IBM Spectrum Scale Container Native Storage Access
5.1.7

IBM Spectrum Scale Container Native Storage Access Guide

SC28-3168-21

Note

Before using this information and the product it supports, read the information in "Notices" on page 117.

This edition applies to Version 5 release 1 modification 7 of the following products, and to all subsequent releases and modifications until otherwise indicated in new editions:

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

© Copyright International Business Machines Corporation 2020, 2023.
US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Tables

About this information

This edition applies to IBM Spectrum Scale version 5.1.7 for AIX®, Linux®, and Windows.

IBM Spectrum 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 Spectrum Scale is running on a particular AIX node, enter:

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

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

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

The IBM Spectrum 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 Spectrum 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.

© Copyright IBM Corp. 2020, 2023

Table 1. IBM Spectrum Scale library information units

Information unit	**Type of information**	**Intended users**
IBM Spectrum Scale: Concepts, Planning, and Installation Guide	This guide provides the following information: **Product overview** • Overview of IBM Spectrum 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 Spectrum Scale • Introduction to IBM Spectrum Scale GUI • IBM Spectrum Scale management API • Introduction to Cloud services • Introduction to file audit logging • Introduction to clustered watch folder • Understanding call home • IBM Spectrum Scale in an OpenStack cloud deployment • IBM Spectrum Scale product editions • IBM Spectrum Scale license designation • Capacity-based licensing	System administrators, analysts, installers, planners, and programmers of IBM Spectrum Scale clusters who are very experienced with the operating systems on which each IBM Spectrum Scale cluster is based

Table 1. IBM Spectrum Scale library information units (continued)

Information unit	**Type of information**	**Intended users**
IBM Spectrum Scale: Concepts, Planning, and Installation Guide	**Planning** • Planning for GPFS • Planning for protocols • Planning for Cloud services • Planning for IBM Spectrum Scale on Public Clouds • Planning for AFM • Planning for AFM DR • Planning for AFM to cloud object storage • Planning for performance monitoring tool	
IBM Spectrum 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 Spectrum Scale library information units (continued)

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

Table 1. IBM Spectrum Scale library information units (continued)

Information unit	Type of information	Intended users
IBM Spectrum Scale: Concepts, Planning, and Installation Guide	• Upgrading IBM Spectrum® Scale non-protocol Linux nodes • Upgrading IBM Spectrum Scale protocol nodes • Upgrading GPUDirect Storage • Upgrading AFM and AFM DR • Upgrading object packages • Upgrading SMB packages • Upgrading NFS packages • Upgrading call home • Manually upgrading the performance monitoring tool • Manually upgrading pmswift • Manually upgrading the IBM Spectrum 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 Spectrum Scale components with the installation toolkit • Protocol authentication configuration changes during upgrade • Changing the IBM Spectrum Scale product edition • Completing the upgrade to a new level of IBM Spectrum Scale • Reverting to the previous level of IBM Spectrum Scale	System administrators, analysts, installers, planners, and programmers of IBM Spectrum Scale clusters who are very experienced with the operating systems on which each IBM Spectrum Scale cluster is based

Table 1. IBM Spectrum Scale library information units (continued)

Information unit	Type of information	Intended users
IBM Spectrum Scale: Concepts, Planning, and Installation Guide	• Coexistence considerations • Compatibility considerations • Considerations for IBM Spectrum Protect for Space Management • Applying maintenance to your IBM Spectrum Scale system • Guidance for upgrading the operating system on IBM Spectrum Scale nodes • Considerations for upgrading from an operating system not supported in IBM Spectrum Scale 5.1.x.x • Servicing IBM Spectrum Scale protocol nodes • Offline upgrade with complete cluster shutdown	

Table 1. IBM Spectrum Scale library information units (continued)		
Information unit	**Type of information**	**Intended users**
IBM Spectrum Scale: Administration Guide	This guide provides the following information: **Configuring** • Configuring the GPFS cluster • Configuring GPUDirect Storage for IBM Spectrum 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 Spectrum Scale • Configuring file audit logging • Configuring clustered watch folder • 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 Spectrum Scale Cinder driver with Red Hat OpenStack Platform 16.1 • Configuring Multi-Rail over TCP (MROT)	System administrators or programmers of IBM Spectrum Scale systems

Table 1. IBM Spectrum Scale library information units (continued)

Information unit	**Type of information**	**Intended users**
IBM Spectrum Scale: Administration Guide	**Administering** • Performing GPFS administration tasks • Performing parallel copy with mmxcp command • Protecting file data: IBM Spectrum Scale safeguarded copy • Verifying network operation with the mmnetverify command • Managing file systems • File system format changes between versions of IBM Spectrum Scale • Managing disks	System administrators or programmers of IBM Spectrum Scale systems

Table 1. IBM Spectrum Scale library information units (continued)

Information unit	**Type of information**	**Intended users**
IBM Spectrum 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 Spectrum 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 • Cloud services: Transparent cloud tiering and Cloud data sharing • Managing file audit logging • RDMA tuning • Configuring Mellanox Memory Translation Table (MTT) for GPFS RDMA VERBS Operation • Administering AFM • Administering AFM DR	System administrators or programmers of IBM Spectrum Scale systems

Table 1. IBM Spectrum Scale library information units (continued)

Information unit	Type of information	Intended users
IBM Spectrum 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 Spectrum Scale systems

Table 1. IBM Spectrum Scale library information units (continued)

Information unit	**Type of information**	**Intended users**
IBM Spectrum Scale: Problem Determination Guide	This guide provides the following information: **Monitoring** • Monitoring system health by using IBM Spectrum Scale GUI • Monitoring system health by using the mmhealth command • 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 Spectrum 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 Spectrum Scale: Concepts, Planning, and Installation Guide*

Table 1. IBM Spectrum Scale library information units (continued)

Information unit	Type of information	Intended users
IBM Spectrum 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 Spectrum Scale library information units (continued)

Information unit	Type of information	Intended users
IBM Spectrum 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 Spectrum Scale systems • Application programmers who are experienced with IBM Spectrum Scale systems and familiar with the terminology and concepts in the XDSM standard

Table 1. IBM Spectrum Scale library information units (continued)

Information unit	Type of information	Intended users
IBM Spectrum 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 Spectrum Scale systems • Application programmers who are experienced with IBM Spectrum Scale systems and familiar with the terminology and concepts in the XDSM standard

Table 1. IBM Spectrum Scale library information units (continued)

Information unit	Type of information	Intended users
IBM Spectrum 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 Spectrum Scale systems • Application programmers who are experienced with IBM Spectrum Scale systems and familiar with the terminology and concepts in the XDSM standard

Table 1. IBM Spectrum Scale library information units (continued)		
Information unit	**Type of information**	**Intended users**
IBM Spectrum 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 Spectrum Scale Data Management API for GPFS information • GPFS programming interfaces • GPFS user exits • IBM Spectrum Scale management API endpoints • Considerations for GPFS applications	• System administrators of IBM Spectrum Scale systems • Application programmers who are experienced with IBM Spectrum Scale systems and familiar with the terminology and concepts in the XDSM standard

Table 1. IBM Spectrum Scale library information units (continued)

Information unit	**Type of information**	**Intended users**
IBM Spectrum 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 Spectrum 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 Spectrum Scale Hadoop performance tuning guide • Overview • Performance overview • Hadoop Performance Planning over IBM Spectrum Scale • Performance guide	• System administrators of IBM Spectrum Scale systems • Application programmers who are experienced with IBM Spectrum Scale systems and familiar with the terminology and concepts in the XDSM standard

Table 1. IBM Spectrum Scale library information units (continued)

Information unit	**Type of information**	**Intended users**
IBM Spectrum 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 Spectrum Scale systems • Application programmers who are experienced with IBM Spectrum Scale systems and familiar with the terminology and concepts in the XDSM standard
IBM Spectrum 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 Spectrum Scale systems • Application programmers who are experienced with IBM Spectrum Scale systems and familiar with the terminology and concepts in the XDSM standard

Table 1. IBM Spectrum Scale library information units (continued)

Information unit	**Type of information**	**Intended users**
IBM Spectrum Scale Erasure Code Edition Guide	IBM Spectrum Scale Erasure Code Edition • Summary of changes • Introduction to IBM Spectrum Scale Erasure Code Edition • Planning for IBM Spectrum Scale Erasure Code Edition • Installing IBM Spectrum Scale Erasure Code Edition • Uninstalling IBM Spectrum Scale Erasure Code Edition • Creating an IBM Spectrum Scale Erasure Code Edition storage environment • Using IBM Spectrum Scale Erasure Code Edition for data mirroring and replication • Upgrading IBM Spectrum Scale Erasure Code Edition • Incorporating IBM Spectrum Scale Erasure Code Edition in an Elastic Storage Server (ESS) cluster • Incorporating IBM Elastic Storage Server (ESS) building block in an IBM Spectrum Scale Erasure Code Edition cluster • Administering IBM Spectrum Scale Erasure Code Edition • Troubleshooting • IBM Spectrum Scale RAID Administration	• System administrators of IBM Spectrum Scale systems • Application programmers who are experienced with IBM Spectrum Scale systems and familiar with the terminology and concepts in the XDSM standard

Table 1. IBM Spectrum Scale library information units (continued)

Information unit	Type of information	Intended users
IBM Spectrum Scale Container Native Storage Access	This guide provides the following information: • Overview • Planning • Installation prerequisites • Installing the IBM Spectrum Scale container native operator and cluster • Upgrading • Configuring IBM Spectrum Scale Container Storage Interface (CSI) driver • Using IBM Spectrum Scale GUI • Maintenance of a deployed cluster • Cleaning up the container native cluster • Monitoring • Troubleshooting • References	• System administrators of IBM Spectrum Scale systems • Application programmers who are experienced with IBM Spectrum Scale systems and familiar with the terminology and concepts in the XDSM standard
IBM Spectrum Scale Data Access Service	This guide provides the following information: • Release notes • Product overview • Planning • Installing • Upgrading • Administering • Monitoring • Troubleshooting • Command reference (mmdas command) • Programming reference (REST APIs)	• System administrators of IBM Spectrum Scale systems • Application programmers who are experienced with IBM Spectrum Scale systems and familiar with the terminology and concepts in the XDSM standard

Table 1. IBM Spectrum Scale library information units (continued)

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

Prerequisite and related information

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

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

Conventions used in this information

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

Note: Users of IBM Spectrum 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.

Table 2. Conventions (continued)

Convention	Usage
`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.
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 Spectrum 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 Spectrum Scale development organization, send your comments to the following e-mail address:

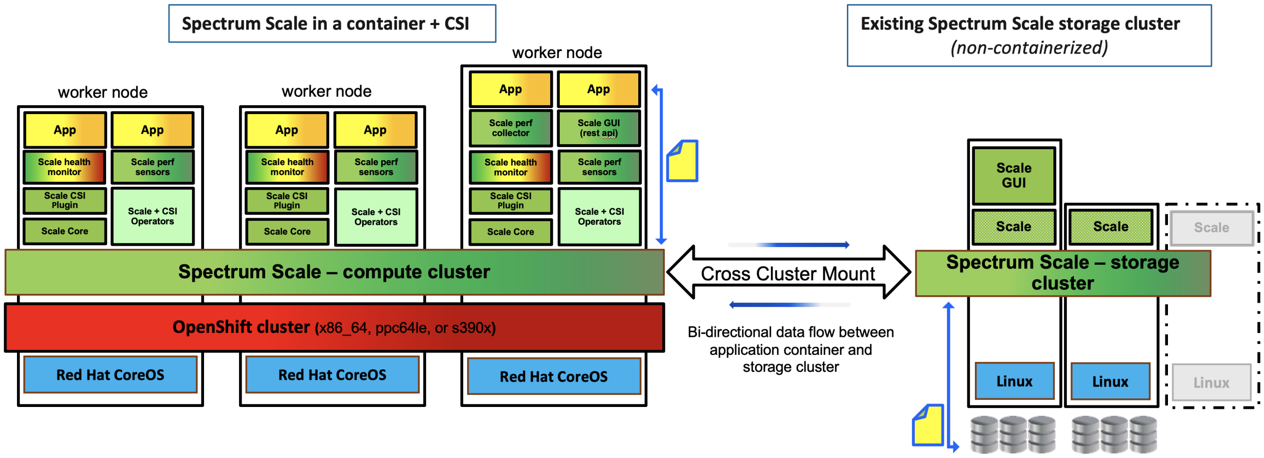
`scale@us.ibm.com`

Chapter 1. Overview

IBM Spectrum Scale container native is a containerized version of IBM Spectrum Scale. IBM Spectrum 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 about IBM Spectrum Scale features, see Product overview in IBM Spectrum Scale documentation.

IBM Spectrum Scale container native allows the deployment of the cluster file system in a Red Hat OpenShift cluster. Using a remote mount attached file system, the container native deployment provides a persistent data store to be accessed by the applications through the IBM Spectrum Scale CSI driver by using Persistent Volumes (PVs). For more information, see IBM Spectrum Scale Container Storage Interface Driver in IBM Spectrum Scale CSI documentation.

Figure 1. Remote mount



For more information about direct storage attachment, see "Deployment considerations" on page 9.

What's new?

The following enhancements are made in this release:

- Ability to upgrade IBM Spectrum Scale container native from 5.1.6.x to 5.1.7.0. For more information, see Chapter 8, "Upgrading," on page 73.

  Note: If upgrading from IBM Spectrum Scale container native less than 5.1.5.0, it is required to first upgrade to IBM Spectrum Scale container native 5.1.5.0 before continuing to higher levels.

- Support for IBM Spectrum Scale Container Storage Interface (CSI) 2.9.0. For more information, see What's New in IBM Spectrum Scale CSI documentation.
- Support for IBM Spectrum Scale Data Access Services (DAS) 5.1.7. For more information, see *Release Notes* in DAS documentation.
- Support for Red Hat OpenShift 4.12.
- Scalability improvements for Security-Enhanced Linux (SELinux) relabeling on Container Storage Interface (CSI) volume attachment. For more information, see "IBM Spectrum Scale container native and SELinux" on page 15.

© Copyright IBM Corp. 2020, 2023

- Ability to install IBM Spectrum Scale container native on Red Hat OpenShift Service on AWS (ROSA) with a remote mount to an IBM Spectrum Scale cloudkit installed storage cluster. For more information, see Installing IBM Spectrum Scale on public cloud with the cloud kit.
- Specific placement of IBM Spectrum Scale container native core pods using custom node selector.

  Important: The sample cluster CR `spec.daemon.nodeSelector` is changed from using the kubernetes worker role, `node-role.kubernetes.io/worker: ""` to using a custom label `scale.spectrum.ibm.com/daemon-selector: ""`.
- Cluster manifest is split to help facilitate ease of installation. For more information, see "Install" on page 46.
- Reduced memory footprint of IBM Spectrum Scale container native operator.
- Memory profiling for the IBM Spectrum Scale container native operator.
- Memory profiling included in must-gather.
- Support of custom tolerations for IBM Spectrum Scale container native core pods. For more information, see "Tolerations" on page 50.
- Update of remote mount access key on IBM Spectrum Scale storage cluster. For more information, see "Updating remote mount access key on IBM Spectrum Scale storage cluster" on page 85.

Note: If the storage cluster is running a GUI high availability configuration, for example, having 2 or more GUI nodes installed, ensure the storage cluster is running IBM Spectrum Scale 5.1.6.1 or higher prior to use with IBM Spectrum Scale container native and IBM Spectrum Scale CSI.

Supported features

IBM Spectrum Scale container native with Red Hat OpenShift Container Platform supports the following features:

- IBM Spectrum Scale node labels to establish node affinity
- Automated client-only cluster creation
- Automated remote file system mount for IBM Spectrum Scale Storage cluster
- Integrated IBM Spectrum Scale Container Storage Interface (CSI) driver for application persistent storage
- Automated deployment of IBM Spectrum Scale Container Storage Interface (CSI) driver
- IBM Spectrum Scale container native client cluster node expansion on Red Hat OpenShift Container Platform
- Cluster monitoring by using Red Hat OpenShift Container Platform Liveness and Readiness probe
- Call home
- Performance data collection
- Storage cluster encryption
- Rolling upgrade
- Automated IBM Spectrum Scale performance monitoring bridge for Grafana
- File audit logging (FAL)
- Compression
- Quotas on the storage cluster
- ACLs on the storage cluster
- ILM support on the storage cluster
- File clones on the storage cluster
- Snapshots on the storage cluster
- TCP/IP network connectivity among cluster nodes

- Direct storage attachment on s390x, x86, and power servers
- Automatic quorum selection is Kubernetes topology aware

IBM Spectrum Scale Data Access Services (DAS) supports the following features:

- S3 object access for Artificial Intelligence and analytics workloads
- Up to 10 TB single object size
- Up to 100 locally managed S3 accounts
- Up to 500 S3 buckets
- Up to 1,000,000 objects per S3 bucket
- Each IBM Spectrum Scale DAS cluster can be attached to one IBM Spectrum Scale storage cluster and to one IBM Spectrum Scale file system only.
- Each IBM Spectrum Scale storage cluster can be attached to one IBM Spectrum Scale DAS cluster only.

For more information, see *IBM Spectrum Scale Data Access Services*.

Limitations

- IBM Spectrum Scale container native currently supports only remote mount of the file system. It does not support local disks and NSD nodes.
- Deployment of IBM Spectrum Scale pods on master nodes is not supported with the exception of compact OpenShift clusters.
- Deployment of IBM Spectrum Scale pods on RHEL worker nodes is not supported.
- Deployment of IBM Spectrum Scale pods on nodes with ARM CPUs is not supported.
- Single node OpenShift clusters are not supported.

Scalability constraints

Table 3. Maximum Capacity Specification

Description	Max Supported
Number of worker nodes	128
Number of remote clusters	4
Number of remote file systems	16

Chapter 2. Planning

The planning for IBM Spectrum Scale container native includes the following topics:

Prerequisites

The planning process to install IBM Spectrum Scale on Red Hat OpenShift consists of many steps.

These steps are built on top of each other, so it is critical to follow the sequence defined in the following sections. Before you begin installation, there are several things that needs to be considered. The list of questions provided helps you to be prepared for the procedure.

- What version of Red Hat OpenShift Container Platform do you need?
- What are the hardware requirements?
- Have the necessary ports been opened?
- Is the Red Hat OpenShift Container Platform cluster in a restricted network environment?
- What is the minimum level of IBM Spectrum Scale that is needed on the storage cluster?

Preparations for deploying the IBM Spectrum Scale container native cluster

The following section summarizes the prerequisites required before deploying the IBM Spectrum Scale container native cluster:

- Validate that the OpenShift cluster, or the node from where you are managing the OpenShift cluster, has access to the manifest files in IBM Spectrum Scale container native repository of GitHub.

  For more information, see IBM Spectrum Scale container native repository on GitHub.

  Note: GitHub YAML manifests are inline with the Installation steps and are either accessed directly or pulled through `curl` through an existing internet connection. If an air gapped environment is running, the manifest files must be made locally available for use.
- Validate and apply the configuration to the Red Hat OpenShift installation settings.
- Obtain IBM Cloud Container Registry entitlement key to access the container images of IBM Spectrum Scale container native.
- If you are in a restricted network environment, then mirror the container images of IBM Spectrum Scale container native into a site-managed private image registry.
- Create an OpenShift global pull secret for the image registry that the cluster uses (either IBM Cloud Container Registry or private image registry).

Deploying the IBM Spectrum Scale container native cluster

To deploy a cluster, complete the following steps:

1. Create the IBM Spectrum Scale container native and IBM Spectrum Scale CSI operators by deploying the operator installer file `install.yaml` from GitHub.
2. Download the sample `cluster.yaml` CR from the GitHub repository, make necessary change, and apply to the cluster.

   i. Specify the IBM Spectrum Scale Edition in the license field and accept the license.

   ii. Configure appropriate node selectors for the IBM Spectrum Scale container native deployment.

   iii. Configure host aliases or ensure that the proper DNS is configured for your environment to allow for communication to storage cluster.

   iv. Configure Ephemeral Port Range, if necessary.

© Copyright IBM Corp. 2020, 2023

v. Enable the optional Grafana Bridge.

3. Download the sample `callhome.yaml` CR from GitHub, make necessary changes, and apply to the cluster.
4. Download the sample `remotecluster.yaml` CR from GitHub, make necessary changes, and apply to the cluster.
5. Download the sample `filesystem.remote.yaml` CR from GitHub, make necessary changes, and apply to the cluster.
6. If accessing encrypted data on the storage cluster, download the sample `encryptionconfig.remote.yaml` CR from GitHub, make necessary changes, and apply to the cluster..
7. Complete the storage cluster configuration.

   a. Create a GUI user on the storage cluster with the `ContainerOperator` role.

   b. Create a GUI user on the storage cluster with the `CsiAdmin` role.

   c. Configure CSI prerequisites on storage cluster.
8. Create a secret by using the storage cluster GUI user credentials for the `ContainerOperator` GUI user in the `ibm-spectrum-scale` namespace.
9. Create a secret by using the storage cluster GUI user credentials for `CsiAdmin` GUI user in the `ibm-spectrum-scale-csi` namespace.
10. Create a storage class to create volumes to use with your container native cluster.

Hardware requirements

The following sections describe hardware requirements to consider when deploying IBM Spectrum Scale container native.

Network

- All nodes in a compute cluster must be able to communicate with all nodes in a storage cluster.
- A minimum of 10 Gb network is needed but 40 - 100 Gb is recommended.
- RDMA for InfiniBand or RoCE for Ethernet is not supported.

Worker node requirements

IBM Spectrum Scale takes at least 2 GiB per node. Therefore, 8 GiB or more total memory is recommended for worker nodes.

IBM Spectrum Scale container native supports x86, ppc64le, and s390 CPU architectures. All nodes in the OpenShift cluster must have the same architecture. The ARM architecture is not supported.

IBM Spectrum Scale container native deploys several pods in the cluster. The following table shows the resource consumption of those pods.

Table 4. Hardware requirements

Pods	Where deployed	CPU request	Memory request	Storage	Description
core (created with the k8s Node shortname)	Nodes labeled with nodeSelector in cluster CR	If >=1000mCPU, will request 25% of capacity. Override to "2" CPU in sample CR client role	If >=2GiB, will request 25% of capacity. Override to "4Gi" in sample CR in client role	Config in /var (~25GiB)	This is the pod that provides the filesystem service for the node. It is required to be deployed on all nodes where PVs are accessed from application pods. The CPU and memory requests can be customized in the cluster CR.
operator	Single Node	100mCPU	40MiB	-	The controller runtime that manages all custom resources.
gui	Two Nodes	630mCPU	1.25GiB	Local PV for DB	The graphical user interface and ReST API.
pmcollector	Two Nodes	120 mCPU	3-7GiB depending on cluster size	Local PV for DB	The performance collector database.
grafana-bridge	Single Node	100mCPU	1GiB	-	The bridge for accessing pmcollector from grafana.

Note: The shown values are requests. For more information, see Kubernetes resource management in Kubernetes documentation. The limits are higher. This means that for CPU the pods might have bursts with more CPU usage at times where the CPU has free cycles. For memory the pods should not exceed their request significantly.

By default the core fs pods take 25% of the worker node capacity. This might be oversized in many applications. For more information about configuring the requests for both CPU and memory, see Cluster Custom Resource.

- Allocating more resources to IBM Spectrum Scale will result in better storage performance.
- Allocating less will allow more applications to be scheduled on a node.

For CPU, allocation can be reduced if the core fs pods consistently stay below the request. This can be monitored on the OpenShift console. When going too low, the filesystem daemon might starve on CPU cycles which destabilizes the whole cluster and can result in outages. For memory there is no real monitoring, allocating more will result in more data being cached which can boost performance. But this will be only seen indirectly by observing application performance.

Note: The CPU request can be dialed down below the 1000mCPU minimum. Your system might run just fine with, for example, 100mCPU. But, if a service ticket is opened for an issue that might be in any way related to this setting you will be asked to go up to 1000mCPU. The ticket is accepted only if the problem keeps showing up. Examples for related issues are, node expells, lag on PV creation in CSI, slow policy runs, bad performance, long waiters, etc.

The OpenShift console will report all worker nodes as overcommitted. The reason is that the CPU and memory limits of the pods add up to more than the total capacity of the node. Pods are scheduled based on their requests and the scheduler ensures that nodes will not be overcommitted in this regard. Higher limits allow pods to use resources that are free at the moment, but only the requested resources are guaranteed to them by Kubernetes. For more information about pods scheduling, see Kubernetes resource management.

This list does not include pods of the CSI driver which come on top of this. For more information, see IBM Spectrum Scale CSI documentation.

Software requirements

Use the following table to determine the software requirement levels for each release:

Table 5. Software requirements

IBM Spectrum Scale container native	IBM Spectrum Scale Container Storage Interface	Architecture	IBM Spectrum Scale remote storage cluster level	File system version cannot be newer than	OpenShift Container Platform level	Red Hat CoreOS	UBI level
5.1.7.0	2.9.0	x86,ppc64le,x390x	5.1.3.0+	31.00	4.10, 4.11,4.12	4.10, 4.11,4.12	8.7
5.1.6.0	2.8.0	x86,ppc64le,x390x	5.1.3.0+	30.00	4.9, 4.10, 4.11	4.9, 4.10, 4.11	8.6
5.1.5.0	2.7.0	x86,ppc64le,x390x	5.1.3.0+	29.00	4.9, 4.10, 4.11	4.9, 4.10, 4.11	8.6
5.1.4.0	2.6.0	x86,ppc64le,x390x	5.1.3.0+	28.00	4.9, 4.10	4.9, 4.10	8.6
5.1.3.0	2.5.0	x86,ppc64le,x390x	5.1.3.0+	27.00	4.9, 4.10	4.9, 4.10	8.5

Note:

The storage cluster is supported to be down-level from the IBM Spectrum Scale container native cluster, but it is ideal that the versions match. CSI functionality is highly dependent upon the IBM Spectrum Scale release, filesystem level, and version, installed on the storage cluster. If the storage cluster is running an earlier version, some functionality may not be available. For more information about CSI features and required levels, see *Table 1 in Hardware and Software Requirements* in IBM Spectrum Scale CSI documentation. For more information about compatibility and software matrix, see Section 17.3 in IBM Spectrum Scale FAQ documentation.

IBM Spectrum Scale Container Storage Interface (CSI)

- CSI 2.9.0 is installed in conjunction with IBM Spectrum Scale container native 5.1.7.0.

Storage cluster

- The storage cluster must be at IBM Spectrum Scale 5.1.1.2 or later.
- To take advantage of all functions provided by IBM Spectrum Scale Container Storage Interface Driver (CSI) 2.9, the storage cluster must be at IBM Spectrum Scale 5.1.7.0 or later and the file system must be at file system format level 31.00. For earlier levels restrictions may apply. For more information, see *Hardware and software requirements* in IBM Spectrum Scale CSI documentation.

  For more information, see Upgrading multi-cluster environments in the IBM Spectrum Scale documentation.
- Determine if the storage cluster is running a GUI high availability configuration, for example, having 2 or more GUI nodes installed.

  On the storage cluster, issue the following command. If two or more GUI nodes are displayed, then the storage cluster is running a GUI high availability configuration.

  ```
  /usr/lpp/mmfs/gui/cli/lsnode
  ```

  Note: If the storage cluster is running a GUI high availability configuration, ensure the storage cluster is running IBM Spectrum Scale 5.1.6.1 or higher prior to use with IBM Spectrum Scale container native and IBM Spectrum Scale CSI.

  For more information, see Ensuring high availability of the GUI service in IBM Spectrum Scale documentation.
- Enable the `--auto-inode-limit` parameter on the remotely mounted file system.

  For more information about `auto-inode-limit` parameter, see mmchfs command in IBM Spectrum Scale documentation.

  Note: The `--auto-inode-limit` option is available only with file system format level 28.00 or later.
- Encrypted file systems are supported. Configure the EncryptionConfig custom resource with the necessary key client and key server information. For more information, see EncryptionConfig.

External container images

There are some external container images that are required to run IBM Spectrum Scale container native. If running in an air gap environment, these images are required for successful deployment. For more information, see Container image list for IBM Spectrum Scale container native.

Auxiliary helper applications

- `curl` is used to retrieve some files required for the IBM Spectrum Scale container native installation.
- `jq` 1.5+ is used to help parse and format json output.

Deployment considerations

Before deployment, ensure that you are aware of the Red Hat OpenShift version, cluster network, persistent storage, and the IBM Spectrum Scale storage cluster considerations.

Red Hat OpenShift cluster considerations

The following list includes the Red Hat OpenShift cluster considerations:

- A minimum configuration of three master nodes and three worker nodes, with a maximum of 128 worker nodes is required.

- Deploying IBM Spectrum Scale pods on master nodes is not supported. An exception is when deploying in a compact cluster configuration. For more information, see Compact Cluster Support.
- Single node Red Hat OpenShift clusters are not supported, simply access data on a IBM Spectrum Scale storage cluster through NFS.
- Red Hat Enterprise Linux CoreOS (RHCOS) restricts new file system mounts to the `/mnt` subtree. IBM Spectrum Scale can mount any file system under `/mnt` on the Red Hat OpenShift cluster regardless of the default mount point that is defined on the storage cluster.

Red Hat OpenShift cluster network considerations

IBM Spectrum Scale container native comes with a collection of different pods. A subset of these pods can be considered regular pods that behave like typical application pods. Those pods are the operator, the GUI pods, and the performance data collector pods. The exception is what we refer to as the core pods as they provide the actual filesystem services. The core pods are not controlled by the Kubernetes scheduler through a regular DaemonSet. Instead, the IBM Spectrum Scale container native operator handles the management of those pods.

- The filesystem daemon running inside the core pods requires a static IP address for communication between daemons on different nodes.
- All core pods must be able to communicate with each other through the chosen network.

There are two network configurations that can be employed: host network or Container Network Interface (CNI) network. Only one network configuration can be chosen.

Host network

By default, the IBM Spectrum Scale pods use the host network. While this is the simplest configuration, it has some disadvantages:

- Using the host network breaks the network isolation that usually comes with containers. For example, any network port opened by IBM Spectrum Scale may conflict with a network port opened by another component on the host.
- Security features, like network policies, are not available for the host network.
- If the node has multiple network adapters, there is no way to select the adapter. Host network will always use the network adapter the worker node IP is assigned to.

Container Network Interface (CNI) network

As an alternative to host network, IBM Spectrum Scale can use a CNI network. There is more configuration effort to set up the CNI:

- In this configuration, core pods will have an IP address on the usual OpenShift SDN and another one on the CNI network.
  - OpenShift SDN will be used for communication with other pods.
  - CNI network will be used for communication between filesystem daemons, both inter-cluster and with the storage cluster.
- If the node is equipped with high speed network, the CNI should be attached to that.
  - This will be the daemon network where the filesystem I/O runs on. High bandwidth and low latency are highly beneficial for performance.
- The CNI network will be used exclusively by IBM Spectrum Scale and eliminates the potential for port conflicts with other components.
- Security features like network policies work on `MACVLAN` CNIs.
- The DNS must be configured properly to allow the worker nodes the ability to resolve the storage cluster nodes.
  - For more information, see Host aliases.

Note: Advanced features of `SR-IOV` type CNIs, such as RDMA and GPUdirect, are not yet supported.

For more information about configuring CNI with IBM Spectrum Scale container native, see Container network interface (CNI) configuration.

Red Hat OpenShift cluster persistent storage considerations

The following list includes the Red Hat OpenShift cluster persistent storage considerations:

- The IBM Spectrum Scale pods use host path mounts to store IBM Spectrum Scale cluster metadata and various logs.
- The IBM Spectrum Scale container native operator creates two local PersistentVolumes (PVs) on two eligible worker nodes. At least 25 GB free space must be available in the file system that contains the `/var` directory on all eligible worker nodes to avoid potential failures during the deployment. These PVs are created with the ReadWriteOnce (RWO) access mode.
- Both the host path mounts and local PVs are not automatically cleaned up when you delete the associated IBM Spectrum Scale container native cluster. You must manually clean these up. For more information about cleaning up the persistent storage, see “Cleanup OpenShift nodes” on page 89 and “Cleanup IBM Spectrum Scale container native” on page 88.
- IBM Spectrum Scale container native does not support the use of dynamically created or pre-created PVs.

Enterprise grade image registry considerations

The following list includes the considerations for enterprise grade image registry:

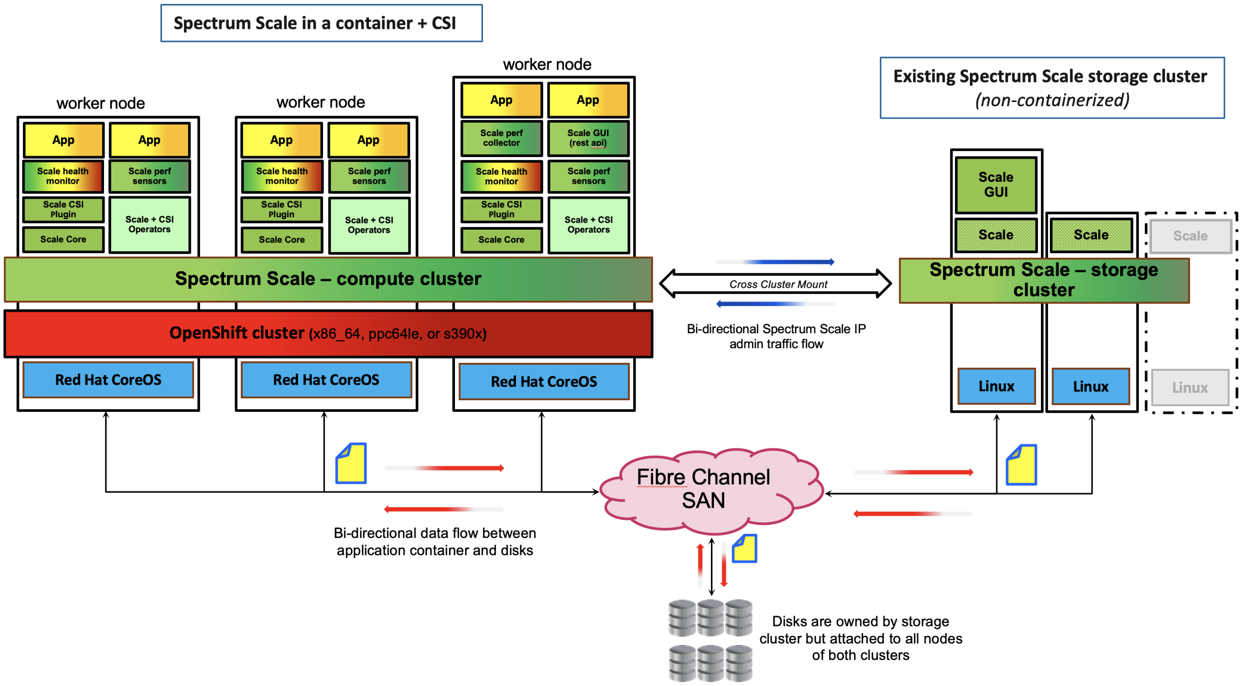
- In a restricted network environment where the Red Hat OpenShift Container Platform cluster cannot pull IBM Spectrum Scale images from the IBM Container Repository, images must be mirrored to a production grade enterprise image registry that the Red Hat OpenShift Container Platform cluster can access.
- In a restricted network environment, there must be a node that can communicate externally and also with the target Red Hat OpenShift Container Platform cluster.
- Any registry that is used for hosting the container images of IBM Spectrum Scale container native must not be accessible to external users. Also, it must be restricted to the service account used for IBM Spectrum Scale container native management. All users and machines that are accessing these container images must be authorized per IBM Spectrum Scale license agreement.

Direct storage attachment considerations

The following list includes the considerations for direct storage attachment:

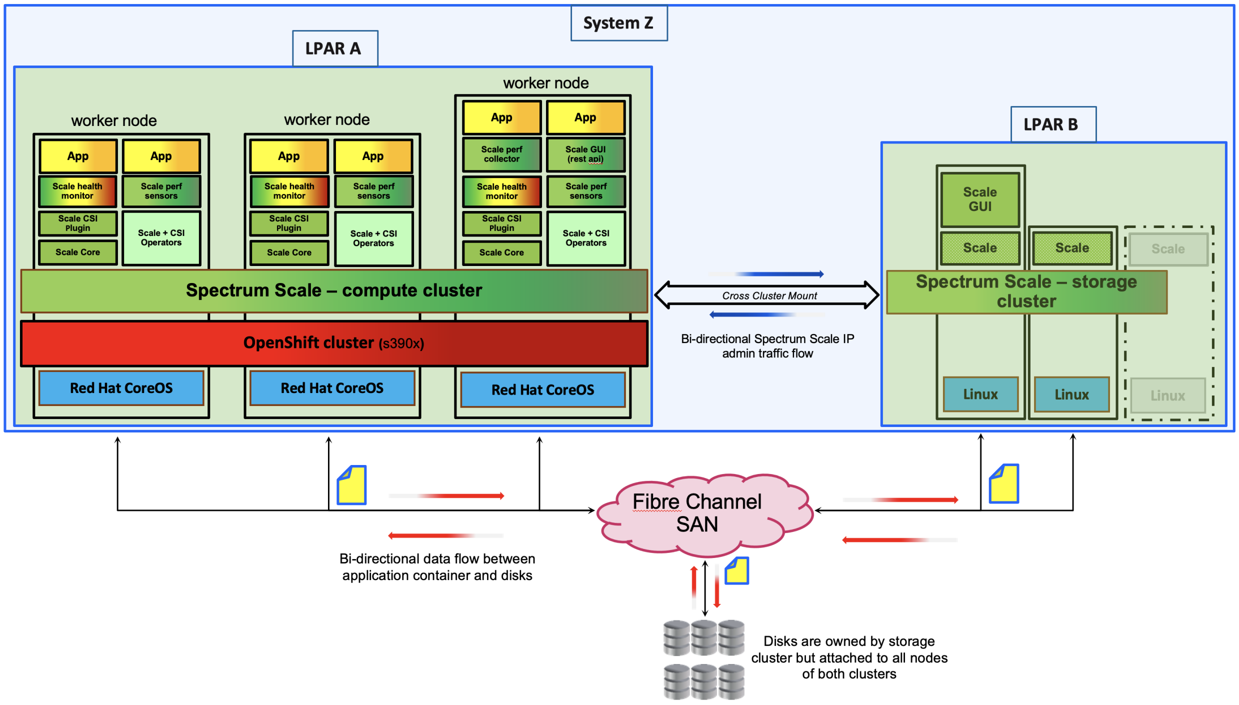
- Support for direct storage attachment on x86, power, and Z servers. In direct storage attachment configuration, the worker nodes use the SAN fabric instead of the IBM Spectrum Scale NSD protocol for I/O traffic.
- If using power or x86 servers, it might be necessary to load multi-path drivers through Red Hat CoreOS before storage can be seen.

  Figure 2. Direct attach config


Spectrum Scale in a container + CSI
worker node
worker node
worker node
App
Scale health monitor
Scale perf sensors
Scale perf collector
Scale GUI (rest api)
Scale CSI Plugin
Scale + CSI Operators
Scale Core
Spectrum Scale – compute cluster
OpenShift cluster (x86_64, ppc64le, or s390x)
Red Hat CoreOS
Existing Spectrum Scale storage cluster (non-containerized)
Scale GUI
Scale
Spectrum Scale – storage cluster
Linux
Cross Cluster Mount
Bi-directional Spectrum Scale IP admin traffic flow
Fibre Channel SAN
Bi-directional data flow between application container and disks
Disks are owned by storage cluster but attached to all nodes of both clusters

- The virtualization layers of an IBM Z server allow the physical connection of the disks containing the IBM Spectrum Scale file system data to both the storage cluster and the IBM Spectrum Scale container native cluster.
- For more information about setting up a direct storage attachment, see Attaching direct storage on IBM Z in IBM Spectrum Scale documentation.

Figure 3. Direct attach SystemZ


System Z
LPAR A
worker node
worker node
worker node
App
Scale health monitor
Scale perf sensors
Scale perf collector
Scale GUI (rest api)
Scale CSI Plugin
Scale + CSI Operators
Scale Core
Spectrum Scale – compute cluster
OpenShift cluster (s390x)
Red Hat CoreOS
LPAR B
Scale GUI
Scale
Spectrum Scale – storage cluster
Linux
Cross Cluster Mount
Bi-directional Spectrum Scale IP admin traffic flow
Fibre Channel SAN
Bi-directional data flow between application container and disks
Disks are owned by storage cluster but attached to all nodes of both clusters

Network and firewall requirements

Internal IBM Spectrum Scale container native core pod communication

IBM Spectrum Scale container native core pods use two network interfaces, an "admin" and a "daemon" interface. The "admin" interface is used for monitoring and management, and the "daemon" interface is used for file system I/O. In a default configuration, both networks use the Kubernetes node internal network that is exposed on the host, `hostNetwork: true`, and the pods are exposed to all host networking.

When using the default host networking, both the core pod daemon network requirements and the core pod admin network requirements, must be satisfied and communication verified to be open across the Kubernetes node internal network.

However, it is recommended to use the Container Network Interface (CNI) for the daemon network instead of host networking because it allows for better security and isolation. In the configurations using CNI, only core pod daemon network requirements are required for the daemon network configured with CNI. The core pod admin network requirements apply to the default pod network, which is automatically managed by IBM Spectrum Scale container native. For more information, see "Core pod daemon network requirements" on page 13 and "Core pod admin network requirements" on page 14.

Core pod daemon network requirements

The following list of ports and protocols are used to communicate between all nodes within the IBM Spectrum Scale container native cluster and between nodes in any configured remote storage clusters. Each node acts as a server that might initiate connections to any other node. Ensure that the list of ports and protocols is open for both inbound and outbound packet flows.

Port number	Protocol	Use	Initiated Direction
1191	TCP	GPFS	All node to all node
-	ICMP	GPFS	All node to all node
Configurable ephemeral port range	TCP	GPFS Policy Engine & compatibility with clusters less than 5.1.3	All node to all node

Ephemeral port ranges

Ephemeral ports are used by the GPFS Policy Engine, which is used by the CSI snapshot and compression features. When `tscCmdAllowRemoteConnections=yes` is configured in the Cluster CR, ephemeral port ranges are also used for communication and compatibility with remote clusters with a earlier version of IBM Spectrum Scale 5.1.3.

Note: If ephemeral ports are configured on the remote storage cluster, ensure that they are also configured in the Cluster CR for the IBM Spectrum Scale container native deployment.

For more information about how to set ephemeral port ranges, see Ephemeral port range.

Name resolution between a IBM Spectrum Scale container native cluster and a storage cluster

The IBM Spectrum Scale container native core pods require name resolution to all storage cluster nodes, specifically the node's daemon node name. If these names cannot be configured in the environment's domain name service (DNS), then host aliases may be added in the Cluster CR `.spec.daemon.hostAliases`. This configuration adds these host aliases to the internal DNS managed by IBM Spectrum Scale container native and applies only to name resolution performed by the IBM Spectrum Scale container native core pods.

Core pod admin network requirements

The following network requirements must be met when using the host networking, which is the default deployment method when Container Network Interface (CNI) is not used. These are only needed to be open between nodes within the local IBM Spectrum Scale container native cluster.

Port number	Protocol	Use	Initiated Direction
12345	TCP	Administrative SSH	all node to all node within local cluster

Other container native scale requirements

The following applies to the IBM Spectrum Scale container native operator and IBM Spectrum Scale CSI operator and driver. These components require name resolution for the host name(s) configured as Remote Cluster REST APIs.

Port number	Protocol	Use	Initiated Direction
443	TCP	Storage cluster REST API	pod network to storage cluster GUI nodes

Note: The pod network encapsulates the requirement that IBM Spectrum Scale container native operator and IBM Spectrum Scale CSI operator and driver require the ability to reach the storage cluster GUI nodes REST API. These respective pods use the default pod network and might schedule to any node selected.

Container Network Interface (CNI) configuration

IBM Spectrum Scale container native core pods uses two network interfaces, an `admin` and a `daemon` interface. The `admin` interface is used for monitoring and management, while the `daemon` interface is used for filesystem I/O. In a default configuration, both networks utilize the Kubernetes node internal network exposed on the host, and the pods are exposed to all host networking.

Configuring CNI provides advantages over the default configuration. The `admin` and `daemon` network interfaces become private to the IBM Spectrum Scale container native core pod. The `admin` interface moves to use the Kubernetes pod network, which includes security and isolation provided by Kubernetes and NetworkPolicies. The `daemon` interface, if configured by CNI, can be setup to be isolated, private, and high-speed.

When configuring custom network interfaces, CNI is the only supported method for IBM Spectrum Scale container native network interface. It provides security and isolation advantages above and beyond configuring network interfaces on host, for example, interfaces managed by the `nmstate` operator.

How to configure CNI

Complete the following steps:

1. To configure OpenShift for CNI, refer steps on Red Hat website.

   For more information, see Adding CNI networks in Red Hat OpenShift documentation.

   Considerations

   - If the node has only a single physical network attachment, then the network adapter needs to be shared between networks. There are several CNI flavors that allow this: `Bridge`, `IPVLAN`, and `MACVLAN`.

     – The `MACVLAN` CNI supports also network policies and should be the default choice.

   - If the node has multiple physical network attachments and you want to dedicate one of the physical networks to IBM Spectrum Scale, select `host-adapter` CNI. It will map a physical network adapter into a pod, making it inaccessible to the host and other pods.

- SR-IOV surpasses the capabilities of `host-adapter`. For more information, see About Single Root I/O Virtualization (SR-IOV) hardware networks in Red Hat OpenShift documentation. Advanced features like RDMA, GPUdirect, and bonding of network ports are accessible via `SR-IOV` hardware network.

  Note: The features RDMA, GPUdirect, and bonding of network ports are not currently supported by IBM Spectrum Scale. Also `SR-IOV` allows to partition the hardware adapter and hand those partitions to different pods. Configuration is more complex compared to other CNIs and choice of supported network adapters is limited. As of today, IBM Spectrum Scale is not tested with `SR-IOV`.
- The IP address mapping is required to be static. This can be achieved by setting up static IPs or by configuring DHCP static mapping. For remote mount of a filesystem from a IBM Spectrum Scale storage cluster, this network must be routed to the storage cluster's daemon network.

2. Configure each of the OpenShift nodes that comprise the IBM Spectrum Scale CNSA cluster:

   a. Create runtime configuration node annotation that has the CNI definition. The specific node annotations for IBM Spectrum Scale is `scale.spectrum.ibm.com/daemon-network`. The format of the CNI annotation value is formed in the same format of a single network as defined in the format specified by `k8s.v1.cni.cncf.io/networks`.

   Example:

   ```
   annotations:
   scale.spectrum.ibm.com/daemon-network: |-
       {
           "name": "daemon-network",
           "mac": "22:22:0a:11:37:b2",
           "ips": [
               "10.17.99.63"
           ]
       }
   ```

   b. `ips` field must be set as this is the static IP desired for this CNI network.

   c. `mac` field may be set if you use DHCP ipam (backed by a statically mapped DHCP). `ips` still must be set in addition to `mac`.

   Note: This might seem redundant, but IBM Spectrum Scale container native uses `ips` to set up its own name resolution. This process is asynchronous and independent of the pod actually being created. If `ips` was not set, then DHCP address would not be discovered until after pod creation.

IBM Spectrum Scale container native and SELinux

IBM Spectrum Scale container native offers a default for Container Storage Interface (CSI) volume attachment behavior for Security-Enhanced Linux (SELinux) labels. This default is introduced beginning with IBM Spectrum Scale container native 5.1.7.0 release.

What is SELinux?

SELinux, or Security-Enhanced Linux, defines access controls for the applications, processes, and files on a system. It uses security policies, which are a set of rules that tell SELinux what can or cannot be accessed, to enforce the access allowed by a policy.

For more information about SELinux, see What is SELinux? in Red Hat Documentation.

How is SELinux controlled in Kubernetes?

Container processes are started with the SELinux context in the `.spec.securityContext.seLinuxOptions` field, while CSI-based volumes have all their files labeled to match on pod attachment during container creation.

SELinux Multi-Category Security (MCS) is used in Red Hat OpenShift. By default, containers have their SELinux level set to values annotated on the namespace. The category defaults annotated on the namespace are assigned automatically by OpenShift to limit overlap.

Note: Red Hat OpenShift is based on Kubernetes.

Problems with SELinux relabel on volume attach

When attaching CSI-based volumes to pods in Kubernetes the container-runtime SELinux relabels all files in the volume. This is problematic for shared filesystems that support SELinux, such as the IBM Spectrum Scale filesystem.

Relabeling on volume attachment moves the security control to the consumer of file volumes instead of the owner of the files. In classic shared filesystem environments, such as IBM Spectrum Scale, access is controlled by node administrators and file owners. In Kubernetes, volume access is isolated to a namespaced volume claim and controlled using Kubernetes role-based access control (RBAC). While in classic shared filesystem environments, such as IBM Spectrum Scale, access is controlled by node administrators and file owners. This Kubernetes behavior makes it very difficult to maintain access controls for volumes and files that are shared outside of Kubernetes. This is because Kubernetes ignores and overwrites any SELinux security isolation set by external administrators.

In addition, relabeling all files in a volume is a non-trivial operation, which may generate a large I/O load on the backend storage system. This introduces a performance and denial of service concern. Volumes with too many files, or shared volumes that are attached concurrently, may easily swamp the storage subsystem and cause cascading pod creation timeouts.

Upstream Kubernetes limitation for SELinux relabel

The SELinux relabel issue is not unique to IBM Spectrum Scale. Upstream Kubernetes does not give CSI volume driver implementations control of SELinux relabel on volume attachment.

Legacy in-tree volume drivers, such as the in-tree NFS volume driver, have the ability to control SELinux relabeling. However, legacy in-tree volume drivers are deprecated.

In Red Hat OpenShift, the container-runtime that performs the SELinux relabel will skip the relabeling if the container SELinux context is set to `spc_t`, which is the super privileged container. It is an uncontained type and may access any file on the system allowed by standard file permissions.

Default volume attachment behavior by IBM Spectrum Scale container native

To prevent Kubernetes from relabeling SELinux file labels,IBM Spectrum Scale container native by default will mount the filesystem with a container permissive context. This disallows the `security.selinux` label from being set, and all files inside the filesystem will be considered to have the context defined on the filesystem mount. By default, that context would allow all containers running as `container_t` SELinux type to access files on the filesystem allowed by standard file permissions. By default, the mount context is set to `system_u:object_r:container_file_t:s0`.

Comparing SELinux relabel on attach and SELinux mount context

Security considerations

Setting a container permissive SELinux mount context or doing an SELinux relabel on volume attachment have similar access control within Kubernetes. Any container that can claim a volume may access its files.

However, if a process could escape a container or the host was able to be compromised, then any `container_t` constrained context would have access to the filesystem from a SELinux access perspective. Standard linux file permissions would still prevent access. This means containers that run within the restricted SecurityContextConstraint defined by OpenShift would not be able to access files in a volume that it did not explicitly share.

If a volume is shared with Kubernetes from an external application, then SELinux relabeling breaks SELinux security isolation managed externally. SELinux relabel on attach allows a volume consumer to ignore and overwrite SELinux labels, potentially exposing the files to other external systems. Setting the SELinux mount context will only allow the external SELinux labels to be ignored within the confines of the Kubernetes cluster.

Performance considerations

Since SELinux mount context disallows SELinux relabeling, volume attachments to pods do not generate extra I/O load and will attach faster than if SELinux relabeling is done.

If SELinux relabeling occurs, it must complete within a minute or the container creation will timeout and fail. Multiple relabel operations occurring concurrently are more likely to trigger timeout and failure. This cascading failure may only be exposed due to wider outages (node, cluster, lab), or maintenance.

External access considerations

External access refers to any components outside of the IBM Spectrum Scale container native cluster. This includes the storage cluster that is remotely mounted to the IBM Spectrum Scale container native cluster.

SELinux mount context means all new files are created without SELinux labels, and are considered `unlabeled_t`. This would include files created by application containers using IBM Spectrum Scale container native volumes. Generally, access to `unlabeled_t` files is only allowed by uncontained or more privileged process contexts. To access the unlabeled files, the external applications outside of Kubernetes must be given access to `unlabeled_t` via SELinux user policies, the files must be labeled manually, or the mount used by the application has a valid SELinux context configured.

The behavior without mount context, which does an SELinux relabel of all files, is similar in that the SELinux context of the container should be configured to match what is desired externally, or the external application should be granted access to files created by the container SELinux context.

How to change mount context

The mount context may be set on the Filesystem kind. This applies to all applications using volumes within the filesystem. Run `oc explain fs.spec.seLinuxOptions` command for more details.

When changing the SELinux context, applications with ReadWriteMany volumes running on nodes with mixed SELinux contexts may experience `Permission Denied` errors. Applications running on nodes with differing SELinux contexts may not have access to each other's files during the duration of the SELinux context update.

Note: Changing the `fs.spec.seLinuxOptions` field of a Filesystem kind will cause pods to restart, and the restart of the pod will cause a reboot of the node itself. Ensure that the proper maintenance window and precautions, similar to upgrade, have been taken prior to changing this field.

```
spec:
           seLinuxOptions:
             user: <user>
             role: <object>
             type: <type>
             level: <level>
```

Legacy relabel on attach behavior

Earlier behavior of relabel on attach is considered legacy and the support is limited for it. If this behavior is required, contact IBM support.

Related Links

- Red Hat solution - pods fail to start due to volumes with high volume counts in Red Hat Openshift
- Kubernetes enhancement 1710
  - 1710 SELinux relabeling documentation
- Red Hat OpenShift - Understanding host and VM security
- Red Hat OpenShift and SELinux

Roles and persona

Different roles, cluster roles, and levels of access are needed to deploy a fully functioning IBM Spectrum Scale container native cluster.

Personas

Red Hat OpenShift Cluster Administrator

A user with cluster administrator privileges is required to deploy the IBM Spectrum Scale container native cluster. Cluster admin privileges are required to deploy higher privilege items such as:

- Namespaces
- Custom Resource Definitions
- Cluster Roles and Cluster Role Bindings
- Security Context Constraints

Since the above items are created by the OpenShift Cluster Administrator, this allows the operator to run in a more restricted manner with minimal privilege.

IBM Spectrum Scale Storage Cluster Administrator

A user with privilege and access to configure the existing IBM Spectrum Scale storage cluster for remote access is also required for a successful deployment of an IBM Spectrum Scale container native cluster. Since the container native cluster utilizes remote mounts, the storage cluster admin must be able to execute commands against the storage cluster. For more information, see “IBM Spectrum Scale storage cluster” on page 27.

Lab Administrator

A user with access to customer infrastructure and network is required to ensure a successful IBM Spectrum Scale container native cluster deployment. All OpenShift nodes that comprise the IBM Spectrum Scale container native cluster must be able to communicate with the remote Spectrum Scale storage cluster. This may require a Lab Administrator to tune the customer network firewall to allow such communications.

Operator permissions

The IBM Spectrum Scale container native operator is a cluster-scoped operator. The operator watches all namespaces on the OpenShift cluster it is deployed into. Since the operator is cluster scoped, it requires access to cluster level resources to successfully deploy. Access to cluster level resources is handled through a cluster role that is deployed via RBAC YAML files. The cluster role is bound to the custom `ibm-spectrum-scale-operator` ServiceAccount, which the operator uses to create the IBM Spectrum Scale container native cluster.

To view the permissions of the scale core operator, use the following query on a system that has a deployed CNSA operator.

```
oc describe clusterrole ibm-spectrum-scale-operator
```

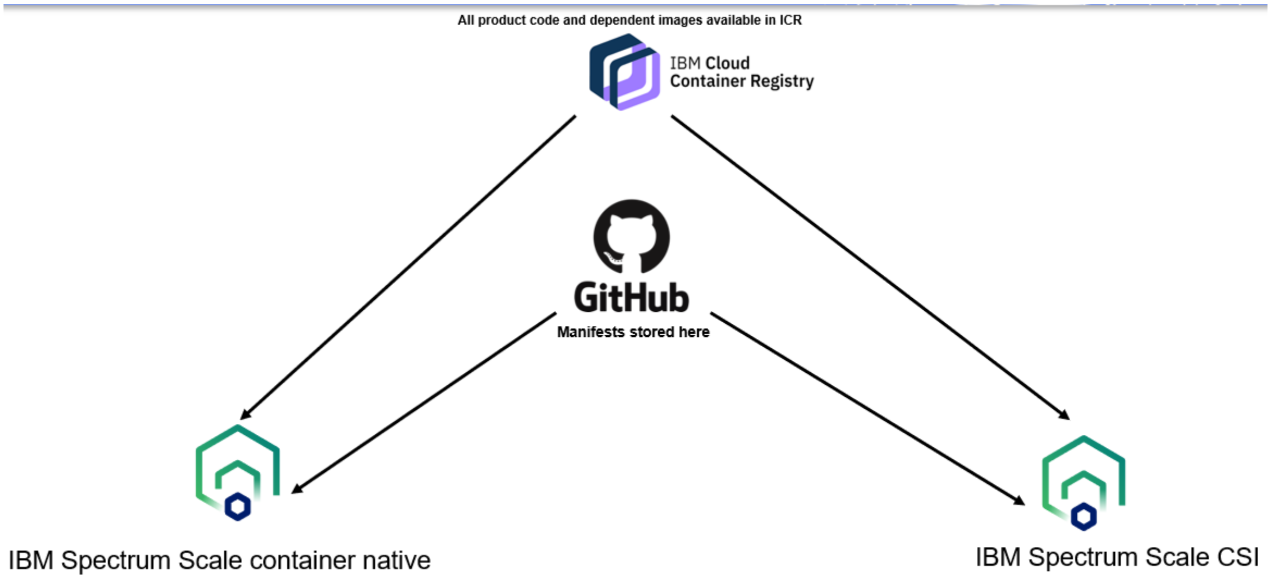
This command will list out every resource the operator has access to and what it can do with them.

Roles

Once a CNSA cluster is operational, users can authenticate to the IBM Spectrum Scale container native GUI via existing OCP roles. For more information, see “OpenShift Users” on page 71.

Container images

The container images are required for the successful deployment of IBM Spectrum Scale container native. All images required for the deployment of IBM Spectrum Scale container native cluster are sourced from the IBM Container Repository.


All product code and dependent images available in ICR
IBM Cloud Container Registry
GitHub
Manifests stored here
IBM Spectrum Scale container native
IBM Spectrum Scale CSI

Note: It is recommended to use the latest fixpack release available.

Figure 4. Dependent images available in ICR

Container image list for IBM Spectrum Scale container native 5.1.7.0

IBM Spectrum Scale images acquired from non-entitled IBM Container Repository

The images listed in the following table are the container images that are obtained through the IBM Container Repository that do not require entitlement. These images can be anonymously pulled.

Table 6. Images acquired from non-entitled IBM Container Repository

Pod	Container	Repository	Image
ibm-spectrum-scale-controller-manager-XXXXXXXXX-XXXXX	manager	icr.io/cpopen/	ibm-spectrum-scale-operator@sha256:eb727060999daea0319c3d67ea7eeb1ca24df6984670272f47f8b6774f451a94
ibm-spectrum-scale-csi-operator	operator	icr.io/cpopen/	ibm-spectrum-scale-csi-operator@sha256:da7ada19c06b20edc9b3c8067a8380f6879899022dda8a5c1cbed7c15b2a381d
must-gather-XXXXX	must-gather	icr.io/cpopen/	ibm-spectrum-scale-must-gather@sha256:f9b4e6570a9ff8194840bbb97cd7f021485dabc806a5115c0e14f06813d580e7

IBM Spectrum Scale images acquired from entitled IBM Container Repository

The images listed in the following table are the container images that are obtained through entitlement to the IBM Container Repository:

Table 7. Images acquired from entitled IBM Container Repository

Pod	Container	Repository	Image
[1] workerX/masterX	mmbuildgpl	cp.icr.io/cp/spectrum/scale	ibm-spectrum-scale-core-init@sha256:f64d3b53d6283068d8bf215a123c65e2e23955bc702bf004aaa0ddd2c43eba64
[1] workerX/masterX	config	cp.icr.io/cp/spectrum/scale	ibm-spectrum-scale-core-init@sha256:f64d3b53d6283068d8bf215a123c65e2e23955bc702bf004aaa0ddd2c43eba64
[1] workerX/masterX	gpfs (if using Data Access Edition)	cp.icr.io/cp/spectrum/scale/data-access	ibm-spectrum-scale-daemon@sha256:8c0db58f8b570cdebebfb611f1ee4da18f62b1ce14e765708e69458929d83049
[1] workerX/masterX	gpfs (if using Data Management Edition)	cp.icr.io/cp/spectrum/scale/data-management	ibm-spectrum-scale-daemon@sha256:50b8a205375e8c0ec15076388db334749e51da223f34627cfb56692efe05bc31
[1] workerX/masterX	logs	cp.icr.io/cp/spectrum/scale	ubi-minimal@sha256:65a240ad8bd3f2fff3e18a22ebadc40da0b145616231fc1e16251f3c6dee087a
ibm-spectrum-scale-gui-X	liberty	cp.icr.io/cp/spectrum/scale	ibm-spectrum-scale-gui@sha256:b2026fd3f989dca9cbaded2157d0dc14c4d89dc1d1f3db0613c07924eb03e852
ibm-spectrum-scale-gui-X	sysmon	cp.icr.io/cp/spectrum/scale	ibm-spectrum-scale-monitor@sha256:70766c93b2bf352ea42b153913e8eacb156a298e750ddb8d8274d3eecc913c5a
ibm-spectrum-scale-gui-X	postgres	cp.icr.io/cp/spectrum/scale	postgres@sha256:c2a30d08a6f9e6c365595fd086c9e0436064c52425f15f72379ecf0807bac518
ibm-spectrum-scale-gui-X	logs	cp.icr.io/cp/spectrum/scale	ubi-minimal@sha256:65a240ad8bd3f2fff3e18a22ebadc40da0b145616231fc1e16251f3c6dee087a
ibm-spectrum-scale-pmcollector-X	pmcollector	cp.icr.io/cp/spectrum/scale	ibm-spectrum-scale-pmcollector@sha256:59e635e0d6c7e84158fda612469a02acd11d4ccc8a4ec85541c47acd449bd8b0
ibm-spectrum-scale-pmcollector-X	sysmon	cp.icr.io/cp/spectrum/scale	ibm-spectrum-scale-monitor@sha256:70766c93b2bf352ea42b153913e8eacb156a298e750ddb8d8274d3eecc913c5a
ibm-spectrum-scale-csi-snapshotter	csi-snapshotter	cp.icr.io/cp/spectrum/scale/csi	csi-snapshotter@sha256:0d8d81948af4897bd07b86046424f022f79634ee0315e9f1d4cdb5c1c8d51c90
ibm-spectrum-scale-csi-attacher	ibm-spectrum-scale-csi-attacher	cp.icr.io/cp/spectrum/scale/csi	csi-attacher@sha256:08721106b949e4f5c7ba34b059e17300d73c8e9495201954edc90eeb3e6d8461

Table 7. Images acquired from entitled IBM Container Repository (continued)

Pod	Container	Repository	Image
ibm-spectrum-scale-csi-provisioner	csi-provisioner	cp.icr.io/cp/spectrum/scale/csi	csi-provisioner@sha256:e468dddcd275163a042ab297b2d8c2aca50d5e148d2d22f3b6ba119e2f31fa79
ibm-spectrum-scale-csi-driver-XXXXX	liveness-probe	cp.icr.io/cp/spectrum/scale/csi	livenessprobe@sha256:2b10b24dafdc3ba94a03fc94d9df9941ca9d6a9207b927f5dfd21d59fbe05ba0
ibm-spectrum-scale-csi-driver-XXXXX	driver-registrar	cp.icr.io/cp/spectrum/scale/csi	csi-node-driver-registrar@sha256:4a4cae5118c4404e35d66059346b7fa0835d7e6319ff45ed73f4bba335cf5183
ibm-spectrum-scale-csi-resizer-X	ibm-spectrum-scale-csi-resizer	cp.icr.io/cp/spectrum/scale/csi	csi-resizer@sha256:3a7bdf5d105783d05d0962fa06ca53032b01694556e633f27366201c2881e01d
ibm-spectrum-scale-csi-driver-XXXXX	ibm-spectrum-scale-csi	cp.icr.io/cp/spectrum/scale/csi	ibm-spectrum-scale-csi-driver@sha256:573b3b2d349359d7871d53060a0fc7df6e03de2e2900d1be46b4146ab1972fb7
ibm-spectrum-scale-grafana-bridge-X	grafanabridge	cp.icr.io/cp/spectrum/scale	ibm-spectrum-scale-grafana-bridge@sha256:bc9eb6ac3a92075cb872c45dc5af2c05422868bdb18e2202ccf928d3cc31d889
coredns-XXXXX	coredns	cp.icr.io/cp/spectrum/scale	ibm-spectrum-scale-coredns@sha256:29f943685acbf4c0a111ae70889465130bac94a4d6d5a6bf5efa0f879c2a79b1

[1] Pod names that contain the mmbuildgpl, config, and gpfs containers may vary. The pod name is based on the shortname of the node it was scheduled to.

Note:

No user action is required to obtain or define this list of images when in a non-airgapped environment. There are instructions to mirror the list of images in an air gap environment. For more information, see Air gap setup for network restricted Red Hat OpenShift Container Platform clusters.

Disconnected Installs

The following section describes installation process in an air gap environment.

Note: It is recommended to use the latest fixpack release available.

Air gap setup for network restricted Red Hat OpenShift Container Platform clusters (optional) 5.1.7.0

Air gap setup is done for Red Hat OpenShift Container Platform clusters that are in a restricted network environment.

Note: You need to do the Air gap setup if the worker nodes are not able to access the repository due to network and firewall restrictions.

Prerequisites

Following are the prerequisites before setting up the air gap environment:

- A production grade Docker V2 compatible registry, such as Quay Enterprise, JFrog Artifactory, or Docker Registry. The Red Hat OpenShift Internal Registry is not supported.
- An online node that can copy images from the source image registry to the production grade internal image registry.
- The online node must have `skopeo` installed.
- Access to the Red Hat OpenShift Container Platform cluster as a user with the `cluster-admin` role.

Note: For Red Hat OpenShift Container Platform clusters that are in a restricted network environment, the obtained files must be transferred to a bastion/infrastructure node that can communicate with the target cluster before applying the `yaml` files. This is likely the same node in your Red Hat OpenShift Container Platform cluster where the `oc` command is executed.

Configuring the registry mirror

Create a new `ImageContentSourcePolicy` on your Red Hat OpenShift cluster to enable the redirection of requests to pull images from a repository on a mirrored image registry.

Complete the following steps from the `inf` node of your Red Hat OpenShift cluster:

1. Paste the following in a file, `registrymirror.yaml` and replace `example.io/subdir` with your internal image registry repository:

   ```
   apiVersion: operator.openshift.io/v1alpha1
   kind: ImageContentSourcePolicy
   metadata:
     name: icr-mirror
   spec:
     repositoryDigestMirrors:
     - mirrors:
       - example.io/subdir
       source: cp.icr.io/cp/spectrum/scale
     - mirrors:
       - example.io/subdir
       source: icr.io/cpopen
   ```

   Note: Do not prefix mirrors with `http://` or `https://` and ensure that they do not have trailing / characters as this causes an issue while resolving them correctly.

2. Create `ImageContentSourcePolicy` named `icr-mirror` by entering the following command:

   ```
   oc apply -f registrymirror.yaml
   ```

   This update is rolled out to all nodes, which can take some time depending on the size of your cluster.

3. If you want to verify the `ImageContentSourcePolicy` is applied, enter the `oc  debug` command to query the mirrors on the host nodes.

   ```
   $  oc debug node/worker0.subdomain
   Starting pod/worker0examplecom-debug ...
   To use host binaries, run `chroot /host`
   Pod IP: 12.34.56.789
   If you don't see a command prompt, try pressing enter.

   # chroot /host
   # cat /etc/containers/registries.conf
   unqualified-search-registries = ["registry.access.redhat.com", "docker.io"]

   [[registry]]
     prefix = ""
     location = "cp.icr.io/cp/spectrum/scale"
     mirror-by-digest-only = true
   ```

```
  [[registry.mirror]]
    location = "example.io/subdir"

[[registry]]
  prefix = ""
  location = "icr.io/cpopen"
  mirror-by-digest-only = true

  [[registry.mirror]]
    location = "example.io/subdir"
```

Note: For more information, see Configuring image registry repository mirroring in Red Hat OpenShift documentation.

Copying images from source image registry to target internal image registry

The OpenShift cluster is configured to redirect external image registry requests to an internal registry through the `ImageContentSourcePolicy`. Now, the internal registry must be populated with the images from the source image registry.

Complete the following steps from the `online` node described in the prerequisites:

1. Log in to the IBM Entitled Container Registry with the credentials by entering the `skopeo` command.

   ```
   skopeo login cp.icr.io
   ```

2. Log in to your internal production grade image registry with the credentials by entering the `skopeo` command.

   ```
   skopeo login example.io
   ```

3. Use `skopeo  copy` to copy the following images from the IBM Entitled Container Registry to your internal production grade image registry.

   ```
   icr.io/cpopen/ibm-spectrum-scale-
   operator@sha256:eb727060999daea0319c3d67ea7eeb1ca24df6984670272f47f8b6774f451a94
   cp.icr.io/cp/spectrum/scale/ibm-spectrum-scale-core-
   init@sha256:f64d3b53d6283068d8bf215a123c65e2e23955bc702bf004aaa0ddd2c43eba64
   cp.icr.io/cp/spectrum/scale/ibm-spectrum-scale-
   pmcollector@sha256:59e635e0d6c7e84158fda612469a02acd11d4ccc8a4ec85541c47acd449bd8b0
   cp.icr.io/cp/spectrum/scale/ibm-spectrum-scale-
   monitor@sha256:70766c93b2bf352ea42b153913e8eacb156a298e750ddb8d8274d3eecc913c5a
   cp.icr.io/cp/spectrum/scale/ibm-spectrum-scale-
   gui@sha256:b2026fd3f989dca9cbaded2157d0dc14c4d89dc1d1f3db0613c07924eb03e852
   cp.icr.io/cp/spectrum/scale/ibm-spectrum-scale-grafana-
   bridge@sha256:bc9eb6ac3a92075cb872c45dc5af2c05422868bdb18e2202ccf928d3cc31d889
   icr.io/cpopen/ibm-spectrum-scale-must-
   gather@sha256:f9b4e6570a9ff8194840bbb97cd7f021485dabc806a5115c0e14f06813d580e7
   cp.icr.io/cp/spectrum/scale/ibm-spectrum-scale-
   coredns@sha256:29f943685acbf4c0a111ae70889465130bac94a4d6d5a6bf5efa0f879c2a79b1
   cp.icr.io/cp/spectrum/scale/ubi-
   minimal@sha256:65a240ad8bd3f2fff3e18a22ebadc40da0b145616231fc1e16251f3c6dee087a
   cp.icr.io/cp/spectrum/scale/
   postgres@sha256:c2a30d08a6f9e6c365595fd086c9e0436064c52425f15f72379ecf0807bac518
   icr.io/cpopen/ibm-spectrum-scale-csi-
   operator@sha256:da7ada19c06b20edc9b3c8067a8380f6879899022dda8a5c1cbed7c15b2a381d
   cp.icr.io/cp/spectrum/scale/csi/ibm-spectrum-scale-csi-
   driver@sha256:573b3b2d349359d7871d53060a0fc7df6e03de2e2900d1be46b4146ab1972fb7
   cp.icr.io/cp/spectrum/scale/csi/csi-
   snapshotter@sha256:0d8d81948af4897bd07b86046424f022f79634ee0315e9f1d4cdb5c1c8d51c90
   cp.icr.io/cp/spectrum/scale/csi/csi-
   provisioner@sha256:e468dddcd275163a042ab297b2d8c2aca50d5e148d2d22f3b6ba119e2f31fa79
   cp.icr.io/cp/spectrum/scale/csi/csi-node-driver-
   registrar@sha256:4a4cae5118c4404e35d66059346b7fa0835d7e6319ff45ed73f4bba335cf5183
   cp.icr.io/cp/spectrum/scale/csi/csi-
   attacher@sha256:08721106b949e4f5c7ba34b059e17300d73c8e9495201954edc90eeb3e6d8461
   cp.icr.io/cp/spectrum/scale/csi/
   livenessprobe@sha256:2b10b24dafdc3ba94a03fc94d9df9941ca9d6a9207b927f5dfd21d59fbe05ba0
   cp.icr.io/cp/spectrum/scale/csi/csi-
   resizer@sha256:3a7bdf5d105783d05d0962fa06ca53032b01694556e633f27366201c2881e01d
   ```

To deploy a cluster using the Data Access edition of IBM Spectrum Scale container native, copy the following image:

```
cp.icr.io/cp/spectrum/scale/data-access/ibm-spectrum-scale-
daemon@sha256:8c0db58f8b570cdebebfb611f1ee4da18f62b1ce14e765708e69458929d83049
```

To deploy a cluster using the Data Management edition of IBM Spectrum Scale container native, copy the following image:

```
cp.icr.io/cp/spectrum/scale/data-management/ibm-spectrum-scale-
daemon@sha256:50b8a205375e8c0ec15076388db334749e51da223f34627cfb56692efe05bc31
```

Note: The destination is up to the user and depends on how the registry mirror was configured in the first section. Using the same `example.io/subdir` repository, a sample `skopeo  copy` command is shown:

```
skopeo copy --all docker://cp.icr.io/cp/spectrum/scale/ibm-
spectrum-scale-gui@sha256:b2026fd3f989dca9cbaded2157d0dc14c4d89dc1d1f3db0613c07924eb03e852
docker://example.io/subdir/ibm-spectrum-scale-
gui@sha256:b2026fd3f989dca9cbaded2157d0dc14c4d89dc1d1f3db0613c07924eb03e852
```

Note: The ibm-spectrum-scale-daemon image is edition specific. When copying it, you must put it in a folder that indicates its edition. The folder it resides in must be `data-access` or `data-management` depending on the image you are entitled to.

The sample command for copying the Data Access Edition `ibm-spectrum-scale-daemon` image is shown:

```
skopeo copy --all docker://cp.icr.io/cp/spectrum/scale/data-access/ibm-spectrum-
scale-daemon@sha256:8c0db58f8b570cdebebfb611f1ee4da18f62b1ce14e765708e69458929d83049
docker://example.io/subdir/data-access/ibm-spectrum-scale-
daemon@sha256:8c0db58f8b570cdebebfb611f1ee4da18f62b1ce14e765708e69458929d83049
```

The sample command for copying the Data Management Edition `ibm-spectrum-scale-daemon` image is:

```
skopeo copy --all docker://cp.icr.io/cp/spectrum/scale/data-management/ibm-spectrum-
scale-daemon@sha256:50b8a205375e8c0ec15076388db334749e51da223f34627cfb56692efe05bc31
docker://example.io/subdir/data-management/ibm-spectrum-scale-
daemon@sha256:50b8a205375e8c0ec15076388db334749e51da223f34627cfb56692efe05bc31
```

A generic `skopeo  copy` command is shown:

```
skopeo copy --all docker://<source image registry>/<image> docker://<internal image
registry>/<image>
```

4. Log out of the IBM Entitled Container Registry by entering the `skopeo` command.

   ```
   skopeo logout cp.icr.io
   ```

5. Log out of your internal production grade image registry by entering the `skopeo` command.

   ```
   skopeo logout example.io
   ```

Testing the pull of images from the mirrored registry

Complete the following steps from the `inf` node of your OpenShift cluster:

1. Pick a worker node from `oc  get  nodes` and start a node to debug it.

   ```
   oc debug node/<worker node>
   ```

   A command prompt must be presented.

2. Switch to host binaries by entering the `chroot /host` command.

   ```
   # oc debug node/worker0.example.com
   Starting pod/worker0examplecom-debug ...
   To use host binaries, run `chroot /host`
   Pod IP: 12.34.56.789
   If you don't see a command prompt, try pressing enter.
   # chroot /host
   ```

3. Enter the `podman login` command to authenticate your mirrored image registry.

   ```
   # podman login example.io
   Username: sampleemail@email.com
   Password:
   Login Succeeded!
   ```

4. Attempt to pull one of the images from the source image registry through podman. The OpenShift cluster must be able to redirect the request from the external image registry to the internal image registry and successfully pull the image.

   ```
   # podman pull cp.icr.io/cp/spectrum/scale/ibm-spectrum-scale-
   gui@sha256:b2026fd3f989dca9cbaded2157d0dc14c4d89dc1d1f3db0613c07924eb03e852
   ```

5. Verify that the image is pulled.

   ```
   # podman images | grepcp.icr.io/cp/spectrum/scale/ibm-spectrum-scale-gui
   cp.icr.io/cp/spectrum/scale/ibm-spectrum-scale-gui    <none>     9c215ae62f37   22 hours
   ago    851 MB
   ```

Red Hat OpenShift Container Registry pull secret

For images to be properly pulled at the pod level, the OpenShift global pull secrets must be modified to contain credentials to access your internal container registry.

Complete the following steps:

1. Create a base64 encoded string of the credentials used to access your internal container registry.

   Note: The following example uses `example.io/subdir` as the internal container registry.

   - Use the credentials to access your `example.io/subdir` internal container registry.

     ```
     echo -n "<username>:<password>" | base64 -w0
     ```

2. Create an `authority.json` to include the base64 encoded string of your credentials. Use your username and password to access internal container registry `example.io/subdir`.

   ```
   {
     "auth": "<base64 encoded string from previous step>",
     "username":"<example.io username>",
     "password":"<example.io generated entitlement key>"
   }
   ```

3. Enter the following command to include the `authority.json` as a new authority in your `.dockerconfigjson` and store it as `temp_config.json`:

   Note: For the example internal container registry of `example.io/subdir`, use `example.io` as the input key for the contents of `authority.json`.

   ```
   oc get secret/pull-secret -n openshift-config -ojson | \
   jq -r '.data[".dockerconfigjson"]' | \
   base64 -d - | \
   jq '.[]."example.io" += input' - authority.json > temp_config.json
   ```

   Note: This command is supported with `jq` 1.5.

- Enter the following command to verify that your authority credentials were created in the resulting file:

```
# cat temp_config.json
{
   "auths": {
      "quay.io": {
        "auth": "",
        "email": ""
      },
      "registry.connect.redhat.com": {
         "auth": "",
         "email": ""
      },
      "registry.redhat.io": {
         "auth": "",
         "email": ""
      },
      "example.io": {
         "auth": "<base64 encoded string created in previous step>",
         "username": "<example.io username>",
         "password": "<example.io password>"
      }
   }
}
```

4. Verify that your pull-secret is updated with your new authority, enter the following command and confirm your authority is present:

```
oc get secret/pull-secret -n openshift-config -ojson | \
jq -r '.data[".dockerconfigjson"]' | \
base64 -d -
```

5. Apply the pull secret configuration to the OpenShift cluster.

```
oc set data secret/pull-secret -n openshift-config --from-
file=.dockerconfigjson=temp_config.json
```

   This update is rolled out to all nodes, which can take some time depending on the size of your cluster.

6. Remove the temporary files that were created.

```
rm authority.json temp_config.json
```

Chapter 3. Storage Cluster

IBM Spectrum Scale container native supports remote mounting filesystems running in the following scenarios.

On-premise

Some additional tasks need to be performed on the IBM Spectrum Scale storage cluster running on-premise.

IBM Spectrum Scale storage cluster

The operators of CNSA and CSI interact with the storage cluster through REST API (which is part of the GUI stack). To enable this, user IDs need to be created on the storage cluster GUI. There are tailored roles that grant those user IDs only the operations needed to provide their functionality. In addition, some settings on the cluster and the filesystem are required for interoperability with CSI.

Note: If the storage cluster is running a GUI high availability configuration, for example, having 2 or more GUI nodes installed, ensure the storage cluster is running IBM Spectrum Scale 5.1.6.1 or higher prior to use with IBM Spectrum Scale container native and IBM Spectrum Scale CSI.

Creating Container Operator and CSI Users

Complete the following steps in the shell of the GUI node of the storage cluster:

1. To create container native operator GUI user, enter the following command:

   ```
   /usr/lpp/mmfs/gui/cli/mkuser cnsa_storage_gui_user -p cnsa_storage_gui_password -g
   ContainerOperator
   ```

   By default, user passwords expire after 90 days. If the security policy of your organization permits it, use the `-e 1` option at the end of the above command to create a user with a password that does not expire.

2. To create the CSI GUI user, enter the following commands:

   ```
   /usr/lpp/mmfs/gui/cli/mkuser csi_storage_gui_user -p  csi_storage_gui_password -g CsiAdmin
   ```

   By default, user passwords expire after 90 days. If the security policy of your organization permits it, use the `-e 1` option at the end of the above command to create a user with a password that does not expire.

   Note: The `-e 1` parameter is only available for IBM Spectrum Scale storage cluster 5.1.1.0 or later.

Storage cluster configuration for Container Storage Interface (CSI)

Complete the following steps on the storage cluster to ensure the IBM Spectrum Scale CSI driver can operate successfully:

1. Ensure that the perfileset quota on the file systems used by IBM Spectrum Scale Container Storage Interface driver is set to No.

   Note: The IBM Spectrum Scale Container Storage Interface driver will create a large number of filesets (one per PV). Tracking user and group quotas on a per-fileset basis will significantly increase the overhead of quota management. As a result, the file system performance may suffer.

   ```
   mmlsfs fs1 --perfileset-quota
   ```

2. Enter the following command to enable the Quota in the file systems:

© Copyright IBM Corp. 2020, 2023

Note: The IBM Spectrum Scale Container Storage Interface driver translates capacity of persistent volumes to fileset quotas. For this to work, quotas are required to be enabled in the file system.

```
mmchfs fs1 -Q yes
```

```
mmlsfs fs1 -Q
```

3. Enable the quota for root user by entering the following command:

   Note: On Kubernetes, the containers may run as root, so ensure that quotas are enforced for the root user as well.

   ```
   mmchconfig enforceFilesetQuotaOnRoot=yes -i
   ```

4. Ensure that the `controlSetxattrImmutableSELinux` parameter is set to `yes` by entering the following command:

   Note: Kubernetes does not honor immutability of files/directories when setting SELinux labels. This creates issues, for example, with the immutable `.snapshot` directory.

   ```
   mmchconfig controlSetxattrImmutableSELinux=yes -i
   ```

5. Enable `filesetdf` of the file system by entering the following command:

   Note: IBM Spectrum Scale Container Storage Interface driver will only be able to report free space on persistant volumes if `filesetdf` is set correctly.

   ```
   mmchfs fs1 --filesetdf
   ```

6. Enable `auto-inode-limit` of the file system by entering the following command:

   Note: IBM Spectrum Scale Container Storage Interface driver has no information about the number of inodes a persistent volume will consume. Therefore, corresponding independent filesets are created with default values for maxIndoes. The above setting enables automatic expansion of the inode space, so persistent volumes do not run out of inodes.

   ```
   mmchfs fs1 --auto-inode-limit
   ```

   Note: The `--auto-inode-limit` option is available only with filesystem format level 28.00 or later. Enable `auto-inode-limit` as soon as the filesystem format level is updated to 28.00 or later. On older filesystem levels the administrator of the storage cluster needs to manually increase the inode limit when warnings for low inodes are raised by the health monitoring.

   For more information about `auto-inode-limit` parameter, see mmchfs command in IBM Spectrum Scale documentation.

Configure the cluster profile with tscCmdAllowRemoteConnections

Starting from IBM Spectrum Scale and IBM Spectrum Scale container native 5.1.3, the `tscCmdAllowRemoteConnections` configuration is recommended to be set to no. If the storage cluster and all client clusters (including IBM Spectrum Scale container native) are at versions >= 5.1.3, it is recommended to set this value to no. However, if any version is < 5.1.3, `tscCmdAllowRemoteConnections` must be set to `yes` on the storage cluster and client clusters to successfully communicate between the clusters.

Use the following table as a reference:

Storage cluster version	IBM Spectrum Scale container native version	`tscCmdAllowRemoteConnections`
< 5.1.3	< 5.1.3.0	yes
>= 5.1.3	< 5.1.3.0	yes
>= 5.1.3	>= 5.1.3	no

- To change this value on the storage cluster, issue the following command:

```
# mmchconfig tscCmdAllowRemoteConnections='yes|no'
```

- To change this value on the IBM Spectrum Scale container native cluster, set `tscCmdAllowRemoteConnections: yes|no` in the `clusterProfile` section of the cluster spec:

```
kind: Clustermetadata:
name: ibm-spectrum-scalespec:
  ...daemon:
    ......clusterProfile:
      tscCmdAllowRemoteConnections: "yes"
```

For more information about how to configure the `clusterProfile` section of the cluster spec, see Cluster profile.

For more information about all IBM Spectrum Scale services, see Securing the IBM Spectrum Scale system using firewall in IBM Spectrum Scale documentation.

AWS storage cluster

IBM Spectrum Scale container native with Red Hat OpenShift Service on AWS (ROSA) provides a highly scalable and performant solution for containerized applications. You can use the IBM Spectrum Scale Storage cluster on public clouds (such as AWS) to store its persistent data. IBM Spectrum Scale storage cluster on public clouds can be deployed and managed using the tool `cloudkit`. For more information, see cloudkit in IBM Spectrum Scale documentation.

`cloudkit` is included in the IBM Spectrum Scale Self-Extracting (SE) package and enables users to deploy IBM Spectrum Scale clusters across multiple cloud providers (currently limited to AWS), providing deployment topology flexibility. In addition, `cloudkit` provides a user-friendly interface for cluster deployment, reducing the complexity and time required for setting up the storage cluster. It also offers the ability to configure various cluster parameters, such as disk type, instance type, and network configurations, through its interactive interface.

Furthermore, `cloudkit` offers integration with other IBM Spectrum Scale features, such as data replication, data tiering, and data encryption, providing a complete storage solution for various use cases.

Before you begin

- Before deploying a storage cluster, it is important to understand your requirements in terms of performance, scalability, data availability, and data protection. Decide how you want to structure your storage, including the number of storage nodes, the filesystem size, and the number of replicas you want to maintain.
- ROSA and `cloudkit` both utilize Multi-Availability Zone (AZ) to ensure resiliency, but in Multi-AZ mode, the inter-AZ network limit will constrain the throughput.
- Review the AWS quota limits to ensure that your existing Virtual Private Cloud (VPC) has enough free IP addresses to accommodate both ROSA and `cloudkit`. If you are planning to create a new VPC, you should plan the network to accommodate the IP requirements for both services.
- Users have the flexibility to start with either ROSA in new VPC mode and `cloudkit` in existing VPC mode or `cloudkit` in new VPC mode and ROSA in existing VPC mode, depending on the requirements.
- For optimal performance, it is recommended to choose storage cluster deployment via `cloudkit` within the same subnet that is in use by the ROSA. This ensures that the data traffic between ROSA and the IBM Spectrum Scale storage cluster is within the same AWS Availability Zone, providing lower latency and higher throughput.

About this task

This task assumes that the installation of IBM Spectrum Scale container native on AWS ROSA is completed and the `cloudkit` is being used to deploy the storage cluster using the same VPC created by the ROSA installer.

Extract the IBM Spectrum Scale SE package and locate the cloudkit binary

For more information related to this step, see cloudkit in IBM Spectrum Scale documentation.

Deploy storage cluster

```
# ./cloudkit create cluster
I: Logging at /root/scale-cloudkit/logs/cloudkit-15-2-2023_23-12-14.log
? Cloud platform name:  AWS
? Cluster name:  scale-strg-cls
? VPC Mode:  Existing
? IBM Spectrum Scale deployment model:  Storage-only
? Connectivity method to cloud:  JumpHost
? Tuning profile:  Throughput-Performance-Persistent-Storage
? Storage cluster management GUI username:  administrator
? Storage cluster management GUI password:  ********
? Filesystem mount point:  /gpfs/fs1
? Filesystem block size:  4M
? Filesystem capacity (Gi):  10000Gi
? Operator Email (Optional):
? Region: us-east-2
? Select availability zones (1):  us-east-2a
? VPC ID:  vpc-0e4d0912c5c73bacb | 10.0.0.0/16 | rosa-scale-r9mzd-vpc
? Public Subnet IDs:  subnet-044ff2a60a56b4622 | us-east-2a | rosa-scale-r9mzd-public-us-east-2a
? Private Subnet IDs:  subnet-0c7c2a420957d8696 | us-east-2a | rosa-scale-r9mzd-private-us-
east-2a
? Do you wish to continue by creation of bastion/jumphost:  Yes
? Bastion OS:  amzn-ami-hvm-2018.03.0.20230207.0-x86_64-gp2 | ami-0e765ecc7a91f4a2f | ec2-user
? Bastion instance type:  t3.micro    | vCPU(1)  | RAM (1.0 GiB) | CPU Credits/hr (6)
? Select key pair to be used for launching Bastion/Jump host instance(s):  aws_keypair
? Bastion/JumpHost SSH private key file path (will be used only for configuration):  /root/
bastion_pvt_key
? Bastion CIDR block allow list:  129.41.87.4/32
? Do you prefer to use AWS offered stock image for storage nodes:  No
? Select existing AMI ID:  spectrum-scale-5.1.7.0-34 | ami-054062979cfd084f3 | 5.1.7.0 | IBM
Spectrum Scale AMI created with 5.1.7.0 version
? Storage instance type:  m4.large    | vCPU(2)  | RAM (8.0 GiB) | Dedicated EBS Bandwidth (450
Mbps)
? Select key pair to be used for launching Storage host instance(s):  aws_keypair
? Storage node count:  4
? EBS Disk type:  gp2
? Do you wish to view cost before provisioning the resources (this requires infracost
api_key):  No
I: To create this cluster again in the future, you can run:
    cloudkit create cluster --region us-east-2 --cloud-platform AWS --resource-name scale-strg-
cls --gpfs-rpms /usr/lpp/mmfs/5.1.7.0 --avail-zones "us-east-2a" --vpc-type Existing --
deployment-mode Storage-only --network-mode JumpHost --profile-name Throughput-Performance-
Persistent-Storage --fs-mount-point /gpfs/fs1 --fs-block-size 4M --fs-capacity 10000Gi --strg-
gui-username administrator --skip-email true --vol-type gp2 --vpc-id vpc-0e4d0912c5c73bacb --
public-subnet-id subnet-044ff2a60a56b4622 --jumphost-login-user ec2-user --jumphost-image-id
ami-0e765ecc7a91f4a2f --jumphost-instance-type t3.micro --jumphost-key-pair aws_keypair --
jumphost-key-path /root/bastion_pvt_key --strg-image-id ami-054062979cfd084f3 --strg-ins-type
m4.large --strg-ins-keypair aws_keypair --strg-node-count 4 --is-stock-image false

I: Writing JumpHost inputs to '/root/scale-cloudkit/workarea/cluster/aws/scale-strg-cls/
bastion_template/inputs.auto.tfvars.json' completed.
I: Initializing the working directory '/usr/lpp/mmfs/5.1.7.0/cloudkit/dependencies/ibm-spectrum-
scale-cloud-install/aws_scale_templates/sub_modules/bastion_template' to be used by terraform.
I: Initiating resource provisioning actions proposed in the terraform plan.
 100% |
████████████████████████████████████████████████████████████████████████████████████████████████
███████████████| (10/10, 4 it/min)
I: JumpHost resource provisioning actions proposed in the plan completed.

I: Writing storage cluster inputs to '/root/scale-cloudkit/workarea/cluster/aws/scale-strg-cls/
instance_template/inputs.auto.tfvars.json' completed.
I: Initializing the working directory '/usr/lpp/mmfs/5.1.7.0/cloudkit/dependencies/ibm-spectrum-
scale-cloud-install/aws_scale_templates/sub_modules/instance_template' to be used by terraform.
I: Initiating resource provisioning actions proposed in the terraform plan.
 100% |
████████████████████████████████████████████████████████████████████████████████████████████████
███████████████| (10/10, 4 it/min)
I: Storage cluster resource provisioning actions proposed in the plan completed.
```

```
I: Waiting for instances to obtain running state.
I: Initiating cluster configuration based on the provisioned resource inventory.
 100% |
████████████████████████████████████████████████████████████████████████████████████████████████
███████████████| (10/10, 1 it/min)
I: Spectrum Scale cluster configuration completed.
I: Spectrum Scale cluster 'scale-strg-cls' has been created.
```

Note: `cloudkit` provides various deployment profiles, including `Throughput-Performance-Persistent-Storage`, `Throughput-Performance-Scratch-Storage`, and `Balanced`.

If you are deploying to a Single AZ, it is advisable to use the `Throughput-Performance-Persistent-Storage` profile, while in the case of multi-AZ, the `Balanced` profile is recommended.

⚠ **CAUTION:** It is not recommended to use the `Throughput-Performance-Scratch-Storage` profile for IBM Spectrum Scale container native workloads, as this profile uses instance storage and results in data loss when storage nodes are shut down.

Details of the cluster created through `cloudkit` can be viewable using the `cloudkit describe cluster` command:

```
# ./cloudkit describe cluster
I: Logging at /root/scale-cloudkit/logs/cloudkit-16-2-2023_0-5-7.log
? Cloud platform name:  AWS
? Select cluster name:  scale-strg-cls
+------------------------------------------+-------------------------------------------+
|                  FIELDS                  |                  DETAILS                  |
+------------------------------------------+-------------------------------------------+
| VPC_ID                                   | vpc-0e4d0912c5c73bacb                     |
| STORAGE_CLUSTER_INSTANCE_IDS             | [i-0285f16f3c1594b54                      |
|                                          | i-0294f558e273063a5                       |
|                                          | i-05bdc01d916b419a5                       |
|                                          | i-06cd0dfdbddc8720d]                      |
| STORAGE_CLUSTER_INSTANCE_PRIVATE_IPS     | [10.0.143.20 10.0.156.67                  |
|                                          | 10.0.163.242 10.0.228.45]                 |
| STORAGE_CLUSTER_WITH_DATA_VOLUME_MAPPING | map[10.0.143.20:[/dev/xvdf]               |
|                                          | 10.0.156.67:[/dev/xvdf]                   |
|                                          | 10.0.163.242:[/dev/xvdf]                  |
|                                          | 10.0.228.45:[/dev/xvdf]]                  |
| BASTION_SECURITY_GROUP_ID                | sg-011f51e6b54f5e49a                      |
| STORAGE_CLUSTER_SECURITY_GROUP_ID        | <sg-storage-hash>                         |
| PROFILE_NAME                             | Throughput-Performance-Persistent-Storage |
| BASTION_INSTANCE_ID                      | [i-052aa97a047f0ec9d]                     |
| BASTION_INSTANCE_PRIVATE_IP              | [10.0.67.224]                             |
| BASTION_INSTANCE_PUBLIC_IP               | [xxx.xxx.xxx.]                            |
+------------------------------------------+-------------------------------------------+
```

Perform grant filesystem operation

`cloudkit` performs setting up security rules for data access and exchange between the ROSA worker security group and storage cluster. It performs the prerequisite configuration such as creating users in `ContainerOperator`, `CsiAdmin` group on the storage cluster. Using `cloudkit`, remote-mounted storage filesystems can greatly simplify data management across multiple IBM Spectrum Scale container native clusters.

```
# ./cloudkit grant filesystem
I: Logging at /root/scale-cloudkit/logs/cloudkit-16-2-2023_0-10-57.log
? Cloud platform name:  AWS
? Select storage cluster name (cluster that owns and serves the file system):  scale-strg-cls
? Compute cluster type:  Spectrum-Scale-CNSA-Cluster
? Select OCP Worker security group:  rosa-scale-r9mzd-worker-sg |
terraform-20230216064317352500000003 | <sg-ocp-group-hash>
? CNSA operator cluster management GUI username:  cnsa_administrator
? CNSA operator cluster management GUI password:  ********
? CSI admin cluster management GUI username:  csi_administrator
? CSI admin cluster management GUI password:  ********
? Storage cluster management GUI username:  administrator
? Storage cluster management GUI password:  ********
? Connectivity method to cloud:  JumpHost
? Bastion/JumpHost instance login username:  ec2-user
? Bastion/JumpHost instance public ip address:  xxx.xxx.xxx.195
? Bastion/JumpHost SSH private key file path (will be used only for configuration):  /root/
bastion_pvt_key
```

```
I: Obtaining spectrum scale storage cluster definition.
I: Updating storage security group '<sg-storage-hash>' to allow traffic from OCP security group
'<sg-ocp-group-hash>'.
I: Initiating remote mount configuration.
 100% |
██████████████████████████████████████████████████████████████████████████████████████████████████
███████████████| (10/10, 1 it/min)
I: Spectrum Scale cluster configuration completed.
I: Access to Spectrum Scale storage cluster 'scale-strg-cls' has been granted for compute
cluster 'rosa-scale'.
```

Details of the filesystem access grant through `cloudkit` can be viewable using the `cloudkit describe grant` command:

```
# ./cloudkit describe grant
I: Logging at /root/scale-cloudkit/logs/cloudkit-16-2-2023_0-13-52.log
? Cloud platform name:  AWS
? Select remotemount name:  scale-strg-cls-rosa-scale
+---------------------+--------------------------------+
|       FIELDS        |            DETAILS             |
+---------------------+--------------------------------+
| OCP_CLS_SG_ID       | <sg-ocp-group-hash>            |
| SCALE_GUI_IPADDRESS | 10.0.143.20                    |
| SCALE_CONTACT_NODES | [10.0.156.67 10.0.163.242      |
|                     | 10.0.228.45]                   |
| PLATFORM            | AWS                            |
| STRG_CLS_NAME       | scale-strg-cls                 |
| STRG_CLS_SG_ID      | <sg-storage-hash>              |
| CMP_CLS_NAME        | rosa-scale                     |
| GRANT_NAME          | scale-strg-cls-rosa-scale      |
| REGION              | us-east-2                      |
| GRANT_MODE          | Spectrum-Scale-CNSA-Cluster    |
+---------------------+--------------------------------+
```

Perform revoke filesystem operation

Once the IBM Spectrum Scale container native has been uninstalled, it is recommended to revoke the filesystem access using the `cloudkit revoke filesystem` command:

```
# ./cloudkit revoke filesystem
I: Logging at /root/scale-cloudkit/logs/cloudkit-16-2-2023_0-16-37.log
? Cloud platform name:  AWS
=========================================================================
|                              ! Note !                                 |
========================================================================|
|  Revoke cluster involves unmount of filesystem. Make sure the I/O is  |
|  stopped and data is flushed before proceeding further.               |
=========================================================================
? Select remotemount name:  scale-strg-cls-rosa-scale
? IBM Spectrum Scale Container Native cluster name:  ibm-spectrum-scale.stg.rosa-
scale.example.com
? Storage cluster management GUI username:  administrator
? Storage cluster management GUI password:  ********
? Connectivity method to cloud:  JumpHost
? Bastion/JumpHost instance login username:  ec2-user
? Bastion/JumpHost instance public ip address:  xxx.xxx.xxx.195
? Bastion/JumpHost SSH private key file path (will be used only for configuration):  /root/
bastion_pvt_key
I: Obtaining spectrum scale storage cluster definition.
I: Initiating remote mount revoke configuration.
 100% |
██████████████████████████████████████████████████████████████████████████████████████████████████
███████████████| (10/10, 1 it/min)
I: Spectrum Scale cluster configuration completed.
I: Updating storage security group '<sg-storage-hash>' to revoke traffic from OCP security
group '<sg-ocp-group-hash>'.
I: Access to Spectrum Scale cluster 'scale-strg-cls' has been revoked.
```

Delete storage cluster

CAUTION: It is crucial to take appropriate data backups before proceeding with the deletion of the storage cluster, as this operation will irrevocably terminate all scale instances and wipe out the disks used for filesystem creation. Failing to do so may result in the permanent loss of critical data.

```
# ./cloudkit delete cluster
I: Logging at /root/scale-cloudkit/logs/cloudkit-16-2-2023_0-34-7.log
? Cloud platform name:  AWS
? Select cluster name:  scale-strg-cls
W: Deleting a cluster may lead to failure if there are other resources that reference the
resources created by cloudKit, or if there are additional resources sharing the network
resources created by cloudKit that are not recognized by cloudKit.
? W: Data Deletion Alert! Are you sure you want to delete 'scale-strg-cls'. To proceed further,
type `permanently delete`:  permanently delete
I: Destroy all remote objects managed by terraform (instance_template) configuration.
I: Destroy all remote objects managed by terraform (bastion_template) configuration.
I: Spectrum Scale cluster 'scale-strg-cls' has been deleted.
```

Chapter 4. Red Hat OpenShift Container Platform

Refer to the following sections to help prepare Red Hat OpenShift environments.

IBM Cloud Container Registry (ICR) Pull Secrets

IBM Spectrum Scale container native images are hosted in IBM Cloud Container Registry. Obtain an entitlement key from IBM container software library.

Entitlement keys determine whether the IBM Spectrum Scale operator can automatically pull the required IBM Spectrum Scale container native images. Image pull failures may occur due to an invalid entitlement key or a key belonging to an account that does not have entitlement to either IBM Spectrum Scale Data Access Edition or IBM Spectrum Scale Data Management Edition.

Adding IBM Cloud Container Registry credentials

For images to be properly pulled at the pod level, the OpenShift global pull secrets must be modified to contain credentials to access the IBM Cloud Container Registry.

Note: The following steps are for users whose OpenShift cluster is accessing the IBM Cloud Container Registry. For more information, see Disconnected Installs.

1. Create a base64 encoded string of the credentials used to access the image registry.

   - For using IBM Cloud Container Registry, the credentials must use the `cp` user along with the entitlement key.

     ```
     echo -n "cp:REPLACE_WITH_GENERATED_ENTITLEMENT_KEY" | base64 -w0
     ```

2. Create an `authority.json` to include the base64 encoded string of your credentials, the fixed username `cp` (used to access `cp.icr.io` repository), and entitlement key for the IBM Cloud Container Registry.

   ```
   {
     "auth": "REPLACE_WITH_BASE64_ENCODED_KEY_FROM_PREVIOUS_STEP",
     "username":"cp",
     "password":"REPLACE_WITH_GENERATED_ENTITLEMENT_KEY"
   }
   ```

3. Enter the following command to include the `authority.json` as a new authority in your `.dockerconfigjson` and store it as `temp_config.json`:

   Note: Using the IBM Cloud Container Registry as the authority, use `cp.icr.io` as the input key for the contents of `authority.json`.

   ```
   oc get secret/pull-secret -n openshift-config -ojson | \
   jq -r '.data[".dockerconfigjson"]' | \
   base64 -d - | \
   jq '.[]."cp.icr.io" += input' - authority.json > temp_config.json
   ```

   Note: This command is supported with `jq` 1.5.

   - To verify that your authority credentials were created in the resulting file:

     ```
     # cat temp_config.json
     {
        "auths": {
           "quay.io": {
             "auth": "",
             "email": ""
           },
           "registry.connect.redhat.com": {
              "auth": "",
     ```

© Copyright IBM Corp. 2020, 2023

```
         "email": ""
      },
      "registry.redhat.io": {
         "auth": "",
         "email": ""
      },
      "cp.icr.io": {
         "auth": "REPLACE_WITH_BASE64_ENCODED_KEY_FROM_PREVIOUS_STEP",
         "username": "cp",
         "password": "REPLACE_WITH_GENERATED_ENTITLEMENT_KEY"
      }
   }
}
```

4. Use the contents of the `temp_config.json` file, and apply the updated config to the OpenShift cluster.

   ```
   oc set data secret/pull-secret -n openshift-config --from-
   file=.dockerconfigjson=temp_config.json
   ```

   To verify that your pull-secret is updated with your new authority, issue the following command and confirm that your authority is present:

   ```
   oc get secret/pull-secret -n openshift-config -ojson | \
   jq -r '.data[".dockerconfigjson"]' | \
   base64 -d -
   ```

5. This update is rolled out to all nodes, which can take some time depending on the size of your cluster.
6. Enter the following command to remove the temporary files that were created:

   ```
   rm authority.json temp_config.json
   ```

On-premise

Before installing IBM Spectrum Scale container native on an OpenShift cluster running on-premise, ensure the following is completed.

Red Hat OpenShift Configuration on-premise

Apply additional configurations to Red Hat OpenShift installation for IBM Spectrum Scale container native cluster to operator correctly. For more information, see Installing in Red Hat OpenShift documentation.

Note: The configuration tasks shown can also be handled during the Red Hat OpenShift Container Platform installation by adding day-1 kernel arguments. For more information, see Installation Configuration in Red Hat OpenShift documentation.

Applying Machine Config Operator (MCO) Settings

The following machine configuration is provided as a convenience to easily change the following configuration:

- **Pid Limits**: Ensure that the pid limits is at least set to 4096. Insufficient pid limits will cause the GPFS daemon to crash during I/O.
  - On OpenShift Container Platform < 4.11, increase this by using the provided MCO sample files below.
  - On OpenShift Container Platform >= 4.11, the `KubeletConfig` defaults `podPidsLimit` to 4096.
- **Kernel Devel/Header Packages**: Install the kernel related packages for IBM Spectrum Scale to successfully build its portability layer.
- **Increase vmalloc kernel parameter**: Modify the kernel parameters that are required to operate properly with Red Hat CoreOS. It applies only to the IBM Spectrum Scale running on Linux on Z.

To use them, apply the corresponding file based on your OCP version and machines architecture.

- If you are running x86_64, enter the following command:

For 4.10:

```
oc apply -f https://raw.githubusercontent.com/IBM/ibm-spectrum-scale-container-native/
v5.1.7.0/generated/scale/mco/ocp4.10/mco_x86_64.yaml
```

For 4.11:

```
oc apply -f https://raw.githubusercontent.com/IBM/ibm-spectrum-scale-container-native/
v5.1.7.0/generated/scale/mco/ocp4.11/mco_x86_64.yaml
```

For 4.12:

```
oc apply -f https://raw.githubusercontent.com/IBM/ibm-spectrum-scale-container-native/
v5.1.7.0/generated/scale/mco/ocp4.12/mco_x86_64.yaml
```

- If you are running ppc64le, enter the following command:

  For 4.10:

  ```
  oc apply -f https://raw.githubusercontent.com/IBM/ibm-spectrum-scale-container-native/
  v5.1.7.0/generated/scale/mco/ocp4.10/mco_ppc64le.yaml
  ```

  For 4.11:

  ```
  oc apply -f https://raw.githubusercontent.com/IBM/ibm-spectrum-scale-container-native/
  v5.1.7.0/generated/scale/mco/ocp4.11/mco_ppc64le.yaml
  ```

  For 4.12:

  ```
  oc apply -f https://raw.githubusercontent.com/IBM/ibm-spectrum-scale-container-native/
  v5.1.7.0/generated/scale/mco/ocp4.12/mco_ppc64le.yaml
  ```

- If you are running s390x, enter the following command:

  For 4.10:

  ```
  oc apply -f https://raw.githubusercontent.com/IBM/ibm-spectrum-scale-container-native/
  v5.1.7.0/generated/scale/mco/ocp4.10/mco_s390x.yaml
  ```

  For 4.11:

  ```
  oc apply -f https://raw.githubusercontent.com/IBM/ibm-spectrum-scale-container-native/
  v5.1.7.0/generated/scale/mco/ocp4.11/mco_s390x.yaml
  ```

  For 4.12:

  ```
  oc apply -f https://raw.githubusercontent.com/IBM/ibm-spectrum-scale-container-native/
  v5.1.7.0/generated/scale/mco/ocp4.12/mco_s390x.yaml
  ```

Compact clusters support

You can deploy compact-3-node clusters on resource constrained environments in Red Hat OpenShift Container Platform 4.5 and later.

For more information, see Delivering a Three-node Architecture for Edge Deployments in Red Hat Hybrid Cloud documentation.

In this configuration, ensure that the system is sized correctly to operate smoothly. In compact clusters, the Kubernetes control plane, IBM Spectrum Scale container native, and user applications will all compete for the same resources in the cluster (CPU, memory, network, local disk, etc). High application load can impact the control plane resources, causing the OpenShift cluster to become unusable or unstable. etcd is very sensitive to latency and could present as frequent leader elections, and other instabilities. IBM Spectrum Scale container native may also take down the filesystem if resources are constrained.

For more information, see Recommended etcd practices in Red Hat OpenShift documentation.

Schedulable control plane nodes

To allow pod placement for master nodes also known as control plane nodes, ensure that they are configured as schedulable. By default, control plane nodes are not schedulable.

Verify that `mastersSchedulable` is set to `true` by entering the following command:

```
oc get schedulers.config.openshift.io cluster -ojson | jq -r ".spec.mastersSchedulable"
```

If this value is not `true`, patch the cluster by entering the following command:

```
oc patch schedulers.config.openshift.io cluster --type='json' \
-p='[{"op": "replace", "path": "/spec/mastersSchedulable", "value":true}]'
```

For more information, see Configuring control plane nodes as schedulable in Red Hat OpenShift documentation.

Applying Machine Config Operator (MCO) settings

Similar to the configuration tasks that are required for the workers nodes, these MCO settings must also be applied to the master nodes in a compact-cluster environment.

1. Apply the correct sample file based on the OCP version and machine architecture.

   - If you are running x86_64, enter the following commands for the relevant versions:

     For 4.10:

     ```
     curl https://raw.githubusercontent.com/IBM/ibm-spectrum-scale-container-native/v5.1.7.0/
     generated/scale/mco/ocp4.10/mco_x86_64.yaml | sed 's/worker/master/g' | oc apply -f -
     ```

     For 4.11:

     ```
     curl https://raw.githubusercontent.com/IBM/ibm-spectrum-scale-container-native/v5.1.7.0/
     generated/scale/mco/ocp4.11/mco_x86_64.yaml | sed 's/worker/master/g' | oc apply -f -
     ```

     For 4.12:

     ```
     curl https://raw.githubusercontent.com/IBM/ibm-spectrum-scale-container-native/v5.1.7.0/
     generated/scale/mco/ocp4.12/mco_x86_64.yaml | sed 's/worker/master/g' | oc apply -f -
     ```

   - If you are running ppc64le, enter the following command:

     For 4.10:

     ```
     curl https://raw.githubusercontent.com/IBM/ibm-spectrum-scale-container-native/v5.1.7.0/
     generated/scale/mco/ocp4.10/mco_ppc64le.yaml | sed 's/worker/master/g' | oc apply -f -
     ```

     For 4.11:

     ```
     curl https://raw.githubusercontent.com/IBM/ibm-spectrum-scale-container-native/v5.1.7.0/
     generated/scale/mco/ocp4.11/mco_ppc64le.yaml | sed 's/worker/master/g' | oc apply -f -
     ```

     For 4.12:

     ```
     curl https://raw.githubusercontent.com/IBM/ibm-spectrum-scale-container-native/v5.1.7.0/
     generated/scale/mco/ocp4.12/mco_ppc64le.yaml | sed 's/worker/master/g' | oc apply -f -
     ```

   - If you are running s390x, enter the following command:

     For 4.10:

```
curl https://raw.githubusercontent.com/IBM/ibm-spectrum-scale-container-native/v5.1.7.0/
generated/scale/mco/ocp4.10/mco_s390x.yaml | sed 's/worker/master/g' | oc apply -f -
```

For 4.11:

```
curl https://raw.githubusercontent.com/IBM/ibm-spectrum-scale-container-native/v5.1.7.0/
generated/scale/mco/ocp4.11/mco_s390x.yaml | sed 's/worker/master/g' | oc apply -f -
```

For 4.12:

```
curl https://raw.githubusercontent.com/IBM/ibm-spectrum-scale-container-native/v5.1.7.0/
generated/scale/mco/ocp4.12/mco_s390x.yaml | sed 's/worker/master/g' | oc apply -f -
```

2. Validate the MCO settings against the `master` pool.

   For more information, see "Validating OpenShift Configuration" on page 42.

3. Add the required node selector specified in the `Cluster` CR node selector to the master nodes where you want to run a core pod. For more information, see "Node selectors" on page 51.

Red Hat OpenShift Service on AWS

Red Hat OpenShift service on AWS (ROSA) is a fully managed service that provides an enterprise-ready Red Hat OpenShift cluster to run applications on AWS. It offers an easy and quick deploy of an operational Red Hat OpenShift cluster without the concerns to administer the underlying infrastructure. The service also provides a range of features to help improve the security and reliability of Red Hat OpenShift clusters, including automated backups and recovery, fine-grained access control, and integration with AWS security services.

For more information on ROSA, see Red Hat OpenShift Service on AWS Documentation.

Support matrix for IBM Spectrum Scale container native and ROSA

The following table lists the support matrix for IBM Spectrum Scale container native and Red Hat OpenShift Service on AWS (ROSA).

Table 8.

ROSA CLI version	OCP version	Autoscaling Workers	Spot Instances
1.2.14	4.11.x, 4.12.x	No	No

Note: For issues related to IBM Spectrum Scale container native, contact IBM Support.

For problems related to Red Hat OpenShift and AWS infrastructure (ROSA), contact AWS support.

Red Hat OpenShift Configuration on AWS

You must modify the Red Hat OpenShift Service on AWS (ROSA) platform for IBM Spectrum Scale container native to operate correctly.

Add IBM Spectrum Scale labels to the machine pool

Modify the default machine pool.

ROSA provides a Command Line Interface (CLI) which can be obtained from GitHub for editing node labels, which can be used to customize the machine pool used for running IBM Spectrum Scale container native. When you install ROSA, it creates a machine pool named `Default` that includes nodes with roles `worker` and `worker,infra`.

To use these nodes for running IBM Spectrum Scale container native, you can modify the `Default` machine pool as follows:

```
rosa edit machinepool -c <rosa-cluster-name> Default --labels node-role.kubernetes.io/
scale=,scale.spectrum.ibm.com/daemon-selector= --replicas 3
```

Note: It is recommended to configure a minimum of 3 replicas.

If you prefer to isolate your workload to a custom machine pool, you can create one and assign labels to the nodes using the following command:

```
rosa create machinepool -c <rosa-cluster-name> --name=<custom-pool-name> --labels node-
role.kubernetes.io/scale=,scale.spectrum.ibm.com/daemon-selector=  --replicas 3
```

Note: It is recommended to configure a minimum of 3 replicas.

Create a service account for applying Machine Config Operator (MCO)

As a regular user, you are prevented to modify Red Hat OpenShift managed resources. Therefore, it is advised to create a service account for Machine Config Operator (MCO) by following the steps below:

1. Create a service account for Machine Config Operator (MCO):

   ```
   oc create serviceaccount scale-mco-sa
   ```

2. Add a role to service account. Below command adds a `cluster-admin` role to a service account `scale-mco-sa`.

   ```
   oc adm policy add-cluster-role-to-user cluster-admin -z scale-mco-sa
   ```

3. Execute the command below to create a new token `scale-mco-sa` for the service account and store it in the `MCO_SA_TOKEN` environment variable.

   ```
   export MCO_SA_TOKEN=$(oc create token scale-mco-sa)
   ```

4. Log in to Red Hat OpenShift using the new token.

   ```
   oc login --token=$MCO_SA_TOKEN
   ```

For more information on how to use service account in applications, see Using service accounts in applications Documentation in the Red Hat OpenShift documentation.

Create a new Machine Config Pool

To separate out nodes with labels `scale` and `worker` into a new machine pool, execute the command below:

```
echo '
apiVersion: machineconfiguration.openshift.io/v1
kind: MachineConfigPool
metadata:
  name: scale-mcp
spec:
  machineConfigSelector:
    matchExpressions:
      - {key: machineconfiguration.openshift.io/role, operator: In, values: [worker,scale]}
  nodeSelector:
    matchLabels:
      node-role.kubernetes.io/scale: ""
' | oc apply -f -
```

Applying Machine Config Operator (MCO) Settings

Apply the following MCO setting to install the kernel related packages forIBM Spectrum Scale to successfully build its portability layer.

```
echo '
apiVersion: machineconfiguration.openshift.io/v1
kind: MachineConfig
metadata:
  labels:
    machineconfiguration.openshift.io/role: "scale"
  name: 00-worker-ibm-spectrum-scale-kernel-devel
spec:
  selector:
    matchLabels:
      machineconfiguration.openshift.io/role: "scale"
  config:
    ignition:
      version: 3.2.0
  extensions:
     - kernel-devel
' | oc apply -f -
```

For more information about validating the kernel-devel packages installed onto the machines, see "Validate Kernel Packages" on page 42.

Cleanup the service account

Once the Machine Config Operator task is successfully completed, complete the following steps to clean up the service account:

1. Log out of the Red Hat OpenShift cluster:

   ```
   oc logout
   ```

2. Log in to the Red Hat OpenShift cluster as a regular user:

   ```
   oc login <OpenShift_URL> -u <regular_user> -p <regular_user_password>
   ```

3. Delete the service account that was created for performing MCO task:

   ```
   oc delete sa scale-mco-sa
   ```

Authorize ROSA worker security group to allow IBM Spectrum Scale container native ports

Complete the following steps to authorize ROSA worker security group:

1. In AWS Management Console, navigate to the **EC2 Dashboard** in the region where ROSA is installed and select the **Security Groups** option from the navigation pane. Locate the Security Group for ROSA worker security group ID and export the following variable.

   ```
   export AWS_ROSA_WORKER_SECURITY_GROUP=<security_group_id>
   ```

2. Use the following commands to authorize Ingress Traffic to allow ports needed by IBM Spectrum Scale container native.

   ```
   aws ec2 authorize-security-group-ingress --group-id ${AWS_ROSA_WORKER_SECURITY_GROUP} --
   protocol tcp --port 12345  --source-group ${AWS_ROSA_WORKER_SECURITY_GROUP}
   aws ec2 authorize-security-group-ingress --group-id ${AWS_ROSA_WORKER_SECURITY_GROUP} --
   protocol tcp --port 1191  --source-group ${AWS_ROSA_WORKER_SECURITY_GROUP}
   aws ec2 authorize-security-group-ingress --group-id ${AWS_ROSA_WORKER_SECURITY_GROUP} --
   protocol tcp --port 60000-61000  --source-group ${AWS_ROSA_WORKER_SECURITY_GROUP}
   ```

Validating OpenShift Configuration

Red Hat OpenShift machine configuration changes are handled by the Machine Config Operator (MCO). To see the status of MCO and the resources it manages, run the following command:

```
oc get MachineConfingPool
```

Validate PID Limits

Validate that the PID limits are correctly set to the intended configured value on the nodes which will deploy core pods.

In the following example, the label `scale.spectrum.ibm.com/daemon-selector=` is used to select the nodes which will run core pods. Replace this selector to match what is configured in your cluster.

For OCP version >= 4.11, use the following command to check the kubelet `podPidsLimit` value:

```
oc get nodes -lscale.spectrum.ibm.com/daemon-selector= \
    -ojsonpath="{range .items[*]}{.metadata.name}{'\n'}" |\
    xargs -I{} oc debug node/{} -T -- chroot /host grep podPidsLimit  /etc/kubernetes/
kubelet.conf
```

If `podPidsLimit` is lower than `pids_limit` and `pids_limit` is not set to 0, then the effective container pids limit is defined by the value set in `podPidsLimit`. For more information about how to resolve this, see "pid_limits set higher than podPidLimits, but not being honored" on page 100.

For OCP version <= 4.10, use the following command to check the container runtime pids limit value:

```
 oc get nodes -lscale.spectrum.ibm.com/daemon-selector= \
    -ojsonpath="{range .items[*]}{.metadata.name}{'\n'}" |\
    xargs -I{} oc debug node/{} -T -- chroot /host crio-status config | grep pids_limit
```

Note: This command creates a debug pod for all nodes returned by the label selector. Use it with discretion if you have a large system.

Validate Kernel Packages

Validate that the `kernel-devel` package is installed on the nodes which will deploy core pods.

In the following example, the label `scale.spectrum.ibm.com/daemon-selector=` is used to select the nodes which will run core pods. Replace this selector to match what is configured in your cluster.

```
oc get nodes -lscale.spectrum.ibm.com/daemon-selector= \
-ojsonpath="{range .items[*]}{.metadata.name}{'\n'}" |\
xargs -I{} oc debug node/{} -T -- chroot /host sh -c "rpm -q kernel-devel"
```

Note: This command creates a debug pod for all nodes returned by the label selector. Use it with discretion if you have a large system.

s390x specific validation

Perform the extra validation steps if deployed on the s390x architecture.

Validate that the `vmalloc` kernel parameter is applied on the Red Hat OpenShift Container Platform worker nodes by entering the following command:

```
oc get nodes -lscale.spectrum.ibm.com/daemon-selector= \
-ojsonpath="{range .items[*]}{.metadata.name}{'\n'}" |\
xargs -I{} oc debug node/{} -T -- cat /proc/cmdline
```

In the following example, the value `vmalloc=4096G` is seen in the output at the end:

```
# oc debug node/worker1.example.com -- cat /proc/cmdline
Starting pod/worker1examplecom-debug ...
To use host binaries, run `chroot /host`
```

```
rhcos.root=crypt_rootfs random.trust_cpu=on ignition.platform.id=metal
rd.luks.options=discard $ignition_firstboot ostree=/ostree/boot.1/rhcos/
51e4c768b7c3dcec3bb63b01b9de9e8741486bf00dd4ae4df2d1ff1f872efe2e/0 vmalloc=4096G
```

Note: This command creates a debug pod for all nodes returned by the label selector. Use it with discretion if you have a large system.

Chapter 5. Installing the IBM Spectrum Scale container native operator and cluster

The installation of the IBM Spectrum Scale container native operator and cluster includes several procedures.

Labels and annotations

IBM Spectrum Scale container native assigns labels to worker nodes and allows to set memory and CPU limits on a per node basis by using a node annotation.

Designation labels

IBM Spectrum Scale container native automatically assigns designations to some worker nodes. You do not need to explicitly designate the worker nodes but if it is required then it can be done using node labels.

The following mechanisms are supported to designate IBM Spectrum Scale container native nodes:

- **Automatic** *(Recommended)* - Allows the Operator to designate the nodes automatically.
- **Manual** *(Optional)* - Allows administrators to have more control of the placement of IBM Spectrum Scale node designations (like the quorum designation) to pods on specific worker nodes.

  Note: Manual labeling requires insight about IBM Spectrum Scale and should only be used by experienced administrators.

Automatic

If a user does not label any nodes as quorum nodes, the Operator automatically applies quorum annotations to a subset of the nodes in the cluster. The number of nodes to be annotated depends on the number of nodes in a cluster:

- If the number of nodes in the cluster definition is less than 4, all nodes are designated as quorum nodes.
- If the number of nodes in the cluster definition is between 4 and 9 inclusive, 3 nodes are designated as quorum nodes.
- If the number of nodes in the cluster definition is between 10 and 18 inclusive, 5 nodes are designated as quorum nodes.
- If the number of nodes in the cluster definition is greater than 18, 7 nodes are designated as quorum nodes.

Kubernetes zones are considered if they are configured in the OpenShift cluster. The operator selects quorum nodes across all zones. For example, if there are 3 zones and 3 quorum nodes are to be designated as quorum nodes, then one node of each zone is designated as quorum node. For more information, see Kubernetes zones.

Note: The automatic node designation only works at initial cluster creation time. Once the cluster is created, the operator does not change node designations automatically, for example, if the nodes are added to the cluster. If node designations need to be changed on an existing cluster, the Manual steps can be performed, even if Automatic mode is used at cluster creation time. For more information, see “Manual” on page 46 and “Automatic” on page 45 sections.

© Copyright IBM Corp. 2020, 2023

Manual

Supported designation label values are `quorum` and `manager`. The nodes designated as `quorum` nodes also automatically assume the role of `manager`. If nodes are left without a designation label and sufficient `quorum` nodes are designated, unlabeled nodes become client nodes within the cluster.

IBM Spectrum Scale quorum designation

For more information about IBM Spectrum Scale quorum designation, see Quorum in IBM Spectrum Scale documentation. It is recommended to configure an odd number of nodes, with 3, 5, or 7 nodes being the typical numbers used.

IBM Spectrum Scale manager designation

For more information about IBM Spectrum Scale manager designation, see Manager in IBM Spectrum Scale documentation.

Node Labeling

To see the list of nodes in your cluster, enter the `oc get nodes` command:

```
# oc get nodes
NAME                   STATUS   ROLES    AGE   VERSION
master0.example.com    Ready    master   50d   v1.16.2
worker0.example.com    Ready    worker   50d   v1.16.2
worker1.example.com    Ready    worker   50d   v1.16.2
worker2.example.com    Ready    worker   50d   v1.16.2
```

The following labels can be applied to nodes in the Red Hat OpenShift cluster to dictate how the pods deployed on those nodes are designated:

```
scale.spectrum.ibm.com/designation=quorum
scale.spectrum.ibm.com/designation=manager
```

To apply a label to a node, enter the `oc label node <node name> scale.spectrum.ibm.com/designation=<designation>` command as follows:

```
oc label node worker0.example.com scale.spectrum.ibm.com/designation=quorum
```

To verify that the label was applied to the node, enter the `oc describe node <node name>` command as follows:

```
# oc describe node worker0.example.com
Name:               worker0.example.com
...
Labels:             ...
                    ...
                    scale.spectrum.ibm.com/designation=quorum
...
```

To remove a label from a node, enter the following command:

```
oc label node <node name> scale.spectrum.ibm.com/designation-
```

Note: Quorum node designations can be changed on the IBM Spectrum Scale container native cluster by manually applying or removing node labels. Manual labeling requires insight about IBM Spectrum Scale and should only be performed by experienced administrators.

Install

The installation process for IBM Spectrum Scale container native begins with applying the `install.yaml` to create and define kubernetes configuration across the following namespaces:

- ibm-spectrum-scale-operator
- ibm-spectrum-scale-dns
- ibm-spectrum-scale-csi
- ibm-spectrum-scale

Run the following command to configure the environment:

```
oc apply -f https://raw.githubusercontent.com/IBM/ibm-spectrum-scale-container-native/v5.1.7.0/
generated/scale/install.yaml
```

Verification

Validate that the following namespaces are created by running the command:

```
oc get namespaces | grep ibm-spectrum-scale
```

```
$ oc get namespaces | grep ibm-spectrum-scale
ibm-spectrum-scale                                 Active   4s
ibm-spectrum-scale-csi                             Active   27s
ibm-spectrum-scale-dns                             Active   26s
ibm-spectrum-scale-operator                        Active   26s
```

Validate that operator pods are running in the following two namespaces:

- ibm-spectrum-scale-operator

  ```
  oc get pods -n ibm-spectrum-scale-operator
  ```

  ```
  $ oc get pods -n ibm-spectrum-scale-operator
  NAME                                                     READY   STATUS    RESTARTS   AGE
  ibm-spectrum-scale-controller-manager-78df9cf866-jd89q   1/1     Running   0          78s
  ```

- ibm-spectrum-scale-csi

  ```
  oc get pods -n ibm-spectrum-scale-csi
  ```

  ```
  $ oc get pods -n ibm-spectrum-scale-csi
  NAME                                               READY   STATUS    RESTARTS   AGE
  ibm-spectrum-scale-csi-operator-7f94bfd897-w88fr   1/1     Running   0          40s
  ```

Image pull secrets (Optional)

If you are in a connected environment, create the `ibm-entitlement-key` pull secret so that deployed resources can gain permission to pull images from the IBM Cloud Container Registry.

```
oc create secret docker-registry ibm-entitlement-key \
--docker-server=cp.icr.io \
--docker-username cp \
--docker-password <REPLACE WITH ICR ENTITLEMENT KEY> -n ibm-spectrum-scale
```

Kubernetes Resources

The following sections describe Kubernetes resources that need to be defined to the OpenShift cluster to drive features of IBM Spectrum Scale container native cluster.

The following table describes the custom resource definitions (CRDs) managed by the IBM Spectrum Scale container native operator:

Table 9. IBM Spectrum Scale container native cluster custom resources

Resource	Short name	Description
`cluster`	`gpfs`	Set attributes for the IBM Spectrum Scale container native cluster.
`callhome`	none	Configures IBM Spectrum Scale callhome functionality.
`remoteclusters`	`remotegpfs`	Provide configuration details to the IBM Spectrum Scale remote cluster. For more information, see Filesystem section.
`filesystem`	`fs`	Configure the filesystems for the container native cluster.
`encryptionconfig`	`ec`	Allows users to configure encryption functionality.

Cluster

A cluster definition is required to declare the properties of the IBM Spectrum Scale container native cluster. The following steps describe creating a cluster CR.

1. Download a copy of the sample `cluster.yaml` from GitHub.

   ```
   curl -fs https://raw.githubusercontent.com/IBM/ibm-spectrum-scale-container-native/v5.1.7.0/
   generated/scale/v1beta1/cluster/cluster.yaml > cluster.yaml || echo "Failed to download
   Cluster Sample CR"
   ```

2. Make changes specific to your installation.

   For more details on how to fill out the sections of the Cluster specification, see “Cluster spec” on page 48.

3. Set the labels for the `nodeSelector` in the cluster `spec.daemon.nodeSelector` field. The `nodeSelector` in the sample Cluster CR is defined as `scale.spectrum.ibm.com/daemon-selector: ""`. For example, if you want scale core pods to run on the OpenShift worker nodes, `node-role.kubernetes.io/worker`, then apply the following command to set the `nodeSelector` label on the worker nodes:

   ```
   oc label nodes -lnode-role.kubernetes.io/worker= scale.spectrum.ibm.com/daemon-selector=
   ```

4. After you have made the changes, apply the cluster yaml using the following command:

   ```
   oc apply -f cluster.yaml
   ```

   Note: If you apply the cluster CR and do not see any core pods being created, check that you have correctly labeled your nodes with the configured `daemon.nodeSelector`.

After deployed, use the command `'oc edit cluster ibm-spectrum-scale'` to modify properties of the custom resource.

Cluster spec

The following table describes the properties for `Cluster`:

Table 10. Cluster property and description

Property	Required	Default	Description
license	Yes	Not accepted	The license section allows the user to accept the license and specify the edition of IBM Spectrum Scale. For more information, see License.
license.accept	Yes	False	The license must be accepted. Read the license and change to `true` to accept. For more information, see License.
license.license	Yes	data-access	It specifies the IBM Spectrum Scale edition, `data-access` or `data-management`. For more information, see License.
daemon	Yes	N/A	It specifies the configuration of the GPFS daemons. For more information, see "Daemon" on page 50.
daemon.nodeSelector	No	scale.spectrum.ibm.com /daemon-selector	Allows user to set a `nodeSelector` to indicate which OpenShift nodes to run scale core pods. For more information, see "Daemon" on page 50.
grafanaBridge	No	Disabled	It specifies the configuration of the Grafana Bridge. For more information, see "Grafana bridge" on page 53.
gui	No	N/A	It specifies additional configuration of the GUI.
pmcollector	No	N/A	It specifies the additional configuration of the pmcollector.

License

The `license` section allows you to accept and choose the IBM Spectrum Scale edition that needs to be deployed in the IBM Spectrum Scale container native cluster. You must complete the following activities:

- Review the appropriate license documentation through the URL in the CR.
- Accept the license by specifying `true` in the `license.accept` field.

- Supply the edition being used in the `license.license` field.

The sample CR defaults to `data-access` under the `license.license` field, indicating IBM Spectrum Scale Data Access Edition. If you need the IBM Spectrum Scale Data Management Edition, then change the value in `license.license` to `data-management`.

Specifying an edition without proper entitlement results in image pull failures during deployment.

```
apiVersion: scale.spectrum.ibm.com/v1beta1
kind: Cluster
spec:
  ...
  license:
    accept: true
    license: data-access
```

Enter the `oc explain cluster.spec.license` command to view more details.

Daemon

The daemon section in the cluster specification specifies configuration for the IBM Spectrum Scale core pods.

Cluster name override

If you have multiple container native clusters that are attempting to remote mount from a single IBM Spectrum Scale storage cluster, each client GPFS cluster must have a unique GPFS Cluster name. Depending on the configuration of your Red Hat OpenShift cluster, the default cluster may not be unique and cause failures to mount the filesystem.

The cluster name is created by taking `ibm-spectrum-scale` and adding it to the base domain of the Red Hat OpenShift cluster (`oc get dns cluster -ojson | jq -r '.spec.baseDomain'`). If this cluster name is not unique, use the `daemon.clusterNameOverride` field to override the generated name.

Note: The `clusterNameOverride` field must be configured prior to cluster creation. This field cannot be changed **after** a cluster is created. Setting **after** a cluster is created will require a full cleanup and redeployment of the cluster.

If using the following value in the clusterNameOverride field:

```
apiVersion: scale.spectrum.ibm.com/v1beta1
kind: Cluster
...
spec:
  daemon:
    clusterNameOverride: ocp.c2.example.com
```

The container native cluster created will have the following GPFS cluster name:

```
mmlssh-4.4# mmlscluster

GPFS cluster information
========================
  GPFS cluster name:         ocp.c2.example.com
...
...
```

Tolerations

The `spec.daemon.tolerations` section allows the user to configure additional tolerations to determine where IBM Spectrum Scale core pods can be scheduled. For more information on tolerations, see Taints and Tolerations in Kubernetes documentation.

The operator will roll out the configured tolerations to the IBM Spectrum Scale core pods.

Enter the `oc explain cluster.spec.daemon.tolerations` command to view more details.

Node selectors

The `daemon.nodeSelector` section allows you to configure a nodeSelector to determine where IBM Spectrum Scale pods can be deployed.

The Operator checks that a node has all defined labels present in order to deem a node eligible to deploy IBM Spectrum Scale pods. In the cluster CR sample, the operator will deploy IBM Spectrum Scale pods on nodes with the `scale.spectrum.ibm.com/daemon-selector` label:

```
apiVersion: scale.spectrum.ibm.com/v1beta1
kind: Cluster
spec:
  ...
  daemon:
    nodeSelector:
      scale.spectrum.ibm.com/daemon-selector: ""
```

Enter the `oc explain cluster.spec.daemon.nodeSelector` command to view more details. For more information, see "Compact clusters support" on page 37.

Host aliases

It is highly recommended that a proper DNS is configured in your environment.

The `daemon.hostAliases` section allows for user defined entries to be added into the IBM Spectrum Scale CoreDNS service handling name resolution for the core pods.

For example, if the core pods are unable to resolve hostname of the servers in the storage cluster by DNS, their hostname and their IP addresses can be specified in the `hostAliases` as follows:

```
apiVersion: scale.spectrum.ibm.com/v1beta1
kind: Cluster
spec:
  ...
  daemon:
    hostAliases:
      - hostname: node1.example.com
        ip: 10.0.0.1
      - hostname: node2.example.com
        ip: 10.0.0.2
```

The IBM Spectrum Scale CoreDNS service only handles name resolution for the core pods. For `RemoteCluster` CR, the hostname provided in the `remotecluster.spec.gui.host` field must be DNS resolvable and using host aliases is not a valid workaround.

Enter the `oc explain cluster.spec.daemon.hostAliases` command to view more details.

Cluster profile

The `daemon.clusterProfile` allows the user to set default IBM Spectrum Scale configuration parameters for the cluster at cluster creation time.

```
apiVersion: scale.spectrum.ibm.com/v1beta1
kind: Cluster
spec:
  daemon:
    ...
    clusterProfile:
      controlSetxattrImmutableSELinux: "yes"
      enforceFilesetQuotaOnRoot: "yes"
      ignorePrefetchLUNCount: "yes"
      initPrefetchBuffers: "128"
      maxblocksize: 16M
      prefetchPct: "25"
      prefetchTimeout: "30"
```

Note: Changing the values in the `clusterProfile` is not supported and must be avoided unless advised by IBM Support.

There are two exceptions where changing values in the `clusterProfile` is supported. For more information, see "Cluster profile - ephemeral port range" on page 52 and "Ephemeral port ranges" on page 13.

Enter the `oc explain cluster.spec.daemon.clusterProfile` command to view more details.

Cluster profile - ephemeral port range

If the storage cluster has the ephemeral port range configured, you need to set `tscCmdPortRange` on the container native cluster to match the range.

For example, if the storage cluster is configured to use port range, 60000-61000, set this value under the `clusterProfile` section in the `Cluster` CR.

```
apiVersion: scale.spectrum.ibm.com/v1beta1
kind: Cluster
spec:
  ...
  daemon:
    clusterProfile:
      ...
      tscCmdPortRange: "60000-61000"
```

Roles

The `daemon.roles` section under the Cluster spec allows the user to fine tune memory and CPU requests using the `resources` object on the nodes that are part of specific IBM Spectrum Scale roles. For more information on Request and Limits, see Resource Management for Pods and Containers in Kubernetes documentation.

- `client` role: For `client` role, the configuration recommendation is 2 CPU and 4GiB. On systems with a lot of CPU cores, big memory, and/or high speed network, the storage performance might increase with higher values. Encryption and compression of PVs result in higher CPU load, therefore, higher resource values can be beneficial. On smaller systems and/or applications with low I/O workload, 1 CPU and 2GiB can be set.

  Note: Low resource configurations may yield poor performance.

The following describes the resource properties of the core pods:

- Limits are set to the capacity of the nodes
- Requests, if not specified in the cluster spec for the roles is set to 25% of the capacity of the nodes

For example, to set memory and CPU requests for the `client` role, specify the values under `spec.daemon.roles.resources`:

Note: These values must be set at cluster creation time. Changes made after the cluster is created will not take effect until the pods restart.

```
apiVersion: scale.spectrum.ibm.com/v1beta1
kind: Cluster
spec:
  ...
  daemon:
    roles:
    - name: client
      resources:
        memory: "40G"
        cpu: "4"
```

Note: For s390x, the sample cluster CR ships with "4G" memory request and you may need to reduce the memory request to "2G" if your hardware does not have enough physical memory.

Enter the oc explain cluster.spec.daemon.roles command to view more details.

Grafana bridge

The grafanaBridge section allows the user to enable the deployment of the IBM Spectrum Scale bridge for Grafana application. For more information, see IBM Spectrum Scale bridge for Grafana repository in Github.

Specify grafanaBridge: {} to enable Grafana Bridge:

```
apiVersion: scale.spectrum.ibm.com/v1beta1
kind: Cluster
spec:
  ...
  grafanaBridge: {}
```

Enter the oc explain grafanabridge.spec command to view more details.

Infrastructure nodes

The GUI pods, pmcollector pods, and Grafana bridge pods can be placed on OpenShift infrastructure nodes. At least two infrastructure nodes are required because two replicas of GUI and pmcollector pods have to run on different infrastructure nodes. For more information, see "Grafana bridge" on page 53.

GUI, pmcollector, and Grafana Bridge pods

The gui, pmcollector, and grafanaBridge sections allow to specify a Kubernetes Node Selector and Kubernetes Taints and Tolerations.

If the OpenShift infrastructure nodes are labeled with node-role.kubernetes.io/infra="" and have node-role.kubernetes.io/infra:NoSchedule and node-role.kubernetes.io/infra:NoExecute taints, add the following lines to run the gui, pmcollector, and grafanaBridge sections:

```
apiVersion: scale.spectrum.ibm.com/v1beta1
kind: Cluster
spec:
  ...
  gui:
    nodeSelector:
      node-role.kubernetes.io/infra: ""
    tolerations:
    - effect: NoSchedule
      key: node-role.kubernetes.io/infra
      operator: Exists
    - effect: NoExecute
      key: node-role.kubernetes.io/infra
      operator: Exists
  pmcollector:
    nodeSelector:
      node-role.kubernetes.io/infra: ""
    tolerations:
    - effect: NoSchedule
      key: node-role.kubernetes.io/infra
      operator: Exists
    - effect: NoExecute
      key: node-role.kubernetes.io/infra
      operator: Exists
  grafanaBridge:
    nodeSelector:
      node-role.kubernetes.io/infra: ""
    tolerations:
    - effect: NoSchedule
      key: node-role.kubernetes.io/infra
      operator: Exists
    - effect: NoExecute
      key: node-role.kubernetes.io/infra
      operator: Exists
```

The Grafana Bridge is optional.

Spectrum Scale core pods

When using infrastructure pods as described above, it is required that IBM Spectrum Scale core pods run on the infrastructure nodes as well.

- Label all worker nodes and infrastructure nodes with a common label. The sample cluster CR ships with a node selector of `scale.spectrum.ibm.com/daemon-selector: ""`, use the following commands to set the label onto worker and infrastructure nodes:

  ```
  oc label nodes --selector node-role.kubernetes.io/worker scale.spectrum.ibm.com/daemon-
  selector=""

    oc label nodes --selector node-role.kubernetes.io/infra scale.spectrum.ibm.com/daemon-
  selector=""
  ```

  Note: The IBM Spectrum Scale core pods tolerate the `NoExecute` and `NoSchedule` taint. Therefore, the core pods will run on infrastructure nodes even if these taints are present.

Cluster Status

Status `Conditions` can be viewed as a snapshot of the current and most up-to-date status of a `Cluster`.

- The `Success` condition is set to `True` if the `Cluster` is successfully configured.

Callhome

To enable call home functionality, create a `CallHome` custom resource to the kubernetes cluster. The following steps describe creating a `CallHome` CR.

1. Download a copy of the sample `callhome.yaml` from GitHub.

   ```
   curl -fs https://raw.githubusercontent.com/IBM/ibm-spectrum-scale-container-native/v5.1.7.0/
   generated/scale/v1beta1/callhome/callhome.yaml > callhome.yaml || echo "Failed to download
   Callhome Sample CR"
   ```

2. Edit the callhome.yaml file and make changes specific to your installation.

   For more details on how to fill out the sections of the Callhome specification, see "Callhome spec" on page 54.

3. After you have made your changes, apply the callhome yaml using the following command:

   ```
   oc apply -f callhome.yaml
   ```

4. View the callhome resources using the following:

   ```
   oc get callhome -n ibm-spectrum-scale
   ```

Once deployed, use the **`oc edit callhome -n ibm-spectrum-scale`** command to modify properties of the resource.

Note: Call home can be enabled, modified, or disabled at any time.

For more information, see Understanding call home in the IBM Spectrum Scale documentation.

Callhome spec

The following table describes the properties for `Callhome`:

Table 11. Callhome property and description

Property	Required	Default	Description
companyEmail	Yes	None	The address of the system administrator who can be contacted by the IBM Support. Usually this e-mail address is directed towards a group or task e-mail address. For example, itsupport@mycompanyname.com.
companyName	Yes	None	The company to which the contact person belongs. This name can consist of any alphanumeric characters and these non-alphanumeric characters are '-', '_', '.', ' ', ','.
countryCode	Yes	None	The two-letter upper-case country codes as defined in ISO 3166-1 alpha-2.
customerID	Yes	None	The customer ID of the system administrator who can be contacted by the IBM Support. This can consist of any alphanumeric characters and these non-alphanumeric characters are '-', '_', '.'.
license.accept	Yes	None	License must be accepted by the end user to enable Callhome.
proxy	No	None	If specified, defines a proxy server configuration.
proxy.host	Yes, if `proxy` is specified	None	The host of proxy server as hostname or IP address.
proxy.port	Yes, if `proxy` is specified	None	The port of proxy server.
proxy.secretName	Yes, if `proxy` is specified	None	The secret name of a basic authentication secret, which contains username and password for proxy server.

Note: Define only one Callhome resource to the namespace.

License agreement

To agree and accept the license, set `license.accept` property to `true`. If you do not accept the license, call home is not enabled.

```
apiVersion: scale.spectrum.ibm.com/v1beta1
kind: Callhome
...
spec:
  ...
  license:
    accept: true
```

Personal information

Under the `spec` for `Callhome`, enter your `companyName`, the `customerID` that IBM provided to you, the `companyEmail` and the `countryCode`.

Note: The countryCode is a two-letter upper case country codes as defined in ISO 3166-1 alpha-2. For example, US for the United States or DE for Germany.

Type

Set the `spec.type` to reflect the type of cluster, `test` or `production`.

Proxy (optional)

If you are using a proxy for communication, enter information about the proxy service in the `spec.proxy` field. Enter the `oc explain callhome.spec.proxy` command to view more details.

If your proxy requires authentication, you must create a kubernetes secret containing the credentials. For example, to create a secret `proxyServerSecret`, you can enter the following command:

```
oc create secret generic proxyServerSecret --from-literal=username='<proxy_username>' \
--from-literal=password='<proxy_password>' -n ibm-spectrum-scale
```

Then add your configuration into the CR:

```
apiVersion: scale.spectrum.ibm.com/v1beta1
kind: Callhome
...
spec:
  ...
  proxy:
    host: proxyserver.example.com
    port: 443
    secretName: proxyServerSecret
```

Enter the `oc explain callhome` command to view more details.

Call home Status

Status `Conditions` can be viewed as a snapshot of the current and most up-to-date status of `Callhome`.

- The `Enabled` condition is set to `True` if `Callhome` functionality is enabled by accepting the license.
- The `Success` condition is set to `True` if `Callhome` configured successfully and is able to communicate with the IBM Callhome server.

RemoteClusters

To allow the IBM Spectrum Scale container native cluster to access remote IBM Spectrum Scale storage clusters, a `'RemoteCluster'` custom resources (CR) must be defined for each storage cluster.

Before creating the `'RemoteCluster'` custom resource, ensure that you have already prepared the storage cluster. For more information, see Storage Cluster.

To assist in filling out fields in the `'RemoteCluster'` custom resource specification, see the sections below:

- "Creating secrets for storage cluster GUI Users" on page 59
- "Configuring Certificate Authority (CA) certificates" on page 60

The following steps will help guide you in creating a `RemoteCluster` resource.

1. Download a copy of the sample `remotecluster.yaml` from GitHub to use as a starting point.

   ```
   curl -fs https://raw.githubusercontent.com/IBM/ibm-spectrum-scale-container-native/v5.1.7.0/
   generated/scale/v1beta1/remotecluster/remotecluster.yaml > remotecluster.yaml || echo
   "Failed to download RemoteCluster Sample CR"
   ```

   Note: The sample `remotecluster.yaml` uses `remotecluster-sample` as the resource name. You can change this name to your liking. When creating multiple remote cluster resources, each resource must have a unique `metadata.name`.

2. Edit the `remotecluster.yaml` file and make change to the fields that are specific to your installation. For details on how to fill out the sections of the Remote Cluster specification, see RemoteCluster Spec.
3. Apply the yaml using the following command:

   ```
   oc apply -f remotecluster.yaml
   ```

4. View the remote cluster resources using the following command:

   ```
   oc get remotecluster -n ibm-spectrum-scale
   ```

Once deployed, use the `oc edit remotecluster <remotecluster-name> -n ibm-spectrum-scale>` command to modify properties of the resource.

RemoteCluster spec

The following table describes the properties for `RemoteCluster`:

Table 12. RemoteCluster field and description

Field	Required	Default	Description
metadata.name	Yes	None	The name of the CR. This is used to identify the remote storage cluster in the Filesystem CR.

Table 12. RemoteCluster field and description (continued)

Field	Required	Default	Description
contactNodes	No	None	This property is optional and provides a list of nodes from the storage cluster to be used as the remote cluster contact nodes. The names should be the daemon node names. If not specified, the operator uses any 3 nodes detected from the storage cluster.
gui	Yes	None	It specifies the details for the IBM Spectrum Scale Remote Cluster GUI.
gui.cacert	No	None	It specifies the name of the RootCA ConfigMap.
gui.csiSecretName	Yes	csi-remote-mount-storage-cluster-1	It references the secret that contains the username and password of the CSI admin user in the ibm-spectrum-scale-csi namespace.
gui.host	Yes	None	The hostname for the GUI endpoint on the storage cluster.
gui.insecureSkipVerify	No	None	The parameter controls whether a client verifies the storage cluster's GUI certificate chain and host name. If set to `true`, TLS is susceptible to machine-in-the-middle attacks. The default value is `false`.
gui.port	No	443	It specifies the port of the Remote Cluster.
gui.scheme	No	https	The default value is 'https'. No other value is supported.
gui.secretName	Yes	None	The name of the Kubernetes secret created during the storage cluster configuration.

You can define 1 or more `RemoteClusters` to the cluster, one for each Storage Cluster you want to mount filesystems from.

GUI

The `spec.gui` contains the properties that are needed to communicate with the IBM Spectrum Scale remote storage cluster.

- cacert - provides the name of the ConfigMap storing the CA Certificate for the Storage Cluster GUI. For more information, see Configuring Certificate Authority (CA) certificates.

- insecuritySkipVerify - controls whether a client verifies the server's certificate chain and host name. Default to false.
- host - provides the Remote Cluster GUI host endpoint.
- secretName - kubernetes secret name containing the credentials for the ContainerOperator user on the storage cluster.
- csiSecretName - kubernetes secret name containing the credentials for the CsiAdmin user on the storage cluster.

Limitations

Deleting a `RemoteCluster` custom resource definition does not delete the access permission of an IBM Spectrum Scale container native cluster to the file systems on a remote storage cluster.

Enter the `oc explain remotecluster.spec` command to view more details.

RemoteCluster Status

Status `Conditions` can be viewed as a snapshot of the current and most up-to-date status of a `Remotecluster` instance.

- The `Ready` condition is set to `True` if the `Remotecluster` credentials are established.

Examples

Additional remote clusters

It is possible to mount additional filesystems that are served by multiple remote clusters. Each remote cluster would need to have a resource defined into the Kubernetes cluster.

If applying the sample `remotecluster.yaml` CR from the example above, you would have a single resource defined:

```
$ oc get remotecluster -n ibm-spectrum-scale
NAME                   HOST                             READY   AGE
remotecluster-sample   guihost.example.com              True    25h
```

Duplicate the sample CR and make changes to reflect your second remote storage cluster. It is important that the `metadata.name` is changed to something unique, in this example: `xyz-storagecluster`.

After applying, you should see a second remote cluster resource defined:

```
$ oc get remotecluster -n ibm-spectrum-scale
NAME                   HOST                             READY   AGE
remotecluster-sample   guihost.example.com              True    25h
xyz-storagecluster     xyz.example.com                  True    10m
```

Creating secrets for storage cluster GUI Users

Create a secret on the Red Hat OpenShift cluster that holds the credentials for GUI users defined on the IBM Spectrum Scale Storage cluster. The secret is used by the operator to communicate with the storage cluster to configure the remote mount.

Note: The username and password specified below must match the GUI user that was created on the storage cluster. For more information, see “Creating Container Operator and CSI Users” on page 27.

Two new secrets must be added for each storage cluster being configured.

1. Create a secret for the `ContainerOperator` GUI user defined on the storage cluster.

To create a secret named `cnsa-remote-mount-storage-cluster-1` in the `ibm-spectrum-scale` namespace, enter the following command:

```
oc create secret generic cnsa-remote-mount-storage-cluster-1 --from-
literal=username='cnsa_storage_gui_user' \
   --from-literal=password='cnsa_storage_gui_password' -n ibm-spectrum-scale
```

2. Create a secret for the `CsiAdmin` GUI user defined on the storage cluster.

   To create the secret named `csi-remote-mount-storage-cluster-1` in the `ibm-spectrum-scale-csi` namespace, enter the following command:

```
oc create secret generic csi-remote-mount-storage-cluster-1 --from-
literal=username='csi_storage_gui_user' \
   --from-literal=password='csi_storage_gui_password' -n ibm-spectrum-scale-csi
```

3. Label the secret, enter the following command:

```
oc label secret csi-remote-mount-storage-cluster-1 -n ibm-spectrum-scale-csi product=ibm-
spectrum-scale-csi
```

Note: When the passwords on the storage cluster for these users change, the credentials in the secrets must be updated.

Configuring Certificate Authority (CA) certificates

IBM Spectrum Scale container native uses Transport Layer Security (TLS) verification to guarantee secure HTTPS communication with the storage cluster GUI. It verifies the server’s certificate chain and host name.

Configure a security protocol

A security protocol must be configured for use with IBM Spectrum Scale container native in one of three different ways.

Option 1 - CA Certificate ConfigMap

A ConfigMap containing the CA certificate of the storage cluster GUI must be created to allow the IBM Spectrum Scale container native operator to perform TLS verification. CA certificate data can exist in base64 encoded or decoded forms.

In the following example, we create a ConfigMap from `storage-cluster-1.crt` file. This file contains the storage cluster CA certificate data in decoded form. The decoded form must appear as shown:

```
# cat storage-cluster-1.crt
-----BEGIN CERTIFICATE-----
MIIDZDC.....................................................
............................................................
...........n/J9OJFdoXs=
-----END CERTIFICATE-----
```

Create the ConfigMap with one of the following two commands. The second command is provided to assist the users who wish to trust the self-signed certificate of the storage cluster GUI.

```
oc create configmap cacert-storage-cluster-1 --from-file=storage-cluster-1.crt=storage-
cluster-1.crt -n ibm-spectrum-scale
```

Note: By default, the storage cluster GUI self-signs a certificate that can be used in lieu of a CA certificate. This certificate can be obtained and used to create the cacert ConfigMap by entering the following command. Replace the gui host with the hostname of the storage cluster GUI.

```
oc create configmap cacert-storage-cluster-1 --from-literal=storage-cluster-1.crt="$(openssl
s_client -showcerts -connect <gui host>:443 </dev/null 2>/dev/null|openssl x509 -outform PEM)"
-n ibm-spectrum-scale
```

Option 2 - Storage Cluster uses the OpenShift Container Platform CA or a Red Hat Default CA

IBM Spectrum Scale container native automatically includes the OpenShift Container Platform CA and the default Red Hat CA bundle for storage cluster GUI communication. If the storage cluster uses the OpenShift Container Platform CA or a Red Hat trusted CA, a ConfigMap, as described in Option 1, does not need to be created for the CA certificate and the `cacert` field should be deleted from the RemoteCluster Custom Resource. For more information, see "RemoteClusters" on page 57.

Option 3 - Skip Verification

Storage cluster verification may be skipped if desired, however, TLS is susceptible to machine-in-the-middle attacks. To skip verification, the `insecureSkipVerify` option must be set to `true`, when configuring the RemoteCluster Custom Resource. For more information, see "RemoteClusters" on page 57.

Storage cluster verification

Events are posted onto the `RemoteCluster` resource if configuration is missing. For example, if secrets and ConfigMaps are missing, you may see events similar to the following sample:

```
$ oc describe remotecluster remotecluster-sample
...
Events:
  Type     Reason           Age                From           Message
  ----     ------           ----               ----           -------
  Warning  RemoteConnError  6m3s               RemoteCluster  Secret "cnsa-remote-mount-storage-
cluster-1" not found
  Warning  RemoteConnError  3s (x6 over 5m3s)  RemoteCluster  ConfigMap "cacert-storage-
cluster-1" not found
```

Filesystems

To configure a file system in the IBM Spectrum Scale for the container native cluster, a `'Filesystem'` custom resource (CR) must be defined for each filesystem you want mounted.

The filesystem is served by an IBM Spectrum Scale remote storage cluster. Ensure that you have already created a `RemoteCluster` custom resource for that storage cluster before proceeding. For more information, see "RemoteClusters" on page 57.

The following steps will guide you to create a `Filesystem` resource.

1. Download a copy of the sample `filesystem.remote.yaml` from GitHub to use as a starting point.

   ```
   curl -fs https://raw.githubusercontent.com/IBM/ibm-spectrum-scale-container-native/v5.1.7.0/
   generated/scale/v1beta1/filesystem/filesystem.remote.yaml > filesystem.remote.yaml || echo
   "Failed to download Filesystem Sample CR"
   ```

2. Edit the '`filesystem.remote.yaml`' file and make change to the fields that are specific to your installation. For details on how to fill out the sections of the Filesystem specification, see "Filesystem spec" on page 62.
3. After you have made your changes, apply the filesystem yaml using the following command:

   ```
   oc apply -f filesystem.remote.yaml
   ```

4. View the filesystem resources using the following command:

   ```
   oc get filesystem -n ibm-spectrum-scale
   ```

   Once deployed, use the `oc edit filesystem <filesystem-name> -n ibm-spectrum-scale` command to modify properties of the custom resource.

   Note: If you choose to configure an encrypted remote mounted filesystem for the IBM Spectrum Scale container native cluster you must also create an `EncryptionConfig` custom resource. For more information, see EncryptionConfig.

Filesystem spec

The following table describes the properties for `Filesystem`:

Table 13. Filesystem property and description

Property	Required	Default	Description
metadata.name	Yes	None	The name of the CR.
remote	No	None	If specified, describes the file system to be remote mounted filesystem.
remote.fs	Yes, if `remote` is specified	None	It is the name of the filesystem to mount, served by the remote cluster.
remote.cluster	Yes, if `remote` is specified	None	It is the name of the `Remote Cluster` resource.

You can define `1` or more `Filesystems`, one for each filesystem that you want to mount from the storage cluster.

Note: All filesystems will be mounted under `/mnt`. This cannot be changed.

Remote

The `spec.remote` section defines the remote filesystem properties and consists of two fields:

- `remote.cluster`: This specifies the name of the `RemoteCluster` CR that is defined that is serving the filesystem.
- `remote.fs`: This specifies the filesystem name on the remote storage cluster that we want to mount into the container native cluster.

In the following example:

```
apiVersion: scale.spectrum.ibm.com/v1beta1
kind: Filesystem
...
spec:
  remote:
    cluster: remotecluster-sample
    fs: fs1
```

The filesystem `fs1` provided by the remote cluster defined in `remotecluster-sample` will be made available in the container.

Limitations

Deleting a `Filesystem` custom resource does not un-mount or delete the file system configuration from the IBM Spectrum Scale cluster.

Enter the `oc explain filesystem.spec.remote` command to view more details.

Filesystem Status

Status `Conditions` can be viewed as a snapshot of the current and most up-to-date status of a `Filesystem` instance.

- The `Success` condition is set to `True` if the `Filesystem` is created and mounted.

Examples

Additional filesystems

To have more filesystems created into the container native cluster, you would define a custom resource for each filesystem and apply it to the Kubernetes cluster.

After following the example above, you would have a single filesystem resource defined in the cluster:

```
$ oc get filesystem -n ibm-spectrum-scale
NAME            ESTABLISHED   AGE
remote-sample   True          25h
```

1. To create another filesystem served by the same remote cluster, `remotecluster-sample`, define a yaml as follows:

   ```
   apiVersion: scale.spectrum.ibm.com/v1beta1
   kind: Filesystem
   metadata:
     labels:
       app.kubernetes.io/instance: ibm-spectrum-scale
       app.kubernetes.io/name: cluster
     name: c1-fs2
     namespace: ibm-spectrum-scale
   spec:
     remote:
       cluster: remotecluster-sample
       fs: fs2
   ```

   When applied, filesystem `fs2` hosted by the remote cluster `remotecluster-sample` will be made available to the container native cluster.

   At this point, displaying the filesystems will show the following output:

   ```
   $ oc get filesystem -n ibm-spectrum-scale -o wide
   NAME            ESTABLISHED   REMOTE CLUSTER         MAINTENANCE MODE   AGE
   remote-sample   True          remotecluster-sample   not supported      25h
   c1-fs2          True          remotecluster-sample   not supported      10m
   ```

2. To create another filesystem served by a different remote cluster, `remotecluster-sample-2`, define a yaml as follows:

   ```
   apiVersion: scale.spectrum.ibm.com/v1beta1
   kind: Filesystem
   metadata:
     labels:
       app.kubernetes.io/instance: ibm-spectrum-scale
       app.kubernetes.io/name: cluster
     name: xyz-fs2
     namespace: ibm-spectrum-scale
   spec:
     remote:
       cluster: remotecluster-sample-2
       fs: fs2
   ```

   Note: This assumes that you have already created a RemoteCluster resource called `remotecluster-sample-2` per the instructions in the RemoteClusters section.

   When applied, filesystem `fs2` hosted by the remote cluster `remotecluster-sample-2` will be made available to the container native cluster.

   At this point, displaying the filesystems will show the following output:

   ```
   $ oc get filesystem -n ibm-spectrum-scale -o wide
   NAME            ESTABLISHED   REMOTE CLUSTER           MAINTENANCE MODE   AGE
   remote-sample   True          remotecluster-sample     not supported      25h
   c1-fs2          True          remotecluster-sample     not supported      20m
   xyz-fs2         True          remotecluster-sample-2   not supported      10m
   ```

Encryption

IBM Spectrum Scale container native supports remote mount of an encrypted filesystem.

Encryption is managed through use of encryption keys stored on key server.

The following key servers are supported:

- IBM Security Guardium Key Lifecycle Manager (SKLM)

EncryptionConfig Specs

The following table describe the properties for `EncryptionConfig`:

Property	Required	Default	Description
`metadata.name`	Yes	None	The name of the CR.
`server`	Yes	None	The key server host name or IP in which encryption keys are stored.
`backupServers`	No	None	The backup key servers configured for high availability. This field is optional.
`port`	No	None	It can be used to override the default port for the key server.
`cacert`	No	None	The ConfigMap storing CA and endpoint certificates used while adding/renewing key server certificate chain.
`secret`	Yes	None	The name of the basic-auth secret containing the username and password to the key server.
`tenant`	Yes	None	The tenant name on the key server that contains encryption keys. This has to be the same tenant name that is used to store the encryption keys of the remote storage file system.
`client`	Yes	None	The key client to communicate with the key Server.
`remoteRKM`	Yes	None	The RKM ID from the storage cluster corresponding to given key server and tenant.

Limitations:

- Deleting a `EncryptionConfig` custom resource does not delete the encryption configuration from the IBM Spectrum Scale container native cluster.

  Warning: Updating `client` and `tenant` is not recommended as it causes loss of master encryption keys for that tenant.

For more information, enter the oc explain encryptionconfig.spec command.

EncryptionConfig status

Status Conditions can be viewed as a snapshot of the current and most up-to-date status of a EncryptionConfig instance.

- The Success condition is set to True if the EncryptionConfig is successfully configured.

Examples

To give IBM Spectrum Scale container native access to the encryption key server, an EncryptionConfig custom resource must be created. The configuration must add the same key server and tenant as configured on storage cluster hosting the filesystem. You can define more than one EncryptionConfig custom resource.

For more information, see Encryption in IBM Spectrum Scale documentation.

Prerequisites

- Create a secret containing the administrator username and password credentials to the key server.

  ```
  oc create secret generic keyserver-credentials -n ibm-spectrum-scale \
  --from-literal=username=<keyserver_admin_name> \
  --from-literal=password=<keyserver_admin_password>
  ```

- If using CA certificates, create the ConfigMap holding the CA certificate chain.
  1. Obtain CA certificates and endpoint/server certificates. Separate the root certificate and the intermediate certificates into the following .crt files:
     - Root certificate :root.crt
     - Server or Endpoint certificate: endpoint.crt
     - Intermediate certificates: intermediate<numeric_index>.crt
  2. Create the ConfigMap with the following command:

     ```
     oc create ConfigMap sample-ca-cert \
     --from-file=/path/to/root.crt \
     --from-file=/path/to/intermediate1.crt \
     --from-file=/path/to/intermediate2.crt \
     --from-file=/path/to/intermediate3.crt \
     --from-file=/path/to/endpoint.crt \
     -n ibm-spectrum-scale
     ```

- Encryption details from storage cluster, specifically, tenant and RKMID.

Note: If the core pods are unable to resolve the IP address of the IBM SKLM server, you can add hostAliases entries in the Cluster custom resource. For more information, see Cluster.

Configure EncryptionConfig custom resource

The following steps describe creating a EncryptionConfig CR:

1. Download a copy of the sample encrpytionconfig.remote.yaml from GitHub.

   ```
   curl -fs https://raw.githubusercontent.com/IBM/ibm-spectrum-scale-container-native/
   v5.1.7.0/generated/scale/v1beta1/encrpytionconfig/encrpytionconfig.remote.yaml >
   encryptionconfig.remote.yaml || echo "Failed to download EncryptionConfig Sample CR"
   ```

2. Make changes specific to your installation. For more information about the EncryptionConfig specification, see "EncryptionConfig Specs" on page 64.
   - Replace keyserver.example.com with your keyserver hostname

   - Replace `keyserver1.example.com`, `keyserver2.example.com`, etc with your backup keyserver hostnames
   - Replace `sampleTenant` with your tenant name
   - Replace `sampleClient` with your client name
   - Replace `sampleRKM` with your RKMID
   - If using self-signed certificates, comment out the `cacert` field in the spec

3. Apply the cluster yaml using the following command:

   ```
   oc apply -f encryptionconfig.remote.yaml
   ```

   Once deployed, use the command **`oc edit encryptionconfig <encrypyionconfig-name> -n ibm-spectrum-scale`** to modify properties of the resource.

Chapter 6. Validating Installation

The following sections will help validate the installation.

Verifying an IBM Spectrum Scale container native cluster

Verify whether the deployment of an IBM Spectrum Scale container native cluster is done correctly.

Complete the following steps:

Note: For more information, see "Debugging IBM Spectrum Scale deployment" on page 105.

1. Verify that the Operator has created a cluster by checking the pods.

   ```
   oc get pods -n ibm-spectrum-scale
   ```

   A sample output is shown:

   ```
   # oc get pods -n ibm-spectrum-scale
      NAME                                 READY   STATUS    RESTARTS    AGE
      ibm-spectrum-scale-gui-0             4/4     Running   0           5m45s
      ibm-spectrum-scale-gui-1             4/4     Running   0           2m9s
      ibm-spectrum-scale-pmcollector-0     2/2     Running   0           5m15s
      ibm-spectrum-scale-pmcollector-1     2/2     Running   0           4m11s
      worker0                              2/2     Running   0           5m43s
      worker1                              2/2     Running   0           5m43s
      worker3                              2/2     Running   0           5m45s
   ```

   Note: The following list includes considerations about the IBM Spectrum Scale cluster creation and its pods:

   - The cluster takes some time to create.
   - One core pod per node gets created on nodes matching the `nodeSelector`.
   - Core pods can take several minutes to move to Running status.
   - GUI pods do not achieve the Running status until all the core pods are in a Running status.
   - Two GUI pods are created, where the second is created after the first is moved to Running status.
   - Two pmcollector pods are created, where the second is created after the first is moved to Running status.
   - Resulting cluster should have one core pod per node as specified by the `nodeSelector`, two GUI pods, and two pmcollector pods.

2. Verify that the IBM Spectrum Scale cluster is created correctly:

   a. Enter the `mmlscluster` command:

   ```
   oc exec $(oc get pods -lapp.kubernetes.io/name=core \
   -ojsonpath="{.items[0].metadata.name}" -n ibm-spectrum-scale)  \
   -c gpfs -n ibm-spectrum-scale -- mmlscluster
   ```

   The output from the command should show that an IBM Spectrum Scale cluster is created, and all nodes as specified by the `nodeSelector` are present.

   ```
   GPFS cluster information
         ========================
         GPFS cluster name:         ibm-spectrum-scale.mycluster.example.com
         GPFS cluster id:           835278197609441888
         GPFS UID domain:           ibm-spectrum-scale.mycluster.example.com
         Remote shell command:      /usr/bin/ssh
         Remote file copy command:  /usr/bin/scp
         Repository type:           CCR

         Node  Daemon node name  IP address    Admin node name  Designation
         --------------------------------------------------------------------
   ```

© Copyright IBM Corp. 2020, 2023

```
        1   worker2.daemon.ibm-spectrum-scale.stg.mycluster.example.com.  172.29.0.145
worker2.admin.ibm-spectrum-scale.stg.mycluster.example.com.  quorum-manager-perfmon
        2   worker1.daemon.ibm-spectrum-scale.stg.mycluster.example.com.  172.29.0.146
worker1.admin.ibm-spectrum-scale.stg.mycluster.example.com.  quorum-manager-perfmon
        3   worker3.daemon.ibm-spectrum-scale.stg.mycluster.example.com.  172.29.0.148
worker3.admin.ibm-spectrum-scale.stg.mycluster.example.com.  quorum-manager-perfmon
```

b. Enter the `mmgetstate` command:

```
oc exec $(oc get pods -lapp.kubernetes.io/name=core \
      -ojsonpath="{.items[0].metadata.name}" -n ibm-spectrum-scale)  \
      -c gpfs -n ibm-spectrum-scale -- mmgetstate -a
```

The output from the command should show that the `GPFS state` for all nodes are listed as `active`.

```
Node number  Node name     GPFS state
-------------------------------------------
      1      worker0        active
      2      worker1        active
      3      worker3        active
```

3. Verify that the Remote Cluster authentication is successfully created.

   a) Get a list of the remote clusters.

   ```
   oc get remotecluster.scale  -n ibm-spectrum-scale
   ```

   b) Inspect the remote clusters and ensure that the value for `READY` is `True`.

   Example:

   ```
    # oc get remotecluster.scale -n ibm-spectrum-scale
         NAME                    HOST                           READY   AGE
         remotecluster-sample   cnsa-storage.example.ibm.com   True    30h
   ```

4. Verify that the storage cluster file system is configured:

   a) Get a list of the file systems:

   ```
   oc get filesystem.scale -n ibm-spectrum-scale
   ```

   b) Inspect the file systems and ensure that the value for `ESTABLISHED` is `True`.

   ```
    # oc get filesystem.scale -n ibm-spectrum-scale
      NAME            ESTABLISHED   AGE
      remote-sample   True          30h
   ```

5. Manually verify that the file system is mounted by using the `mmlsmount` command.

```
oc exec $(oc get pods -lapp.kubernetes.io/name=core \
   -ojsonpath="{.items[0].metadata.name}" -n ibm-spectrum-scale)  \
   -c gpfs -n ibm-spectrum-scale -- mmlsmount remote-sample -L
```

Example output:

```
File system remote-sample (gpfs1.local:fs1) is mounted on ...
   ...
   172.29.0.148    worker3.daemon.ibm-spectrum-scale.stg.mycluster.example.com. ibm-spectrum-
scale.mycluster.example.com
   172.29.0.146    worker1.daemon.ibm-spectrum-scale.stg.mycluster.example.com. ibm-spectrum-
scale.mycluster.example.com
   172.29.0.145    worker2.daemon.ibm-spectrum-scale.stg.mycluster.example.com. ibm-spectrum-
scale.mycluster.example.com
```

6. Verify that there are no problems reported in the operator status and events. For more information, see "Status and events" on page 69.

7. Verify that the CSI pods are up and running.

```
oc get pods -n ibm-spectrum-scale-csi
```

8. Verify that the Core DNS pods are up and running. There will be at least one Core DNS pod per core pod.

   ```
   oc get pods -n ibm-spectrum-scale-dns
   ```

Status and events

The custom resource (CR) objects contain helpful information which can be retrieved by entering the oc describe command.

For each object, a Status attribute provides the last observed state of the resource. In the retrieved information, a log of recent Events pertaining to the resource is also shown. This information can be helpful to check the desired state of the resource or when debugging with the IBM Spectrum Scale container native cluster. For more information, see Application Introspection and Debugging in Kubernetes documentation.

The oc describe <CR> -n ibm-spectrum-scale command is used to view the *status* and *events* of the custom resources, such as cluster, daemon, filesystem, remotecluster, callhome, and others.

The *Status* can be seen in the Conditions section:

```
$ oc describe callhome.scale -n ibm-spectrum-scale
...
Status:
  Conditions:
    Last Transition Time:  2021-08-31T12:54:05Z
    Message:               Callhome is enabled.
    Reason:                Enabled
    Status:                True
    Type:                  Enabled
    Last Transition Time:  2021-08-31T12:54:07Z
    Message:               Successfully tested connection to the IBM Callhome Server.
    Reason:                TestPassed
    Status:                True
    Type:                  Success
  Mode:                    test
...
```

A *Condition* has the following fields:

- *Type:* Type of condition.
- *Status:* Status of the condition, one of True, False or Unknown.
- *Reason:* The reason contains a programmatic identifier indicating the reason for the condition's last transition.
- *Message:* Message is a human readable message indicating details about the transition.
- *Last Transition Time:* This is the last time the condition transitioned from one status to another (For example, from False to True).

The Events section of oc describe output lists the *Events*:

```
$ oc describe callhome.scale -n ibm-spectrum-scale
...
Events:
  Type    Reason      Age   From      Message
  ----    ------      ----  ----      -------
  Normal  NodeUpdate  44m   Callhome  Callhome was enabled on 0 nodes before, but now it's
enabled on all 5 nodes.
  Normal  Configured  44m   Callhome  Successfully updated callhome configuration.
Customer=IBM, CustomerID=123456, Email=sroth@de.ibm.de, Country=DE, Type=test
  Normal  Enabled     44m   Callhome  Callhome has been enabled.
```

Enter the oc get crd | grep ibm command to see a full list of CRs that can be checked for status and events with the oc describe command.

Note:

- The *Events* disappear after they are created.
- The *Status* and *Events* listed above are examples and they look different on your system.

Chapter 7. Using IBM Spectrum Scale GUI

The following section describes how to use functionality in the IBM Spectrum Scale container native cluster.

IBM Spectrum Scale container native GUI

You can manage and monitor cluster and node information through the IBM Spectrum Scale container native GUI.

OpenShift Users

OpenShift roles are mapped to two IBM Spectrum Scale GUI user groups. Details are provided in the following table:

Table 14. Roles and privileges

Roles			Privileges		
OCP role	**GUI role**	**View**	[1] **Download snap**	[2] **Manage events**	**Test connection for call home**
cluster-admin	Maintenance	Yes	Yes	Yes	Yes
ibm-spectrum-scale-maintenance	Maintenance	Yes	Yes	Yes	Yes
ibm-spectrum-scale-monitor	Monitor	Yes	No	No	No

[1] Ability to download master and non-master snaps.

[2] Ability to mark events as resolved, hiding resolved tips and notifications.

Add a role to a user or a group

Users created on the Openshift Container Platform (OCP) having a role of `ibm-spectrum-scale-maintenance` or `ibm-spectrum-scale-monitor` can log in to the IBM Spectrum Scale container native GUI through Single Sign On using the OAuth implementation.

```
oc adm policy add-cluster-role-to-user <role> <user>
```

```
oc adm policy add-cluster-role-to-group <role> <group>
```

Accessing the IBM Spectrum Scale GUI

To access the IBM Spectrum Scale GUI, complete the following steps:

1. In a browser, navigate to `https://ibm-spectrum-scale-gui-ibm-spectrum-scale.apps.<ocp domain>/`, where `<ocp domain>` is the domain of your OpenShift cluster. You should see the IBM Spectrum Scale GUI login page.

   If the domain is `ocp4.example.com`, the URL would be `https://ibm-spectrum-scale-gui-ibm-spectrum-scale.apps.ocp4.example.com`.
2. Click **Sign in**, which redirects to the Red Hat Openshift Container Platform login page.

© Copyright IBM Corp. 2020, 2023

3. Authenticate by using your OCP user credentials.

   On success, you are redirected back to the IBM Spectrum Scale GUI home page.

Chapter 8. Upgrading

Refer to the following sections to upgrade IBM Spectrum Scale container native to the next version:

Supported upgrade paths

Use the following table to understand the supported upgrade paths for IBM Spectrum Scale container native.

Table 15. Supported upgrade paths

Upgrade from	Upgrade to 5.1.3.x	Upgrade to 5.1.4.x	Upgrade to 5.1.5.0	Upgrade to 5.1.6.0	Upgrade to 5.1.7.0
5.1.6.0	--	--	--	--	Yes
5.1.5.0	--	--	--	Yes	Yes
5.1.4.x	--	Yes	Yes	No	No
5.1.3.x	Yes	Yes	No	No	No
5.1.2.1	Yes	Yes	No	No	No

Note: If upgrading from IBM Spectrum Scale container native less than 5.1.5.0, it is required to first upgrade to 5.1.5.0 before continuing to higher levels.

Upgrading IBM Spectrum Scale container native

The following section describes how to upgrade the IBM Spectrum Scale container native cluster.

While an upgrade is in progress, do not perform the following:

- Do not make changes to the `Cluster` custom resource.
- Do not attempt to add a node to the cluster.

During an upgrade, the IBM Spectrum Scale operator orchestrates the upgrade procedure in a rolling node-by-node fashion. Each node will be:

- Cordoned (tainted unschedulable)
- Drained (its pods safely evicted and rescheduled to other available nodes)
- Rebooted, if necessary
- Uncordoned (returning it to normal service)

After the node is schedulable, IBM Spectrum Scale and IBM Spectrum Scale Container Storage Interface (CSI) pods will start. Applications may fail to attach storage until the system is started.

Note: If upgrading from IBM Spectrum Scale container native less than 5.1.5.0, it is required to first upgrade toIBM Spectrum Scale container native 5.1.5.0 before continuing to higher levels. For more information, see "Supported upgrade paths" on page 73.

Prerequisites

All the core pods need to be in running status.

Use the following command to check the status of the core pods:

```
oc get daemons ibm-spectrum-scale -n ibm-spectrum-scale -ojson | jq -r '.status.podsStatus'
```

Ensure that there are no pods in any of the following states:

© Copyright IBM Corp. 2020, 2023

- starting
- terminating
- unknown
- waitingForDelete

In the following example, the output shows 1 pod in `waitingForDelete`, so the upgrade should not be done at this time.

```
$ oc get daemons ibm-spectrum-scale -n ibm-spectrum-scale -ojson | jq -r '.status.podsStatus'
  {
  "running": "4",
  "starting": "0",
  "terminating": "0",
  "unknown": "0",
  "waitingForDelete": "1"
  }
```

Upgrade Procedure

Complete the following steps to upgrade:

1. Stop the running operator pod by setting the `replicas` in the deployment to 0.

   ```
   oc scale deployment ibm-spectrum-scale-controller-manager -n ibm-spectrum-scale-operator --
   replicas=0
   ```

2. Delete the old security context constraint.

   ```
   oc delete scc ibm-spectrum-scale-privileged
   ```

3. Delete the old role binding for privilege.

   ```
   oc delete rolebinding -n ibm-spectrum-scale ibm-spectrum-scale-privileged --ignore-not-found
   ```

4. Delete the `MutatingWebhookConfiguration` and `ValidatingWebhookConfiguration`. These will be created in later steps.

   ```
   oc delete MutatingWebhookConfiguration ibm-spectrum-scale-mutating-webhook-configuration
   oc delete ValidatingWebhookConfiguration ibm-spectrum-scale-validating-webhook-configuration
   ```

5. Special handling for data sharing and SELinux.

   a. Determine if any filesystems require special considerations with respect to data sharing and SELinux.

      The SELinux changes introduced in this release are aimed to improve performance of volume attachments, while maintaining existing security standards.

      Special consideration for data sharing is required if:

      - Non-containerized workloads produce/ingest filesystem data with SELinux labels.
      - Containerized workloads are using a non-default SELinux context to share data.

      Special consideration for data sharing is not needed if all workloads accessing the filesystem are containerized and fall within the default Red Hat Openshift SELinux context range.

      When changing SELinux context, applications with ReadWriteMany volumes running on nodes with mixed SELinux contexts may experience "Permission Denied" errors. Applications running on nodes with differing SELinux contexts may not have access to each other's files during the duration of the SELinux context update.

      During an upgrade of IBM Spectrum Scale container native, applications with ReadWriteMany volumes running on nodes with mixed IBM Spectrum Scale container native may experience `Permission Denied` errors. Applications running on a node prior to IBM Spectrum Scale container native 5.1.7.0 will not have access to new files that are written on a system with

IBM Spectrum Scale container native 5.1.7.0+. If the file exists prior to upgrade, it will remain accessible during upgrade and afterwards.

For more information, see "IBM Spectrum Scale container native and SELinux" on page 15.

If no filesystems require a modified SELinux context, skip this section and proceed to the next step.

b. Apply the new manifests (excluding operator).

```
oc apply -f https://raw.githubusercontent.com/IBM/ibm-spectrum-scale-container-native/
v5.1.7.0/generated/scale/install-excluding-operator.yaml
```

c. Delete the `MutatingWebhookConfiguration` and `ValidatingWebhookConfiguration`. These will be created in later steps.

```
oc delete MutatingWebhookConfiguration ibm-spectrum-scale-mutating-webhook-configuration
oc delete ValidatingWebhookConfiguration ibm-spectrum-scale-validating-webhook-
configuration
```

Note: Do not change any other custom resource settings during this time, besides the SELinux options noted in the next step.

d. Adjust SELinux context for any special considerations.

Use the `fs.spec.seLinuxOptions` field of the filesystem resource to set the SELinux context for any filesystems that were determined to have special data sharing considerations.

This may be accomplished in a number of different ways depending on a user's environment. Some users may choose to source control resources and use `oc apply`, while others may choose to directly use `oc edit`.

6. Apply the new manifests (including operator).

```
oc apply -f https://raw.githubusercontent.com/IBM/ibm-spectrum-scale-container-native/
v5.1.7.0/generated/scale/install.yaml
```

Verification

When the new IBM Spectrum Scale container native operator is deployed, the upgrade process begins. It takes some time to complete as the new code is rolled out into the cluster.

You can check the progress of pod restarts and node reboots by looking at information provided in the Daemon CR under `.status.statusDetails`. Query the Daemon CR using the following command:

```
oc describe daemon -n ibm-spectrum-scale
```

Code version updated

The version details will be listed under `.status.versions` in the Daemon CR and will be updated as the pods roll. The following command will show the versions that core pods currently have on them. Wait until all the pods are reporting the same new version.

```
oc get daemon -n ibm-spectrum-scale -ojson  | jq -r .items[].status.versions
```

Post upgrade tasks

It describes actions that should be performed on the cluster to complete the upgrade of IBM Spectrum Scale container native to the new code levels.

Approve the new release level

It is recommended to first use the cluster with the new code of IBM Spectrum Scale installed, until you are sure to permanently upgrade the cluster to the new level. When you are ready to enable the new functionality of the installed release and lock in the new level, you need to approve an `UpgradeApproval` resource. An `UpgradeApproval` resource is automatically created by the operator if a release level change is detected after the upgrade.

For more information, see File system format changes between versions of IBM Spectrum Scale in IBM Spectrum Scale documentation.

Complete the following steps:

1. Check to see if any cluster upgrade approvals are present that require action.

   Note: An upgrade approval that shows nothing under the `COMPLETED` field are ones that require some action.

   ```
   oc get upgradeapproval -n ibm-spectrum-scale
   ```

   Note: If an upgrade approval does not appear, check the Daemon CR status to ensure that all pods are on the new version using **`oc get daemon -n ibm-spectrum-scale -ojson | jq -r .items[].status.versions`**
2. Check the `minReleaseLevel` of the cluster:

   ```
   oc exec $(oc get pods -lapp.kubernetes.io/name=core -ojsonpath="{.items[0].metadata.name}" \
      -n ibm-spectrum-scale) -c gpfs -n ibm-spectrum-scale -- mmlsconfig release
   ```
3. To approve the upgrade approval job, execute the following command:

   ```
   oc patch upgradeapproval <upgradeapproval-name> -n ibm-spectrum-scale --type='json' \
      -p='[{"op": "replace", "path": "/spec/approved", "value":true}]'
   ```

   Full Example:

   ```
   # Check for any upgrade approvals for TYPE=cluster
   $ oc get upgradeapproval -n ibm-spectrum-scale
   NAME            TYPE      FILESYSTEM   LAST SCHEDULE TIME   LAST SUCCESSFUL TIME   RUNNING
   COMPLETED
   upgrade-rmlp4   cluster

   # check the daemon status for the versions deployed on each core pod
   $ oc get daemon -n ibm-spectrum-scale -ojson | jq  .items[].status.versions
   [
     {
       "count": "3",
       "version": "5.1.7.0"
     }
   ]

   # check the current cluster release version
   $ oc exec $(oc get pods -lapp.kubernetes.io/name=core -ojsonpath="{.items[0].metadata.name}"
   -n ibm-spectrum-scale) -c gpfs -n ibm-spectrum-scale -- mmlsconfig release
   minReleaseLevel 5.1.5.0

   # patch the upgrade approval
   $ oc patch upgradeapproval upgrade-rmlp4  -n ibm-spectrum-scale \
   > --type='json'  -p='[{"op": "replace", "path": "/spec/approved", "value":true}]'
   upgradeapproval.scale.spectrum.ibm.com/upgrade-rmlp4 patched

   # query the upgrade approval to see it running
   $ oc get upgradeapproval -n ibm-spectrum-scale
   NAME            TYPE      FILESYSTEM   LAST SCHEDULE TIME   LAST SUCCESSFUL TIME
   RUNNING                                                 COMPLETED
   upgrade-rmlp4   cluster                23s                                        ibm-
   spectrum-scale/worker2/gpfs/upgradeCluster_p6Jhz9

   # upgrade approval job completed
   $ oc get upgradeapproval -n ibm-spectrum-scale
   NAME            TYPE      FILESYSTEM   LAST SCHEDULE TIME   LAST SUCCESSFUL TIME   RUNNING
   COMPLETED
   upgrade-rmlp4   cluster                38s                  6s
   Successful

   # verify that the cluster release level has been updated
   $ oc exec $(oc get pods -lapp.kubernetes.io/name=core -ojsonpath="{.items[0].metadata.name}"
   ```

```
-n ibm-spectrum-scale) -c gpfs -n ibm-spectrum-scale -- mmlsconfig release
minReleaseLevel 5.1.7.0
```

For more information, see Completing the upgrade to a new level of IBM Spectrum Scale in IBM Spectrum Scale documentation.

Remote storage cluster considerations

The storage cluster is supported to be down-level from the IBM Spectrum Scale container native cluster, but it is strongly recommended that the versions match. CSI functionality is highly dependent upon the IBM Spectrum Scale release, filesystem level, and version, installed on the storage cluster. If the storage cluster is running an earlier version, some functionality may not be available. For more information about CSI features and required levels, see *Table 1 in Hardware and Software Requirements* in IBM Spectrum Scale CSI documentation. For more information about compatibility and software matrix, see Section 17.3 in IBM Spectrum Scale FAQ documentation.

- Run the following step for each filesystem to upgrade to the latest metadata format:

  ⚠ **Warning:** If the storage cluster is being mounted by other GPFS client clusters that are running lower version of code, performing this step makes those client cluster unable to mount filesystems from this storage cluster.

  ```
  mmchfs <Filesystem> -V full
  ```

  Note: This step is optional but recommended for enabling the functionality provided at the latest levels of code.

- Enable `auto-inode-limit` of the file system.

  ```
  mmchfs <Filesystem> --auto-inode-limit
  ```

  Note: The `--auto-inode-limit` option is available only at filesystem format level of 28.00 or later. Enable this option as soon as the filesystem is updated to 28.00 or later.

Chapter 9. Configuring IBM Spectrum Scale Container Storage Interface (CSI) driver

Use the following sections to help with deploying IBM Spectrum Scale CSI with IBM Spectrum Scale container native:

Configuring storage class to use CSI driver

Storage class is used for creating lightweight volumes and fileset based volumes.

Lightweight (directory) based volumes

A storage class example for creating directory (lightweight) based volumes is provided.

Note: Adjust the parameters as per your environment.

```
# cat storageClass_Lightweight.yaml
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
    name: ibm-spectrum-scale-csi-lt
provisioner: spectrumscale.csi.ibm.com
parameters:
    volBackendFs: "fs1"
    volDirBasePath: "pvfileset/lwdir" # relative path from filesystem mount point for creating
lightweight volume
reclaimPolicy: Delete
```

```
oc create -f storageClass_Lightweight.yaml
```

Fileset based volumes

A storage class example for creating fileset based volumes is provided.

Note: Adjust the parameters as per your environment.

```
# cat storageClass_fileset.yaml
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
    name: ibm-spectrum-scale-csi-fileset
provisioner: spectrumscale.csi.ibm.com
parameters:
    volBackendFs: fs1
    clusterId: "17797813605352210071" # cluster ID of storage cluster
reclaimPolicy: Delete
```

A sample fileset based storage class is created by using a primary file system as the `volBackendFs`. It can be used to create other storage classes with the remote cluster ID that is provided. Enter the **`oc get storageclass -oyaml > storageClass_fileset.yaml`** command to create a copy of this storage class. Then configure parameters as desired and create the configured storage class using the command below:

```
oc create -f storageClass_fileset.yaml
```

Note: For more information, see Storage Class in IBM Spectrum Scale CSI documentation.

© Copyright IBM Corp. 2020, 2023

Managed CSI fields

In the CSI Custom Resource (CR) that is created by the CSI Controller, there are some fields that are managed by the controller. If these fields are changed, they are overridden by the controller. If required, you can change any field that is not managed by the controller.

Managed fields

Note: The following fields are populated with default values by the CSI Controller. Any new values are honored, however, any values that are manually removed are repopulated upon the next controller reconcile cycle.

Table 16. Managed fields description

Field	**Default Value(s)**
clusters	Two entries are created by default (local and remote clusters).
clusters.id	Local Cluster ID / Cluster ID of Remote cluster.
clusters.secrets	`ibm-spectrum-scale-gui-csiadmin`
clusters.secureSSLMode	`false`
clusters.primary.primaryFs	The name of the first file system created (only applicable in local. cluster entry).
clusters.restApi.guiHost	`ibm-spectrum-scale.<container-native-namespace>` for local cluster entry and the host specified in the remote cluster CR for the remote cluster entry.
tolerations	`NoSchedule`, `NoExecute` and `CriticalAddonsOnly`
attacherNodeSelector	`scale=true`
provisionerNodeSelector	`scale=true`
pluginNodeSelector	`scale=true`
snapshotterNodeSelector	`scale=true`

Editing the CSI CR

To edit a CSI CR, enter this command and fill the desired field:

```
oc edit csiscaleoperator -n ibm-spectrum-scale-csi
```

Setting primary file set

After the CSI CR is created by the CSI controller a primary file set needs to be set in order to avoid the naming conflict. Once this field is added the CSI driver pods are deleted and recreated one by one.

Enter the `oc edit csiscaleoperator -n ibm-spectrum-scale-csi` command and add the `primaryFset` field:

```
clusters:
  - id: "11171289193543683780"
    secrets: "secret-cnsa"
    secureSslMode: false
    primary:
```

```
    primaryFs: "fs5"
    primaryFset: "cluster1-fset" #<---- example
    remoteCluster: "2303539379337927879"
   restApi:
    - guiHost: "ibm-spectrum-scale-gui.ibm-spectrum-scale"

  - id: "2303539379337927879"
   secrets: "secret-storage"
   restApi:
   - guiHost: "koopa-gui-1.fyre.ibm.com"
```

Chapter 10. Maintenance of a deployed cluster

The maintenance of a deployed IBM Spectrum Scale container native cluster includes certain procedures.

Shutting down a cluster

Before you begin the maintenance procedure, the IBM Spectrum Scale container native cluster must be shut down to avoid any issues.

Note: For more information, see On the nodes running CSI sidecars in IBM Spectrum Scale CSI documentation.

Complete the following steps to shut down a cluster:

1. Stop the running operator pod by setting the `replicas` in the deployment to 0:

   ```
   oc scale deployment ibm-spectrum-scale-controller-manager -n ibm-spectrum-scale-operator --
   replicas=0
   ```

2. Enter the following command to remove a CSI label:

   ```
   oc label node --all scale-
   ```

3. Enter the following command to delete the running core pods:

   ```
   oc delete pods -lapp.kubernetes.io/name=core -n ibm-spectrum-scale
   ```

IBM Spectrum Scale container native cluster and node maintenance

The following section provides information on changing the IBM Spectrum Scale cluster configuration.

Limitations

While a configuration change is in progress, do not perform the following:

- Do not upgrade IBM Spectrum Scale container native.
- Do not attempt to add a node to the cluster.
- Do not upgrade Red Hat OpenShift.

Updating existing cluster configuration

IBM Spectrum Scale cluster configuration is controlled by the Cluster resource.

To edit configuration for an existing cluster:

```
oc edit cluster.scale
```

When a core pod configuration requires an update, the IBM Spectrum Scale operator will cordon, drain, reboot (if necessary), and uncordon the node. This is performed one node at a time. Once the drain is complete, the core pod will be updated with the new configuration.

Note: If the node was previously cordoned before the update, the operator will not uncordon the node.

© Copyright IBM Corp. 2020, 2023

Updating cluster configuration through upgrade

An upgrade of IBM Spectrum Scale container native is also considered an update of the cluster configuration, as this will likely introduce changes to pod specification, for example, update of container images.

For more information about how IBM Spectrum Scale Operator orchestrates an upgrade, see Chapter 8, “Upgrading,” on page 73.

Red Hat OpenShift maintenance

When a Red Hat OpenShift administrator needs to perform maintenance on a node that involves a drain, the IBM Spectrum Scale operator will intercept and handle the updates safely. If the operator is not running, the interception and drain fails.

Limitations

While Red Hat OpenShift configuration or maintenance is in progress, do not perform the following:

- Do not update IBM Spectrum Scale container native configuration.
- Do not upgrade IBM Spectrum Scale container native.
- Do not attempt to add a node to the cluster.

In addition, deadlock is possible waiting for applications to safely evict from nodes undergoing maintenance. This can occur if too many nodes are undergoing maintenance concurrently, and application workload is disrupted if further action is taken. In these cases, the OpenShift administrator should safely evict and reschedule impacted applications.

For more information about troubleshooting cluster maintenance issues, see Identifying applications preventing cluster maintenance

Red Hat OpenShift cluster configuration update

Red Hat OpenShift machine configuration is managed by the Machine Config Operator (MCO). When configuration changes impact node operation, MCO cordon and drain nodes to perform the maintenance action. Existing pods on the node will be evicted and rescheduled to another available node. The IBM Spectrum Scale operator intercepts requests from the Kubernetes scheduler to ensure that the applications running IBM Spectrum Scale storage workloads are removed before the IBM Spectrum Scalecore pod on the corresponding node. This allows the application to gracefully shutdown before storage access on the node is disrupted.

Once the core pod is safely removed, the MCO update continues and reboots the node, if necessary. After the MCO update is complete, the node gets uncordoned and schedulable.

OpenShift node administrator maintenance

When an OpenShift administrator needs to perform manual maintenance on a node, it should be cordoned and safely drained of pods. The cordon will taint the node as unschedulable and safely evicts applications. This allows the application to reschedule to other available nodes, and notifies the IBM Spectrum Scale operator about the desired node maintenance.

1. Drain the node requiring maintenance. The drain command also cordons the node.

   ```
   oc adm drain <node name>
   ```

2. Once drain completes without error, the node can then have maintenance performed, for example, powering it down.
3. When maintenance is complete, uncordon the node to allow it to resume normal operation and scheduling.

```
oc adm uncordon <node name>
```

Starting the cluster after shutdown

If the IBM Spectrum Scale cluster was shut down, start the cluster by using the following steps:

Note: Ensure that the worker nodes are in the Ready state before restarting the IBM Spectrum Scale cluster by entering the `oc get nodes` command. If any of the worker nodes are in a state other than Ready, the IBM Spectrum Scale cluster fails to restore.

Scale up the operator pods by setting the `replicas` in the deployment to 1.

```
oc scale deployment ibm-spectrum-scale-controller-manager -n ibm-spectrum-scale-operator --
replicas=1
```

After the operator pod comes back up, the core pods are rescheduled and the default CSI label is re-applied.

Adding a new node to an existing cluster

To add a new node:

- Ensure the new node belongs to the existing Machine Config Pool configured to include the kernel-devel extensions. This was configured as a prerequisite to IBM Spectrum Scale container native installation, and defaults to Red Hat OpenShift `worker` nodes. For more information, see Red Hat OpenShift Container Platform configuration.
- Ensure the new node has a node selector that matches the existing cluster's node selector. For more information, see "Node selectors" on page 51.

When the node selector is recognized by the IBM Spectrum Scale Operator, a pod is created on the new node and it goes to running state within a few minutes.

Check the progress of the creation of the new pod by entering the following command:

```
oc get pods -n ibm-spectrum-scale
```

Ensure that the new pod is ready by entering the following command:

```
oc exec <scale-pod> -n ibm-spectrum-scale -- mmgetstate -a
```

The output appears as shown:

```
 Node number  Node name        GPFS state
-------------------------------------------
       1      worker1          active
       2      new node         arbitrating
       3      worker0          active
```

Once the pod has finished arbitrating and enters the active state, CSI will automatically be configured for use by the pod by the IBM Spectrum Scale and CSI Operators. Once CSI has completed configuration, then the newly added node can be used for running applications.

Note: When adding nodes, it makes sense to select additional quorum nodes. For more information, see "Labels and annotations" on page 45.

Updating remote mount access key on IBM Spectrum Scale storage cluster

An access key is used in the IBM Spectrum Scale storage cluster to grant filesystem access to the IBM Spectrum Scale container native cluster. The access key can be changed by following these steps:

1. Identify the `RemoteCluster` resource which you intend to change the access key.

   ```
   oc get remotecluster
   ```

   The `HOST` column identifies the IBM Spectrum Scale storage cluster GUI endpoint. The `NAME` of the resource is used in the later steps, identified by `<remote-cluster>`.
2. On the IBM Spectrum Scale storage cluster, generate a new access key by entering the following command:

   ```
   mmauth genkey new
   ```

   Now, the storage cluster temporarily has two access keys.
3. Begin the update of the new generated key from the IBM Spectrum Scale container native cluster by labeling the `RemoteCluster` resource:

   ```
   oc label remotecluster <remote-cluster> scale.spectrum.ibm.com/update= -n ibm-spectrum-scale
   ```

4. Check the success by watching status and events of the remote cluster resource:

   ```
   oc describe remotecluster <remote-cluster> -n ibm-spectrum-scale
   ```

5. Ensure that all your client clusters that are remote mounting from this storage cluster have applied the new key. After that, commit the new access key by using the following command:

   ```
   mmauth genkey commit
   ```

   At this point, the IBM Spectrum Scale storage cluster has only the new key, the old key is removed.

   For more information, see mmauth command.

Chapter 11. Cleanup

The following procedures outline the steps required to cleanup various pieces of the IBM Spectrum Scale container native solution. To completely uninstall the product, run through each of the following sections.

Removing applications

If there are applications that are accessing the storage filesystem through IBM Spectrum Scale CSI, they must be stopped before continuing.

Perform the following steps to remove all applications:

1. Stop all the applications that are accessing storage via the IBM Spectrum Scale CSI driver.
2. Delete all the Persistent Volume Claims (PVCs) and Persistent Volumes (PVs) provisioned by IBM Spectrum Scale CSI Driver.

Cleanup Kubernetes Resource

The following sections will help describe how to clean up kubernetes custom resources that may be defined in IBM Spectrum Scale container native cluster.

Filesystems

Deleting a `Filesystem` custom resource does not result in the operator un-mounting or deleting the remote mount file system configuration on the IBM Spectrum Scale container native cluster.

Before removing the configuration of the remote mounted file system, ensure that there are no applications actively writing to the file system.

In this example, the `Filesystem` to be removed is named as `remote-sample`:

```
kind: Filesystem
metadata:
  ...
  name: remote-sample
spec:
  remote:
    cluster: remotecluster-sample
    fs: fs1
```

Complete the following steps:

1. Enter the following command to delete the file system from OpenShift.

   ```
   oc delete filesystem remote-sample -n ibm-spectrum-scale
   ```

2. Log in to a core pod by using the following command to remove the file system from IBM Spectrum Scale.

   ```
   oc rsh -n ibm-spectrum-scale worker0
   ```

   - Unmount the file system on all the container native pods.

     ```
     mmunmount remote-sample -a
     ```

   - Delete the remote file system.

     ```
     mmremotefs delete remote-sample
     ```

© Copyright IBM Corp. 2020, 2023

3. If the remote storage cluster is only configured to mount and serve the single `remote-sample` file system, you can delete the remote cluster definition. Otherwise, the other file system(s) must be deleted by using the same process mentioned in the above step.

   - Find the remote clusters.

     ```
     mmremotecluster show all
     ```

   - Delete the remote cluster that is serving the remote file system. For example, to delete a remote cluster named `gpfs.storage`.

     ```
     mmremotecluster delete gpfs.storage
     ```

Remote Clusters

Deleting a `RemoteCluster` custom resource does not result in the operator deleting the access permission of an IBM Spectrum Scale container native cluster to the file systems on a remote storage cluster. The `RemoteCluster` controller only handles creating the access permissions.

Before removing the remote cluster credentials, ensure that no file systems are using this credential. For more information on removing file system resources, see "Filesystems" on page 87.

For this example, the sample `RemoteCluster` is used:

```
kind: RemoteCluster
metadata:
   name: remotecluster-sample
spec:
  ...
```

Perform the following steps:

1. Delete the `RemoteCluster` definition from OpenShift by entering the following command:

   ```
   oc delete remotecluster remotecluster-sample -n ibm-spectrum-scale
   ```

2. Delete the secure credentials on the storage cluster. For more information, see "Cleanup storage cluster" on page 90.

Cluster

Perform the following steps to delete the cluster:

1. Record the GPFS cluster name before deleting the cluster.

   ```
   oc get daemon -n ibm-spectrum-scale -o json | jq -r '.items[].status.clusterName'
   ```

   Note: This GPFS cluster name is required in subsequent steps when removing the authorization on the storage cluster.

2. Enter the following command to delete the IBM Spectrum Scale cluster.

   ```
   oc delete cluster.scale ibm-spectrum-scale
   ```

Cleanup IBM Spectrum Scale container native

Complete the following steps:

1. Change to a default namespace:

   ```
   oc project default
   ```

2. Enter the following command to uninstall the operator, kubernetes objects, namespaces, and more.

```
oc delete -f https://raw.githubusercontent.com/IBM/ibm-spectrum-scale-container-native/
v5.1.7.0/generated/scale/install.yaml --ignore-not-found=true
```

3. Enter the following command to clean up the performance monitoring and IBM Spectrum Scale CSI artifacts.

   a) Enter the following command to list the PVs with claim of `datadir-ibm-spectrum-scale-scale-pmcollector`. Two PVs are returned.

   ```
   oc get pv -lapp.kubernetes.io/instance=ibm-spectrum-scale,app.kubernetes.io/
   name=pmcollector
   oc delete pv  -lapp.kubernetes.io/instance=ibm-spectrum-scale,app.kubernetes.io/
   name=pmcollector
   ```

   b) Enter the following command to delete the Storage Classes created by performance monitoring and IBM Spectrum Scale CSI artifacts:

```
oc delete sc  -lapp.kubernetes.io/instance=ibm-spectrum-scale,app.kubernetes.io/
name=pmcollector
oc delete sc ibm-spectrum-scale-sample
```

Cleanup OpenShift nodes

IBM Spectrum Scale container native requires host path volume mounts and will create persistent directories and files on each of the OpenShift nodes running a core pod. When uninstalling, or before reinstalling, removal of these persistent files are required.

Complete the following steps to remove the persistent files on the OpenShift nodes:

1. Switch to the default project.

   ```
   oc project default
   ```

2. Enter the following command to list the nodes.

   ```
   oc get nodes -o jsonpath="{range .items[*]}{.metadata.name}{'\n'}"
   ```

   Note: If you know the nodeSelector that was used, use the label selector to reduce the set of nodes to cleanup on, or just run cleanup against all the nodes.

3. For each of nodes from the output above, enter the following command to create a debug pod that removes the kernel modules and the host path volume mounted directories used by IBM Spectrum Scale container native:

   ```
   oc debug node/<openshift_node> -T -- chroot /host sh -c "rm -rf /var/mmfs; rm -rf /var/adm/
   ras; rmmod tracedev mmfs26 mmfslinux;"
   ```

   Example:

   ```
   oc debug node/worker0.example.com -T -- chroot /host sh -c "rm -rf /var/mmfs; rm
   -rf /var/adm/ras; rmmod tracedev mmfs26 mmfslinux;"
      Starting pod/worker0examplecom-debug ...
      To use host binaries, run `chroot /host`
      Removing debug pod ...
   ```

4. Ensure that none of the artifacts are left by entering the following validation command:

   ```
   oc debug node/<openshift_node> -T -- chroot /host sh -c "ls /var/mmfs; ls /var/adm/ras;
   rmmod tracedev mmfs26 mmfslinux;"
   ```

   Example:

   ```
   oc debug node/worker0.example.com -T -- chroot /host sh -c "ls /var/mmfs; ls /var/adm/ras;
   rmmod tracedev mmfs26 mmfslinux;"
      Starting pod/worker0examplecom-debug ...
      To use host binaries, run `chroot /host`
      ls: cannot access '/var/mmfs': No such file or directory
   ```

```
   ls: cannot access '/var/adm/ras': No such file or directory
   rmmod: ERROR: Module tracedev is not currently loaded
   rmmod: ERROR: Module mmfs26 is not currently loaded
   rmmod: ERROR: Module mmfslinux is not currently loaded
   Removing debug pod ...
   error: non-zero exit code from debug container
```

5. Remove any node labels that are created by the IBM Spectrum Scale container native operator:

```
oc label node --all scale.spectrum.ibm.com/role-
   oc label node --all scale.spectrum.ibm.com/designation-
   oc label node --all scale-
```

Cleanup storage cluster

Note: If your storage cluster is on AWS ROSA, skip this section and see "Revoke filesystem access from the storage cluster" on page 90.

Delete the access permission that is granted to the IBM Spectrum Scale client cluster for mounting the remote file system by running the following steps on the storage cluster:

1. Enter the following command to query the name of the GPFS clusters that have established authorization to this storage cluster.

```
mmauth show all
```

2. If the name of your GPFS cluster is `ibm-spectrum-scale.clustername.example.com`, run the following command to remove the client cluster authorization.

```
mmauth delete ibm-spectrum-scale.clustername.example.com
```

Cleanup on AWS ROSA

This section provides cleanup procedures that needs to be done on an IBM Spectrum Scale container native deployment on Red Hat OpenShift Service on AWS (ROSA).

Cleaning up the ROSA worker security group

Complete the following steps to clean up ROSA worker security group:

1. In AWS management console, navigate to the **EC2 Dashboard** in the region where ROSA is installed and select the **Security Groups** option from the navigation pane. Locate the Security Group for ROSA worker security group ID and export the following variable.

```
export AWS_ROSA_WORKER_SECURITY_GROUP=<security_group_id>
```

2. Use the following commands to revoke the IBM Spectrum Scale container native ingress traffic rules.

```
aws ec2 revoke-security-group-ingress --group-id ${AWS_ROSA_WORKER_SECURITY_GROUP} --
protocol tcp --port 12345  --source-group ${AWS_ROSA_WORKER_SECURITY_GROUP}
aws ec2 revoke-security-group-ingress --group-id ${AWS_ROSA_WORKER_SECURITY_GROUP} --
protocol tcp --port 1191  --source-group ${AWS_ROSA_WORKER_SECURITY_GROUP}
aws ec2 revoke-security-group-ingress --group-id ${AWS_ROSA_WORKER_SECURITY_GROUP} --
protocol tcp --port 60000-61000  --source-group ${AWS_ROSA_WORKER_SECURITY_GROUP}
```

Revoke filesystem access from the storage cluster

Note: Before revoking access from the storage cluster, ensure that you have fully cleaned up the IBM Spectrum Scale container native cluster and no longer have any applications utilizing the filesystems.

After IBM Spectrum Scale container native has been uninstalled, use the below command to revoke the filesystem access:

```
# ./cloudkit revoke filesystem
I: Logging at /root/scale-cloudkit/logs/cloudkit-16-2-2023_0-16-37.log
? Cloud platform name:  AWS
==========================================================================
|                              ! Note !                                  |
=========================================================================|
|  Revoke cluster involves unmount of filesystem. Make sure the I/O is   |
|  stopped and data is flushed before proceeding further.                |
==========================================================================
? Select remotemount name:  scale-strg-cls-rosa-scale
? IBM Spectrum Scale Container Native cluster name:  ibm-spectrum-scale.stg.rosa-
scale.example.com
? Storage cluster management GUI username:  administrator
? Storage cluster management GUI password:  ********
? Connectivity method to cloud:  JumpHost
? Bastion/JumpHost instance login username:  ec2-user
? Bastion/JumpHost instance public ip address:  xxx.xxx.xxx.195
? Bastion/JumpHost SSH private key file path (will be used only for configuration):  /root/
bastion_pvt_key
I: Obtaining spectrum scale storage cluster definition.
I: Initiating remote mount revoke configuration.
 100% |
██████████████████████████████████████████████████████████████████████████████████████████████|
(10/10, 1 it/min)
I: Spectrum Scale cluster configuration completed.
I: Updating storage security group '<sg-storage-hash>' to revoke traffic from OCP security
group '<sg-ocp-group-hash>'.
I: Access to Spectrum Scale cluster 'scale-strg-cls' has been revoked.
```

Chapter 12. Monitoring

The IBM Spectrum Scale container native cluster is monitored by sending the health status and events between its pods.

System monitor and Kubernetes readiness probe

The scale-monitor sidecar container has the following objectives:

- Runs the containermon service which is monitoring the service (GUI, pmcollector) in the same pod.
- Provides a readiness probe API (HTTPS).
- Sends the health status and events back to the core pod on the same worker node.
- Core pod is forwarding the events to GUI or mmhealth.
- Provides an API for call home data collection.
- Has several debug tools installed and can be used for problem determination.

Note:

For more information, see Container probes in Kubernetes documentation.

If the monitoring status is HEALTHY, the probe returns success 200. When the `unreadyOnFailed` option is enabled in containermon.conf (default=true), any FAILED state causes the probe to return 500. When a critical event occurred which has the `container_unready=True` flag, the probe returns 501. When the service faces an issue, for example, no service found, it returns 502.

Viewing and analyzing the performance data with the IBM Spectrum Scale bridge for Grafana

IBM Spectrum Scale has built-in performance monitoring tool that collects metrics from various GPFS components.

These metrics can provide you with a status overview and trends of the key performance indicators. You can view and analyze the collected performance data with Grafana, a third-party visualization software.

For using Grafana, you need a running Grafana instance and the IBM Spectrum Scale performance monitoring bridge for Grafana deployed on your IBM Spectrum Scale container native cluster. For more information, see IBM Spectrum Scale bridge for grafana repository in GitHub.

The IBM Spectrum Scale bridge for Grafana is an open source tool, available for free usage on IBM Spectrum Scale devices. It translates the metadata and performance data collected by the IBM Spectrum Scale performance monitoring tool to query requests acceptable by the Grafana-integrated openTSDB plugin.

The IBM Spectrum Scale performance monitoring bridge for Grafana could be deployed automatically through the operator. For more information, see “Grafana bridge” on page 53.

For more information about setting up a Grafana instance for monitoring the IBM Spectrum Scale container native cluster, see Setup Grafana for monitoring a IBM Spectrum Scale container native cluster in a k8s OCP environment in GitHub documentation.

© Copyright IBM Corp. 2020, 2023

Chapter 13. Support

There may be occasions where you need to troubleshoot or open a support case in IBM for further assistance.

Common issues

Error: daemon and kernel extension do not match

This error occurs when there is an unintentional upgrade of GPFS code.

The issue presents itself as the GPFS state is down and the above error is found in the GPFS logs.

To resolve the issue, follow proper upgrade procedures. The issue occurs because the kernel module cannot be unloaded when a file system is in use. Rebooting the node resolves the problem, or follow procedures to remove application workloads and then enter the following command on the node issue:

```
rmmod tracedev mmfs26 mmfslinux
```

Note: For more information, see "Removing applications" on page 87.

RestError: Failed to get storage cluster information. errmsg: 401 Unauthorized GET

When describing the `RemoteCluster` CR using `oc describe remotecluster.scale` you may see Events that show errors indicating "401 Unauthorized". The IBM Spectrum Scale GUI REST credentials for storage clusters are stored in kubernetes secrets. The 401/Unauthorized error indicates that the credentials provided by the kubernetes secrets do not match the GUI user credentials in the storage cluster. For more information, see Chapter 3, "Storage Cluster," on page 27.

There are different possible root causes:

- A GUI user was never created as described in the procedure for creating operator user and group. For more information, see "Creating Container Operator and CSI Users" on page 27.
- The GUI user password has expired in the storage cluster and must be changed.
- The GUI user password is changed in the storage cluster.
- The GUI user is deleted in the storage cluster.

Complete the following steps to solve this problem:

1. Get the name of the secret by entering `oc describe remotecluster.scale -n ibm-spectrum-scale` command and looking for `Secret Name`:

```
...
Spec:
  Contact Nodes:
    storagecluster1node1
    storagecluster1node2
  Gui:
    Cacert:                 cacert-storage-cluster-1
    Csi Secret Name:        csi-remote-mount-storage-cluster-1
    Host:                   guihost.example.com
    Insecure Skip Verify:   false
    Port:                   443
    Secret Name:            cnsa-remote-mount-storage-cluster-1
...
```

2. Read the credentials from the kubernetes secret for accessing the storage cluster IBM Spectrum Scale GUI REST API.

```
oc get secret cnsa-remote-mount-storage-cluster-1 -n ibm-spectrum-scale
-ojsonpath='{.data.username}' | base64 -d -
```

```
oc get secret cnsa-remote-mount-storage-cluster-1 -n ibm-spectrum-scale
-ojsonpath='{.data.password}' | base64 -d -
```

Note: In some shells, the end of the line has a highlighted %. This denotes there is no new line and should not be included when updating the password.

3. If the password differs from the one that is set for a GUI user in the storage cluster, then delete and re-create the secret as configured during installation.
4. If a GUI user does not exist in a storage cluster, create an IBM Spectrum Scale GUI user in the `ContainerOperator` group by either using the GUI or by issuing the following command in the shell of the GUI node of the storage cluster:

```
/usr/lpp/mmfs/gui/cli/mkuser cnss_storage_gui_user -p cnss_storage_gui_password -g
ContainerOperator
```

MountVolume.SetUp failed for volume "ssh-keys"

```
Warning FailedMount 83m (x5 over 83m) kubelet, worker-0.example.ibm.com  MountVolume.SetUp
failed for volume "ssh-keys" : secret "ibm-spectrum-scale-ssh-key-secret" not found
```

The pod create times show that the ssh key secret was created after the deployment. This means that the deployment rightfully could not find the secret to mount, as it did not yet exist.

This message can be misleading as the pods should resolve themselves once the secret is created. If core pods are not in a `Running` state, and the secret is already created, deleting the ibm-spectrum-scale-core pods should resolve the issue. This restarts the pods and allow the mount to complete successfully for the already created SSH key.

A pmcollector pod is in pending state during the OpenShift Container Platform upgrade or reboot

```
Events:
  Type     Reason            Age                    From               Message
  ----     ------            ----                   ----               -------
  Warning  FailedScheduling  65s (x202 over 4h43m)  default-scheduler  0/6 nodes are available:
1 node(s) were unschedulable, 2 node(s) had volume node affinity conflict, 3 node(s) had taint
{node-role.kubernetes.io/master:}, that the pod didn't tolerate.
```

This issue is caused by a problem during the OpenShift Container Platform Upgrade or when a worker node has not been reset to schedulable after reboot. The pmcollector remains in a `Pending` state until the pod itself and its respective Persistent Volume can be bound to a worker node.

```
# oc get nodes
NAME                   STATUS                       ROLES    AGE     VERSION
master0.example.com    Ready                        master   5d18h   v1.18.3+2fbd7c7
master1.example.com    Ready                        master   5d18h   v1.18.3+2fbd7c7
master2.example.com    Ready                        master   5d18h   v1.18.3+2fbd7c7
worker0.example.com    Ready                        worker   5d18h   v1.17.1+45f8ddb
worker1.example.com    Ready,SchedulingDisabled     worker   5d18h   v1.17.1+45f8ddb
worker2.example.com    Ready                        worker   5d18h   v1.17.1+45f8ddb
```

If the Persistent Volume has `Node  Affinity` to the host that has `SchedulingDisabled`, the pmcollector pod remains in `Pending` state until the node associated with the PV becomes schedulable.

```
# oc describe pv worker1.example.com-pv
Name:              worker1.example.com-pv
Labels:            app=scale-pmcollector
Annotations:       pv.kubernetes.io/bound-by-controller: yes
Finalizers:        [kubernetes.io/pv-protection]
StorageClass:      ibm-spectrum-scale-internal
Status:            Bound
Claim:             example/datadir-ibm-spectrum-scale-pmcollector-1
Reclaim Policy:    Delete
Access Modes:      RWO
VolumeMode:        Filesystem
Capacity:          25Gi
Node Affinity:
  Required Terms:
    Term 0:        kubernetes.io/hostname in [worker1.example.com]
Message:
Source:
    Type:  LocalVolume (a persistent volume backed by local storage on a node)
    Path:  /var/mmfs/pmcollector
```

If the issue was with OpenShift Container Platform Upgrade, fixing the upgrade issue should resolve the pending pod.

If the issue is due to worker node in `SchedulingDisabled` state and not due to a failed OpenShift Container Platform Upgrade, re-enable scheduling for the worker with the `oc adm uncordon` command.

Failed to establish remote cluster connection when cacert ConfigMap does not exist

When describing the remote cluster objects, you may see an error: `Error: ConfigMap "cacert-storage-cluster-1" not found`.

This issue is caused by not configuring TLS verification of CA certificates for the remote storage GUI.

For more information and resolution options, see "Configuring Certificate Authority (CA) certificates" on page 60.

Known issues

http: TLS handshake error... remote error: tls: bad certificate errors found in the operator log

After upgrading IBM Spectrum Scale container native, you may see many messages in the operator log similar to the following:

```
http: TLS handshake error from <IP>:<PORT> remote error: tls: bad certificate
```

Verify that the Webhooks have a `caBundle` injected for each defined `path:`. You can use these commands to check this:

- To check the `MutatingWebhookConfiguration`:

  ```
  oc get MutatingWebhookConfiguration ibm-spectrum-scale-mutating-webhook-configuration -oyaml
  | egrep "caBundle:|path:"
  ```

- To check the `ValidatingWebhookConfiguration`:

  ```
  oc get ValidatingWebhookConfiguration ibm-spectrum-scale-validating-webhook-configuration
  -oyaml | egrep "caBundle:|path:"
  ```

For example, if looking at the `ibm-spectrum-scale-mutating-webhook-configuration`, you may see an output similar to the following:

```

caBundle: <SOME CA BUNDLE DEFINED, LONG KEY>
  path: /mutate-scale-spectrum-ibm-com-v1beta1-cluster
  path: /mutate-scale-spectrum-ibm-com-v1beta1-filesystem
```

The above shows that you have more path endpoints than `caBundle`, meaning that the `caBundle` was not injected for the second path: `/mutate-scale-spectrum-ibm-com-v1beta1-filesystem`. This is caused by a bug in the OpenShift service CA operator injecting the caBundle into a patched webhook.

If this has occurred in your cluster, you should perform the following actions:

1. Delete the `MutatingWebhookConfiguration` and `ValidatingWebhookConfiguration`.

   ```
   oc delete MutatingWebhookConfiguration ibm-spectrum-scale-mutating-webhook-configuration
   oc delete ValidatingWebhookConfiguration ibm-spectrum-scale-validating-webhook-configuration
   ```

2. Re-apply the manifest for IBM Spectrum Scale container native. For example, if you are upgrading to IBM Spectrum Scale container native 5.1.7.0, then run:

   ```
   oc apply -f https://raw.githubusercontent.com/IBM/ibm-spectrum-scale-container-native/
   v5.1.7.0/generated/scale/install.yaml
   ```

3. After applying the manifests, querying the webhook again should result in having a `caBundle` injected to each endpoint path:

   ```
   caBundle: <SOME CA BUNDLE DEFINED, LONG KEY>
    path: /mutate-scale-spectrum-ibm-com-v1beta1-cluster
   caBundle: <SOME CA BUNDLE DEFINED, LONG KEY>
    path: /mutate-scale-spectrum-ibm-com-v1beta1-filesystem
   ```

For more information on specific upgrade procedures, see "Upgrading IBM Spectrum Scale container native" on page 73.

Adding a remote cluster to an existing IBM Spectrum Scale container native cluster taking long time to appear

When adding a RemoteCluster custom resource after initial installation of IBM Spectrum Scale container native, it can take some time for the IBM Spectrum Scale container native operator to propagate this information to the CSI custom resource.

To resolve this, manually trigger a reconcile of the operator by deleting the operator pod and allowing it to be recreated.

```
oc delete pod -nibm-spectrum-scale-operator -lapp.kubernetes.io/name=operator
```

Once the operator reconciles, it updates the CSI custom resource with the new RemoteCluster custom resource.

pmsensors showing null after failure of pmcollector node

If a node that is running the pmcollector pod is drained, when the node is uncordoned, the pmcollector pods get new IPs assigned. This leads to the pmsensors process issue. It displays the following message:

```
Connection to scale-pmcollector-0.scale-pmcollector successfully established.
```

But an error is reported:

```
Error on socket to scale-pmcollector-0.scale-pmcollector: No route to host (113)
```

See `/var/log/zimon/ZIMonSensors.log`. This issue can also be seen on the pmcollector pod:

```
# echo "get metrics cpu_user bucket_size 5 last 10" | /opt/IBM/zimon/zc 0
1:      worker1
2:      worker2
Row Timestamp               cpu_user
1   2020-11-16 05:27:25     null
2   2020-11-16 05:27:30     null
3   2020-11-16 05:27:35     null
4   2020-11-16 05:27:40     null
5   2020-11-16 05:27:45     null
6   2020-11-16 05:27:50     null
7   2020-11-16 05:27:55     null
8   2020-11-16 05:28:00     null
9   2020-11-16 05:28:05     null
10  2020-11-16 05:28:10     null
```

If the scale-pmcollector pods get their IP addresses changed, the pmsensors process needs to be killed and restarted manually on all scale-core pods, to get the performance metrics collection resumed.

To kill the pmsensor process, run these commands on all the ibm-spectrum-scale-core pods. The `PMSENSORPID` variable holds the results of the `oc exec` command. If this variable is empty, there is no process running, and you do not need to enter the following command to kill the process.

```
PMSENSORPID=`oc exec <ibm-spectrum-scale-core>  -n ibm-spectrum-scale -- pgrep
-fx '/opt/IBM/zimon/sbin/pmsensors -C /etc/scale-pmsensors-configuration/ZIMonSensors.cfg
-R /var/run/perfmon'`
echo $PMSENSORPID
oc exec <scale-pod> -n ibm-spectrum-scale -- kill $PMSENSORPID
```

To start the service again, enter this command on all the scale pods.

```
oc exec <scale-pod> -n ibm-spectrum-scale -- /opt/IBM/zimon/sbin/pmsensors -C /etc/scale-
pmsensors-configuration/ZIMonSensors.cfg -R /var/run/perfmon
```

Remote file systems are defined but not mounted on all nodes

If the RemoteMount controller shows that a target storage cluster file system is established, but the remote file system is not mounted on all the nodes in the ibm-spectrum-scale-core pods, execute the following command to mount the file system manually from one of the scale-core pods:

```
# Replace FILESYSTEM with the name of your filesystem
FILESYSTEM="fs1"
oc exec $(oc get pods -lapp=ibm-spectrum-scale-core -ojsonpath="{.items[0].metadata.name}") --
mmmount $FILESYSTEM -a
```

Remote file systems unable to mount successfully

On the Filesystem CR, if you see events that indicate the filesystem is unable to mount, check in the pod to see if running **`mmlsfs <filesystem>`** results in 'Operation not permitted' error message.

Starting with IBM Spectrum Scale 5.1.3.0 and IBM Spectrum Scale container native 5.1.3.0, the **`tscCmdAllowRemoteConnections`** configuration is recommended to be set to no. If a storage cluster and all client clusters (including IBM Spectrum Scale container native) are at versions >= 5.1.3.0, it is recommended to set this value to no. However, if any version is < 5.1.3.0, **`tscCmdAllowRemoteConnections`** needs to be set to `yes` on the storage cluster and client clusters to successfully communicate between the clusters.

Use the following table as a reference.

Table 17. Storage cluster and IBM Spectrum Scale container native versions

Storage Cluster version	IBM Spectrum Scale container native	tscCmdAllowRemoteConnections
< 5.1.3.0	< 5.1.3.0	yes
>= 5.1.3.0	< 5.1.3.0	yes
>= 5.1.3.0	>= 5.1.3.0	no

To change this value on a storage cluster, run **`mmchconfig tscCmdAllowRemoteConnections:yes|no`**.

To change this value on an IBM Spectrum Scalecontainer native cluster, set the **`tscCmdAllowRemoteConnections:yes|no`** in the **`clusterProfile`** section of the cluster spec by entering the following command:

```
  kind: Cluster
  metadata:
  name: ibm-spectrum-scale
  spec:
    ...
    daemon:
      ...
    ...
    clusterProfile:
      tscCmdAllowRemoteConnections: "yes"
```

For more information to configure the **`clusterProfile`** section of the cluster spec, see "Cluster profile" on page 51.

pid_limits set higher than podPidLimits, but not being honored

With OpenShift Container Platform 4.11, some CRI-O fields introduced before kubelet supported those values are being deprecated in favor of using the fields defined in kubelet. One of those deprecated fields is `pids_limit` set by the `ContainerRuntimeConfig` CR. For more information, see CRI-O should deprecate log size max and pids limit options in RedHat JIRA dashboard.

If you had applied a custom MCO configuration with a `pids_limit` value higher than 4096, the container limits is restricted by the default `podPidsLimit` value in `kubelet.conf`. This default is set to 4096 on OCP 4.11. To increase this value, perform the following:

Note:

It is highly recommended that you are at IBM Spectrum Scale container native 5.1.7.0 or higher before making changes to `MachineConfig` as the IBM Spectrum Scale container native operator will orchestrate the updates to `MachineConfig` as an attempt to keep the IBM Spectrum Scale cluster operational.

1. Define the `podPidsLimit` in the `KubeletConfig` custom resource.

   ```
   yaml
       ---
       apiVersion: machineconfiguration.openshift.io/v1
       kind: KubeletConfig
       metadata:
         name: 01-worker-ibm-spectrum-scale-increase-pid-limit
       spec:
         machineConfigPoolSelector:
           matchLabels:
             pools.operator.machineconfiguration.openshift.io/worker: ''
         kubeletConfig:
           podPidsLimit: 8192
   ```

2. Delete the IBM Spectrum Scale container native `ContainerRuntimeConfig` CR to set the default back to 0 (unlimited):

   ```
    oc delete ContainerRuntimeConfig 01-worker-ibm-spectrum-scale-increase-pid-limit
   ```

Adding a node fails with "The node appears to already belong to a GPFS cluster"

When adding a worker node into OpenShift, and using the nodeSelector of `node-role.kubernetes.io/worker` in the Cluster CR, the IBM Spectrum Scale container native operator will deploy a core pod to the newly added node and attempt to add this node into the GPFS cluster. There may be a situation where the core pod will be in "Init:1/2" state with no sign of recovery.

The operator log will contain entries matching ERROR `Failed to add` node and mmaddnode failing with the reason "The node appears to already belong to a GPFS cluster".

To recover from this scenario, perform the following steps:

1. Create a debug pod to the node where the pod is failing to start and delete the GPFS metadata.

   ```
   oc debug node/<openshift_worker_node> -T -- chroot /host sh -c "rm -rf /var/mmfs; rm
   -rf /var/adm/ras"
   ```

   Example:

   ```
   oc debug node/worker0.example.com -T -- chroot /host sh -c "rm -rf /var/mmfs; rm
   -rf /var/adm/ras"
   Starting pod/worker0examplecom-debug ...
   ```

```
To use host binaries, run `chroot /host`
Removing debug pod ...
```

2. Delete the core pod. If the core pod is called `worker3`, run the following command:

   ```
   oc delete pod worker3 -n ibm-spectrum-scale
   ```

3. The operator should reconcile and attempt to create the pod again and succeed.

GUI or Grafana bridge pods fails to start, no data returned from pmcollector to frontend applications

There exists an issue where no data is returned to frontend applications that are actively consuming performance metrics from IBM Spectrum Scale pmcollector. This also has a signature of Grafana Bridge pod failing to start. If this is experienced, apply the following workaround.

1. Check NodeNetworkConfigurationPolicy's to determine which network interfaces are configured for a node network.

   - List the NodeNetworkConfigurationPolicies

     ```
     oc get nnce
     ```

     Example:

     ```
     # oc get nnce
     NAME                                                         STATUS
     compute-0.mycluster.example.com.bond1-ru5-policy   SuccessfullyConfigured
     compute-1.mycluster.example.com.bond1-ru6-policy   SuccessfullyConfigured
     compute-2.mycluster.example.com.bond1-ru7-policy   SuccessfullyConfigured
     control-0.mycluster.example.com.bond1-ru2-policy   SuccessfullyConfigured
     control-1.mycluster.example.com.bond1-ru3-policy   SuccessfullyConfigured
     control-2.mycluster.example.com.bond1-ru4-policy   SuccessfullyConfigured
     ```

   - Describe the NodeNetworkConfigurationPolicy to identify the network interface being used.

     Example:

     ```
     # oc describe nnce compute-0.mycluster.example.com.bond1-ru5-policy | grep Name
     Name:         compute-0.mycluster.example.com.bond1-ru5-policy
     Namespace:
       Name:           bond1-ru5-policy
         Name:         bond1
         Name:         bond1.3201
     ```

     Note: In this particular example, the bond interfaces are configured for the node network traffic.

2. Change the Performance Data Collection rules to limit the discovery of the Network adapters to only the configured interfaces.

   - Stop the sensors activities on all Core nodes

     ```
     oc get pods -lapp.kubernetes.io/name=core -n ibm-spectrum-scale \
     -ojsonpath="{range .items[*]}{.metadata.name}{'\n'}" | \
     xargs -I{} oc exec {} -n ibm-spectrum-scale -c gpfs -- \
     kill $(pgrep -fx '/opt/IBM/zimon/sbin/pmsensors -C /etc/scale-pmsensors-configuration/
     ZIMonSensors.cfg -R /var/run/perfmon')
     ```

   - Review the current filter settings for the Network sensor in the Performance Data Collection rules. These are stored in the `ibm-spectrum-scale-pmsensors-config` ConfigMap.

     ```
     oc describe cm ibm-spectrum-scale-pmsensors-config -n ibm-spectrum-scale | grep filter |
     grep netdev
     ```

     Example output:

     ```
     # oc describe cm ibm-spectrum-scale-pmsensors-config -n ibm-spectrum-scale | grep filter |
     grep netdev
     filter = "netdev_name=veth.*|docker.*|flannel.*|cali.*|cbr.*"
     ```

    Note: The above filter is used for exclusion logic.

- Edit the `ibm-spectrum-scale-pmsensors-config` ConfigMap and replace the substring `netdev_name=veth.|docker.|flannel.|cali.|cbr.` with `netdev_name=^((?!bond).)`

  ```
  oc edit ibm-spectrum-scale-pmsensors-config -n ibm-spectrum-scale
  ```

  Note: Bond interface is being used in this example, replace *bond* with the respective adapter name used by the customer's network interface.

- Verify that the `ibm-spectrum-scale-pmsensors-config` ConfigMap now reflects the desired adapter.

  ```
  oc describe cm ibm-spectrum-scale-pmsensors-config -n ibm-spectrum-scale|grep filter |
  grep netdev
  ```

3. Cleanup the metadata keys in the pmcollector database not related to the configured node network interfaces. Remote shell into each pmcollector pod and issue the following commands.

   ```
   oc rsh -cpmcollector ibm-spectrum-scale-pmcollector-0
   ```

   ```
   echo "delete key .*|Network|[a-f0-9]{15}|.*" | /opt/IBM/zimon/zc 0
   ```

   ```
   echo "topo -c -d 6" | /opt/IBM/zimon/zc 0| grep Network | cut -d'|' -f2-3 | sort | uniq -c |
   sort -n | tail -50
   ```

   Then exit the container.

   Example:

   ```
   # oc rsh -cpmcollector ibm-spectrum-scale-pmcollector-0
   sh-4.4$ echo "delete key .*|Network|[a-f0-9]{15}|.*" | /opt/IBM/zimon/zc 0
   sh-4.4$ echo "topo -c -d 6" | /opt/IBM/zimon/zc 0| grep Network | cut -d'|' -f2-3 | sort |
   uniq -c | sort -n | tail -50
       96 Network|bond0
       96 Network|bond1
       96 Network|bond1.3201
       96 Network|lo
   sh-4.4$ exit

   # oc rsh -cpmcollector ibm-spectrum-scale-pmcollector-1
   sh-4.4$ echo "delete key .*|Network|[a-f0-9]{15}|.*" | /opt/IBM/zimon/zc 0
   sh-4.4$ echo "topo -c -d 6" | /opt/IBM/zimon/zc 0| grep Network | cut -d'|' -f2-3 | sort |
   uniq -c | sort -n | tail -50
       96 Network|bond0
       96 Network|bond1
       96 Network|bond1.3201
       96 Network|lo
   sh-4.4$ exit
   ```

4. Start the sensors jobs on all Core nodes.

   ```
   oc get pods -lapp.kubernetes.io/name=core -n ibm-spectrum-scale \
   -ojsonpath="{range .items[*]}{.metadata.name}{'\n'}" | \
   xargs -I{} oc exec {} -n ibm-spectrum-scale -c gpfs -- \
   /opt/IBM/zimon/sbin/pmsensors -C /etc/scale-pmsensors-configuration/ZIMonSensors.cfg
   -R /var/run/perfmon
   ```

5. Delete the pmcollector and grafana bridge pods to update the configuration changes.

   ```
   oc delete pod -lapp.kubernetes.io/instance=ibm-spectrum-scale,app.kubernetes.io/
   name=pmcollector
   oc delete pod -lapp.kubernetes.io/instance=ibm-spectrum-scale,app.kubernetes.io/
   name=grafanabridge
   ```

   After some time, the pmcollector and grafana bridge pods are redeployed by the ibm-spectrum-scale-operator.

Gathering data about your cluster

When opening an issue in IBM Support, you may be asked to provide information that will help assist in performing root cause analysis.

must-gather

The `oc adm must-gather` CLI command can be used to collect information about your cluster that is most likely needed to debug issues.

Prerequisites

- Access to the cluster as a user with the `cluster-admin` role
- OpenShift Container Platform CLI (OC) command should be installed

Gather data about a Red Hat OpenShift Container Platform cluster

For issues with a Red Hat OpenShift Container Platform cluster where a ticket must be opened with Red Hat Support, provide the debugging information about the cluster for problem determination. For more information, see Gathering data about your cluster in Red Hat OpenShift documentation.

Note: Executing a default `must-gather` for OpenShift Container Platform debug does not collect information for IBM Spectrum Scale container native.

Gather data about the IBM Spectrum Scale container native cluster

To gather data specifically for the IBM Spectrum Scale container native cluster to provide to IBM Support, point the `must-gather` tool to the `ibm-spectrum-scale-must-gather` image. This will collect Kubernetes resources associated with the IBM Spectrum Scale container native cluster as well as retrieving a GPFS snap.

Note: `oc adm must-gather --image` requires the image stored in a repository where it can be anonymously pulled (no credentials required). In an airgapped environment, the `ibm-spectrum-scale-must-gather` image must be pulled from the IBM Cloud Container Registry and then uploaded to an internal image registry allowing anonymous pull. For more information about air gap instructions, see Air gap setup for network restricted Red Hat OpenShift Container Platform clusters.

1. In the directory where the must-gather contents need to be stored, enter the must-gather command by using the `ibm-spectrum-scale-must-gather` image:

   ```
   oc adm must-gather --image=icr.io/cpopen/ibm-spectrum-scale-must-gather:v5.1.7.0
   ```

2. Once completed, a new directory with `must-gather` prefix is created in your working directory.

   For example:

   ```
   # ls -ltr
   drwxr-xr-x  3 root root   229 Jun 14 09:11 must-gather.local.681612165636007567
   ```

3. Create a compressed file from the must-gather directory that was just created in your working directory.

   ```
   tar cvaf must-gather.tar.gz must-gather.local.681612165636007567/
   ```

   Note: Replace the directory name used in this command with your respective must-gather directory.

Generating GPFS trace reports

Some issues might require low-level system detail accessible only through the IBM Spectrum Scale daemon and the IBM Spectrum Scale Linux kernel trace facilities.

In such instances the IBM Support Center might request such GPFS trace reports to facilitate rapid problem determination of failures.

The level of detail that is gathered by the trace facility is controlled by setting the trace levels using the mmtracectl command. For more information, see mmtracectl command in IBM Spectrum Scale documentation.

Note: The following steps must be performed under the direction of the IBM Support Center.

1. Enter the following command to access a running ibm-spectrum-scale-core pod:

   ```
   oc rsh -n ibm-spectrum-scale <ibm-spectrum-scale-core-pod>
   ```

   Note: The pod must be in `Running` status to connect. It is best to pick a pod running on a node that is not exhibiting issues.

   The remaining steps should be completed while connected to this shell running inside the `gpfs` container of this running core pod.
2. Enter the `mmchconfig` command to change the `dataStructureDump` field to point to `/var/adm/ras`. This changes the default location where trace data is stored to a directory that persists on the host machine:

   ```
   mmchconfig dataStructureDump=/var/adm/ras/
   ```

3. Set desired trace classes and levels.

   Note: This part of the process is identical to classic IBM Spectrum Scale installs. For more information, see Generating GPFS trace reports in IBM Spectrum Scale documentation.

   ```
   mmtracectl --set --trace={io | all | def | "Class Level [Class Level ...]"}
   ```

4. Start the trace facility on all nodes by entering the following command:

   ```
   mmtracectl --start
   ```

5. Re-create the problem.
6. Stop the trace generation as soon as the problem to be captured occurs, by entering the following command:

   ```
   mmtracectl --stop
   ```

7. Turn off trace generation by entering the following command:

   ```
   mmtracectl --off
   ```

Kernel crash dumps

Red Hat Enterprise Linux CoreOS (RHCOS) based machines do not support configuring kdump or generating kernel crash dumps for Red Hat OpenShift Container Platform 4.6 and earlier. For more information, see How to configure kdump in Red Hat CoreOS in Red Hat OpenShift documentation.

In some virtual machine installations, it may be possible to generate a vmcore crash dump from the hypervisor.

In lieu of kernel dumps, CoreOS currently recommends using pstore, even if only small snippets of diagnostic data can be collected. For more information, see Using pstore in CoreOS documentation on GitHub.

Chapter 14. Troubleshooting

Use the following sections to help troubleshoot and debug specific issues with the IBM Spectrum Scale container native deployment.

Debugging the IBM Spectrum Scale operator

Problem: The operator pod is not successfully deployed

No operator pod appears when running `oc get pods -n ibm-spectrum-scale-operator`.

- Verify that all worker nodes in the Red Hat OpenShift Container Platform cluster are in a Ready state. If not, the operator pod may not have an eligible node to be deployed to.

```
# oc get nodes
NAME                   STATUS      ROLES    AGE   VERSION
master0.example.com    Ready       master   65d   v1.18.3+6c42de8
master1.example.com    Ready       master   65d   v1.18.3+6c42de8
master2.example.com    Ready       master   65d   v1.18.3+6c42de8
worker0.example.com    NotReady    worker   65d   v1.18.3+6c42de8
worker1.example.com    NotReady    worker   65d   v1.18.3+6c42de8
worker2.example.com    NotReady    worker   65d   v1.18.3+6c42de8
```

- Inspect the operator namespace and look for details that may point to any problems.

```
oc get deployment -n ibm-spectrum-scale-operator
oc describe deployment ibm-spectrum-scale-controller-manager -n ibm-spectrum-scale-operator

oc get replicasets -n ibm-spectrum-scale-operator
oc describe replicaset <replicaset name> -n ibm-spectrum-scale-operator
```

Problem: Operator pod shows container restarts

- Kubernetes keeps the logs of the current container and the previous container. Take a look at the previous container's logs for any clues by using the following command:

```
oc logs -p <operator pod> -n ibm-spectrum-scale-operator
```

Debugging IBM Spectrum Scale deployment

Problem: No endpoints available for service "ibm-spectrum-scale-webhook-service"

When applying the various CRs or making changes to defined IBM Spectrum Scale container native Custom Resources, it is possible that validating or mutating webhooks can fail if the operator pod is unavailable. If you receive an error regarding `no endpoints for service "ibm-spectrum-scale-webhook-service"`, check the status of the operator pod.

Example error:

```
# oc apply -f remotecluster.yaml
remotecluster.scale.spectrum.ibm.com/remotecluster-sample unchanged
Error from server (InternalError): error when creating "remotecluster.yaml": Internal
error occurred: failed calling webhook "mcluster.scale.spectrum.ibm.com": failed to call
webhook: Post "https://ibm-spectrum-scale-webhook-service.ibm-spectrum-scale-operator.svc:443/
mutate-scale-spectrum-ibm-com-v1beta1-cluster?timeout=10s": no endpoints available for service
"ibm-spectrum-scale-webhook-service"
```

Checking status of the operator pod:

© Copyright IBM Corp. 2020, 2023

```
# oc get pods -n ibm-spectrum-scale-operator
NAME                                                         READY   STATUS
RESTARTS         AGE
pod/ibm-spectrum-scale-controller-manager-64bb4798df-rrj4j   0/1     ImagePullBackOff   10
(4m14s ago)   34m
```

In the above example, it appears that there might be some issue with image pull credentials. As a remedy to the `no endpoints available` issue, the operator issue must be resolved first. Once it is resolved, perform the steps again that failed with `no endpoints available`.

Problem: Core, GUI, or collector pods are in ErrImgPull or ImagePullBackOff state

When viewing `oc get pods -n ibm-spectrum-scale`, if any of the pods are in `ErrImgPull` or `ImagePullBackOff` state, use `oc describe pod` <podname> to get more details on the pod and look for any errors that may be happening.

```
oc describe pod <pod-name> -n ibm-spectrum-scale
```

Problem: Core, GUI, or collector pods are not up

- If the pods are not deployed in the `ibm-spectrum-scale` namespace, or a cluster is not created, examine the operator pod logs:

  ```
  oc logs $(oc get pods -n ibm-spectrum-scale-operator -ojson  | jq -r
  ".items[0].metadata.name") -n ibm-spectrum-scale-operator
  ```

Problem: Core, GUI, or collector pods show container restarts

- Kubernetes keeps the logs of the current container and the previous container. Check the previous container's logs for any clues by using the following command:

  ```
  oc logs -p <scale pod> -n ibm-spectrum-scale
  ```

Problem: Core pods are stuck in Init:1/2

If for some reason, an IBM Spectrum Scale container native cluster fails to create the core pods on the worker nodes get stuck in the Init container.

```
# oc get pods
NAME                              READY   STATUS    RESTARTS   AGE
...
worker0                           2/2     Init:1/2   0         2h
worker1                           2/2     Init:1/2   0         2h
worker2                           2/2     Init:1/2   0         2h
worker3                           2/2     Init:1/2   0         2h
```

There is no recovery from this. For more information about clean up, see "Cleanup IBM Spectrum Scale container native" on page 88 and "Cleanup OpenShift nodes" on page 89. For more information on redeploying, see Chapter 5, "Installing the IBM Spectrum Scale container native operator and cluster," on page 45 .

Problem: All pods have been deployed but a GPFS cluster is stuck in the "arbitrating" state

If the cluster is stuck in the arbitrating state:

- Check the output of `mmlscluster`.

  ```
  oc exec $(oc get pods -lapp.kubernetes.io/name=core -n ibm-spectrum-scale -o json | jq -r
  ".items[0].metadata.name") -- mmlscluster
  ```

- Check the GPFS logs.

```
oc logs $(oc get pods -lapp.kubernetes.io/name=core -n ibm-spectrum-scale  -o json | jq -r
".items[0].metadata.name") -c logs | grep mmfs.log.latest
```

Problem: A remote mount file system not getting configured or mounted

- Check the `RemoteCluster` objects and the `Filesystem` objects. The `Filesystem` controller waits until a RemoteCluster object is Ready before attempting to configure the remote mount file system. Describe the objects and check `Status` or `Events` for any reasons for failures.
  - Remote Clusters

    ```
    oc get remotecluster.scale -n ibm-spectrum-scale
    oc describe remotecluster.scale <name> -n ibm-spectrum-scale
    ```

  - Filesystems

    ```
    oc get filesystem.scale -n ibm-spectrum-scale
    oc describe filesystem.scale <name> -n ibm-spectrum-scale
    ```

  Check the `Status` and `Events` for any reason of failures.

  If nothing, check the operator logs for any errors:

  ```
  oc logs $(oc get pods -n ibm-spectrum-scale-operator -ojson  | jq -r
  ".items[0].metadata.name") -n ibm-spectrum-scale-operator
  ```

- Enter the **`mmnetverify`** command to verify the network between the clusters. For more information, see mmnetverify command in IBM Spectrum Scale documentation.

Debugging the IBM Spectrum Scale Container Storage Interface (CSI) deployment

Problem: CSI pods stuck in CrashLoopBackOff (Unauthorized GET request)

```
# oc get pods
NAME                                               READY   STATUS             RESTARTS   AGE
ibm-spectrum-scale-csi-9566l                       1/2     CrashLoopBackOff   9          26m
ibm-spectrum-scale-csi-attacher-0                  1/1     Running            0          85m
ibm-spectrum-scale-csi-klr7x                       1/2     CrashLoopBackOff   9          26m
ibm-spectrum-scale-csi-operator-56955949c4-mzn7g   1/1     Running            0          90m
ibm-spectrum-scale-csi-provisioner-0               1/1     Running            0          85m
ibm-spectrum-scale-csi-xlxkl                       1/2     CrashLoopBackOff   9          26m
```

```
# oc logs ibm-spectrum-scale-csi-9566l -c ibm-spectrum-scale-csi

...

I1218 17:27:33.875884       1 http_utils.go:60] http_utils FormatURL. url: https://ibm-spectrum-
scale-gui-ibm-spectrum-scale.apps.example.com:443/
I1218 17:27:33.875894       1 rest_v2.go:586] rest_v2 doHTTP. endpoint: https://ibm-spectrum-
scale-gui-ibm-spectrum-scale.apps.example.com:443/scalemgmt/v2/cluster, method: GET, param:
<nil>
I1218 17:27:33.875900       1 http_utils.go:74] http_utils HttpExecuteUserAuth. type:
GET, url: https://ibm-spectrum-scale-gui-ibm-spectrum-scale.apps.example.com:443/scalemgmt/v2/
cluster, user: csi-cnsa-gui-user
```

- Check that the `csi-cnsa-gui-user` role was created.

  ```
  # oc exec ibm-spectrum-scale-gui-0 -n ibm-spectrum-scale -- /usr/lpp/mmfs/gui/cli/lsuser
  Defaulting container name to liberty.
  Use 'oc describe pod/ibm-spectrum-scale-gui-0 -n ibm-spectrum-scale' to see all of the
  containers in this pod.
  Name              Long name Password status Group names       Failed login attempts Target
  Feedback Date
  ContainerOperator           active          ContainerOperator 0
  EFSSG1000I The command completed successfully.
  ```

In this case, the `csi-cnsa-gui-user` role was not created. To resolve the issue, enter the following command to create a GUI user:

```
# oc exec -c liberty ibm-spectrum-scale-gui-0 -n ibm-spectrum-scale -- /usr/lpp/mmfs/gui/cli/
mkuser csi-cnsa-gui-user -p csi-cnsa-gui-password -g CsiAdmin
EFSSG0019I The user csi-cnsa-gui-user has been successfully created.
EFSSG1000I The command completed successfully.
```

- Check that the `csi-remote-mount-storage-cluster-1` secret was created with correct credentials.

```
# oc get secrets csi-remote-mount-storage-cluster-1 -n ibm-spectrum-scale-csi
-ojsonpath='{.data.username}' | base64 --decode
csi-cnsa-gui-user

# oc get secrets csi-remote-mount-storage-cluster-1  -n ibm-spectrum-scale-csi
-ojsonpath='{.data.password}' | base64 --decode
this-is-a-bad-password
```

In this case, the `csi-remote-mount-storage-cluster-1` secret was created without a correct password. To resolve the issue, enter the following command to delete the secret and recreate it with correct values:

```
# oc delete secrets csi-remote-mount-storage-cluster-1  -n ibm-spectrum-scale-csi
secret "csi-remote-mount-storage-cluster-1" deleted

# oc create secret generic csi-remote-mount-storage-cluster-1 --from-literal=username=csi-
cnsa-gui-user --from-literal=password=csi-cnsa-gui-password -n ibm-spectrum-scale-csi
secret/csi-remote-mount-storage-cluster-1 created

# oc label secret csi-remote-mount-storage-cluster-1 product=ibm-spectrum-scale-csi -n ibm-
spectrum-scale-csi
secret/csi-remote-mount-storage-cluster-1 labeled
```

Problem: CSI CR is never created

If all the core pods are running and an IBM Spectrum Scale container native cluster appears to be in a good state, the CSI CR should be created automatically. In some error paths this does not happen and causes the driver pods to not be scheduled:

Note: Only the operator pod is listed and no results are found for csiscaleoperators.

```
# oc get po,csiscaleoperator -n ibm-spectrum-scale-csi
NAME                                                   READY   STATUS    RESTARTS   AGE
pod/ibm-spectrum-scale-csi-operator-79bd756d58-ht6hf   1/1     Running   0          47h
```

- Check that the GUI pod(s) are up and running.

```
# oc get pods -n ibm-spectrum-scale
NAME                                   READY   STATUS    RESTARTS   AGE
ibm-spectrum-scale-gui-0               4/4     Running   0          3m58s
ibm-spectrum-scale-gui-1               4/4     Running   0          95s
ibm-spectrum-scale-pmcollector-0       2/2     Running   0          3m59s
worker0                                2/2     Running   0          3m59s
worker1                                2/2     Running   0          3m58s
worker2                                2/2     Running   0          3m58s
```

All GUI pods must be up and running before the CSI CR is created. Each pod can take a few minutes for all containers in the pod to enter the `Running` state.

- Check that the daemon status has a non-empty cluster ID.

```
# oc describe daemon -n ibm-spectrum-scale
```

Find the status section and ensure that the `Cluster ID` field exists and is not empty.

```
Status:
  Cluster ID:    3004252500454687654
  Cluster Name:  example.cluster.com
```

If those fields are missing then the IBM Spectrum Scale container native cluster is experiencing an issue. Check the operator logs for more information.

Debugging PVC creation issue : PVC binding in pending state

If there are PVC creation issues:

1. Check if multiple GUI's are installed at the remote storage cluster.
2. SSH to one of the GUI node and run the following command:

   ```
   mmlsfileset <filesystem name>
   ```

3. Check if fileset is unlinked.

If multiple GUI’s are installed then contact IBM support for further guidance.

Debugging OCP upgrade

Problem: GUI mount not getting refreshed as multiple OCP clusters are remote mounted on the same FS

To resolve the issue, unmount the FS from another OCP cluster.

```
# /usr/lpp/mmfs/gui/cli/runtask FILESYSTEM_MOUNT
err: Batch entry 3 INSERT INTO FSCC.FILESYSTEM_MOUNTS
(CLUSTER_ID, DEVICENAME, HOST_NAME, MOUNT_MODE, LAST_UPDATE)
VALUES ('5228226002706731921','fs1','worker1.example.com','RW','2021-07-28
19:06:15.111000+00'::timestamp) was aborted: ERROR: duplicate key value violates unique
constraint "filesystem_mounts_pk"
  Detail: Key (host_name, cluster_id, devicename)=(worker1.example.com, 5228226002706731921 ,
fs1) already exists.  Call getNextException to see other errors in the batch.
EFSSG1150C Running specified task was unsuccessful.
# /usr/lpp/mmfs/gui/cli/runtask FILESYSTEM_MOUNT
EFSSG1000I The command completed successfully.
# exit
exit
```

Identifying applications preventing cluster maintenance

Debug steps to determine when a cluster maintenance action is not being completed.

When would we see this?

A drain occurs when a core pod is selected for deletion. There are a number of actions that prompt a core pod deletion:

- Pod evictions
  - Red Hat OpenShift Machine Config Operator
  - User-initiated drains
- Pod spec updates
  - Resource requests, i.e. changing core pod requests for CPU and/or memory
  - Image updates prompted by a new release

What does this look like?

An update driven by pod spec updates will present as core pods awaiting deletion. An update driven by Red Hat OpenShift Machine Config Operator (MCO) ceases to update nodes. The signature will look similar between the two scenarios. Use the following steps to determine where to direct the support case.

When to open a support case?

To determine if the MCO is stuck due to IBM Spectrum Scale container native, run the following command on the node that is failing to update:

```
oc adm drain <node> --force --ignore-daemonsets --delete-emptydir-data --pod-
selector='app.kubernetes.io/instance notin(ibm-spectrum-scale)'
```

If this command completes without errors, then IBM Spectrum Scale container native is blocking the ongoing drain. To resolve it, raise a support ticket to IBM. For more information, see Gather data to submit a support ticket to IBM.

If errors are presented from the `oc adm drain` command, raise a support ticket to Red Hat. For more information, see Gather data to submit a support ticket to Red Hat OpenShift.

Identifying signatures of an ongoing update

Complete the following steps:

1. Check the Daemon status to verify if any pods are awaiting deletion.

   ```
   oc describe daemon -n ibm-spectrum-scale
   ```

   Example output:

   ```
   Status Details:
           Nodes Rebooting:
           Nodes Unreachable:
           Nodes Waiting For Reboot:
           Pods Starting:
           Pods Terminating:
           Pods Unknown:
           Pods Waiting For Delete:   worker0, worker1, worker2
           Quorum Pods:               worker0, worker1, worker2
   ```

2. Check if any nodes are cordoned.

   ```
   oc get nodes
   ```

   Example:

   ```
   NAME                      STATUS                     ROLES    AGE   VERSION
       master0.example.ibm.com   Ready                      master   23d   v1.24.0+3882f8f
       master1.example.ibm.com   Ready                      master   23d   v1.24.0+3882f8f
       master2.example.ibm.com   Ready                      master   23d   v1.24.0+3882f8f
       worker0.example.ibm.com   Ready,SchedulingDisabled   worker   23d   v1.24.0+3882f8f
       worker1.example.ibm.com   Ready                      worker   23d   v1.24.0+3882f8f
       worker2.example.ibm.com   Ready                      worker   23d   v1.24.0+3882f8f
   ```

3. Check events on the core pod of the failing node.

   ```
   oc describe <pod> -n ibm-spectrum-scale
   ```

   Example event:

   ```
   error when evicting pods/"mypod-0" -n "myworkload" (will retry after 5s): Cannot evict pod
   as it would violate the pod's disruption budget.
   ```

Recovery of IBM Spectrum Scale container native cluster after Red Hat OpenShift node failure

IBM Spectrum Scale container native operator does not have the managed support for the removal of a node. In some cases, a node is not recovered after the node suffers an outage, for example, after a Red Hat OpenShift update.

Before you begin

Before the maintenance of a failed node, familiarize with the node selectors, quorum and manager designations, and network configuration annotations. Because these configurations will be applied after the removal of the node to ensure parity of the cluster.

- Log the node selector labels required for the IBM Spectrum Scale core pods that are currently being used by the Cluster CR.
- Understand if the node to be removed is a quorum node for the IBM Spectrum Scale container native cluster.
- Identify if a CNI network configuration is currently in use for the IBM Spectrum Scale container native cluster.

Collect configuration prior to servicing the OCP cluster

Running an IBM Spectrum Scale container native must-gather will collect the state of the cluster, allowing a reference point of the cluster prior to servicing the Red Hat OpenShift cluster.

Collect an IBM Spectrum Scale container native must-gather to ensure that the current configuration is captured prior to attempting service of Red Hat OpenShift cluster.

```
oc adm must-gather --image=icr.io/cpopen/ibm-spectrum-scale-must-gather:v5.1.7.0
```

Delete one Scale node from IBM Spectrum Scale container native cluster

To delete the IBM Spectrum Scale node from IBM Spectrum Scale container native cluster, complete the following steps:

1. Stop the running operator pod by setting the `replicas` in the deployment to 0.

   ```
   oc scale deployment ibm-spectrum-scale-controller-manager -n ibm-spectrum-scale-operator --
   replicas=0
   ```

2. If the failed Red Hat OpenShift node exists, delete the corresponding IBM Spectrum Scale core pod.

   ```
   oc delete pod REPLACE_WITH_CORE_POD_OF_FAILED_OCP_NODE -n ibm-spectrum-scale
   ```

3. Enter a currently running IBM Spectrum Scale core pod to execute the remaining commands.

   ```
   oc -n ibm-spectrum-scale rsh REPLACE_WITH_RUNNING_CORE_POD
   ```

4. Note the quorum and manager designations as denoted in `mmlscluster`. This information is needed when to add a new node back.

   ```
   mmlscluster
   ```

5. Map the affected Red Hat OpenShift node to the GPFS admin interface. It must show a status as `Unknown`.

   Note: The GPFS admin interface would have a short name of the affected Red Hat OpenShift node.

   ```
   mmgetstate -a
   ```

6. Ensure that the most of IBM Spectrum Scale nodes are in the `Active` state. Then, force delete the desired GPFS admin interface from the IBM Spectrum Scale cluster.

   Caution: When shutting down GPFS on quorum nodes or deleting quorum nodes from the GPFS cluster, if the number of remaining quorum nodes is less than the requirement for a quorum, you cannot perform file system operations.

   ```
   mmdelnode -N REPLACE_WITH_FAILED_GPFS_ADMIN_NODE --force
   ```

7. Ensure that the bad GPFS admin node is deleted from the IBM Spectrum Scale cluster.

   ```
   mmgetstate -a
   ```

8. Exit from the IBM Spectrum Scale core pod.

   ```
   exit
   ```

Add a new Red Hat OpenShift node in an existing IBM Spectrum Scale container native cluster

The previous node configuration of the failed node must be applied to the new Red Hat OpenShift node. So that, the IBM Spectrum Scale container native operator can successfully add it to the existing IBM Spectrum Scale container native cluster.

1. Ensure that the quorum and manager designations matches with the previous node the quorum and manager designations, and the GPFS cluster is not in minority of quorum. If the number of remaining quorum nodes is less than the requirement for a quorum, you cannot perform file system operations.

   Note: If a node is removed without addition of a replacement node, ensure that the sufficient nodes are assigned to the quorum.

2. If you are using CNI, ensure that the network configuration is valid according to the CNI documentation. Ensure that the `scale.spectrum.ibm.com/daemon-network` annotation is present on the new Red Hat OpenShift node. For more information, see "Container Network Interface (CNI) configuration" on page 14.
3. Ensure the node selector labels (required for the IBM Spectrum Scale core pods) from the cluster CR are present on the newly added Red Hat OpenShift node.
4. Scale up the operator pods by setting the `replicas` in the deployment to 1.

   ```
   oc scale deployment ibm-spectrum-scale-controller-manager -n ibm-spectrum-scale-operator --
   replicas=1
   ```

5. Ensure that the new IBM Spectrum Scale core pod is created and runs immediately.

Validate the IBM Spectrum Scale container native cluster after the removal and replacement of the Red Hat OpenShift node

After the removal and replacement of the failed Red Hat OpenShift node completes, perform the following steps. For more information, see "Verifying an IBM Spectrum Scale container native cluster" on page 67.

- Ensure that all pods are in running state.

  Note: Ensure that the CSI pod on the replaced node is running. This validates that file systems are active and running.

- Validate that the IBM Spectrum Scale cluster has successfully added the node with the desired node designations.
- Validate that the nodes are active in the IBM Spectrum Scale cluster.

Chapter 15. References

IBM Spectrum Scale

- Administration Guide
- For Linux on Z: Changing the kernel settings
- mmchconfig command
- mmnetverify command
- Accessing a remote GPFS file system
- Defining the cluster topology for the installation toolkit
- Node quorum
- Installing IBM Spectrum Scale Container Storage Interface driver using CLI

Red Hat OpenShift or Kubernetes

- Display which Pods have the PVC in use
- Red Hat OpenShift Container Platform 4 now defaults to CRI-O as underlying container engine
- How to configure kdump in Red Hat CoreOS?
- Installing and configuring OpenShift Container Platform clusters
- Installation Configuration
- Configuring an HTPasswd identity provider

© Copyright IBM Corp. 2020, 2023

Accessibility features for IBM Spectrum 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 Spectrum 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.

© Copyright IBM Corp. 2020, 2023

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

© Copyright IBM Corp. 2020, 2023

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 Spectrum 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 Object 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 Spectrum 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 Spectrum 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.

© Copyright IBM Corp. 2020, 2023

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 Spectrum 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 Spectrum 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 Spectrum 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 Spectrum 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 Spectrum 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 Spectrum 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 Spectrum Scale file systems. For more information, a sample program is provided in the following directory on IBM Spectrum Scale nodes: `/usr/lpp/mmfs/samples/util` called tswf that can be modified according to the user's needs.

Index

Part Number:

SC28-3168-21

(1P) P/N: