IBM Spectrum Scale Container Native Storage Access
5.1.4

IBM Spectrum Scale
Container Native Storage Access Guide

SC28-3168-18

Note

Before using this information and the product it supports, read the information in "Notices" on page 97.

This edition applies to Version 5 release 1 modification 4 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)

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

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

Contents

Tables

About this information

This edition applies to IBM Spectrum Scale version 5.1.4 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 viii.

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.

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 **Planning** • Planning for GPFS • Planning for protocols • Planning for Cloud services • Planning for AFM • Planning for AFM DR • Planning for AFM to cloud object storage • Firewall recommendations	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	• Considerations for GPFS applications • Security-Enhanced Linux support • Space requirements for call home data upload	
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 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 issues • 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 • 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	This guide provides the following information: **Command reference** • 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	• 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	• 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 • 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	• mmrestripefs command • mmrpldisk command • mmsdrrestore command • mmsetquota command • mmshutdown command • mmsmb command • mmsnapdir command • mmstartup 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 • 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 Cloudera HDP 2.6 • Planning • Installation • Upgrading software stack • Configuration • Administration • Troubleshooting • Limitations • FAQ	• 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 • Incorporating IBM Spectrum Scale Erasure Code Edition in an Elastic Storage Server (ESS) cluster • 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 • 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

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 xxv 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.
`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

The overview of IBM Spectrum Scale container native includes the following topics:

- Introduction
- What's new?
- Supported features
- Limitations

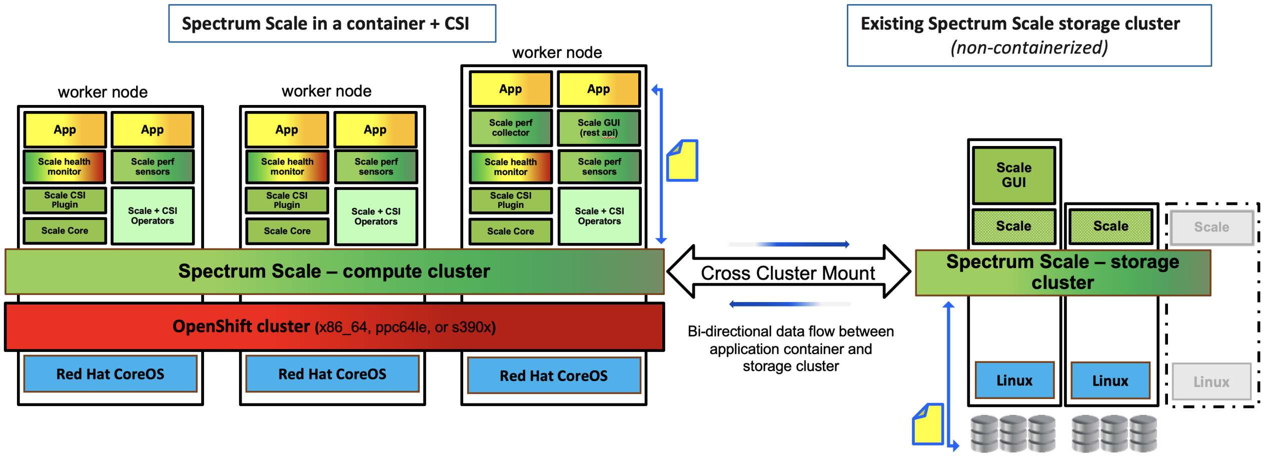
Introduction

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. Generic remote mount


Spectrum Scale in a container + CSI
worker node
worker node
worker node
App
App
Scale health monitor
Scale perf sensors
Scale CSI Plugin
Scale + CSI Operators
Scale Core
Scale perf collector
Scale GUI (rest api)
Spectrum Scale – compute cluster
OpenShift cluster (x86_64, ppc64le, or s390x)
Red Hat CoreOS
Existing Spectrum Scale storage cluster
(non-containerized)
Cross Cluster Mount
Bi-directional data flow between application container and storage cluster
Scale GUI
Scale
Spectrum Scale – storage cluster
Linux

For more information about direct storage attachment, see "Deployment considerations" on page 8.

What's new?

The following enhancements are made in this release:

- Ability to upgrade IBM Spectrum Scale container native from 5.1.3.x to 5.1.4. For more information, see Chapter 5, "Upgrading IBM Spectrum Scale container native," on page 59.
- Support for IBM Spectrum Scale Container Storage Interface (CSI) 2.6.0. For more information, see What's New in IBM Spectrum Scale CSI documentation.
- Support for IBM Spectrum Scale Data Access Services (DAS) 5.1.3.1. For more information, see IBM Spectrum Scale Data Access Services.

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.

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
- Hardware requirements
- Software requirements
- Deployment considerations
- Container image list for IBM Spectrum Scale container native
- Roles and personas

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

Complete the following steps:

1. 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.
2. Validate and apply the configuration to the Red Hat OpenShift installation settings.
3. Obtain IBM Cloud Container Registry entitlement key in order to access the container images of IBM Spectrum Scale container native.
4. 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.
5. 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.
2. Download the sample YAML file from the GitHub repository.

Note: The `scale_v1beta1_cluster_cr.yaml` sample file is a collection of multiple custom resources and kinds.

a. Configure the Cluster custom resource that is used for deployment of the Operator.

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

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

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

iv. Configure Ephemeral Port Range, if necessary.

v. Enable the optional Grafana Bridge.

b. Configure the Callhome custom resource.

c. Configure the Filesystem custom resource.

i. Define the RemoteCluster resource.

ii. Define the file system on the RemoteCluster to mount.

d. Configure the RemoteCluster custom resource.

i. Populate the details of the storage cluster GUI.

3. Create an IBM Spectrum Scale container native cluster by deploying the configured `scale_v1beta1_cluster_cr.yaml` file.
4. If accessing encrypted data on the storage cluster, download and configure the `EncryptionConfig` custom resource YAML file, `scale_v1beta1_encryptionconfig_cr.yaml`, from the GitHub repository.
5. 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.
6. Create a secret by using the storage cluster GUI user credentials for the `ContainerOperator` GUI user in the `ibm-spectrum-scale` namespace.
7. Create a secret by using the storage cluster GUI user credentials for `CsiAdmin` GUI user in the `ibm-spectrum-scale-csi` namespace.
8. Create a storage class to create volumes to use with your container native cluster.

Hardware requirements

Note: IBM Spectrum Scale container native 5.1.4 supports only on-premises environments (customer infrastructure) and does not support cloud environments.

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

The following table lists the minimum and recommended worker node requirements per each OpenShift Container Platform worker node.

The IBM Spectrum Scale container native that is running on the OpenShift Container Platform recommends a minimum of three worker nodes for the cluster. It supports a maximum of 128 worker nodes.

Table 4. Hardware requirements

Architecture	Minimum		Recommended	
	CPU (Cores)	Memory (GB)	CPU (Cores)	Memory (GB)
x86_64	8	16	16	64
PPC64LE	8	16	16	64
s390x	4	8	8	16

IBM Spectrum Scale storage cluster (remote cluster)

The IBM Spectrum Scale storage cluster that is used as a remote cluster must run IBM Spectrum Scale 5.1.4 or later.

Software requirements

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

Table 5. Software requirements

IBM Spectrum Scalecontainer native	IBM Spectrum ScaleContainer 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
5.1.4.0	2.6.0	x86,ppc64le,x390x	5.1.3.0+	28.00	4.8, 4.9, 4.10	4.8, 4.9, 4.10	8.6
5.1.3.0	2.5.0	x86,ppc64le,x390x	5.1.3.0+	27.00	4.8, 4.9, 4.10	4.8, 4.9, 4.10	8.5
5.1.2.1	2.4.0	x86,ppc64le,x390x	5.1.2.1+	26.00	4.8, 4.9	4.8, 4.9	8.5
5.1.1.4	2.3.1	x86,ppc64le,s390x	5.1.1.4+	25.00	4.8	4.8	8.4
5.1.1.3	2.3.0	x86, ppc64le, s390x	5.1.1.3+	25.00	4.8	4.8	8.4

Note: For more information about the minimum levels required for specific CSI functions, see 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 2.6.0 is installed in conjunction with IBM Spectrum Scale container native 5.1.4, and requires an IBM Spectrum Scale storage cluster running on 5.1.4 or later.

Storage cluster

- The remotely mounted file system should be at file system format level 28.00 or earlier. For more information, see Upgrading multi-cluster environments in IBM Spectrum Scale documentation.

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, Red Hat OpenShift cluster persistent storage, and storage cluster considerations.

The following list includes the Red Hat OpenShift cluster considerations:

- The IBM Spectrum Scale pods use the host network. They do not use the Container Network Interface (CNI) overlay network.
- The DNS must be configured properly so that the worker nodes can resolve the storage cluster nodes. For more information, see Host aliases.
- All worker nodes must be able to communicate with each other through the host network.
- 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.
- A minimum configuration of three master nodes and three worker nodes, with a maximum of 128 worker nodes is required.

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 Cleaning up the worker nodes and Cleaning up IBM Spectrum Scale operator.
- IBM Spectrum Scale container native does not support the use of dynamically created or pre-created PVs.

The following list includes the storage cluster considerations:

- The storage cluster must be at IBM Spectrum Scale 5.1.4 or later to support new functions introduced with IBM Spectrum Scale Container Storage Interface Driver (CSI) 2.6.0.

- The remotely mounted file system should be at file system format level 28.00 or earlier.
- Encrypted file systems are supported. Configure the EncryptionConfig custom resource with the necessary key client and key server information. For more information, see EncryptionConfig.

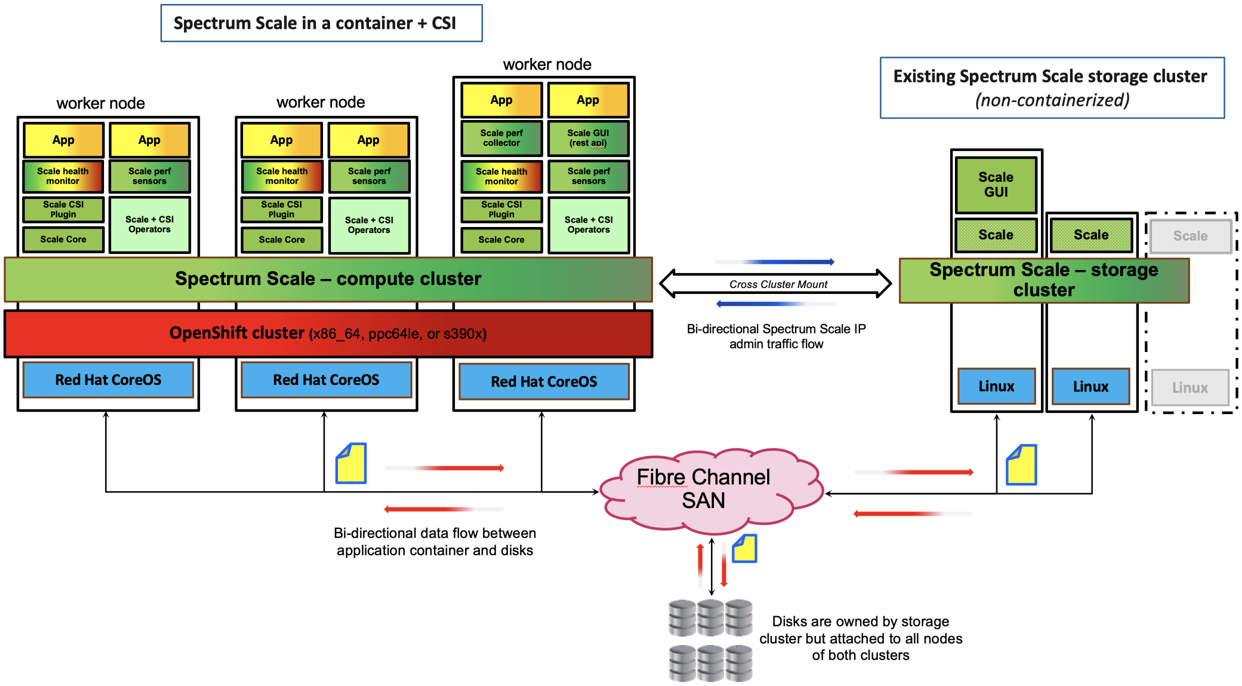
The following list includes the considerations for enterprise grade image registry:

- 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.

