy the Operator on your cluster.
2. Use the Operator for deploying IBM Storage Scale Container Storage Interface driver.

Phase 1: Deploying the Operator

To deploy Operator on your cluster, do the following steps:

1. Create a namespace.

   ```
   kubectl create namespace ibm-spectrum-scale-csi-driver
   ```

   Note: For Red Hat OpenShift, use this command.

   ```
   oc new-project ibm-spectrum-scale-csi-driver
   ```

2. Issue the following command to download the operator manifest for CSI 2.11.1:

   ```
   curl -O https://raw.githubusercontent.com/IBM/ibm-spectrum-scale-csi/v2.11.1/generated/
   installer/ibm-spectrum-scale-csi-operator.yaml
   ```
   ```

If you are using OCP cluster with RHEL nodes, issue the following command:

```
curl -O https://raw.githubusercontent.com/IBM/ibm-spectrum-scale-csi/v2.11.1/generated/
installer/ibm-spectrum-scale-csi-operator-ocp-rhel.yaml
```

Note: To conduct an air gap upgrade, you must first download and modify the manifests as described in “Air gap installation” on page 17. Then, use those modified manifests to follow the next steps.

3. Issue the following command to apply the operator manifest to deploy the operator.

```
kubectl create -f ibm-spectrum-scale-csi-operator.yaml
```

For OCP cluster with RHEL nodes, issue the following command:

```
kubectl create -f ibm-spectrum-scale-csi-operator-ocp-rhel.yaml
```

4. Verify that the Operator is deployed, and the Operator pod is in running state.

```
kubectl get pod,deployment -n ibm-spectrum-scale-csi-driver
```

```
NAME                                                READY   STATUS    RESTARTS    AGE
pod/ibm-spectrum-scale-csi-operator-6ff9cf6979-v5g4c   1/1     Running      0        6d3h

NAME                                                READY   UP-TO-DATE  AVAILABLE AGE
deployment.apps/ibm-spectrum-scale-csi-operator        1/1        1            1     6d3h
```

Phase 2: Deploying IBM Storage Scale Container Storage Interface driver

Now that the Operator is up and running, you must access the Operator's API and request a deployment by using the *CSIScaleOperator* custom resource.

Do the following steps:

1. Create a secret with IBM Storage Scale GUI server’s credentials in the `ibm-spectrum-scale-csi-driver` namespace. For more information, see “Secrets” on page 23.

   Note: If you are using a remote cluster setup, then create a secret object for the GUI server of each cluster.
2. Modify the parameters in the file to suit your environment. For more information, see “Operator” on page 23.
3. Apply the custom resource file to deploy IBM Storage Scale Container Storage Interface driver:

```
kubectl apply -f csiscaleoperators.csi.ibm.com_cr.yaml
```

4. Verify that the IBM Storage Scale Container Storage Interface driver is installed, Operator and driver resources are ready, and pods are in running state.

```
kubectl get pod,daemonset,deployment -n ibm-spectrum-scale-csi-driver
```

```
NAME                                                READY   STATUS    RESTARTS   AGE
pod/ibm-spectrum-scale-csi-9tvlj                    3/3     Running   0          5d7h
pod/ibm-spectrum-scale-csi-attacher-66879bb7f9-f25h4    1/1     Running   3(10h ago) 5d8h
pod/ibm-spectrum-scale-csi-attacher-66879bb7f9-ggnlv    1/1     Running   1(5d7h ago)5d8h
pod/ibm-spectrum-scale-csi-bvc7p                    3/3     Running   0          5d7h
pod/ibm-spectrum-scale-csi-operator-df7ddcf8d-54pdg     1/1     Running   0(36s ago) 5d7h
pod/ibm-spectrum-scale-csi-provisioner-59b777f96d-xppxs 1/1     Running   3(10h ago) 5d7h
pod/ibm-spectrum-scale-csi-resizer-6d854f78bd-r29tc     1/1     Running   3(10h ago) 5d7h
pod/ibm-spectrum-scale-csi-snapshotter-6fbb5cf945-q8fsq 1/1     Running   3(10h ago) 5d7h


NAME                                   DESIRED   CURRENT   READY   UP-TO-DATE   AVAILABLE
NODE SELECTOR   AGE
daemonset.apps/ibm-spectrum-scale-csi   2         2         2       2            2
scale=true      5d22h
```

```
NAME                                                READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/ibm-spectrum-scale-csi-attacher     2/2     2            2           5d22h
deployment.apps/ibm-spectrum-scale-csi-operator     1/1     1            1           7d
deployment.apps/ibm-spectrum-scale-csi-provisioner 1/1     1            1           5d22h
deployment.apps/ibm-spectrum-scale-csi-resizer      1/1     1            1           5d22h
deployment.apps/ibm-spectrum-scale-csi-snapshotter 1/1     1            1           5d22h
```

For more information, see IBM Storage Scale Container Storage Interface (CSI) project in the IBM GitHub repository.

Air gap installation

Installation or upgrade of the IBM Storage Scale Container Storage Interface driver via an air gap is performed for clusters that are in a restricted network environment. The following steps are applicable for Vanilla Kubernetes and Red Hat OpenShift with RHEL worker nodes.

Before you install the IBM Storage Scale Container Storage Interface driver, make sure that the prerequisites are met. For more information, see “Performing pre-installation tasks” on page 13.

The following steps are needed for air gap installation or upgrade:

1. Download all the CSI images that are mentioned in Table 1 of Deployment Considerations and upload those CSI images to private registry.
2. Download the operator manifest for CSI 2.11.1:

   ```
   curl -O https://raw.githubusercontent.com/IBM/ibm-spectrum-scale-csi/v2.11.1/generated/
   installer/ibm-spectrum-scale-csi-operator.yaml
   ```

   If you are using a Red Hat OpenShift Container Platform cluster with RHEL nodes, issue the following command:

   ```
   curl -O https://raw.githubusercontent.com/IBM/ibm-spectrum-scale-csi/v2.11.1/generated/
   installer/ibm-spectrum-scale-csi-operator-ocp-rhel.yaml
   ```

3. Download the following sample of a custom resource file on your cluster.

   ```
   curl -O https://raw.githubusercontent.com/IBM/ibm-spectrum-scale-csi/v2.11.1/operator/config/
   samples/csiscaleoperators.csi.ibm.com_cr.yaml
   ```

4. Update `ibm-spectrum-scale-csi-operator.yaml or ibm-spectrum-scale-csi-operator-ocp-rhel.yaml` (if you are using Red Hat OpenShift Container Platform with RHEL worker nodes) to use offline images.
   - Update the operator image location.
   - Update an environment variable CSI_DRIVER_IMAGE.
   - Add environment variables:
     - CSI_SNAPSHOTTER_IMAGE
     - CSI_ATTACHER_IMAGE
     - CSI_PROVISIONER_IMAGE
     - CSI_LIVENESSPROBE_IMAGE
     - CSI_NODE_REGISTRAR_IMAGE
     - CSI_RESIZER_IMAGE

   ```
   containers:

   - name: operator
   image: <CSI Operator Image>

   env:
   - name: CSI_DRIVER_IMAGE
   value: <CSI Driver Image>
   - name: CSI_SNAPSHOTTER_IMAGE
   value: <CSI Snapshotter Image>
   - name: CSI_ATTACHER_IMAGE
   ```

```
value: <CSI Attacher Image>
- name: CSI_PROVISIONER_IMAGE
value: <CSI Provisioner Image>
- name: CSI_LIVENESSPROBE_IMAGE
value: <CSI Livenessprobe Image>
- name: CSI_NODE_REGISTRAR_IMAGE
value: <CSI node registrar Image>
- name: CSI_RESIZER_IMAGE
value: <CSI Resizer Image>
```

5. If the images are being stored in a private image registry, `imagePullSecret` is required to pull an image from that image registry. Create an `imagePullSecret` with the `ibm-spectrum-scale-csi-registry` key name in the `ibm-spectrum-scale-csi-driver` namespace. For more information about creating an `imagePullSecret`, see Pull an Image from a Private Registry in the Kubernetes documentation.
6. After the configurations are done, install or upgrade CSI. Follow the instructions for CSI installation or CSI upgrade.

Chapter 5. Upgrading

You can upgrade the IBM Storage Scale Container Storage Interface driver to use the enhanced feature.

IBM Storage Scale CSI 2.10.x to 2.11.x

You can upgrade IBM Storage Scale Container Storage Interface driver 2.10.x to 2.11.x to use the new updates.

Do the following steps to upgrade IBM Storage Scale Container Storage Interface driver 2.10.x to version 2.11.x:

1. Download the Operator manifest file on your cluster by issuing the following command:

   ```
   curl -O https://raw.githubusercontent.com/IBM/ibm-spectrum-scale-csi/v2.11.1/generated/
   installer/ibm-spectrum-scale-csi-operator.yaml
   ```

   Note: To conduct an air gap upgrade, you must first download and modify the manifests as described in “Air gap installation” on page 17. Then, use those modified manifests to follow the next steps.
2. Apply the manifest file by issuing the following command:

   ```
   kubectl apply -f ibm-spectrum-scale-csi-operator.yaml
   ```

   The preceding step upgrades both the Operator and the IBM Storage Scale Container Storage Interface driver. The Operator and the pods are restarted with the upgraded image.
3. Verify that the pods are back in the running state by issuing the following command:

   ```
   kubectl get pod -n ibm-spectrum-scale-csi-driver
   ```

   The system displays the following output:

   ```
   NAME                                                READY   STATUS    RESTARTS   AGE
   ibm-spectrum-scale-csi-7kzzn                        3/3     Running   0          3m47s
   ibm-spectrum-scale-csi-attacher-0                   1/1     Running   0          3m47s
   ibm-spectrum-scale-csi-attacher-1                   1/1     Running   0          3m43s
   ibm-spectrum-scale-csi-operator-8947b76cb-k8ggx     1/1     Running   0          3m52s
   ibm-spectrum-scale-csi-provisioner-0                1/1     Running   0          3m47s
   ibm-spectrum-scale-csi-resizer-0                    1/1     Running   0          3m47s
   ibm-spectrum-scale-csi-s85bn                        3/3     Running   0          3m47s
   ibm-spectrum-scale-csi-snapshotter-0                1/1     Running   0          3m47s
   ```
4. Verify that the Operator and the pods are using the upgraded images by issuing the following command:

   Note: If the `CustomResource` is updated to use custom image names, then the `CustomResource` must be updated to use new version of the images. Users are not recommended to use the custom images in the `CustomResource`.

   ```
   kubectl describe pod ibm-spectrum-scale-csi-operator-8947b76cb-k8ggx -n ibm-spectrum-scale-
   csi-driver |\
    grep "Image:" | grep ibm-spectrum-scale

       Image:         quay.io/ibm-spectrum-scale/ibm-spectrum-scale-csi-
   operator@sha256:c5ab8375e746233fe3370af25c4b6431742e95d04d042b4b2587002c8c3e71a6
   ```

   ```
   kubectl describe pod ibm-spectrum-scale-csi-7kzzn -n ibm-spectrum-scale-csi-driver |\
   grep "Image:" | grep ibm-spectrum-scale

       Image:         quay.io/ibm-spectrum-scale/ibm-spectrum-scale-csi-
   driver@sha256:fb25463d85c1a81555e481118b24c30d337397a9719547a02d3a408bb645ae0f
   ```

Handling upgrade failure

If IBM Storage Scale Container Storage Interface driver upgrade fails, you can revert to the earlier IBM Storage Scale Container Storage Interface driver version. First, uninstall the current IBM Storage Scale Container Storage Interface driver version and reinstall the earlier version.

Note:

- The new objects that are created on a newer IBM Storage Scale Container Storage Interface driver version might not work on an older version.
- In a few cases, the `CustomResource` file needs to be adjusted based on the IBM Storage Scale Container Storage Interface driver version that is installed.

Air gap upgrade

Installation or upgrade of the IBM Storage Scale Container Storage Interface driver via an air gap is performed for clusters that are in a restricted network environment. The following steps are applicable for Vanilla Kubernetes and Red Hat OpenShift with RHEL worker nodes.

Before you upgrade the IBM Storage Scale Container Storage Interface driver, make sure that the prerequisites are met. For more information, see “Performing pre-installation tasks” on page 13.

The following steps are needed for air gap installation or upgrade:

1. Download all the CSI images that are mentioned in Table 1 of Deployment Considerations and upload those CSI images to private registry.
2. Download the operator manifest for CSI 2.11.1:

   ```
   curl -O https://raw.githubusercontent.com/IBM/ibm-spectrum-scale-csi/v2.11.1/generated/
   installer/ibm-spectrum-scale-csi-operator.yaml
   ```

   If you are using a Red Hat OpenShift Container Platform cluster with RHEL nodes, issue the following command:

   ```
   curl -O https://raw.githubusercontent.com/IBM/ibm-spectrum-scale-csi/v2.11.1/generated/
   installer/ibm-spectrum-scale-csi-operator-ocp-rhel.yaml
   ```

3. Download the following sample of a custom resource file on your cluster.

   ```
   curl -O https://raw.githubusercontent.com/IBM/ibm-spectrum-scale-csi/v2.11.1/operator/config/
   samples/csiscaleoperators.csi.ibm.com_cr.yaml
   ```

4. Update `ibm-spectrum-scale-csi-operator.yaml or ibm-spectrum-scale-csi-operator-ocp-rhel.yaml` (if you are using Red Hat OpenShift Container Platform with RHEL worker nodes) to use offline images.
   - Update the operator image location.
   - Update an environment variable CSI_DRIVER_IMAGE.
   - Add environment variables:
     - CSI_SNAPSHOTTER_IMAGE
     - CSI_ATTACHER_IMAGE
     - CSI_PROVISIONER_IMAGE
     - CSI_LIVENESSPROBE_IMAGE
     - CSI_NODE_REGISTRAR_IMAGE
     - CSI_RESIZER_IMAGE

   ```
   containers:

   - name: operator
   image: <CSI Operator Image>

   env:
   - name: CSI_DRIVER_IMAGE
   ```

```
value: <CSI Driver Image>
- name: CSI_SNAPSHOTTER_IMAGE
value: <CSI Snapshotter Image>
- name: CSI_ATTACHER_IMAGE
value: <CSI Attacher Image>
- name: CSI_PROVISIONER_IMAGE
value: <CSI Provisioner Image>
- name: CSI_LIVENESSPROBE_IMAGE
value: <CSI Livenessprobe Image>
- name: CSI_NODE_REGISTRAR_IMAGE
value: <CSI node registrar Image>
- name: CSI_RESIZER_IMAGE
value: <CSI Resizer Image>
```

5. If the images are being stored in a private image registry, `imagePullSecret` is required to pull an image from that image registry. Create an `imagePullSecret` with the `ibm-spectrum-scale-csi-registry` key name in the `ibm-spectrum-scale-csi-driver` namespace. For more information about creating an `imagePullSecret`, see Pull an Image from a Private Registry in the Kubernetes documentation.
6. After the configurations are done, install or upgrade CSI. Follow the instructions for CSI installation or CSI upgrade.

Chapter 6. Configurations

You can configure the IBM Storage Scale Container Storage Interface driver at your site.

IBM Storage Scale Container Storage Interface driver configurations

During IBM Storage Scale Container Storage Interface driver plug-in deployment, the parameters that are required for communication with IBM Storage Scale must be configured in Kubernetes' Secrets.

Secrets

Secret is needed to store credentials to connect to IBM Storage Scale REST API server. The GUI user must have *csiadmin* role.

Perform the following steps:

1. Create a secret in the CSI namespace by issuing the following command:

```
kubectl create secret generic [secret_name] --from-literal=username=[gui_username] --from-
literal=password=[gui_password] -n ibm-spectrum-scale-csi-driver
```

2. Apply the CSI product label to the secret by issuing the following command:

```
kubectl label secret [secret_name] product=ibm-spectrum-scale-csi -n ibm-spectrum-scale-csi-
driver
```

Certificates

For secure SSL mode, a CA certificate must be specified. This certificate is used in SSL communication with the IBM Storage Scale GUI server. The certificate must be created as a `ConfigMap`. There must be as many ConfigMaps as the number of clusters with secure SSL enabled.

`ConfigMap` command syntax:

```
kubectl create configmap <name of configmap> --from-file=<same value provided as name of
configmap>=/path/to/mycertificate.pem -n ibm-spectrum-scale-csi-driver
```

For example:

```
kubectl create configmap storage-cacert --from-file=storage-cacert=/path/to/mycertificate.pem
-n ibm-spectrum-scale-csi-driver
```

Note:

- Configmap name and `--from-file` value must match and this `--from-file` value must be used as "`cacert`" value in the Operator.
- Specifying different CA signed certificate for GUI host is not supported while configuring the GUI High Availability feature in the CSI driver custom resource.

Operator

You can define the configuration parameters that are needed for creating a CSIScaleOperator custom resource that is used to configure the IBM Storage Scale Container Storage Interface driver.

For more information, see a sample CSIScaleOperator custom resource configuration YAML file.

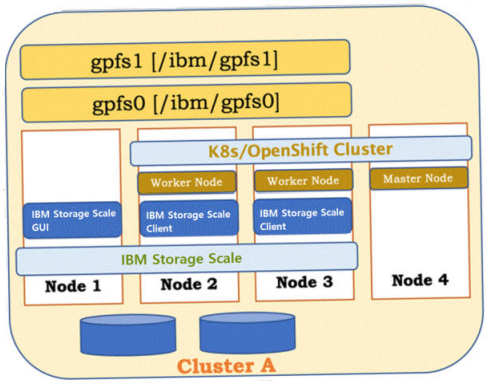

Figure 2. Operator configuration

- Primary cluster: IBM Storage Scale cluster where some or all of the client nodes are also worker nodes of Red Hat OpenShift or Kubernetes cluster. The aim of running IBM Storage Scale on worker node is to provide persistent storage from IBM Storage Scale to the application that is running on Kubernetes or Red Hat OpenShift.
- Primary file system: One of the existing IBM Storage Scale file systems from the primary cluster must be designated as the primary file system. One fileset from this file system is used by the IBM Storage Scale Container Storage Interface driver internally to store the volume references. This fileset is referred to as the primary fileset. For proper functioning of IBM Storage Scale Container Storage Interface driver, the primary file system must be mounted on all worker nodes all the time.

The CSIScaleOperator custom resource for a sample deployment looks like the following sample shows. There are two file systems **gpfs0** and **gpfs1**. For this deployment, **gpfs0** was chosen as the PrimaryFs.

csiscaleoperators.csi.ibm.com_cr.yaml file

```
---
apiVersion: csi.ibm.com/v1
kind: "CSIScaleOperator"
metadata:
  name: "ibm-spectrum-scale-csi"
  namespace: "ibm-spectrum-scale-csi-driver"
  labels:
    app.kubernetes.io/name: ibm-spectrum-scale-csi-operator
    app.kubernetes.io/instance: ibm-spectrum-scale-csi-operator
    app.kubernetes.io/managed-by: ibm-spectrum-scale-csi-operator
    release: ibm-spectrum-scale-csi-operator
status: {}
spec:
   clusters:
    - id: "<cluster id of IBM Storage Scale running on node1,node2,node3>"
      secrets: "guisecret"
      secureSslMode: false
      primary:
        primaryFs: "gpfs0"
      restApi:
        - guiHost: "<FQDN/IP of GUI Node 1>"
        #- guiHost: "<FQDN/IP of GUI Node 2>" #Optional - Multiple GUI nodes can be specified
if the storage cluster has GUI installed on multiple nodes.
    attacherNodeSelector:
    - key: "scale"
      value: "true"
  provisionerNodeSelector:
    - key: "scale"
      value: "true"
  pluginNodeSelector:
    - key: "scale"
      value: "true"
  snapshotterNodeSelector:
    - key: "scale"
      value: "true"
  resizerNodeSelector:
    - key: "scale"
```

```
      value: "true"
---
```

Table 8. CSIScaleOperator configuration parameter description

Parameter	Usage	Description
id	Mandatory	Cluster ID of the primary IBM Storage Scale cluster. For more information, see *mmlscluster command* in the *IBM Storage Scale: Concepts, Planning, and Installation Guide*.
primaryFs	Mandatory	Primary file system name.
primaryFset	Optional	Primary fileset name. This will be created if the fileset does not exist. Default value: **spectrum-scale-csi-volume-store**
inodeLimit	Optional	Inode limit for the primary fileset. If not specified, fileset is created with 1 M inodes, which is the IBM Storage Scale default.
cacert	Mandatory if *secureSslMode* is true.	Name of the pre-created CA certificate configmap that is used to connect to the GUI server (running on the "guiHost"). For more information, see “Certificates” on page 23.
secrets	Mandatory	Name of the pre-created Secret containing username and password that are used to connect to the GUI server for the cluster specified against the **id** parameter. For more information, see “Secrets” on page 23.
guiHost	Mandatory	FQDN or IP address of the GUI node of the IBM Storage Scale cluster that is specified against the **id** parameter. Optionally, multiple GUI hosts can be specified for a storage cluster with multiple GUIs. If multiple guiHosts are specified, use the same port number for all the GUIs on a storage cluster.
imagePullSecrets	Optional	An array of imagePullSecrets to be used for pulling images from a private registry. This parameter is a pass-through option that distributes the imagePullSecrets array to the containers generated by the Operator. For more information about creating imagePullSecrets, see *Pull an Image from a Private Registry* in the Kubernetes documentation.[1]
kubeletRootDirPath	Optional	The kubelet root directory path is used in case the Kubernetes setup uses a nondefault kubelet root directory path. The default kubelet's root directory is at `/var/lib/kubelet`.

[1]Do not update CR to create an `imagePullSecret` array with the `ibm-spectrum-scale-csi-registrykey` name because the `ibm-spectrum-scale-csi-registrykey` name is used as a default secret internally.

For a deployment that involves two or more IBM Storage Scale clusters, see “Remote cluster support” on page 26.

Status

To check the status of the `csiScaleOperator` resource, issue the following command:

```
kubectl get csiscaleoperator -n ibm-spectrum-scale-csi-driver -ojson | jq -r '.items[0].status'
```

Output:

```
{
   "conditions": [
```

```
    {
      "lastTransitionTime": "2024-02-22T02:54:08Z",
      "message": "The CSI driver resources have been created/updated successfully",
      "reason": "CSIConfigured",
      "status": "True",
      "type": "Success"
    }
],
"versions": [
    {
      "name": "ibm-spectrum-scale-csi",
      "version": "2.11.1"
    }
 ]
}
```

Table 9. Output status description

Condition type	Condition status	Description
Success	TRUE	The operator created all the resources in the cluster required by the CSI driver.
Success	FALSE	An error occurred while the operator was creating the resources required by the CSI driver.
Success	Unknown	The initial status when the operator reconciliation is in-progress.

Note:

- If `status.condition.status` is *False*, look for `status.condition.reason` and `status.condition.message` to identify the cause of an error. For more information, see operator logs.
- From IBM Storage Scale Container Storage Interface driver 2.9.0 onward, after an instance of the custom resource CSIScaleOperator is created, the change in primary stanza of the instance is not allowed as changing the primary stanza causes loss of access to version 1 volumes. If you do not need access to these volumes, you can delete CSIScaleOperator instance and create a new instance with the required primary stanza.
- If you change any value in a cluster stanza of a CSIScaleOperator instance, the changed value is passed from the operator to the driver only if it is valid. For an invalid value, you can see corresponding error message in the status of CSIScaleOperator instance.

Remote cluster support

IBM Storage Scale provides a feature to mount IBM Storage Scale file systems from one IBM Storage Scale cluster (owning cluster) to another IBM Storage Scale cluster (accessing cluster). You can configure an IBM Storage Scale Container Storage Interface driver to work with a remotely mounted IBM Storage Scale.

The cluster that owns the file system is responsible for administering the file system and granting access to other clusters on a per-cluster basis. After access to a file system is granted to nodes in another IBM Storage Scale cluster, the nodes can mount the file system and do data operations as if the file systems are locally owned.

For more information about the remote mount setup, see *Accessing a remote GPFS file system* in the *IBM Storage Scale: Command and Programming Reference Guide*.

Note: Remote mount setup must be done before you configure IBM Storage Scale Container Storage Interface driver.

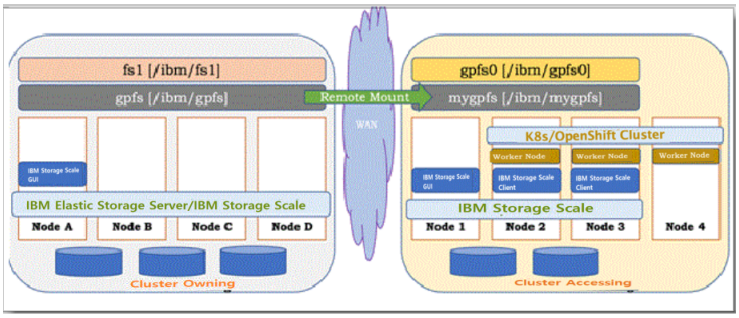

Figure 3. Deployment of two IBM Storage Scale clusters with remote-mounted file systems

The primary cluster is the IBM Storage Scale cluster where Red Hat OpenShift or Kubernetes worker nodes coexist with IBM Storage Scale client nodes. In this example deployment, cluster A is designated as the primary cluster.

The cluster O is another IBM Storage Scale cluster that has two file systems `gpfs` and `fs1`. The file system `gpfs` is mounted on Cluster A as file system `mygpfs` while file system `fs1` is not exposed to Cluster A.

For each IBM Storage Scale cluster, cluster entry must be added under the **clusters** section of the custom resource.

```
- id: "<cluster id of IBM Storage Scale>"
    restApi:
     - guiHost: "<FQDN/IP of GUI Node 1>"
     #- guiHost: "<FQDN/IP of GUI Node 2>" #Optional
  secrets: "<secret name for GUI of IBM Storage Scale cluster>"
  secureSslMode: false
```

One IBM Storage Scale cluster must be the primary cluster for IBM Storage Scale Container Storage Interface driver deployment. Primary cluster is marked by adding the **primary** section in the respective cluster entry. In the example described in the figure, deployment Cluster A is the primary cluster.

An example of an entry for a primary cluster is shown:

```
- id: "<cluster id of IBM Storage Scale Cluster which is Primary cluster >"
    primary:
    primaryFs: <name of primary filesystem>
    restApi:
      - guiHost: "<FQDN/IP of GUI Node 1>"
     #- guiHost: "<FQDN/IP of GUI Node 2>" #Optional
    secrets: "<secret name for GUI of Primary IBM Storage Scale cluster >"
    secureSslMode: false
```

In the deployment example, there are two IBM Storage Scale clusters, hence two entries of clusters are added, one for the primary cluster (Cluster A) and another one for cluster O (Owning cluster).

The custom resource configuration slightly changes based on whether the primary file system is locally owned (`gpfs0` in the example deployment) or remotely mounted (`mygpfs` in the example deployment). The changes are in the `primary` section of the primary cluster entry.

The custom resource for the example deployment when primaryFS is a locally owned file system (`gpfs0`) appears as shown:

```
---
apiVersion: csi.ibm.com/v1
kind: "CSIScaleOperator"
metadata:
  name: "ibm-spectrum-scale-csi"
  namespace: "ibm-spectrum-scale-csi-driver"
  labels:
    app.kubernetes.io/name: ibm-spectrum-scale-csi-operator
    app.kubernetes.io/instance: ibm-spectrum-scale-csi-operator
    app.kubernetes.io/managed-by: ibm-spectrum-scale-csi-operator
```

```
    release: ibm-spectrum-scale-csi-operator
status: {}
spec:

  clusters:
    - id: "<cluster id of IBM Storage Scale Cluster A>"
      secrets: "guisecret"
      secureSslMode: false
      primary:
        primaryFs: "gpfs0"
      restApi:
        - guiHost: "<FQDN/IP of GUI Node 1>"
        #- guiHost: "<FQDN/IP of GUI Node 2>" #Optional - Multiple GUI nodes can be specified
if the storage cluster has GUI installed on multiple nodes.

    - id: "<cluster id of IBM Storage Scale Cluster O>"
      secrets: "remoteguisecret"
      secureSslMode: false
      restApi:
        - guiHost: "<FQDN/IP of GUI Node A>" # Multiple GUIs can be provided here also similar
to primary cluster.
  attacherNodeSelector:
    - key: "scale"
      value: "true"
  provisionerNodeSelector:
    - key: "scale"
      value: "true"
  pluginNodeSelector:
    - key: "scale"
      value: "true"
  snapshotterNodeSelector:
    - key: "scale"
      value: "true"
  resizerNodeSelector:
    - key: "scale"
      value: "true"
---
```

The custom resource, for example, deployment when primaryFs is a remotely mounted file system (mygpfs) appears as shown:

```
---
apiVersion: csi.ibm.com/v1
kind: "CSIScaleOperator"
metadata:
  name: "ibm-spectrum-scale-csi"
  namespace: "ibm-spectrum-scale-csi-driver"
  labels:
    app.kubernetes.io/name: ibm-spectrum-scale-csi-operator
    app.kubernetes.io/instance: ibm-spectrum-scale-csi-operator
    app.kubernetes.io/managed-by: ibm-spectrum-scale-csi-operator
    release: ibm-spectrum-scale-csi-operator
status: {}
spec:

  clusters:
    - id: "<cluster id of IBM Storage Scale Cluster A>"
      secrets: "guisecret"
      secureSslMode: false
      primary:
        primaryFs: "mygpfs"
    remoteCluster: "<cluster id of IBM Storage Scale Cluster O (Owning cluster)>"
      restApi:
        - guiHost: "<FQDN/IP of GUI Node 1>"
    - id: "<cluster id of IBM Storage Scale Cluster O(owning cluster)>"
      secrets: "remoteguisecret"
      secureSslMode: false
      restApi:
        - guiHost: "<FQDN/IP of GUI Node A>" # Multiple GUIs can be provided here also similar
to primary cluster.
  attacherNodeSelector:
    - key: "scale"
      value: "true"
  provisionerNodeSelector:
    - key: "scale"
      value: "true"
  pluginNodeSelector:
    - key: "scale"
      value: "true"
  snapshotterNodeSelector:
```

```
    - key: "scale"
      value: "true"
  resizerNodeSelector:
    - key: "scale"
      value: "true"
---
```

Table 10. Parameter description

Parameter name	Status	Parameter description
id	Mandatory	Cluster ID of IBM Storage Scale cluster. For more information, see *mmlscluster* in the *IBM Storage Scale: Command and Programming Reference Guide*.
primaryFs	Mandatory if cluster is primary.	Name of the primary file system on the primary cluster.
primaryFset	Optional	Primary fileset name. Fileset of the specified name is created if it does not exist. Default: `spectrum-scale-csi-volume-store`
remoteCluster	Mandatory if the primary file system (PrimaryFS) is a remotely mounted file system.	Cluster ID of the remote cluster, who is the owner of the file system that is specified against **primaryFs**.
inodeLimit	Optional	Inode limit for the primary fileset. If not specified, fileset is created with 1 M inodes, which is the IBM Storage Scale default value.
cacert	Mandatory if *secureSslMode* is true.	Name of the pre-created CA certificate configmap that is used to connect to the GUI server that is running on the **guiHost**. For more information, see “Certificates” on page 23.
secrets	Mandatory	Name of the pre-created Secret containing username and password to connect to the GUI running on the guiHost for cluster specified against the **id** parameter. For more information, see “Secrets” on page 23.
guiHost	Mandatory	FQDN or IP address of the GUI node of the IBM Storage Scale cluster that is specified against the **id** parameter. Optionally, multiple GUI hosts can be specified for a storage cluster with multiple GUIs. If multiple guiHosts are specified, use the same port number for all the GUIs on a storage cluster.

Table 10. Parameter description (continued)		
Parameter name	**Status**	**Parameter description**
imagePullSecrets	Optional	An array of imagePullSecrets to be used for pulling images from a private registry. This parameter is a pass-through option that distributes the imagePullSecrets array to the containers generated by the Operator. For more information about creating imagePullSecrets, see *Pull an Image from a Private Registry* in the Kubernetes documentation.

Note:

- Owning cluster might have more than one file system and not all file systems need to be remotely mounted on the accessing cluster.
- There can be more than one owning cluster that exposes their file systems to the accessing cluster.
- Accessing cluster or primary cluster can be compute-only cluster without any of its own file system.
- Secrets contain the credentials to connect to the GUI for a specified cluster. For each cluster in the custom resource, there should be a pre-created secret before Operator deployment. For more information, see "Secrets" on page 23. Same secret cannot be used for multiple clusters even if the credentials are same.
- Custom resource also contains other parameters that are optional, so those parameters should be added according to your requirement.

Using the node selector

By default, the IBM Storage Scale Container Storage Interface driver gets deployed on all worker nodes. The node selector controls the Kubernetes worker nodes on which the IBM Storage Scale Container Storage Interface driver must run. It helps in cases where new worker nodes are added to Kubernetes cluster but do not have IBM Storage Scale installed. It helps in ensuring that sidecar pods are running on the desired nodes.

To configure node selector, perform the following steps:

1. Label the Kubernetes worker nodes where sidecar pods should run, as shown in the following example:

   - ```
     kubectl label node node1 infranode=1 --overwrite=true
     ```
   - ```
     kubectl label node node2 infranode=2 --overwrite=true
     ```

   Note:

   - Use specific labels like the one for attacher and provisioner sidecar pods, only if there is a requirement of running these sidecar pods for very specific nodes. Otherwise, use single label like `scale=true` for running sidecar pods and IBM Storage Scale Container Storage Interface driver DaemonSet.
   - Nodes marked for running sidecar pods must be a subset of the nodes marked with the `scale=true` label.

2. Label the Kubernetes worker nodes where IBM Storage Scale Container Storage Interface driver must run, as shown:

```
kubectl label node node1 scale=true --overwrite=true
```

3. Configure the following parameters in the Operator custom resource (`csiscaleoperators.csi.ibm.com_cr.yaml`) under the "spec" section. IBM Storage Scale client must be installed and running on the nodes that have the `scale=true` label.

```
attacherNodeSelector:
   - key: "scale"
     value: "true"
#  - key: "infranode"       # Only if there is requirement of running Attacher
#    value: "2"             # on specific Node

provisionerNodeSelector:
   - key: "scale"
     value: "true"
#  - key: "infranode".      # Only if there is requirement of running Provisioner
#    value: "1"             # on specific Node

snapshotterNodeSelector:
    - key: "scale"
      value: "true"
#   - key: "infranode"      # Only if there is requirement of running Snapshotter
#     value: "2"            # on specific Node

pluginNodeSelector:
   - key: "scale"
     value: "true"

resizerNodeSelector:
    - key: "scale"
      value: "true"
#   - key: "infranode"      # Only if there is requirement of running Resizer
#     value: "2"            # on specific Node
```

Note: If you choose to run IBM Storage Scale Container Storage Interface driver on selective nodes using the **`nodeSelector`**, then make sure that the pod using IBM Storage Scale Container Storage Interface driver PVC is scheduled on the nodes where IBM Storage Scale Container Storage Interface driver is running.

Kubernetes to IBM Storage Scale node mapping

In some environments, Kubernetes node names might be different from the IBM Storage Scale node names. This difference results in failure during mounting of pods. Kubernetes node to IBM Storage Scale node mapping must be configured to address this condition during the Operator configuration.

To configure this, add "`nodeMapping`" section under "spec" in the `csiscaleoperators.csi.ibm.com_cr.yaml`, as shown in the following example:

```
nodeMapping:
   - k8sNode: "kubernetesNode1"
     spectrumscaleNode: "scaleNode1"
   - k8sNode: "kubernetesNode2"
     spectrumscaleNode: "scaleNode2"
```

If Kubernetes node name starts with a number, then add node mapping for such nodes in the following format:

```
    - k8sNode: "K8sNodePrefix_<Kubernetes Node Name/ID>"
      spectrumscaleNode: "<Spectrum Scale Node Name/ID>"
```

For example, if Kubernetes node name is 198.51.100.10, then use the following node mapping:

```
    - k8sNode: "K8sNodePrefix_198.51.100.10"
      spectrumscaleNode: "spectrumscalenode11"
```

Note:

- Kubernetes node name is listed by issuing the `kubectl get nodes` command.
- You can list the IBM Storage Scale node name by issuing the following command. Look for the field `nodesMountedReadWrite`.

```
curl --insecure -u '<gui_username>:<gui_username_password>' -X GET <https://<gui> host IP/
name>:443/scalemgmt/v2/filesystems/<filesystemname>?fields=mount
```

- All entries for nodes that differ in name must be added. Consider only nodes where IBM Storage Scale CSI is expected to run along with IBM Storage Scale.

Tolerations

Tolerations are applied to pods, and allow (but do not require) the pods to be scheduled on nodes with matching taints.

Taints and tolerations work together to ensure that pods are not scheduled onto inappropriate nodes. For more information, see Taints and Tolerations in the Kubernetes documentation.

To allow the IBM Storage Scale Container Storage Interface driver pods to be scheduled on nodes with taints, configure the CSIScaleOperator custom resource `csiscaleoperators.csi.ibm.com_cr.yaml` under "spec" section as shown in the following example:

```
tolerations:
     - key: "key1"  # Node taint key name. Mandatory
       operator: "Equal”   # Valid values are "Exists" and "Equal". Mandatory
       value: "value1”  # Required if operator is "Equal"
       effect: "NoExecute”  # Valid values are "NoSchedule", "PreferNoSchedule" and
"NoExecute”. An empty effect matches all effects with given key. Mandatory
       tolerationSeconds: 3600  # Used only when effect is "NoExecute". It determines how long
the pod will stay bound to the node after the taint is added.
```

Changing the configuration after deployment

IBM Storage Scale Container Storage Interface driver configuration can be changed after the driver is deployed. Any change in the configuration post deployment reinitializes IBM Storage Scale Container Storage Interface driver.

Updating a Secret

The IBM Storage Scale Container Storage Interface driver uses secrets to store API authentication. If the password is expired or you want to change the password, you must update the secret in Kubernetes.

To update the secret and have the operator apply it, do the following steps:

1. Delete the old secret.

```
kubectl delete secret -n ibm-spectrum-scale-csi-driver [secret_name]
```

2. Change the password of the GUI user on the IBM Storage Scale cluster.
3. Create a secret with new credentials and apply the required labels.

```
kubectl create secret generic [secret_name] --from-literal=username=[gui_username] --from-
literal=password=[new_gui_password] -n ibm-spectrum-scale-csi-driver

kubectl label secret [secret_name] product=ibm-spectrum-scale-csi app.kubernetes.io/name=ibm-
spectrum-scale-csi-operator -n ibm-spectrum-scale-csi-driver
```

It is mandatory to label the secret issuing `kubectl label` command to trigger reconciliation. The process can then be monitored in the Operator logs.

Also, if the Operator's custom resource was deployed before the secrets were created, the previous process can be used to start the operator without deleting the Custom Resource.

Cluster Details

To change cluster details such as *guiHost*, remote cluster information or node mapping, edit the **CSIScaleOperator** by using the following command.

```
kubectl edit CSIScaleOperator ibm-spectrum-scale-csi  -n  ibm-spectrum-scale-csi-driver
```

When this command is issued, a `vi` editor opens up, which contains a temporary YAML file with the contents for *CSIScaleOperator* object. You must update the cluster details, save the file, and exit. The Operator restarts the IBM Storage Scale Container Storage Interface driver with the new configuration.

Note: The parameter does not work after you edit the CSIScaleOperator. To make the parameter work, you must delete the custom resource, update the parameter in the custom resource, and re-create the custom resource.

Advanced configuration

The IBM Storage Scale Container Storage Interface driver supports some advanced configurations that can be applied by using an optional ConfigMap. Any change in the ConfigMap after the driver is deployed reinitializes the IBM Storage Scale Container Storage Interface driver.

1. Create an optional ConfigMap.

   An optional ConfigMap is used to manage advanced features of IBM Storage Scale Container Storage Interface driver. Use the following YAML file to manage the features.

   ```
   kind: ConfigMap

   apiVersion: v1

   metadata:
     name: ibm-spectrum-scale-csi-config
     namespace: ibm-spectrum-scale-csi-driver

   data:
     VAR_DRIVER_LOGLEVEL: TRACE
     VAR_DRIVER_PERSISTENT_LOG: ENABLED
     DRIVER_CPU_LIMITS: "600m"
     DRIVER_MEMORY_LIMITS: "700Mi"
     SIDECAR_CPU_LIMITS: "401m"
     SIDECAR_MEMORY_LIMITS: "801Mi"
   ```

2. Apply the ConfigMap.

   ```
   kubectl apply -f <ConfigMap file name>
   ```

3. Update the existing ConfigMap.

   ```
   kubectl edit configmap ibm-spectrum-scale-csi-config -n <namespace>
   ```

   a. Update the value to the corresponding field and save the file to update the modified value in IBM Storage Scale Container Storage Interface driver.

   b. To use the default configuration, delete the applied ConfigMap by using the following command.

   ```
   kubectl delete -f <ConfigMap file name>
   ```

4. Update the logger level of IBM Storage Scale Container Storage Interface driver by changing the **VAR_DRIVER_LOGLEVEL** parameter of optional ConfigMap. The driver uses six levels of logger: *TRACE*, *DEBUG*, *INFO*, *WARNING*, *ERROR*, and *FATAL*. The default logger level of IBM Storage Scale Container Storage Interface driver would be *INFO* if a wrong value or no value is provided in the optional configuration map.
5. Resource limits of IBM Storage Scale Container Storage Interface driver can be configured with the higher limits if the user notices that the pods are being stopped due to an out of memory (OOM) error, which is reported as `OOMKilled` or `status as OOMKilled`. The new values cannot be lower than the required resource limits. For more information, see Table 3.

Update *DRIVER_CPU_LIMITS* to configure the CPU limit of the driver container, and *DRIVER_MEMORY_LIMITS* to update the memory limit of the same container.

Update *SIDECAR_CPU_LIMITS* to configure the CPU limits, and *SIDECAR_MEMORY_LIMITS* to configure the memory limit of other containers which includes `driver-registrar`, `liveness-probe`, `Attacher`, `Snapshotter`, `Resizer`, and `Provisioner`.

Chapter 7. Using IBM Storage Scale Container Storage Interface driver

You can create storage volumes such as PVCs and PVs to suit your requirements.

Storage class

Storage class is used for creating lightweight volumes, fileset-based volumes, and consistency group volumes.

Storage class for creating lightweight volumes

Create lightweight volumes by using storage class.

The configuration is as follows:

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: ibm-spectrum-scale-csi-lt
provisioner: spectrumscale.csi.ibm.com
parameters:
  volBackendFs: "gpfs0"
  volDirBasePath: "pvfileset/lwdir"
reclaimPolicy: Delete
```

The following fields come under the parameters section of storageClass. Parameters section is mandatory for IBM Storage Scale CSI driver storageClass:

Field name	Description
volBackendFs	The file system on which the directory-based volume must be created.
volDirBasePath	The base path under which all volumes with this storage class is created. This path must exist. The relative path from the file system mount point.
uid (optional)	The username of the directory. The uid/gid must exist on the IBM Storage Scale GUI node.
gid (optional)	The group name of the directory. The uid/gid must exist on the IBM Storage Scale GUI node.
shared (optional)	Use `shared=true`, if you have pods with non-root users that are using PVCs with `ReadWriteMany`(RWX) access mode. Default value is false.
nodeClass (optional)	The user can specify a pre-existing `nodeClass` in `storageClass`, so that the pre-existing `nodeClass` is used when a volume is created from an existing volume or snapshot. If a `nodeClass` is specified, only the node belonging to it participates in data copy. **Note:** For more information about `nodeClass`, see IBM Storage Scale `nodeClass` command.

Note:

- Since lightweight volume does not enforce quota, it can grow beyond defined size, which may result in consuming whole file system. To avoid this, you must manually create or use an existing fileset to host the lightweight PVC volumes. This can be done by specifying the directory inside fileset for `volDirBasePath` option.
- Do not use filesets that are created by the IBM Storage Scale Container Storage Interface driver for hosting lightweight volumes. When fileset-based volumes get deleted, all the data under the fileset, including lightweight PVC if created, also gets deleted.

Storage class for creating fileset-based volumes

Create PVCs under a file system that is owned by the primary cluster or a file system that is owned by a different cluster other than the primary cluster.

To create the fileset-based volumes, use the storageClass details that are provided in the following example.

In this example, it is assumed that the user wants to create PVCs under the file system, **gpfs0**, as in the sample deployment explained in “Remote cluster support” on page 26. The same storageClass format is applicable for the sample deployment that is explained in “Operator” on page 23.

Independent fileset storage class:

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: ibm-spectrum-scale-csi-fileset
provisioner: spectrumscale.csi.ibm.com
parameters:
  volBackendFs: gpfs0
  uid: "1000"
  gid: "1000"
  nodeClass: gpfs_siteA
reclaimPolicy: Delete
```

Dependent fileset storage class:

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: ibm-spectrum-scale-csi-fileset-dependent
provisioner: spectrumscale.csi.ibm.com
parameters:
  volBackendFs: "gpfs0"
  uid: "1000"
  gid: "1000"
  filesetType: "dependent"
  parentFileset: "independent-fileset-fset1"
reclaimPolicy: Delete
```

The following fields are available for the "parameters:" section in a storageClass manifest. The "parameters:" section is a mandatory section for the IBM Storage Scale Container Storage Interface driver storageClass:

Field name	Description
volBackendFs	The name of the file system under which the fileset should be created. When a file system that is owned by a different cluster, the file system name is the name of the remotely mounted file system on the primary cluster.
clusterId (optional)	The Cluster ID of the owning cluster for the remote files system.

Field name	Description
	The Cluster ID of the primary cluster for local file system.
uid (optional)	The username of the fileset. The uid/gid must exist on the IBM Storage Scale GUI node of accessing and owning clusters. The default value is "root".
gid (optional)	The group name of the fileset. The gid/group name must exist on the IBM Storage Scale GUI node of the accessing and owning clusters. The default value is "root".
filesetType	Valid options are "independent", "dependent". The default value is "independent".
parentFileset	Parent fileset name. Valid with filesetType=dependent. Default value is "root".
inodeLimit (optional)	Inode limit for fileset. Valid with `filesetType=independent`. If not specified, `inodeLimit` is calculated by using this formula: volume size/block size of the file system. **Note:** With IBM Storage Scale 5.1.4 and later, auto inode expansion can be used instead of `inodeLimit`. For more information, see the mmchfs command.
shared (optional)	Use `shared="true"`, if you have pods with non-root users that are using PVCs with `ReadWriteMany`(RWX) access mode. Default value is "false".
nodeClass (optional)	The user can specify a pre-existing `nodeClass` in `storageClass`, so that the pre-existing `nodeClass` is used when a volume is created from an existing volume or snapshot. If a `nodeClass` is specified, only the node belonging to it participates in data copy. **Note:** For more information about `nodeClass`, see IBM Storage Scale `nodeClass` command.

Storage class for creating consistency group volumes

Create PVCs in a consistency group on a file system that is owned by the primary cluster or a file system that is owned by a different cluster other than the primary cluster.

To create the consistency group volumes, use the storageClass details that are provided in the following example.

In this example, it is assumed that you want to create PVCs under the file system, **gpfs0**, as in the sample deployment explained in “Remote cluster support” on page 26. The same storageClass format is applicable for the sample deployment that is explained in “Operator” on page 23.

Consistency group storage class:

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: ibm-spectrum-scale-csi-consistency-group
provisioner: spectrumscale.csi.ibm.com
```

```
parameters:
    version: "2"
    volBackendFs: gpfs0
reclaimPolicy: Delete
allowVolumeExpansion: true
```

The following fields come under the parameters section of storageClass. Parameters section is mandatory for IBM Storage Scale CSI driver storageClass:

Field name	Description
version	Represents the version of the storage class. • Value "1" indicates that the storage class is not for a consistency group. • Value "2" indicates that the storage class is for a consistency group. **Note:** The default value is "1".
volBackendFs (mandatory)	The name of the file system under which the fileset must be created. File system name is the name of the remotely mounted file system on the primary cluster.
clusterId (optional)	The Cluster ID of the owning cluster for remote files system. The Cluster ID of the primary cluster for local file system.
uid (optional)	The username of the fileset. The uid/gid must exist on the IBM Storage Scale GUI node of accessing and owning clusters. The default value is "root".
gid (optional)	The group name of the fileset. The gid/group name must exist on the IBM Storage Scale GUI node of the accessing and owning clusters. The default value is "root".
inodeLimit (optional)	Inode limit for consistency group. If not specified, **`inodeLimit`** is set to 1M. **Note:** With IBM Storage Scale 5.1.4 and later, auto inode expansion can be used instead of `inodeLimit`. For more information, see the mmchfs command.
shared (optional)	Use `shared=true`, if you have pods with non-root users that are using PVCs with `ReadWriteMany`(RWX) access mode. Default value is false.
compression (optional)	Specifies whether the compression is enabled. Allowed values are: • true • false The default value is "false".
tier (optional)	Name of the tier or pool where the volume data is to be placed.

Field name	Description
	Note: Tiering feature requires IBM Storage Scale file system 27.00 or later.
nodeClass (optional)	The user can specify a pre-existing `nodeClass` in `storageClass`, so that the pre-existing `nodeClass` is used when a volume is created from an existing volume or snapshot. If a `nodeClass` is specified, only the node belonging to it participates in data copy. **Note:** For more information about `nodeClass`, see IBM Storage Scale `nodeClass` command.

Note: For using the consistency group feature, you must use IBM Storage Scale 5.1.3.0 or later.

Consistency Group (CG)

Group a list of volumes that are backed-up and restored in a consistent manner.

Application data protection scenarios (backup, restore, and disaster recovery) need to ensure that the data is being backed-up or replicated in a consistent way. An application can consist of one or more groups, which can require the data within group to be consistently backed up and restored. The group where the data needed to be consistent is referred to as Consistency Group (CG). In the context of IBM Storage Scale Container Storage Interface driver or IBM Storage Scale, CG is represented as a list of volumes that are backed up and restored in a consistent way.

The consistency of volumes that belong to the same CG is achieved by taking the snapshot of all at a single point. In IBM Storage Scale, the snapshot of volumes refers volumes to the same snapshot of their parent.

The CG of a volume is decided at the time of volume creation. CG is defined at namespace level, indicating that all the PVs created in the same namespace belong to the same CG.

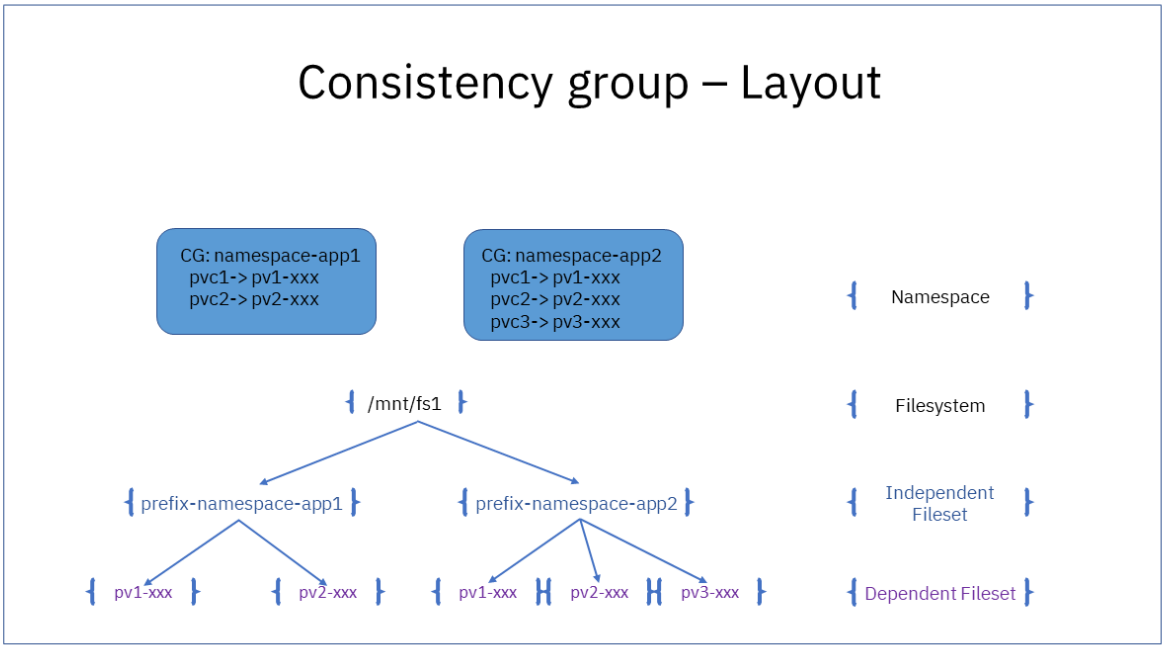

Figure 4. Consistency Group Layout

Volume Creation and Deletion

A special parameter (**`version`**=*2*) is introduced in storageClass to differentiate volumes for CG and classic volumes.

When PVC is created for CG (*namespace*) in Kubernetes, IBM Storage Scale Container Storage Interface driver will create an independent fileset on IBM Storage Scale representing the consistency group, if not already available. IBM Storage Scale Container Storage Interface driver will then create the dependent fileset within the independent fileset with quota equal to size of requested PVC.

If the PVC that belongs to the CG is deleted, IBM Storage Scale Container Storage Interface driver will delete the dependent fileset that represents the PVC. If it is the last PVC in the CG, then the independent fileset is also deleted.

You can have volumes from different `storageClasses` into the same CG, provided `storageClass` has version 2 and same `volBackendFs`.

Volume Snapshot Creation and Deletion

Volume snapshot plays a main role in maintaining the consistency of a CG. The consistency of a CG is achieved by taking a snapshot of the independent fileset instead of the dependent fileset presenting PVC.

If you invoke the snapshot from Kubernetes for a particular PVC that belongs to a CG, IBM Storage Scale Container Storage Interface driver will take the snapshot of the parent, which is an independent fileset representing a CG.

You are expected to create a volume snapshot for all the PVCs that belong to a CG in a short span of time. It is essential to create Kubernetes snapshot for all the PVCs that belong to a CG because the snapshot details are used to restore or create the PVC from the snapshot.

IBM Storage Scale Container Storage Interface driver has introduced a new parameter that is called **`snapWindow`** in `volumeSnapshotClass`. If you request the snapshot of all volumes that belong to a CG, it results in a single snapshot on an independent fileset on IBM Storage Scale. The first snapshot request results in an actual snapshot on IBM Storage Scale while subsequent snapshots will be no-op and returned successfully with reference to successful independent fileset snapshot.

The next snapshot of independent fileset is taken if there is a snapshot request from Kubernetes, and no snapshot of independent fileset taken within the time specified in `volumeSnapshotClass`. Otherwise, the snapshot request from Kubernetes keeps referring to the previous independent fileset snapshot.

The snapshot in IBM Storage Scale is also deleted when you delete all the snapshots that are referring it in Kubernetes.

You must verify that all the snapshots created for a CG that require consistency are referring to same IBM Storage Scale snapshot.

Other features like volume expansion, volume cloning, tearing support, compression support works in the same way as with classic volumes.

Note:

- There is no mechanism to define CG other than **`namespace`**.
- You cannot move volume in or out of CG, but it is possible to delete volume from CG.
- You can have maximum of 998 CG per file system if you are using IBM Storage Scale 5.1.3.0 or earlier.
- You can have maximum of 2998 CG per file system if you are using IBM Storage Scale 5.1.4.0 or later.
- Avoid having **`snapWindow`** for less than 30 Minutes.
- Last volumes deletion in CG fails if there are **`VolumeSnapshot`** present in CG.
- Avoid using multiple **`volumeSnapshotClass`** with different **`snapWindow`** for snapshots of the same CG.

- Avoid creating volumes or snapshots for the same CG with a combination of reclaim policy delete and retain. If you create the volume or snapshot with **reclaimPolicy** as *retain* then the independent fileset or snapshot might remain in the system forever and will require the human intervention.

Dynamic provisioning

Administrators use dynamic volume provisioning to create storage volumes on-demand.

Do the following steps:

1. Create a traditional storageClass or consistency-group-based storageClass. For more information, see “Storage class” on page 35.
2. Apply the following configuration:

```
kubectl apply -f storageclass.yaml
```

3. Use this storageClass to create a persistent volume claim (PVC), as shown in the following example:

```
# cat pvc.yaml

apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: scale-fset-pvc
spec:
  accessModes:
  - ReadWriteMany
  resources:
    requests:
      storage: 1Gi
  storageClassName: [name_of_your_storageclass]
```

   Modify the PVC name, storage, and storageClassName values according to your requirement.
4. Create a PVC by issuing the following command:

```
kubectl apply -f pvc.yaml
```

Creating pods

To configure a pod, do the following steps:

1. Create a manifest file (`pod.yaml`) with pod definition referencing the persistent volume claim (PVC). Following is an example of a pod definition for creating a nginx container by using a previously created PVC:

```
# cat pod.yaml

apiVersion: v1
kind: Pod
metadata:
  name: csi-scale-staticdemo-pod
  labels:
    app: nginx
spec:
  containers:
   - name: web-server
     image: nginx
     volumeMounts:
       - name: mypvc
         mountPath: /usr/share/nginx/html/scale
     ports:
     - containerPort: 80
  volumes:
   - name: mypvc
     persistentVolumeClaim:
       claimName: [pvc name]
       readOnly: false
```

Note: `claimName` is the PVC name to be used by pod for persistent storage. The *readOnly* flag can be set to true, in which case the pod mounts the PVC in the read-only mode.

2. Issue the following command to create the pod:

```
kubectl apply -f pod.yaml
```

For more information about pods, see Configure a Pod to Use a PersistentVolume for Storage in the Kubernetes documentation.

Considerations for mounting read-write many (RWX) volumes

Red Hat OpenShift relabels the volumes while mounting them inside a pod. In case of volumes with RWX access mode, when multiple pods mount them, they are relabeled multiple times. If the SELinux labeling is not managed, it might cause data access issues from the pod.

Note: If you are running IBM Storage Scale Container Storage Interface driver with IBM Storage Scale Container Native 5.1.7, you need not to do anything for the data access issue. For more information, see the Limitations section.

You can address this issue in either of the following ways:

- Use an SCC (Security Context Constraints) with `seLinuxContext.type` set as "MustRunAs".

```
seLinuxContext:
     type: MustRunAs
```

  Ensure that correct SCC is used by the pod:

```
metadata:
     annotations:
       openshift.io/scc: <scc_name>
```

- Specify appropriate `seLinuxOptions.level` in the deployment specification of the pod as shown in the following example:

```
securityContext:
    seLinuxOptions:
      level: <SELinux level label>
```

  An example of SELinux level label is "s0:c123,c456".

Access PVC by using a non-root user

All the persistent volumes are accessible only by the root user by default. To access persistent volumes through non-root users, add fsGroup in application pod spec.

Use the following snippet to add `fsGroup` in `PodSecurityContext` section of a Pod:

```
spec:
   securityContext:
     fsGroup: 5000
   containers:
```

When fsGroup is specified in Pod Spec, it means that the specified group with ID 5000 is associated with all containers in the pod. When fsGroup is specified, Kubernetes recursively change the ownership of volume content to group specified against fsGroup. Also, the owning GID will be that of fsGroup, and the setuid bit is set to that new files created in the volume that are owned by owning GID.

For more information, see Kubernetes documentation.

Volume Snapshot

The Volume Snapshot feature is used to take a point-in-time snapshot of the IBM Storage Scale Container Storage Interface driver volume. The volume snapshot feature also provides the capability of creating a new IBM Storage Scale Container Storage Interface driver volume from the existing snapshot.

Create a VolumeSnapshot

VolumeSnapshot creates a point-in-time snapshot of the independent fileset based IBM Storage Scale Container Storage Interface driver volume on the IBM Storage Scale storage system.

Create a `VolumeSnapshotClass`

`VolumeSnapshotClass` is like a `StorageClass` that defines driver-specific attributes for the snapshot to be created.

A sample `VolumeSnapshotClass` is created as shown in the following example:

```
# cat volumesnapshotclass.yaml
apiVersion: snapshot.storage.k8s.io/v1
kind: VolumeSnapshotClass
metadata:
  name: ibm-spectrum-scale-snapshot-class-consistency-group
driver: spectrumscale.csi.ibm.com
deletionPolicy: Delete
#parameters:
#        snapWindow: "30"
```

`snapWindow` parameter is valid for only consistency group. It indicates how long a snapshot stays valid for a consistency group after a snapshot is taken. The value specified should indicate snapWindow time in minutes. Default value is "30" minutes.

Note:

- **`snapWindow`** must not be less than 30 minutes while taking snapshots of multiple volumes.
- If there are multiple requests, only one snapshot for consistency group must be taken within the specified **`snapWindow`**.
- **`snapWindow`** time starts when snapshot of any volume that belongs to the consistency group is taken either for the first time or any time after **`snapWindow`** is passed.
- If one needs to take a snapshot of consistency group, request for snapshot for all volumes that belong to the consistency group must be created in a short span of time.

Create a VolumeSnapshot

VolumeSnapshot is a copy of a volume content on a storage system.

Specify the source volume to be used for creating snapshot here as shown in the following sample manifest. Source persistent volume claim (PVC) must be in the same namespace in which the snapshot is being created. Snapshots can be created only from independent fileset-based PVCs or for consistency group-based PVCs.

A sample `VolumeSnapshot` is created as shown in the following example:

```
# cat volumesnapshot.yaml
apiVersion: snapshot.storage.k8s.io/v1
kind: VolumeSnapshot
metadata:
  name: ibm-spectrum-scale-snapshot
spec:
  volumeSnapshotClassName: ibm-spectrum-scale-snapshot-class
  source:
    persistentVolumeClaimName: ibm-spectrum-scale-pvc
```

Verify that snapshot is created

Use the following steps to ensure creation of volume snapshot.

Ensure that snapshot is in the READYTOUSE state and a corresponding file set snapshot must be available on IBM Storage Scale.

- To get the status of volume snapshot, issue the following command.

```
# kubectl get volumesnapshot
NAME    READYTOUSE   SOURCEPVC   SOURCESNAPSHOTCONTENT   RESTORESIZE   SNAPSHOTCLASS
SNAPSHOTCONTENT                                    CREATIONTIME   AGE
ibm-spectrum-scale-snapshot   true         ibm-spectrum-scale-
pvc                           1Gi             ibm-spectrum-scale-snapshot-class
snapcontent-2b478910-28d1-4c29-8e12-556149095094   2d23h          2d23h
```

Note:

- The volume size of the source PVC is used as the restore size of the snapshot. Any volume that is created from this snapshot must be of the same or larger capacity.
- Volume Snapshot is supported only for the independent fileset based PVCs.

Create a volume from a source snapshot

Use the following steps to create a volume from a source snapshot.

Ensure that the source snapshot is in the same namespace as the volume that is created. Volume capacity must be greater than or equal to the source snapshot's restore size. Resultant PVC must contain data from "ibm-spectrum-scale-snapshot".

```
# cat pvcfromsnapshot.yaml
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
   name: ibm-spectrum-scale-pvc-from-snap
spec:
   accessModes:
   - ReadWriteMany
   resources:
      requests:
         storage: 1Gi
   storageClassName: ibm-spectrum-scale-storageclass
   dataSource:
      name: ibm-spectrum-scale-snapshot
      kind: VolumeSnapshot
      apiGroup: snapshot.storage.k8s.io
```

Issue the following command to create a PVC from snapshot.

```
kubectl apply -f pvcfromsnapshot.yaml
```

Restriction:

- Snapshot and new volume must be from file systems that belong to the same cluster.
- Restoring snapshot to lightweight PVC of a remotely mounted file system is not supported.

Create a shallow copy volume from a source snapshot (read-only)

Use the following steps to create a shallow copy volume from a source snapshot (read-only).

Cloning a snapshot for read-only purposes is an expensive operation. Shallow copy volume is primarily used for backup purposes where data is exclusively needed in read-only mode.

Ensure that the source snapshot is in the same namespace as the shallow copy volume that is created. Volume capacity must be greater than or equal to the restore size of the source snapshot. The resultant PVC must contain the path of the snapshot.

Tip: To create a shallow copy volume, it is recommended to use the same storage class as the one used for source PVC.

```
# cat pvcfromsnapshot.yaml
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
   name: ibm-spectrum-scale-pvc-from-snap
spec:
   accessModes:
   - ReadOnlyMany
   resources:
      requests:
         storage: 1Gi
   storageClassName: ibm-spectrum-scale-storageclass
   dataSource:
      name: ibm-spectrum-scale-snapshot
      kind: VolumeSnapshot
      apiGroup: snapshot.storage.k8s.io
```

Issue the following command to create a PVC from the snapshot:

```
kubectl apply -f pvcfromsnapshot.yaml
```

Restriction:

- Snapshot and shallow copy volume must be from file systems that belong to the same cluster.
- Restoring snapshot (shallow copy volume) to lightweight PVC is not supported.
- Restoring snapshot (shallow copy volume) across file systems from different IBM Storage Scale clusters is not supported.
- Restoring snapshot (shallow copy volume) between version 1 and version 2 `storageClass` (or between version 2 and version 1 `storageClass`) is not supported.
- Restoring snapshot (shallow copy volume) is not supported for static snapshots or snapshots that are made from versions previous to CSI 2.5.
- Volume expansion is not supported for shallow copy volume.
- Volume stat is not supported for shallow copy volume.
- Shallow copy volume is not supported for `fsgroup` and `subdir`.
- Shallow copy volume is not supported where Security-Enhanced Linux (SELinux) is enabled.

Volume cloning

Clone your volume to duplicate an existing persistent volume at a particular point-in-time. For data consistency, restrain input and output operations while cloning.

To create a clone, expand the existing `dataSource` field in the `PersistentVolumeClaim` object. So that, field accepts the name of an existing `PersistentVolumeClaim` in the same namespace.

Example of `PersistentVolumeClaim` for cloning:

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
    name: clone-of-scale-pvc
spec:
  accessModes:
  - ReadWriteOnce
  storageClassName: [name_of_your_storageclass]
  resources:
    requests:
      storage: 5Gi
  dataSource:
    kind: PersistentVolumeClaim
    name: scale-pvc
```

Note:

- The storage capacity of the cloned volume must be the same or larger than the capacity of the source volume.
- The destination persistent volume claim (PVC) must exist in the same namespace as the source PVC.
- The source PVC must be bound and available but must not be in use.

Restriction:

- Volume cloning is supported only for dynamically provisioned volumes (with or without consistency group).
- Volume cloning of lightweight volumes on remote file systems is not supported.
- Volume cloning across file systems from different IBM Storage Scale clusters is not supported.
- Volume cloning between version 1 and version 2 storageClass is not supported or vice versa.
- Volume cloning between lightweight volumes and fileset based volumes or vice versa is not supported.

Volume Expansion

Expand the capacity of dynamically provisioned volumes to meet the growing storage needs of your environment.

To enable the expansion of a volume, you must set the `allowVolumeExpansion` parameter to `true` in the StorageClass.

Example of StorageClass:

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: ibm-spectrum-scale-csi-fileset-expansion
provisioner: spectrumscale.csi.ibm.com
parameters:
  volBackendFs: gpfs0
  clusterId: "17797813605352210071"
reclaimPolicy: Delete
allowVolumeExpansion: true
```

To expand volumes where volume expansion is enabled, edit the size of the PVC.

The following example illustrates how to use the `patch` command to expand a volume to 2Gi:

```
kubectl patch pvc scale-pvc  -p '{"spec":{"resources":{"requests":{"storage":"2Gi"}}}}'
```

If the volume is created using IBM Storage Scale Container Storage Interface driver 2.3.0 or earlier, the "allowVolumeExpansion" is unset in StorageClass. You must patch the storage class as given below, before PVC expansion.

```
kubectl patch storageclass ibm-spectrum-scale-csi-fileset  -p '{"allowVolumeExpansion": true}'
```

Available: Volume expansion is supported only for dynamically provisioned volumes (with or without consistency group).

Restriction: Volume shrinking is not supported.

Static provisioning

In static provisioning, an administrator creates a number of persistent volumes (PVs), which include information about the storage that is available to each user in the cluster.

To use the existing volume on the storage system, do the following steps:

1. Create a persistent volume using the PV manifest file. For more information, see “Creating a persistent volume (PV)” on page 47.

2. Create a persistent volume claim (PVC) using the PVC manifest file. For more information, see “Creating a PersistentVolumeClaim (PVC)” on page 49.

Note: Static provisioning is not supported for consistency group feature.

Related concepts

“Creating pods” on page 41

Generating static provisioning manifests

To generate static provisioning manifests (PV and PVC), run the following script:

```
generate_static_provisioning_yamls.sh
```

You can issue the following command to download the script for CSI 2.11.1:

```
curl -O https://raw.githubusercontent.com/IBM/ibm-spectrum-scale-csi/v2.11.1/tools/
generate_static_provisioning_yamls.sh
```

Note: This script must be run on the IBM Storage Scale cluster node.

Usage of the script is as follows:

```
Usage: ./generate_static_provisioning_yamls.sh
              -f|--filesystem <Name of Volume's Source Filesystem>
              -l|--path <full Path of Volume in Primary Filesystem>
              -F|--fileset <name of source fileset>
              -s|--size <size in GB>
              -u|--username <Username of IBM Storage Scale GUI user account>
              -p|--password <Password of IBM Storage Scale GUI user account>
              -r|--guihost <HostName(or route) used to access IBM Storage Scale GUI service
running on Primary Cluster>
             [-P|--pvname <name for pv>]
             [-c|--storageclass <StorageClass for pv>]
             [-a|--accessmode <AccessMode for pv>]
             [-h|--help]
```

Note: `--path` and `--fileset` options are mutually exclusive. At least one of the options must be specified.

Example 1: Directory based static volume

The following example illustrates how to create a volume from a directory `/mnt/fs1/staticpv` within the file system 'fs1'.

```
./generate_static_provisioning_yamls.sh --filesystem fs1 --path /mnt/fs1/staticpv
--size 10 --pvname mystaticpv --guihost ibm-spectrum-scale-gui-ibm-spectrum-
scale.apps.cluster.cp.fyre.ibm.com
```

Example 2: Fileset based volume

The following example illustrates how to create a volume from a fileset 'fileset1' within the file system 'fs1'.

```
./generate_static_provisioning_yamls.sh --filesystem fs1 --fileset f1 --size 10 --pvname
mystaticpv --guihost ibm-spectrum-scale-gui-ibm-spectrum-scale.apps.cluster.cp.fyre.ibm.com
```

Note: The Path specified for option `--path` must be valid a gpfs path from the primary file system.

Creating a persistent volume (PV)

A persistent volume (PV) is the storage that is statically provisioned by an administrator or dynamically provisioned by using the storage classes.

To create a PV, do the following steps:

1. Download the sample file by issuing the following command and update with necessary parameters:

```
curl -O https://raw.githubusercontent.com/IBM/ibm-spectrum-scale-csi/v2.11.1/driver/examples/
version1/volume/staticprovisioning/static_pv.yaml
```

2. Configure persistent volume (PV) manifest file with a volumeHandle as described in this example.

```
# cat pv.yaml
apiVersion: v1
kind: PersistentVolume
metadata:
  name: static-scale-static-pv
spec:
  capacity:
    storage: 1Gi
  accessModes:
    - ReadWriteMany
  csi:
    driver: spectrumscale.csi.ibm.com
    volumeHandle: 0;2;7171748422707577770;13280B0A:61F4048E;;fset2;/ibm/fs1/fset2
```

Field Name	Description
volumeHandle	This input must follow this format: `0;[Volume type];[Cluster ID];[Filesystem UUID];;[Fileset name];[Path to the directory or fileset linkpath]`. For statically provisioned PV, the first field is "0" and the fifth field is always empty. For directory based volume, fileset name is always empty. Volume type is "0" for directory based volume, "1" for dependent fileset based volume and "2" for independent fileset based volume.
clusterID	The ID of the primary cluster. **Note:** The **`mmlscluster`** command displays the current configuration, including the cluster ID.
Filesystem UID	This identifier is the UID of the file system that consists of the directory to be provisioned. **Note:** The **`mmlsfs <filesystem name> --uid`** command displays the file system UID.
path	The complete path of volume directory in OpenShift or Kubernetes cluster.
Fileset ID	The **`Fileset ID`** field must be used if you want to create a snapshot volume from an independent fileset.

Note: This manifest file can be auto-generated by using the `generate_static_provisioning_yamls.sh` tool.

For more information, see Static provisioning YAML file.

3. Issue this command to create a PV:

```
kubectl apply -f pv.yaml
```

Creating a PersistentVolumeClaim (PVC)

A PVC is a request for storage by a user. There are two types of PVCs, static provisioning, and dynamic provisioning.

Create a PVC manifest as follows:

1. Create a `pvc.yaml` file:

```
# cat pvc.yaml
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: scale-static-pvc
spec:
  volumeName: static-scale-static-pv
  accessModes:
  - ReadWriteMany
  resources:
    requests:
      storage: 1Gi
```

2. Create a PVC by issuing the following command:

```
kubectl apply -f pvc.yaml
```

   This PVC is bound to an available PV with storage equal to or greater than what is specified in the `pvc.yaml` file.

Tiering support

Manage the location of newly created files in a specific storage pool with tiering support.

When the storage class is assigned to a specific "tier", files that are created in volumes that belong to that storage class are placed in the assigned "tier". For more information about storage pools in IBM Storage Scale, see the IBM Storage Scale documentation.

A sample storage class with tiering is created by applying a configuration like the following one:

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
    name: ibm-spectrum-scale-tiering
parameters:
    version: "2"
    volBackendFs: fs0
    tier: "storagePoolName"
provisioner: spectrumscale.csi.ibm.com
reclaimPolicy: Delete
```

Where, the `tier` field is optional. This optional field can be set to one of the storage pools that are defined in the IBM Storage Scale file system that is specified in the `volBackendFs` field. If the storage pool that is specified in the `tier` field does not exist in IBM Storage Scale `volBackendFs`, the request fails.

Requirements: IBM Storage Scale 5.1.3 or later and file system 27.00 or later

Compression Support

Provides support for compressing files, typically on some schedule with cronjobs.

The files will not be compressed until a cronjob runs. As files are created or modified, they will be uncompressed until the job runs again. When the storage class is created with `compression: true`, the fileset created in the IBM Storage Scale file system will have a name appended to it such as **`-COMPRESSZcsi`**. The compression field is optional, and by default is "false", which means that no files are compressed within that PV.

A sample storage class with compression enabled:

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
    name: ibm-spectrum-scale-compression
parameters:
    version: "2"
    volBackendFs: fs0
    compression: "true" # Default: false
provisioner: spectrumscale.csi.ibm.com
reclaimPolicy: Delete
```

Perform the following steps to create a cronjob:

1. Create compression policy file.

   ```
   $ echo "RULE 'FSETCOMPRESSION' MIGRATE COMPRESS('z') WHERE FILESET_NAME LIKE
   '%COMPRESSZcsi%'" > zcompress.policy
   $ chmod 400 zcompress.policy
   ```

2. Create a crontab entry on the owning cluster of the file system to run compression policy.

   ```
   $ crontab -e
   # insert line
   0 0 * * * /usr/lpp/mmfs/bin/mmapplypolicy <filesystemName> -P zcompress.policy -I yes
   # save and quit
   ```

Chapter 8. Managing IBM Storage Scale when used with IBM Storage Scale Container Storage Interface driver

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

Adding a new node to the Kubernetes or Red Hat OpenShift cluster

Describes the procedure for adding a node to the Kubernetes or Red Hat OpenShift cluster.

Do the following steps:

1. Add the node to the IBM Storage Scale cluster. For more information, see *mmaddnode command* in the *IBM Storage Scale: Command and Programming Reference Guide*.
2. Mount the required file systems on the newly added node.
3. Add the new node into the Kubernetes cluster. For more information, see the Kubernetes documentation.
4. Add the node mapping if required by the new node.
5. Add a label to the node. For example, `scale=true`
6. Check whether the IBM Storage Scale Container Storage Interface driver pods are running correctly.

Unmounting IBM Storage Scale file system

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

1. Move the containers that are using the file system, which is being unmounted, to other nodes.
2. Ensure that the new pods that are using the IBM Storage Scale file system, which is being unmounted, are not scheduled on the node.
3. Unmount the IBM Storage Scale file system using the **mmunmount** command.

For more information on the **mmunmount** command, see *mmunmount command* in the *IBM Storage Scale: Command and Programming Reference* Guide.

If you face an issue in unmounting the IBM Storage Scale file systems, see *File system fails to unmount* topic in the *IBM Storage Scale: Problem Determination Guide*.

Shutting down IBM Storage Scale

Follow these steps to shut down IBM Storage Scale when using IBM Storage Scale Container Storage Interface driver:

1. Move the containers that are using the file system, which is being unmounted, to other nodes.
2. Ensure that the new pods that are using the IBM Storage Scale file systems that are being unmounted are not scheduled on the node.
3. Stop Kubernetes and Docker.
4. Shut down the IBM Storage Scale file system using the **mmshutdown** command.

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

For more information on the **mmshutdown** command, see *mmshutdown command* in the *IBM Storage Scale: Command and Programming Reference* Guide.

IBM Storage Scale monitoring considerations

Consider the following information for IBM Storage Scale when using IBM Storage Scale Container Storage Interface driver:

- If an IBM Storage Scale file system that is being used by Kubernetes gets unmounted, or if there is an issue with the IBM Storage Scale file system mounted on a particular node, then the applications in the containers that are using the PVC from IBM Storage Scale throw an I/O error.
- Users must directly monitor IBM Storage Scale for any IBM Storage Scale specific issues, as such monitoring is not done by Kubernetes or IBM Storage Scale Container Storage Interface driver.

Upgrading IBM Storage Scale on IBM Storage Scale Container Storage Interface driver nodes

IBM Storage Scale can be upgraded on the nodes where IBM Storage Scale Container Storage Interface driver is already running.

On the worker nodes

As a first step, upgrade IBM Storage Scale on the worker nodes. Perform the following steps to upgrade IBM Storage Scale on worker nodes:

1. Cordon the worker node so that scheduling is disabled.

   ```
   kubectl cordon <node>
   ```

2. Move any workload off the worker node.
3. Remove the `scale` label from the node.

   ```
   kubectl label node <node> scale-
   ```

4. Prepare the worker node to shut down IBM Storage Scale.
   a. Check for open files.

      ```
      lsof <filesystem>
      ```

   b. Check for mounted kernel modules.

      ```
      lsmod | grep mm
      ```

   c. If there is any mm* present, then unmount and shut down file systems on the worker node. You can use the following options to view the details of the mounted file systems:
      - Check the GPFS status:

        ```
        /usr/lpp/mmfs/bin/mmgetstate
        ```

        Note: The GPFS state should be active.
      - List the mounted file systems across all nodes by issuing the following command:

        ```
        /usr/lpp/mmfs/bin/mmlsmount all
        ```

      - List disk space usage to see what file systems are mounted by issuing the **df** command:

        ```
        df
        ```

   d. Unmount all file systems for the current node.

```
/usr/lpp/mmfs/bin/mmunmount all
```

e. Shut down GPFS on the current node.

```
/usr/lpp/mmfs/bin/mmshutdown
```

f. Check for open files.

```
lsof <file system>
```

g. Check for mounted kernel modules.

```
lsmod | grep mm
```

Note:

- All file systems must be unmounted, and GPFS must be shut down. Continue to step 5 to proceed with IBM Storage Scale upgrade.
- If there are any file systems or mounted kernel modules (mm*) present, then you require a reboot of the worker node to clean up the state. Ensure **autoload** is set to `off` for the node before rebooting.

h. Set **autoload** to `off`.

```
/usr/lpp/mmfs/bin/mmchconfig autoload=no -N <node>
```

i. Reboot the worker node.

```
reboot
```

5. Upgrade IBM Storage Scale by using the toolkit, set the worker node as an offline node, and exclude the other nodes.
6. After the upgrade is completed, do the following steps:

a. Log on to the worker node and ensure that **autoload** is set back to on.

```
/usr/lpp/mmfs/bin/mmchconfig autoload=yes -N <node>
```

b. Log on to the worker node and start GPFS.

```
/usr/lpp/mmfs/bin/mmstartup
```

c. Uncordon the worker node.

```
kubectl uncordon <node>
```

d. Relabel the node `scale`.

```
kubectl label node <node> scale=true
```

On the nodes running CSI sidecars (Provisioner, Attacher, Snapshotter, Resizer etc)

As the next step, you must upgrade IBM Storage Scale on the nodes where provisioner, attacher, resizer, and snapshotter pods are running.

1. Move or stop all pods that use volumes that are managed by the IBM Storage Scale Container Storage Interface driver.
2. Drain the nodes so that sidecar pods move to other nodes.

```
kubectl drain <nodename> --ignore-daemonsets —delete-local-data
```

3. Remove the `PluginSelector` label that is assigned to the infrastructure node.

```
kubectl label node <nodename> scale-
```

4. Prepare the node to shut down IBM Storage Scale.
   a. Check for open files.
   ```
   lsof <filesystem>
   ```
   b. Check for mounted kernel modules.
   ```
   lsmod | grep mm
   ```
   c. If there is any mm* present, then unmount and shut down file systems on the node.
      - Ensure that GPFS is active on the node.
      ```
      /usr/lpp/mmfs/bin/mmgetstate
      ```
      - To list the mounted file systems across all nodes.
      ```
      /usr/lpp/mmfs/bin/mmlsmount all
      ```
      - To list disk space usage and more importantly to see what file systems are mounted.
      ```
      df
      ```
   d. Unmount all file systems for the current node.
   ```
   /usr/lpp/mmfs/bin/mmunmount all
   ```
   e. Shut down GPFS on the current node.
   ```
   /usr/lpp/mmfs/bin/mmshutdown
   ```
   f. Check for open files.
   ```
   lsof <filesystem>
   ```
   g. Check for mounted kernel modules.
   ```
   lsmod | grep mm
   ```
   Note:
   - If all file systems are unmounted and GPFS is shut down, continue to step 5 to proceed with IBM Storage Scale upgrade.
   - If there are any file systems or mounted kernel modules (mm*) present, then do a reboot of the worker node to clean up the state. Ensure that **autoload** is set to `off` for the node before rebooting.

   h. Disable **autoload** by issuing the following command:
   ```
   /usr/lpp/mmfs/bin/mmchconfig autoload=no -N <node>
   ```
   i. Reboot the worker node.
   ```
   reboot
   ```
5. Upgrade the IBM Storage Scale using the IBM Storage Scale installation toolkit, set the worker node as an offline node, and exclude other nodes.
6. After the upgrade is completed, do the following steps:
   a. Log on to the worker node and issue the following command to enable autoload:
   ```
   /usr/lpp/mmfs/bin/mmchconfig autoload=yes -N <node>
   ```

b. Log on to the worker node and start GPFS.

```
/usr/lpp/mmfs/bin/mmstartup
```

c. Uncordon the node.

```
kubectl uncordon <node>
```

d. Relabel the node with IBM Storage Scale Container Storage Interface driver.

```
kubectl label node <node> scale=true
```

Chapter 9. Cleanup

Cleaning up IBM Storage Scale Container Storage Interface driver and Operator by using CLIs

Clean up or uninstall the IBM Storage Scale Container Storage Interface driver and the Operator by using the command line interface (CLI).

To manage IBM Storage Scale Container Storage Interface driver, the Operator must be running on your cluster. If the Operator is deleted for some reason, ensure to redeploy it by using the `kubectl apply -f ibm-spectrum-scale-csi-operator.yaml` command before you proceed with the following steps:

1. To stop and uninstall IBM Storage Scale Container Storage Interface driver, issue the following command:

   ```
   kubectl delete CSIScaleOperator ibm-spectrum-scale-csi -n ibm-spectrum-scale-csi-driver
   ```

   Note: Ensure that `CustomResource` is properly deleted before you proceed for operator deletion. This operation takes few minutes to delete all the resources created by the Operator.
2. To uninstall Operator and clean up all resources, issue the following commands:

   ```
   kubectl delete -f https://raw.githubusercontent.com/IBM/ibm-spectrum-scale-csi/v2.11.1/
   generated/installer/ibm-spectrum-scale-csi-operator.yaml
   ```

   If you are using OCP cluster with RHEL nodes, issue the following command:

   ```
   kubectl delete -f https://raw.githubusercontent.com/IBM/ibm-spectrum-scale-csi/v2.11.1/
   generated/installer/ibm-spectrum-scale-csi-operator-ocp-rhel.yaml
   ```

   Note: Delete the secrets for GUI credentials and configmap for CA certificates (if any) under the `ibm-spectrum-scale-csi-driver` namespace.
3. Delete the ibm-spectrum-scale-csi-driver namespace:

   ```
   kubectl delete namespace ibm-spectrum-scale-csi-driver
   ```
4. Remove all the IBM Storage Scale Container Storage Interface driver container images from the Kubernetes or OCP worker nodes.

   Note: If you use a different container engine than Docker, replace the Docker commands with the commands of the container engine that you use.
5. To delete PVC data, unlink and delete the primary fileset that is defined in the `csiscaleoperators.csi.ibm.com_cr.yaml` file from your IBM Storage Scale cluster, by issuing the following commands:

   ```
   /usr/lpp/mmfs/bin/mmunlinkfileset gpfs0 spectrum-scale-csi-volume-store
   /usr/lpp/mmfs/bin/mmdelfileset gpfs0 spectrum-scale-csi-volume-store
   ```

   Note: The command in "step 4" completely deletes the PVC data, and any PVCs that are created before would no longer be useful even if the IBM Storage Scale Container Storage Interface driver is reinstalled.

Chapter 10. Limitations

The known limitations of IBM Storage Scale Container Storage Interface driver are provided in this section.

- IBM Storage Scale Container Storage Interface driver does not honor the size that is specified in `PersistentVolumeClaim` for lightweight volume.
- The **`df`** command inside a container does not show the correct volume size in the container for lightweight volumes. Instead, this command shows the size of the file system or fileset based on the `filesetdf` configuration.
- The **`df`** command inside the container shows only one entry per file system even when multiple PVCs from IBM Storage Scale are attached to the container or pod. Use `--all` option of `df` command to list all the entries.
- Maximum number of supported volumes that can be created by using independent fileset storage class is 998, excluding the root fileset and primary fileset reserved for IBM Storage Scale Container Storage Interface driver. The maximum number of volumes that can be created are 2998 in IBM Storage Scale 5.1.4.0 or later. This limitation is based on the number of filesets that are supported by IBM Storage Scale. For more information, see *IBM Storage Scale FAQ* in the IBM Storage Scale documentation.

  Note: `fsType` mentioned in the `storageClass` definition does not have any impact on the IBM Storage Scale Container Storage Interface driver. The `fsType` value (gpfs) is displayed in the `PersistentVolume` (PV) details.
- IBM Storage Scale Container Storage Interface driver relies on the GUI server for doing IBM Storage Scale operations during volume provisioning or deprovisioning and attach or detach. If the GUI password or CA certificate expires, then manual intervention is needed by the administrator to reset the password on GUI or generate a new certificate and update the configuration in IBM Storage Scale Container Storage Interface driver.
- IBM Storage Scale Container Storage Interface driver supports the `ReadOnlyMany` access mode of PVC only to create a restore volume (shallow copy volume) from a read-only snapshot. To enable the `ReadOnly` access, use the **`readOnly: true`** parameter in the pod specification.
- IBM Storage Scale file systems must remain mounted on the worker nodes while IBM Storage Scale Container Storage Interface driver is running. If unmounted for some reason, it might affect pods on a node mounting the volumes from these file systems, and you need to restart the node to mount the file system back.
- Volume provisioning and attachment operations rely on REST API status. Occasionally, there will be some delay in reflecting the file system status from the cluster into the REST API. In such instances, you might experience interim failures in volume provisioning and attach or detach operations, which go away when the REST API status is updated.
- IBM Storage Scale Container Storage Interface driver does not support the rolling upgrade of IBM Storage Scale while IBM Storage Scale Container Storage Interface driver or Operator pods are running on the worker nodes.
- In dynamic provisioning or consistency group, if the PVC is deleted before it goes into bound state then the fileset that is created for that PVC might not get deleted. You need to delete the fileset manually.
- In the static provisioning, the softlink must be resolved on the Kubernetes cluster for proper functioning. If the softlinks are not resolved, then you might have different file system mount points on the owning and the accessing cluster. This setup can lead to mounting or data access issue.
- While using fsGroup for large volumes, checking and changing the ownership and permissions can take longer that slowing down the Pod startup. You can use the `fsGroupChangePolicy` control in such a way that Kubernetes checks and manages the ownership and permissions for a volume.
- fsGroup support is applicable to previously created volume as well and is applied as soon as existing pod is restarted with IBM Storage Scale Container Storage Interface driver 2.6 or later. If you do not desire to have this effect, remove `fsGroup` from pod's `securityContext`.

- The following limitations are specific to the Volume Snapshots and Volume Cloning features:
  - Hard links from the source snapshot are copied as regular files during volume creation from the snapshot.
  - Performance of the volume creation from the snapshot operation depends on the level of subdirectories that exist in the snapshot and the data resides within the subdirectories. As the number of nested sub directories increases, the volume creation operation gets slower.
  - While creating PVC from the snapshot, the minimum size of the new PVC must be at least 1 GB as the initial fileset-based PVC, whose snapshot is used as a datasource, results in 1 GB PV.
  - You must have the default snapshots directory as `.snapshots` for filesets in IBM Storage Scale to use the snapshot feature for a consistency group.
- Specifying different CA signed certificate for GUI host is not supported while configuring the GUI High Availability feature in the CSI driver custom resource.
- The SELinux behavior and requirement changes are as follows:
  - SELinux enablement on a storage cluster is optional.
  - By using IBM Storage Scale Container Native, a file system is mounted with the predefined `system_u:object_r:container_file_t:s0` label.
  - From IBM Storage Scale Container Storage Interface driver 2.9.0, using the `container_file_t` label to access existing data in a container is optional.
  - SElinux is not relabeled when a pod is started.
  - Files created from an application pod have the system_u:object_r:unlabeled_t:s0 label on a storage cluster, which has SELinux labels. This is default behavior and might change based on SELinux policies that are defined on a storage cluster.
  - If you do not want any changes in the SELinux behavior, contact IBM Support before you upgrade to the latest CNSA version. For more information about SELinux behavior, see the *IBM Storage Scale container native and SELinux* section in the IBM Storage Scale Container Native documentation.
- Changing the **consistencyGroupPrefix** field in the `CSIScaleOperator` custom resource of IBM Storage Scale CSI is not supported.
- Using the same volume handle for different statically provisioned volumes is not supported.
- If the `/var/lib/kubelet` directory is mounted as a `symlink` on a worker node, any application pod in the termination state within CSI 2.10 on that node persists in the termination state even after upgrading to CSI 2.11.0.

Chapter 11. Troubleshooting

For any issue with the IBM Storage Scale Container Storage Interface driver functions, you must obtain the logs, which can be done by running the `storage-scale-driver-snap.sh` tool. These logs along with the output of the **`gpfs.snap`** command can be used for debugging the issue. Also, detailed output of k8s resources like PVC, snapshot, and pod for which failures are being seen is needed. The output can be generated by issuing the command in the format of `kubetctl get resource <resource name> -o yaml`.

Debug data collection

IBM Storage Scale Container Storage Interface driver provides the `storage-scale-driver-snap.sh` tool to collect the debug data. This tool gathers the state of required Kubernetes resources like nodes, pods, service accounts, and so on and collects `Deployment` and `DaemonSet` logs from all nodes. It collects the definition of resources in the given namespace with the label, `product=ibm-spectrum-scale-csi`. The collected logs are stored in the given output directory.

Issue the following command and download the tool for CSI 2.11.x:

```
curl -O https://raw.githubusercontent.com/IBM/ibm-spectrum-scale-csi/v2.11.1/tools/storage-scale-driver-snap.sh
```

Usage of the tool

```
storage-scale-driver-snap.sh [-l | -n | -o | -p | -s | -v | -h]
-l: Collects logs from driver pods that are running only on the same node as sidecar pods. Logs will be collected from all the driver pods if the -l option is not specified.
-n: Namespace from which the debug data for CSI resources is to be collected. If not specified, the default namespace is used. The tool returns error if CSI is not running under the given namespace.
-o: Output directory where debug data will be stored. If not specified, the debug data is stored in current directory.
-p: Previous instance of the container in a pod. If set to False, the logs for the previous instance of the container in a pod are not collected.
-s: Only returns the newer logs, which correspond to a specific time frame like the last 2 hours (2h) or 4 days (4d). By default, returns all logs.
-v: Prints the CSI version that is used on this cluster.
-h: Prints help information about the usage of the tool for debug data collection.
```

The resultant folder should contain the following file structure with debug information:

```
├── cluster-scoped-resources
│   ├── clusterrolebindings
│   ├── clusterroles
│   ├── csinodes
│   ├── nodes
│   ├── pv
│   ├── storageclass
│   └── volumeattachment
├── kube-system
│   ├── daemonsets.json
│   ├── deployments.json
│   ├── events.json
│   ├── pods.json
│   ├── replicasets.json
│   ├── replication-controllers.json
│   └── services.json
├── namespaces
│   └── ibm-spectrum-scale-csi-driver
├── nodes.json
└── version
```

In addition to these logs, the following details are necessary for troubleshooting:

- Include the following details if an issue is related to PVC.

```
kubectl describe namespace < namespace name >
kubectl get pvc < pvc name > -o yaml
kubectl describe pvc < pvc name >
kubectl describe pv < pv name from above pvc >
kubectl describe StorageClass < storage class name for pvc >
```

- Include the following details in addition to the previous if an issue is related to a pod.

```
kubectl describe pod < pod name >
kubectl get pod < pod name > -o yaml
kubectl describe VolumeAttachment < volume attachment name >
```

- Include the following details in addition to the previous if an issue is related to a snapshot.

```
kubectl describe VolumeSnapshot < snapshot name >
kubectl get VolumeSnapshot < snapshot name > -o yaml
kubectl describe VolumeSnapshotContent <snapshot content name >
kubectl describe VolumeSnapshotClass < snapshot name >
```

Debugging initialization issues

This section outlines how to debug IBM Storage Scale Container Storage Interface driver initialization issues.

Issue: IBM Storage Scale Container Storage Interface driver pod and sidecar pods do not come up during deployment

```
# kubectl get pod -n ibm-spectrum-scale-csi-driver
NAME                                              READY   STATUS     RESTARTS   AGE
ibm-spectrum-scale-csi-operator-58c54949ff-rqbzl  1/1     Running    0          98m
```

How to troubleshoot?

Operator validates the configuration parameters for CSIScaleOperator. If there is any failure in validation, driver and sidecar pods do not come up. Check the CSIScaleOperator status in the CSIScaleOperator custom resource as shown in the following example, where you can see the reason for the failure.

```
# kubectl describe CSIScaleOperator ibm-spectrum-scale-csi -n ibm-spectrum-scale-csi-driver
Status:
   Conditions:
        Last Transition Time: 2023-02-27T10:29:16Z
        Message: The Secret guisecret1 is not found. Please make sure to create Secret named
guisecret1
        Reason: GetFailed
        Status: False
        Type: Success
```

For example, the previous status indicates that the secret 'guisecret1' used to connect to the GUI server does not exist. In order to fix this issue, make sure that the secret used in the CSIScaleOperator custom resource, exists in the namespace ibm-spectrum-scale-csi-driver. To get more details about the failure check operator logs.

Debugging PVC creation issues

This section discusses the troubleshooting of PVC creation issues.

Issue: PVC remains in the pending state

```
# kubectl get pvc  scale-fset-pvc
NAME             STATUS    VOLUME   CAPACITY   ACCESS MODES   STORAGECLASS
AGE
```

```
scale-fset-pvc   Pending                                      ibm-spectrum-scale-csi-fileset
9s
```

How to troubleshoot?

Look for the PVC description. It should highlight any error prohibiting the volume creation, as shown in the following example:

```
# kubectl describe pvc scale-fset-pvc
Name:          scale-fset-pvc
Namespace:     ibm-spectrum-scale-csi-driver
StorageClass:  ibm-spectrum-scale-csi-fileset
Status:        Pending
Volume:
Labels:        <none>
Annotations:   volume.beta.kubernetes.io/storage-provisioner: spectrumscale.csi.ibm.com
Finalizers:    [kubernetes.io/pvc-protection]
Capacity:
Access Modes:
VolumeMode:    Filesystem
Events:
  Type       Reason                Age
From
     Message
  ----       ------                ----
----
     -------
  Normal     Provisioning          11s                spectrumscale.csi.ibm.com_ibm-spectrum-
scale-csi-provisioner-0_c58323a3-436a-11ea-9c1a-920ed99f44ce  External provisioner is
provisioning volume for claim "ibm-spectrum-scale-csi-driver/scale-fset-pvc"
  Warning    ProvisioningFailed    10s                spectrumscale.csi.ibm.com_ibm-spectrum-
scale-csi-provisioner-0_c58323a3-436a-11ea-9c1a-920ed99f44ce  failed to provision volume with
StorageClass "ibm-spectrum-scale-csi-fileset": rpc error: code = Internal desc = Unable to
create fileset [pvc-4696a3e4-5006-11ea-8b62-000c2932e5ce] in FS [scale0]. Error [[EFSSG0072C
File set myscalefileset does not exist.]]
  Normal     ExternalProvisioning  10s (x2 over 10s)  persistentvolume-
controller                                                                    waiting for
a volume to be created, either by external provisioner "spectrumscale.csi.ibm.com" or manually
created by system administrator
Mounted By:  <none>
```

Cleanup of PVCs in Pending state

When volume provisioning fails and you see a PVC in `Pending` state, follow the steps for manual cleanup of the PVCs and IBM Storage Scale filesets that are created on the storage cluster, if any:

1. Before you delete the PVC that is in `Pending` state, get the UID of the PVC by issuing following command:

   ```
   kubectl get pvc <PVC name> -oyaml | grep uid
   ```

2. Issue following command to delete the PVC:

   ```
   kubectl delete pvc <PVC name>
   ```

3. From storage cluster, delete the fileset with name `pvc-<UID>` by issuing following commands:

   ```
   /usr/lpp/mmfs/bin/mmunlinkfileset <filesystem name> pvc-<UID>
   /usr/lpp/mmfs/bin/mmdelfileset <filesystem name> pvc-<UID>
   ```

Debugging pod mounting issues

This section discusses the troubleshooting of issues related to pod mounting.

Issue

Application pod fails to start and does not go in the `Running` state.

How to troubleshoot?

Look for pod description for the root cause of failure.

```
# kubectl describe pod my-csi-pod -n ibm-spectrum-scale-csi-driver
.
.
Events:
  Type     Reason              Age                    From                     Message
  ----     ------              ----                   ----                     -------
  Normal   Scheduled           8s                     default-scheduler        Successfully
assigned spectrum-scale-csi/csi-scale-fsetdemo-pod to scuttleclaw-compute4
  Warning  FailedAttachVolume  <invalid> (x6 over 8s)  attachdetach-controller
AttachVolume.Attach failed for volume "pvc-f3024f7a-06be-11ea-9384-00505695e231" : rpc error:
code = Internal desc = ControllerPublishVolume : SKIP_MOUNT_UNMOUNT == yes and either fs1 or
fs1 in not mounted on node scuttleclaw-compute4.
```

Root cause

If the above error is seen despite the file system being mounted on a given node, then the root cause is that IBM Storage Scale node names and Kubernetes node names are different, and node mapping is not configured. For more information, see “Kubernetes to IBM Storage Scale node mapping” on page 31.

Debugging GUI issues

Debugging IBM Storage Scale GUI issues that affects IBM Storage Scale Container Storage Interface driver operations.

Symptoms

- PVC and Snapshot creation and deletion operation fails.
- Attach and Detach operation of PVC to Pod fails.
- IBM Storage Scale Container Storage Interface driver pod fails repeatedly during the GUI start.

Causes

- IBM Storage Scale GUI does not function as expected.
- GUI user credentials that are needed for IBM Storage Scale Container Storage Interface driver setup are no longer valid.
- GUI SSL CA certificate is expired.

If the GUI user credentials are reset or expired, then you can see the related error messages during various volume-specific operations. The following example shows pending status for Create PVC operation due to issues as written in the messages.

```
[root@cnss-deeghuge7-inf fileset]# oc describe pvc scale-fset-pvc-1
Name:          scale-fset-pvc-1
Namespace:     ibm-spectrum-scale-csi-driver
StorageClass:  ibm-spectrum-scale-csi-fileset
Status:        Pending
Volume:
Labels:        <none>
Annotations:   volume.beta.kubernetes.io/storage-provisioner: spectrumscale.csi.ibm.com
Finalizers:    [kubernetes.io/pvc-protection]
Capacity:
Access Modes:
VolumeMode:    Filesystem
Mounted By:    <none>
Events:
  Type     Reason                Age
From
     Message
  ----     ------                ----
----
     -------
  Normal   ExternalProvisioning  9s (x2 over 13s)  persistentvolume-
controller                                                                    waiting for
a volume to be created, either by external provisioner "spectrumscale.csi.ibm.com" or manually
```

```
created by system administrator
  Normal   Provisioning          5s (x4 over 13s)  spectrumscale.csi.ibm.com_ibm-spectrum-scale-
csi-provisioner-0_1fc0b34c-fc13-46b5-bf04-552bad562df0  External provisioner is provisioning
volume for claim "ibm-spectrum-scale-csi-driver/scale-fset-pvc-1"
  Warning  ProvisioningFailed    5s (x4 over 13s)  spectrumscale.csi.ibm.com_ibm-spectrum-scale-
csi-provisioner-0_1fc0b34c-fc13-46b5-bf04-552bad562df0  failed to provision volume with
StorageClass "ibm-spectrum-scale-csi-fileset": rpc error: code = Internal desc = unable to
check type of filesystem [fs0]. Error: rpc error: code = Unauthenticated desc = Unauthorized
GET request to https://10.11.48.228:443/scalemgmt/v2/filesystems/fs0: 401 Unauthorized
```

Resolving the problem

1. To ensure proper functioning of IBM Storage Scale GUI, monitoring of GUI health status helps debug and resolve the IBM Storage Scale Container Storage Interface driver GUI issues. You can check the health of the GUI node by issuing the **mmhealth** command. The following output sample shows no error when **mmhealth** command is issued to check the GUI status.

   ```
   [root@remote-deeghuge7-1 ~]# /usr/lpp/mmfs/bin/mmhealth node show GUI -N remote-
   deeghuge7-2.ibm.com
   Node name:      remote-deeghuge7-2.ibm.com
   Component     Status        Status Change     Reasons
   ------------------------------------------------------
   GUI           HEALTHY       21 days ago
   ```

2. To resolve the GUI access issue, reset the IBM Storage Scale GUI user credential and update the secret for IBM Storage Scale Container Storage Interface driver. For more information about updating a secret, see “Changing the configuration after deployment” on page 32.

   Note: The PVC creation issue, as previously mentioned, can be resolved by fixing the secret as given in Step 2.

3. To benefit from the GUI HA function (multiple GUIs on a storage cluster), a GUI node must be completely down.

   If a GUI node is available but the file system is unmounted, the GUI HA function cannot be availed and a volume provisioning failure occurs. In such case, the GUI node where the file system is unmounted must be shut down manually to benefit from the GUI HA function.

For more information about how to monitor, administer, and troubleshoot GUI-related issues, see monitoring, administration, and troubleshooting sections in the specific versions of IBM Storage Scale documentation.

Appendix A. Installing IBM Storage Scale CSI on a Kubernetes cluster with RHEL 8 or RHEL 9 nodes

Note: No official Kubernetes community or Red Hat support documentation is available for the Kubernetes installation on RHEL 8 or RHEL 9. Although many unofficial documentations are available on the Internet, the Kubernetes configuration in these documentations is not validated with IBM Storage Scale CSI driver. Ideally, if Kubernetes works properly, CSI is also expected to work. If CSI does not work because of some Kubernetes configuration issues, these issues can be fixed with a limited support. The following configuration is used on RHEL 8 or RHEL 9 nodes.

1. Enable IP forwarding on the nodes on which you are setting up a Kubernetes cluster, if it is not enabled.

   ```
   echo 1 > /proc/sys/net/ipv4/ip_forward
   ```

2. Install the kubeadm toolbox to create a Kubernetes cluster.

   For more information, see Installing kubeadm in the Kubernetes documentation.

   a) Install a container runtime to run a container.

      For more information, see Installing a container runtime in the Kubernetes documentation.

      For x86_64 architecture, use the `cri-o` container runtime and choose VERSION=1.28, and OS=CentOS_8 for RHEL 8 or OS=CentOS_9_Stream for RHEL 9. For more information, see CRI-O installation instructions.

      For `ppc64le` architecture, use `containerd` container runtime. For more information, see containerd installation instructions.

3. Verify the RPM versions that are installed.

   For x86_64 architecture:

   ```
   # rpm -qa | grep kubelet
   kubelet-1.28.2-0.x86_64
   # rpm -qa | grep kubectl
   kubectl-1.28.2-0.x86_64
   # rpm -qa | grep kubeadm
   kubeadm-1.28.2-0.x86_64
   ```

   For `ppc64le` architecture:

   ```
   # rpm -qa | grep kubelet
   kubelet-1.29.3-150500.1.1.ppc64le
   # rpm -qa | grep kubectl
   kubectl-1.29.3-150500.1.1.ppc64le
   # rpm -qa | grep kubeadm
   kubeadm-1.29.3-150500.1.1.ppc64le
   ```

   The following configuration is tested and supported on the cluster:

   - For x86_64 architecture:
     - IBM Storage Scale: 5.1.9.3
     - Kubernetes: 1.28
     - RHEL: 9.2
     - Linux Kernel: RHEL 9:5.14.0-284.30.1.el9_2.x86_64
     - `cri-o`: 1.28
   - For `ppc64le` architecture:
     - IBM Storage Scale: 5.2.0.0

- Kubernetes: 1.29
- RHEL: 8.8
- Kernel Linux: 4.18.0-477.21.1.el8_8.ppc64le
- `containerd`: 1.6.9

Note: If any issues are observed with any variation of this configuration, a limited support can be provided.

4. Install CSI on the Kubernetes cluster. For more information about the CSI installation, see the Installation chapter.

Accessibility features for IBM Storage Scale

Accessibility features help users who have a disability, such as restricted mobility or limited vision, to use information technology products successfully.

Accessibility features

The following list includes the major accessibility features in IBM Storage Scale:

- Keyboard-only operation
- Interfaces that are commonly used by screen readers
- Keys that are discernible by touch but do not activate just by touching them
- Industry-standard devices for ports and connectors
- The attachment of alternative input and output devices

IBM Documentation, and its related publications, are accessibility-enabled.

Keyboard navigation

This product uses standard Microsoft Windows navigation keys.

IBM and accessibility

See the IBM Human Ability and Accessibility Center (www.ibm.com/able) for more information about the commitment that IBM has to accessibility.

Notices

This information was developed for products and services offered in the US. This material might be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing IBM Corporation North Castle Drive, MD-NC119 Armonk, NY 10504-1785 US

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

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

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

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

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

IBM Director of Licensing IBM Corporation North Castle Drive, MD-NC119 Armonk, NY 10504-1785 US

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

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

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

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and

cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

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

All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to actual people or business enterprises is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Each copy or any portion of these sample programs or any derivative work must include a copyright notice as follows:

© (your company name) (year).
Portions of this code are derived from IBM Corp.
Sample Programs. © Copyright IBM Corp. _enter the year or years_.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at Copyright and trademark information at www.ibm.com/legal/copytrade.shtml.

Intel is a trademark of Intel Corporation or its subsidiaries in the United States and other countries.

Java™ and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

The registered trademark Linux is used pursuant to a sublicense from the Linux Foundation, the exclusive licensee of Linus Torvalds, owner of the mark on a worldwide basis.

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

Red Hat, OpenShift, and Ansible® are trademarks or registered trademarks of Red Hat, Inc. or its subsidiaries in the United States and other countries.

UNIX is a registered trademark of the Open Group in the United States and other countries.

Terms and conditions for product documentation

Permissions for the use of these publications are granted subject to the following terms and conditions.

IBM Privacy Policy

At IBM we recognize the importance of protecting your personal information and are committed to processing it responsibly and in compliance with applicable data protection laws in all countries in which IBM operates.

Visit the IBM Privacy Policy for additional information on this topic at https://www.ibm.com/privacy/details/us/en/.

Applicability

These terms and conditions are in addition to any terms of use for the IBM website.

Personal use

You can reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You cannot distribute, display, or make derivative work of these publications, or any portion thereof, without the express consent of IBM.

Commercial use

You can reproduce, distribute, and display these publications solely within your enterprise provided that all proprietary notices are preserved. You cannot make derivative works of these publications, or reproduce, distribute, or display these publications or any portion thereof outside your enterprise, without the express consent of IBM.

Rights

Except as expressly granted in this permission, no other permissions, licenses, or rights are granted, either express or implied, to the Publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions that are granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or as determined by IBM, the above instructions are not being properly followed.

You cannot download, export, or reexport this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.

Glossary

This glossary provides terms and definitions for IBM Storage Scale.

The following cross-references are used in this glossary:

- *See* refers you from a nonpreferred term to the preferred term or from an abbreviation to the spelled-out form.
- *See also* refers you to a related or contrasting term.

For other terms and definitions, see the IBM Terminology website (www.ibm.com/software/globalization/terminology) (opens in new window).

B

block utilization
The measurement of the percentage of used subblocks per allocated blocks.

C

cluster
A loosely coupled collection of independent systems (nodes) organized into a network for the purpose of sharing resources and communicating with each other. See also *GPFS cluster*.

cluster configuration data
The configuration data that is stored on the cluster configuration servers.

Cluster Export Services (CES) nodes
A subset of nodes configured within a cluster to provide a solution for exporting GPFS file systems by using the Network File System (NFS), Server Message Block (SMB), and S3 protocols.

cluster manager
The node that monitors node status using disk leases, detects failures, drives recovery, and selects file system managers. The cluster manager must be a quorum node. The selection of the cluster manager node favors the quorum-manager node with the lowest node number among the nodes that are operating at that particular time.

Note: The cluster manager role is not moved to another node when a node with a lower node number becomes active.

clustered watch folder
Provides a scalable and fault-tolerant method for file system activity within an IBM Storage Scale file system. A clustered watch folder can watch file system activity on a fileset, inode space, or an entire file system. Events are streamed to an external Kafka sink cluster in an easy-to-parse JSON format. For more information, see the *mmwatch command* in the *IBM Storage Scale: Command and Programming Reference Guide*.

control data structures
Data structures needed to manage file data and metadata cached in memory. Control data structures include hash tables and link pointers for finding cached data; lock states and tokens to implement distributed locking; and various flags and sequence numbers to keep track of updates to the cached data.

D

Data Management Application Program Interface (DMAPI)
The interface defined by the Open Group's XDSM standard as described in the publication *System Management: Data Storage Management (XDSM) API Common Application Environment (CAE) Specification C429*, The Open Group ISBN 1-85912-190-X.

deadman switch timer
A kernel timer that works on a node that has lost its disk lease and has outstanding I/O requests. This timer ensures that the node cannot complete the outstanding I/O requests (which would risk causing file system corruption), by causing a panic in the kernel.

dependent fileset
A fileset that shares the inode space of an existing independent fileset.

disk descriptor
A definition of the type of data that the disk contains and the failure group to which this disk belongs. See also *failure group*.

disk leasing
A method for controlling access to storage devices from multiple host systems. Any host that wants to access a storage device configured to use disk leasing registers for a lease; in the event of a perceived failure, a host system can deny access, preventing I/O operations with the storage device until the preempted system has reregistered.

disposition
The session to which a data management event is delivered. An individual disposition is set for each type of event from each file system.

domain
A logical grouping of resources in a network for the purpose of common management and administration.

E

ECKD
See *extended count key data (ECKD)*.

ECKD device
See *extended count key data device (ECKD device)*.

encryption key
A mathematical value that allows components to verify that they are in communication with the expected server. Encryption keys are based on a public or private key pair that is created during the installation process. See also *file encryption key*, *master encryption key*.

extended count key data (ECKD)
An extension of the count-key-data (CKD) architecture. It includes additional commands that can be used to improve performance.

extended count key data device (ECKD device)
A disk storage device that has a data transfer rate faster than some processors can utilize and that is connected to the processor through use of a speed matching buffer. A specialized channel program is needed to communicate with such a device. See also *fixed-block architecture disk device*.

F

failback
Cluster recovery from failover following repair. See also *failover*.

failover
(1) The assumption of file system duties by another node when a node fails. (2) The process of transferring all control of the ESS to a single cluster in the ESS when the other clusters in the ESS fails. See also *cluster*. (3) The routing of all transactions to a second controller when the first controller fails. See also *cluster*.

failure group
A collection of disks that share common access paths or adapter connections, and could all become unavailable through a single hardware failure.

FEK
See *file encryption key*.

fileset
A hierarchical grouping of files managed as a unit for balancing workload across a cluster. See also *dependent fileset*, *independent fileset*.

fileset snapshot
A snapshot of an independent fileset plus all dependent filesets.

file audit logging
Provides the ability to monitor user activity of IBM Storage Scale file systems and store events related to the user activity in a security-enhanced fileset. Events are stored in an easy-to-parse JSON format. For more information, see the *mmaudit command* in the *IBM Storage Scale: Command and Programming Reference Guide*.

file clone
A writable snapshot of an individual file.

file encryption key (FEK)
A key used to encrypt sectors of an individual file. See also *encryption key*.

file-management policy
A set of rules defined in a policy file that GPFS uses to manage file migration and file deletion. See also *policy*.

file-placement policy
A set of rules defined in a policy file that GPFS uses to manage the initial placement of a newly created file. See also *policy*.

file system descriptor
A data structure containing key information about a file system. This information includes the disks assigned to the file system (*stripe group*), the current state of the file system, and pointers to key files such as quota files and log files.

file system descriptor quorum
The number of disks needed in order to write the file system descriptor correctly.

file system manager
The provider of services for all the nodes using a single file system. A file system manager processes changes to the state or description of the file system, controls the regions of disks that are allocated to each node, and controls token management and quota management.

fixed-block architecture disk device (FBA disk device)
A disk device that stores data in blocks of fixed size. These blocks are addressed by block number relative to the beginning of the file. See also *extended count key data device*.

fragment
The space allocated for an amount of data too small to require a full block. A fragment consists of one or more subblocks.

G

GPUDirect Storage
IBM Storage Scale's support for NVIDIA's GPUDirect Storage (GDS) enables a direct path between GPU memory and storage. File system storage is directly connected to the GPU buffers to reduce latency and load on CPU. Data is read directly from an NSD server's pagepool and it is sent to the GPU buffer of the IBM Storage Scale clients by using RDMA.

global snapshot
A snapshot of an entire GPFS file system.

GPFS cluster
A cluster of nodes defined as being available for use by GPFS file systems.

GPFS portability layer
The interface module that each installation must build for its specific hardware platform and Linux distribution.

GPFS recovery log
A file that contains a record of metadata activity and exists for each node of a cluster. In the event of a node failure, the recovery log for the failed node is replayed, restoring the file system to a consistent state and allowing other nodes to continue working.

I

ill-placed file
A file assigned to one storage pool but having some or all of its data in a different storage pool.

ill-replicated file
A file with contents that are not correctly replicated according to the desired setting for that file. This situation occurs in the interval between a change in the file's replication settings or suspending one of its disks, and the restripe of the file.

independent fileset
A fileset that has its own inode space.

indirect block
A block containing pointers to other blocks.

inode
The internal structure that describes the individual files in the file system. There is one inode for each file.

inode space
A collection of inode number ranges reserved for an independent fileset, which enables more efficient per-fileset functions.

ISKLM
IBM Security Key Lifecycle Manager. For GPFS encryption, the ISKLM is used as an RKM server to store MEKs.

J

journaled file system (JFS)
A technology designed for high-throughput server environments, which are important for running intranet and other high-performance e-business file servers.

junction
A special directory entry that connects a name in a directory of one fileset to the root directory of another fileset.

K

kernel
The part of an operating system that contains programs for such tasks as input/output, management and control of hardware, and the scheduling of user tasks.

M

master encryption key (MEK)
A key used to encrypt other keys. See also *encryption key*.

MEK
See *master encryption key*.

metadata
Data structures that contain information that is needed to access file data. Metadata includes inodes, indirect blocks, and directories. Metadata is not accessible to user applications.

metanode
The one node per open file that is responsible for maintaining file metadata integrity. In most cases, the node that has had the file open for the longest period of continuous time is the metanode.

mirroring
The process of writing the same data to multiple disks at the same time. The mirroring of data protects it against data loss within the database or within the recovery log.

Microsoft Management Console (MMC)

A Windows tool that can be used to do basic configuration tasks on an SMB server. These tasks include administrative tasks such as listing or closing the connected users and open files, and creating and manipulating SMB shares.

multi-tailed
A disk connected to multiple nodes.

N

namespace
Space reserved by a file system to contain the names of its objects.

Network File System (NFS)
A protocol, developed by Sun Microsystems, Incorporated, that allows any host in a network to gain access to another host or netgroup and their file directories.

Network Shared Disk (NSD)
A component for cluster-wide disk naming and access.

NSD volume ID
A unique 16-digit hex number that is used to identify and access all NSDs.

node
An individual operating-system image within a cluster. Depending on the way in which the computer system is partitioned, it may contain one or more nodes.

node descriptor
A definition that indicates how GPFS uses a node. Possible functions include: manager node, client node, quorum node, and nonquorum node.

node number
A number that is generated and maintained by GPFS as the cluster is created, and as nodes are added to or deleted from the cluster.

node quorum
The minimum number of nodes that must be running in order for the daemon to start.

node quorum with tiebreaker disks
A form of quorum that allows GPFS to run with as little as one quorum node available, as long as there is access to a majority of the quorum disks.

non-quorum node
A node in a cluster that is not counted for the purposes of quorum determination.

Non-Volatile Memory Express (NVMe)
An interface specification that allows host software to communicate with non-volatile memory storage media.

P

policy
A list of file-placement, service-class, and encryption rules that define characteristics and placement of files. Several policies can be defined within the configuration, but only one policy set is active at one time.

policy rule
A programming statement within a policy that defines a specific action to be performed.

pool
A group of resources with similar characteristics and attributes.

portability
The ability of a programming language to compile successfully on different operating systems without requiring changes to the source code.

primary GPFS cluster configuration server
In a GPFS cluster, the node chosen to maintain the GPFS cluster configuration data.

private IP address
An IP address used to communicate on a private network.

public IP address
An IP address used to communicate on a public network.

Q

quorum node
A node in the cluster that is counted to determine whether a quorum exists.

quota
The amount of disk space and number of inodes assigned as upper limits for a specified user, group of users, or fileset.

quota management
The allocation of disk blocks to the other nodes writing to the file system, and comparison of the allocated space to quota limits at regular intervals.

R

Redundant Array of Independent Disks (RAID)
A collection of two or more disk physical drives that present to the host an image of one or more logical disk drives. In the event of a single physical device failure, the data can be read or regenerated from the other disk drives in the array due to data redundancy.

recovery
The process of restoring access to file system data when a failure has occurred. Recovery can involve reconstructing data or providing alternative routing through a different server.

remote key management server (RKM server)
A server that is used to store master encryption keys.

replication
The process of maintaining a defined set of data in more than one location. Replication consists of copying designated changes for one location (a source) to another (a target) and synchronizing the data in both locations.

RKM server
See *remote key management server*.

rule
A list of conditions and actions that are triggered when certain conditions are met. Conditions include attributes about an object (file name, type or extension, dates, owner, and groups), the requesting client, and the container name associated with the object.

S

SAN-attached
Disks that are physically attached to all nodes in the cluster using Serial Storage Architecture (SSA) connections or using Fibre Channel switches.

Scale Out Backup and Restore (SOBAR)
A specialized mechanism for data protection against disaster only for GPFS file systems that are managed by IBM Storage Protect for Space Management.

secondary GPFS cluster configuration server
In a GPFS cluster, the node chosen to maintain the GPFS cluster configuration data in the event that the primary GPFS cluster configuration server fails or becomes unavailable.

Secure Hash Algorithm digest (SHA digest)
A character string used to identify a GPFS security key.

session failure
The loss of all resources of a data management session due to the failure of the daemon on the session node.

session node
The node on which a data management session was created.

Small Computer System Interface (SCSI)
An ANSI-standard electronic interface that allows personal computers to communicate with peripheral hardware, such as disk drives, tape drives, CD-ROM drives, printers, and scanners faster and more flexibly than previous interfaces.

snapshot
An exact copy of changed data in the active files and directories of a file system or fileset at a single point in time. See also *fileset snapshot*, *global snapshot*.

source node
The node on which a data management event is generated.

stand-alone client
The node in a one-node cluster.

storage area network (SAN)
A dedicated storage network tailored to a specific environment, combining servers, storage products, networking products, software, and services.

storage pool
A grouping of storage space consisting of volumes, logical unit numbers (LUNs), or addresses that share a common set of administrative characteristics.

stripe group
The set of disks comprising the storage assigned to a file system.

striping
A storage process in which information is split into blocks (a fixed amount of data) and the blocks are written to (or read from) a series of disks in parallel.

subblock
The smallest unit of data accessible in an I/O operation, equal to one thirty-second of a data block.

system storage pool
A storage pool containing file system control structures, reserved files, directories, symbolic links, special devices, as well as the metadata associated with regular files, including indirect blocks and extended attributes. The `system storage pool` can also contain user data.

T

token management
A system for controlling file access in which each application performing a read or write operation is granted some form of access to a specific block of file data. Token management provides data consistency and controls conflicts. Token management has two components: the token management server, and the token management function.

token management function
A component of token management that requests tokens from the token management server. The token management function is located on each cluster node.

token management server
A component of token management that controls tokens relating to the operation of the file system. The token management server is located at the file system manager node.

transparent cloud tiering (TCT)
A separately installable add-on feature of IBM Storage Scale that provides a native cloud storage tier. It allows data center administrators to free up on-premise storage capacity, by moving out cooler data to the cloud storage, thereby reducing capital and operational expenditures.

twin-tailed
A disk connected to two nodes.

U

user storage pool
A storage pool containing the blocks of data that make up user files.

V

VFS
See *virtual file system*.

virtual file system (VFS)
A remote file system that has been mounted so that it is accessible to the local user.

virtual node (vnode)
The structure that contains information about a file system object in a virtual file system (VFS).

W

watch folder API
Provides a programming interface where a custom C program can be written that incorporates the ability to monitor inode spaces, filesets, or directories for specific user activity-related events within IBM Storage Scale file systems. For more information, a sample program is provided in the following directory on IBM Storage Scale nodes: `/usr/lpp/mmfs/samples/util` called tswf that can be modified according to the user's needs.

Index

SC28-3113-19