The following list includes the considerations for direct storage attachment:

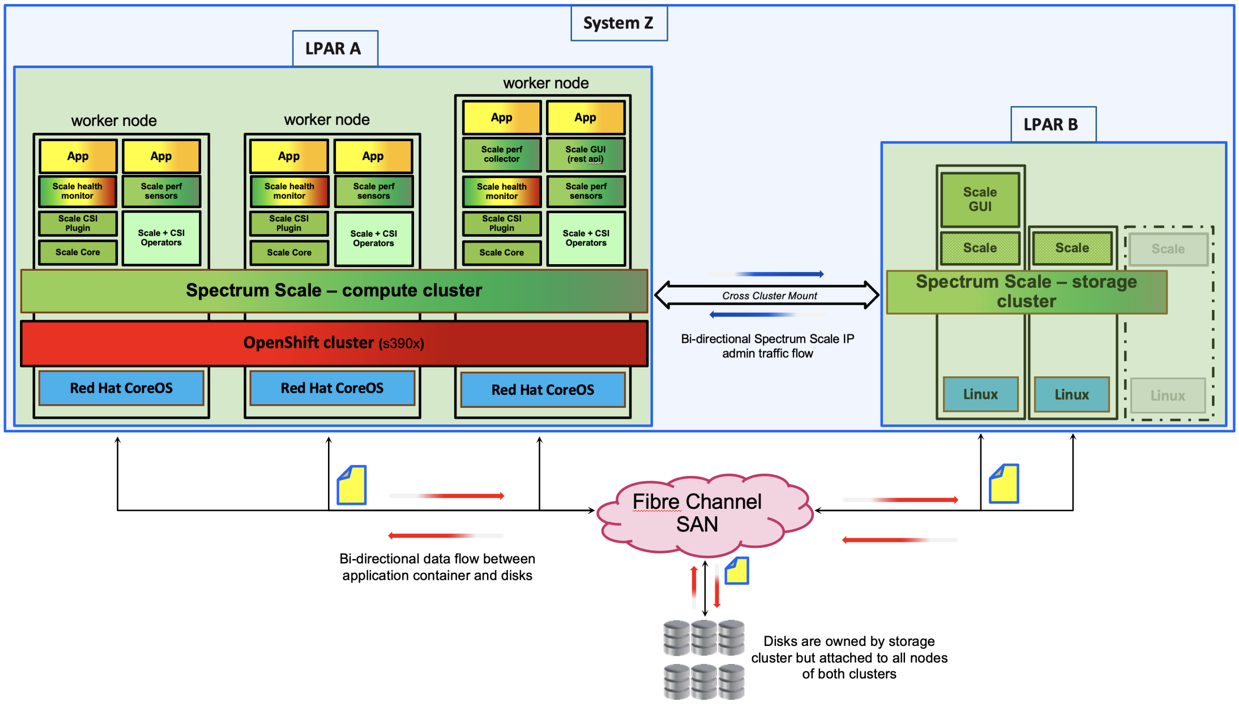
- Support for direct storage attachment on x86, power, and Z servers. In direct storage attachment configuration, the worker nodes use the SAN fabric instead of the IBM Spectrum Scale NSD protocol for I/O traffic.
- If using Power or x86 servers, it might be necessary to load multi-path drivers through 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
App
Scale health monitor
Scale perf sensors
Scale CSI Plugin
Scale Core
Scale + CSI Operators
Scale perf collector
Scale GUI (rest api)
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.
- If using Power or x86 servers, it might be necessary to load multi-path drivers through Red Hat CoreOS before storage can be seen.
- In direct storage attachment configuration, the worker nodes use the SAN fabric instead of the IBM Spectrum Scale NSD protocol for I/O traffic. 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 CSI Plugin
Scale + CSI Operators
Scale Core
Scale perf collector
Scale GUI (rest api)
Spectrum Scale – compute cluster
OpenShift cluster (s390x)
Red Hat CoreOS
Cross Cluster Mount
Bi-directional Spectrum Scale IP admin traffic flow
LPAR B
Scale GUI
Scale
Spectrum Scale – storage cluster
Linux
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

Container image list for IBM Spectrum Scale container native

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.

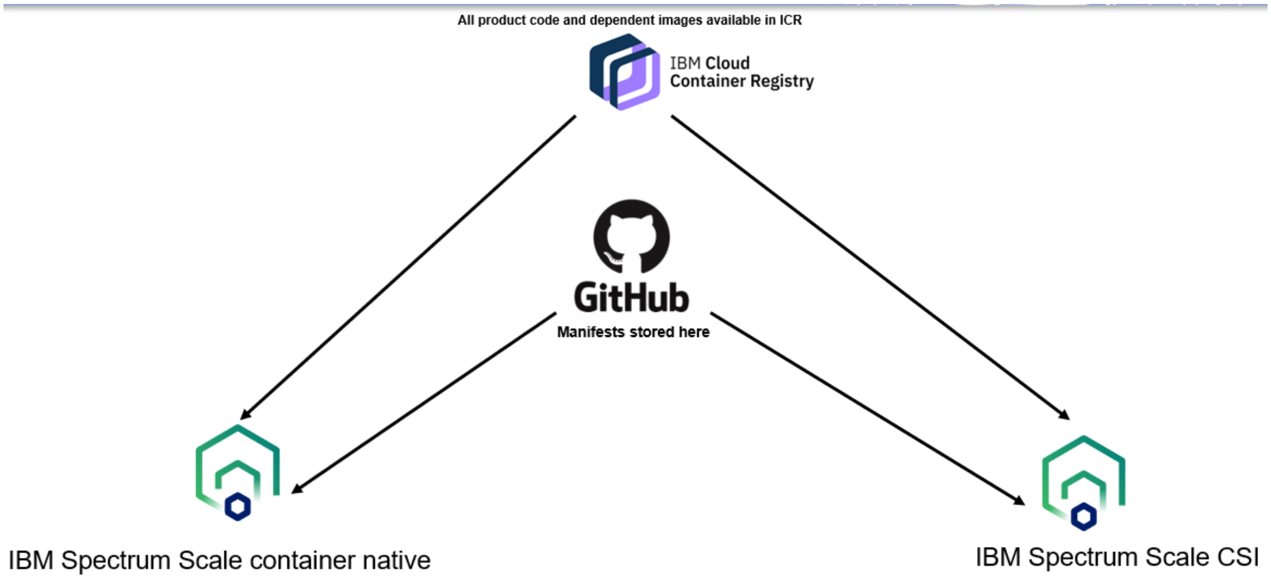

All product code and dependent images available in ICR
IBM Cloud Container Registry
GitHub
Manifests stored here
IBM Spectrum Scale container native
IBM Spectrum Scale CSI

Figure 4. Dependent images available in ICR

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

- "Container image list for IBM Spectrum Scale container native 5.1.4.0" on page 10
- "Container image list for IBM Spectrum Scale container native 5.1.4.1" on page 13

Container image list for IBM Spectrum Scale container native 5.1.4.0

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

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:aa81c3c1572c4b29dcac048563f0bd6242b649da7c63acc9dbb6612e1f00574c
ibm-spectrum-scale-csi-operator	operator	icr.io/cpopen/	ibm-spectrum-scale-csi-operator@sha256:b62ad588fbc338090f405740fd8bceaf59b3267cdb532c60ae965643ccae4d5d
must-gather-XXXXX	must-gather	icr.io/cpopen/	ibm-spectrum-scale-must-gather@sha256:76f85ac3be474a1dea0266e3b391aeee3a3ea7c44d8ffcbea0e715b8e5ce0839

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@sha256:9319f52cd83d0576fb72c0eb31bfc998039366a3766d5afcf431870423cd9bd5
[1] workerX/masterX	config	cp.icr.io/cp/spectrum/scale	ibm-spectrum-scale-core-init@sha256:9319f52cd83d0576fb72c0eb31bfc998039366a3766d5afcf431870423cd9bd5
[1] workerX/masterX	gpfs (if using Data Access Edition)	cp.icr.io/cp/spectrum/scale/data-access	ibm-spectrum-scale-daemon@sha256:4635ab99adee4c2645a952805b9a4494390368fbc3a0badd4f4cef552b05147a
[1] workerX/masterX	gpfs (if using Data Management Edition)	cp.icr.io/cp/spectrum/scale/data-management	ibm-spectrum-scale-daemon@sha256:84087b10860da922ef1ba481ffe747a5192fd1a96696f5fb8e7188a1006d0bc9
[1] workerX/masterX	logs	cp.icr.io/cp/spectrum/scale	ubi-minimal@sha256:e83a3146aa8d34dccfb99097aa79a3914327942337890aa6f73911996a80ebb8
ibm-spectrum-scale-gui-X	liberty	cp.icr.io/cp/spectrum/scale	ibm-spectrum-scale-gui@sha256:797701bf382fb1125f71281382564a1ee82f7ebe1fecb42df465d11c86f06234
ibm-spectrum-scale-gui-X	sysmon	cp.icr.io/cp/spectrum/scale	ibm-spectrum-scale-monitor@sha256:c52ca470b2ef8e7f4020413d10ad82a4c7c52f8988a10b62dbaba4a254c03a5b

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

Pod	Container	Repository	Image
ibm-spectrum-scale-gui-X	postgres	cp.icr.io/cp/spectrum/scale	postgres@sha256:3162a6ead070474b27289f09eac4c865e75f93847a2d7098f718ee5a721637c4
ibm-spectrum-scale-gui-X	logs	cp.icr.io/cp/spectrum/scale	ubi-minimal@sha256:e83a3146aa8d34dccfb99097aa79a3914327942337890aa6f73911996a80ebb8
ibm-spectrum-scale-pmcollector-X	pmcollector	cp.icr.io/cp/spectrum/scale	ibm-spectrum-scale-pmcollector@sha256:ed565dea825dc16423ebd84a9532fb8498a55b56f75d6a4e9ae060b9c0fdad18
ibm-spectrum-scale-pmcollector-X	sysmon	cp.icr.io/cp/spectrum/scale	ibm-spectrum-scale-monitor@sha256:c52ca470b2ef8e7f4020413d10ad82a4c7c52f8988a10b62dbaba4a254c03a5b
ibm-spectrum-scale-csi-snapshotter	csi-snapshotter	cp.icr.io/cp/spectrum/scale/csi	csi-snapshotter@sha256:89e900a160a986a1a7a4eba7f5259e510398fa87ca9b8a729e7dec59e04c7709
ibm-spectrum-scale-csi-attacher	ibm-spectrum-scale-csi-attacher	cp.icr.io/cp/spectrum/scale/csi	csi-attacher@sha256:8b9c313c05f54fb04f8d430896f5f5904b6cb157df261501b29adc04d2b2dc7b
ibm-spectrum-scale-csi-provisioner	csi-provisioner	cp.icr.io/cp/spectrum/scale/csi	csi-provisioner@sha256:122bfb8c1edabb3c0edd63f06523e6940d958d19b3957dc7b1d6f81e9f1f6119
ibm-spectrum-scale-csi-driver-XXXXX	liveness-probe	cp.icr.io/cp/spectrum/scale/csi	livenessprobe@sha256:933940f13b3ea0abc62e656c1aa5c5b47c04b15d71250413a6b821bd0c58b94e
ibm-spectrum-scale-csi-driver-XXXXX	driver-registrar	cp.icr.io/cp/spectrum/scale/csi	csi-node-driver-registrar@sha256:4fd21f36075b44d1a423dfb262ad79202ce54e95f5cbc4622a6c1c38ab287ad6
ibm-spectrum-scale-csi-resizer-X	ibm-spectrum-scale-csi-resizer	cp.icr.io/cp/spectrum/scale/csi	csi-resizer@sha256:9ebbf9f023e7b41ccee3d52afe39a89e3ddacdbb69269d583abfc25847cfd9e4
ibm-spectrum-scale-csi-driver-XXXXX	ibm-spectrum-scale-csi	cp.icr.io/cp/spectrum/scale/csi	ibm-spectrum-scale-csi-driver@sha256:1efa61d65ee33c574cd2826f1bc6194631b1905996eddaa2ad51b66073e0c409
ibm-spectrum-scale-grafana-bridge-X	grafanabridge	cp.icr.io/cp/spectrum/scale	ibm-spectrum-scale-grafana-bridge@sha256:6c66b16b228f32a7d755794fe7abb1792ef4796ca189eaa9447a10869bcf216a

[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.

Container image list for IBM Spectrum Scale container native 5.1.4.1

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 8. 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@sha256:3f6e90c08d232338bc21127805a285c7debdbf5cec512ef84de9899a700c438e
ibm-spectrum-scale-csi-operator	operator	icr.io/cpopen/	ibm-spectrum-scale-csi-operator@sha256:9ac13e80b2a9299807f4249b23967fc2517a4aa34528d1ec524c5c8eb2691fd0
must-gather-XXXXX	must-gather	icr.io/cpopen/	ibm-spectrum-scale-must-gather@sha256:76f85ac3be474a1dea0266e3b391aeee3a3ea7c44d8ffcbea0e715b8e5ce0839

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 9. 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@sha256:4c35ecba95011443362cdc7cf07acba14eea22c48d5938a19a2027789130dfeb
[1] workerX/masterX	config	cp.icr.io/cp/spectrum/scale	ibm-spectrum-scale-core-init@sha256:4c35ecba95011443362cdc7cf07acba14eea22c48d5938a19a2027789130dfeb
[1] workerX/masterX	gpfs (if using Data Access Edition)	cp.icr.io/cp/spectrum/scale/data-access	ibm-spectrum-scale-daemon@sha256:c1f4b9be01d8083f43634038daf3e3a3c855042687a83e36c39d0afe144effc3
[1] workerX/masterX	gpfs (if using Data Management Edition)	cp.icr.io/cp/spectrum/scale/data-management	ibm-spectrum-scale-daemon@sha256:ca9d2b844b27583e67c4101b2c74d5959b3f5c310b2a1a995e8df056c2e92fd7

Table 9. Images acquired from entitled IBM Container Repository (continued)

Pod	Container	Repository	Image
[1] workerX/masterX	logs	cp.icr.io/cp/spectrum/scale	ubi-minimal@sha256:e83a3146aa8d34dccfb99097aa79a3914327942337890aa6f73911996a80ebb8
ibm-spectrum-scale-gui-X	liberty	cp.icr.io/cp/spectrum/scale	ibm-spectrum-scale-gui@sha256:522af21446583c5a9bf2797d0e952149379264881472678ffe6706b647313430
ibm-spectrum-scale-gui-X	sysmon	cp.icr.io/cp/spectrum/scale	ibm-spectrum-scale-monitor@sha256:88844b0f29e32b54d869e831538b0a6063e3a5a5a0b284691da1c63ec1336d80
ibm-spectrum-scale-gui-X	postgres	cp.icr.io/cp/spectrum/scale	postgres@sha256:3162a6ead070474b27289f09eac4c865e75f93847a2d7098f718ee5a721637c4
ibm-spectrum-scale-gui-X	logs	cp.icr.io/cp/spectrum/scale	ubi-minimal@sha256:e83a3146aa8d34dccfb99097aa79a3914327942337890aa6f73911996a80ebb8
ibm-spectrum-scale-pmcollector-X	pmcollector	cp.icr.io/cp/spectrum/scale	ibm-spectrum-scale-pmcollector@sha256:d859df59dc1937ad525562be97e811ddc05526bf2597717b99df11f5053f2dd1
ibm-spectrum-scale-pmcollector-X	sysmon	cp.icr.io/cp/spectrum/scale	ibm-spectrum-scale-monitor@sha256:88844b0f29e32b54d869e831538b0a6063e3a5a5a0b284691da1c63ec1336d80
ibm-spectrum-scale-csi-snapshotter	csi-snapshotter	cp.icr.io/cp/spectrum/scale/csi	csi-snapshotter@sha256:89e900a160a986a1a7a4eba7f5259e510398fa87ca9b8a729e7dec59e04c7709
ibm-spectrum-scale-csi-attacher	ibm-spectrum-scale-csi-attacher	cp.icr.io/cp/spectrum/scale/csi	csi-attacher@sha256:8b9c313c05f54fb04f8d430896f5f5904b6cb157df261501b29adc04d2b2dc7b
ibm-spectrum-scale-csi-provisioner	csi-provisioner	cp.icr.io/cp/spectrum/scale/csi	csi-provisioner@sha256:122bfb8c1edabb3c0edd63f06523e6940d958d19b3957dc7b1d6f81e9f1f6119
ibm-spectrum-scale-csi-driver-XXXXX	liveness-probe	cp.icr.io/cp/spectrum/scale/csi	livenessprobe@sha256:933940f13b3ea0abc62e656c1aa5c5b47c04b15d71250413a6b821bd0c58b94e
ibm-spectrum-scale-csi-driver-XXXXX	driver-registrar	cp.icr.io/cp/spectrum/scale/csi	csi-node-driver-registrar@sha256:4fd21f36075b44d1a423dfb262ad79202ce54e95f5cbc4622a6c1c38ab287ad6

Table 9. Images acquired from entitled IBM Container Repository (continued)

Pod	Container	Repository	Image
ibm-spectrum-scale-csi-resizer-X	ibm-spectrum-scale-csi-resizer	cp.icr.io/cp/spectrum/scale/csi	csi-resizer@sha256:9ebbf9f023e7b41ccee3d52afe39a89e3ddacdbb69269d583abfc25847cfd9e4
ibm-spectrum-scale-csi-driver-XXXXX	ibm-spectrum-scale-csi	cp.icr.io/cp/spectrum/scale/csi	ibm-spectrum-scale-csi-driver@sha256:8340faaee3bd3a4ac19047dc17fa3cf01c7350d49a783e1729468d7771f5d5be
ibm-spectrum-scale-grafana-bridge-X	grafanabridge	cp.icr.io/cp/spectrum/scale	ibm-spectrum-scale-grafana-bridge@sha256:6c66b16b228f32a7d755794fe7abb1792ef4796ca189eaa9447a10869bcf216a

[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.

Roles and personas

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

Persona

Red Hat OpenShift Cluster administrator must deploy the IBM Spectrum Scale container native cluster.

Operator permissions

The IBM Spectrum Scale container native operator is a namespace-scoped operator. The operator watches the namespace that it is deployed into. As part of the operator installation, you can deploy various role-based access control (RBAC) related YAML files that control the operator's access to resources within the namespace it is watching. While the operator is running with a namespace scope, it requires access to cluster level resources to successfully deploy. Access to cluster level resources is handled through a cluster role that is deployed during the deployment of RBAC YAML files. The role and cluster role are bound to the custom `ibm-spectrum-scale-operator` ServiceAccount, which the operator uses to create the IBM Spectrum Scale container native cluster.

ibm-spectrum-scale-operator role

Table 10. ibm-spectrum-scale-operator role

Resources	Verbs	API Groups
pods, pods/exec, services, serviceaccounts, configmaps, secrets, services/finalizers	*	-
roles, rolebindings	*	rbac.authorization.k8s.io

Table 10. ibm-spectrum-scale-operator role (continued)

Resources	Verbs	API Groups
leases	get, create, update	coordination.k8s.io
daemonsets, replicasets, statefulsets	*	apps
servicemonitors	get, create	monitoring.coreos.com
deployments, deployments\finalizers (resourceName=ibm-spectrum-scale-operator only)	get, update	apps
*	*	ibm.com
scaleclusters/status	get, patch, update	scale.ibm.com
scaleclusters, scaleclusters/finalizers	create, delete, get, list, patch, update, watch	scale.ibm.com

ibm-spectrum-scale-operator cluster role

Table 11. ibm-spectrum-scale-operator cluster role

Resources	Verbs	API Groups
nodes, services, events	get, list, create, patch, watch	-
persistentvolumes, persistentvolumes/finalizers, persistentvolumeclaims	get, list, create, patch, delete	-
statefulsets	get	apps
securitycontextconstraints	get, list, watch, create, update, patch, delete	security.openshift.io
storageclasses	get, list, patch, create	storage.k8s.com
clusterroles, clusterrolebindings	get, list, watch, create, update, patch, delete	rbac.authorization.k8s.io

Core pod permissions

You can collect a gpfs.snap from any running Spectrum Scale core pod for diagnostic log collection when seeking problem determination. The gpfs.snap contains both gpfs logs and captured output relevant to kubernetes and OpenShift resources. In order to successfully query Kubernetes and OpenShift resources, the daemonset must be given permission to access said resources. This permission is given by a role that is bound to the `ibm-spectrum-scale-core` service account, which is used exclusively by the daemonset.

Table 12. Core pod permissions

Resources	Verbs	API Groups
pods, services	get, list	-
deployments, statefulsets	get, list	apps

Chapter 3. Installation prerequisites

Prior to the installation of IBM Spectrum Scale container native, the following are the prerequisites:

- Red Hat OpenShift Container Platform configuration
  - Compact clusters support
- IBM Cloud container registry
  - IBM Cloud Container Registry (ICR) entitlement
  - Adding IBM Cloud container registry credentials
  - Air gap setup for network restricted Red Hat OpenShift Container Platform clusters

Red Hat OpenShift Container Platform configuration

You must modify the Red Hat OpenShift Container Platform installation for IBM Spectrum Scale container native to operate correctly.

For more information, see Installing in Red Hat OpenShift documentation.

For the instructions that follow, it is assumed that the Red Hat OpenShift Container Platform is already installed.

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 the machine configuration provided drives a rolling update of the OpenShift nodes and could take several minutes to complete. For the new configuration to take effect, the nodes within the pool must be rebooted. On applying the supplied YAML files, you can complete the following tasks:

- **Increase pids_limit**: Increase the `pids_limit` to 4096. Without this change, the GPFS daemon crashes during I/O by running out of PID resources.
- **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.

Applying Machine Config Operator (MCO) Settings

Note:

The Machine Configuration files have not changed between IBM Spectrum Scale container native fixpack releases. Continue using the provided links to retrieve the files.

Apply the following set of MCO settings depending on your OCP version and machine's architecture:

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

  For 4.8:

  ```
  oc apply -f https://raw.githubusercontent.com/IBM/ibm-spectrum-scale-container-native/
  v5.1.4.0/generated/mco/ocp4.8/mco_x86_64.yaml
  ```

  For 4.9:

  ```
  oc apply -f https://raw.githubusercontent.com/IBM/ibm-spectrum-scale-container-native/
  v5.1.4.0/generated/mco/ocp4.9/mco_x86_64.yaml
  ```

For 4.10:

```
oc apply -f https://raw.githubusercontent.com/IBM/ibm-spectrum-scale-container-native/
v5.1.4.0/generated/mco/ocp4.10/mco_x86_64.yaml
```

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

  For 4.8:

  ```
  oc apply -f https://raw.githubusercontent.com/IBM/ibm-spectrum-scale-container-native/
  v5.1.4.0/generated/mco/ocp4.8/mco_ppc64le.yaml
  ```

  For 4.9:

  ```
  oc apply -f https://raw.githubusercontent.com/IBM/ibm-spectrum-scale-container-native/
  v5.1.4.0/generated/mco/ocp4.9/mco_ppc64le.yaml
  ```

  For 4.10:

  ```
  oc apply -f https://raw.githubusercontent.com/IBM/ibm-spectrum-scale-container-native/
  v5.1.4.0/generated/mco/ocp4.10/mco_ppc64le.yaml
  ```

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

  For 4.8:

  ```
  oc apply -f https://raw.githubusercontent.com/IBM/ibm-spectrum-scale-container-native/
  v5.1.4.0/generated/mco/ocp4.8/mco_s390x.yaml
  ```

  For 4.9:

  ```
  oc apply -f https://raw.githubusercontent.com/IBM/ibm-spectrum-scale-container-native/
  v5.1.4.0/generated/mco/ocp4.9/mco_s390x.yaml
  ```

  For 4.10:

  ```
  oc apply -f https://raw.githubusercontent.com/IBM/ibm-spectrum-scale-container-native/
  v5.1.4.0/generated/mco/ocp4.10/mco_s390x.yaml
  ```

Verifying Machine Config Operator (MCO) settings

Complete the following steps:

1. Check the status of the update by entering the following command:

   ```
   oc get MachineConfigPool
   ```

   Note: The status might take a while to display after you enter the command.

2. Verify that the `pids_limit` is increased on the worker nodes by entering the following command:

   ```
   oc get nodes -lnode-role.kubernetes.io/worker= \
   -ojsonpath="{range .items[*]}{.metadata.name}{'\n'}" |\
   xargs -I{} oc debug node/{} -T -- chroot /host crio-status config | grep pids_limit
   ```

   Note: This command runs through all the worker nodes. Use it with discretion if you have a large system.

3. Enter the following command to validate that the `Kernel-devel` package is successfully applied on the Red Hat OpenShiftcontainer worker nodes.

   ```
   oc get nodes -lnode-role.kubernetes.io/worker= \
   -ojsonpath="{range .items[*]}{.metadata.name}{'\n'}" |\
   xargs -I{} oc debug node/{} -T -- chroot /host sh -c "rpm -q kernel-devel"
   ```

4. If running s390x, validate that the machine config has the vmalloc kernel parameter set by entering the following command:

```
oc describe machineconfig | grep vmalloc
```

Note: Perform this step only if you are running Linux on Z.

5. If running s390x, 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 -lnode-role.kubernetes.io/worker= \
-ojsonpath="{range .items[*]}{.metadata.name}{'\n'}" |\
xargs -I{} oc debug node/{} -T -- cat /proc/cmdline
```

Note: Perform this step only if you are running Linux on Z.

You can see `vmalloc=4096G` in the following output:

```
# 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
```

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 compact OpenShift cluster, all applications including the services of the OpenShift master nodes, IBM Spectrum Scale and applications share the same resources. Very high load on one component might impact the availability of other components. In such environments, the resource utilization must be monitored carefully to assure the availability and the performance of all solution components.

You must not run user container workload on the Control Plane. For more information, see Control Plane Components in Kubernetes 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. For more information, see Red Hat OpenShift Container Platform configuration.

You can take the sample mco `yaml` files as a base template that can be modified and applied to your cluster.

Note:

The Machine Configuration files have not changed between IBM Spectrum Scale container native fixpack releases. Continue using the provided links to retrieve the files.

1. Download the correct sample file based on your OCP version and machine architecture and save it as `master_mco.yaml`.

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

     For 4.8:

     ```
     curl https://raw.githubusercontent.com/IBM/ibm-spectrum-scale-container-native/v5.1.4.0/
     generated/mco/ocp4.8/mco_x86_64.yaml > master_mco.yaml
     ```

     For 4.9:

     ```
     curl https://raw.githubusercontent.com/IBM/ibm-spectrum-scale-container-native/v5.1.4.0/
     generated/mco/ocp4.9/mco_x86_64.yaml > master_mco.yaml
     ```

     For 4.10:

     ```
     curl https://raw.githubusercontent.com/IBM/ibm-spectrum-scale-container-native/v5.1.4.0/
     generated/mco/ocp4.10/mco_x86_64.yaml > master_mco.yaml
     ```

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

     For 4.8:

     ```
     curl https://raw.githubusercontent.com/IBM/ibm-spectrum-scale-container-native/v5.1.4.0/
     generated/mco/ocp4.8/mco_ppc64le.yaml > master_mco.yaml
     ```

     For 4.9:

     ```
     curl https://raw.githubusercontent.com/IBM/ibm-spectrum-scale-container-native/v5.1.4.0/
     generated/mco/ocp4.9/mco_ppc64le.yaml > master_mco.yaml
     ```

     For 4.10:

     ```
     curl https://raw.githubusercontent.com/IBM/ibm-spectrum-scale-container-native/v5.1.4.0/
     generated/mco/ocp4.10/mco_ppc64le.yaml > master_mco.yaml
     ```

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

     For 4.8:

     ```
     curl https://raw.githubusercontent.com/IBM/ibm-spectrum-scale-container-native/v5.1.4.0/
     generated/mco/ocp4.8/mco_s390x.yaml > master_mco.yaml
     ```

     For 4.9:

     ```
     curl https://raw.githubusercontent.com/IBM/ibm-spectrum-scale-container-native/v5.1.4.0/
     generated/mco/ocp4.9/mco_s390x.yaml > master_mco.yaml
     ```

     For 4.10:

     ```
     curl https://raw.githubusercontent.com/IBM/ibm-spectrum-scale-container-native/v5.1.4.0/
     generated/mco/ocp4.10/mco_s390x.yaml > master_mco.yaml
     ```

2. Modify the sample file for the `master` role and apply to your cluster:

   ```
   cat master_mco.yaml | sed 's/worker/master/g' | oc apply -f -
   ```

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

   For more information, see "Verifying Machine Config Operator (MCO) settings" on page 18.

4. Remove the `node-role.kubernetes.io/worker: ""` selector from the default `Cluster CR` node selector.

Removing this selector enables the deployment of IBM Spectrum Scale **core** pods on master and worker nodes. For more information, see Node Selectors.

Obtaining a deployment image from IBM Cloud Container Registry

Starting with IBM Spectrum Scale container native 5.1.1.1, the container images have moved from Fix Central to the IBM Cloud Container Registry.

Note:

If your cluster is already configured with IBM Cloud Container Registry, you do not need to create an entitlement key nor create the global pull secret since they already exist there.

- IBM Cloud Container Registry (ICR) entitlement
- Adding IBM Cloud container registry credentials
- "Air gap setup for network restricted Red Hat OpenShift Container Platform clusters (optional) 5.1.4.0" on page 23

IBM Cloud Container Registry (ICR) entitlement

To obtain an entitlement key, complete the following steps:

1. Log in to the IBM container software library with an IBM id and a password that is associated with the entitled software.
2. Click `Get entitlement key` on the left navigation bar.
3. On the `Access your container software` page, click `Copy key` to copy the generated entitlement key.
4. Save the key to a secure location for future use.

Note: Entitlement keys determine whether the IBM Spectrum Scale operator can automatically pull the required IBM Spectrum Scale container native images. During installation, 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. It is therefore important to generate a key from an account that already has entitlement to the desired edition of IBM Spectrum Scale software.

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 Air gap setup for network restricted Red Hat OpenShift Container Platform clusters.

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

   - For using IBM Cloud Container Registry, the credentials are the fixed `cp` user and the generated entitlement key.

     For more information, see IBM Cloud Container Registry (ICR) entitlement.

     ```
     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 generated 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. The following step takes the `authority.json` and include it as a new authority in your `.dockerconfigjson`, stored as a `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": "",
              "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. The updated config is now rolled out to all the nodes in the OpenShift cluster. Nodes are cycled through one at a time and are not schedulable before rebooting. Enter the `watch oc get nodes` command to observe nodes.

   ```
   # ocgetnodes

   NAME                    STATUS                           ROLES     AGE      VERSION
   master0.example.com   NotReady,SchedulingDisabled       master    99d   v1.19.0+43983cd
   master1.example.com   Ready                             master    99d   v1.19.0+43983cd
   master2.example.com   Ready                             master    99d   v1.19.0+43983cd
   worker0.example.com   NotReady,SchedulingDisabled       worker    99d   v1.19.0+43983cd
   worker1.example.com   Ready                             worker    99d   v1.19.0+43983cd
   worker2.example.com   Ready                             worker    99d   v1.19.0+43983cd
   ```

   Note: Red Hat OpenShift Container Platform 4.7 and above versions do not reboot the nodes. For more information, see Updating the global cluster pull secret in Red Hat OpenShift documentation.

6. When the global pull secret is updated, enter the following command to remove the temporary files that were created.

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

Air gap setup for network restricted Red Hat OpenShift Container Platform clusters

Specified instructions per fixpack for installing IBM Spectrum Scale container native in an air gap setup.

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

- "Air gap setup for network restricted Red Hat OpenShift Container Platform clusters (optional) 5.1.4.0" on page 23
- "Air gap setup for network restricted Red Hat OpenShift Container Platform clusters (optional) 5.1.4.1" on page 28

Air gap setup for network restricted Red Hat OpenShift Container Platform clusters (optional) 5.1.4.0

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 (example: registrymirror.yaml) and replace your internal image registry repository with `example.io/subdir`:

```
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 issue while resolving them correctly.

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

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

   The mirror gets rolled out to all nodes in the OpenShift cluster. Nodes are cycled one at a time and are made unschedulable before rebooting.

3. Enter the following command to observe the nodes:

   ```
   watch oc get nodes
   ```

   Note: Red Hat OpenShift Container Platform 4.7 and later do not reboot the nodes.

4. Once all nodes have finished updating and rebooting, verify that the `ImageContentSourcePolicy` is applied by entering 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:aa81c3c1572c4b29dcac048563f0bd6242b649da7c63acc9dbb6612e1f00574c
   cp.icr.io/cp/spectrum/scale/ibm-spectrum-scale-core-
   init@sha256:9319f52cd83d0576fb72c0eb31bfc998039366a3766d5afcf431870423cd9bd5
   cp.icr.io/cp/spectrum/scale/ibm-spectrum-scale-
   pmcollector@sha256:ed565dea825dc16423ebd84a9532fb8498a55b56f75d6a4e9ae060b9c0fdad18
   cp.icr.io/cp/spectrum/scale/ibm-spectrum-scale-
   monitor@sha256:c52ca470b2ef8e7f4020413d10ad82a4c7c52f8988a10b62dbaba4a254c03a5b
   cp.icr.io/cp/spectrum/scale/ibm-spectrum-scale-
   gui@sha256:797701bf382fb1125f71281382564a1ee82f7ebe1fecb42df465d11c86f06234
   cp.icr.io/cp/spectrum/scale/ibm-spectrum-scale-grafana-
   bridge@sha256:6c66b16b228f32a7d755794fe7abb1792ef4796ca189eaa9447a10869bcf216a
   icr.io/cpopen/ibm-spectrum-scale-must-
   gather@sha256:76f85ac3be474a1dea0266e3b391aeee3a3ea7c44d8ffcbea0e715b8e5ce0839
   cp.icr.io/cp/spectrum/scale/ubi-
   minimal@sha256:e83a3146aa8d34dccfb99097aa79a3914327942337890aa6f73911996a80ebb8
   cp.icr.io/cp/spectrum/scale/
   postgres@sha256:3162a6ead070474b27289f09eac4c865e75f93847a2d7098f718ee5a721637c4
   icr.io/cpopen/ibm-spectrum-scale-csi-
   operator@sha256:b62ad588fbc338090f405740fd8bceaf59b3267cdb532c60ae965643ccae4d5d

   cp.icr.io/cp/spectrum/scale/csi/ibm-spectrum-scale-csi-
   driver@sha256:1efa61d65ee33c574cd2826f1bc6194631b1905996eddaa2ad51b66073e0c409
   cp.icr.io/cp/spectrum/scale/csi/csi-
   snapshotter@sha256:89e900a160a986a1a7a4eba7f5259e510398fa87ca9b8a729e7dec59e04c7709
   cp.icr.io/cp/spectrum/scale/csi/csi-
   provisioner@sha256:122bfb8c1edabb3c0edd63f06523e6940d958d19b3957dc7b1d6f81e9f1f6119
   cp.icr.io/cp/spectrum/scale/csi/csi-node-driver-
   registrar@sha256:4fd21f36075b44d1a423dfb262ad79202ce54e95f5cbc4622a6c1c38ab287ad6
   cp.icr.io/cp/spectrum/scale/csi/csi-
   attacher@sha256:8b9c313c05f54fb04f8d430896f5f5904b6cb157df261501b29adc04d2b2dc7b
   cp.icr.io/cp/spectrum/scale/csi/
   livenessprobe@sha256:933940f13b3ea0abc62e656c1aa5c5b47c04b15d71250413a6b821bd0c58b94e
   cp.icr.io/cp/spectrum/scale/csi/csi-
   resizer@sha256:9ebbf9f023e7b41ccee3d52afe39a89e3ddacdbb69269d583abfc25847cfd9e4
   ```

   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@sha256:4635ab99adee4c2645a952805b9a4494390368fbc3a0badd4f4cef552b05147a
   ```

   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@sha256:84087b10860da922ef1ba481ffe747a5192fd1a96696f5fb8e7188a1006d0bc9
   ```

   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@sha256:797701bf382fb1125f71281382564a1ee82f7ebe1fecb42df465d11c86f06234
   docker://example.io/subdir/ibm-spectrum-scale-
   gui@sha256:797701bf382fb1125f71281382564a1ee82f7ebe1fecb42df465d11c86f06234
   ```

   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@sha256:4635ab99adee4c2645a952805b9a4494390368fbc3a0badd4f4cef552b05147a
   docker://example.io/subdir/data-access/ibm-spectrum-scale-
   daemon@sha256:4635ab99adee4c2645a952805b9a4494390368fbc3a0badd4f4cef552b05147a
   ```

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@sha256:84087b10860da922ef1ba481ffe747a5192fd1a96696f5fb8e7188a1006d0bc9
docker://example.io/subdir/data-management/ibm-spectrum-scale-
daemon@sha256:84087b10860da922ef1ba481ffe747a5192fd1a96696f5fb8e7188a1006d0bc9
```

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@sha256:797701bf382fb1125f71281382564a1ee82f7ebe1fecb42df465d11c86f06234
   ```

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. Use the contents of the `temp_config.json` file, and apply the updated configuration 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, 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 -
     ```

   The updated configuration is now rolled out to all nodes in the OpenShift cluster. Nodes are cycled one at a time and are made unavailable for scheduling before rebooting.

5. Enter the `watch oc get nodes` command to observe the nodes.

   ```
   # ocgetnodes

   NAME                    STATUS                           ROLES     AGE      VERSION
   master0.example.com   NotReady,SchedulingDisabled       master    99d   v1.19.0+43983cd
   master1.example.com   Ready                             master    99d   v1.19.0+43983cd
   master2.example.com   Ready                             master    99d   v1.19.0+43983cd
   worker0.example.com   NotReady,SchedulingDisabled       worker    99d   v1.19.0+43983cd
   worker1.example.com   Ready                             worker    99d   v1.19.0+43983cd
   worker2.example.com   Ready                             worker    99d   v1.19.0+43983cd
   ```

   Note: Red Hat OpenShiftContainer Platform 4.7 and above versions do not reboot the nodes. For more information, see Updating the global cluster pull secret in Red Hat OpenShift documentation.

6. When the global pull secret is updated, remove the temporary files that were created.

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

Air gap setup for network restricted Red Hat OpenShift Container Platform clusters (optional) 5.1.4.1

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 (example: registrymirror.yaml) and replace your internal image registry repository with `example.io/subdir`:

   ```
   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 issue while resolving them correctly.

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

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

   The mirror gets rolled out to all nodes in the OpenShift cluster. Nodes are cycled one at a time and are made unschedulable before rebooting.

3. Enter the following command to observe the nodes:

   ```
   watch oc get nodes
   ```

   Note: Red Hat OpenShift Container Platform 4.7 and later do not reboot the nodes.

4. Once all nodes have finished updating and rebooting, verify that the `ImageContentSourcePolicy` is applied by entering 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@sha256:3f6e90c08d232338bc21127805a285c7debdbf5cec512ef84de9899a700c438e
   cp.icr.io/cp/spectrum/scale/ibm-spectrum-scale-core-
   ```

```
init@sha256:4c35ecba95011443362cdc7cf07acba14eea22c48d5938a19a2027789130dfeb
cp.icr.io/cp/spectrum/scale/ibm-spectrum-scale-
pmcollector@sha256:d859df59dc1937ad525562be97e811ddc05526bf2597717b99df11f5053f2dd1
cp.icr.io/cp/spectrum/scale/ibm-spectrum-scale-
monitor@sha256:88844b0f29e32b54d869e831538b0a6063e3a5a5a0b284691da1c63ec1336d80
cp.icr.io/cp/spectrum/scale/ibm-spectrum-scale-
gui@sha256:522af21446583c5a9bf2797d0e952149379264881472678ffe6706b647313430
cp.icr.io/cp/spectrum/scale/ibm-spectrum-scale-grafana-
bridge@sha256:6c66b16b228f32a7d755794fe7abb1792ef4796ca189eaa9447a10869bcf216a
icr.io/cpopen/ibm-spectrum-scale-must-
gather@sha256:76f85ac3be474a1dea0266e3b391aeee3a3ea7c44d8ffcbea0e715b8e5ce0839
cp.icr.io/cp/spectrum/scale/ubi-
minimal@sha256:e83a3146aa8d34dccfb99097aa79a3914327942337890aa6f73911996a80ebb8
cp.icr.io/cp/spectrum/scale/
postgres@sha256:3162a6ead070474b27289f09eac4c865e75f93847a2d7098f718ee5a721637c4
icr.io/cpopen/ibm-spectrum-scale-csi-
operator@sha256:9ac13e80b2a9299807f4249b23967fc2517a4aa34528d1ec524c5c8eb2691fd0
cp.icr.io/cp/spectrum/scale/csi/ibm-spectrum-scale-csi-
driver@sha256:8340faaee3bd3a4ac19047dc17fa3cf01c7350d49a783e1729468d7771f5d5be
cp.icr.io/cp/spectrum/scale/csi/csi-
snapshotter@sha256:89e900a160a986a1a7a4eba7f5259e510398fa87ca9b8a729e7dec59e04c7709
cp.icr.io/cp/spectrum/scale/csi/csi-
provisioner@sha256:122bfb8c1edabb3c0edd63f06523e6940d958d19b3957dc7b1d6f81e9f1f6119
cp.icr.io/cp/spectrum/scale/csi/csi-node-driver-
registrar@sha256:4fd21f36075b44d1a423dfb262ad79202ce54e95f5cbc4622a6c1c38ab287ad6
cp.icr.io/cp/spectrum/scale/csi/csi-
attacher@sha256:8b9c313c05f54fb04f8d430896f5f5904b6cb157df261501b29adc04d2b2dc7b
cp.icr.io/cp/spectrum/scale/csi/
livenessprobe@sha256:933940f13b3ea0abc62e656c1aa5c5b47c04b15d71250413a6b821bd0c58b94e
cp.icr.io/cp/spectrum/scale/csi/csi-
resizer@sha256:9ebbf9f023e7b41ccee3d52afe39a89e3ddacdbb69269d583abfc25847cfd9e4
```

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:c1f4b9be01d8083f43634038daf3e3a3c855042687a83e36c39d0afe144effc3
```

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:ca9d2b844b27583e67c4101b2c74d5959b3f5c310b2a1a995e8df056c2e92fd7
```

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@sha256:522af21446583c5a9bf2797d0e952149379264881472678ffe6706b647313430
docker://example.io/subdir/ibm-spectrum-scale-
gui@sha256:522af21446583c5a9bf2797d0e952149379264881472678ffe6706b647313430
```

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:c1f4b9be01d8083f43634038daf3e3a3c855042687a83e36c39d0afe144effc3
docker://example.io/subdir/data-access/ibm-spectrum-scale-
daemon@sha256:c1f4b9be01d8083f43634038daf3e3a3c855042687a83e36c39d0afe144effc3
```

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:ca9d2b844b27583e67c4101b2c74d5959b3f5c310b2a1a995e8df056c2e92fd7
```

```
docker://example.io/subdir/data-management/ibm-spectrum-scale-
daemon@sha256:ca9d2b844b27583e67c4101b2c74d5959b3f5c310b2a1a995e8df056c2e92fd7
```

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@sha256:90a556ed3e13926d60e9e140903a3c2e4fe9463e4806833c12f9eff2dba4ae93
```

5. Verify that the image is pulled.

```
# podman images | grep cp.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. Use the contents of the `temp_config.json` file, and apply the updated configuration 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, 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 -
     ```

   The updated configuration is now rolled out to all nodes in the OpenShift cluster. Nodes are cycled one at a time and are made unavailable for scheduling before rebooting.

5. Enter the `watch oc get nodes` command to observe the nodes.

   ```
   # ocgetnodes
   ```

```
NAME                    STATUS                           ROLES     AGE      VERSION
master0.example.com   NotReady,SchedulingDisabled       master    99d   v1.19.0+43983cd
master1.example.com   Ready                             master    99d   v1.19.0+43983cd
master2.example.com   Ready                             master    99d   v1.19.0+43983cd
worker0.example.com   NotReady,SchedulingDisabled       worker    99d   v1.19.0+43983cd
worker1.example.com   Ready                             worker    99d   v1.19.0+43983cd
worker2.example.com   Ready                             worker    99d   v1.19.0+43983cd
```

Note: Red Hat OpenShiftContainer Platform 4.7 and above versions do not reboot the nodes. For more information, see Updating the global cluster pull secret in Red Hat OpenShift documentation.

6. When the global pull secret is updated, remove the temporary files that were created.

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

Chapter 4. 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.

- Node labels and annotations
- Firewall recommendations
- IBM Spectrum Scale storage cluster configuration
- Deploy the operator
- Configuring the IBM Spectrum Scale container native cluster custom resources
  - Cluster
  - Callhome
  - Filesystems
  - Encryption
- Creating the IBM Spectrum Scale container native cluster
- Creating secrets for storage cluster GUI
- Configuring Certificate Authority (CA) certificates
- Verifying the IBM Spectrum Scale container native cluster
- Status and events

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.

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 accross 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.

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

For more information about IBM Spectrum Scale quorum, 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.

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 cannot be changed after creation of the IBM Spectrum Scale container native cluster.

Firewall recommendations

Ensure that ports 12345, 1191, 443, ping, and the ephemeral port ranges are open on a storage cluster and on any network switches between a storage and container native cluster. Otherwise, the container native cluster cannot remotely mount a file system from the storage cluster.

Table 13. Recommended port numbers

Port number	Protocol	Service name
12345	TCP	Config, GPFS
1191	TCP	Config, GPFS

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

Configure cluster profile with tscCmdAllowRemoteConnections

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 the 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 14. Configure tscCmdAllowRemoteConnections

Storage cluster version	IBM Spectrum Scalecontainer native version	tscCmdAllowRemoteConnections
< 5.1.3	< 5.1.3.0	yes
>= 5.1.3	< 5.1.3.0	yes
>= 5.1.3	>= 5.1.3.0	no

- To change this value on the storage cluster, enter 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: 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.

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

IBM Spectrum Scale storage cluster configuration

Some additional tasks need to be performed on the IBM Spectrum Scale storage cluster in order for the container native cluster operator to configure the remote mount of the storage cluster file systems.

Creating Operator User and Group

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

1. To verify whether the IBM Spectrum Scale GUI user group `ContainerOperator` exists, enter the following command:

   ```
   /usr/lpp/mmfs/gui/cli/lsusergrp ContainerOperator
   ```

2. To create the `ContainerOperator` GUI user group if it does not exist, enter the following command:

   ```
   /usr/lpp/mmfs/gui/cli/mkusergrp ContainerOperator --role containeroperator
   ```

3. To verify whether an IBM Spectrum Scale GUI user exists within the `ContainerOperator` group, enter the following command:

   ```
   /usr/lpp/mmfs/gui/cli/lsuser | grep ContainerOperator
   ```

4. To create a GUI user for the `ContainerOperator` group, 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, then enter the following command to create the user with a password that never expires:

   ```
   /usr/lpp/mmfs/gui/cli/mkuser cnsa_storage_gui_user -p cnsa_storage_gui_password -g 
   ContainerOperator -e 1
   ```

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

Container Storage Interface (CSI) configuration

Complete the following steps on a *storage cluster* to ensure the IBM Spectrum Scale CSI driver is deployed successfully.

1. Create an IBM Spectrum Scale user group `CsiAdmin` by entering the following command:

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

2. Create an IBM Spectrum Scale user in the `CsiAdmin` group. This user must be used on IBM Spectrum Scale Container Storage Interface driver configuration. Enter this command on the GUI node to create the user.

   ```
   /usr/lpp/mmfs/gui/cli/mkuser csi-storage-gui-user -p  csi-storage-gui-password -g CsiAdmin
   ```

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

   ```
   $ mmlsfs fs1 --perfileset-quota
   flag                value                    description
   ------------------- ------------------------ -------------------------------
   --perfileset-quota No                       Per-fileset quota enforcement
   ```

4. Enter the following command to enable the Quota in the file systems that is used by the IBM Spectrum Scale Container Storage Interface driver.

   ```
   mmchfs fs1 -Q yes
   ```

5. Enter the following command to verify that the quota is enabled.

```
$ mmlsfs fs1 -Q

flag                value                    description
------------------- ------------------------ -----------------------------------
-Q                 user;group;fileset       Quotas accounting enabled
                     user;group;fileset       Quotas enforced
                     none                     Default quotas enabled
```

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

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

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

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

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

   ```
   mmchfs fs1 --filesetdf
   ```

Deploy the operator

Deploy the IBM Spectrum Scale container native operator by entering the following command:

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

For IBM Spectrum Scale container native 5.1.4.1

```
oc apply -f https://raw.githubusercontent.com/IBM/ibm-spectrum-scale-container-native/v5.1.4.1/
generated/installer/ibm-spectrum-scale-operator.yaml
```

For IBM Spectrum Scale container native 5.1.4.0

```
oc apply -f https://raw.githubusercontent.com/IBM/ibm-spectrum-scale-container-native/v5.1.4.0/
generated/installer/ibm-spectrum-scale-operator.yaml
```

Validate that the operator pods are running in the following 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
  ```

Configuring the IBM Spectrum Scale container native cluster custom resources

Before deploying a cluster, you need to make changes to the sample `scale_v1beta1_cluster_cr.yaml` file.

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

Save the sample YAML file from the GitHub by entering the following command:

For IBM Spectrum Scale container native 5.1.4.1

```
curl -fs https://raw.githubusercontent.com/IBM/ibm-spectrum-scale-container-native/v5.1.4.1/
generated/scale_v1beta1_cluster_cr.yaml > scale_v1beta1_cluster_cr.yaml || echo "Failed to curl
Cluster CR"
```

For IBM Spectrum Scale container native 5.1.4.0

```
 curl -fs https://raw.githubusercontent.com/IBM/ibm-spectrum-scale-container-native/v5.1.4.0/
generated/scale_v1beta1_cluster_cr.yaml > scale_v1beta1_cluster_cr.yaml || echo "Failed to curl
Cluster CR"
```

This file is used to set configuration. When deployed, it initiates the IBM Spectrum Scale cluster creation.

The table below describes the Custom Resource Definitions (CRDs) managed by the IBM Spectrum Scale container native operator:

Table 15. 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.
`filesystem`	`fs`	Configures remote mounted filesystems for the container native cluster.
`remoteclusters`	`remotegpfs`	Provide configuration to the remote cluster and establishes the secure authorizations. For more information, see Filesystem section.
`encryptionconfig`	`ec`	Allows users to configure encryption functionality.

The following sections guides through this process:

- Cluster
- Callhome
- Filesystems
- Encryption

Cluster

The sample `Cluster` custom resource can be found under `kind: Cluster` in the `scale_v1beta1_cluster_cr.yaml` file.

For more information, see Configuring the IBM Spectrum Scale container native cluster custom resources. Once deployed, enter the `oc edit cluster` command to modify the properties.

Cluster spec

The following table describe the properties for `Cluster`:

Table 16. Cluster property and description

Property	Required	Default	Description
license	Yes	None	The license must be accepted by the end user that provides a way to specify the IBM Spectrum ScaleEdition.
license.accept	Yes	None	Read the license and specify `true` to accept or `false` to not accept.
license.license	Yes	None	It specifies the IBM Spectrum Scale edition, `data-access` or `data-management`.
daemon	Internal CR	N/A	It tells the operator how to configure the gpfs daemons.
grafanaBridge	Internal CR	Disabled	It tells the operator how to configure Grafana Bridge.
gui	Internal CR	N/A	It tells the operator how to configure the GUIs.
pmcollector	Internal CR	N/A	It tells the operator how to configure the pmcollectors.

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 link 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.

Node selectors

The daemon.nodeSelector section allows you to configure a nodeSelector to determine where IBM Spectrum Scale pods can be deployed. The default location in the sample is to deploy core pods to Kubernetes worker roles:

```
apiVersion: scale.spectrum.ibm.com/v1beta1
kind: Cluster
spec:
  ...
  daemon:
    nodeSelector:
      node-role.kubernetes.io/worker: ""
```

You may configure multiple node selector values by adding labels to the nodeSelector list. 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 following example, the Operator deploys IBM Spectrum Scale pods on nodes with both the worker label and "scale" component label.

```
apiVersion: scale.spectrum.ibm.com/v1beta1
kind: Cluster
spec:
  ...
  daemon:
    nodeSelector:
      node-role.kubernetes.io/worker: ""
      app.kubernetes.io/component: "scale"
```

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

Host aliases

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

The daemon.hostAliases section allows for entries to be created by kubernetes into the IBM Spectrum Scale **core** pod's /etc/hosts file.

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 hostAliases section does not handle creating entries in /etc/hosts in any pods except 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.

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.

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`:

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

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

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.

Enter the following command 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.

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

The sample `Callhome` custom resource can be found under `kind: Callhome` in the `scale_v1beta1_cluster_cr.yaml` file.

For more information, see Configuring the IBM Spectrum Scale container native cluster custom resources. Fill out the following details to enable call home functionality.

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

Note: You must always configure call home. If you choose not to configure call home, delete or comment out the call home section in the custom resource.

Call home can be enabled, modified, or disabled at any time. Enter the `oc explain callhome` command to view more details.

Callhome spec

The following table describes the properties for `Callhome`:

Table 17. 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.

Table 17. Callhome property and description (continued)

Property	Required	Default	Description
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.

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
```

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.

Filesystems

Remote filesystem

To configure a remote mounted file system for a container native cluster, you must create a `Filesystem` custom resource and a `RemoteCluster` custom resource.

Filesystem

Filesystem spec

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

The following table describe the properties for `Filesystem`:

Table 18. Filesystem property and description

Property	Required	Default	Description
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 on the remote cluster to mount.
remote.cluster	Yes, if `remote` is specified	None	It is the name of the Remote Cluster custom resource.

The sample Filesystem custom resource can be found under kind: Filesystem in scale_v1beta1_cluster_cr.yaml file. For more information, see Configuring the IBM Spectrum Scale container native cluster custom resources.

The name that you set for the Filesystem CR under metadata.name is the name of the custom resource and also becomes the name of the remote file system mount point at /mnt/<metadata.name>. In the sample, the name of the local file system is remote-sample and is mounted at /mnt/remote-sample. You can define more than one Filesystem CR.

Set the details under the remote section to reflect the storage cluster file system being mounted as fs and the name of the RemoteCluster created as a cluster.

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

Limitations

Deleting a Filesystem custom resource does not un-mount or delete the file system configuration from an 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.

RemoteCluster

The sample RemoteCluster custom resource can be found under kind: RemoteCluster in scale_v1beta1_cluster_cr.yaml file.

RemoteCluster spec

The following table describe the properties for RemoteCluster:

Table 19. RemoteCluster field and description

Field	Required	Default	Description
metadata.name	Yes	None	The name of the CR, that is used to identify the remote storage cluster in the `filesystem` CR.
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.

The name that you set for the `RemoteCluster` CR under `metadata.name` identifies the remote storage cluster you want to create an authentication to. This name is used as a reference in the `Filesystem` CR `remote.cluster` to identify the remote storage cluster serving the file system. You can define more than one RemoteCluster.

To create RemoteCluster spec, complete the following steps:

1. Validate that a secret for the storage cluster is created.

   For more information, see Creating secrets for storage cluster GUI.
2. Set the GUI details to match your remote storage GUI in the `gui` section:

```
apiVersion: scale.spectrum.ibm.com/v1beta1
kind: RemoteCluster
...
metadata:
name: remotecluster-sample
spec:
  contactNodes:
  - storagecluster1node1
  - storagecluster1node2
  gui:
      cacert: cacert-storage-cluster-1
      host: guihost.example.com
      insecureSkipVerify: false
      secretName: cnsa-remote-mount-storage-cluster-1
```

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.

EncryptionConfig

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)

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.

EncryptionConfig spec

The following table describes the properties for `EncryptionConfig`:

Table 20. Encryption property and description

Property	Required	Default	Description
metadata.name	Yes	None	The name of the CR.
server	Yes	None	The key server hostname or IP in which encryption keys are stored.

Table 20. Encryption property and description (continued)

Property	Required	Default	Description
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.

The sample `EncryptionConfig` custom resource file can be downloaded by entering the following command:

```
curl https://raw.githubusercontent.com/IBM/ibm-spectrum-scale-container-native/v5.1.4.0/
generated/scale_v1beta1_encryptionconfig_sample.yaml \
> scale_v1beta1_encryptionconfig_cr.yaml
```

This file is used to set configuration. When deployed, it initiates the configuration of key-server in a cluster.

Here, `metadata.name` is the name of the CR with the configuration parameters. Under `spec`:

- `server` is the hostname of the key-server in which a key is stored.
- `tenant` is the name of the tenant that contains the encryption keys. This has to be the same tenant name that is used to store the encryption keys of a remote storage file system.
- `remoteRKM` is the RKM ID that is used in the storage cluster to register a client with the tenant. In case the key-server uses a CA signed server certificate, `cacert` is used to pass the CA certificates to the client for validation of the endpoint certificate.

```
apiVersion: scale.spectrum.ibm.com/v1beta1
kind: EncryptionConfig
metadata:
  ...
  name: encryption-config-sample
  ...
```

```
spec:
  cacert: sample-ca-cert
  client: sampleClient
  port: 9443
  remoteRKM: sampleRKM
  secret: keyserver-credentials
  server: keyserver.example.com
  tenant: sampleTenant
```

Additional parameters defined by `filesystems` object are used to specify an encryption algorithm to be used to decrypt the local file system.

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

Configure EncryptionConfig custom resource

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 Encryption

Download and save a copy of the sample `scale_v1beta1_encryptionconfig_cr.yaml` from GitHub to make changes specific to your installation.

```
curl -fs https://raw.githubusercontent.com/IBM/ibm-spectrum-scale-container-native/v5.1.4.0/
generated/scale_v1beta1_encryptionconfig_cr.yaml > scale_v1beta1_encryptionconfig_cr.yaml ||
echo "Failed to curl EncryptionConfig CR"
```

Set the details in the `EncryptionConfig` custom resource:

- Replace `keyserver.example.com` with your keyserver hostname
- 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

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

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.

Creating an IBM Spectrum Scale container native cluster

Deploy a cluster by applying the custom resource modified in the *Configuring the IBM Spectrum Scale container native cluster custom resources* procedure.

For more information, see Configuring the IBM Spectrum Scale container native cluster custom resources.

Once the custom resources file is applied, IBM Spectrum Scale Operator creates all the pods that make up an IBM Spectrum Scale container native cluster. Enter the following command to apply the YAML file:

```
oc apply -f ./scale_v1beta1_cluster_cr.yaml
```

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
```

Creating secrets for the storage cluster GUI

Create a secret on the Red Hat OpenShift cluster to store a username and a password for an IBM Spectrum Scale Storage cluster GUI user and password.

This secret is used by the Operator to communicate with the storage cluster while configuring for a remote mount.

Two new secrets must be added for each storage cluster being configured for remote mount on the IBM Spectrum Scale container native cluster.

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

   The username and password specified in this topic must match the GUI user that was created on the storage cluster in the *Creating Operator User and Group* procedure. For more information, see Creating Operator User and Group.

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

   Note: The name of this secret must match the secretName field defined for the RemoteCluster CR. For more information, see Filesystems.

   ```
   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 storage cluster CsiAdmin GUI user.

   The username and password specified in this topic must match the GUI user that was created on the storage cluster of the *Container Storage Interface (CSI) configuration* procedure. For more information, see Container Storage Interface (CSI) configuration.

   Note: The name of this secret should match the csiSecretName field defined for the RemoteCluster CR. For more information, see Filesystems.

3. To create the storage cluster GUI user 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
   ```

4. To 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
   ```

Configuring Certificate Authority (CA) certificates for storage cluster

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 Filesystem Custom Resource. For more information, see Filesystems.

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 Filesystem Custom Resource. For more information, see Filesystems.

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
```

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 77.

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 resulting cluster contains two `gui` pods, two `pmcollector` pods, and one core pod per node that matches the specified `nodeSelector`.

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

   - 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
     ```

     Output:

     ```
     GPFS cluster information
           ========================
           GPFS cluster name:         ibm-spectrum-scale.ocp4.example.com
           GPFS cluster id:           835278197609441888
           GPFS UID domain:           ibm-spectrum-scale.ocp4.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
           --------------------------------------------------------------------
              1   worker0           172.29.0.145  worker0          quorum-manager-perfmon
              2   worker1           172.29.0.146  worker1          quorum-manager-perfmon
              3   worker3           172.29.0.148  worker3          quorum-manager-perfmon
     ```

- 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
```

Example output:

```
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) Check the status of the `RemoteCluster` to ensure that the Status `Ready` is `True`.

   - Get a list of the remote clusters.

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

   - Describe the remote cluster to check its status.

   ```
   oc describe remotecluster remotecluster-sample -n ibm-spectrum-scale
   ```

   b) If the storage cluster authentication was created successfully, you should see `Status` similar to the sample shown:

```
# oc get remoteclusters remotecluster-sample -n ibm-spectrum-scale
  ...
  ...
  Status:
  Conditions:
      ...
      Message:               The remote cluster has been configured successfully.
      Reason:                AuthCreated
      Status:                True
      Type:                  Ready
   Events:
   Type     Reason            Age                  From           Message
   ----     ------            ----                 ----           -------
   ...
   ...
   Normal   Created           66s                  RemoteCluster  The remote cluster has 
been configured successfully.
```

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

   a) Check the status of the `Filesystem` to ensure that the Status`Success` is `True`.

   - Get a list of the file systems:

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

   - Describe the sample file systems to check status:

   ```
   oc describe filesystems remote-sample -n ibm-spectrum-scale
   ```

   b) If the file system was created successfully, you should see a `Status` similar to the sample shown:

```
 Status:
   Conditions:
     ...
     Message:               The remote filesystem has been created and mounted.
     Reason:                FilesystemEstablished
     Status:                True
     Type:                  Success
   Events:
   Type     Reason   Age                 From        Message
   ----     ------   ----                ----        -------
   Warning  Failed   16m (x20 over 28m)  Filesystem  Unable to register storage cluster 
on client cluster.
   Warning  Failed   15m                 Filesystem  Unable to create remote filesystem
```

```
on client cluster.
   Normal   Created  14m (x4 over 15m)   Filesystem  Attempting to mount filesystem on:
[worker0 worker1]
   Normal   Created  14m                 Filesystem  Attempting to mount filesystem on:
[worker1 worker0]
   Normal   Created  13m (x2 over 13m)   Filesystem  Attempting to mount filesystem on:
[worker0]
   Normal   Created  8m8s                Filesystem  The filesystem has been created and
mounted.
```

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                   ibm-spectrum-scale.ocp4-c4.example.com
      172.29.0.146    worker1                   ibm-spectrum-scale.ocp4-c4.example.com
      172.29.0.145    worker0                   ibm-spectrum-scale.ocp4-c4.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 56.

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 -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 -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 5. Upgrading IBM Spectrum Scale container native

Before upgrading IBM Spectrum Scale container native to a new version, refer the supported upgrade paths.

Supported upgrade paths

The following table lists the supported upgrade paths for IBM Spectrum Scale container native:

Table 21. Supported upgrade paths

Upgrade from	Upgrade to 5.1.1.3	Upgrade to 5.1.1.4	Upgrade to 5.1.2.1	Upgrade to 5.1.3.x	Upgrade to 5.1.4.x
5.1.3.x	--	--	--	--	Yes
5.1.2.1	--	--	--	Yes	Yes
5.1.1.4	--	--	Yes	Yes	No
5.1.1.3	--	Yes	Yes	Yes	No
5.1.1.1	[1] Yes	Yes	No	No	No

[1] Do not upgrade IBM Spectrum Scale container native 5.1.1.1 to 5.1.1.3 if your configuration includes 2 IBM Spectrum Scale container native clusters remote mounting the same remote cluster file system. Upgrading causes one of the IBM Spectrum Scale container native clusters to lose access to the remote cluster. Instead, directly upgrade to IBM Spectrum Scale container native 5.1.1.4.

Upgrading IBM Spectrum Scale container native

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

- "Upgrade IBM Spectrum Scale container native to 5.1.4.x" on page 60

Configure cluster profile with tscCmdAllowRemoteConnections

If you are upgrading to 5.1.3.x from any previous version, review the Table 14 on page 37 to determine the `tscCmdAllowRemoteConnections` configuration for your deployment.

Determine storage cluster version for IBM Spectrum Scale container native 5.1.4.x and IBM Spectrum Scale CSI 2.6.x

To ensure the compatibility for desired features of IBM Spectrum Scale container native 5.1.4.x and IBM Spectrum Scale CSI 2.6.x, determine the required storage cluster version. For more information, see Table 1 in Hardware and Software Requirements of IBM Spectrum Scale CSI documentation.

Post upgrade tasks

After you have upgraded the IBM Spectrum Scale container native cluster:

Note: Ensure that all clusters are at the desired level before proceeding with the next steps.

This includes the IBM Spectrum Scale container native cluster, the storage cluster, and any other clusters that are mounting the storage cluster. The next two steps enable new functionality of the installed release, and in doing so, it locks the new level in place. For more information, see File system format changes between versions of IBM Spectrum Scale in IBM Spectrum Scale documentation.

- To complete the upgrade, the release level must be updated after installation.

  ```
  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 -- mmchconfig release=LATEST
  ```

- On the storage cluster, update the file system format.

  Note: If the storage cluster is being mounted by other clusters at down level versions of code, this step causes these clusters to be unable to mount the storage cluster.

  ```
  mmchfs <device> -V full
  ```

Upgrade IBM Spectrum Scale container native to 5.1.4.x

To upgrade IBM Spectrum Scale container native to 5.1.4.x, complete the following steps:

Note: It is recommended to upgrade to the latest available fixpack.

1. Delete the old security context constraint.

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

2. Apply the new manifests.

   ```
   oc apply -f https://raw.githubusercontent.com/IBM/ibm-spectrum-scale-container-native/
   v5.1.4.0/generated/installer/ibm-spectrum-scale-operator.yaml
   ```

Chapter 6. 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" on page 61
- "Managed CSI fields" on page 62
- "Setting primary file set" on page 62

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.

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 22. 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 7. Using IBM Spectrum Scale GUI

You can refer to the mapping of OpenShift users to IBM Spectrum Scale GUI user groups for accessing the IBM Spectrum Scale GUI.

- "IBM Spectrum Scale container native GUI" on page 65

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

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

Table 23. 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
Kubeadmin	Maintenance	Yes	Yes	Yes	Yes
View	Monitor	Yes	Yes	No	Yes

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

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

Accessing the IBM Spectrum Scale GUI

Users created on the Openshift Container Platform (OCP) can log in to the IBM Spectrum Scale container native GUI through single-sign-on (SSO) by using the OAuth implementation.

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.
3. Authenticate by using your OCP user credentials.

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

Chapter 8. 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. Enter the following command to scale the number of IBM Spectrum Scale container native operators to 0.

   ```
   oc edit deploy -n ibm-spectrum-scale-operator
   ```

   Set number of replicas to 0:

   ```
   ...
      spec:
        progressDeadlineSeconds: 600
        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
   ```

Upgrading Red Hat OpenShift Container Platform

Preparation

During Red Hat OpenShift upgrade, the Machine Config Operator (MCO) reboots master and worker nodes to apply the new configuration across a cluster. It cordons the number of nodes that is specified by the `maxUnavailable` field on the machine configuration pool and marks them as unavailable. By default, this value is set to 1. It then applies the new configuration and reboots each node.

As the nodes are updated by the Red Hat OpenShift Container Platform, the IBM Spectrum Scale pods are restarted and returned to a schedulable state.

The MCO has no awareness of the state of an IBM Spectrum Scale cluster. Therefore, there is a risk that the rolling reboot driven by the MCO can bring IBM Spectrum Scale temporarily offline by taking down too many quorum nodes.

This risk can be mitigated in the following ways:

1. Make sure that the MachineConfigPools are configured so that only one node is rebooted at a time.

2. Larger clusters are less vulnerable, as quorum nodes are more widely distributed across the cluster. For more information, see “Labels and annotations” on page 35.
3. The 3 nodes case is considered critical because if the MCO does not wait for a node to fully come back online, IBM Spectrum Scale has the potential to lose a quorum.

Note:

If you encounter any problems during Red Hat OpenShift upgrade, open a support case by gathering cluster information. The details help in providing debugging information to Red Hat OpenShift support on your cluster. For more information, see Gathering data about your cluster in Red Hat OpenShift documentation.

Optional steps to protect from workload failures

To ensure that no workload is affected, stop the workload before completing the following steps:

1. Shut down the IBM Spectrum Scale cluster. For more information, see “Shutting down a cluster” on page 67.
2. Configure the worker nodes. For more information, see “Red Hat OpenShift Container Platform configuration” on page 17.

   Note: If machine configurations are applied to your Red Hat OpenShift Container Platform cluster in a previous IBM Spectrum Scale container native deployment, do not reapply the same machine configuration again. Reapplying an existing Machine Configuration could cause reboots across the worker nodes.
3. Start the IBM Spectrum Scale cluster by following the steps for starting the cluster. For more information, see “Starting the cluster after shutdown” on page 68.

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 the number of operator pods back to 1.

```
oc edit deploy -n ibm-spectrum-scale-operator
```

Set number of replicas to 1:

```
...
   spec:
     progressDeadlineSeconds: 600
     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, you need to add it to an existing cluster and configure CSI on it.

When a new node with labels that matches the existing cluster's node selector is added, a pod is created on the new node. The new pod is up and running within a few minutes. For more information, see “Labels and annotations” on page 35.

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 is ready to be enabled on this node.

Configuring CSI on new nodes

Note: CSI must only be configured on new nodes after they are finished arbitrating and in Active state. Applying the CSI node label before nodes are in an active state can cause unexpected behavior.

For CSI to recognize the newly added node, apply the label to the node:

```
oc label node <node-name> scale=true
```

The newly added node can now be used for running applications.

Chapter 9. Cleaning up the container native cluster

To safely remove the pods or perform other maintenance actions, the IBM Spectrum Scale container native cluster needs to be manually shut down prior to performing these operations. The following procedures outline the steps to complete these actions and validate that it is safe to shut down the cluster.

- "Deleting a cluster" on page 71
- "Removing applications" on page 71
- "Custom Resource" on page 71
  - "Filesystems" on page 71
  - "Remote Clusters" on page 72
- "Cleaning up IBM Spectrum Scale operator" on page 73
- "Cleaning up the worker nodes" on page 73
- "Cleaning up on a storage cluster" on page 74

Deleting a cluster

When deleting an entire cluster, all applications and the IBM Spectrum Scale Container Storage Interface driver must be unloaded prior to the unmount and shutdown steps.

Removing applications

Complete the following steps:

Note: Ensure that you are in the project for IBM Spectrum Scale Container Storage Interface (CSI) driver.

1. Enter the following command to query the PVC to identify the applications that are active.

```
oc describe <csi pvc>
```

2. Enter the following command to remove all the applications. This requires the node to be drained of all data.

```
oc delete <application deployment or daemonSet from csi pvc describe output>
```

Custom Resource

There can be situations when you need to change the custom resource definitions but not clean up the whole container native cluster. The following sections describe how to clean up the IBM Spectrum Scale artifacts when only deleting custom resource definitions.

- "Filesystems" on page 71
- "Remote Clusters" on page 72

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 sample `Filesystem` is used:

```
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
     ```

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) needs to 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 additional file systems are using this credential.

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 "Cleaning up on a storage cluster" on page 74.

Cleaning up IBM Spectrum Scale operator

Complete the following steps:

1. Enter the following command to delete the IBM Spectrum Scale Custom Resources.

```
oc delete -f scale_v1beta1_cluster_cr.yaml -n ibm-spectrum-scale
```

2. Enter the following command to uninstall the Operator, related objects, and namespaces.

```
oc delete -f https://raw.githubusercontent.com/IBM/ibm-spectrum-scale-container-native/
v5.1.4.0/generated/installer/ibm-spectrum-scale-operator.yaml
```

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
```

Note: If the current namespace was deleted in these steps (`ibm-spectrum-scale`, `ibm-spectrum-scale-operator`, `ibm-spectrum-scale-csi`), then the working namespace should be changed to an existing one.

```
oc project default
```

Cleaning up the worker nodes

IBM Spectrum Scale requires host path volume mounts and creates directories on each worker node.

Note: At this point, the project is deleted. Ensure that you are in the default namespace by entering **`oc project default`** command.

Complete the following steps:

1. Enter the following command to list the nodes that have the `node-role.kubernetes.io/worker=` label.

```
oc get nodes -l 'node-role.kubernetes.io/worker=' -o jsonpath="{range .items[*]}
{.metadata.name}{'\n'}"
```

2. For each of the listed worker nodes, enter the following command to create a debug pod that removes the host path volume mounted directories used by IBM Spectrum Scale:

```
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 ...
```

3. Ensure that none of the files are left by entering the following command:

```
oc debug node/<openshift_worker_node> -T -- chroot /host sh -c "ls /var/mmfs; ls /var/adm/
ras"
```

Example:

```
oc debug node/worker0.example.com -T -- chroot /host sh -c "ls /var/mmfs; ls /var/adm/ras"
   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
   Removing debug pod ...
   error: non-zero exit code from debug container
```

4. Remove node labels 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-
```

Cleaning up on a storage cluster

Delete the access permission that is granted to the IBM Spectrum Scale client cluster for mounting a remote file system.

Perform the following steps on the IBM Spectrum Scale storage cluster:

1. Enter the following command to query the name of the containerized client cluster:

```
 $ mmauth show all | grep ibm-spectrum-scale
   Cluster name: ibm-spectrum-scale.clustername.example.com
```

2. Enter the following command to remove the client cluster authorization:

```
$ mmauth delete ibm-spectrum-scale.clustername.example.com
   mmauth: Propagating the cluster configuration data to all affected nodes.
   mmauth: Command successfully completed
```

Chapter 10. 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 "Configuring the IBM Spectrum Scale container native cluster custom resources" on page 39.

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.

Chapter 11. 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” on page 77
- “Debugging IBM Spectrum Scale deployment” on page 77
- “Debugging the IBM Spectrum Scale Container Storage Interface (CSI) deployment” on page 79
- “Debugging OCP upgrade” on page 81
- “Common issues” on page 81
- “Known issues” on page 84
- “Collecting data for support” on page 88

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 cluster CR or making changes to 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 scale_v1beta1_cluster_cr.yaml
namespace/ibm-spectrum-scale unchanged
serviceaccount/ibm-spectrum-scale-core configured
serviceaccount/ibm-spectrum-scale-default configured
serviceaccount/ibm-spectrum-scale-gui configured
serviceaccount/ibm-spectrum-scale-pmcollector configured
role.rbac.authorization.k8s.io/ibm-spectrum-scale-sysmon unchanged
rolebinding.rbac.authorization.k8s.io/ibm-spectrum-scale-privileged unchanged
rolebinding.rbac.authorization.k8s.io/ibm-spectrum-scale-sysmon unchanged
callhome.scale.spectrum.ibm.com/callhome unchanged
remotecluster.scale.spectrum.ibm.com/remotecluster-sample unchanged
Error from server (InternalError): error when creating "scale_v1beta1_cluster_cr.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"
Error from server (InternalError): error when creating "scale_v1beta1_cluster_cr.yaml":
Internal error occurred: failed calling webhook "vfilesystem.scale.spectrum.ibm.com":
Post "https://ibm-spectrum-scale-webhook-service.ibm-spectrum-scale-operator.svc:443/validate-
scale-spectrum-ibm-com-v1beta1-filesystem?timeout=10s": no endpoints available for service "ibm-
spectrum-scale-webhook-service"
```

Checking status of the operator pod:

```
# 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 “Cleaning up IBM Spectrum Scale operator” on page 73 and “Cleaning up the worker nodes” on page 73. For more information about redeploy, see Chapter 4, “Installing the IBM Spectrum Scale container native operator and cluster,” on page 35 .

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 remoteclusters -n ibm-spectrum-scale
    oc describe remotecluster <name> -n ibm-spectrum-scale
    ```

  – Filesystems

    ```
    oc get filesystems -n ibm-spectrum-scale
    oc describe filesystem <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 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
```

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 71.

RestError: Failed to get storage cluster information. errmsg: 401 Unauthorized GET

The `oc describe gpfs` command shows the following error:

```
Warning  RestError               48s (x12 over 2m2s)  RemoteMount <filesystem>: [storage
cluster] Failed to get storage cluster information. errmsg: 401 Unauthorized GET request to
https://<storage cluster GUI>:443/scalemgmt/v2/cluster
```

The IBM Spectrum Scale GUI REST credentials for storage clusters are stored in kubernetes secrets. For more information, see "IBM Spectrum Scale storage cluster configuration" on page 38. The `RestError` indicates that the GUI user in the kubernetes secret does not match the GUI user in the storage cluster.

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 Operator User and Group.
- 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 -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 RemoteScale connector when cacert ConfigMap doesn't exist

```
# oc describe gpfs
...
Normal   RemoteMountAttempt            0s (x11 over 11s)      RemoteMount  fs1, Attempting to
configure remote storage file system=fs1 from remoteCluster=storageCluster1 as /mnt/fs1.
Warning  RemoteMountRestError          0s (x11 over 11s)      RemoteMount  fs1, [storage
cluster] Failed to establish RemoteScale connector. 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, see "Configuring Certificate Authority (CA) certificates for storage cluster" on page 53.

To resolve this issue, choose a configuration option from *Configuring certificate authority (CA) certificates for storage cluster* procedure and follow the instructions below for the corresponding option of choice.

For more information, see "Configuring Certificate Authority (CA) certificates for storage cluster" on page 53.

- Option 1

  Create the `cacert-storage-cluster-1` ConfigMap.

  For more information, see "Configuring Certificate Authority (CA) certificates for storage cluster" on page 53.
- Option 2

  Ensure that the storage cluster GUI is using a default trusted CA certificate. If configured correctly, the storage cluster GUI should connect successfully.
- Option 3

  Patch the Custom Resource to use `insecureSkipVerify: true`.

  ```
  oc patch scaleclusters ibm-spectrum-scale --type='json' -n ibm-spectrum-scale \
      -p='[{"op": "replace", "path": "/spec/remoteClusters/0/gui/insecureSkipVerify", "value":
  true}]'
  ```

Known issues

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 letting 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 24. 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.

File system fails to mount because it is already mounted on an IBM Spectrum Scale container native cluster

If a file system is failing to mount to the container native cluster ensure that this is not caused by the single cluster limitation:

The same remote file system cannot be mounted on multiple IBM Spectrum Scale container native clusters.

Grafana bridge pod 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 consuming performance metrics from IBM Spectrum Scale pmcollector. This also has a signature of Grafana Bridge pod failing to start.

To resolve this issue, complete the following steps:

1. Check NodeNetworkConfigurationPolicy to determine which network interfaces are configured for a node network.

   a. 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
```

b. 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 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.

a. Stop the sensors activities on all core pods:

```
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')
```

b. 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` Config Map:

```
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.

c. 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.

d. 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 which are not related to the configured node network interfaces.

a. 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
```

b. 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 on all core pods.

```
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.

Collecting data for support

You need to perform the following procedures to gather data for support:

- “Generating GPFS trace reports” on page 88
- “Configuring GPFS trace reports from cluster creation” on page 89
- “Kernel crash dumps” on page 89
- “Gather data about the IBM Spectrum Scale container native cluster” on page 89
- “Gather data about a Red Hat OpenShift Container Platform cluster” on page 92

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. 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
   ```

Configuring GPFS trace reports from cluster creation

In some situations, it may be required to configure GPFS tracing from cluster creation. This can be accomplished using the cluster core profile and settings directed by IBM Support Center.

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.

Gather data about the IBM Spectrum Scale container native cluster

To gather logs and diagnostic data to assist IBM Support in debugging an issue, enter the `oc adm must-gather` CLI command with the supporting must-gather image specifically for IBM Spectrum Scale container native.

The `ibm-spectrum-scale-must-gather` image collects the Kubernetes objects associated with its namespace and also retrieve a GPFS snap from the IBM Spectrum Scale container native cluster.

Prerequisites

- Running `oc adm must-gather` requires the user to be logged in to an account on a Red Hat OpenShift Container Platform cluster that has sufficient privileges to query OpenShift and Kubernetes resources.

Collaboration with the administrator may be needed to get necessary credentials for `oc login -u <username>` to successfully query OpenShift and Kubernetes resources.

- `oc adm must-gather --image` requires the must-gather image that is stored in a repository where it can be anonymously pulled (no credentials required). In an airgapped environment, the must-gather image must be pulled from the IBM Cloud Container Registry, and then uploaded to an image registry allowing anonymous pull. There are many methods for doing so, two of which are outlined.

  1. If a production grade Docker V2 compatible registry is already configured for a Red Hat OpenShift Container Platform cluster, and it can be set for anonymous pull of this image, then proceed with using this image registry.
  2. If you do not have a pre-configured image registry, one possible temporary solution is to configure the Red Hat OpenShift Container Platform internal registry.

  Note: The Red Hat OpenShift Container Platform internal registry may default to an emptydir storage setup. Any images stored within may be deleted if the image registry restarts. For more information, see Image Registry in Red Hat OpenShift documentation.

- In a connected environment, the Red Hat OpenShift cluster should be able to pull the image directly from ICR anonymously through the **`oc adm must-gather`** command.

Connected (non airgapped) environments

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:v5.1.4.0
   ```

   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
   ```

2. 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.

Airgapped environments

Retrieve ibm-spectrum-scale-must-gather image from IBM Cloud Container Registry

The following instruction is to pull the must-gather image from the IBM Cloud Container Registry's entitled repository.

Note: podman 1.6+ is required to perform the following step.

- Pull the image.

  ```
  podman pull icr.io/cpopen/ibm-spectrum-scale-must-gather:v5.1.4.0
  ```

Upload ibm-spectrum-scale-must-gather to OpenShift Container Platform internal registry

Before performing the following steps, the Red Hat OpenShift Container Platform internal registry must be configured and ready for use.

Note: For more information, see Image Registry in Red Hat OpenShift documentation.

1. Log in as `kubeadmin`.

   ```
   oc login -u kubeadmin
   ```

2. Add roles of registry-editor and edit to user `kube:admin`.

   ```
   oc policy add-role-to-user registry-editor kube:admin
   oc policy add-role-to-user edit kube:admin
   ```

3. Patch the configuration to expose the route to the OpenShift Container Platform internal registry.

   ```
   oc patch configs.imageregistry.operator.openshift.io/cluster \
   --patch '{"spec":{"defaultRoute":true}}' --type=merge
   ```

4. Set the following variables to facilitate with pushing images into the OpenShift Container Platform internal registry:

   - HOST - The name of the OpenShift Internal Container Registry
   - NAMESPACE - The namespace of your IBM Spectrum Scale container native cluster, default is `ibm-spectrum-scale`
   - IMAGE - The image name, `ibm-spectrum-scale-must-gather`
   - TAG - The tag given for this release, `v5.1.4.0`

     ```
     export HOST=$(oc get route default-route -n openshift-image-registry --
     template='{{ .spec.host }}')
     export NAMESPACE=ibm-spectrum-scale
     export IMAGE=ibm-spectrum-scale-must-gather
     export TAG=v5.1.4.0
     ```

5. Log in to the Red Hat OpenShift Container Platform integrated container registry through podman:

   ```
   oc whoami -t | podman login -u kubeadmin --password-stdin --tls-verify=false $HOST
   ```

6. List your images in podman and find the Image ID associated with the ibm-spectrum-scale-must-gather image retrieved from the IBM Cloud Container Registry.

   ```
   # podman images
   REPOSITORY                                                  TAG     IMAGE ID
   CREATED     SIZE
   icr.io/cpopen/ibm-spectrum-scale-must-gather                <none>  663727979b51  3 days
   ago  374 MB
   ```

7. Assign a new image name to the ibm-spectrum-scale-must-gather image by using its Image ID.

   ```
   podman tag 663727979b51 $HOST/$NAMESPACE/$IMAGE:$TAG
   ```

   Listing the images should now yield the image with the new name:

   ```
   # podman images
   REPOSITORY
               TAG       IMAGE ID      CREATED     SIZE
   default-route-openshift-image-registry.example.com/ibm-spectrum-scale/ibm-spectrum-scale-
   must-gather     v5.1.4.0  663727979b51  3 days ago  374 MB
   ```

8. Push the image into the Red Hat OpenShift Container Platform Image Registry.

   ```
   podman push $HOST/$NAMESPACE/$IMAGE:$TAG --tls-verify=false
   ```

9. Ensure that the ImageStream contains the ibm-spectrum-scale-must-gather image.

   ```
   oc get is $IMAGE -n ibm-spectrum-scale -oyaml | egrep "name:|dockerImageRepository|tag"
   ```

   Example:

   ```
   name: ibm-spectrum-scale-must-gather
   dockerImageRepository: image-registry.openshift-image-registry.svc:5000/ibm-spectrum-scale/
   ibm-spectrum-scale-must-gather
   tags:
       tag: v5.1.4.0
   ```

Execute the ibm-spectrum-scale-must-gather image

Complete the following steps:

1. In the directory where your must-gather contents are to be stored, enter the must-gather command using the `ibm-spectrum-scale-must-gather` image:

   ```
   oc adm must-gather --image=image-registry.openshift-image-registry.svc:5000/ibm-spectrum-
   scale/ibm-spectrum-scale-must-gather:v5.1.4.0
   ```

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.

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.

Chapter 12. 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

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.

Notices

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

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

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

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

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

COPYRIGHT LICENSE:

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

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

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

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

Trademarks

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

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

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

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

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

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

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

Terms and conditions for product documentation

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

IBM Privacy Policy

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

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

Applicability

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

Personal use

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

Commercial use

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

Rights

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

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

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

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

Glossary

This glossary provides terms and definitions for IBM 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*.

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

D

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

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

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

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

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

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

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

E

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

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

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

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

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

F

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

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

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

FEK
See *file encryption key*.

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

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

file audit logging
Provides the ability to monitor user activity of IBM 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*.

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:

SC28-3168-18

(1P) P/N: