IBM Storage Scale
5.1.9.1

Data Access Services Guide

SC27-9872-10

Note

Before using this information and the product it supports, read the information in "Notices" on page 155.

This edition applies to Version 5 release 1 modification 9 of the following products, and to all subsequent releases and modifications until otherwise indicated in new editions:

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

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

IBM welcomes your comments; see the topic "How to send your comments" on page xxxii. 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

Figures

Tables

About this information

This edition applies to IBM Storage Scale version 5.1.9 for AIX®, Linux®, and Windows.

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

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

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

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

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

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

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

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

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

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

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

Table 1. IBM Storage Scale library information units

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

Table 1. IBM Storage Scale library information units (continued)		
Information unit	**Type of information**	**Intended users**
IBM Storage Scale: Concepts, Planning, and Installation Guide	**Planning** • Planning for GPFS • Planning for protocols • Planning for cloud services • Planning for IBM Storage Scale on Public Clouds • Planning for AFM • Planning for AFM DR • Planning for AFM to cloud object storage • Planning for performance monitoring tool • Planning for UEFI secure boot	
IBM Storage Scale: Concepts, Planning, and Installation Guide	• Firewall recommendations • Considerations for GPFS applications • Security-Enhanced Linux support • Space requirements for call home data upload	

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

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

Table 1. IBM Storage Scale library information units (continued)		
Information unit	**Type of information**	**Intended users**
IBM Storage Scale: Concepts, Planning, and Installation Guide	• Upgrading IBM Storage Scale non-protocol Linux nodes • Upgrading IBM Storage Scale protocol nodes • Upgrading IBM Storage Scale on cloud • Upgrading GPUDirect Storage • Upgrading AFM and AFM DR • Upgrading object packages • Upgrading SMB packages • Upgrading NFS packages • Upgrading call home • Upgrading the performance monitoring tool • Upgrading signed kernel modules for UEFI secure boot • Manually upgrading pmswift • Manually upgrading the IBM Storage Scale management GUI • Upgrading Cloud services • Upgrading to IBM Cloud Object Storage software level 3.7.2 and above • Upgrade paths and commands for file audit logging and clustered watch folder • Upgrading IBM Storage Scale components with the installation toolkit • Protocol authentication configuration changes during upgrade • Changing the IBM Storage Scale product edition • Completing the upgrade to a new level of IBM Storage Scale • Reverting to the previous level of IBM Storage Scale	System administrators, analysts, installers, planners, and programmers of IBM Storage Scale clusters who are very experienced with the operating systems on which each IBM Storage Scale cluster is based

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

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

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

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

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

Table 1. IBM Storage Scale library information units (continued)		
Information unit	**Type of information**	**Intended users**
IBM Storage Scale: Administration Guide	• Managing protocol services • Managing protocol user authentication • Managing protocol data exports • Managing object storage • Managing GPFS quotas • Managing GUI users • Managing GPFS access control lists • Native NFS and GPFS • Accessing a remote GPFS file system • Information lifecycle management for IBM Storage Scale • Creating and maintaining snapshots of file systems • Creating and managing file clones • Scale Out Backup and Restore (SOBAR) • Data Mirroring and Replication • Implementing a clustered NFS environment on Linux • Implementing Cluster Export Services • Identity management on Windows / RFC 2307 Attributes • Protocols cluster disaster recovery • File Placement Optimizer • Encryption • Managing certificates to secure communications between GUI web server and web browsers • Securing protocol data • 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 cloudkit • Administering AFM • Administering AFM DR	System administrators or programmers of IBM Storage Scale systems

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

Information unit	Type of information	Intended users
IBM Storage Scale: Administration Guide	• Administering AFM to cloud object storage • Highly available write cache (HAWC) • Local read-only cache • Miscellaneous advanced administration topics • GUI limitations	System administrators or programmers of IBM Storage Scale systems

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

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

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

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

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

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

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

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

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

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

Information unit	Type of information	Intended users
IBM Storage Scale: Big Data and Analytics Guide	Cloudera Data Platform (CDP) Private Cloud Base • Overview • Planning • Installing • Configuring • Administering • Monitoring • Upgrading • Limitations • Problem determination	• System administrators of IBM Storage Scale systems • Application programmers who are experienced with IBM Storage Scale systems and familiar with the terminology and concepts in the XDSM standard
IBM Storage Scale: Big Data and Analytics Guide	Cloudera HDP 3.X • Planning • Installation • Upgrading and uninstallation • Configuration • Administration • Limitations • Problem determination Open Source Apache Hadoop • Open Source Apache Hadoop without CES HDFS • Open Source Apache Hadoop with CES HDFS	• System administrators of IBM Storage Scale systems • Application programmers who are experienced with IBM Storage Scale systems and familiar with the terminology and concepts in the XDSM standard

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

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

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

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

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

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

Prerequisite and related information

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

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

Conventions used in this information

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

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

Table 2. Conventions

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

Table 2. Conventions (continued)

<table>
<tr><th>Convention</th><th>Usage</th></tr>
<tr><td><code>constant width</code></td><td>Examples and information that the system displays appear in <code>constant-width</code> typeface.

Depending on the context, <code>constant-width</code> typeface sometimes represents path names, directories, or file names.</td></tr>
<tr><td>italic</td><td>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.</td></tr>
<tr><td><<i>key</i>></td><td>Angle brackets (less-than and greater-than) enclose the name of a key on the keyboard. For example, <code><Enter></code> refers to the key on your terminal or workstation that is labeled with the word <i>Enter</i>.</td></tr>
<tr><td>\</td><td>In command examples, a backslash indicates that the command or coding example continues on the next line. For example:

<code>mkcondition -r IBM.FileSystem -e "PercentTotUsed > 90" \</code>
<code>-E "PercentTotUsed < 85" -m p "FileSystem space used"</code></td></tr>
<tr><td>{item}</td><td>Braces enclose a list from which you must choose an item in format and syntax descriptions.</td></tr>
<tr><td>[item]</td><td>Brackets enclose optional items in format and syntax descriptions.</td></tr>
<tr><td><code><Ctrl-<i>x</i>></code></td><td>The notation <code><Ctrl-<i>x</i>></code> indicates a control character sequence. For example, <code><Ctrl-c></code> means that you hold down the control key while pressing <code><c></code>.</td></tr>
<tr><td>item...</td><td>Ellipses indicate that you can repeat the preceding item one or more times.</td></tr>
<tr><td>|</td><td>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.</td></tr>
</table>

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

How to send your comments

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

`mhvrcfs@us.ibm.com`

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

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

`scale@us.ibm.com`

Chapter 1. Release notes

IBM Storage Scale Data Access Services (DAS) 5.1.9.1 release notes.

IBM Storage Scale DAS supports the S3 access protocol and is part of IBM Storage Scale container native (CNSA) which is a containerized version of IBM Storage Scale. IBM Storage Scale DAS S3 access protocol enables clients to access data that is stored in IBM Storage Scale file systems as objects.

About this release:

IBM Storage Scale DAS 5.1.9.1 is now generally available. The topic includes new features, changes, and known issues that pertain to IBM Storage Scale DAS 5.1.9.1 release.

- Supported software levels
  - IBM Storage Scale DAS 5.1.9.1 is supported on Red Hat OpenShift Container Platform (OCP) 4.14.x and it supports Red Hat OpenShift Data Foundation (ODF) 4.14.x. For more information, see "Software requirements" on page 15.
- Upgrade
  - Ability to upgrade IBM Storage Scale DAS from 5.1.7 to 5.1.9.1. For more information, see Upgrading.
- Security
  - Added ability to use Transport Layer Security (TLS) verification to guarantee secure HTTPS communication with the storage cluster GUI by verifying the server's certificate chain. To use this the security protocol must be configured for use with CNSA. For more information, see the Configuring Certificate Authority (CA) certificates section in the IBM Storage Scale container native documentation.
- Known issues
  - IBM Storage Scale DAS 5.1.9.1 has some known issues. For more information, see "Known issues" on page 116.

Note: IBM Storage Scale DAS is deprecated and it will be removed from future releases.

Chapter 2. Product overview

IBM Storage Scale Data Access Services (DAS) supports the S3 access protocol that enables clients to access data that is stored in IBM Storage Scale file systems as objects.

Architecture

IBM Storage Scale DAS modernizes IBM Storage Scale's in-built support for S3 access. IBM Storage Scale DAS requires a dedicated Red Hat OpenShift cluster that runs only IBM Storage Scale CNSA and IBM Storage Scale DAS.

S3 applications use the S3 protocol to access data in IBM Storage Scale. They run outside the Red Hat OpenShift cluster by using any underlying infrastructure. These include traditional applications on bare metal servers or virtual machines, containerized applications on Red Hat OpenShift, vanilla Kubernetes, or any other container orchestration platform, and embedded applications integrated in hardware appliances.

Administrators use the IBM Storage Scale CLI and the REST API to manage all components of IBM Storage Scale including IBM Storage Scale DAS. They use the Red Hat OpenShift CLI, Web UI, and REST API to manage the underlying Red Hat OpenShift cluster.

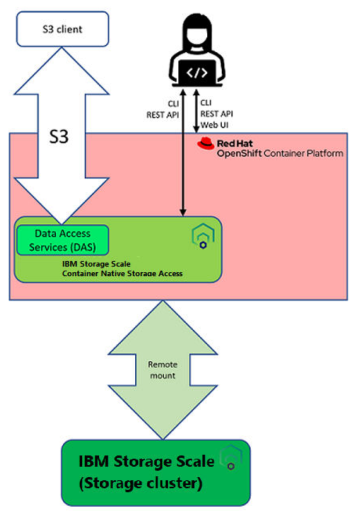

Figure 1. IBM Storage Scale DAS architecture

Infrastructure architecture

IBM Storage Scale DAS on dedicated Red Hat OpenShift clusters requires three x86_64 based bare metal servers. Each server is configured as a Data Access Node (DAN) running Red Hat OpenShift, IBM Storage Scale container native, IBM Storage Scale CSI, and IBM Storage Scale DAS.

The three DANs must be configured as compact Red Hat OpenShift cluster. A compact cluster is a three-node cluster where each Red Hat OpenShift node acts as a combined master and worker node. For more information, see the following Red Hat OpenShift documentation resources:

- Configuring a three-node cluster
- Delivering a Three-node Architecture for Edge Deployments (blog)

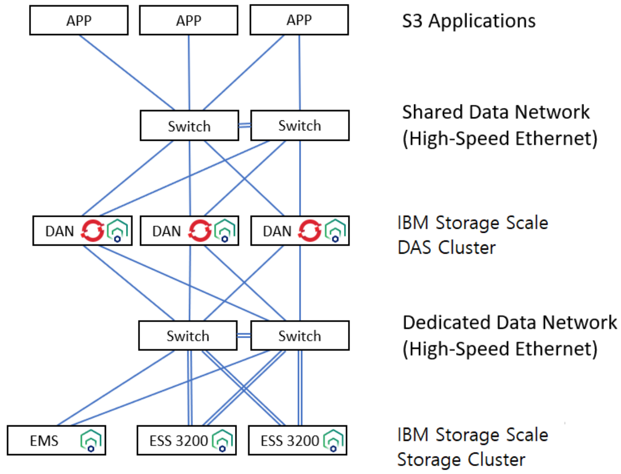

Figure 2. Example infrastructure architecture for IBM Storage Scale DAS deployment

The IBM Storage Scale storage cluster owns the IBM Storage Scale file system that is used to store S3 data. IBM Storage Scale DAS limits storage options to IBM Elastic Storage System (ESS) only. All ESS models are supported. The storage cluster includes one IBM ESS Management Server (EMS) and one or more IBM Elastic Storage System (ESS).

The IBM Storage Scale DAS cluster, or more precisely the IBM Storage Scale container native cluster running on the IBM Storage Scale DAS cluster, remotely mounts an IBM Storage Scale file system provided by the IBM Storage Scale storage cluster.

Each DAN exposes one IP address for S3 access. To provide scalable S3 performance, IBM Storage Scale DAS supports configuring high-speed Ethernet networks in addition to the default network for the Red Hat OpenShift cluster. To provide good S3 performance, it is required to connect the S3 clients through a well-controlled data center network, for example, the same layer 2 network. A dedicated data network must be provided to connect all IBM Storage Scale nodes that are not connected to any shared data network, such as a data center network, a campus network, or the Internet.

IBM Storage Scale DAS supports all bare metal Ethernet configurations which are supported by IBM Storage Scale container native and Red Hat OpenShift:

- IBM Storage Scale container native network requirements
- Red Hat Open Shift Container Platform - Understanding networking

Deployment architecture

IBM Storage Scale DAS is deployed on the top of IBM Storage Scale container native and IBM Storage Scale CSI.

Figure 3 on page 5 illustrates an example deployment of container native and IBM Storage Scale CSI. Only the IBM Storage Scale core pods are in the data path. All the other IBM Storage Scale pods are required to configure and monitor IBM Storage Scale container native. IBM Storage Scale CSI provides application pods running on the same Red Hat OpenShift cluster access to data that is stored in IBM Storage Scale. For detailed description of each pod, see IBM Storage Scale container native and IBM Storage Scale CSI documentation.

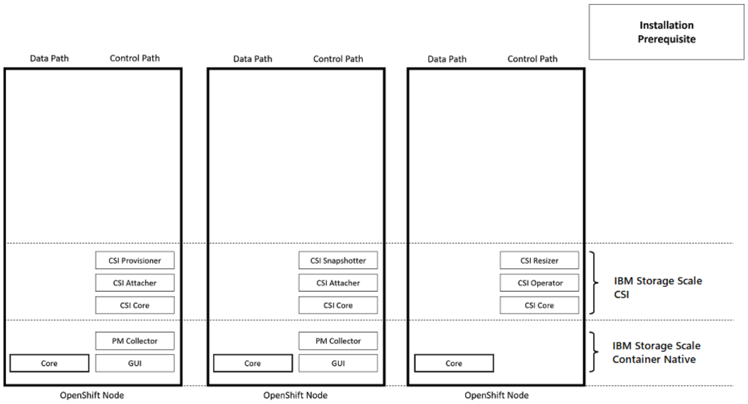

Figure 3. Example deployment of IBM Storage Scale container native and IBM Storage Scale CSI

IBM Storage Scale DAS is deployed by applying the manifest file for IBM Storage Scale DAS, see Figure 4 on page 5. The application of the IBM Storage Scale DAS manifest first deploys the IBM Storage Scale DAS operator. The IBM Storage Scale DAS operator then deploys the IBM Storage Scale DAS endpoints that provide an internal REST API to configure and monitor IBM Storage Scale DAS.

IBM Storage Scale DAS includes an embedded license for Red Hat OpenShift Data Foundation, see Figure 5 on page 6. The IBM Storage Scale DAS operator therefore implicitly deploys Red Hat OpenShift Data Foundation. The use of Red Hat OpenShift Data Foundation is limited to the features that can be configured with the IBM Storage Scale DAS management interfaces. For a detailed description of each Red Hat OpenShift Data Foundation pod, see Red Hat OpenShift Data Foundation documentation.

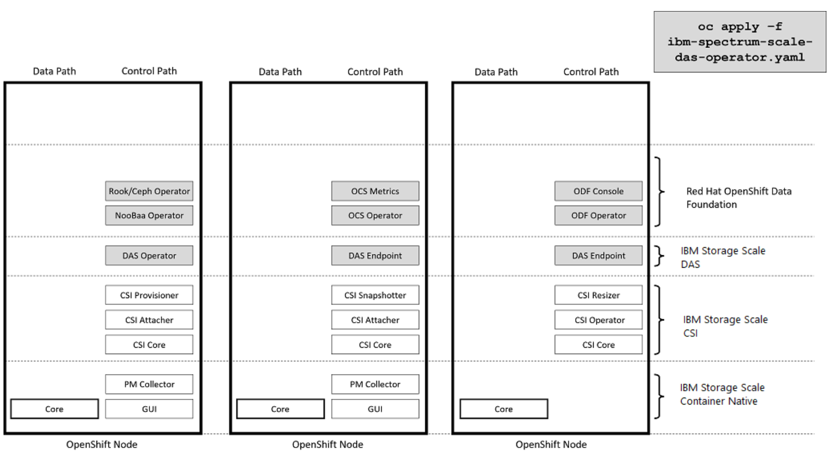

Figure 4. The application of the IBM Storage Scale DAS manifest deploys the control pods for IBM Storage Scale DAS and for Red Hat OpenShift Data Foundation

After deploying IBM Storage Scale DAS, the IBM Storage Scale DAS S3 service can be deployed using the **mmdas service create** command or the respective IBM Storage Scale DAS REST API request. The creation of the IBM Storage Scale DAS S3 service implicitly deploys and configures the NooBaa component of Red Hat OpenShift Data Foundation. The NooBaa component provides S3 access to data stored in IBM Storage Scale. The NooBaa endpoint pods are in the data path and they provide S3 access to data that is stored in IBM Storage Scale file systems. All other NooBaa pods are required to configure and monitor NooBaa. For a detailed description of the NooBaa pods, see the Red Hat OpenShift Data Foundation documentation.

The creation of the IBM Storage Scale DAS S3 service also deploys the NooBaa Monitor pod in the namespace for IBM Storage Scale container native. The NooBaa Monitor pod integrates the monitoring of NooBaa in the IBM Storage Scale management framework.

The creation of the IBM Storage Scale DAS S3 service furthermore deploys and configures the Red Hat OpenShift MetalLB feature. IBM Storage Scale DAS uses MetalLB to provide an S3 endpoint on each Red Hat OpenShift node that is configured for IBM Storage Scale DAS, and it provides resiliency against

Red Hat OpenShift node failures. For a detailed description of each MetalLB pod, see OpenShift MetalLB documentation.

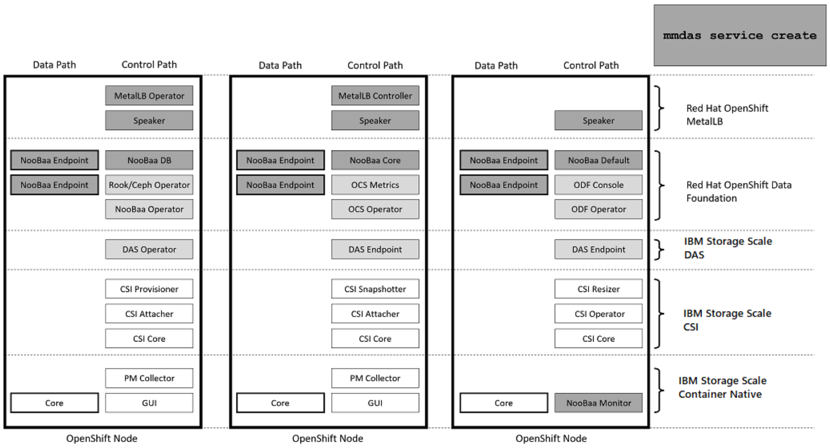

Figure 5. The creation of the IBM Storage Scale DAS S3 service configures the NooBaa component of Red Hat OpenShift Data Foundation, and it installs and configures the OpenShift MetalLB feature

Data path architecture

The data path of IBM Storage Scale DAS comprises three tiers that are parallel to the three tiers of the infrastructure architecture. For more information, see "Infrastructure architecture" on page 3.

Figure 6 on page 6 illustrates how object data is represented at each layer. For information on how to configure the different layers, see Chapter 4, "Installing," on page 27.

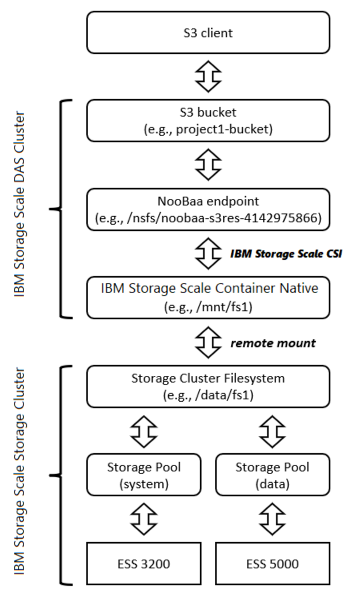

Figure 6. Example IBM Storage Scale DAS data path

The IBM Storage Scale file system for IBM Storage Scale DAS comprises one or more storage pools that contain the disks provided by the storage devices. The example illustrates one IBM Storage Scale file system (`fs1` mounted at `/data/fs1`), which comprises two ESS based storage pools. The file system includes the directory `project1-data` and the file `message`.

```
ls /data/fs1/project1-data
message
```

```
cat /data/fs1/project1-data/message
IBM Storage Scale provides scalable performance.
```

The IBM Storage Scale DAS cluster includes IBM Storage Scale container native. IBM Storage Scale container native remotely mounts the file system `fs1` at `/mnt/fs1`. The directory `project1-data` and the file `message` are available under the respective paths.

```
ls /mnt/fs1/project1-data
message
```

```
cat /mnt/fs1/project1-data/message
IBM Storage Scale provides scalable performance.
```

The NooBaa endpoint pods of Red Hat OpenShift Data Foundation provide S3 access to data in IBM Storage Scale. IBM Storage Scale DAS uses IBM Storage Scale CSI to make IBM Storage Scale file systems available in NooBaa endpoint pods. NooBaa mounts the IBM Storage Scale file systems in sub-directories of directory `/nsfs`. In this example, the file system `fs1` is mounted at `/nsfs/noobaa-s3res-4142975866`. The directory `project1-data` and the file `message` are available under the respective paths.

```
ls /nsfs/noobaa-s3res-4142975866/project1-data
message
```

```
cat /nsfs/noobaa-s3res-4142975866/project1-data/message
IBM Storage Scale provides scalable performance.
```

IBM Storage Scale DAS makes configurable directories in IBM Storage Scale file systems accessible as S3 buckets. In this example, the directory `project1-data` is exported as the S3 bucket `project1-bucket`. The **mmdas** command can report all exported directories and the mapping of S3 buckets to file system directories.

```
mmdas export list project1-bucket

 Name                   Filesystem Path
 -----                  -----------------
 project1-bucket        /mnt/fs1/project1-data/
```

```
mmdas export list

 Name
 ------
 project1-bucket
 project2-bucket
 shared-bucket
```

S3 applications can access such exported directories and files as S3 buckets and S3 objects. In this example, the file `message` in the directory `project1-data` is accessible as S3 object `message` in the S3 bucket `project1-bucket`. In the following output, the command `s3p1` is an alias for the AWS CLI.

Note: To set the alias for s3p1, see "Example configuration of IBM Storage Scale DAS" on page 37.

```
s3p1 ls s3://project1-bucket
2022-03-12 08:40:28         50 message
```

```
s3p1 cp s3://project1-bucket/message mymessage
download: s3://project1-bucket/message to ./mymessage
```

```
cat mymessage
IBM Storage Scale provides scalable performance.
```

Control path architecture

For the control path, IBM Storage Scale DAS adds new endpoints to the IBM Storage Scale REST API of the IBM Storage Scale container native cluster.

The `mmdas` command is a front-end to the IBM Storage Scale REST API to configure and manage all resources of IBM Storage Scale DAS. IBM Storage Scale container native GUI pods forward IBM Storage Scale REST API requests that are related to IBM Storage Scale DAS through an internal REST API to the IBM Storage Scale DAS endpoint pods. The IBM Storage Scale DAS endpoint pods use Kubernetes Custom Resources (CRs) and internal RPC calls to serve IBM Storage Scale DAS related REST API requests.

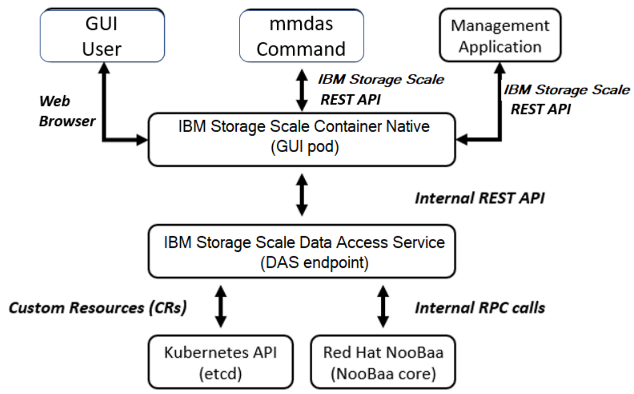

Figure 7. IBM Storage Scale DAS control path

S3 object access for AI and analytics workloads

IBM Storage Scale DAS provides a new S3 object access service that is built into IBM Storage Scale. The S3 object access service is optimized for AI and analytics workloads that use large objects.

S3 objects and S3 buckets are mapped 1:1 to files and directories in IBM Storage Scale file systems and vice versa. An IBM Storage Scale file system provides the storage capacity for the object data. All data must be created, processed, and deleted by using the S3 object access protocol. For more information, see "Example configuration of IBM Storage Scale DAS" on page 37.

Scaling

The topic describes IBM Storage Scale DAS scaling options.

The IBM Storage Scale DAS supports the following scaling options:

- 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 Storage Scale DAS cluster can be attached to one IBM Storage Scale storage cluster and to one IBM Storage Scale file system only
- Each IBM Storage Scale storage cluster can be attached to one IBM Storage Scale DAS cluster only

Performance

The performance of IBM Storage Scale DAS is highly dependent on your underlying infrastructure and workload.

IBM published the following benchmark results for IBM Storage Scale DAS:

- COSBench using objects with a size of 1 GB running against a three-node IBM Storage Scale DAS cluster and using IBM Elastic Storage System 3200 as the back-end storage:
  - More than 60 GB/s aggregated throughput for read workloads
  - More than 20 GB/s aggregated throughput for write workloads

For more information, see following resources:

- IBM Data Access Services (DAS) performance evaluation using COSBench and large objects
- IBM Data Access Services (DAS) read performance evaluation of small objects using COSBench

.

Security

As a feature of IBM Storage Scale and running on Red Hat OpenShift, IBM Storage Scale DAS inherits the in-built security of IBM Storage Scale and Red Hat OpenShift.

IBM Storage Scale DAS uses S3 accounts, access control lists (ACLs), allows Security-Enhanced Linux (SELinux), encryption, and audit logging to secure your data.

Other security considerations are as follows:

- IBM Storage Scale DAS sets network policy to allow incoming connection requests from pods from trusted Kubernetes namespaces only.
- IBM Storage Scale DAS sets network policy to allow outgoing connection requests to pods/resources of trusted Kubernetes namespaces only.
- All containers in IBM Storage Scale DAS pods in the `ibm-spectrum-scale-das` namespace run with non-root user permissions. Similarly, containers in IBM Storage Scale DAS monitoring pod run with non-root user permissions.
- All the containers in IBM Storage Scale DAS pods run in the non-privileged mode.
- Secured TLS connections to NooBaa in `openshift-storage` namespace.
- IBM Storage Scale DAS supports the ability to use Transport Layer Security (TLS) verification to guarantee secure HTTPS communication with the storage cluster GUI by verifying the certificate chain of server.

For more information, see “Security requirements” on page 15.

Deployment

IBM Storage Scale DAS requires a dedicated and compact Red Hat OpenShift cluster. Compact Red Hat OpenShift clusters are three-node clusters in which each Red Hat OpenShift node acts as a combined master and worker node.

The Red Hat OpenShift cluster must be dedicated to IBM Storage Scale DAS. You must not have other application pods on the same Red Hat OpenShift cluster. S3 applications must run on collocated and separate servers (same layer 2 network) running any operating system or any Kubernetes platform.

IBM Storage Scale DAS requires the Red Hat OpenShift cluster to be configured with IBM Storage Scale container native and IBM Storage Scale Container Storage Interface. The IBM Storage Scale container native cluster imports (remotely mounts) one IBM Storage Scale file system that is provided by a collocated IBM Storage Scale storage cluster. The IBM Storage Scale file system must be configured with NFSv4 ACLs. The storage cluster must be based on IBM Elastic Storage Systems (ESS).

IBM Storage Scale DAS includes an embedded license for Red Hat OpenShift Data Foundation the SKU MCT4201 Red Hat Cloud Data Federation for IBM Storage Scale. IBM Storage Scale DAS installs and configures the supported version of Red Hat OpenShift Data Foundation. The use of Red Hat OpenShift Data Foundation is limited to the integration in IBM Storage Scale. The use of Red Hat OpenShift Data Foundation features that are not configured by IBM Storage Scale DAS is not supported.

To improve scaling and performance of S3 object access, IBM Storage Scale DAS supports deployments on compact Red Hat OpenShift clusters that, in addition to the default Red Hat OpenShift network, are configured with high-speed Ethernet. For more information on configuring multiple networks for Red Hat OpenShift, see Red Hat OpenShift documentation.

Built on Red Hat OpenShift Container Platform and IBM Storage Scale, IBM Storage Scale DAS is resilient against infrastructure outages such as failures of Red Hat OpenShift Container Platform nodes and storage failures. IBM Storage Scale DAS uses the MetalLB feature of Red Hat OpenShift Container Platform to provide high-availability and load distribution of S3 object access.

For more information on deployment, see Chapter 3, “Planning,” on page 13 and “Deployment architecture” on page 4.

Installation

To install IBM Storage Scale Data Access Services (DAS), customers must provide an IBM Storage Scale storage cluster based on IBM Elastic Storage Systems. In addition, a dedicated compact Red Hat OpenShift Container Platform Cluster running IBM Storage Scale container native and the required networks.

The installation procedure of IBM Storage Scale DAS customizes the IBM Storage Scale file system and Red Hat OpenShift Container Platform to provide storage for internal metadata, S3 buckets, and S3 objects, and then installs all components of IBM Storage Scale DAS. IBM Storage Scale DAS supports disconnected deployments (air gap installation).

For more information, see Chapter 4, “Installing,” on page 27.

Management

IBM Storage Scale DAS is an IBM Storage Scale feature that seamlessly integrates with IBM Storage Scale’s existing configuration and monitoring stack.

IBM Storage Scale DAS adds new endpoints to the IBM Storage Scale REST API for IBM Storage Scale container native clusters and the new **mmdas** command to manage S3 service, S3 accounts, and S3 buckets.

The existing IBM Storage Scale commands **mmhealth** and **gpfs.snap**, IBM Container Native Storage Access MustGather, Red Hat OpenShift Data Foundation MustGather, and IBM Storage Scale call home are enhanced to include IBM Storage Scale DAS related configuration and status.

For more information, see “Collecting data for support” on page 92 and Chapter 7, “Monitoring,” on page 89.

S3 service

The S3 service of IBM Storage Scale DAS provides the data path for S3 object access to files and directories stored in IBM Storage Scale file systems.

IBM Storage Scale DAS allows administrators to manage the S3 service using the **mmdas** command or the IBM Storage Scale REST API. Basic management of the S3 service includes creating, deleting, enabling, disabling, and reporting the status of the S3 service.

Advanced configuration options allow administrators to configure the IP addresses for S3 object access to disable the automatic failover and failback of IP addresses in case of Red Hat OpenShift node failures, configure the scaling of S3 object access to optimally use the underlying servers and networks, and optionally generate MD5 based ETags to support applications that require MD5 based ETags.

For more information, see “Managing S3 object service instance” on page 69.

S3 accounts

IBM Storage Scale DAS uses S3 accounts to manage S3 access keys for S3 clients and their respective UIDs and GIDs.

IBM Storage Scale DAS allows administrators to manage S3 accounts by using the **mmdas** command or the IBM Storage Scale REST API. Basic management of S3 accounts include creating, deleting, and listing of S3 accounts. It also allows administrators to update the S3 access keys and the default path for new S3 buckets that are created with the S3 CreateBucket request.

For more information, see "Managing accounts for S3 object access" on page 74.

S3 buckets

IBM Storage Scale DAS maps each S3 bucket to a directory in the IBM Storage Scale file system.

In IBM Storage Scale DAS S3 buckets are referred to as S3 exports. IBM Storage Scale DAS allows administrators to create, delete, and list S3 buckets using the **mmdas** command or the IBM Storage Scale REST API.

IBM Storage Scale DAS allows S3 clients to manage S3 buckets by using the following S3 REST API requests:

- S3 CreateBucket
- S3 ListObjects
- S3 ListObjectsV2
- S3 DeleteBucket
- S3 HeadBucket
- S3 ListBuckets
- S3 ListMultipartUploads

For more information, see "Managing S3 object exports" on page 80.

S3 objects

IBM Storage Scale DAS maps each S3 object to a file in the IBM Storage Scale file system.

IBM Storage Scale DAS allows S3 clients to manage S3 objects by using the following S3 REST API requests:

- S3 PutObject
- S3 GetObject
- S3 HeadObject
- S3 CopyObject
- S3 DeleteObject
- S3 DeleteObjects
- S3 CreateMultipartUpload
- S3 CompleteMultipartUpload
- S3 AbortMultipartUpload
- S3 UploadPart
- S3 UploadPartCopy
- S3 ListParts

IBM Storage Scale DAS allows S3 applications to store user-defined object metadata in addition to the object data itself.

Data management

IBM Storage Scale DAS stores S3 objects and S3 buckets as files and directories in the IBM Storage Scale file system that is owned by the IBM Storage Scale storage cluster.

IBM Storage Scale DAS supports the use of selected data management features that are in-built in IBM Storage Scale. These include following features:

- IBM Storage Scale filesets to prepare the underlying IBM Storage Scale file system for the use of fileset based data management.
- IBM Storage Scale storage pools and IBM Storage Scale information lifecycle management (ILM) to integrate storage media with varying performance and capacity into the same file system, such as NVMe, SSD, and NL-SAS.
- Backup and restore the files and directories using the IBM Storage Scale **mmbackup** command.

For more information, see Information lifecycle management and Protecting data in a file system using backup in IBM Storage Scale documentation.

Multi-protocol data sharing with S3, NFS, POSIX, and IBM Storage Scale CSI

Multi-protocol data sharing for file and object access allows use cases where you can access data by using object and file interfaces.

Some of the key unified file and object access use cases are as follows:

- Accessing object by using file interfaces and accessing file by using object interfaces help legacy applications that are designed for file to start integrating into the object world.
- It allows files exported using NFS or IBM Storage Scale CSI, or files available on POSIX, to be accessible as objects using HTTP to the end clients.
- Multi-protocol access for file and object that is available in different environments allows supporting and sharing data with multiple access options. For more information about the NFS protocol, see the Configuring the CES and protocol configuration section.

Unified file and object access allows users to access the same data as an object and as a file. Data can be stored and retrieved through IBM Storage Scale DAS for object storage or through IBM Storage Scale as files from POSIX and NFS interfaces, or through IBM Storage Scale CSI. The unified file and object access provides the following capabilities:

- Ingest data by using the object interface, and access this data from the file interface.
- Ingest data by using the file interface, and access this data from the object interface.
- Ingest data by using IBM Storage Scale CSI, and access this data from the file or object interface.

For more information about the unified file and object access, check the *Multiprotocol data sharing across Data Access Services (S3) - NFS - CSI - POSIX* blog.

Limitations

- This feature is tested with basic authentication only. It is not tested with any external authentication mechanism on IBM Storage Scale.
- Concurrent data access with locking enabled has not been tested because the locking feature needs to be enabled or designed across the containerized and noncontainerized clusters.

Known issues

The IBM Storage Scale DAS 5.1.9 release has some known issues.

For more information, see “Known issues” on page 116.

Chapter 3. Planning

This section enables you to prepare for IBM Storage Scale DAS installation. To plan your IBM Storage Scale DAS installation, review the information in Chapter 2, “Product overview,” on page 3, “Architecture” on page 3, and “Security requirements” on page 15.

Hardware requirements

The topic lists IBM Storage Scale DAS 5.1.9.1 hardware requirements.

Solution components

An IBM Storage Scale DAS deployment includes Data Access Nodes (DAN) based on a dedicated compact Red Hat OpenShift cluster, an IBM Storage Scale storage cluster based on IBM Elastic Storage System (ESS) and networks. For the overall solution architecture, see “Infrastructure architecture” on page 3.

Dedicated Red Hat OpenShift clusters

IBM Storage Scale DAS on dedicated Red Hat OpenShift clusters requires three x86_64 based bare metal servers. Each server is configured as a DAN running Red Hat OpenShift, IBM Storage Scale container native, IBM Storage Scale CSI, and IBM Storage Scale DAS. The three DANs must be configured as compact Red Hat OpenShift cluster. A compact cluster is a three-node cluster where each Red Hat OpenShift node acts as a combined master and worker node. For more information, see the following Red Hat OpenShift documentation resources:

- Configuring a three-node cluster
- Delivering a Three-node Architecture for Edge Deployments (blog)

Temporary bootstrap node

For installing Red Hat OpenShift, you require a temporary bootstrap node. You can remove the bootstrap node after Red Hat OpenShift is installed. The bootstrap node can be a VM in your infrastructure or on your laptop but it must meet the installation prerequisites. These prerequisites include CPU, memory, DNS, and network connectivity. For more information, see the following Red Hat OpenShift installation documentation resources:

- Required machines
- Minimum resource requirements

Network considerations

IBM recommends configuring a dedicated data network and a shared data network in addition to the default network for the Red Hat OpenShift cluster. For the recommended network architecture, see “Infrastructure architecture” on page 3.

The dedicated data network connects all IBM Storage Scale nodes of the Storage Cluster. It is not connected to any shared data network, such as a data center network, a campus network, or the Internet.

To provide the best performance, it is recommended to connect the S3 clients through a well-controlled data network, for example, the same layer 2 network.

IBM Storage Scale DAS supports all bare metal Ethernet configurations which are supported by IBM Storage Scale container native and Red Hat OpenShift:

- IBM Storage Scale container native network requirements
- Red Hat Open Shift Container Platform - Understanding networking

Example: Dedicated IBM Storage Scale DAS Cluster optimized for minimal rack space

Choose 1U servers for rack-space optimized configurations. The following figure depicts an example deployment with three 1U DANs where each DAN is configured with 2x dual-port 100 Gb/s network interface cards (NICs) providing 4x100 Gb/s ports in total. This allows connecting each DAN with two 100 Gb/s links to the shared data network and with two 100 Gb/s links to the dedicated data network, providing high availability and good performance.

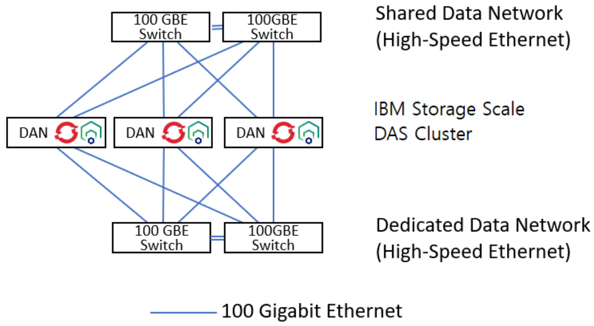

Figure 8. Connecting each DAN with two 100 Gb/s links

Example: Dedicated IBM Storage Scale DAS cluster optimized for performance

Choose 2U servers for performance optimized configurations. 2U servers allow adding more NICs than 1U servers. The following figure depicts an example deployment with three 2U DANs where each DAN is configured with 2x dual-port 100 Gb/s NICs and 2x single-port 200 Gb/s NICs. This allows connecting each DAN with four 100 Gb/s links to the shared data network and with two 200 Gb/s links to the dedicated data network providing high availability and high performance.

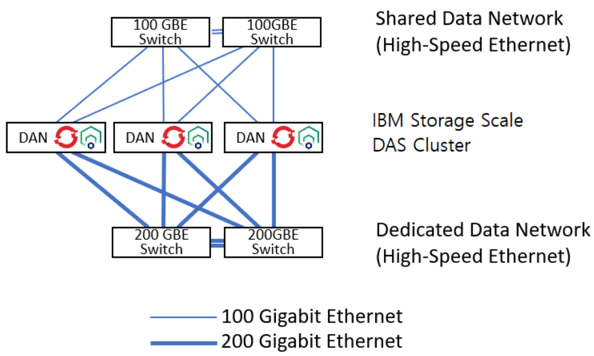

Figure 9. Connecting each DAN with four 100 Gb/s links and with two 200 Gb/s links

Data access node (DAN) requirements

The minimum requirements for a DAN in an IBM Storage Scale DAS on a dedicated Red Hat OpenShift cluster deployment are as follows:

- 16 vCPU
- 64 GB of RAM
- 200 GB of disk space (SSD or NVMe)

The recommended configuration for a DAN in an IBM Storage Scale DAS on a dedicated Red Hat OpenShift cluster deployment are as follows:

- 2x CPU

- 256 GB of RAM
- 400 GB of disk space (NVMe, mirrored using RAID1)
- High-speed network ports for dedicated data network and shared data network

Plan for some head room to have sufficient CPU and memory in case of failure situations, such as the outage of a DAN. The actual required resources are highly dependent on your workload requirements and your chosen network configuration. It is recommended to validate your chosen node configuration in a pre-production environment.

Software requirements

The topic lists IBM Storage Scale DAS 5.1.9.1 software requirements.

Supported software levels:

IBM Storage Scale DAS 5.1.9.1 supports the following software levels:

- Red Hat OpenShift Container Platform (OCP) 4.14.x
- Red Hat OpenShift Data Foundation (ODF) 4.14.x
- IBM Storage Scale container native (CNSA) 5.1.9.1
- IBM Storage Scale Container Storage Interface (CSI) 2.10.0
- IBM Storage Scale 5.1.8 cluster, IBM Storage Scale 5.1.8 (ESS 6.1.8.2)

External container images

There are some external container images that are required to run IBM Storage Scale DAS. If you are running IBM Storage Scale DAS in an air gap environment, these images are required for the successful deployment. For more information, see "Container image list for IBM Storage Scale DAS" on page 20.

Security requirements

To prevent unauthorized access to data that is stored in IBM Storage Scale cluster file systems, it is important to understand how to properly secure various aspects of your IBM Storage Scale DAS deployment.

General security hardening

IBM Storage Scale DAS inherits the in-built security functions of IBM Storage Scale, such as multi-factor authentication for administrative access, audit logging of configuration changes, check summing for data-in-flight between IBM Storage Scale nodes, and replication and erasure coding for data at rest.

Running on Red Hat OpenShift Container Platform, IBM Storage Scale DAS benefits from the in-built security of a modern infrastructure platform including Core OS, Security-Enabled Linux (SELinux), and audit logging of infrastructure events.

All the containers in IBM Storage Scale DAS pods in the `ibm-spectrum-scale-das` namespace run with the non-root user permissions. In addition, all the containers in IBM Storage Scale DAS pods run in the non-privileged mode.

Authentication and ID mapping

IBM Storage Scale DAS uses S3 accounts to secure S3 access. Each S3 account comprises an S3 account name, S3 access keys, a UID, a GID, and other metadata.

Secret access keys are stored encrypted in an internal database. The master encryption key is stored as a Kubernetes secret. When loaded into memory of NooBaa endpoint pods, secrets are wrapped to avoid leaking out to logs.

IBM Storage Scale DAS uses the S3 access keys to identify and authenticate S3 applications. In case of successful authentication of an S3 client, IBM Storage Scale DAS proceeds with authorization. In case of unsuccessful authentication or authorization, the access to data in IBM Storage Scale is denied. IBM Storage Scale DAS does not support unauthenticated (anonymous) S3 access.

Authorization

Authorization in AWS S3 is based on S3 bucket access policies and object ACLs. IBM Storage Scale DAS uses a different approach for authorization to seamlessly integrate S3 access into IBM Storage Scale to support workflows that require multiple access protocols including S3.

Note: IBM Storage Scale DAS 5.1.9.1 supports S3 access only.

IBM Storage Scale DAS uses the standard UNIX access policy based on the user, group, and other permissions, known as Discretionary Access Control (DAC), and Security Enhanced Linux (SELinux) policies, known as Mandatory Access Control (MAC), to secure S3 access to files and directories in IBM Storage Scale.

After successful authentication of an S3 client, IBM Storage Scale DAS looks up the corresponding S3 account's UID and GID from the internal user database and uses them to authorize access to S3 buckets and S3 objects.

In case of S3 read access, IBM Storage Scale DAS enforces the ACLs stored in the IBM Storage Scale file system. Access to S3 buckets and S3 objects is denied when the S3 application has no proper permissions in the IBM Storage Scale file system to access the underlying directories and files.

In case of write access, IBM Storage Scale DAS stores each sent S3 object as file in the IBM Storage Scale file system and sets the owner of the new file to the respective UID and GID of the prior identified and authenticated S3 account. IBM Storage Scale DAS sets the permissions of new files to 660 that allows sharing of S3 objects with other S3 accounts which have the same GID.

Directories can be created by different means. An IBM Storage Scale DAS administrator can create a directory on the IBM Storage Scale storage cluster before creating an S3 export using the **mmdas** command or the IBM Storage Scale DAS REST API. In this case, the administrator is responsible to configure the desired owner and access permissions or ACLs of the new directory using standard Linux and IBM Storage Scale commands.

S3 applications can use the **CreateBucket** S3 API request to create a new S3 bucket. In this case, IBM Storage Scale DAS tries to create a new directory for the new S3 bucket. The creation of a new S3 bucket will fail, in case the respective S3 account does not have the permission in the file system to create the new directory. In case the creation of the new directory is successful, IBM Storage Scale DAS sets the owner of the new directory to the respective UID and GID of the prior identified and authenticated S3 account. IBM Storage Scale DAS sets the permissions of new directory to 770 which allows sharing of S3 buckets with other S3 accounts that have the same GID.

IBM Storage Scale DAS uses the slash (/) as delimiter in object names. When an S3 application uploads an object that has the delimiter in the object name, then IBM Storage Scale DAS creates respective sub directories. In this case, IBM Storage Scale DAS sets the owner of the new sub directory to the respective UID and GID of the prior identified and authenticated S3 account. IBM Storage Scale DAS sets the permissions of new sub directories to 770 which allows sharing of S3 objects that have a delimiter in their object name with other S3 accounts which have the same GID.

In addition, IBM Storage Scale DAS supports usage of SELinux Multi-Category Security (MCS) to confine all IBM Storage Scale DAS processes. IBM Storage Scale DAS inherits SELinux MCS from Red Hat OpenShift that isolates running pods using SELinux MCS by default. If you have SELinux enabled on storage cluster, the deployment procedure of IBM Storage Scale DAS ensures that the SELinux context of IBM Storage Scale DAS pods that access data in IBM Storage Scale match the SELinux context of data in IBM Storage Scale. In this way, other pods and therefore other applications running on the same Red Hat OpenShift cluster by default cannot access the same data in IBM Storage Scale because they run with a different SELinux MCS context.

Protecting data in flight

IBM Storage Scale DAS uses standard methods to secure S3 access on the network layer. S3 object access is protected by SSL certificates.

Clients connect to endpoints over HTTPS and validate the certificate chain up to a well-known root CA to ensure the server identity is authentic. TLS encrypts the data in motion to keep the channel private. TCP checksums the data in motion to detect data corruption over the network.

S3 clients use their secret key to cryptographically sign S3 requests using Signature Version 4 (SigV4) method or Signature Version 2 (SigV2) for backwards compatibility with older clients. For more information, see Signature Version 4 signing process and Signature Version 2 signing process. Request signatures authenticate the sender identity and the request integrity path and headers. This prevents unauthorized requests such as impersonation or tampering.

Clients optionally also sign the request payload by pre-calculating the content checksum and add it as a header to extend the signature coverage to include the payload integrity. Payload checksums are meant to prevent man-in-the-middle content tampering, but data integrity in motion is covered by the network layers.

IBM Storage Scale DAS supports the “Content-MD5” header, which require significant CPU resources from the clients and server. AWS S3 SDK disables payload checksums, if connection is over HTTPS.

IBM Storage Scale DAS calculates and proofs MD5 checksums only if S3 applications send the optional “Content-MD5” header. In case the MD5 checksum sent as value of the HTTP request header “Content-MD5” does not match with the checksum of the data received by IBM Storage Scale DAS, IBM Storage Scale DAS returns an error, which for instance fails a request to write an S3 object. This behavior is in line with the HTTP standard.

Note: IBM Storage Scale DAS has a known issue with the validation of the `Content-MD5` headers. For more information, see “Known issues” on page 116.

Data integrity of responses is typically not checksummed in the API layer and integrity in motion is deferred to the network. For end-to-end data integrity the client is required to explicatively validate the expected data based on pre-calculated checksum that it stored with the data or externally.

Protecting data at rest

Data at rest can be protected against unauthorized access attempts by enforcing file system access permissions and SELinux MCS policies. For more information, see “Authorization” on page 16.

In addition, the security for data at rest can be improved by configuring the IBM Storage Scale storage cluster with encryption, end-to-end checksums for GNR based storage (ESS), file system audit logging, and Security Integration, and Event Management (SIEM) integration to log and detect suspicious activity on the file system.

- File audit logging
- Encryption
- IBM Storage Scale Erasure Code Edition

Roles and persona

Different roles, cluster roles, and levels of access are needed to deploy a fully functioning IBM Storage Scale DAS.

For IBM Storage Scale DAS, same roles and persona are applicable as those for IBM Storage Scale container native. For more information, see Roles and persona (IBM Storage Scale container native).

Persona

The Red Hat OpenShift Cluster administrator must deploy the IBM Storage Scale DAS.

Operator permissions

The IBM Storage Scale DAS 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 Storage Scale DAS.

ibm-spectrum-scale-das-operator role

Resources	Verbs	API Groups
configmaps	get,list,watch,update	-
configmaps/status services/status	get,update,patch	-
endpoints	create,get,list,patch,watch	-
namespaces	create,delete,get,update	-
nodes	get,list,patch,watch,update	-
persistentvolumeclaims, persistentvolumes	create,delete,get,list	-
pods	*	-
secrets	create,delete,get,list,watch	-
serviceaccounts	create,delete,get,list	-
services	create,delete,get,list,patch	-
customresourcedefinitions	*	apiextensions.k8s.io
daemonsets	get,list,watch	apps
deployments deployments/scale statefulsets	*	apps
clusterversions	get,list	config.openshift.io
csiscaleoperators	get,list	csi.ibm.com
leases	get,list, create, update	coordination.k8s.io
s3services,haservices	*	das.scale.ibm.com
s3services/status, haservice/status	get,patch,update	das.scale.ibm.com
ipaddresspools,metallbs	*	metallb.io
l2advertisements	create,delete	metallb.io

Resources	Verbs	API Groups
noobaas, namespacestores	*	noobaa.io
catalogsources, operatorgroups, subscriptions	create,delete,get	operators.coreos.com
clusterserviceversions	get,list,watch	operators.coreos.com
installplans	get,patch	operators.coreos.com
packagemanifests	get,list,watch	packages.operators.coreos.com
podsecuritypolicies, controller, speaker	create,delete,use	policy
clusterrolebindings, clusterroles, rolebindings, roles	*	rbac.authorization.k8s.io
scaleclusters	get,list	scale.ibm.com
clusters, filesystems, remoteclusters	get,list	scale.spectrum.ibm.com
privileged, securitycontextconstraints	get,list,use,watch	security.openshift.io
storageclasses	get,list	storage.k8s.io
networkpolicies	get;list;create;delete	networking.k8s.io

Deployment considerations

You must consider the following for the deployment of IBM Storage Scale DAS.

Considerations for IBM Storage Scale container native

Review the deployment considerations for IBM Storage Scale container native. For more information, see IBM Storage Scale container native deployment considerations.

Considerations for Red Hat OpenShift Container Platform (OCP)

The following Red Hat OpenShift Container Platform (OCP) cluster considerations are in addition to those applicable for IBM Storage Scale container native.

- IBM Storage Scale DAS restricts the configuration options for Red Hat OpenShift. For more information, see “Dedicated Red Hat OpenShift clusters” on page 13.

- IBM Storage Scale DAS uses Red Hat OpenShift MetalLB for the scaling and the high availability of S3 access. The installation of IBM Storage Scale DAS includes the installation and the configuration of the MetalLB feature of Red Hat OpenShift.

Persistent storage for IBM Storage Scale DAS

The following Red Hat OpenShift Container Platform (OCP) cluster persistent volume considerations are in addition to those applicable for IBM Storage Scale container native.

- The IBM Storage Scale DAS implicitly installs an embedded version of Red Hat OpenShift Data Foundation (ODF). ODF includes NooBaa.
- NooBaa requires one local PersistentVolumes (PV) for NooBaa's internal Postgres database. IBM recommends installing this database on an IBM Storage Scale file system.
- This PV must have 50 GB free space created by Noobaa for its internal Postgres database storage and it must be created with the ReadWriteOnce (RWO) access mode.

Container image list for IBM Storage Scale DAS

The installation of IBM Storage Scale DAS requires prior installation of IBM Storage Scale container native and IBM Storage Scale CSI. For information about the containers required for the successful deployment of IBM Storage Scale container native, see Container image list for IBM Storage Scale container native.

IBM Storage Scale DAS includes an embedded version of Red Hat OpenShift Data Foundation (ODF). All images required for the deployment of IBM Storage Scale DAS are sourced from the IBM Cloud Container repository and the Red Hat repository.

Red Hat OpenShift Container Platform (OCP) and OpenShift Data Foundation (ODF) images acquired from Red Hat Container repository

The images listed in the following table are the container images that are obtained through the Red Hat Container repository. They are included with Red Hat OpenShift Container Platform (OCP) version 4.14.x and Red Hat OpenShift Data Foundation (ODF). The Red Hat OpenShift Container Platform (OCP) images are required for IBM Storage Scale DAS. The global Red Hat OpenShift pull secret provides the required permissions to access the Red Hat OpenShift Data Foundation (ODF) images. IBM Storage Scale DAS 5.1.9.1 supports installation and upgrade from OCP 4.12.x.

Table 3. OCP and ODF container images

Pod	Container	Repository	Image
cephcsi-rhel9	cephcsi-rhel9	registry.redhat.io/odf4/	registry.redhat.io/odf4/cephcsi-rhel9@sha256:d82715ef8ec3ba2e501b3a3e735e94c38b96e7e240ba68803e98dee166966117
mcg-core-rhel9	mcg-core-rhel9	registry.redhat.io/odf4/	registry.redhat.io/odf4/mcg-core-rhel9@sha256:cff46f07dc041aa5f75238002edfd856ead5837746dfe3b9ed12c9b087a5f691

Table 3. OCP and ODF container images (continued)			
Pod	**Container**	**Repository**	**Image**
mcg-rhel9-operator	mcg-rhel9-operator	registry.redhat.io/odf4/	registry.redhat.io/odf4/mcg-rhel9-operator@sha256:4b7d43fe5c44ababff41b1fd0bc6ac8f80ca09a13ae80549cfd5ffa389ac5a68
odf-must-gather-rhel9	odf-must-gather-rhel9	registry.redhat.io/odf4/	registry.redhat.io/odf4/odf-must-gather-rhel9@sha256:5387b313ec66b227a5450a285f167005ed191145082e969bb4849f4edca559ff
ocs-rhel9-operator	ocs-rhel9-operator	registry.redhat.io/odf4/	registry.redhat.io/odf4/ocs-rhel9-operator@sha256:5e3624774a3e53020922702d3b2dafb7ead0237c3d11796dbc146d63f2743a1f
odf-console-rhel9	odf-console-rhel9	registry.redhat.io/odf4/	registry.redhat.io/odf4/odf-console-rhel9@sha256:754ac2f10dad7642a8a657da31a9809ca8d2c6381fbfd3e61dc4d65296adae85
rook-ceph-rhel9-operator	rook-ceph-rhel9-operator	registry.redhat.io/odf4/	registry.redhat.io/odf4/rook-ceph-rhel9-operator@sha256:b7665835b58e80400870e54e2a6744ecf22131160566f5b31de88443b3e18258
odf-rhel9-operator	odf-rhel9-operator	registry.redhat.io/odf4/	registry.redhat.io/odf4/odf-rhel9-operator@sha256:31bff2b28ba9d5fbd4a08de914585d3ab4286b90860abb1617fa554017b93beb
odf-csi-addons-sidecar-rhel9	odf-csi-addons-sidecar-rhel9	registry.redhat.io/odf4/	registry.redhat.io/odf4/odf-csi-addons-sidecar-rhel9@sha256:793881fc64ba39d786e0db2e17d075f1d5a7a3f52be832ba15cfdd9c0796cf14

Pod	Container	Repository	Image
odf-csi-addons-rhel9-operator	odf-csi-addons-rhel9-operator	registry.redhat.io/odf4/	registry.redhat.io/odf4/odf-csi-addons-rhel9-operator@sha256:43752953058915f5570da37c164f414e828fa21462f6f63df06754e1acd4afbe
ose-csi-external-attacher	ose-csi-external-attacher	registry.redhat.io/openshift4/	registry.redhat.io/openshift4/ose-csi-external-attacher-rhel8@sha256:0c86d78e8136bf6b6119608727ae6b5a943a5b546e4ac457fa827e724c5ccc2
ose-csi-external-provisioner	ose-csi-external-provisioner	registry.redhat.io/openshift4/	registry.redhat.io/openshift4/ose-csi-external-provisioner@sha256:b453a5c76ba4e975a978e31a51531b1d6233723b0d944622caf7844dedf9ad5a
ose-csi-external-resizer	ose-csi-external-resizer	registry.redhat.io/openshift4/	registry.redhat.io/openshift4/ose-csi-external-resizer@sha256:7ee0257998b7f804fcde9c095b4dc240c510eb316d7223e8485f701b5c9f2fbf
ose-csi-external-snapshotter	ose-csi-external-snapshotter	registry.redhat.io/openshift4/	registry.redhat.io/openshift4/ose-csi-external-snapshotter-rhel8@sha256:06590e3725e496d47eb9893187acb163b1e5a9a0fd3f33d98bb43518176bc27f
ose-csi-node-driver-registrar	ose-csi-node-driver-registrar	registry.redhat.io/openshift4/	registry.redhat.io/openshift4/ose-csi-node-driver-registrar@sha256:caa0bbab808d8cbed476e8fa3e296ceb90f8d7d253e36588fa77e639ea389d55

Table 3. OCP and ODF container images (continued)

Pod	Container	Repository	Image
ose-kube-rbac-proxy	ose-kube-rbac-proxy	registry.redhat.io/ openshift4/	registry.redhat.io/ openshift4/ose-kube-rbac-proxy@sha256:97cade2c1ee468261aec5400728c8d44de387b459134aec7a4c3b5ec5a335d2c registry.redhat.io/ openshift4/ose-kube-rbac-proxy@sha256:1dddb0988d1612c996707d43eb839bc49fc7e7554afaf085436eeddb37a12438
rhceph-6-rhel9	rhceph-6-rhel9	registry.redhat.io/ rhceph/	registry.redhat.io/ rhceph/rhceph-6-rhel9@sha256:3ee8169c13d824d96c0494d5e58d6376f3fa8b947d81cf3e98f722e5d33028e5
postgresql-12	postgresql-12	registry.redhat.io/rhel8/	registry.redhat.io/rhel8/ postgresql-12@sha256:d8c5112a34ad9a5b93223389fbb64e6429f299c1703876b6652c8810fcacd59d
ocs-metrics-exporter-rhel9	ocs-metrics-exporter-rhel9	registry.redhat.io/odf4/	registry.redhat.io/odf4/ ocs-metrics-exporter-rhel9@sha256:76073045c3f55781b59a969201e51f4463ddeb8a20f191fae5ed3801d27581fc
odf-cosi-sidecar-rhel9	odf-cosi-sidecar-rhel9	registry.redhat.io/odf4/	registry.redhat.io/odf4/ odf-cosi-sidecar-rhel9@sha256:82ea95b6380804c86b77f4c1a4186cafe0be2625ef4f79a4def485862669008f
mcg-cli-rhel9	mcg-cli-rhel9	registry.redhat.io/odf4/	registry.redhat.io/odf4/ mcg-cli-rhel9@sha256:755dabb03aefc8235af2679c8a36d0e4d842e2e01c8e31162a490faaa1e88d10

Table 3. OCP and ODF container images (continued)

Note: No user action is required to obtain or define this list of images when in a non air-gapped environment. There are instructions to mirror the list of images in an air gap environment. For more information, see Air-gapped installs for IBM Storage Scale container native.

Red Hat MetalLB images

The following table lists the Red Hat MetalLB images.

Table 4. Red Hat MetalLB images

Pod	Container	Repository	Image
metallb-rhel9-operator	metallb-rhel9-operator	registry.redhat.io/ openshift4/	registry.redhat.io/ openshift4/metallb-rhel9-operator@sha256:f62cb1852e48a3c7e7075e7e6e4cf8561a7e0fe91be5f4435d520808909f35a0
metallb-rhel9	metallb-rhel9	registry.redhat.io/ openshift4/	registry.redhat.io/ openshift4/metallb-rhel9@sha256:d0586171932a1995eebdae11b8968349433d9526da4fbc0dac1d85190c259160
frr-rhel9	frr-rhel9	registry.redhat.io/ openshift4/	registry.redhat.io/ openshift4/frr-rhel9@sha256:ccfef625e8e03cd2d3e7fd4509f6ebe0f77ce2828d5e321b24dc2a82aa3e57ab
ose-kube-rbac-proxy	ose-kube-rbac-proxy	registry.redhat.io/ openshift4/	registry.redhat.io/ openshift4/ose-kube-rbac-proxy@sha256:97cade2c1ee468261aec5400728c8d44de387b459134aec7a4c3b5ec5a335d2c

IBM Storage Scale DAS images

The image listed in the following table does not require entitlement.

Table 5. Container images that do not require entitlement

Pod	Container	Repository	Image
ibm-spectrum-scale-das-operator	das-operator	icr.io/cpopen/ibm-spectrum-scale-das-operator	icr.io/cpopen/ibm-spectrum-scale-das-operator@sha256:9c82e815a75f4ceacd0c3f300c1221993bf6fb5438fc2db1209a4105361bba6b

The images listed in the following table are the container images that are obtained through entitlement.

Pod	Container	Repository	Image
ibm-spectrum-scale-das-endpoint	das-endpoint	cp.icr.io/cp/spectrum/scale/das/s3/ibm-spectrum-scale-das-endpoint	cp.icr.io/cp/spectrum/scale/das/s3/ibm-spectrum-scale-das-endpoint@sha256:5ec18d36e114d1f3938fc740a7c70a5978fe371be9bd847364ae34843c9d62e4
ibm-spectrum-scale-noobaamonitoring	sysmon	cp.icr.io/cp/spectrum/scale/ibm-spectrum-scale-monitor	cp.icr.io/cp/spectrum/scale/ibm-spectrum-scale-monitor@sha256:781f86296525f7519d212f8b1887d1a680157b5ebf55f3a989d76579778abb77
ibm-spectrum-scale-pmsensors	pmsensors	cp.icr.io/cp/spectrum/scale/ibm-spectrum-scale-pmsensors	cp.icr.io/cp/spectrum/scale/ibm-spectrum-scale-pmsensors@sha256:69ae2d99018bd3ce064125677fbd8d00c3829fee64597a83b5da20844f966157
ibm-spectrum-scale-das-operator	kube-rbac-proxy	gcr.io/kubebuilder/kube-rbac-proxy	gcr.io/kubebuilder/kube-rbac-proxy@sha256:d4883d7c622683b3319b5e6b3a7edfbf2594c18060131a8bf64504805f875522

Table 6. Container images that require entitlement

Chapter 4. Installing

Information required before installation and configuration

Before installing and configuring IBM Storage Scale DAS, you must have the following information.

- The name of the storage cluster that owns the IBM Storage Scale file system that is used for IBM Storage Scale DAS. For example,

```
Example storage cluster name: sc42
Example file system name: fs1
```

- Three consecutive IP addresses for IBM Storage Scale DAS S3 access. For example, `192.0.2.12-192.0.2.14`.

  Note: These IP addresses must not be used for any other purpose.

- If SELinux is enabled on the storage cluster, the SELinux MCS labels must be set for securing the stored data.

  Attention: The multi-protocol data sharing feature is supported when SELinux is disabled on the storage cluster.

- A user ID (UID) and a group ID (GID) that will own the object data of the first user to be stored in IBM Storage Scale. For example, UID 1602, GID 1996.
- A user ID (UID) and a group ID (GID) that will own the object data of the second user to be stored in IBM Storage Scale. For example, UID 1606, GID 1996. This account will be used to demonstrate the data sharing with the first S3 account. Both S3 accounts must have different UIDs and the same GID.

Configuring and verifying the installation prerequisites

Use the following steps to configure and verify the prerequisite software components for an IBM Storage Scale DAS deployment.

- IBM Storage Scale storage cluster
- IBM Storage Scale file system
- Red Hat OpenShift Container Platform (OCP) cluster
- IBM Storage Scale container native
- IBM Storage Scale CSI

All steps must be executed in the specified order.

The IBM Storage Scale storage cluster must be installed and configured. For more information about installing and configuring IBM Storage Scale, see IBM Storage Scale documentation.

1. Verify that the IBM Storage Scale storage cluster has the required software version and it is in a healthy state.
   a) From one of the storage cluster nodes, view the software version.

      For example,

```
mmdsh -N all rpm -q gpfs.base
```

      A sample output is as follows:

```
emsdas-hs.test.net:  gpfs.base-5.1.3-0.220203.103103.ppc64le
ess3200das1a-hs.test.net:  gpfs.base-5.1.3-0.x86_64
ess3200das2a-hs.test.net:  gpfs.base-5.1.3-0.x86_64
```

```
ess3200das1b-hs.test.net:  gpfs.base-5.1.3-0.x86_64
ess3200das2b-hs.test.net:  gpfs.base-5.1.3-0.x86_64
```

b) From one of the storage cluster nodes, view the storage cluster health state.

For example,

```
mmhealth cluster show
```

A sample output is as follows:

```
Component            Total         Failed       Degraded        Healthy          Other
--------------------------------------------------------------------------------------
NODE                     5              0              0              5              0
GPFS                     4              0              0              5              0
NETWORK                  4              0              0              4              0
FILESYSTEM               4              0              0              1              0
DISK                    20              0              0             20              0
FILESYSMGR               3              0              0              3              0
GUI                      1              0              0              1              0
NATIVE_RAID              3              0              0              4              0
PERFMON                  4              0              0              4              0
THRESHOLD                4              0              0              4              0
```

To report unauthorized access to S3 buckets and S3 objects, IBM Storage Scale DAS requires that the IBM Storage Scale storage cluster is configured with Security-Enhance Linux (SELinux) in permissive mode.

2. If you plan to enable SELinux on the storage cluster, then enable SELinux in permissive mode on the EMS and ESS I/O nodes. For more information, see Enabling SELinux in ESS.

   You can view the SELinux mode enabled on the storage cluster by issuing the following command from one of the storage cluster nodes.

   ```
   mmdsh -N all getenforce
   ```

   A sample output is as follows:

   ```
   emsdas-hs.test.net:  Permissive
   ess3200das2b-hs.test.net:  Permissive
   ess3200das1a-hs.test.net:  Permissive
   ess3200das2a-hs.test.net:  Permissive
   ess3200das1b-hs.test.net:  Permissive
   ```

The IBM Storage Scale file system used for IBM Storage Scale DAS must be configured in the NFSv4 mode.

3. From one of the storage cluster nodes, verify that the `-D` and `-k` options are set to nfsv4 for the IBM Storage Scale file system that is being used for IBM Storage Scale DAS.

   For example,

   ```
   mmlsfs fs1 -D -k
   ```

   A sample output is as follows:

   ```
   flag                value                    description
   ------------------- ------------------------ -----------------------------------
    -D                 nfs4                     File locking semantics in effect
    -k                 nfs4                     ACL semantics in effect
   ```

IBM Storage Scale CSI requires the quota configuration of IBM Storage Scale file system to be customized. For more information, see Performing pre-installation tasks for CSI Operator deployment.

4. From one of the storage cluster nodes, verify the quota configuration of the IBM Storage Scale file system that is configured for IBM Storage Scale CSI.

   For example,

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

   A sample output is as follows:

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

IBM Storage Scale DAS supports protecting data by using SELinux Multi-Category Security (MCS) labels. Therefore, the IBM Storage Scale file system that is used for IBM Storage Scale DAS can be configured with SELinux MCS labels.

5. If you have enabled SELinux on the storage cluster, then from one of the storage cluster nodes, configure the SELinux MCS labels on the IBM Storage Scale file system.

   a) List the default mount point of the file system.

      For example,

      ```
      mmlsfs fs1 -T
      ```

      A sample output is as follows:

      ```
      flag                value                    description
      ------------------- ------------------------ -----------------------------------
       -T                 /data/fs1                Default mount point
      ```

   b) Set the SELinux MCS labels for the mount point of the file system.

      For example,

      ```
      chcon system_u:object_r:container_file_t:s0:c111,c234 /data/fs1
      ```

   c) List the security context for the mount point of the file system to verify that the SELinux MCS levels are set correctly.

      For example,

      ```
      ls -laZ /data/fs1/
      ```

      A sample output is as follows:

      ```
      total 257
      drwxr-xr-x. 2 root root system_u:object_r:container_file_t:s0:c111,c234 262144 Mar 10
      10:07 .
      drwxr-xr-x  3 root root ?                                                   17 Mar 10
      10:07 ..
      dr-xr-xr-x  2 root root ?                                                 8192 Dec 31
      1969 .snapshots
      ```

IBM Storage Scale DAS requires customers to provide a compact Red Hat OpenShift Container Platform (OCP) cluster. For more information about installing and configuring OCP, see Red Hat OpenShift Container Platform documentation.

6. Verify that the OCP cluster has the required software version and it is in a healthy state.

   a) From a node configured to work with the OCP cluster, view the software version.

      For example,

      ```
      oc get clusterversion
      ```

      A sample output is as follows:

      ```
      NAME      VERSION   AVAILABLE   PROGRESSING   SINCE   STATUS
      version   4.13.13   True        False         14d     Cluster version is 4.13.13
      ```

   b) From a node configured to work with the OCP cluster, view the OCP node status.

      For example,

      ```
      oc get nodes
      ```

      A sample output is as follows:

```
NAME                        STATUS   ROLES           AGE   VERSION
dan1.dasocp4.example.com   Ready    master,worker   21d   v1.22.3+e790d7f
dan2.dasocp4.example.com   Ready    master,worker   21d   v1.22.3+e790d7f
dan3.dasocp4.example.com   Ready    master,worker   21d   v1.22.3+e790d7f
```

IBM Storage Scale DAS requires customers to install and configure IBM Storage Scale container native. For more information about installing and configuring IBM Storage Scale container native cluster, see IBM Storage Scale container native documentation.

7. Verify that the IBM Storage Scale container native cluster has the required software version, all pods are running, and file system is mounted and properly configured for IBM Storage Scale DAS.

   a) From a node configured to work with the OCP cluster, view the IBM Storage Scale container native pods that are running.

      For example,

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

      A sample output is as follows:

      ```
      NAME                                                    READY   STATUS    RESTARTS   AGE
      das-dan1                                                2/2     Running   0          3d13h
      das-dan2                                                2/2     Running   0          3d13h
      das-dan3                                                2/2     Running   0          3d13h
      ibm-spectrum-scale-gui-0                                4/4     Running   0          3d13h
      ibm-spectrum-scale-gui-1                                4/4     Running   0          3d13h
      ibm-spectrum-scale-pmcollector-0                        2/2     Running   0          3d13h
      ibm-spectrum-scale-pmcollector-1                        2/2     Running   0          3d13h
      ```

   b) From a node configured to work with the OCP cluster, view the software version.

      For example,

      ```
      oc -n ibm-spectrum-scale rsh -c gpfs $(oc -n ibm-spectrum-scale get pods -l
      app.kubernetes.io/name=core -o=jsonpath='{.items[0].metadata.name}') mmdiag --version
      ```

      A sample output is as follows:

      ```
      Current GPFS build: "5.1.9.1 ".
      Built on Feb 20 2023 at 14:31:36
      Running 3 days 22 hours 29 minutes 14 secs, pid 1364
      ```

   c) From a node configured to work with the OCP cluster, verify that the IBM Storage Scale file system is mounted on the IBM Storage Scale container native cluster.

      For example,

      i) List the nodes on which the file system is mounted.

         ```
         oc -n ibm-spectrum-scale rsh -c gpfs $(oc -n ibm-spectrum-scale get pods -l
         app.kubernetes.io/name=core -o=jsonpath='{.items[0].metadata.name}') mmlsmount fs1 -L
         ```

         A sample output is as follows:

         ```
         File system fs1 (dasnode1.example.com:fs1) is mounted on 6 nodes:
           192.0.2.64        sc42-n3.example.com       dasnode1.example.com
           192.0.2.78        sc42-n1.example.com       dasnode1.example.com
           192.0.2.27        sc42-n2.example.com       dasnode1.example.com
           198.51.100.212    worker0                   ibm-spectrum-scale.example.com
           198.51.100.134    worker1                   ibm-spectrum-scale.example.com
           198.51.100.161    worker2                   ibm-spectrum-scale.example.com
         ```

      ii) List the file system mount point.

         ```
         oc -n ibm-spectrum-scale rsh -c gpfs $(oc -n ibm-spectrum-scale get pods -l
         app.kubernetes.io/name=core -o=jsonpath='{.items[0].metadata.name}') mmlsfs fs1 -T
         ```

         A sample output is as follows:

```
flag                value                    description
------------------- ------------------------ -----------------------------------
 -T                 /mnt/fs1                 Default mount point
```

iii) If you have enabled SELinux on the storage cluster, then view the security context of the file system.

```
oc -n ibm-spectrum-scale rsh -c gpfs $(oc -n ibm-spectrum-scale get pods -l
app.kubernetes.io/name=core -o=jsonpath='{.items[0].metadata.name}') ls -laZ /mnt/fs1
```

A sample output is as follows:

```
total 257
drwxrwxrwx. 3 root root system_u:object_r:container_file_t:s0 262144 Mar 11 17:31 .
drwxr-xr-x. 3 root root system_u:object_r:var_t:s0                        17 Mar 11
16:12 ..
dr-xr-xr-x. 2 root root system_u:object_r:container_file_t:s0                8192 Jan
1  1970 .snapshots
drwxrwx--x. 3 root root system_u:object_r:container_file_t:s0                4096 Mar
11 17:08 primary-fileset-fs1-1036623172086751852
```

Note: The IBM Storage Scale fileset `primary-fileset-fs1-`nnnnnnnnnnn is implicitly created during the installation of IBM Storage Scale container native. The fileset name includes the name of the remote file system - in this example, it is `fs1`. The nnnnnnnnnnn value refers to the cluster ID of the IBM Storage Scale container native cluster. This fileset is discussed in more detail in the following step.

d) From a node configured to work with the OCP cluster, verify that the IBM Storage Scale file system that is configured for IBM Storage Scale DAS is properly configured by checking the status of the `gpfs`, `remotecluster`, and `filesystem` resources created in the `ibm-spectrum-scale` namespace.

i) Verify that `Status.Conditions.Status` is `True` for the IBM Storage Scale storage cluster that owns the IBM Storage Scale file system configured for IBM Storage Scale DAS.

For example,

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

A sample output is as follows:

```
NAME   HOST                 READY   AGE
sc42   sc42-n1.example.com  True    177m
```

```
oc -n ibm-spectrum-scale describe remoteclusters | grep ^Status -A 7
```

A sample output is as follows:

```
Status:
  Conditions:
    Last Transition Time:  2022-03-11T16:12:01Z
    Message:               The remote cluster has been configured successfully.
    Reason:                AuthCreated
    Status:                True
    Type:                  Ready
Events:
```

ii) Verify that `Status.Conditions.Status` is `True` for the IBM Storage Scale container native cluster.

For example,

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

A sample output is as follows:

```
NAME                 EDITION       AGE
ibm-spectrum-scale   data-access   179m
```

```
oc -n ibm-spectrum-scale describe gpfs ibm-spectrum-scale | grep ^Status -A 7
```

A sample output is as follows:

```
Status:
  Conditions:
    Last Transition Time:  2022-03-11T17:11:07Z
    Message:               The cluster resources have been created successfully.
    Reason:                Configured
    Status:                True
    Type:                  Success
Events:                    <none>
```

iii) Verify that `Status.Conditions.Status` is `True` for the IBM Storage Scaleer file system configured for IBM Storage Scale DAS.

For example,

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

A sample output is as follows:

```
NAME   ESTABLISHED   AGE
fs1    True          3h1m
```

```
oc -n ibm-spectrum-scale describe filesystems fs1 | grep ^Status -A 7
```

A sample output is as follows:

```
Status:
  Conditions:
    Last Transition Time:  2022-03-11T17:11:06Z
    Message:               Filesystem is created.
    Reason:                Created
    Status:                True
    Type:                  Success
  Maintenance Mode:        not supported
```

IBM Storage Scale container native implicitly installs IBM Storage Scale Container Storage Interface (CSI).

8. Verify that the IBM Storage Scale CSI has the required software version and all pods are running.

a) From a node configured to work with the OCP cluster, view the running IBM Storage Scale CSI pods.

For example,

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

A sample output is as follows:

```
NAME                                                READY   STATUS    RESTARTS         AGE
ibm-spectrum-scale-csi-attacher-0                   1/1     Running   20 (138m ago)    179m
ibm-spectrum-scale-csi-attacher-1                   1/1     Running   21 (25m ago)     179m
ibm-spectrum-scale-csi-ff5lh                        3/3     Running   13 (142m ago)    179m
ibm-spectrum-scale-csi-gvwxr                        3/3     Running   13 (142m ago)    179m
ibm-spectrum-scale-csi-operator-78979c7c59-97g8h    1/1     Running   7 (26m ago)      9h
ibm-spectrum-scale-csi-provisioner-0                1/1     Running   0                179m
ibm-spectrum-scale-csi-resizer-0                    1/1     Running   1 (25m ago)      179m
ibm-spectrum-scale-csi-snapshotter-0                1/1     Running   0                179m
ibm-spectrum-scale-csi-vt9f5                        3/3     Running   0                25m
```

b) From a node configured to work with the OCP cluster, view the software version.

For example,

```
oc -n ibm-spectrum-scale-csi get pod -l app=ibm-spectrum-scale-csi
-o=jsonpath='{.items[0].metadata.annotations.productVersion}{"\n"}'
```

A sample output is as follows:

```
2.10.0
```

c) From a node configured to work with the OCP cluster, view the default IBM Storage Scale CSI storage classes.

For example,

```
oc get storageclass
```

A sample output is as follows:

```
NAME                            PROVISIONER                    RECLAIMPOLICY
VOLUMEBINDINGMODE      ALLOWVOLUMEEXPANSION   AGE
ibm-spectrum-scale-internal   kubernetes.io/no-provisioner   Delete
WaitForFirstConsumer   false                  4h38m
ibm-spectrum-scale-sample     spectrumscale.csi.ibm.com      Delete
Immediate              false                  3h2m
```

Note: By default, IBM Storage Scale CSI configures two storage classes.

d) From one of the IBM Storage Scale storage cluster nodes, view the details of the IBM Storage Scale CSI primary fileset.

Note: IBM Storage Scale CSI configures an IBM Storage Scale fileset, referred to as the primary fileset, for IBM Storage Scale CSI internal metadata. For more information, see IBM Storage Scale CSI documentation.

i) List the filesets in the IBM Storage Scale file system configured for IBM Storage Scale DAS.

For example,

```
mmlsfileset fs1 -L
```

A sample output is as follows:

```
Filesets in file system 'fs1':
Name                            Id      RootInode  ParentId Created                      InodeSpace  MaxInodes  AllocInodes Comment
root                             0              3        -- Thu Mar 10 10:07:17 2022      0         615424     503808      root
                                                                                                                          fileset
primary-fileset-fs1-             1         524291         0 Fri Mar 11 09:08:52 2022      1         1048576    55296       Fileset
1036623172086751852                                                                                                       created by IBM
                                                                                                                          Container Storage
                                                                                                                          Interface driver
```

ii) List the details of the primary fileset.

For example,

```
mmlsfileset fs1 primary-fileset-fs1-1036623172086751852
```

A sample output is as follows:

```
Filesets in file system 'fs1':
Name                                     Status    Path
primary-fileset-fs1-1036623172086751852    Linked    /data/fs1/primary-fileset-fs1-1036623172086751852
```

iii) If you have enabled SELinux on the storage cluster, then view the security context of the primary fileset.

For example,

```
ls -laZ /data/fs1/primary-fileset-fs1-1036623172086751852
```

A sample output is as follows:

```
total 258
drwxrwx--x. 3 root root system_u:object_r:unlabeled_t:s0                4096 Mar 11
09:08 .
drwxr-xr-x. 3 root root system_u:object_r:container_file_t:s0:c111,c234 262144 Mar 11
09:31 ..
dr-xr-xr-x. 2 root root system_u:object_r:unlabeled_t:s0                8192 Dec 31
1969 .snapshots
drwxrwx--x. 2 root root system_u:object_r:unlabeled_t:s0                4096 Mar 11
09:08 .volumes
```

Network policy considerations

The `cluster.spec.networkPolicy` object in the CNSA cluster indicates whether the container native operator and DAS operator create and manage the network policy rules within the IBM Storage Scale container native namespaces. The default sample cluster CR in the CNSA cluster setup enables the `networkPolicy: {}` option.

For more information about enabling and disabling network policies, refer the *Network policies* section in the Cluster topic in the IBM Storage Scale container native documentation.

Installing IBM Storage Scale DAS

After configuring and verifying the installation prerequisites, complete the following steps to install IBM Storage Scale DAS in your Red Hat OpenShift Container Platform (OCP) cluster.

To install IBM Storage Scale DAS, you need the manifest file from the GitHub repository.

1. To install IBM Storage Scale DAS, apply the manifest file from the GitHub repository, as shown in the following example:

```
oc apply -f https://raw.githubusercontent.com/IBM/ibm-spectrum-scale-container-native/
v5.1.9.x/generated/das/install.yaml
```

Running the preceding step sets up the Red Hat OpenShift namespace for IBM Storage Scale DAS (`ibm-spectrum-scale-das`) and tries to pull the operator image. The IBM Storage Scale DAS images are pulled from IBM Cloud Container Registry (ICR), using the global pull secret configured to pull IBM Storage Scale container native images. For more information, see Adding IBM Cloud container registry credentials.

2. After some time, the IBM Storage Scale DAS namespace will have three running pods, one IBM Storage Scale operator and two IBM Storage Scale DAS endpoint pods for the management of IBM Storage Scale DAS.
3. From a node configured to work with the OCP cluster, view the details of the `ibm-spectrum-scale-das` namespace.

   For example,

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

   A sample output is as follows:

```
NAME                                                    READY   STATUS    RESTARTS   AGE
ibm-spectrum-scale-das-controller-manager-5778d55476-9mgt9  2/2     Running   0
102s
ibm-spectrum-scale-das-endpoint-696bc8fcb9-k7fcp        1/1     Running   0          67s
ibm-spectrum-scale-das-endpoint-696bc8fcb9-rtkb8        1/1     Running   0          67s
```

   The IBM Storage Scale DAS operator deploys and configures Red Hat OpenShift Data Foundation (ODF). At this stage of the installation process, the IBM Storage Scale DAS operator sets up the namespace for openshift-storage and deploys the initial pods. You can view the details of the openshift-storage namespace as follows:

```
oc -n openshift-storage get pods
```

   A sample output is as follows:

```
NAME                                              READY   STATUS    RESTARTS   AGE
csi-addons-controller-manager-5cf799f75d-wc6g4    2/2     Running   0          3m20s
noobaa-operator-777fd9f598-k9tm6                  1/1     Running   0          3m20s
ocs-metrics-exporter-646b65d57b-pvcwn             1/1     Running   0          3m20s
ocs-operator-6db866c6fd-h5kgj                     1/1     Running   0          3m20s
odf-console-5b96f969cb-xzxxv                      1/1     Running   0          3m20s
odf-operator-controller-manager-6b47f4fb68-6t7ss  2/2     Running   0          3m20s
rook-ceph-operator-5b5c67ff7b-7h45x               1/1     Running   0          3m20s
```

By default, Red Hat OpenShift sets the Security Context Constraints (SCCs) for the new Red Hat OpenShift namespaces. All pods started in a namespace inherit their SCCs from their namespace.

4. If you have enabled SELinux on the IBM Storage Scale cluster, then follow this step. Verify the Red Hat OpenShift SCCs for the openshift-storage namespace.

   For example,

   ```
   oc describe namespace openshift-storage | grep scc
   ```

   A sample output is as follows:

   ```
   Annotations:  openshift.io/sa.scc.mcs: s0:c26,c25
                 openshift.io/sa.scc.supplemental-groups: 1000700000/10000
                 openshift.io/sa.scc.uid-range: 1000700000/10000
   ```

   Note: The example output shows the SCCs for the openshift-storage namespace and its pods after initial IBM Storage Scale DAS installation. The SELinux Multi-Category Security (MCS) labels that are configured for the IBM Storage Scale file system (`s0:c111,c234`) are different MCS labels chosen by Red Hat OpenShift for the SCCs of the openshift-storage namespace and its pods.

   ```
   oc -n openshift-storage get pods -o yaml | grep "level: s"
   ```

   A sample output is as follows:

   ```
   level: s0:c26,c25
   level: s0:c26,c25
   level: s0:c26,c25
   level: s0:c26,c25
   level: s0:c26,c25
   level: s0:c26,c25
   ```

   Note: The Red Hat OpenShift SCCs for SELinux MCS labels of the pods in the openshift-storage namespace must match the SELinux MCS labels that are configured for the IBM Storage Scale file system. You can do this by updating the Red Hat OpenShift SCCs of the openshift-storage namespace and restarting all the pods in the namespace.

5. If you have enabled SELinux on the IBM Storage Scalecluster, then follow this step. Set the Red Hat OpenShift SCC of the openshift-storage namespace to the MCS labels for the IBM Storage Scale file system, which is s0:c11,c324.

   For example,

   ```
   oc annotate namespace openshift-storage --overwrite openshift.io/sa.scc.mcs="s0:c111,c234"
   ```

   a) View the Red Hat OpenShift SCCs of the openshift-storage namespace.

      For example,

      ```
      oc describe namespace openshift-storage | grep scc
      ```

      A sample output is as follows:

      ```
      Annotations:  openshift.io/sa.scc.mcs: s0:c111,c234
                    openshift.io/sa.scc.supplemental-groups: 1000700000/10000
                    openshift.io/sa.scc.uid-range: 1000700000/10000
      ```

      Note: Running pods retain their OpenShift SCCs. Therefore, all pods in the openshift-storage namespace must be terminated, so that they get re-created with the updated Red Hat OpenShift SCCs.

   b) Terminate all pods in the openshift-storage namespace.

      For example,

      ```
      oc -n openshift-storage delete --all pods
      ```

      A sample output is as follows:

```
pod "noobaa-operator-849c98d5fc-pn4mz" deleted
pod "ocs-metrics-exporter-6667498545-xzmjt" deleted
pod "ocs-operator-6bffb7469d-857lb" deleted
pod "odf-console-67cdbb6855-drdtd" deleted
pod "odf-operator-controller-manager-64fcc74877-kbq42" deleted
pod "rook-ceph-operator-7f9fc99d87-dmfpj" deleted
```

c) List all re-created pods in the openshift-storage namespace.

For example,

```
oc -n openshift-storage get pods
```

A sample output is as follows:

```
NAME                                               READY   STATUS    RESTARTS   AGE
csi-addons-controller-manager-5cf799f75d-r8r7s     2/2     Running   0          20s
noobaa-operator-777fd9f598-6vrjx                   1/1     Running   0          20s
ocs-metrics-exporter-646b65d57b-tgmg4              1/1     Running   0          20s
ocs-operator-6db866c6fd-f586t                      1/1     Running   0          20s
odf-console-5b96f969cb-59jsq                       1/1     Running   0          20s
odf-operator-controller-manager-6b47f4fb68-pddtk   2/2     Running   0          20s
rook-ceph-operator-5b5c67ff7b-77jgj                1/1     Running   0          20s
```

d) If you have enabled SELinux on the IBM Storage Scale cluster, then follow this step. Verify that the SCC of the openshift-storage namespace are updated to the IBM Storage Scale MCS labels.

For example,

```
oc -n openshift-storage get pods -o yaml | grep "level: s"
```

A sample output is as follows:

```
        level: s0:c111,c234
        level: s0:c111,c234
        level: s0:c111,c234
        level: s0:c111,c234
        level: s0:c111,c234
        level: s0:c111,c234
```

IBM Storage Scale DAS CLI and REST API require access to the IBM Storage Scale GUI of the IBM Storage Scale container native cluster. This involves configuring an administrator user for IBM Storage Scale DAS in the IBM Storage Scale GUI and a respective secret in the `ibm-spectrum-scale-das` namespace.

6. From a node configured to work with the OCP cluster, configure access to the IBM Storage Scale GUI.

a) Configure an administrator user in the IBM Storage Scale GUI of the IBM Storage Scale container native cluster.

For example,

```
oc -n ibm-spectrum-scale exec -c liberty ibm-spectrum-scale-gui-0 -- /usr/lpp/
mmfs/gui/cli/mkuser s3-admin -p Passw0rd -g 'ProtocolAdmin'
```

A sample output is as follows:

```
EFSSG0019I The user s3-admin has been successfully created.
EFSSG1000I The command completed successfully.
```

b) Configure the secret with the credentials of the administrator user in the IBM Storage Scale DAS namespace.

For example,

```
oc -n ibm-spectrum-scale-das create secret generic das-gui-user --from-
literal=username='s3-admin' --from-literal=password='Passw0rd'
```

A sample output is as follows:

```
secret/das-gui-user created
```

Note: GUI user passwords expire after 90 days by default. Changing these passwords requires you to schedule a short maintenance window for IBM Storage Scale DAS. For more information, see "Changing GUI user passwords" on page 87.

The IBM Storage Scale DAS CLI, **mmdas**, is shipped with the IBM Storage Scale DAS endpoint pods.

7. From a node configured to work with the OCP cluster, install the IBM Storage Scale DAS CLI.

   a) Verify that the IBM Storage Scale DAS endpoint pods are running.

   For example,

```
oc -n ibm-spectrum-scale-das get pods -l app=das-endpoint
```

   A sample output is as follows:

```
NAME                                               READY   STATUS    RESTARTS   AGE
ibm-spectrum-scale-das-endpoint-696bc8fcb9-k7fcp   1/1     Running   0          16m
ibm-spectrum-scale-das-endpoint-696bc8fcb9-rtkb8   1/1     Running   0          16m
```

   b) Copy the IBM Storage Scale DAS CLI from a running `ibm-spectrum-scale-das-endpoint` pod to the node configured to work with the OCP cluster.

   For example,

```
oc cp ibm-spectrum-scale-das/$(oc -n ibm-spectrum-scale-das get pods -l app=das-endpoint
-o=jsonpath='{.items[0].metadata.name}'):mmdas /usr/local/bin/mmdas
```

   c) Make the IBM Storage Scale DAS CLI executable.

   For example,

```
chmod 755 /usr/local/bin/mmdas
```

   The IBM Storage Scale DAS CLI is now ready to use. You can try the `mmdas service list` command to validate that IBM Storage Scale DAS is successfully installed. The command shows that the S3 service is not found. This is expected, because IBM Storage Scale DAS is deployed but not yet configured.

   For example,

```
mmdas service list
```

   A sample output is as follows:

```
Setting up REST API endpoint URL ...
No Service found
```

   If you get an error message such as "`Something went wrong, check the das-endpoint logs`", see "Known issues" on page 116.

   d) To check the product version of the deployed `ibm-spectrum-scale-das` operator, issue the command as follows:

```
oc get deploy ibm-spectrum-scale-das-controller-manager -n ibm-spectrum-scale-das -o json
| jq .metadata.annotations.productVersion
```

   The version of the `ibm-spectrum-scale-das` is shown as follows:

```
"5.1.9.1"
```

Example configuration of IBM Storage Scale DAS

The following steps illustrate an example configuration and key concepts of IBM Storage Scale DAS.

Before you can configure IBM Storage Scale DAS, the configuration of installation prerequisites and the installation of IBM Storage Scale DAS must be completed successfully.

The following steps walk you through an example configuration of the IBM Storage Scale DAS S3 service and accessing data stored in IBM Storage Scale using the S3 access protocol. Customize the following steps according to your workload requirements.

To create and configure the S3 service, you need to accept the license and provide an IP address range for S3 access and the scaling factor.

1. From a node configured to work with the OCP cluster, create and configure the IBM Storage Scale DAS S3 service.

   For example,

   ```
   mmdas service create s3 --acceptLicense --ipRange "192.0.2.12-192.0.2.14" --scaleFactor 1
   ```

   A sample output is as follows:

   ```
   Create request for Spectrum Scale Data Access Service: 's3' is accepted
   ```

   View the status of the IBM Storage Scale DAS S3 service.

   ```
   mmdas service list
   ```

   A sample output is as follows:

   ```
   Name   Enable  Phase
    -----  -----   ------
    s3     true    Creating
   ```

   Note:

   - As the creation and configuration of the IBM Storage Scale DAS S3 service progresses, the status shown in the **Phase** column varies according to the progress of the S3 service configuration.
   - IBM Storage Scale DAS endpoint and NooBaa pods also recycle until the **Phase** column shows the Ready state.
   - Before proceeding with the next steps, administrators must wait for the **Phase** column to show the Ready state and until all fields are populated in the output of the **mmdas service list s3** command.

   After the successful creation of the IBM Storage Scale DAS S3 service, **mmdas service list** reports the status of the S3 service as Ready and **mmdas service list s3** reports status and configuration details.

   ```
   mmdas service list
   ```

   ```
   Name   Enable  Phase
    -----  -----   ------
    s3     true    Ready
   ```

   ```
   mmdas service list s3
   ```

   ```
   Name   AcceptLicense   DbStorageClass                  Enable  EnableMD5
    -----  -----           --------------                  ------  ---------
    s3     true            ibm-spectrum-scale-sample       true    true

    ScaleDataBackend       Phase   S3Endpoints
    -----                  -----   --------------
    [/mnt/fs1]             Ready   [https://192.0.2.12 https://192.0.2.13 https://192.0.2.14]

    IpRange                EnableAutoHA    ScaleFactor
    -----                  -----           --------------
    192.0.2.12-192.0.2.14  true            1
   ```

   The IBM Storage Scale DAS S3 service is now ready to use. For information about how IBM Storage Scale DAS uses resources in Red Hat OpenShift namespaces, see "Understanding Red Hat OpenShift resources used by IBM Storage Scale DAS" on page 47.

Before configuring IBM Storage Scale DAS S3 accounts and S3 exports, validate the IBM Storage Scale DAS configuration.

The S3 service can be accessed through the S3 endpoints shown in the preceding steps. A `curl` command can be used to confirm that the S3 endpoints are accessible. The response will show `Access Denied` that confirms that the S3 service is accessible. Authenticated S3 access is covered in a later step.

2. From a node that can connect to the IBM Storage Scale S3 service IP address, issue an unauthenticated **curl** command to verify access to the S3 service.

   For example,

   ```
   curl 192.0.2.12
   ```

   A sample output is as follows:

   ```
   <?xml version="1.0" encoding="UTF-8"?><Error><Code>AccessDenied</Code><Message>Access
   Denied</Message>
   <Resource>/</Resource><RequestId>l07cquox-6zmwye-ef9</RequestId></Error>
   ```

   S3 accounts are required to authenticate access attempts to the IBM Storage Scale DAS S3 service. To create an S3 account, you need to provide an account name, a UID, and a GID, and optionally a path for new S3 buckets. The account name is used for IBM Storage Scale DAS management purposes, and the UID and the GID are used to store S3 objects in the IBM Storage Scale file system. S3 account creation generates S3 access keys which are used by S3 applications to authenticate access. The configuration of the path for new S3 buckets is shown in a later step.

3. From a node configured to work with the OCP cluster, create an S3 account.

   For example,

   ```
   mmdas account create project1 --uid 1602 --gid 1996
   ```

   A sample output is as follows:

   ```
   Account is created successfully. The secret and access keys are as follows.
   Secret Key                                     Access Key
    ----------                                     -----------
   czAjbq8/CzyMHJfKWvGi50nTRrS4/Id3DA/P3Hau       P71YOPyNAYCdfmIjIuv4
   ```

   ```
   mmdas account list
   ```

   ```
   Name           UID     GID     New buckets path
   ----           ---     ---     ----------------
   project1       1602    1996    /mnt/fs1/
   ```

   ```
   mmdas account list project1
   ```

   ```
   Name        UID    GID    Accesskey               Secretkey                                  New buckets path
   ----        ---    ---    ---------               ---------                                  ----------------
   project1    1602   1996   P71YOPyNAYCdfmIjIuv4    czAjbq8/CzyMHJfKWvGi50nTRrS4/Id3DA/P3Hau     /mnt/fs1/
   ```

   The S3 access keys generated in the preceding step can be used by S3 applications to submit authenticated S3 requests to the S3 service.

   For demonstrative purpose, the S3 command of the AWS command line interface is used in the following step. An alias is created for the AWS CLI that uses the S3 access keys for the S3 service endpoint that are configured in the preceding steps.

   The listing of buckets and objects does not show any results, because no buckets or objects are created so far. The creation of a new S3 bucket fails. This will be resolved in a later step.

4. From a node that can connect to the IBM Storage Scale DAS S3 service IP address, use the S3 account to access the S3 service with the AWS CLI.

```
alias s3p1='AWS_ACCESS_KEY_ID=P71YOPyNAYCdfmIjIuv4 AWS_SECRET_ACCESS_KEY=czAjbq8/
CzyMHJfKWvGi50nTRrS4/Id3DA/P3Hau aws --endpoint https://192.0.2.12 --no-verify-ssl s3'
```

```
s3p1 ls
```

```
s3p1 mb s3://mybucket
```

A sample output is as follows:

```
make_bucket failed: s3://mybucket An error occurred (AccessDenied) when calling the
CreateBucket operation: Access Denied
```

In a preceding step, the bucket creation command by an S3 application failed with the message: `An error occurred (AccessDenied) when calling the CreateBucket operation: Access Denied`. S3 applications use the S3 **`CreateBucket`** request to create new S3 Buckets.

The following step illustrates basic usage of S3 buckets and S3 objects.

For using S3 buckets and S3 objects, the directories in the IBM Storage Scale file system must be configured with proper owner, group, permissions, and SELinux settings. The owner and the group of the directories must match the UID and the GID of the S3 account that is configured in the preceding steps. The owner and the group must have permissions to read, write, and access the directories. The SELinux settings must match the settings that are configured on the storage cluster.

Using S3 buckets and S3 objects

1. From one of the storage cluster nodes, prepare directories in the IBM Storage Scale file system for S3 access.

   a. Create the directories.

   ```
   mkdir /data/fs1/project1-data /data/fs1/project1-buckets
   ```

   b. Assign read and write access to the owner and the group of the directories.

   ```
   chmod 770 /data/fs1/project1-data /data/fs1/project1-buckets
   ```

   c. Change the owner and the group of the directories to match with the UID and GID of the S3 account that is created in a preceding step.

   ```
   chown 1602:1996 /data/fs1/project1-data /data/fs1/project1-buckets
   ```

   d. If you have enabled SELinux on the IBM Storage Scale cluster, then follow this step. Change the SELinux settings for the directories to match with the SELinux settings of the IBM Storage Scale file system configured during installation prerequisites.

   ```
   chcon system_u:object_r:container_file_t:s0:c111,c234 /data/fs1/project1-data /data/fs1/
   project1-buckets
   ```

   You can list the details of the directories including their security context as follows:

   ```
   ls -ldZ  /data/fs1/project1-*
   ```

   A sample output is as follows:

   ```
   drwxrwx---. 2 1602 1996 system_u:object_r:container_file_t:s0:c111,c234 4096 Mar 12
   08:23 /data/fs1/project1-buckets
   drwxrwx---. 2 1602 1996 system_u:object_r:container_file_t:s0:c111,c234 4096 Mar 12
   08:23 /data/fs1/project1-data
   ```

2. From a node configured to work with the OCP cluster, create an S3 export by making the directory accessible as an S3 bucket.

For example,

```
mmdas export create project1-bucket --filesystemPath /mnt/fs1/project1-data
```

A sample output is as follows:

```
Export is successfully created
```

```
mmdas export list
```

A sample output is as follows:

```
Name
 ------
 project1-bucket
```

An S3 application can access such an exported directory as an S3 bucket and, for instance, upload S3 objects.

3. From a node that can connect to the IBM Storage Scale DAS S3 service IP address, use the S3 bucket.
   a. View the AWS CLI alias created in step "4" on page 39 of previous example.

   ```
   alias s3p1
   ```

   A sample output is as follows:

   ```
   alias s3p1='AWS_ACCESS_KEY_ID=P71YOPyNAYCdfmIjIuv4 AWS_SECRET_ACCESS_KEY=czAjbq8/
   CzyMHJfKWvGi50nTRrS4/Id3DA/P3Hau aws --endpoint https://192.0.2.12 --no-verify-ssl s3'
   ```

   b. List the S3 buckets.

   ```
   s3p1 ls
   ```

   A sample output is as follows:

   ```
   2022-03-12 08:35:23 project1-bucket
   ```

   c. Create a file.

   ```
   echo "IBM Spectrum Scale provides scalable performance." > message
   ```

   ```
   md5sum message
   ```

   A sample output is as follows:

   ```
   c927f038344fd0ecfbfa8d69230dc0d4  message
   ```

   d. Copy the file to the S3 bucket.

   ```
   s3p1 cp message s3://project1-bucket
   ```

   A sample output is as follows:

   ```
   upload: ./message to s3://project1-bucket/message
   ```

   e. List the contents of the S3 bucket.

   ```
   s3p1 ls s3://project1-bucket
   ```

   A sample output is as follows:

   ```
   2022-03-12 08:39:40         51 message
   ```

   The uploaded file is listed.

The S3 access protocol has no awareness of the underlying file systems. Therefore, IBM Storage Scale DAS needs to define where to create the directories that represent new S3 Buckets. The `newBucketPath` property of S3 accounts defines for each S3 Account where IBM Storage Scale DAS creates the directories for new S3 Buckets.

The default value for `newBucketPath` is the mount point of the IBM Storage Scale file system on the IBM Storage Scale container native cluster that is used for IBM Storage Scale DAS. The permissions of the root directory are configured in a preceding step and they do not allow users to create new directories. Therefore, the creation of a new S3 bucket failed in a preceding step.

The directory `/data/fs1/project1-buckets` has the required permissions for the S3 account `project1` to create directories. To enable S3 account `project1`, the value of their `newBucketPath` must be updated respectively.

4. From a node configured to work with the OCP cluster, update the value of the `newBucketPath` parameter of an S3 account.

   ```
   mmdas account list
   ```

   A sample output before the update is as follows:

   ```
   Name           UID     GID     New buckets path
    ----           ---     ---     ----------------
    project1       1602    1996    /mnt/fs1/
   ```

   ```
   mmdas account update project1 --newBucketsPath /mnt/fs1/project1-buckets
   ```

   A sample output is as follows:

   ```
   Account is successfully updated
   ```

   ```
   mmdas account  list
   ```

   A sample output after the update is as follows:

   ```
   Name           UID     GID     New buckets path
    ----           ---     ---     ----------------
    project1       1602    1996    /mnt/fs1/project1-buckets/
   ```

   After updating the `newBucketPath` value for the S3 account project1, the account can create new S3 buckets using the S3 **CreateBucket** request.

5. From a node that can connect to the IBM Storage Scale DAS S3 service IP address, create S3 buckets by using the S3 **CreateBucket** request.

   ```
   s3p1 mb s3://mybucket
   ```

   A sample output is as follows:

   ```
   make_bucket: mybucket
   ```

   ```
   s3p1 ls
   ```

   A sample output is as follows:

   ```
   2022-03-12 08:36:04 mybucket
   2022-03-12 08:36:04 project1-bucket
   ```

   From an S3 application's perspective, there is no difference between S3 buckets that are created by using the **mmdas** command and S3 buckets that are created using the S3 **CreateBucket** request. For instance, S3 objects can be seamlessly copied between S3 buckets that are created by using different means.

Note: Due to IBM Storage Scale CNSA SeLinux enablement, the SeLinux type parameter has changed as `unlabeled_t` instead of `container_file_t` for buckets and files created using the **s3** command.

Note: In Red Hat OpenShift Data Foundation (ODF) 4.13, new policies are introduced for sharing buckets across the S3 users that share the same group id (`gid`).

6. From a node that can connect to the IBM Storage Scale DAS S3 service IP address, copy S3 objects between S3 buckets that are created by using different means.

```
s3p1 cp s3://project1-bucket/message s3://mybucket
```

A sample output is as follows:

```
copy: s3://project1-bucket/message to s3://mybucket/message
```

IBM Storage Scale DAS stores S3 buckets and S3 objects as files and directories in IBM Storage Scale file systems. The following command shows the file in the IBM Storage Scale file system for the S3 object that is uploaded to the S3 bucket that is created by using **mmdas** command.

Note: The owner, the group, the permissions, and the SELinux settings for the file are set by IBM Storage Scale DAS.

7. From one of the storage cluster nodes, list the data in the IBM Storage Scale file system that is generated by using the S3 access protocol.

```
tree /data/fs1/project1-data
```

A sample output is as follows:

```
/data/fs1/project1-data
└── message

0 directories, 1 file
```

```
md5sum /data/fs1/project1-data/message
```

A sample output is as follows:

```
c927f038344fd0ecfbfa8d69230dc0d4  /data/fs1/project1-data/message
```

```
ls -lZR /data/fs1/project1-data
```

A sample output is as follows:

```
/data/fs1/project1-data:
total 1
-rw-rw----. 1 1602 1996 system_u:object_r:unlabeled_t:s0 51 Mar 12 08:40 message
```

The following command shows the directory for the S3 bucket that is created by using the S3 **CreateBucket** request and the file for the S3 object that is copied into that S3 bucket.

Note: The owner, the group, the permissions, and the SELinux settings for the file are set by IBM Storage Scale DAS.

```
tree /data/fs1/project1-buckets/
```

A sample output is as follows:

```
/data/fs1/project1-buckets/
└── mybucket
    └── message

1 directory, 1 file
```

```
md5sum /data/fs1/project1-buckets/mybucket/message
```

A sample output is as follows:

```
c927f038344fd0ecfbfa8d69230dc0d4  /data/fs1/project1-buckets/mybucket/message
```

```
ls -lZR /data/fs1/project1-buckets/
```

A sample output is as follows:

```
/data/fs1/project1-buckets/:
total 1
drwxrwx---. 3 1602 1996 system_u:object_r:unlabeled_t:s0 4096 Mar 12 08:39 mybucket

/data/fs1/project1-buckets/mybucket:
total 1
-rw-rw-----. 1 1602 1996 system_u:object_r:unlabeled_t:s0 51 Mar 12 08:39 message
```

This step illustrates basic data sharing between S3 accounts. Both the example S3 accounts have different UIDs, but the same GID. This step also shows how different permissions of directories and files in the file system affect the access of S3 objects and S3 buckets by using the S3 access protocol.

Three different directories are used that are configured with varying owner, group, and permissions:

- Directory `project1-data` is owned by UID `1602` and has permissions `700`. The S3 export of this directory will be accessible for S3 account `project1` only.
- Directory `project2-data` is owned by UID `1606` and has permissions `700`. The S3 export of this directory will be accessible for S3 account `project2` only.
- Directory `shared-data` is owned by GID `1996` and has permissions `770`. The S3 export of this directory will be accessible for both the S3 accounts.

Sharing data between S3 accounts

1. From one of the storage nodes, view the details of the directories that are prepared for S3 access.

   ```
   ls -ladZ /data/fs1/*data
   ```

   A sample output is as follows:

   ```
   drwx------. 3 1602 1996 system_u:object_r:container_file_t:s0:c111,c234 4096 Mar 12 08:40 /
   data/fs1/project1-data
   drwx------. 2 1606 1996 system_u:object_r:container_file_t:s0:c111,c234 4096 Mar 12 10:24 /
   data/fs1/project2-data
   drwxrwx---. 2 1602 1996 system_u:object_r:container_file_t:s0:c111,c234 4096 Mar 12 10:24 /
   data/fs1/shared-data
   ```

   Two S3 buckets are already created that can be reported by using the **mmdas** command. In the following step, create S3 exports for the two additional directories `project2-data` and `shared-data`. Three different directories are being used that are configured with varying owner, group, and permissions.

   Note: S3 exports and the resulting S3 buckets have no awareness of UID, GID, or permissions.

2. From a node configured to work with the OCP cluster, create additional S3 exports.

   ```
   mmdas export list
   ```

   A sample output is as follows:

   ```
    Name
    ------
    mybucket
    project1-bucket
   ```

   ```
   mmdas export create project2-bucket --filesystemPath /mnt/fs1/project2-data
   ```

   A sample output is as follows:

```
Export is successfully created
```

```
mmdas export create shared-bucket --filesystemPath /mnt/fs1/shared-data
```

A sample output is as follows:

```
Export is successfully created
```

```
mmdas export list
```

A sample output is as follows:

```
 Name
 ------
 shared-bucket
 project2-bucket
 mybucket
 project1-bucket
```

Before accessing the new S3 exports as S3 buckets, create a second S3 account `project2` that has a different UID than the S3 account `project1`. Both the S3 accounts have the same GID.

Note: The UIDs and GIDs of both the S3 accounts match the owner and the group of the directories configured in a preceding step.

3. From a node configured to work with the OCP cluster, create the 2nd S3 account.

```
mmdas account create project2 --uid 1606 --gid 1996
```

A sample output is as follows:

```
Account is created successfully. The secret and access keys are as follows.

 Secret Key                                       Access Key
 ----------                                       -----------
 6POQr6s03Dzu1qKHeaJ3/C4XYcQX4EMFawiQMA6O         IG8hr2UoQzgGoNOtV151
```

```
mmdas account list
```

A sample output is as follows:

```
 Name            UID     GID     New buckets path
 ----            ---     ---     ----------------
 project2        1606    1996    /mnt/fs1/
 project1        1602    1996    /mnt/fs1/project1-buckets/
```

The owner, the group, and the permissions of the directories that are accessible as S3 buckets determine which S3 accounts can access which S3 buckets and S3 objects. For instance, the S3 account `project1` can access the S3 buckets `project1-bucket` and `shared-bucket`, and it can copy an S3 object from the S3 bucket `project1-bucket` to the S3 bucket `shared-bucket`. The S3 account `project2` cannot access the S3 bucket `project1-bucket`.

4. From a node that can connect to the IBM Storage Scale DAS S3 service IP address, as account `project1`, access the data that is stored in IBM Storage Scale by using the S3 access protocol.

Note: The alias command used in this step is set up in a preceding step.

```
s3p1 ls
```

A sample output is as follows:

```
2022-03-12 14:53:46 shared-bucket
2022-03-12 14:53:46 mybucket
2022-03-12 14:53:46 project1-bucket
```

```
s3p1 cp s3://project1-bucket/message s3://shared-bucket
```

A sample output is as follows:

```
copy: s3://project1-bucket/message to s3://shared-bucket/message
```

The S3 account `project2` can access the S3 buckets `project2-bucket` and `shared-bucket`, and it can copy an S3 object from the S3 bucket `shared-bucket` to the S3 bucket `project2-bucket`. The S3 account `project2` cannot access the S3 bucket `project1-bucket` and it cannot access S3 objects stored in the S3 bucket `project1-bucket`.

5. From a node that can connect to the IBM Storage Scale DAS S3 service IP address, as account `project2`, access the data that is stored in IBM Storage Scale by using the S3 access protocol.

```
alias s3p2='AWS_ACCESS_KEY_ID=IG8hr2UoQzgGoNOtV151
AWS_SECRET_ACCESS_KEY=6POQr6s03Dzu1qKHeaJ3/C4XYcQX4EMFawiQMA6O aws --endpoint https://
192.0.2.156 --no-verify-ssl s3'
```

A sample output is as follows:

```
AWS_SECRET_ACCESS_KEY=6POQr6s03Dzu1qKHeaJ3/C4XYcQX4EMFawiQMA6O aws --endpoint https://
192.0.2.156 --no-verify-ssl s3'
```

```
s3p2 ls
```

A sample output is as follows:

```
2022-03-12 14:58:04 shared-bucket
2022-03-12 14:58:04 project2-bucket
```

```
s3p2 cp s3://shared-bucket/message s3://project2-bucket
```

A sample output is as follows:

```
copy: s3://shared-bucket/message to s3://project2-bucket/message
```

```
s3p2 ls s3://project2-bucket/message
```

A sample output is as follows:

```
2022-03-12 14:59:58         51 message
```

```
s3p2 ls s3://project1-bucket/message
```

A sample output is as follows:

```
An error occurred (AccessDenied) when calling the ListObjectsV2 operation: Access Denied
```

In the next step, check owner, group, permissions and SELinux settings which are created by using the S3 access protocol. S3 objects are mapped 1:1 to files in the IBM Storage Scale file system. Their owner and their group are derived from the S3 accounts that have created the respective S3 objects. In this way, data can be shared between S3 accounts.

6. From one of the storage nodes, inspect the data in the IBM Storage Scale file system that are created by using the S3 access protocol.

```
ls -lZ /data/fs1/*data/*
```

A sample output is as follows:

```
-rw-rw----. 1 1602 1996 system_u:object_r:unlabeled_t:s0 51 Mar 12 08:40 /data/fs1/project1-
data/message
-rw-rw----. 1 1606 1996 system_u:object_r:unlabeled_t:s0 51 Mar 12 14:59 /data/fs1/project2-
data/message
-rw-rw----. 1 1602 1996 system_u:object_r:unlabeled_t:s0 51 Mar 12 14:54 /data/fs1/shared-
data/message
```

```
md5sum /data/fs1/*data/*
```

A sample output is as follows:

```
c927f038344fd0ecfbfa8d69230dc0d4  /data/fs1/project1-data/message
c927f038344fd0ecfbfa8d69230dc0d4  /data/fs1/project2-data/message
c927f038344fd0ecfbfa8d69230dc0d4  /data/fs1/shared-data/message
```

```
for f in /data/fs1/*data/*; do echo -n "$f - "; cat $f ; done
```

A sample output is as follows:

```
/data/fs1/project1-data/message - IBM Spectrum Scale provides scalable performance.
/data/fs1/project2-data/message - IBM Spectrum Scale provides scalable performance.
/data/fs1/shared-data/message - IBM Spectrum Scale provides scalable performance.
```

Related concepts

"REST API authentication process" on page 137
The REST API services require authentication with a user ID and a password.

"Administering" on page 69
Use the following procedures to manage your S3 object service, S3 user accounts, and S3 exports.

Related reference

"Command reference (mmdas command)" on page 127
The **mmdas** command manages IBM Storage Scale Data Access Services (DAS) service instances, accounts, and exports.

Understanding Red Hat OpenShift resources used by IBM Storage Scale DAS

You can use the following steps to understand the resources in Red Hat OpenShift namespaces that are used by IBM Storage Scale DAS.

oc get pvc

The creation of the IBM Storage Scale DAS S3 service implicitly configures Red Hat OpenShift Data Foundation (ODF), which can be seen by the NooBaa pods running in the Red Hat OpenShift namespace for ODF. The NooBaa endpoint pods of the ODF NooBaa component provide S3 access to data that is stored in IBM Storage Scale. The NooBaa endpoint pods are equally distributed across all Red Hat OpenShift nodes. The scaling factor determines the number of NooBaa endpoint pods that run on each Red Hat OpenShift node.

Note: The use of ODF is restricted to features that can be configured with the IBM Storage Scale DAS management interfaces.

1. From a node configured to work with the OCP cluster, list the pods running in the Red Hat OpenShift namespace for ODF.

   For example,

   ```
   oc -n openshift-storage get pods
   ```

   A sample output is as follows:

   ```
   NAME                                                READY   STATUS    RESTARTS       AGE
   noobaa-core-0                                       1/1     Running   0              33m
   noobaa-db-pg-0                                      1/1     Running   0              33m
   noobaa-default-backing-store-noobaa-pod-fe881f75    1/1     Running   0              31m
   noobaa-endpoint-77647c98b8-hz98j                    1/1     Running   0              27m
   noobaa-endpoint-77647c98b8-kgsqw                    1/1     Running   0              29m
   noobaa-endpoint-77647c98b8-tnzt4                    1/1     Running   0              29m
   noobaa-operator-5b4bb8cb68-tcbch                    1/1     Running   0              53m
   ocs-metrics-exporter-5c9f94ff66-2rjhq               1/1     Running   0              53m
   ocs-operator-d5bcf7ff4-t8btz                        1/1     Running   1 (44m ago)    53m
   odf-console-69d58f5c6d-6fr5p                        1/1     Running   0              53m
   ```

```
odf-operator-controller-manager-d46ffcbf8-hzrnd    2/2     Running   0             53m
rook-ceph-operator-86748bd7cd-qv6gw                1/1     Running   0             53m
```

```
oc -n openshift-storage get pods -l noobaa-s3=noobaa -o wide
```

A sample output is as follows:

```
NAME                                READY   STATUS    RESTARTS   AGE   IP
NODE                           NOMINATED NODE   READINESS GATES
noobaa-endpoint-77647c98b8-hz98j   1/1     Running   0          32m   192.0.2.131
worker2.example.com   <none>           <none>
noobaa-endpoint-77647c98b8-kgsqw   1/1     Running   0          34m   192.0.2.132
worker0.example.com   <none>           <none>
noobaa-endpoint-77647c98b8-tnzt4   1/1     Running   0          34m   192.0.2.177
worker1.example.com   <none>           <none>
```

Red Hat OpenShift Data Foundation creates three physical volumes, one for the NooBaa internal metadata database, one to provide S3 access to data that is stored in IBM Storage Scale, and one that is not required for the integration of ODF in IBM Storage Scale.

2. From a node configured to work with the OCP cluster, list the physical volumes and the physical volume claims created by ODF.

   For example,

```
oc get pv | grep noobaa
```

   A sample output is as follows:

```
noobaa-s3respv-4142975866                  50Gi       RWX
Retain           Bound    openshift-storage/noobaa-s3resvol-
pvc-4142975866                                                            22m
pvc-69775ca5-d2d8-452f-959f-88906a35c6ae   50Gi       RWO            Delete
Bound    openshift-storage/noobaa-default-backing-store-noobaa-pvc-268fb925   ibm-spectrum-
scale-sample              22m
pvc-744e2a15-bf50-436a-9131-cc51b380071f   50Gi       RWO            Delete
Bound    openshift-storage/db-noobaa-db-pg-0                                  ibm-spectrum-
scale-sample              22m
```

```
oc get pvc -n openshift-storage
```

   A sample output is as follows:

```
NAME                                                STATUS
VOLUME                                      CAPACITY   ACCESS MODES
STORAGECLASS                 AGE
db-noobaa-db-pg-0                                   Bound    pvc-744e2a15-bf50-436a-9131-
cc51b380071f   50Gi       RWO            ibm-spectrum-scale-sample   22m
noobaa-default-backing-store-noobaa-pvc-268fb925   Bound    pvc-69775ca5-
d2d8-452f-959f-88906a35c6ae   50Gi       RWO            ibm-spectrum-scale-sample   22m
noobaa-s3resvol-pvc-4142975866                      Bound    noobaa-
s3respv-4142975866                  50Gi       RWX                                       22m
```

The physical volume with the name `noobaa-s3respv-nnnnnnnnnn` represents the IBM Storage Scale file system that is configured for IBM Storage Scale DAS. The nnnnnnnnnn value refers to the cluster ID of the IBM Storage Scale container native cluster. This volume is managed by IBM Storage Scale CSI and claimed by all NooBaa endpoint pods.

3. From a node configured to work with the OCP cluster, list the physical volumes and the physical volume claim for the IBM Storage Scale file system.

   For example,

```
oc describe pv -n openshift-storage noobaa-s3respv-4142975866
```

   A sample output is as follows:

```
Name:            noobaa-s3respv-4142975866
Labels:          <none>
Annotations:     pv.kubernetes.io/bound-by-controller: yes
Finalizers:      [kubernetes.io/pv-protection external-attacher/spectrumscale-csi-ibm-com]
```

```
StorageClass:
Status:          Bound
Claim:           openshift-storage/noobaa-s3resvol-pvc-4142975866
Reclaim Policy:  Retain
Access Modes:    RWX
VolumeMode:      Filesystem
Capacity:        50Gi
Node Affinity:   <none>
Message:
Source:
    Type:              CSI (a Container Storage Interface (CSI) volume source)
    Driver:            spectrumscale.csi.ibm.com
    FSType:
    VolumeHandle:      103662317208675185 2;4E530B0A:622A3E5B;path=/mnt/fs1
    ReadOnly:          false
    VolumeAttributes:  <none>
Events:                <none>
```

```
oc describe pvc -n openshift-storage noobaa-s3resvol-pvc-4142975866
```

A sample output is as follows:

```
Name:          noobaa-s3resvol-pvc-4142975866
Namespace:     openshift-storage
StorageClass:
Status:        Bound
Volume:        noobaa-s3respv-4142975866
Labels:        <none>
Annotations:   pv.kubernetes.io/bind-completed: yes
Finalizers:    [kubernetes.io/pvc-protection]
Capacity:      50Gi
Access Modes:  RWX
VolumeMode:    Filesystem
Used By:       noobaa-endpoint-6f948cc6d8-dg5vb
               noobaa-endpoint-6f948cc6d8-j5qlc
               noobaa-endpoint-6f948cc6d8-mmjp8
Events:        <none>
```

Each NooBaa endpoint pod mounts the IBM Storage Scale file system under the `/nsfs` directory, so that the NooBaa processes can access files and directories in IBM Storage Scale and make them accessible as S3 objects and S3 buckets.

4. From a node configured to work with the OCP cluster, list the details of the IBM Storage Scale file system inside a NooBaa endpoint pod.

   For example,

```
oc -n openshift-storage rsh $(oc -n openshift-storage get pods -l noobaa-s3=noobaa
-o=jsonpath='{.items[0].metadata.name}') ls -laZ /nsfs
```

A sample output is as follows (ignore the security context labels if you have not set MCS labels for the SCC of `openshift-storage` namespace):

```
total 256
drwxrwxrwx. 3 root root system_u:object_r:container_file_t:s0:c111,c234     37 Mar 16 00:37 .
dr-xr-xr-x. 1 root root system_u:object_r:container_file_t:s0:c111,c234     29 Mar 16
00:37 ..
drwxr-xr-x. 9 root root system_u:object_r:container_file_t:s0:c111,c234 262144 Mar 29 09:23
noobaa-s3res-4142975866
```

```
oc -n openshift-storage rsh $(oc -n openshift-storage get pods -l noobaa-s3=noobaa
-o=jsonpath='{.items[0].metadata.name}') ls -laZ /nsfs/noobaa-s3res-4142975866
```

A sample output is as follows (ignore the security context labels if you have not set MCS labels for the SCC of `openshift-storage` namespace):

```
total 260
drwxr-xr-x. 9 root root system_u:object_r:container_file_t:s0:c111,c234 262144 Mar 29 09:23 .
drwxrwxrwx. 3 root root system_u:object_r:container_file_t:s0:c111,c234     37 Mar 16
00:37 ..
dr-xr-xr-x. 2 root root system_u:object_r:unlabeled_t:s0                  8192 Jan  1
1970 .snapshots
drwxrwx--x. 3 root root system_u:object_r:container_file_t:s0:c111,c234   4096 Mar 11 17:08
primary-fileset-fs1-1036623172086751852
```

```
drwxrwx--x. 3 root root system_u:object_r:container_file_t:s0:c111,c234   4096 Mar 13 22:35
pvc-69775ca5-d2d8-452f-959f-88906a35c6ae
drwxrwx--x. 3 root root system_u:object_r:container_file_t:s0:c111,c234   4096 Mar 13 22:33
pvc-744e2a15-bf50-436a-9131-cc51b380071f
```

The creation of the IBM Storage Scale DAS S3 service implicitly installs and configures the Red Hat OpenShift MetalLB feature to provide a highly-available S3 service.

Note: The use of MetalLB is restricted to features that can be configured with the IBM Storage Scale DAS management interfaces.

5. From a node configured to work with the OCP cluster, list the details of the MetalLB configuration.

   For example,

   ```
   oc -n metallb-system get pods
   ```

   A sample output is as follows:

   ```
   NAME                                                   READY   STATUS    RESTARTS       AGE
   controller-66c8949699-m464v                            1/1     Running   0           28m
   metallb-operator-controller-manager-7d9f49cf6-jr4d6    1/1     Running   0           29m
   speaker-7wpf6                                          1/1     Running   0           28m
   speaker-gz4qb                                          1/1     Running   0           28m
   speaker-lrs4k                                          1/1     Running   0           28m
   speaker-q5pvr                                          1/1     Running   0           28m
   speaker-sfcwx                                          1/1     Running   0           28m
   speaker-vx86v                                          1/1     Running   0           28m
   ```

   ```
   oc -n openshift-storage get service
   ```

   A sample output is as follows:

   ```
   NAME                                               TYPE           CLUSTER-IP       EXTERNAL-IP
   PORT(S)                                                                AGE
   das-s3-worker0                                     LoadBalancer   203.0.113.188    192.0.2.12     80:32278/TCP,443:31154/TCP,8444:32446/
   TCP,7004:32679/TCP   2d13h
   das-s3-worker1                                     LoadBalancer   203.0.113.240    192.0.2.13     80:31847/TCP,443:31574/TCP,8444:32386/
   TCP,7004:30302/TCP   2d13h
   das-s3-worker2                                     LoadBalancer   203.0.113.29     192.0.2.14     80:30052/TCP,443:31396/TCP,8444:32075/
   TCP,7004:31465/TCP   2d13h
   noobaa-db-pg                                       ClusterIP      203.0.113.22     <none>
   5432/TCP                                                               2d13h
   noobaa-mgmt                                        LoadBalancer   203.0.113.200    <pending>      80:32129/TCP,443:30781/TCP,8445:32213/
   TCP,8446:30764/TCP   2d13h
   odf-console-service                                ClusterIP      203.0.113.9      <none>
   9001/TCP                                                               3d14h
   odf-operator-controller-manager-metrics-service    ClusterIP      203.0.113.218    <none>
   8443/TCP                                                               3d14h
   s3                                                 LoadBalancer   203.0.113.26     <pending>      80:30713/TCP,443:31174/TCP,8444:31605/
   TCP,7004:32248/TCP   2d13h
   ```

IBM Storage Scale DAS integrates with the IBM Storage Scale management framework. This can be seen by the additional NooBaa monitoring pod running in the Red Hat OpenShift namespace for IBM Storage Scale and the `NOOBAA` line in the output of `mmhealth cluster show` command.

6. From a node configured to work with the OCP cluster, list the details of the IBM Storage Scale DAS integration with the IBM Storage Scale management framework.

   For example,

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

   A sample output is as follows:

   ```
   NAME                                                    READY   STATUS    RESTARTS   AGE
   IP              NODE                               NOMINATED NODE   READINESS GATES
   ibm-spectrum-scale-gui-0                                 4/4     Running   0          70m
   192.0.2.169   worker1.example.com   <none>             <none>
   ibm-spectrum-scale-gui-1                                 4/4     Running   0          72m
   192.0.2.108   worker2.example.com   <none>             <none>
   ibm-spectrum-scale-noobaamonitoring-566b9c6bdb-4697g   1/1     Running   0          31m
   192.0.2.174   worker1.example.com   <none>             <none>
   ibm-spectrum-scale-pmcollector-0                         2/2     Running   0          6h55m
   192.0.2.20    worker0.example.com   <none>             <none>
   ibm-spectrum-scale-pmcollector-1                         2/2     Running   0          6h52m
   192.0.2.167   worker1.example.com   <none>             <none>
   worker0                                                  2/2     Running   0          5h50m
   192.0.2.212   worker0.example.com   <none>             <none>
   worker1                                                  2/2     Running   0          5h50m
   192.0.2.134   worker1.example.com   <none>             <none>
   ```

```
worker2                                              2/2     Running   0          5h49m
192.0.2.161   worker2.example.com   <none>           <none>
```

```
oc -n ibm-spectrum-scale rsh -c gpfs $(oc -n ibm-spectrum-scale get pods -l
app.kubernetes.io/name=core -o=jsonpath='{.items[0].metadata.name}') mmhealth cluster show
```

A sample output is as follows:

```
Component            Total         Failed       Degraded        Healthy          Other
---------------------------------------------------------------------------------------------
--------------------
NODE                     3              0              0              3              0
GPFS                     3              0              0              3              0
NETWORK                  3              0              0              3              0
FILESYSTEM               2              0              0              2              0
CALLHOME                 1              0              0              1              0
GUI                      2              0              0              2              0
HEALTHCHECK              1              0              0              1              0
NOOBAA                   1              0              0              1              0
PERFMON                  3              0              0              3              0
THRESHOLD                3              0              0              3              0
```

The command `mmhealth node show noobaa` displays more details about the NooBaa status. It must be issued from inside the IBM Storage Scale container native pod of the Red Hat OpenShift node that runs the NooBaa monitoring pod. Therefore, first you must determine on which node the NooBaa monitoring pod is running by issuing the `mmhealth cluster show noobaa` command. Thereafter, you can issue the `mmhealth node show` command on the respective Red Hat OpenShift node.

7. From a node configured to work with the OCP cluster, monitor the NooBaa health status.

   For example,

```
oc -n ibm-spectrum-scale rsh -c gpfs $(oc -n ibm-spectrum-scale get pods -l
app.kubernetes.io/name=core -o=jsonpath='{.items[0].metadata.name}') mmhealth cluster show
noobaa
```

   A sample output is as follows:

```
Component     Node        Status            Reasons
---------------------------------------------------
NOOBAA        worker2     HEALTHY             -
```

```
oc -n ibm-spectrum-scale rsh worker2
```

   A sample output is as follows:

```
Defaulted container "gpfs" out of: gpfs, logs, mmbuildgpl (init), config (init)
```

```
mmhealth node show
```

   A sample output is as follows:

```
Node name:      worker2
Node status:    HEALTHY
Status Change:  5 hours ago

Component      Status        Status Change     Reasons & Notices
----------------------------------------------------------------
GPFS           HEALTHY       5 hours ago       -
NETWORK        HEALTHY       5 hours ago       -
FILESYSTEM     HEALTHY       5 hours ago       -
GUI            HEALTHY       1 hour ago        -
NOOBAA         HEALTHY       37 min. ago       -
PERFMON        HEALTHY       5 hours ago       -
THRESHOLD      HEALTHY       5 hours ago       -
```

```
mmhealth node show noobaa
```

   A sample output is as follows:

```
Node name:      worker2

Component     Status        Status Change     Reasons & Notices
----------------------------------------------------------------
NOOBAA        HEALTHY       2 days ago        -

There are no active error events for the component NOOBAA on this node (worker2).
```

```
mmhealth node show noobaa -v
```

A sample output is as follows:

```
Node name:      worker2

Component     Status        Status Change     Reasons & Notices
----------------------------------------------------------------
NOOBAA        HEALTHY       38 min. ago       -

Event                   Parameter     Severity    Active Since             Event Message
-------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------
service_pod_data        NOOBAA        INFO        2022-02-26 00:51:41      The request to
ibm-spectrum-scale-noobaamonitoring-6f5bdbd44d-q8rsk did return health data as expected.
noobaa_api_active       NOOBAA        INFO        2022-02-26 00:51:41      Noobaa Data was
retrieved successfully
active_ns_rsc           NOOBAA        INFO        2022-02-26 00:51:41      Namespace
Resource noobaa-s3res-4080029599 is active in Noobaa
ns_rsc_data_present     NOOBAA        INFO        2022-02-26 00:51:41      Data for Noobaa
Namespace Resources was retrieved successfully
```

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

Air gap environment is set up 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

Refer to the following prerequisites before you set up the air gap environment:

- A production grade Docker V2 compatible registry, such as Quay Enterprise, JFrog Artifactory, or Docker Registry. The Watson™ 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 the `skopeo` utility installed.
- Access to the Red Hat OpenShift Container Platform cluster as a user with the `cluster-admin` role.

Configuring the registry mirror

Create an `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 `Infrastructure` node of your Red Hat OpenShift cluster:

1. Paste the following content 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 you do not have trailing / characters as it can cause issues while resolving them correctly.

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

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

The mirror gets rolled out to all nodes in the Red Hat OpenShift cluster. Nodes are cycled one at a time and are unavailable for scheduling before rebooting.

3. Issue the following command to observe the nodes:

```
watch oc get nodes
```

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

4. After all nodes are updated and restarted, 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"

[[registry]]
  prefix = ""
  location = "registry.redhat.io"
  mirror-by-digest-only = true

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

Copying images from source image registry to target internal image registry

The Red Hat 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 issuing the **skopeo** command:

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

2. Log in to your internal production grade image registry with the credentials by issuing the **skopeo** command:

   ```
   skopeo login example.io
   ```

3. Log in to the Red Hat Container Repository with the credentials by issuing the **skopeo** command:

   ```
   skopeo login registry.redhat.io
   ```

4. For an upgrade, Red Hat OpenShift Container Platform (OCP) version 4.13.x should be used. Use the **skopeo** copy command to copy the following images from the IBM Entitled Container Registry and Red Hat image registry:

   Note: The IBM Storage Scale DAS 5.1.7 to 5.1.9.1 upgrade is done in two phases. Refer to the Upgrading section for more information about these phases of the components wise upgrades.

   - OCP 4.12 to OCP 4.13 (w `metallb-system` and ODF at 4.13.4 GA)
   - OCP 4.13.x to OCP 4.14 (w `metallb-system` and ODF at 4.14.0 GA)

   After the installation of IBM Storage Scale DAS 5.1.9.1, it pulls the following mentioned ODF 4.13.4 images.

   The following images are of IBM Storage Scale DAS 5.1.9.1, ODF 4.13.4, and `metallb-system` (OCP 4.13.x) images:

   ```
   registry.redhat.io/openshift4/frr-
   rhel8@sha256:e5d9049fc3480dffda7ba81fe6f4321b3fef865cd93daa51be6e8376df7f3b1b
   registry.redhat.io/openshift4/metallb-
   rhel8@sha256:4bdd2b158dc2bc67d22b211ff193052688356532c40fe175a6490ac7e040999a
   registry.redhat.io/openshift4/metallb-rhel8-
   operator@sha256:67603c19e65b661377c1d628a37054a629a0398e719b6d6ec61a13b9ebc0107c
   registry.redhat.io/openshift4/ose-kube-rbac-
   proxy@sha256:e33874a51971a90917cf437fc6cbeea1da569ba07217f66f1bba96d594f58aed

   registry.redhat.io/odf4/cephcsi-
   rhel9@sha256:f03e20d8f977c0d8df56f0f273f77783a01a79b08b611628d80e11f54db427e1
   registry.redhat.io/odf4/mcg-core-
   rhel9@sha256:d528db6812dc3c6d5aa4cdee58c3008a4389e4dfceda5b0ce99ca9a1f13cb6af
   registry.redhat.io/odf4/mcg-rhel9-
   operator@sha256:d9158fa378da67f21db6deaf32be49a1148a588754bd207ecd28e0af18f2f135
   registry.redhat.io/odf4/ocs-metrics-exporter-
   rhel9@sha256:9fee67e374cedabb23b770b9f97fdea06df2984087030a5614ab30a8ceb38098

   registry.redhat.io/odf4/ocs-rhel9-
   operator@sha256:55019c2f2fa4e5130fb5a8d6da58dcbf807424f0278a1174ed728f2eb88c94cc
   registry.redhat.io/odf4/odf-console-
   rhel9@sha256:de209057465f3aad32dda80a2872b5edd0de8fe68671bc9275ee0f2e032685fb
   registry.redhat.io/odf4/odf-csi-addons-rhel9-
   operator@sha256:47618497e18163f0c5d5db92515b28ea0cedced6afcfb8a901a4ff7c720c4915
   registry.redhat.io/odf4/odf-csi-addons-sidecar-
   rhel9@sha256:2ec00e3ae8e07f8ef2bd44855ef00483e349757dc676ae805944da05ee353fc9
   registry.redhat.io/odf4/odf-rhel9-
   operator@sha256:fdafdc743a6f161d9482444bc2f70a890dea6fc56426b3f5cf64274e0a84a2d0
   registry.redhat.io/odf4/rook-ceph-rhel9-
   operator@sha256:fffe000b41b61af9f820684dfd78ff4915e646557ffe874d551b49e14c681630

   registry.redhat.io/openshift4/ose-csi-external-attacher-
   rhel8@sha256:7fd8a302d074236870c68bde5704846d613ecd705b6c10788d9c05eb4f245507
   registry.redhat.io/openshift4/ose-csi-external-
   provisioner@sha256:740d29cdfd566fcf095fe5dbd1c47b7e6f8f45703200b331f1711613e76c1311
   registry.redhat.io/openshift4/ose-csi-external-
   resizer@sha256:19e23f3e6e555e4482dfd44fa4d892df060da41056548a5e8293b58dbc01d8d4
   registry.redhat.io/openshift4/ose-csi-external-snapshotter-
   rhel8@sha256:18fc7765f56f84b79bfe02464a75141f3039818c26807575d35b8cfb2bdbd4b6
   registry.redhat.io/openshift4/ose-csi-node-driver-
   registrar@sha256:799ede70bffe6756828749566b49d75d5a1f0e16de7bdddcacfa77adf3418910
   registry.redhat.io/openshift4/ose-kube-rbac-
   proxy@sha256:9cef2fa363236e26022dcf98791cd1d0443127c41be2594db99804d6d74d0318

   registry.redhat.io/rhceph/rhceph-6-
   rhel9@sha256:f83b4a9ae2f3f32df537cafbe15bafe50b2bdee0ba2348d0bda3f2e109160936
   registry.redhat.io/rhel8/
   postgresql-12@sha256:e4847abb2f309f86541684d433c3190c678b5723187c0e8c2757f139d6768ff9

   registry.redhat.io/odf4/odf-must-gather-
   ```

```
rhel9@sha256:ae65763f7c81bdb4401b9af2527c9ff601ca49fa433a6950eec33bf33931aa3b
registry.redhat.io/odf4/mcg-cli-
rhel9@sha256:34058a917de1f6a4fa7ea4351cdd57db032c6cd1b8d7c630e7481cebdd8b9577

cp.icr.io/cp/spectrum/scale/ibm-spectrum-scale-
monitor@sha256:781f86296525f7519d212f8b1887d1a680157b5ebf55f3a989d76579778abb77
cp.icr.io/cp/spectrum/scale/ibm-spectrum-scale-
pmsensors@sha256:69ae2d99018bd3ce064125677fbd8d00c3829fee64597a83b5da20844f966157
icr.io/cp/spectrum/scale/das/s3/ibm-spectrum-scale-das-
endpoint@sha256:5ec18d36e114d1f3938fc740a7c70a5978fe371be9bd847364ae34843c9d62e4
icr.io/cpopen/ibm-spectrum-scale-das-
operator@sha256:9c82e815a75f4ceacd0c3f300c1221993bf6fb5438fc2db1209a4105361bba6b
gcr.io/kubebuilder/kube-rbac-
proxy@sha256:d4883d7c622683b3319b5e6b3a7edfbf2594c18060131a8bf64504805f875522
```

An upgrade in an air gap environment, OCP and ODF upgrade from IBM Storage Scale DAS 5.1.7 to IBM Storage Scale DAS 5.1.9.1 is done in two phases. Use the **skopeo copy** command to copy the following images from the IBM Entitled Container Registry to your internal production grade image registry:

```
cp.icr.io/cp/spectrum/scale/ibm-spectrum-scale-
monitor@sha256:781f86296525f7519d212f8b1887d1a680157b5ebf55f3a989d76579778abb77
cp.icr.io/cp/spectrum/scale/ibm-spectrum-scale-
pmsensors@sha256:69ae2d99018bd3ce064125677fbd8d00c3829fee64597a83b5da20844f966157
icr.io/cp/spectrum/scale/das/s3/ibm-spectrum-scale-das-
endpoint@sha256:5ec18d36e114d1f3938fc740a7c70a5978fe371be9bd847364ae34843c9d62e4
icr.io/cpopen/ibm-spectrum-scale-das-
operator@sha256:9c82e815a75f4ceacd0c3f300c1221993bf6fb5438fc2db1209a4105361bba6b
gcr.io/kubebuilder/kube-rbac-
proxy@sha256:d4883d7c622683b3319b5e6b3a7edfbf2594c18060131a8bf64504805f875522
```

Important: For a fresh air gap installation and for an upgrade (after ODF 4.13 upgrade is performed) Red Hat OpenShift Container Platform version 4.14.x must be used.

The following example shows a sample command to copy the image to IBM Entitled Container Registry and Red Hat registry.:

```
skopeo copy --all docker://cp.icr.io/cp/spectrum/scale/das/s3/ibm-spectrum-scale-das-
endpoint@sha256:ea814626c9c8ab10bfbed1ae91f430d50266fe76a277422d565c492c39e896b4 docker://
example.io/subdir/ibm-spectrum-scale-das-endpoint:v5.1.9.1
```

IBM Storage Scale DAS images are required to be downloaded along with Container Registry for a fresh air gap installation.

For a fresh installation and an upgrade, a generic **skopeo copy** command is shown in the following example:

```
skopeo copy --all docker://<source image registry>/<image_name>@sha256:<digest> docker://
<internal image registry>/<image_name>:<tag>
```

Remember: Any string can be used as a `tag`. It can be used for an upgrade from IBM Storage Scale DAS 5.1.7 to 5.1.9.1 and not for a fresh install. For a fresh installation in an air-gapped environment, you need to directly copy the following images of ODF 4.14.0 and `metallb-system`.

For an installation and an upgrade, Red Hat OpenShift Container Platform (OCP) version 4.14.x should be used. Use the skopeo copy command to copy the following images from the IBM Entitled Container Registry and Red Hat registry:

```
registry.redhat.io/openshift4/frr-
rhel9@sha256:ccfef625e8e03cd2d3e7fd4509f6ebe0f77ce2828d5e321b24dc2a82aa3e57ab
registry.redhat.io/openshift4/metallb-
rhel9@sha256:d0586171932a1995eebdae11b8968349433d9526da4fbc0dac1d85190c259160
registry.redhat.io/openshift4/metallb-rhel9-
operator@sha256:f62cb1852e48a3c7e7075e7e6e4cf8561a7e0fe91be5f4435d520808909f35a0
registry.redhat.io/openshift4/ose-kube-rbac-
proxy@sha256:97cade2c1ee468261aec5400728c8d44de387b459134aec7a4c3b5ec5a335d2c

registry.redhat.io/odf4/odf-cosi-sidecar-
rhel9@sha256:82ea95b6380804c86b77f4c1a4186cafe0be2625ef4f79a4def485862669008f
registry.redhat.io/odf4/mcg-cli-
rhel9@sha256:755dabb03aefc8235af2679c8a36d0e4d842e2e01c8e31162a490faaa1e88d10
```

```
registry.redhat.io/odf4/mcg-core-
rhel9@sha256:cff46f07dc041aa5f75238002edfd856ead5837746dfe3b9ed12c9b087a5f691
registry.redhat.io/rhel8/
postgresql-12@sha256:d8c5112a34ad9a5b93223389fbb64e6429f299c1703876b6652c8810fcacd59d
registry.redhat.io/odf4/mcg-rhel9-
operator@sha256:4b7d43fe5c44ababff41b1fd0bc6ac8f80ca09a13ae80549cfd5ffa389ac5a68

registry.redhat.io/odf4/ocs-rhel9-
operator@sha256:5e3624774a3e53020922702d3b2dafb7ead0237c3d11796dbc146d63f2743a1f
registry.redhat.io/odf4/rook-ceph-rhel9-
operator@sha256:b7665835b58e80400870e54e2a6744ecf22131160566f5b31de88443b3e18258
registry.redhat.io/odf4/cephcsi-
rhel9@sha256:d82715ef8ec3ba2e501b3a3e735e94c38b96e7e240ba68803e98dee166966117
registry.redhat.io/openshift4/ose-csi-node-driver-
registrar@sha256:caa0bbab808d8cbed476e8fa3e296ceb90f8d7d253e36588fa77e639ea389d55
registry.redhat.io/openshift4/ose-csi-external-
resizer@sha256:7ee0257998b7f804fcde9c095b4dc240c510eb316d7223e8485f701b5c9f2fbf
registry.redhat.io/openshift4/ose-csi-external-
provisioner@sha256:b453a5c76ba4e975a978e31a51531b1d6233723b0d944622caf7844dedf9ad5a
registry.redhat.io/openshift4/ose-csi-external-snapshotter-
rhel8@sha256:06590e3725e496d47eb9893187acb163b1e5a9a0fd3f33d98bb43518176bc27f
registry.redhat.io/openshift4/ose-csi-external-attacher-
rhel8@sha256:0c86d78e8136bf6b6119608727ae6b5a943a5b546e4ac457fa827e724c5ccc2f
registry.redhat.io/rhceph/rhceph-6-
rhel9@sha256:3ee8169c13d824d96c0494d5e58d6376f3fa8b947d81cf3e98f722e5d33028e5
registry.redhat.io/odf4/odf-csi-addons-sidecar-
rhel9@sha256:793881fc64ba39d786e0db2e17d075f1d5a7a3f52be832ba15cfdd9c0796cf14
registry.redhat.io/odf4/odf-must-gather-
rhel9@sha256:5387b313ec66b227a5450a285f167005ed191145082e969bb4849f4edca559ff
registry.redhat.io/odf4/ocs-metrics-exporter-
rhel9@sha256:76073045c3f55781b59a969201e51f4463ddeb8a20f191fae5ed3801d27581fc

registry.redhat.io/odf4/odf-csi-addons-rhel9-
operator@sha256:43752953058915f5570da37c164f414e828fa21462f6f63df06754e1acd4afbe

registry.redhat.io/openshift4/ose-kube-rbac-
proxy@sha256:1dddb0988d1612c996707d43eb839bc49fc7e7554afaf085436eeddb37a12438

registry.redhat.io/odf4/odf-rhel9-
operator@sha256:31bff2b28ba9d5fbd4a08de914585d3ab4286b90860abb1617fa554017b93beb
registry.redhat.io/odf4/odf-console-
rhel9@sha256:754ac2f10dad7642a8a657da31a9809ca8d2c6381fbfd3e61dc4d65296adae85
```

For a fresh installation in an air gap environment, the building image registry includes the following IBM Storage Scale DAS images:

```
cp.icr.io/cp/spectrum/scale/ibm-spectrum-scale-
monitor@sha256:781f86296525f7519d212f8b1887d1a680157b5ebf55f3a989d76579778abb77
cp.icr.io/cp/spectrum/scale/ibm-spectrum-scale-
pmsensors@sha256:69ae2d99018bd3ce064125677fbd8d00c3829fee64597a83b5da20844f966157
icr.io/cp/spectrum/scale/das/s3/ibm-spectrum-scale-das-
endpoint@sha256:5ec18d36e114d1f3938fc740a7c70a5978fe371be9bd847364ae34843c9d62e4
icr.io/cpopen/ibm-spectrum-scale-das-
operator@sha256:9c82e815a75f4ceacd0c3f300c1221993bf6fb5438fc2db1209a4105361bba6b
gcr.io/kubebuilder/kube-rbac-
proxy@sha256:d4883d7c622683b3319b5e6b3a7edfbf2594c18060131a8bf64504805f875522
```

The subscription `metallb-operator-sub` in the `metallb-system` namespace, hard codes the name of the source as `redhat-operators`. Correct this if it does not match with the `catalogsource` in the `openshift-marketplace` namespace.

Issue the following command to get `catalogsource` details in the `openshift-marketplace` namespace:

```
oc get catalogsource -n openshift-marketplace
```

A sample output is shown as follows:

```
NAME                           DISPLAY                TYPE            PUBLISHER         AGE
certified-operators           Certified Operators    grpc            Red Hat          114d
community-operators           Community Operators    grpc            Red Hat          114d
redhat-marketplace            Red Hat Marketplace    grpc            Red Hat          114d
redhat-operators-mirrors      Red Hat Operators      grpc            Red Hat          114d
```

To compare the subscription, issue the following command:

```
oc get subscription -n metallb-system metallb-operator-sub -o yaml
```

A sample output is shown as follows:

```
apiVersion: operators.coreos.com/v1alpha1
kind: Subscription
metadata:
 creationTimestamp: "2022-11-29T19:53:24Z"
 generation: 1
 labels:
   operators.coreos.com/metallb-operator.metallb-system: ""
 name: metallb-operator-sub
 namespace: metallb-system
 resourceVersion: "192605960"
 uid: 89d05da4-8864-4fbd-92b0-3a7c70ccd4f3
spec:
 channel: stable
 installPlanApproval: Manual
 name: metallb-operator
 source: redhat-operators-mirrors
 sourceNamespace: openshift-marketplace
```

Use the **oc edit** command to correct the source to match the name in the catalog source as shown in the following example:

```
oc edit subscription -n metallb-system metallb-operator-sub -o yaml
```

Manually approve the `installplan` if you need:

a. Issue the following command to get `installplan`:

```
oc get installplan -n metallb-system
```

b. Update the `installplan` by using the following command:

```
oc patch installplan install-xthtn -n metallb-system -p '{"spec":{"approved":true}}' --
type=merge
installplan.operators.coreos.com/install-xthtn patched
```

c. Verify that the install plan is updated by issuing the following command:

```
oc get installplan -n metallb-system
```

Sample output:

```
NAME            CSV                                      APPROVAL   APPROVED
install-xthtn   metallb-operator.4.12.0-202211231638   Manual     true
```

In the output, install plan APPROVED status is changed to `true`.

5. Log out of the IBM Entitled Container Registry by issuing the **skopeo** command:

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

6. Log out of your internal production grade image registry by issuing the **skopeo** command:

```
skopeo logout example.io
```

7. Log out of the Red Hat Container Repository by issuing the **skopeo** command:

```
skopeo logout registry.redhat.io
```

Testing the pull of images from the mirrored registry

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

1. Pick a worker node from the **oc get nodes** command and start a node to debug it.

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

A command line must be presented.

2. Switch to host binaries by issuing 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. Issue the **podman login** command to authenticate the 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 Red Hat 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/das/s3/ibm-spectrum-scale-das-
endpoint@sha256:86dc1b822a96878b25a70056acfd70b2464ec255ebf7185eae04bfd5db300552
Trying to pull cp.icr.io/cp/spectrum/scale/das/s3/ibm-spectrum-scale-das-
endpoint@sha256:86dc1b822a96878b25a70056acfd70b2464ec255ebf7185eae04bfd5db300552...
Getting image source signatures
Copying blob 778b09db1441 done
Copying blob 1f1cb952eb33 done
Copying blob 477cdcaeeeba done
Copying blob 1f46c5f67b7e done
Copying blob e285ba5d0a41 done
Copying blob ae2197677ae9 done
Copying blob b92a3b17450a done
Copying blob 5f1bbddb713c done
Copying config b73e0ce7d6 done
Writing manifest to image destination
Storing signatures
b73e0ce7d67b4109b2c83e2f75a18ca1048c28331ba5069ed54070fbf483630a
```

5. Verify that the image is pulled as shown in the following example:

```
podman images | grep cp.icr.io/cp/spectrum/scale/das/s3/ibm-spectrum-scale-das-endpoint
cp.icr.io/cp/spectrum/scale/das/s3/ibm-spectrum-scale-das-endpoint     <none>
b73e0ce7d67b  2 days ago    368 MB
```

Red Hat OpenShift Container Registry pull secret

For images to be properly pulled at the pod level, the Red Hat 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`, as shown in the following example:

```
{
  "auth": "<base64 encoded string from previous step>",
```

```
  "username":"<example.io username>",
  "password":"<example.io generated entitlement key>"
}
```

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

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

   ```
   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 by `jq` 1.5.

   - Issue 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 Red Hat OpenShift cluster by issuing the following command:

   ```
   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 that 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 Red Hat OpenShift cluster. Nodes are cycled one at a time and are unavailable for scheduling before rebooting.

5. Issue the **`watch oc get nodes`** command to observe the nodes:

   ```
   oc get nodes
   NAME                                     STATUS   ROLES           AGE   VERSION
   master0.pokprv.stglabs.ibm.com   Ready    master,worker   95d   v1.24.6+5658434
   master1.pokprv.stglabs.ibm.com   Ready    master,worker   95d   v1.24.6+5658434
   master2.pokprv.stglabs.ibm.com   Ready    master,worker   95d   v1.24.6+5658434
   ```

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

6. After the global pull secret is updated, remove the temporary files by issuing the following command:

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

Note: For IBM Storage Scale DAS installation steps, see "Installing IBM Storage Scale DAS" on page 34.

Cleaning up an IBM Storage Scale DAS deployment

Complete the following steps to clean up your IBM Storage Scale DAS deployment by using **mmdas** and **oc** commands.

1. Check whether any exports are configured with the s3 service.

   ```
   mmdas export list
   ```

   A sample output is as follows:

   ```
   Name
    ------
    bucket1
   ```

2. Delete the exports.

   ```
   mmdas export delete bucket1
   ```

   A sample output is as follows:

   ```
   Export is successfully deleted
   ```

3. Check whether any accounts are configured with the s3 service.

   ```
   mmdas account list
   ```

   A sample output is as follows:

   ```
   Name   UID     GID     New buckets path
   ----   ---     ---     ----------------
   user2  1002    101     /mnt/remote-sample/
   user1  1001    101     /mnt/remote-sample/
   ```

4. Delete all the accounts.

   ```
   mmdas account delete user1
   ```

   ```
   mmdas account delete user2
   ```

   A sample output is as follows:

   ```
   Account is successfully deleted
   ```

5. Delete the s3 service.

   ```
   mmdas service delete s3
   ```

   A sample output is as follows:

   ```
   IBM Spectrum Scale DAS service s3 delete request accepted
   ```

6. View the pods in the `openshift-storage` namespace.

   ```
   oc get pods -n openshift-storage
   ```

   Except for the following pods, all the noobaa pods in the `openshift-storage` namespace enter the `Terminating` state and disappear after a while. This state is expected.

   ```
   NAME                                                   READY   STATUS     RESTARTS    AGE
   noobaa-operator-5c46775cdd-tj5fv                       1/1     Running    0           4h36m
   ```

```
ocs-metrics-exporter-5c7f76665f-mhbxc             1/1     Running   0          4h36m
ocs-operator-5b9b9d89c7-4sbjk                     1/1     Running   0          4h36m
odf-console-9b698b47-zgzq5                        1/1     Running   0          4h36m
odf-operator-controller-manager-6cb768f45b-txdfq  2/2     Running   0          4h36m
rook-ceph-operator-866bbcb854-kb2gv               1/1     Running   0          4h36m
```

7. Make sure that the `namespacestore` is deleted.

   ```
   oc get namespacestore -n openshift-storage
   ```

   A sample output is as follows:

   ```
   No resources found in openshift-storage namespace.
   ```

8. Make sure that the `pv` and `pvc` for the `noobaa s3` resource is deleted.

   ```
   oc get pv | grep noobaa-s3
   ```

9. Make sure that the `metallb-system` namespace is deleted.

   ```
   oc get ns | grep metallb-system
   ```

10. Delete the IBM Storage Scale DAS namespace and resources that are created in it.

    ```
    oc delete -f https://raw.githubusercontent.com/IBM/ibm-spectrum-scale-container-native/
    v5.1.9.1/generated/das/install.yaml
    ```

    A sample output is as follows:

    ```
    namespace "ibm-spectrum-scale-das" deleted
    customresourcedefinition.apiextensions.k8s.io "haservices.das.scale.ibm.com" deleted
    customresourcedefinition.apiextensions.k8s.io "s3services.das.scale.ibm.com" deleted
    serviceaccount "ibm-spectrum-scale-das-operator" deleted
    role.rbac.authorization.k8s.io "ibm-spectrum-scale-das-leader-election-role" deleted
    clusterrole.rbac.authorization.k8s.io "ibm-spectrum-scale-das-manager-role" deleted
    clusterrole.rbac.authorization.k8s.io "ibm-spectrum-scale-das-metrics-reader" deleted
    clusterrole.rbac.authorization.k8s.io "ibm-spectrum-scale-das-proxy-role" deleted
    rolebinding.rbac.authorization.k8s.io "ibm-spectrum-scale-das-leader-election-rolebinding"
    deleted
    clusterrolebinding.rbac.authorization.k8s.io "ibm-spectrum-scale-das-manager-rolebinding"
    deleted
    clusterrolebinding.rbac.authorization.k8s.io "ibm-spectrum-scale-das-proxy-rolebinding"
    deleted
    service "ibm-spectrum-scale-das-controller-manager-metrics-service" deleted
    deployment.apps "ibm-spectrum-scale-das-controller-manager" deleted
    ```

11. Delete the **mmdas** CLI binary from the directory where it was copied to after deploying the IBM Storage Scale DAS operator. For example, `/usr/local/bin/mmdas`
12. Delete the `.scaledasenv` in the `$HOME` directory of the administrator.

    For example, the file is in `/root/.scaledasenv` for the root user.

    Note: The `.scaledasenv` file gets created when the **mmdas** CLI is issued for the first time.

Chapter 5. Upgrading

Refer to the following sections to upgrade IBM Storage Scale DAS 5.1.7 to 5.1.9.1.

Considerations while upgrading IBM Storage Scale DAS

The section lists considerations of the underlying components that are involved during the process of upgrade for IBM Storage Scale DAS deployment.

- IBM Storage Scale DAS is deployed on Red Hat OpenShift Container Platform (OCP) clusters that are bare metal, therefore upgrade of IBM Storage Scale container native may take some time on bare metal clusters.
- IBM Storage Scale container native pods and Red Hat OpenShift Container Platform (OCP) nodes will reboot while the upgrade in progress.
- The `noobaa-db` pod in `openshift-storage` namespace depends on IBM Storage Scale container native for provisioning the database storage volume. As the IBM Storage Scale container native pods restart during the upgrade, `noobaa-db` pod might have multiple restarts and remain in "`Init`" state for few minutes before changing its state to "`Running`", as the pod running node restarts when the IBM Storage Scale container native upgrade is in progress.
- Setting the DAS service attribute `enableAutoHA` to `true` minimizes I/O interruptions during the IBM Storage Scale container native upgrade. If the `enableAutoHA` is set to `true`, IP movement is possible during upgrade.
- Check the `noobaa-db` pod status in the `openshift-storage` namespace. Do not create any accounts or buckets while upgrade is in progress. Because the node can restart, which causes the `noobaa-db` pod to move around.
- Red Hat OCP supports one phase of upgrade at a time. As IBM Storage Scale DAS 5.1.9.1 has two phases of OCP and ODF upgrades, which need to be performed offline (without active I/O).

  Also, network policies are added in the IBM Storage Scale DAS 5.1.9.1. Enable the network policy in a cluster after the End-to-End upgrade is completed.

Prerequisite check on `metallb-system` before the DAS upgrade

Check the current `metallb-system` on the current cluster by issuing the following command:

```
oc get installplan -n metallb-system
```

A sample output is shown as follows:

```
NAME            CSV                                   APPROVAL   APPROVED
install-r99kb   metallb-operator.4.12.0-202310311227  Manual     true
install-acde    metallb-operator.4.12.0-202311061916  Manual     false
```

If there is any `metallb-operator` latest version is available in 4.12.x, then do perform the `installplan` patch step as follows, otherwise you can go to "Upgrading IBM Storage Scale DAS 5.1.7 to 5.1.9.1" on page 64 section.

- Above output shows that there is newer version of `metallb-operator` is released after the IBM Storage Scale DAS 5.1.7 is installed. Hence it is required to upgrade to latest available before moving to the upgrade path.

  Issue the following command to perform `metallb-system` upgrade:

  ```
  oc patch installplan -n metallb-system install-<value> --type=merge --patch '{"spec":
  {"approved":true}}'
  ```

After executing the above command, the `metallb-system` namespace restarts only two pods, the `metallb-operator-controller-manager` and `metallb-operator-webhook-server` pods.

For example:

```
oc get pods -n metallb-system -o wide (posted restarted pods)

metallb-operator-controller-manager-6df9f874d9-bvp97   0/1     ContainerCreating   0              2s
<none>         worker0.rkomandu-516upgrade.cp.fyre.ibm.
com   <none>           <none>
metallb-operator-controller-manager-846689d6b-glz7l    1/1     Running             4 (15d ago)    20d
10.254.12.18   worker0.rkomandu-516upgrade.cp.fyre.ibm.
com   <none>           <none>
metallb-operator-webhook-server-698d86d5-hrjkn         0/1     ContainerCreating   0              2s
<none>         worker0.rkomandu-516upgrade.cp.fyre.ibm.
com   <none>           <none>
metallb-operator-webhook-server-74d85f8685-f5cnd       1/1     Running             0              20d
10.254.16.9    worker2.rkomandu-516upgrade.cp.fyre.ibm.
com   <none>           <none>
```

Upgrading IBM Storage Scale DAS 5.1.7 to 5.1.9.1

IBM Storage Scale DAS 5.1.7 is based on OCP 4.12 and ODF 4.12.x. IBM Storage Scale DAS 5.1.9.1 supports OCP 4.14.x and ODF 4.14.x, hence the upgrade must be done by using the following two phases:

- **Phase 1**

  1. **Upgrading IBM Storage Scale container native**

     Upgrade the IBM Storage Scale container native to version 5.1.9.1. Before you attempt to upgrade IBM Storage Scale DAS, see the ODF 4.13.x upgrade issue in the Known issues.

     For more information, see Upgrading IBM Storage Scale container native.

     Attention:

     – The IBM Storage Scale DAS solution as of today does not ingest data from outside the Red Hat OpenShift Container Platform (OCP), hence it is recommended to use the include operator step provided in the IBM Storage Scale container native documentation in the upgrade section.

     – Do not apply the patch to the cluster CR to enable network policies because network policies will be enabled after the upgrade is completed. Complete remaining steps in the *Approve the new release* section in the Post upgrade tasks topic.

  2. **Upgrading Red Hat OpenShift Container Platform (OCP)**

     For Red Hat OpenShift Container Platform (OCP) upgrade 4.12.x to 4.13.x, administrators are required to check the Red Hat documentation for upgrading the Red Hat OpenShift Container Platform (OCP) cluster across release for any dependencies.

     Remember: Kubernetes has API changes in Red Hat OpenShift Container Platform (OCP) 4.13 as compared to previous versions.

  3. **Upgrading IBM Storage Scale DAS**

     To upgrade IBM Storage Scale DAS, perform the following steps:

     a. Ensure that the IBM Storage Scale DAS 5.1.7 is based on OCP 4.12.x and ODF 4.12.x.

        i) Check IBM Storage Scale DAS version by using the following command:

        ```
        oc get deploy ibm-spectrum-scale-das-controller-manager -n ibm-spectrum-scale-das -o
        json | jq .metadata.annotations.productVersion
        ```

        ii) Check Red Hat OpenShift Data Foundation (ODF) version by using the following command:

        ```
        oc get csv -n openshift-storage
        ```

b. Apply the IBM Storage Scale DAS operator yaml file by using the following command:

```
oc apply -f https://raw.githubusercontent.com/IBM/ibm-spectrum-scale-container-native/
v5.1.9.1/generated/das/install.yaml
```

c. Wait for the pod restart in the `ibm-spectrum-scale-das` and in the `openshift-storage` namespaces.

d. After IBM Storage Scale DAS 5.1.9.1 is installed, it automatically upgrades the `metallb-system` from 4.12.x to 4.13.x as shown the following example (because OCP is already at 4.13.x).

```
oc get installplan -n metallb-system
NAME          CSV                                   APPROVAL APPROVED
install-pd4z5 metallb-operator.v4.13.0-202311031531 Manual   true
install-r99kb metallb-operator.4.12.0-202310311227  Manual   true
```

e. The ODF is upgraded automatically from 4.12.x to 4.13.x when IBM Storage Scale DAS pods restart after install yaml is applied in step b.

Note: You might encounter an issue in the `rook-ceph-operator` pod in the `CreateContainerConfigError` state and in the `openshift-storage` namespace. For more information, refer to the Known issues.

```
oc get pods -n openshift-storage
rook-ceph-operator-6777844fcd-zw8b8 0/1 CreateContainerConfigError 0 63s 10.254.16.8
worker2.rkomandu-517-upgrade.cp.fyre.ibm.com
<none> <none>
```

```
oc get csv -n openshift-storage
rook-ceph-operator-6777844fcd-zw8b8 0/1 CreateContainerConfigError 0 63s 10.254.16.8
worker2.rkomandu-517-upgrade.cp.fyre.ibm.com
<none> <none>
```

f. After the ODF is upgraded to the latest 4.13.x, the noobaa pods in the `openshift-storage` (`noobaa-db/noobaa-core/noobaa-endpoint`) also restart.

```
oc get csv -n openshift-storage
```

An example of the ODF 4.13.4 is as follows:

```
NAME                                    DISPLAY                      VERSION
REPLACES                                PHASE
mcg-operator.v4.13.4-rhodf              NooBaa Operator              4.13.4-rhodf
mcg-operator.v4.12.9-rhodf              Succeeded
metallb-operator.v4.13.0-202311031531 MetalLB Operator             4.13.0-202311031531
metallb-operator.4.12.0-202310311227    Succeeded
ocs-operator.v4.13.4-rhodf              OpenShift Container Storage 4.13.4-rhodf
ocs-operator.v4.12.9-rhodf              Succeeded
odf-csi-addons-operator.v4.13.4-rhodf CSI Addons                   4.13.4-rhodf
odf-csi-addons-operator.v4.12.9-rhodf Succeeded
odf-operator.v4.13.4-rhodf              OpenShift Data Foundation   4.13.4-rhodf
odf-operator.v4.12.9-rhodf              Succeeded
```

4. **Verifying IBM Storage Scale DAS upgrade**

Perform the following steps to verify IBM Storage Scale DAS version:

a. After the IBM Storage Scale DAS pods have restarted, issue the following command to get the name of the controller-manager pod:

```
oc get pod -n ibm-spectrum-scale-das | grep controller-manager
```

To check the current version, issue the command with name of the controller-manager pod as shown in the following example:

```
oc get deploy ibm-spectrum-scale-das-controller-manager -n ibm-spectrum-scale-das -o
json | jq .metadata.annotations.productVersion
```

A sample output is shown as follows:

```
"productVersion": "5.1.9.1"
```

During upgrade from IBM Storage Scale DAS 5.1.7 to IBM Storage Scale DAS 5.1.9.1, the following are observed `metallb-system` namespace and Red Hat OpenShift Data Foundation version are updated to 4.13.x to be in-line with the underlying OCP 4.13.x environment.

- **Phase 2**

  1. **Upgrading Red Hat OpenShift Container Platform (OCP) 4.13.x to OCP 4.14.x**

     For Red Hat OpenShift Container Platform (OCP) upgrade 4.13.x to 4.14.x, you must check the Red Hat documentation for upgrading the Red Hat OpenShift Container Platform (OCP) cluster across releases.

     Remember: Kubernetes has API changes in Red Hat OpenShift Container Platform (OCP) 4.14 as compared to previous versions.

     After OCP is upgraded, you can see the `metallb-system` namespace is also upgraded to the 4.14.0 as shown in the following example:

     ```
     oc get installplan -n metallb-system
     ```

     A sample output is as follows:

     ```
     NAME           CSV                                        APPROVAL  APPROVED
     install-2lfrd  metallb-operator.v4.14.0-202311101708  Manual    true
     install-9j9xr  metallb-operator.v4.13.0-202311031531  Manual    true
     install-pckm4  metallb-operator.v4.13.0-202311142207  Manual    true
     ```

  2. **Upgrade the ODF 4.13.x to ODF 4.14.x**

     By restarting the `das-controller-manager` pod in the `ibm-spectrum-scale-das` namespace the ODF upgrade is triggered automatically. Check the pods restart in the `openshift-storage` namespace.

     For example,

     ```
     oc get csv -n openshift-storage
     ```

     A sample output is as follows:

```
NAME                                       DISPLAY                      VERSION
REPLACES                                   PHASE
mcg-operator.v4.14.0-rhodf                 NooBaa Operator              4.14.0-rhodf         mcg-operator.v4.13.4-
rhodf              Succeeded
metallb-operator.v4.14.0-202311101708 MetalLB Operator             4.14.0-202311101708 metallb-
operator.v4.13.0-202311142207 Succeeded
ocs-operator.v4.14.0-rhodf                 OpenShift Container Storage 4.14.0-rhodf         ocs-operator.v4.13.4-
rhodf              Succeeded
odf-csi-addons-operator.v4.14.0-rhodf CSI Addons                   4.14.0-rhodf         odf-csi-addons-operator.v4.13.4-
rhodf Succeeded
odf-operator.v4.14.0-rhodf                 OpenShift Data Foundation   4.14.0-rhodf         odf-operator.v4.13.4-
rhodf              Succeeded
```

Post upgrade steps

Perform the following steps after the upgrade is complete. If want to enable network policies that are shipped with IBM Storage Scale container native (CNSA) 5.1.9.1, refer the *Apply the patch to the Cluster CR to enable network policies* in the Post upgrade tasks topic in the IBM Storage Scale container native documentation. Otherwise, copy the **mmdas** command in the Copying **mmdas** command section.

Complete the following steps after the network policies are enabled in the IBM Storage Scale container native:

1. Delete all pods in the operator namespace.

   ```
   oc delete pods --all -n ibm-spectrum-scale-operator
   ```

2. Delete pods in the IBM Storage Scale DAS namespace.

```
oc delete pods --all -n ibm-spectrum-scale-das
```

3. Check the network policy of the IBM Storage Scale DAS namespace.

```
oc get networkpolicy -n ibm-spectrum-scale-operator
oc get networkpolicy -n ibm-spectrum-scale
oc get networkpolicy -n ibm-spectrum-scale-dns
oc get networkpolicy -n ibm-spectrum-scale-das
```

The IBM Storage Scale DAS 5.1.7 to 5.1.9.1 upgrade is completed. Now let us move to the extraction of the **mmdas** command.

Copying the mmdas command

Perform the following steps to copy the **mmdas** command and make it executable:

1. Verify that the IBM Storage Scale DAS endpoint pods are running, by issuing the following command:

```
oc -n ibm-spectrum-scale-das get pods -l app=das-endpoint
```

A sample output is shown as follows:

```
NAME                                              READY   STATUS    RESTARTS   AGE
ibm-spectrum-scale-das-endpoint-696bc8fcb9-k7fcp  1/1     Running   0          16m
ibm-spectrum-scale-das-endpoint-696bc8fcb9-rtkb8  1/1     Running   0          16m
```

2. Rename the **/usr/local/bin/mmdas** command to the **/usr/local/bin/mmdas.old** command.

```
mv /usr/local/bin/mmdas /usr/local/bin/mmdas.old
```

3. Copy the IBM Storage Scale DAS CLI from a running `ibm-spectrum-scale-das-endpoint` pod to the node configured to work with the Red Hat OpenShift Container Platform (OCP) cluster by issuing the following command:

```
oc cp ibm-spectrum-scale-das/$(oc -n ibm-spectrum-scale-das get pods -l app=das-endpoint
-o=jsonpath='{.items[0].metadata.name}'):mmdas /usr/local/bin/mmdas
```

4. Make the IBM Storage Scale DAS CLI executable by issuing the following command:

```
chmod 755 /usr/local/bin/mmdas
```

Remove the `.scaledasenv` before the issuing the **mmdas** command:

```
rm -rf ~/.scaledasenv
```

The IBM Storage Scale DAS CLI is now ready to use.

For example:

```
mmdas service list
```

A sample output is as follows:

```
# mmdas service list
Name Enable Phase
----- ----- ------
s3    true  Ready
```

5. Remove the `mmdas.old` command.

```
rm -rf /usr/local/bin/mmdas.old
```

This is a complete End-to-End upgrade process of IBM Storage Scale DAS.

Chapter 6. Administering

Use the following procedures to manage your S3 object service, S3 user accounts, and S3 exports.

Managing S3 object service instance

Use the CLI or the API to manage your S3 object service instance.

You must have the following details before you can create an S3 object service instance.

- Set up the high availability option, the range of 3 IP addresses and the number of IBM Storage Scale DAS labeled nodes on which the `noobaa` endpoints can scale to.
- The name of the storage class to configure a database for the S3 service. If you do not specify this parameter, the default storage class is used.

  Note: The **`dbStorageClass`** parameter is optional. The IBM Storage Scale DAS operator selects the storage classes defined on the OCP cluster by using `spectrumscale.csi.ibm.com`, if there is only one such storage class. If there are more than one storage classes defined on the OCP cluster using `spectrumscale.csi.ibm.com` as the provisioner, the DAS operator cannot automatically select one of those to configure the S3 service with. In such a scenario, you need to specify which of those storage classes must be used to configure the S3 service.
- The name of the IBM Storage Scale file system that acts as the data backend for access by using the S3 object service interface. If it is not specified, the default file system that is mounted on the IBM Storage Scale container native pods would be automatically detected and used.

Use the following information to create, list, delete, or update your S3 object service instance.

- Create an IBM Storage Scale DAS S3 object service instance as follows:
  - CLI

    ```
    mmdas service create s3 --acceptLicense --ipRange "192.0.2.12-192.0.2.14" --scaleFactor 1
    ```

    Note: If we need to increase the number of noobaa endpoints (for example, `scaleFactor`), ensure to have sufficient memory on the system (for example, memory on DAN nodes).

    A sample output is as follows:

    ```
    Create request for Spectrum Scale Data Access Service: 's3' is accepted
    ```

    In these command examples, the following parameters are specified:

    - License acceptance

      Note: The `--acceptLicense` flag is mandatory to create the S3 service. Using this flag is required to register the acceptance to the IBM Storage Scale Data Access Services (DAS) license before you deploy the service. Before deploying the service, carefully read the terms and conditions of the license. For more information, see terms and conditions of the license.
    - Range of IP addresses for high availability configuration

      Note: The IP range can be set up only at the service creation time. Use the IP range to enable the S3 service access over the specified range of IP addresses. These IP addresses can be configured with an external DNS whose domain name can be used by the S3 client applications to access the storage over S3 protocol by using that DNS URL.

      Range of IP addresses has the following requirements:

      - It must be in the format: `x.x.x.x-x.x.x.x`
      - It must be in a sequence. For example, 192.0.2.12-192.0.2.14

- It must match the number of OCP nodes which are labeled for IBM Storage Scale usage; nodes that have the `scale=true` label. You can check the number of nodes that have the `scale=true` label by issuing the following command:

```
oc get nodes --show-labels | grep scale=true
```

- Number of nodes on which the noobaa endpoints can scale to.

Note: Select a `scaleFactor` according to your requirements at the time of creating the service because the `scaleFactor` must not be changed during active I/O.

- REST API

```
curl -k -X POST -k -H "Content-Type: application/json" -H "Authorization: Basic
czMtYWRtaW46UGFzc3cwcmQ=" https://<ibm-spectrumscale_host>/scalemgmt/v2/das/services/
-d '{"name":"s3","enable":true,"acceptLicense":true,
"ipRange":"192.0.2.12-192.0.2.14",
"scaleFactor":1}'
```

A sample output is as follows:

```
{"message":"Create request for Spectrum Scale Data Access Service: 's3' is accepted"}
```

Note: The variable *<ibm-spectrumscale_host>* in the request URL must be replaced with the route host. Obtain the route host by using the following command from a node that is configured to work with the Red Hat OpenShift Container Platform (OCP) cluster:

```
oc get route ibm-spectrum-scale-gui -n <IBM Storage Scale namespace> -o json |
jq .spec.host
```

For example,

```
oc get route ibm-spectrum-scale-gui -n ibm-spectrum-scale -o json | jq .spec.host
```

A sample output is as follows:

```
"ibm-spectrum-scale-gui-ibm-spectrum-scale.example.com"
```

- List the information of the IBM Storage Scale DAS service instance as follows:
  - CLI

```
mmdas service list
```

A sample output is as follows:

```
Name   Enable  Phase
-----  -----   ------
s3     true    Ready
```

- The **Enable** column shows whether the S3 service instance is enabled or disabled.
- The deployment phase of the service instance shown in the **Phase** column can be one of the following values:
  - **Ready:** The service instance is ready to be used for S3 account creation or export creation.
  - **Configuring:** The service instance configuration is in progress.
  - **Connecting:** The service instance is trying to establish communication between the S3 endpoints and the S3 database.
  - **Failed:** The service instance configuration has failed.

Attention: Once you issue the service creation command, for a brief period of time, the **Phase** column might be empty.

To list the detailed information for the IBM Storage Scale DAS S3 object service instance, issue the following command:

```
mmdas service list s3
```

A sample output is as follows:

```
Name   AcceptLicense   DbStorageClass                 Enable  EnableMD5
-----  -----           --------------                 ------
s3     true            ibm-spectrum-scale-sample      true    true

ScaleDataBackend       Phase   S3Endpoints
-----                  -----   --------------
[/mnt/remote-sample]   Ready   [https://192.0.2.12 https://192.0.2.13 https://192.0.2.14]

IpRange                EnableAutoHA    ScaleFactor
-----                  -----           --------------
192.0.2.12-192.0.2.14    true            1
```

- REST API

```
curl -k -X GET -H "accept: application/json" https://<ibm-spectrumscale_host>/
scalemgmt/v2/das/services/s3
```

A sample output is as follows:


```
{
"acceptLicense" : true,
"dbStorageClass" : "ibm-spectrum-scale-sample",
"enable" : true,
"enableAutoHA" : false,
"enableMD5" : false,
"ipRange" : "192.0.2.12-192.0.2.14",
"name" : "s3",
"phase" : "Ready",
"s3Endpoints" : [ "https://192.0.2.12", "https://192.0.2.13", "https://192.0.2.14" ],
"scaleDataBackend" : [ "/mnt/remote-sample" ],
"scaleFactor" : 1
}
```


- Update the IBM Storage Scale DAS service instance as follows:
  - CLI

```
mmdas service update s3 --enableMD5 --disableAutoHA --scaleFactor 2
```

This command enables md5sum calculation, disables automatic IP address failover and failback, and changes the `scaleFactor` to 2. A sample output is as follows:

```
Update request for Spectrum Scale Data Access Service: 's3' is accepted
```

  - REST API

```
curl -X PUT -H "Content-Type: application/json" -H "Authorization: Basic
czMtYWRtaW46UGFzc3cwcmQ=" https://<ibm-spectrumscale_host>/scalemgmt/v2/das/services/
-d '{"name":"s3","enableMD5":true,"enableAutoHA":false,"scaleFactor":2}'
```

A sample output is as follows:

```
{"message":"Update request for Spectrum Scale Data Access Service: 's3' is accepted"}
```

Note: You must not change the `scaleFactor` during active I/O, otherwise I/O failure might occur. Change the `scaleFactor` during a maintenance window when there is no active I/O. For more information, see "Changing scaleFactor might result in I/O failure" on page 118.

- Delete the IBM Storage Scale DAS service instance as follows:
  - CLI

```
mmdas service delete s3
```

A sample output is as follows:

```
Delete request for Spectrum Scale Data Access Service: 's3' is accepted
```

- REST API

```
curl -k -X DELETE -H "Authorization: Basic czMtYWRtaW46UGFzc3cwcmQ=" https://<ibm-
spectrumscale_host>/scalemgmt/v2/das/services/s3
```

A sample output is as follows:

```
{"message":"Delete request for Spectrum Scale Data Access Service: 's3' is accepted"}
```

Related concepts

"Programming reference (REST APIs)" on page 135

IBM Storage Scale Data Access Services (DAS) REST APIs are REST-style APIs that provide interoperability between a client and a server over a network. These APIs allow authenticated users to perform management tasks.

Related reference

"Command reference (mmdas command)" on page 127

The **mmdas** command manages IBM Storage Scale Data Access Services (DAS) service instances, accounts, and exports.

ETags

By default, IBM Storage Scale DAS generates the ETag of object related HTTP requests from the mtime and the inode number of the underlying file in the IBM Storage Scale file system.

Some applications require that an S3 storage returns the MD5 checksum of an S3 object in the response to a write request as value of the ETag. To support such applications, IBM Storage Scale DAS allows administrators to optionally configure the DAS S3 service enable the `EnableMD5` option for having MD5 checksum of an object in the response to respective write requests.

Managing IP address failover and failback manually

In certain scenarios, you might need to manage IP address failover and failback manually.

These scenarios include:

- Servicing Red Hat OpenShift Container Platform (OCP) nodes
- Handling nodes that have a taint of effect `NoExecute`

Complete the following steps to manually failover and failback IP addresses.

1. Disable automatic IP address failover and failback.

   ```
   mmdas service update s3 --disableAutoHA
   ```

   A sample output is as follows:

   ```
   Update request for Spectrum Scale Data Access Service: 's3' is accepted
   ```

   This command disables the monitoring of node state and thus stops the automatic triggering of IP address failover and failback.
2. Depending on your requirement, do manual IP address failover or failback as follows:
   - Complete the following steps for manual IP address failover.
     a. List all the nodes in your OCP cluster.

     ```
     oc get nodes
     ```

A sample output is as follows:

```
NAME                     STATUS      ROLES            AGE      VERSION
master0.example.com      Ready       master,worker    95d      v1.24.6+5658434
master1.example.com      Ready       master,worker    95d      v1.24.6+5658434
master2.example.com      Not Ready   master,worker    95d      v1.24.6+5658434
```

In the example output, master2 node is down.

b. List the services that are currently defined in the `openshift-storage` namespace.

```
oc get svc -o wide -n openshift-storage
```

A sample output is as follows:

```
NAME                               TYPE           CLUSTER-IP    EXTERNAL-IP    PORT(S)
AGE   SELECTOR
scale-das-ip-master0-example-com   LoadBalancer   192.0.2.137   203.0.113.40   80:32489/TCP,443:31026/TCP,8444:31326/TCP,7004:30598/TCP
45h   scale-das-node=master.example.com
scale-das-ip-master1-example-com   LoadBalancer   192.0.2.33    203.0.113.41   80:30568/TCP,443:30599/TCP,8444:32141/TCP,7004:32111/TCP
45h   scale-das-node=master.example.com
scale-das-ip-master2-example-com   LoadBalancer   192.0.2.159   203.0.113.42   80:30895/TCP,443:30526/TCP,8444:32393/TCP,7004:31767/TCP
45h   scale-das-node=master.example.com
```

c. Edit the service object associated with the master2 node to change the selector to a node that is working.

```
oc edit svc scale-das-ip-master2-example-com
```

With the edit operation, change the selector from:

```
selector:scale-das-node: master2.example.com
```

to:

```
selector:scale-das-node: master0.example.com
```

d. List the services that are currently defined in the `openshift-storage` namespace.

```
oc get svc -o wide -n openshift-storage
```

A sample output is as follows:

```
NAME                               TYPE           CLUSTER-IP    EXTERNAL-IP    PORT(S)
AGE   SELECTOR
scale-das-ip-master0-example-com   LoadBalancer   192.0.2.137   203.0.113.40   80:32489/TCP,443:31026/TCP,8444:31326/TCP,7004:30598/TCP
45h   scale-das-node=master.example.com
scale-das-ip-master1-example-com   LoadBalancer   192.0.2.33    203.0.113.41   80:30568/TCP,443:30599/TCP,8444:32141/TCP,7004:32111/TCP
45h   scale-das-node=master.example.com
scale-das-ip-master2-example-com   LoadBalancer   192.0.2.159   203.0.113.42   80:30895/TCP,443:30526/TCP,8444:32393/TCP,7004:31767/TCP
45h   scale-das-node=master.example.com
```

In the example output, the service has shifted to master0 node.

- Complete the following steps for manual IP address failback.

a. List all the nodes in your OCP cluster.

```
oc get nodes
```

A sample output is as follows:

```
NAME                     STATUS      ROLES            AGE      VERSION
master0.example.com      Ready       master,worker    95d      v1.24.6+5658434
master1.example.com      Ready       master,worker    95d      v1.24.6+5658434
master2.example.com      Ready       master,worker    95d      v1.24.6+5658434
```

b. Edit the service object that was earlier associated with the master2 node to change the selector back to the master2 node.

```
oc edit svc scale-das-ip-master2-example-com
```

With the edit operation, change the selector from:

```
selector:scale-das-node: master0.example.com
```

to:

```
selector:scale-das-node: master2.example.com
```

c. List the services that are currently defined in the `openshift-storage` namespace.

```
oc get svc -o wide -n openshift-storage
```

A sample output is as follows:

```
NAME                              TYPE           CLUSTER-IP     EXTERNAL-IP    PORT(S)
AGE   SELECTOR
scale-das-ip-master0-example-com   LoadBalancer   192.0.2.137    203.0.113.40   80:32489/TCP,443:31026/TCP,8444:31326/TCP,7004:30598/TCP
45h   scale-das-node=master0.example.com
scale-das-ip-master1-example-com   LoadBalancer   192.0.2.33     203.0.113.41   80:30568/TCP,443:30599/TCP,8444:32141/TCP,7004:32111/TCP
45h   scale-das-node=master1.example.com
scale-das-ip-master2-example-com   LoadBalancer   192.0.2.159    203.0.113.42   80:30895/TCP,443:30526/TCP,8444:32393/TCP,7004:31767/TCP
45h   scale-das-node=master2.example.com
```

In the example output, the service has shifted back to master2 node.

Managing accounts for S3 object access

Use the CLI or the API to manage your accounts for S3 object access.

Before creating an account, after the S3 service instance is created, make sure that the directory structure corresponding to the new account exists on the storage cluster with the appropriate user ID and group ID.

On the storage cluster, you must have this directory and permissions set before it is passed to the **newBucketsPath** parameter.

```
cd /<mount-point>/fs1
mkdir <create-user-dir>
chown -R uid:gid <preceding-dir-name>
```

As the IBM Storage Scale DAS administrator, you can get this directory created by the storage cluster administrator with the appropriate user ID and group ID or you can create it yourself. If you plan to use the **newBucketsPath** parameter, complete this prerequisite step before creating user accounts.

Account directory in filesets: If you plan to use an account directory that is in a fileset, the following considerations apply:

- You must change the ownership of the directory to the account user ID.
- If you have enabled SELinux on the storage cluster, you must set the SELinux context. Because the SELinux context inheritance breaks, if the account directory is in a fileset.

Use the following information to create, list, update, or delete your accounts for S3 object access.

- Create an IBM Storage Scale DAS S3 object user account as follows:
  - CLI

    ```
    mmdas account create s3user --gid 777 --uid 888 --newBucketsPath "/mnt/fs1/fset1/
    user1_buckets"
    ```

    In this command example, the following parameters are specified:

    - File system absolute path for creating new exports for the S3 user account that you want to create.

      Note: When you specify this parameter for creating an account, the specified path is not validated. If the specified path is not valid, an error occurs when you try to create an export. Administrators must specify the **newBucketsPath** to enable s3 accounts of end users to create exports using the S3 IO path. If **newBucketsPath** is not specified for an S3 account, by default, the S3 user cannot create new exports and gets the `AccessDenied` error while trying to create an export using the S3 IO path.

- User ID that is associated with the S3 user account that you want to create.
- Group ID that is associated with the S3 user account that you want to create.

A sample output is as follows:

```
Account is created successfully. The secret and access keys are as follows.

Secret Key                                      Access Key
----------                                      -----------
q2F4l5tt8/8mFXt8YOroVrUPx80TW6dlrVYm/zGO        47al0MTOuj98WkgHWmti
```

- REST API

```
curl -k -X POST -H "Content-Type: application/json" -H "Authorization: Basic
czMtYWRtaW46UGFzc3cwcmQ=" https://<ibm-spectrumscale_host>/scalemgmt/v2/das/accounts/
-d '{"name":"s3user","uid":5001,"gid":500,"newBucketsPath":"/mnt/fs1/fset1/
s3user_bucket1"}'
```

A sample output is as follows:

```
{"access_key":"UTnMjG1MUTMyXug8U6aT","secret_key":"PfaJm8ueu+4NrlgF8HI4Y8HrpZOElVJg8kVb0Fp
+"}
```

Note: The variable *<ibm-spectrumscale_host>* in the request URL must be replaced with the route host. Obtain the route host by using the following command from a node that is configured to work with the Red Hat OpenShift Container Platform (OCP) cluster:

```
oc get route ibm-spectrum-scale-gui -n <IBM Storage Scale namespace> -o json |
jq .spec.host
```

For example,

```
oc get route ibm-spectrum-scale-gui -n ibm-spectrum-scale -o json | jq .spec.host
```

A sample output is as follows:

```
"ibm-spectrum-scale-gui-ibm-spectrum-scale.example.com"
```

- List the account information for IBM Storage Scale DAS S3 object user accounts as follows:
  - CLI

```
mmdas account list
```

A sample output is as follows:

```
Name       UID    GID    New buckets path
----       ---    ---    -----------------
s3user1    888    777    /mnt/fs1/fset1/user1_buckets/s3user1_buckets

s3user2    679    629    /mnt/fs1/fset1/user1_buckets/s3user2_buckets

s3user3    478    128    /mnt/fs1/fset1/user1_buckets/s3user3_buckets

s3user4    471    127    /mnt/fs1/fset1/user1_buckets/s3user4_buckets

s3user5    431    124    /mnt/fs1/fset1/user1_buckets/s3user5_buckets
```

To list the detailed information for a specified S3 object user account in the JSON format, issue the following command:

```
mmdas account list s3user1 -o json
```

A sample output is as follows:

```
{"name":"s3user1",
"uid":888,
"gid":777,
"new_buckets_path":"/mnt/fs1/fset1/user1_buckets/s3user1_buckets",
```

```
"access_key":"47al0MTOuj98WkgHWmti",
"secret_key":"q2F4l5tt8/8mFXt8YOroVrUPx80TW6dlrVYm/zGO"}
```

Note: The access key and the secret key that are associated with an S3 object user account are only displayed in the output if you specify an account name with this command. If you specify *UserID:GroupID* with this command, they are not displayed.

- REST API

```
curl -k -X GET -H "Authorization: Basic czMtYWRtaW46UGFzc3cwcmQ=" -H "accept: application/
json" https://<ibm-spectrumscale_host>/scalemgmt/v2/das/accounts
```

A sample output is as follows:

```
[
{
    "gid": 52,
    "name": "s3user1",
    "newBucketsPath": "/mnt/fs1/fset1/s3user1_bucket1",
    "uid": 51
  },
  {
    "gid": 101,
    "name": "s3user2",
    "newBucketsPath": "/mnt/fs1/fset1/s3user2_bucket1",
    "uid": 1003
  },
  {
    "gid": 101,
    "name": "s3user3",
    "newBucketsPath": "/mnt/fs1/fset1/s3user3_bucket1",
    "uid": 1001
  },
  {
    "gid": 101,
    "name": "s3user4",
    "newBucketsPath": "/mnt/fs1/fset1/s3user4_bucket1",
    "uid": 1001
  }
]
```

- Update the IBM Storage Scale DAS S3 object user account as follows:
  - CLI

```
mmdas account update s3user2 --newBucketsPath "/mnt/fs1/fset1/sharedBuckets" --resetKeys
```

This command updates the bucket path and resets the access and secret keys. A sample output is as follows:

```
Account is successfully updated
```

  - REST API

```
curl -k -X POST -H "Content-Type: application/json" -H "Authorization: Basic
czMtYWRtaW46UGFzc3cwcmQ=" https://<ibm-spectrumscale_host>/scalemgmt/v2/das/accounts/
-d '{"name":"s3user2", "newBucketsPath":"/mnt/fs1/fset1/sharedBuckets","resetKeys":true}'
```

- Delete an IBM Storage Scale DAS S3 object user account as follows:

Note: You can delete an account only if the exports (buckets) corresponding to the account are deleted.

  - CLI

```
mmdas account delete s3user1
```

A sample output is as follows:

```
Account is successfully deleted
```

  - REST API

```
curl -k -X DELETE -H "Authorization: Basic czMtYWRtaW46UGFzc3cwcmQ=" https://<ibm-
spectrumscale_host>/scalemgmt/v2/das/accounts/s3user1
```

Related concepts

“Programming reference (REST APIs)” on page 135

IBM Storage Scale Data Access Services (DAS) REST APIs are REST-style APIs that provide interoperability between a client and a server over a network. These APIs allow authenticated users to perform management tasks.

Related reference

“Command reference (mmdas command)” on page 127

The **mmdas** command manages IBM Storage Scale Data Access Services (DAS) service instances, accounts, and exports.

Example I/O - Creating user account and uploading object to the bucket

The following example describes an end to end flow of creating a user account and uploading objects into a bucket .

1. On the IBM Storage Scale DAS cluster, create a user with required **uid**, **gid**, and the **newBucketsPath** by using the following command:

```
mmdas account create s3use8502@fvt.com --uid 8502 --gid 8888 --newBucketsPath "/mnt/remote-
sample/s3user-u8502-dir"
Account is created successfully. The secret and access keys are as follows.
 Access Key             Secret Key
 ----------             -----------
 dM5fTvmbpOsRtbRO7CY9   oo1o23wrd6HbJoo0pSM41k+jaDcaZRwS2Sh7QKnZ
```

Note: At the time of user creation, there is no check by the DAS component on the mentioned **newBucketsPath**.

2. On the storage cluster, create the respective directory with appropriate **uid** and **gid** that was created on the IBM Storage Scale DAS cluster.

   a) Create a directory in the IBM Storage Scale file system that is remotely mounted onto a containerized IBM Storage Scale DAS cluster by using the following command:

```
mkdir /mnt/fs1/s3user-u8502-dir
```

   b) If you have enabled SELinux on the storage cluster, then list the directory with the `-Z` option:

```
ls -laZd /mnt/fs1/s3user-u8502-dir
drwxr-xr-x. 2 root root unconfined_u:object_r:container_file_t:s0:c123,c456 4096 Nov 17
02:15 /mnt/fs1/s3user-u8502-dir
```

   c) If you have enabled SELinux on the storage cluster, then change the SELinux `user/role/type/level` to appropriate values as mentioned:

```
chcon system_u:object_r:container_file_t:s0:c123,c456 /mnt/fs1/s3user-u8502-dir
```

   d) Change the owner and group to the IBM Storage Scale DAS user created by using the following command:

```
chown 8502:8888 /mnt/fs1/s3user-u8502-dir/
```

   e) Change the permission to the directory by using the following command:

```
chmod 770 /mnt/fs1/s3user-u8502-dir/
```

   f) List the directory by using the following command (use the `-Z` option if SELinux was enabled on storage cluster):

```
ls -laZd /mnt/fs1/s3user-u8502-dir/
drwxrwx---. 2 8502 8888 system_u:object_r:container_file_t:s0:c123,c456 4096 Nov 17
02:15 /mnt/fs1/s3user-u8502-dir/
```

Note: In this example, MCS labels are set as c123,c456 across the Storage Cluster and Openshift-storage namespace

3. Login to the application node or infrastructure node, wherever the S3 CLI is installed and create an alias for the user.

   a) Create an alias for the user by using the following command:

```
alias s3u8502='AWS_ACCESS_KEY_ID=4cq56JcdnIIVyAY3QcIa
AWS_SECRET_ACCESS_KEY=KaSC57jyAxgDBHJ/p4i9dp/2v0/a/4FaI64Mo/63 aws --endpoint https://
10.17.54.11 --no-verify-ssl s3'
```

   Note: The IP is referred to as one of the MetalLB IP addresses that was provided at S3 Service creation time.

   b) Create a bucket by using **`s3 mb`** command:

```
s3u8502 mb s3://newbucket-u8502
urllib3/connectionpool.py:1045: InsecureRequestWarning: Unverified HTTPS request is being
made to host '10.17.54.11'. Adding certificate verification is strongly advised. See:
https://urllib3.readthedocs.io/en/1.26.x/advanced-usage.html#ssl-warnings
make_bucket: newbucket-u8502
```

   Note: Red Hat OpenShift Data Foundation (ODF) 4.13 introduced changes in bucket policies that affects buckets shared among S3 users belonging to the same group id (`gid`).

   c) List the content of the bucket by using the following command:

```
s3u8502 ls s3://newbucket-u8502
urllib3/connectionpool.py:1045: InsecureRequestWarning: Unverified HTTPS request is being
made to host '10.17.54.11'. Adding certificate verification is strongly advised. See:
https://urllib3.readthedocs.io/en/1.26.x/advanced-usage.html#ssl-warnings
```

   As no objects are uploaded, it shows empty.

   d) Upload an object to the newly created bucket:

```
echo "this is new object created" > /tmp/new-obj-for-u8502

s3u8502 cp /tmp/new-obj-for-u8502 s3://newbucket-u8502
urllib3/connectionpool.py:1045: InsecureRequestWarning: Unverified HTTPS request is being
made to host '10.17.54.11'. Adding certificate verification is strongly advised. See:
https://urllib3.readthedocs.io/en/1.26.x/advanced-usage.html#ssl-warnings
upload: ../tmp/new-obj-for-u8502 to s3://newbucket-u8502/new-obj-for-u8502
```

   e) List the content of the bucket by using the following command:

```
s3u8502 ls s3://newbucket-u8502
urllib3/connectionpool.py:1045: InsecureRequestWarning: Unverified HTTPS request is being
made to host '10.17.54.11'. Adding certificate verification is strongly advised. See:
https://urllib3.readthedocs.io/en/1.26.x/advanced-usage.html#ssl-warnings
2022-11-17 02:31:07         27 new-obj-for-u8502
```

   In this example, once the user is created, it is evident that buckets can be created and data can be uploaded.

Example I/O - Creating user account along with export(bucket creation) and uploading object to the bucket

The following example describes an end to end flow of creating a user account along with export(bucket creation) and uploading object to the bucket.

1. On the IBM Storage Scale DAS cluster, create a user with required **`uid`**, **`gid`**, and the **`newBucketsPath`** by using the following command:

```
mmdas account create s3use8503@fvt.com --uid 8503 --gid 8599 --newBucketsPath "/mnt/remote-
sample/s3user-8503-dir"
Account is created successfully. The secret and access keys are as follows.
 Access Key             Secret Key
 ----------             -----------
 8TjRTpajyftssbV0j922   v4I1GzpBRNJkNINHLraLwgQSGE6LcLOfgTphVUrI
```

Note: At the time of user creation, there is no check by the DAS component on the mentioned **newBucketsPath**.

2. Create an export directory (bucket name) with the **filesystemPath**, which does include the **newBucketsPath** in the command:

   a) Create the export by using the following command:

   ```
   mmdas export create bucket-8503 --filesystemPath "/mnt/remote-sample/s3user-8503-dir/
   newbucket-u8503-dir"
   Export is successfully created
   ```

   b) List the export by using the following command:

   ```
   mmdas export list bucket-8503

    Name           Filesystem Path
    -----          -----------------
    bucket-8503    /mnt/remote-sample/s3user-8503-dir/newbucket-u8503-dir/
   ```

   c) On the storage cluster, perform the following steps to create these directories with appropriate **uid** and **gid**:

   ```
   mkdir -p s3user-8503-dir/newbucket-u8503-dir
   ```

   d) List the directory with the `-Z` option. (Use the `-Z` option, if SELinux was enabled on a storage cluster.)

   ```
   ls -laZd s3user-8503-dir/newbucket-u8503-dir
   drwxr-xr-x. 2 root root unconfined_u:object_r:container_file_t:s0:c123,c456 4096 Nov 17
   02:52 s3user-8503-dir/newbucket-u8503-dir
   ```

   Remember: In this example, the MCS labels are c123, c456, which must be the same SCC of the openshift-storage namespace.

   e) Change the SELinux user `/role/type/level` to appropriate values as mentioned:

   ```
   chcon system_u:object_r:container_file_t:s0:c123,c456 /mnt/fs1/s3user-8503-dir/newbucket-
   u8503-dir
   ```

   f) Change the owner and group to the IBM Storage Scale DAS user created by using the following command:

   ```
   chown 8503:8599 /mnt/fs1/s3user-8503-dir /mnt/fs1/s3user-8503-dir/newbucket-u8503-dir
   ```

   g) Change the permission to the directory by using the following command:

   ```
   chmod 770 /mnt/fs1/s3user-8503-dir /mnt/fs1/s3user-8503-dir/newbucket-u8503-dir
   ```

   h) List the directory with the `-Z` option, if SELinux was enabled on a storage cluster.

   ```
   ls -laZd s3user-8503-dir/newbucket-u8503-dir
   drwxrwx---. 2 8503 8599 system_u:object_r:container_file_t:s0:c123,c456 4096 Nov 17 02:52
   s3user-8503-dir/newbucket-u8503-dir
   ```

3. Log in to the application node or infrastructure node, wherever the S3 CLI is installed and create an alias for the user.

   a) Create an alias for the user by using the following command:

   ```
   alias s3u8503='AWS_ACCESS_KEY_ID=8TjRTpajyftssbV0j922
   AWS_SECRET_ACCESS_KEY=v4I1GzpBRNJkNINHLraLwgQSGE6LcLOfgTphVUrI aws --endpoint https://
   10.17.61.211 --no-verify-ssl s3'
   ```

   Note: The IP is referred to as one of the MetalLB IP addresses that was provided at S3 Service creation time.

   b) List the content of the bucket by using the following command:

```
s3u8503 ls
urllib3/connectionpool.py:1045: InsecureRequestWarning: Unverified HTTPS request is being
made to host '10.17.61.211'. Adding certificate verification is strongly advised. See:
https://urllib3.readthedocs.io/en/1.26.x/advanced-usage.html#ssl-warnings
2022-11-17 02:59:10 bucket-8503
```

c) Upload an object to the newly created bucket:

```
echo "this is new object created that had a bucket created already" > /tmp/new-obj-for-
u8503

s3u8503 cp /tmp/new-obj-for-u8503 s3://bucket-8503
urllib3/connectionpool.py:1045: InsecureRequestWarning: Unverified HTTPS request is being
made to host '10.17.61.211'. Adding certificate verification is strongly advised. See:
https://urllib3.readthedocs.io/en/1.26.x/advanced-usage.html#ssl-warnings
upload: ../tmp/new-obj-for-u8503 to s3://bucket-8503/new-obj-for-u8503
```

d) List the content of the bucket by using the following command:

```
s3u8503 ls s3://bucket-8503
urllib3/connectionpool.py:1045: InsecureRequestWarning: Unverified HTTPS request is being
made to host '10.17.61.211'. Adding certificate verification is strongly advised. See:
https://urllib3.readthedocs.io/en/1.26.x/advanced-usage.html#ssl-warnings
2022-11-17 03:01:01         61 new-obj-for-u8503
```

In this example, after user and export creation, it is evident that the data can be uploaded directly.

Managing S3 object exports

Use the CLI or the API to manage your S3 object exports.

Use the following information to create, list, or delete your S3 object exports.

- Create an IBM Storage Scale DAS S3 object export as follows:
  - CLI

    ```
    mmdas export create bucket2 --filesystemPath /mnt/fs1/fset1/bucket1
    ```

    In this command example, the following parameter is specified:

    - Absolute path that is to be exported

      Note: Make sure that the directory structure corresponding to the new export that is specified with the `--filesystemPath` option exists on the storage cluster.

    A sample output is as follows:

    ```
    Export is successfully created
    ```

  - REST API

    ```
    curl -k -X POST -H "Content-Type: application/json" -H "Authorization: Basic
    czMtYWRtaW46UGFzc3cwcmQ=" https://<ibm-spectrumscale_host>/scalemgmt/v2/das/exports
    -d '{ "name" : "s3project", "filesystemPath": "/mnt/fs1/fset1/s3user_bucket3"}'
    ```

    Note: The variable *<ibm-spectrumscale_host>* in the request URL must be replaced with the route host. Obtain the route host by using the following command from a node that is configured to work with the Red Hat OpenShift Container Platform (OCP) cluster:

    ```
    oc get route ibm-spectrum-scale-gui -n <IBM Storage Scale namespace> -o json |
    jq .spec.host
    ```

    For example,

    ```
    oc get route ibm-spectrum-scale-gui -n ibm-spectrum-scale -o json | jq .spec.host
    ```

    A sample output is as follows:

```
"ibm-spectrum-scale-gui-ibm-spectrum-scale.example.com"
```

- List the information for IBM Storage Scale DAS S3 object exports as follows:
  - CLI

    ```
    mmdas export list
    ```

    A sample output is as follows:

    ```
    Name
    ------
    bucket2
    bucket2user1
    user1bucket1
    ```

    To list the detailed information for a specified S3 object export, issue the following command:

    ```
    mmdas export list bucket2
    ```

    A sample output is as follows:

    ```
    Name       Filesystem Path
    -----      -----------------

    bucket2   /mnt/fs1/fset1/bucket1
    ```

  - REST API

    ```
    curl -k -X GET -H "Authorization: Basic czMtYWRtaW46UGFzc3cwcmQ=" -H "accept: application/
    json" https://<ibm-spectrumscale_host>/scalemgmt/v2/das/v1/exports
    ```

    A sample output is as follows:

    ```
    [
    { "name" : "s3project"},
    { "name" : "s3project1"},
    { "name" : "s3project2"},
    { "name" : "s3project3"}
    ]
    ```

- Delete an IBM Storage Scale DAS S3 object export as follows:
  - CLI

    ```
    mmdas export delete bucket3
    ```

    A sample output is as follows:

    ```
    Export is successfully deleted
    ```

  - REST API

    ```
    curl -k -X DELETE -H "Authorization: Basic czMtYWRtaW46UGFzc3cwcmQ=" https://<ibm-
    spectrumscale_host>/scalemgmt/v2/das/exports/bucket3
    ```

Related concepts

“Programming reference (REST APIs)” on page 135
IBM Storage Scale Data Access Services (DAS) REST APIs are REST-style APIs that provide interoperability between a client and a server over a network. These APIs allow authenticated users to perform management tasks.

Related reference

“Command reference (mmdas command)” on page 127

The **mmdas** command manages IBM Storage Scale Data Access Services (DAS) service instances, accounts, and exports.

Example end to end flow of creating an export and performing I/O

The following example describes an end to end flow of creating an export and performing I/O operation.

Before you can do the following steps, IBM Storage Scale DAS and its prerequisites must be deployed, and **mmdas** CLI and AWS CLI must be configured on respective nodes.

1. On the storage cluster, create a directory under `/<mount-point>/fs1`.

```
mkdir pre-created-export-user

chown -R 8092:9002 pre-created-export-user
chcon system_u:object_r:container_file_t:s0:c111,c234 pre-created-export-user
mkdir pre-created-export-user/newbucket-for-export
ls -lZd newbucket-for-export
drwxr-x---. 3 8092 9002 system_u:object_r:container_file_t:s0 4096 Dec 16 05:14 newbucket-
for-export
```

   Important: If SELinux is enabled on an IBM Storage Scale cluster, set MCS labels for the **chcon** command, and use the `-Z` option when listing by using the **ls** command.

2. On the Red Hat OpenShift cluster, create s3user with the user ID, group ID, and **newBucketsPath** set to these values for the created directory .

```
mmdas account create s3user8092@example.com --gid 9002 --uid 8092 --newBucketsPath /mnt/
remote-sample/pre-created-export-user
Account created successfully, below are the secret and access keys
Secret Key                                       Access Key
 ----------                                       -----------
 NhDgFUW/05FkvIBmx/Bm/v6Wi1s7tqccFOZR3k+S       j3QvSz4IwSNAqVlCPn5l
```

3. On the Red Hat OpenShift cluster, create the export.

```
mmdas export create bucket-for-export --filesystemPath /mnt/remote-sample/pre-created-export-
user/newbucket-for-export

Export is successfully created
```

4. On the node where the AWS CLI is installed, check the s3user listing with the user credentials to show that the export that is created on the Red Hat OpenShift cluster.

```
s3u8092 ls
urllib3/connectionpool.py:1013: InsecureRequestWarning: Unverified HTTPS request is being
made to host 's3-endpoints.example.com'.
Adding certificate verification is strongly advised. See: https://urllib3.readthedocs.io/en/
latest/advanced-usage.html#ssl-warnings

2021-12-16 07:54:58 bucket-for-export


ls -lh /root/file_20G

-rw-r--r-- 1 root root 20G Dec 16 08:01 /root/file_20G
```

5. Upload a file from the node to the exported directory (bucket).

```
s3u8092 cp /root/file_20G s3://bucket-for-export

upload: ./file_20G to s3://bucket-for-export/file_20G
```

6. List the contents of the export.

```
s3u8092 ls s3://bucket-for-export

2021-12-16 08:05:31 21474836480 file_20G
```

Backing up and restoring IBM Storage Scale DAS configuration

IBM Storage Scale DAS provides scripts to back up and restore your S3 configuration files and NooBaa PostgreSQL database.

- Before you use the backup script, make sure that IBM Storage Scale container native, IBM Storage Scale CSI, and IBM Storage Scale DAS (including the S3 service) are configured and running successfully on the OCP cluster.
- Before you use the restore script, make sure that the OCP cluster is set up with IBM Storage Scale container native, IBM Storage Scale CSI, and IBM Storage Scale DAS, except for configuring the S3 service. The restore script restores the S3 service configuration and HA configuration. HA configuration involves MetalLB and related configuration.
- Before you use the backup and restore scripts, enable network policy in the environment. For more information, see Network policy considerations. Otherwise, when you are restoring S3 service the noobaa-endpoint might not run on DAN nodes.

1. Copy the `dasS3Backup.sh` and `dasS3Restore.sh` scripts from one of the IBM Storage Scale DAS endpoints pods.

   ```
   oc cp ibm-spectrum-scale-das-endpoint-b57955bb6-4vv96:scripts/ /tmp/scripts/
   ls -ltr /tmp/scripts

   total 12
   -rw-r--r-- 1 root root 3910 Feb  17 15:33 dasS3Backup.sh
   -rw-r--r-- 1 root root 2694 Feb  17 15:33 dasS3Restore.sh
   ```

2. Make the scripts executable.

   ```
   chmod +x /tmp/scripts
   ```

- Use the `dasS3Backup.sh` script to back up the IBM Storage Scale DAS service configuration and NooBaa secret keys.

  ```
  ./dasS3Backup.sh <backup_directory>
  ```

  Where *<backup_directory>* is the directory where the backup TAR file is created.

  Note: Make sure that the backup directory exists before using the backup script.

  For example,

  ```
  ./dasS3Backup.sh /tmp/dasbackup
  ```

  The script creates a tape archive (TAR) file and an MD5 checksum file of the TAR file. A sample output is as follows:

  ```
  2022-03-21T09:26:29 INFO: Backup process is completed
  2022-03-21T09:26:29 INFO: backup.20220321-092627.tar and backup.20220321-092627.tar.md5sum
  are stored in the /tmp/dasbackup
  ```

- Use the `dasS3Restore.sh` script to restore the IBM Storage Scale DAS configuration files as follows:

  ```
  ./dasS3Restore.sh config <backup_tar_file> [<backup_tar_checksum_file>]
  ```

  Where,

  – *<backup_tar_file>* is the TAR file that is created when you run the **`./dasS3Backup.sh`** script.

  – [Optional] *<backup_tar_checksum_file>* is the file that contains the MD5 checksum of the backup TAR file.

  For example,

  ```
  ./dasS3Restore.sh config /tmp/dasbackup/backup.20220321-074500.tar
  ```

  A sample output is as follows:

```
secret/das-gui-user configured
s3service.das.scale.ibm.com/s3 created
haservice.das.scale.ibm.com/s3 created
Restore DAS config file is completed
```

- Restore the NooBaa PostgreSQL database as follows:
  a) Verify that the S3 service is in the `ready` state.

  ```
  mmdas service list s3
  ```

  b) Verify that all the pods in the `openshift-storage` namespace are in the `running` state.

  ```
  oc get pods -n openshift-storage
  ```

  c) Use the `dasS3Restore.sh` script to restore the NooBaa PostgreSQL database.

  ```
  ./dasS3Restore.sh db <backup_tar_file> [<backup_tar_checksum_file>]
  ```

  Where,

  – *<backup_tar_file>* is the TAR file that is created when you run the **`./dasS3Backup.sh`** script.

  – [Optional] *<backup_tar_checksum_file>* is the file that contains the MD5 checksum of the backup TAR file.

  For example, restore the NooBaa PostgreSQL database as follows:

  ```
  ./dasS3Restore.sh db /tmp/dasbackup/backup.20220321-074500.tar
  ```

  A sample output is as follows:

  ```
  2022-04-05T18:21:31 INFO: Restore process is completed
  ```

Shutting down and starting up an IBM Storage Scale DAS cluster

Shut down and start up your IBM Storage Scale DAS cluster as follows:

1. Verify that the S3 commands are working.
2. Stop all workloads that you are running on the IBM Storage Scale DAS cluster.
3. Back up the S3 configuration files and NooBaa PostgreSQL database. For more information, see "Backing up and restoring IBM Storage Scale DAS configuration" on page 83.
4. Unmount and shut down the file system on all core pods.

   ```
   oc -n ibm-spectrum-scale exec master0 -- mmunmount all
   oc -n ibm-spectrum-scale exec master0 -- mmshutdown
   oc -n ibm-spectrum-scale exec master1 -- mmunmount all
   oc -n ibm-spectrum-scale exec master1 -- mmshutdown
   oc -n ibm-spectrum-scale exec master2 -- mmunmount all
   oc -n ibm-spectrum-scale exec master2 -- mmshutdown
   ```

   A sample output is as follows:

   ```
   Defaulted container "gpfs" out of: gpfs, logs, mmbuildgpl (init), config (init)
   Sun Mar  6 17:34:14 UTC 2022: mmshutdown: Starting force unmount of GPFS file systems
   Sun Mar  6 17:34:19 UTC 2022: mmshutdown: Shutting down GPFS daemons
   Shutting down!
   ```

   Note: The noobaa-db pods go in the `CrashLoopBackOff` state. All the pods that are up and running includes the IBM Storage Scale container native and CSI pods.

5. Shut down the IBM Storage Scale container native cluster by setting `replicas` to 0.

   ```
   oc edit deploy -n ibm-spectrum-scale-operator
   ...
   spec:
     progressDeadlineSeconds: 600
     replicas: 0
   ```

```
...
oc label node --all scale-
oc delete pods -lapp.kubernetes.io/name=core -n ibm-spectrum-scale
```

6. Power off the OCP nodes by using the IPMI tool.

```
ipmitool -I lanplus -H 192.0.2.44 -U USERID -P Hp0cpcluster power off
Chassis Power Control: Down/Off

ipmitool -I lanplus -H 192.0.2.43 -U USERID -P Hp0cpcluster power off
Chassis Power Control: Down/Off

ipmitool -I lanplus -H 192.0.2.42 -U USERID -P Hp0cpcluster power off
Chassis Power Control: Down/Off

oc get nodes
Unable to connect to the server: EOF
```

7. Power on the OCP nodes by using the IPMI tool.

```
ipmitool -I lanplus -H 192.0.2.42 -U USERID -P Hp0cpcluster power on
    Chassis Power Control: Up/On

ipmitool -I lanplus -H 192.0.2.43 -U USERID -P Hp0cpcluster power on
   Chassis Power Control: Up/On

ipmitool -I lanplus -H 192.0.2.44 -U USERID -P Hp0cpcluster power on
   Chassis Power Control: Up/On
```

8. Start the IBM Storage Scale container native cluster by setting `replicas` to 1.

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

...
spec:
  progressDeadlineSeconds: 600
  replicas: 1
...
```

Note: Ensure that the nodes are in Ready state by using the **`oc get nodes`** command before restarting the IBM Storage Scale cluster. If any of the nodes are in a state other than Ready, the IBM Storage Scale cluster fails to start.

After the operator pod comes into the `Running` state, the core pods are rescheduled and the default CSI label is re-applied.

9. Check the `openshift-storage` namespace and make sure all the pods are up and running.
10. Verify that the S3 commands are working.
11. Restore the S3 configuration files and NooBaa PostgreSQL database. For more information, see "Backing up and restoring IBM Storage Scale DAS configuration" on page 83.

Accessing IBM Storage Scale DAS Service GUI

The topic describes steps to access IBM Storage Scale DAS S3 GUI.

Users created on the Red Hat OpenShift Container Platform (OCP) can log in to the IBM Storage Scale GUI through single sign-on (SSO) by using the OAuth implementation.

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

1. In a browser, open `https://ibm-spectrum-scale-gui-ibm-spectrum-scale.apps.<domain>`. You can see the **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 Red Hat OpenShift Container Platform user credentials.

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

Data access service

You can configure, edit and delete IBM Storage Scale DAS service, accounts, and exports.

You must complete the following prerequisites before you start configuring the IBM Storage Scale DAS S3 service.

- Install the IBM Storage Scale container native Storage Access (CNSA) and Container Storage Interface (CSI) driver. For more information, see the *Installing the IBM Storage Scale container native operator and cluster* in the IBM Storage Scale container native documentation.
- Configure and verify the remote storage cluster path. For more information, see the topic *Verifying an IBM Storage Scale container native cluster* in the *IBM Storage Scale container native* documentation.
- Install IBM Storage Scale DAS. For more information, see "Installing IBM Storage Scale DAS" on page 34.

Configuring DAS Service

1. To access **DAS Service GUI**, select from the main menu, **Services**. Then select **DAS S3**.
2. In the **Configure Service** window, click either the **Basic Configuration** or the **Advanced Configuration** tab.
3. In the **Service name** field, type the name of the service instance. For example, S3.

   Note: You can configure only one DAS S3 service instance and it must be in a ready state before configuring the Account and Export.
4. In the **Accept license** field, select `True` to enable the IBM Storage Scale license and allow the configuration of S3 service instance.
5. In the **IP Range** field, type the range of IP addresses that is to be used for the MetalLB configuration. The IP addresses must meet the following criteria.
   - IP Addresses must be in the format: `x.x.x.x-x.x.x.x`
   - IP Addresses must be in a sequence. For example, `192.0.2.11-192.0.2.13`
   - IP Addresses must match the number of OCP nodes which are labeled for IBM Storage Scale usage and display the "`scale=true`" label.
6. In the **Path** field, type the IBM Storage Scale filesystem mount point that will be enabled for S3 access.
7. In the **Storage class** field under the **Advanced Configuration** tab, type the name of the storage class that is used to configure a database for the S3 service.
8. Click **OK**.

Configuring DAS Accounts

Before configuring DAS Accounts, you must ensure that the DAS S3 service instance is configured and is in a ready state.

1. To access the IBM Storage Scale DAS accounts, select **Protocols** from the main menu, then select **DAS S3 Accounts**.
2. On the **DAS** page under **Accounts**, click **Configure**.
3. In the **Configure Account** window, click either the **Basic Configuration** or the **Advanced Configuration** tab.
4. In the **Account name** field, type the S3 user account name.
5. In the **UID** field, type the user ID that is associated with the S3 user account.
6. In the **GID** field, type the group ID that is associated with the S3 user account.
7. In the **Path** field, type the file system absolute path, which acts as a base path for S3 buckets
8. Click **OK**.

Configuring DAS Exports

Before configuring DAS Exports, you must ensure that the DAS S3 service instance is configured and is in a ready state.

1. To access the IBM Storage Scale DAS exports, select **Protocols** from the main menu, then select **DAS S3 Exports**.
2. On the **DAS** page under **Exports**, click **Configure**.
3. In the **Configure Export** window, click either the **Basic Configuration** or the **Advanced Configuration** tab.
4. In the **Export name** field, type the name of the S3 export that uses the path defined in the **File system path** field. The name must meet the following requirements.
   - The name must consist of lower case alphanumeric characters, **-** (dash), or **.** (period)
   - The name must begin and end with an alphanumeric character
   - The name must have a length greater than or equal to 3 characters and less than or equal to 63 characters.
5. In the **File system path** field, type the absolute path that is to be exported.
6. Click **OK**.

You can select a row and click **Actions** > **Edit** to edit the configurations for **Account**, **Services** or **Exports**.

You can also select a row and click **Actions** > **Delete** to delete the configured **Account**, **Services** or **Exports**.

Changing GUI user passwords

The namespaces of IBM Storage Scale container native and IBM Storage Scale CSI components contain secrets. These secrets contain passwords for container native and CSI GUI users on the storage cluster. The passwords for GUI users `cnsa_storage_gui_user` and `csi_storage_gui_user` expire after 90 days by default. Changing these passwords requires you to schedule a short maintenance window for IBM Storage Scale DAS.

To change these passwords, issue the following commands on the GUI node of the storage cluster.

```
cd /usr/lpp/mmfs/gui/cli
./chuser csi-storage-gui-user -p <new password>
./chuser cnsa_storage_gui_user -p <new_password>
```

For more information on creating or updating a secret, see Changing the configuration after deployment and Creating secrets for the storage cluster GUI.

Chapter 7. Monitoring

Use the following information to monitor the health of your IBM Storage Scale DAS components.

Monitoring health of S3 data interface

You can use the IBM Storage Scale **mmhealth** command to monitor the health of the S3 data interface (NooBaa).

1. Change the context to the ibm-spectrum-scale namespace.

```
oc project ibm-spectrum-scale
```

2. List the IBM Storage Scale container native pods.

```
oc get pods -o wide
```

A sample output is as follows:

```
NAME                                                  READY STATUS    RESTARTS AGE
IP               NODE                   NOMINATED NODE   READINESS GATES
ibm-spectrum-scale-gui-0                              4/4   Running   0        16d
192.0.2.122      worker2.example.com    <none>            <none>
ibm-spectrum-scale-gui-1                              4/4   Running   0        16d
192.51.100.111  worker0.example.com    <none>            <none>
ibm-spectrum-scale-noobaamonitoring-7c777c46b5-ljhkv  1/1   Running   0        14d
192.0.2.208      worker2.example.com    <none>            <none>
ibm-spectrum-scale-pmcollector-0                      2/2   Running   0        37d
192.0.2.15       worker1.example.com    <none>            <none>
ibm-spectrum-scale-pmcollector-1                      2/2   Running   0        37d
192.51.100.30    worker0.example.com    <none>            <none>
worker0                                               2/2   Running   0        37d
203.0.113.67     worker0.example.com    <none>            <none>
worker1                                               2/2   Running   0        27d
203.0.113.166    worker1.example.com    <none>            <none>
worker2                                               2/2   Running   0        37d
203.0.113.176    worker2.example.com    <none>            <none>
```

Note: The noobaamonitoring pod gets created when you create the S3 service instance. In this example output, the worker2 node is interacting with the noobaamonitoring pod.

3. Log in using **rsh** to the worker node core pod that is running the noobaamonitoring pod.

```
oc rsh worker2
```

4. On the worker2 core pod running node, view the health information of all components running on the node.

```
mmhealth node show
```

A sample output is as follows:

```
Node name:      worker2
Node status:    TIPS
Status Change:  2 days ago

Component      Status        Status Change     Reasons
--------------------------------------------------------------------------------
CALLHOME       HEALTHY       2 days ago        -
GPFS           TIPS          2 days ago        gpfs_maxstatcache_low
NETWORK        HEALTHY       2 days ago        -
FILESYSTEM     HEALTHY       2 days ago        -
GUI            HEALTHY       2 days ago        -
NOOBAA         HEALTHY       1 day ago         -
PERFMON        HEALTHY       2 days ago        -
THRESHOLD      HEALTHY       2 days ago        -
PERFMON        HEALTHY       Now               -
THRESHOLD      HEALTHY       5 days ago        -
```

5. On the worker2 core pod running node, view the detailed health information for the Red Hat NooBaa component running on the node.

```
mmhealth node show noobaa -v
```

A sample output is as follows:

```
Node name:      worker2.example.com

Component                Status        Status Change            Reasons & Notices
---------------------------------------------------------------------------------
NOOBAA                   HEALTHY       2021-12-10 05:56:04      -
  newbucket-s3user8005   HEALTHY       2021-12-10 07:08:08      -
  newbucket-s3user8006   HEALTHY       2021-12-10 07:16:23      -
  newbucket-user87       HEALTHY       2021-12-13 04:00:00      -


Event                   Parameter                Severity    Active Since             Event
Message
---------------------------------------------------------------------------------------------
---------------------------------------------------------------------------------------------
---
service_pod_data        NOOBAA                   INFO        2021-12-10 05:55:49      The
request to ibm-spectrum-scale-noobaamonitoring-7c777c46b5-ljhkv did return health data as
expected.
noobaa_api_active       NOOBAA                   INFO        2021-12-10 05:48:34      Noobaa
Data was retrieved successfully
ns_rsc_data_present     NOOBAA                   INFO        2021-12-10 05:56:04      Data
for Noobaa Namespace Resources was retrieved successfully
service_pod_data        NOOBAA                   INFO        2021-12-10 05:55:49      The
request to ibm-spectrum-scale-noobaamonitoring-7c777c46b5-ljhkv did return health data as
expected.
active_ns_rsc           NOOBAA                   INFO        2021-12-10 05:56:04
Namespace Resource noobaa-s3res-4080029599 is active in Noobaa
active_ns_bucket        newbucket-s3user8005     INFO        2021-12-10 07:08:08      Bucket
newbucket-s3user8005 is Healthy and Active
active_ns_bucket        newbucket-s3user8006     INFO        2021-12-10 07:16:23      Bucket
newbucket-s3user8006 is Healthy and Active
active_ns_bucket        newbucket-user87         INFO        2021-12-13 04:00:00      Bucket
newbucket-user87 is Healthy and Active
```

Note: You can also monitor the health of the S3 exports (buckets) as seen in the preceding output.

For viewing specific information or for restarting the system health monitor, use the following commands:

View the health information for NooBaa buckets:

```
mmhealth node show noobaa
```

A sample output is as follows:

```
Node name:      worker2

Component                Status        Status Change     Reasons & Notices
--------------------------------------------------------------------------
NOOBAA                   HEALTHY       6 days ago        -
  newbucket-s3user8005   HEALTHY       6 days ago        -
  newbucket-s3user8006   HEALTHY       6 days ago        -
  newbucket-user87       HEALTHY       3 days ago        -

There are no active error events for the component NOOBAA on this node (worker2).
```

View unhealthy events in the NooBaa component:

```
mmhealth node show noobaa --unhealthy
```

A sample output is as follows:

```
Node name:      master0

Component     Status        Status Change     Reasons
--------------------------------------------------------------------------
NOOBAA        DEGRADED      2 days ago        inactive_ns_rsc
```

```
Event               Parameter     Severity    Active Since      Event Message
------------------------------------------------------------------------------------------
inactive_ns_rsc     NOOBAA        WARNING     2 days ago        Namespace Resource is
not created in Noobaa
```

Monitoring NooBaa with call home

You can use the IBM Storage Scale **mmcallhome** command to monitor NooBaa by collecting details of its system health events.

1. Change the context to the `ibm-spectrum-scale` namespace.

```
oc project ibm-spectrum-scale
```

2. List the IBM Storage Scale container native pods.

```
oc get pods -o wide
```

A sample output is as follows:

```
NAME                                                   READY STATUS    RESTARTS AGE
IP              NODE                    NOMINATED NODE   READINESS GATES
ibm-spectrum-scale-gui-0                               4/4   Running   0        16d
192.0.2.122     worker2.example.com     <none>            <none>
ibm-spectrum-scale-gui-1                               4/4   Running   0        16d
192.51.100.111  worker0.example.com     <none>            <none>
ibm-spectrum-scale-noobaamonitoring-7c777c46b5-ljhkv  1/1   Running   0        14d
192.0.2.208     worker2.example.com     <none>            <none>
ibm-spectrum-scale-pmcollector-0                       2/2   Running   0        37d
192.0.2.15      worker1.example.com     <none>            <none>
ibm-spectrum-scale-pmcollector-1                       2/2   Running   0        37d
192.51.100.30   worker0.example.com     <none>            <none>
worker0                                                2/2   Running   0        37d
203.0.113.67    worker0.example.com     <none>            <none>
worker1                                                2/2   Running   0        27d
203.0.113.166   worker1.example.com     <none>            <none>
worker2                                                2/2   Running   0        37d
203.0.113.176   worker2.example.com     <none>            <none>
```

Note: The `noobaamonitoring` pod gets created when you create the S3 service instance. In this example output, the worker2 node is interacting with the `noobaamonitoring` pod.

3. Log in by using **rsh** to the worker core pod that is interacting with the `noobaamonitoring` pod.

```
oc rsh worker2
```

4. Configure the customer information for call home.

```
mmcallhome info change --customer-name CustomerName --customer-id CustomerID --country-code
CountryCode --email Email
```

A sample output is shown as follows:

```
Call home country-code has been set to **
Call home customer-id has been set to ******
Call home customer-name has been set to ******
Call home email has been set to ******
```

5. Enable the call home capability.

```
mmcallhome capability enable accept
```

A sample output is shown as follows:

```
Call home enabled has been set to true
Additional messages:
License acceptance specified on command line. Call home enabled.
```

6. Distribute all compatible cluster nodes into call home groups automatically.

```
mmcallhome group auto
```

A sample output is as follows:

```
[I] Analyzing the cluster...
No ungrouped potential call home server nodes found.
```

```
mmcallhome group list
```

A sample output is as follows:

```
callHomeGroup callHomeNode callHomeChildNodes
--------------- ------------- --------------------
autoGroup_1 worker2 worker0,worker1,worker2
```

7. Set up the call home gather-send task to collect and upload data daily.

```
mmcallhome run GatherSend --task daily
```

A sample output is as follows:

```
One time run completed with success
```

8. View the status of the currently running and the already completed call home tasks.

```
mmcallhome status list --numbers 1 --task daily --verbose
```

A sample output is shown as follows:

```
=== Executed call home tasks ===
Group Task Start Time Updated Time Status RC or Step Package File Name Original Filename
---------------------------------------------------------------------------------------------
---------------------------------------------------------------------------------------------
-------------
autoGroup_1 daily 20220721105037.458 20220721105104 success RC=0 /tmp/mmfs/callhome/
rsENUploaded/
83325621788401.5_1_4_0.123456.IN.Sanvidhan.autoGroup_1.gat_daily.g_daily.cnsa.202207211050374
58.cl0.DC
```

Collecting data for support

Use the following information to collect support data for NooBaa and IBM Storage Scale DAS components including the IBM Storage Scale DAS operator.

Changing log level for IBM Storage Scale DAS components

Change the log level collection for IBM Storage Scale DAS components.

You can change log levels for the following IBM Storage Scale DAS components:

- NooBaa component logs
- IBM Storage Scale DAS operator and IBM Storage Scale DAS endpoint logs

Change the verbosity of openshift-storage namespace for noobaa component logs

For the `openshift-storage` namespace, the following log levels are available.

- `default_level`
- `all`

Note: The `default_level` is `0` and `all` is `5`.

By default, the log level for the `openshift-storage` namespace is set to `default_level`. To increase the verbosity of logs for the NooBaa component, change the log level to a higher value (0 is set as default):

Note: The **noobaa** command is available from the RedHat ODF package as a separate rpm.

For example:

```
noobaa system set-debug-level 3

INFO[0000] □ Exists: NooBaa "noobaa"

INFO[0000] □ Exists: Service "noobaa-mgmt"

INFO[0000] □ Exists: Secret "noobaa-operator"

INFO[0000] □ Exists: Secret "noobaa-admin"

INFO[0000] ✈ RPC: redirector.publish_to_cluster() Request: {Target: MethodAPI:debug_api
MethodName:set_debug_level RequestParams:{Module:core Level:3}}

WARN[0000] RPC: GetConnection creating connection to wss://localhost:43503/rpc/ 0xc000522d20

INFO[0000] RPC: Connecting websocket (0xc000522d20) &{RPC:0xc00009d4a0 Address:wss://
localhost:43503/rpc/ State:init WS:<nil> PendingRequests:map[] NextRequestID:0 Lock:
{state:1 sema:0} ReconnectDelay:0s cancelPings:<nil>}

INFO[0000] RPC: Connected websocket (0xc000522d20) &{RPC:0xc00009d4a0 Address:wss://
localhost:43503/rpc/ State:init WS:<nil> PendingRequests:map[] NextRequestID:0 Lock:
{state:1 sema:0} ReconnectDelay:0s cancelPings:<nil>}

INFO[0000] □ RPC: redirector.publish_to_cluster() Response OK: took 6.0ms

Debug level was set to 3 successfully

Debug level is not persistent and is only effective for the currently running core and
endpoints pods
```

Change the log level for das-operator and das-endpoint pods

For the `das-operator` and `das-endpoint` pods, the following log levels are available.

- `INFO`
- `DEBUG`
- `ERROR`
- `WARN`

As an IBM Storage Scale DAS administrator, change the log level for `das-operator` and `das-endpoint` pods as follows:

1. Change the context to the `ibm-spectrum-scale-das` namespace.

   ```
   oc project ibm-spectrum-scale-das
   ```

2. Change the `LOG_LEVEL` environment variable under the `spec` section for `das-operator-controller-manager` spec.

   ```
   oc edit deployment ibm-spectrum-scale-das-controller-manager
   ```

   For example,

   ```
   spec:
   ...
   ...
      spec:
         containers:
         ...
         ...
            env:
            - name: LOG_LEVEL
              value: DEBUG
   ```

Once the log level is changed for the IBM Storage Scale DAS operator and IBM Storage Scale DAS endpoint, it automatically gets applied to the `das-endpoint` pods, when the `das-operator` reconciles the `das-endpoint` pods with the changed log level.

Collecting support information for NooBaa

Use the IBM Storage Scale **gpfs.snap** command to gather support information for NooBaa such as the pod's deployment, services, and statefulset. Use **oc adm must-gather** to gather NooBaa pod logs and detailed information.

1. Use **gpfs.snap** to gather NooBaa information as follows:

   a) Change the context to the `ibm-spectrum-scale` namespace.

   ```
   oc project ibm-spectrum-scale
   ```

   b) List the IBM Storage Scale container native pods by issuing the following command:

   ```
   oc get pods -o wide
   ```

   A sample output is as follows:

   ```
   NAME                                                  READY STATUS    RESTARTS AGE
   IP               NODE                  NOMINATED NODE   READINESS GATES
   ibm-spectrum-scale-gui-0                              4/4   Running   0        16d
   192.0.2.122      worker2.example.com   <none>            <none>
   ibm-spectrum-scale-gui-1                              4/4   Running   0        16d
   192.51.100.111   worker0.example.com   <none>            <none>
   ibm-spectrum-scale-noobaamonitoring-7c777c46b5-ljhkv  1/1   Running   0        14d
   192.0.2.208      worker2.example.com   <none>            <none>
   ibm-spectrum-scale-pmcollector-0                      2/2   Running   0        37d
   192.0.2.15       worker1.example.com   <none>            <none>
   ibm-spectrum-scale-pmcollector-1                      2/2   Running   0        37d
   192.51.100.30    worker0.example.com   <none>            <none>
   worker0                                               2/2   Running   0        37d
   203.0.113.67     worker0.example.com   <none>            <none>
   worker1                                               2/2   Running   0        27d
   203.0.113.166    worker1.example.com   <none>            <none>
   worker2                                               2/2   Running   0        37d
   203.0.113.176    worker2.example.com   <none>            <none>
   ```

   Note: The `noobaamonitoring` pod gets created when you create the S3 service instance. In this example output, the worker2 node is interacting with the `noobaamonitoring` pod.

   c) Log in by using **rsh** to the worker pod node that is interacting with the `noobaamonitoring` pod.

   ```
   oc rsh worker2
   ```

   d) On the worker2 pod, gather the IBM Storage Scale data by issuing the following command:

   ```
   gpfs.snap
   ```

   A truncated version of the sample output is as follows:

   ```
   gpfs.snap: started at Wed Jun 30 09:40:13 UTC 2021.
   Gathering common data...
   Gathering Linux specific data...
   Gathering extended network data...
   Gathering local noobaa data...
   Gathering local callhome data...
   .
   .
   .
   gpfs.snap:  Spawning remote gpfs.snap calls. Master is worker0.example.com.
   This may take a while.

   .
   .
   .
   Writing * to file /tmp/gpfs.snapOut/2468992/collect/
   gpfs.snap.worker0_master_20210630094013.2468992.out.tar.gz
   Packaging all data.
   Writing . to file /tmp/gpfs.snapOut/2468992/all.20210630094013.2468992.tar
   ```

```
gpfs.snap completed at Wed Jun 30 09:41:38 UTC 2021
###############################################################################
Send file /tmp/gpfs.snapOut/2468992/all.20210630094013.2468992.tar to IBM Service
Examine previous messages to determine additional required data.
###############################################################################
```

This command creates a compressed file of the gathered data.

e) Use the **oc cp** command to transfer the compressed file to one of the nodes that is configured to work with the OCP cluster.

f) Extract the contents of the compressed file by issuing the following command:

```
tar xvf  /tmp/gpfs.snapOut/2468992/all.20210630094013.2468992.tar
```

A sample output is as follows:

```
./gui.snap.cluster.worker0.example.com.20210630_094050.tar.gz
./sysmon.snap.cluster.worker0.example.com.20210630_094103.tar.gz
./cnss.snap.cluster.worker0.example.com.20210630_094103.tar.gz
./callhome.snap.cluster.worker0.example.com.20210630_094103.tar.gz
./perfmon.snap.cluster.worker0.example.com.20210630_094104.tar.gz
./gpfs.snap.worker2_20210630094106.100729.out.tar.gz
./gpfs.snap.worker1_20210630094106.1181837.out.tar.gz
./remote.gpfs.snap.output_20210630094013.2468992
./gpfs.snap.worker0_master_20210630094013.2468992.out.tar.gz
```

The `noobaamonitoring` pod is running on the worker2 node. You can confirm this by using the **oc get pods -o wide** command.

g) Extract the contents of the compressed file for worker2 and search for noobaa in the extracted contents by issuing the following command:

```
tar zxvf ./gpfs.snap.worker2_master_20210630094013.2468992.out.tar.gz | grep noobaa
```

A sample output is as follows:

```
noobaa.snap.worker2.example.com.20210630_094039/
noobaa.snap.worker2.example.com.20210630_094039/SIDECAR/
noobaa.snap.worker2.example.com.20210630_094039/SIDECAR/noobaa/
noobaa.snap.worker2.example.com.20210630_094039/SIDECAR/noobaa/CommandOutput/
noobaa.snap.worker2.example.com.20210630_094039/SIDECAR/noobaa/CommandOutput/
mmsysmon_noobaa_api.py_noobaa_ftdc
noobaa.snap.worker2.example.com.20210630_094039/SIDECAR/noobaa/CommandOutput/
mmsysmon_noobaa_openshift.py
```

The NooBaa related information is located in the following files:

```
noobaa.snap.worker0.example.com.20210630_094039/SIDECAR/noobaa/CommandOutput/
mmsysmon_noobaa_api.py_noobaa_ftdc
noobaa.snap.worker0.example.com.20210630_094039/SIDECAR/noobaa/CommandOutput/
mmsysmon_noobaa_openshift.py
```

h) Remove the **gpfs.snap** from `/tmp` in the pod.

2. Use **oc adm must-gather** to gather NooBaa pod logs as follows:

a) Change the context to the `openshift-storage` namespace by issuing the following command:

```
oc project openshift-storage
```

b) Gather NooBaa pods-related information by issuing the following command:

```
oc adm must-gather --image=registry.redhat.io/odf4/odf-must-gather-rhel9:v4.14 --dest-
dir=<directory-name>
```

3. Use **oc adm must-gather** to gather support information for all nodes in the OCP cluster.

a) Change the context to the `openshift-storage` namespace by issuing the following command:

```
oc project openshift-storage
```

b) Gather information about all nodes in the OCP cluster by issuing the following command:

```
oc adm must-gather
```

Collecting support information for IBM Storage Scale DAS

Use the **oc adm must-gather** command to gather support information required for debugging any IBM Storage Scale DAS operator related issues.

1. Change the context to the `ibm-spectrum-scale-das` namespace.

   ```
   oc project ibm-spectrum-scale-das
   ```

2. Set up OCS must-gather and collect support information by referring to the IBM Storage Scale container native documentation. For more information, see Gathering data about your cluster under IBM Storage Scale container native documentation.

After completing the preceding step, support information including logs files related to IBM Storage Scale DAS get populated in the following sub directories under the `ibm-spectrum-scale-das` directory:

```
ibm-spectrum-scale-das-controller-manager-79bf49b859-d9425
ibm-spectrum-scale-das-endpoint-7b657c859c-6lsx9
ibm-spectrum-scale-das-endpoint-7b657c859c-qf679
```

Monitoring S3 (NooBaa) performance with the mmperfmon query command

Using the **mmperfmon** query command user can monitor and collect the S3 performance statistics data, which is being processed by `dasproxy` and `pmcollector`.

There are different types of operations are executed on S3 buckets and its objects under NooBaa. The operations are categorized the following three types:

- s3io

  Indicates the I/O operations includes `read` and `write` operation statistics.
- s3ops

  Contains different types of S3 operations, can be performed on buckets and its objects.
- s3fsops

  It is a file system operations, which is performed by internal processes, based on the S3 operations.

I/O operations

- `read` operation

  Metrics can be fetch with the following statistics for the `read` operation:

  - `s3op_count`
  - `s3op_size`
- `write` operation

  Metrics can be fetch with the following statistics for the `write` operation:

  - `s3op_count`
  - `s3op_size`

The **mmperfmon** command allows the system administrator to collect performance statistics from the `scale-core` pod.

Example to login into `scale-core` pod to use the `mmperfmon` command:

```
# oc -n ibm-spectrum-scale get pods -o wide
```

Query output:

```
NAME                                                   READY   STATUS    RESTARTS   AGE
IP             NODE                                     NOMINATED NODE   READINESS GATES
ibm-spectrum-scale-gui-0                               4/4     Running   0          4d5h
10.128.5.239   master2.jfmocp4.pokprv.stglabs.ibm.com   <none>           <none>
ibm-spectrum-scale-gui-1                               4/4     Running   0          4d5h
10.128.1.105   master0.jfmocp4.pokprv.stglabs.ibm.com   <none>           <none>
ibm-spectrum-scale-noobaamonitoring-69b8b94db8-b4mj4   2/2     Running   0          20h
10.128.4.91    master2.jfmocp4.pokprv.stglabs.ibm.com   <none>           <none>
ibm-spectrum-scale-pmcollector-0                       2/2     Running   0          4d5h
10.128.5.242   master2.jfmocp4.pokprv.stglabs.ibm.com   <none>           <none>
ibm-spectrum-scale-pmcollector-1                       2/2     Running   0          4d5h
10.128.1.107   master0.jfmocp4.pokprv.stglabs.ibm.com   <none>           <none>
master0                                                2/2     Running   0          4d5h
10.28.20.32    master0.jfmocp4.pokprv.stglabs.ibm.com   <none>           <none>
master1                                                2/2     Running   0          4d5h
10.28.20.33    master1.jfmocp4.pokprv.stglabs.ibm.com   <none>           <none>
master2                                                2/2     Running   0          4d4h
10.28.20.24    master2.jfmocp4.pokprv.stglabs.ibm.com   <none>           <none>
```

Example with the **mmperfmon** query command for the `read` operation:

```
# oc -n ibm-spectrum-scale rsh -c gpfs master2 mmperfmon query s3op_count,s3op_size --filter
"operation=read" -b 60 -n 3

Output:
==================================================================================
Legend:
 1:     dasS3.monitoring|DasS3|s3io|read|s3op_count
 2:     dasS3.monitoring|DasS3|s3io|read|s3op_size

Row           Timestamp s3op_count s3op_size
  1 2023-04-11-07:24:00          0         0
  2 2023-04-11-07:25:00          3       108
  3 2023-04-11-07:26:00          0         0
```

Example with the **mmperfmon** query for the `write` operation:

```
# oc -n ibm-spectrum-scale rsh -c gpfs master2 mmperfmon query s3op_count,s3op_size --filter
"operation=write" -b 60 -n 3
```

Query output:

```
Legend:
 1:     dasS3.monitoring|DasS3|s3io|write|s3op_count
 2:     dasS3.monitoring|DasS3|s3io|write|s3op_size

Row           Timestamp s3op_count s3op_size
  1 2023-04-11-07:24:00          0         0
  2 2023-04-11-07:25:00          3       108
  3 2023-04-11-07:26:00          0         0
```

S3 operations

For all S3 operations metrics statistics can be fetch with these six values:

- `s3op_count`
- `s3op_size`
- `s3op_error_count`
- `s3op_avg_time`
- `s3op_min_time`
- `s3op_max_time`

create_bucket

To collect the create bucket statistics using the **mmperfmon** query, issue the command as follows:

```
# oc -n ibm-spectrum-scale rsh -c gpfs master2 mmperfmon query
s3op_count,s3op_size,s3op_error_count,s3op_avg_time,s3op_min_time,s3op_max_time --filter
"operation=create_bucket" -b 60 -n 3
```

Query output:

```
Legend:
 1:     dasS3.monitoring|DasS3|s3ops|create_bucket|s3op_count
 2:     dasS3.monitoring|DasS3|s3ops|create_bucket|s3op_size
 3:     dasS3.monitoring|DasS3|s3ops|create_bucket|s3op_error_count
 4:     dasS3.monitoring|DasS3|s3ops|create_bucket|s3op_avg_time
 5:     dasS3.monitoring|DasS3|s3ops|create_bucket|s3op_min_time
 6:     dasS3.monitoring|DasS3|s3ops|create_bucket|s3op_max_time

Row           Timestamp s3op_count s3op_size s3op_error_count s3op_avg_time s3op_min_time
s3op_max_time
  1 2023-04-11-07:51:00          0         0                0             0
0             0
  2 2023-04-11-07:52:00          4         0                0           172
31           589
  3 2023-04-11-07:53:00          0         0                0             0
0             0
```

For file system operations use the following command:

```
# oc -n ibm-spectrum-scale rsh -c gpfs master2 mmperfmon query
s3op_count,s3op_size,s3op_error_count,s3op_avg_time,s3op_min_time,s3op_max_time --filter
"operation=mkdir" -b 60 -n 3
```

Query output:

```
Legend:
 1:     dasS3.monitoring|DasS3|s3fsops|mkdir|s3op_count
 2:     dasS3.monitoring|DasS3|s3fsops|mkdir|s3op_size
 3:     dasS3.monitoring|DasS3|s3fsops|mkdir|s3op_error_count
 4:     dasS3.monitoring|DasS3|s3fsops|mkdir|s3op_avg_time
 5:     dasS3.monitoring|DasS3|s3fsops|mkdir|s3op_min_time
 6:     dasS3.monitoring|DasS3|s3fsops|mkdir|s3op_max_time

Row           Timestamp s3op_count s3op_size s3op_error_count s3op_avg_time s3op_min_time
s3op_max_time
  1 2023-04-11-07:51:00          0         0                0             0
0             0
  2 2023-04-11-07:52:00          4         0                0          3004
442         10562
  3 2023-04-11-07:53:00          0         0                0             0
0             0
```

list_buckets

To collect the create bucket statistics using the **mmperfmon** query, issue the command as follows:

```
# oc -n ibm-spectrum-scale rsh -c gpfs master2 mmperfmon query
s3op_count,s3op_size,s3op_error_count,s3op_avg_time,s3op_min_time,s3op_max_time --filter
"operation=list_buckets" -b 60 -n 3
```

Query output:

```
Legend:
 1:     dasS3.monitoring|DasS3|s3ops|list_buckets|s3op_count
 2:     dasS3.monitoring|DasS3|s3ops|list_buckets|s3op_size
 3:     dasS3.monitoring|DasS3|s3ops|list_buckets|s3op_error_count
 4:     dasS3.monitoring|DasS3|s3ops|list_buckets|s3op_avg_time
 5:     dasS3.monitoring|DasS3|s3ops|list_buckets|s3op_min_time
 6:     dasS3.monitoring|DasS3|s3ops|list_buckets|s3op_max_time

Row           Timestamp s3op_count s3op_size s3op_error_count s3op_avg_time s3op_min_time
s3op_max_time
  1 2023-04-11-09:49:00          0         0                0             0
0             0
```

```
  2 2023-04-11-09:50:00          1         0                0             0
0             0
  3 2023-04-11-09:51:00          0         0                0             0
0             0
```

delete_bucket

- S3 operations

  To collect the delete bucket S3 operations statistics using the **mmperfmon** query, issue the command as follows:

  ```
  # oc -n ibm-spectrum-scale rsh -c gpfs master2 mmperfmon query
  s3op_count,s3op_size,s3op_error_count,s3op_avg_time,s3op_min_time,s3op_max_time --filter
  "operation=delete_bucket" -b 60 -n 3
  ```

  Query output:

  ```
  Legend:
   1:     dasS3.monitoring|DasS3|s3ops|delete_bucket|s3op_count
   2:     dasS3.monitoring|DasS3|s3ops|delete_bucket|s3op_size
   3:     dasS3.monitoring|DasS3|s3ops|delete_bucket|s3op_error_count
   4:     dasS3.monitoring|DasS3|s3ops|delete_bucket|s3op_avg_time
   5:     dasS3.monitoring|DasS3|s3ops|delete_bucket|s3op_min_time
   6:     dasS3.monitoring|DasS3|s3ops|delete_bucket|s3op_max_time

  Row           Timestamp s3op_count s3op_size s3op_error_count s3op_avg_time s3op_min_time
  s3op_max_time
    1 2023-04-11-08:10:00          0         0                0             0
  0             0
    2 2023-04-11-08:11:00          4         0                0           147
  29           490
    3 2023-04-11-08:12:00          0         0                0             0
  0             0
  ```

- File system operations

  To collect the delete bucket file system operations statistics using the **mmperfmon** query, issue the command as follows:

  ```
  # oc -n ibm-spectrum-scale rsh -c gpfs master2 mmperfmon query
  s3op_count,s3op_size,s3op_error_count,s3op_avg_time,s3op_min_time,s3op_max_time --filter
  "operation=stat" -b 60 -n 3
  ```

  Query output:

  ```
  Legend:
   1:     dasS3.monitoring|DasS3|s3fsops|stat|s3op_count
   2:     dasS3.monitoring|DasS3|s3fsops|stat|s3op_size
   3:     dasS3.monitoring|DasS3|s3fsops|stat|s3op_error_count
   4:     dasS3.monitoring|DasS3|s3fsops|stat|s3op_avg_time
   5:     dasS3.monitoring|DasS3|s3fsops|stat|s3op_min_time
   6:     dasS3.monitoring|DasS3|s3fsops|stat|s3op_max_time

  Row           Timestamp s3op_count s3op_size s3op_error_count s3op_avg_time s3op_min_time
  s3op_max_time
    1 2023-04-11-08:10:00          0         0                0             0
  0             0
    2 2023-04-11-08:11:00          8         0                0            36
  12            57
    3 2023-04-11-08:12:00          0         0                0             0
  0             0
  ```

  Sample query:

  ```
  # oc -n ibm-spectrum-scale rsh -c gpfs master2 mmperfmon query
  s3op_count,s3op_size,s3op_error_count,s3op_avg_time,s3op_min_time,s3op_max_time --filter
  "operation=realpath" -b 60 -n 6
  ```

  Query output:

  ```
  Legend:
   1:     dasS3.monitoring|DasS3|s3fsops|realpath|s3op_count
  ```

```
 2:     dasS3.monitoring|DasS3|s3fsops|realpath|s3op_size
 3:     dasS3.monitoring|DasS3|s3fsops|realpath|s3op_error_count
 4:     dasS3.monitoring|DasS3|s3fsops|realpath|s3op_avg_time
 5:     dasS3.monitoring|DasS3|s3fsops|realpath|s3op_min_time
 6:     dasS3.monitoring|DasS3|s3fsops|realpath|s3op_max_time

Row           Timestamp s3op_count s3op_size s3op_error_count s3op_avg_time s3op_min_time 
s3op_max_time
  1 2023-04-11-08:10:00          0         0                0             0             
0             0
  2 2023-04-11-08:11:00          4         0                0            61            
23          115
  3 2023-04-11-08:12:00          0         0                0             0             
0             0
```

Sample query:

```
# oc -n ibm-spectrum-scale rsh -c gpfs master2 mmperfmon query 
s3op_count,s3op_size,s3op_error_count,s3op_avg_time,s3op_min_time,s3op_max_time --filter 
"operation=checkaccess" -b 60 -n 6
```

Query output:

```
Legend:
 1:     dasS3.monitoring|DasS3|s3fsops|checkaccess|s3op_count
 2:     dasS3.monitoring|DasS3|s3fsops|checkaccess|s3op_size
 3:     dasS3.monitoring|DasS3|s3fsops|checkaccess|s3op_error_count
 4:     dasS3.monitoring|DasS3|s3fsops|checkaccess|s3op_avg_time
 5:     dasS3.monitoring|DasS3|s3fsops|checkaccess|s3op_min_time
 6:     dasS3.monitoring|DasS3|s3fsops|checkaccess|s3op_max_time

Row           Timestamp s3op_count s3op_size s3op_error_count s3op_avg_time s3op_min_time 
s3op_max_time
  1 2023-04-11-08:10:00          0         0                0             0             
0             0
  2 2023-04-11-08:11:00          4         0                0           141            
64          276
  3 2023-04-11-08:12:00          0         0                0             0             
0             0
```

Sample query:

```
# oc -n ibm-spectrum-scale rsh -c gpfs master2 mmperfmon query 
s3op_count,s3op_size,s3op_error_count,s3op_avg_time,s3op_min_time,s3op_max_time --filter 
"operation=readdir" -b 60 -n 6
```

Query output:

```
Legend:
 1:     dasS3.monitoring|DasS3|s3fsops|readdir|s3op_count
 2:     dasS3.monitoring|DasS3|s3fsops|readdir|s3op_size
 3:     dasS3.monitoring|DasS3|s3fsops|readdir|s3op_error_count
 4:     dasS3.monitoring|DasS3|s3fsops|readdir|s3op_avg_time
 5:     dasS3.monitoring|DasS3|s3fsops|readdir|s3op_min_time
 6:     dasS3.monitoring|DasS3|s3fsops|readdir|s3op_max_time

Row           Timestamp s3op_count s3op_size s3op_error_count s3op_avg_time s3op_min_time 
s3op_max_time
  1 2023-04-11-08:10:00          0         0                0             0             
0             0
  2 2023-04-11-08:11:00          8         0                0           151            
34          354
  3 2023-04-11-08:12:00          0         0                0             0             
0             0
```

Sample query:

```
# oc -n ibm-spectrum-scale rsh -c gpfs master2 mmperfmon query 
s3op_count,s3op_size,s3op_error_count,s3op_avg_time,s3op_min_time,s3op_max_time --filter 
"operation=rmdir" -b 60 -n 6
```

Query output:

```
Legend:
 1:     dasS3.monitoring|DasS3|s3fsops|rmdir|s3op_count
 2:     dasS3.monitoring|DasS3|s3fsops|rmdir|s3op_size
```

```
 3:     dasS3.monitoring|DasS3|s3fsops|rmdir|s3op_error_count
 4:     dasS3.monitoring|DasS3|s3fsops|rmdir|s3op_avg_time
 5:     dasS3.monitoring|DasS3|s3fsops|rmdir|s3op_min_time
 6:     dasS3.monitoring|DasS3|s3fsops|rmdir|s3op_max_time

Row           Timestamp s3op_count s3op_size s3op_error_count s3op_avg_time s3op_min_time 
s3op_max_time
  1 2023-04-11-08:10:00          0         0                0             0 
0             0
  2 2023-04-11-08:11:00          4         0                0           666 
318         1158
  3 2023-04-11-08:12:00          0         0                0             0 
0             0
```

upload_object

- I/O operations

  To collect the upload object I/O operations statistics using the **mmperfmon** query, issue the command as follows:

```
# oc -n ibm-spectrum-scale rsh -c gpfs master2 mmperfmon query 
s3op_count,s3op_size,s3op_error_count,s3op_avg_time,s3op_min_time,s3op_max_time --filter 
"operation=write" -b 60 -n 3
```

  Query output

```
Legend:
 1:     dasS3.monitoring|DasS3|s3io|write|s3op_count
 2:     dasS3.monitoring|DasS3|s3io|write|s3op_size
 3:     dasS3.monitoring|DasS3|s3io|write|s3op_error_count
 4:     dasS3.monitoring|DasS3|s3io|write|s3op_avg_time
 5:     dasS3.monitoring|DasS3|s3io|write|s3op_min_time
 6:     dasS3.monitoring|DasS3|s3io|write|s3op_max_time

Row           Timestamp s3op_count s3op_size s3op_error_count s3op_avg_time s3op_min_time 
s3op_max_time
  1 2023-04-11-07:24:00          0         0                0             0 
0             0
  2 2023-04-11-07:25:00          3       108                0             0 
0             0
  3 2023-04-11-07:26:00          0         0                0             0 
0             0
```

- S3 operations

  To collect the upload object S3 operations statistics using the **mmperfmon** query, issue the command as follows:

```
# oc -n ibm-spectrum-scale rsh -c gpfs master2 mmperfmon query 
s3op_count,s3op_size,s3op_error_count,s3op_avg_time,s3op_min_time,s3op_max_time --filter 
"operation=upload_object" -b 60 -n 3
```

  Query output:

```
Legend:
 1:     dasS3.monitoring|DasS3|s3ops|upload_object|s3op_count
 2:     dasS3.monitoring|DasS3|s3ops|upload_object|s3op_size
 3:     dasS3.monitoring|DasS3|s3ops|upload_object|s3op_error_count
 4:     dasS3.monitoring|DasS3|s3ops|upload_object|s3op_avg_time
 5:     dasS3.monitoring|DasS3|s3ops|upload_object|s3op_min_time
 6:     dasS3.monitoring|DasS3|s3ops|upload_object|s3op_max_time

Row           Timestamp s3op_count s3op_size s3op_error_count s3op_avg_time s3op_min_time 
s3op_max_time
  2 2023-04-11-07:24:00          0         0                0             0 
0             0
  3 2023-04-11-07:25:00          3         0                0            29 
14           41
  4 2023-04-11-07:26:00          0         0                0             0 
0             0
```

- File system operations

To collect the upload object file system operations statistics using the **mmperfmon** query, issue the command as follows:

```
# oc -n ibm-spectrum-scale rsh -c gpfs master2 mmperfmon query
s3op_count,s3op_size,s3op_error_count,s3op_avg_time,s3op_min_time,s3op_max_time --filter
"operation=stat" -b 60 -n 3
```

Query output:

```
Legend:
 1:     dasS3.monitoring|DasS3|s3fsops|stat|s3op_count
 2:     dasS3.monitoring|DasS3|s3fsops|stat|s3op_size
 3:     dasS3.monitoring|DasS3|s3fsops|stat|s3op_error_count
 4:     dasS3.monitoring|DasS3|s3fsops|stat|s3op_avg_time
 5:     dasS3.monitoring|DasS3|s3fsops|stat|s3op_min_time
 6:     dasS3.monitoring|DasS3|s3fsops|stat|s3op_max_time

Row           Timestamp s3op_count s3op_size s3op_error_count s3op_avg_time s3op_min_time
s3op_max_time
  1 2023-04-11-07:24:00          0         0                0             0
0             0
  2 2023-04-11-07:25:00          6         0                0            35
18           52
  3 2023-04-11-07:26:00          0         0                0             0
0             0
```

Sample query:

```
# oc -n ibm-spectrum-scale rsh -c gpfs master2 mmperfmon query
s3op_count,s3op_size,s3op_error_count,s3op_avg_time,s3op_min_time,s3op_max_time --filter
"operation=realpath" -b 60 -n 3
```

Query output:

```
Legend:
 1:     dasS3.monitoring|DasS3|s3fsops|realpath|s3op_count
 2:     dasS3.monitoring|DasS3|s3fsops|realpath|s3op_size
 3:     dasS3.monitoring|DasS3|s3fsops|realpath|s3op_error_count
 4:     dasS3.monitoring|DasS3|s3fsops|realpath|s3op_avg_time
 5:     dasS3.monitoring|DasS3|s3fsops|realpath|s3op_min_time
 6:     dasS3.monitoring|DasS3|s3fsops|realpath|s3op_max_time

Row           Timestamp s3op_count s3op_size s3op_error_count s3op_avg_time s3op_min_time
s3op_max_time
  1 2023-04-11-07:24:00          0         0                0             0
0             0
  2 2023-04-11-07:25:00          4         0                1            58
15          207
  3 2023-04-11-07:26:00          0         0                0             0
0             0
```

Sample query:

```
# oc -n ibm-spectrum-scale rsh -c gpfs master2 mmperfmon query
s3op_count,s3op_size,s3op_error_count,s3op_avg_time,s3op_min_time,s3op_max_time --filter
"operation=fsync" -b 60 -n 3
```

Query output:

```
Legend:
 1:     dasS3.monitoring|DasS3|s3fsops|fsync|s3op_count
 2:     dasS3.monitoring|DasS3|s3fsops|fsync|s3op_size
 3:     dasS3.monitoring|DasS3|s3fsops|fsync|s3op_error_count
 4:     dasS3.monitoring|DasS3|s3fsops|fsync|s3op_avg_time
 5:     dasS3.monitoring|DasS3|s3fsops|fsync|s3op_min_time
 6:     dasS3.monitoring|DasS3|s3fsops|fsync|s3op_max_time

Row           Timestamp s3op_count s3op_size s3op_error_count s3op_avg_time s3op_min_time
s3op_max_time
  1 2023-04-11-07:24:00          0         0                0             0
0             0
  2 2023-04-11-07:25:00          9         0                0          3936
76        18290
  3 2023-04-11-07:26:00          0         0                0             0
0             0
```

Sample query:

```
# oc -n ibm-spectrum-scale rsh -c gpfs master2 mmperfmon query
s3op_count,s3op_size,s3op_error_count,s3op_avg_time,s3op_min_time,s3op_max_time --filter
"operation=fileopen" -b 60 -n 3
```

Query output:

```
Legend:
 1:     dasS3.monitoring|DasS3|s3fsops|fileopen|s3op_count
 2:     dasS3.monitoring|DasS3|s3fsops|fileopen|s3op_size
 3:     dasS3.monitoring|DasS3|s3fsops|fileopen|s3op_error_count
 4:     dasS3.monitoring|DasS3|s3fsops|fileopen|s3op_avg_time
 5:     dasS3.monitoring|DasS3|s3fsops|fileopen|s3op_min_time
 6:     dasS3.monitoring|DasS3|s3fsops|fileopen|s3op_max_time

Row           Timestamp s3op_count s3op_size s3op_error_count s3op_avg_time s3op_min_time
s3op_max_time
  1 2023-04-11-07:24:00          0         0                0             0
0             0
  2 2023-04-11-07:25:00          3         0                0          3008
576          6212
  3 2023-04-11-07:26:00          0         0                0             0
0             0
```

Sample query:

```
# oc -n ibm-spectrum-scale rsh -c gpfs master2 mmperfmon query
s3op_count,s3op_size,s3op_error_count,s3op_avg_time,s3op_min_time,s3op_max_time --filter
"operation=filewritev" -b 60 -n 3
```

Query output:

```
Legend:
 1:     dasS3.monitoring|DasS3|s3fsops|filewritev|s3op_count
 2:     dasS3.monitoring|DasS3|s3fsops|filewritev|s3op_size
 3:     dasS3.monitoring|DasS3|s3fsops|filewritev|s3op_error_count
 4:     dasS3.monitoring|DasS3|s3fsops|filewritev|s3op_avg_time
 5:     dasS3.monitoring|DasS3|s3fsops|filewritev|s3op_min_time
 6:     dasS3.monitoring|DasS3|s3fsops|filewritev|s3op_max_time

Row           Timestamp s3op_count s3op_size s3op_error_count s3op_avg_time s3op_min_time
s3op_max_time
  1 2023-04-11-07:24:00          0         0                0             0
0             0
  2 2023-04-11-07:25:00          3         0                0           260
153          363
  3 2023-04-11-07:26:00          0         0                0             0
0             0
```

Sample query:

```
# oc -n ibm-spectrum-scale rsh -c gpfs master2 mmperfmon query
s3op_count,s3op_size,s3op_error_count,s3op_avg_time,s3op_min_time,s3op_max_time --filter
"operation=filefsync" -b 60 -n 3
```

Query output:

```
Legend:
 1:     dasS3.monitoring|DasS3|s3fsops|filefsync|s3op_count
 2:     dasS3.monitoring|DasS3|s3fsops|filefsync|s3op_size
 3:     dasS3.monitoring|DasS3|s3fsops|filefsync|s3op_error_count
 4:     dasS3.monitoring|DasS3|s3fsops|filefsync|s3op_avg_time
 5:     dasS3.monitoring|DasS3|s3fsops|filefsync|s3op_min_time
 6:     dasS3.monitoring|DasS3|s3fsops|filefsync|s3op_max_time

Row           Timestamp s3op_count s3op_size s3op_error_count s3op_avg_time s3op_min_time
s3op_max_time
  1 2023-04-11-07:24:00          0         0                0             0
0             0
  2 2023-04-11-07:25:00          3         0                0          4342
3675          5080
  3 2023-04-11-07:26:00          0         0                0             0
0             0
```

Sample query:

```
# oc -n ibm-spectrum-scale rsh -c gpfs master2 mmperfmon query
s3op_count,s3op_size,s3op_error_count,s3op_avg_time,s3op_min_time,s3op_max_time --filter
"operation=fileclose" -b 60 -n 3
```

Query output:

```
Legend:
 1:     dasS3.monitoring|DasS3|s3fsops|fileclose|s3op_count
 2:     dasS3.monitoring|DasS3|s3fsops|fileclose|s3op_size
 3:     dasS3.monitoring|DasS3|s3fsops|fileclose|s3op_error_count
 4:     dasS3.monitoring|DasS3|s3fsops|fileclose|s3op_avg_time
 5:     dasS3.monitoring|DasS3|s3fsops|fileclose|s3op_min_time
 6:     dasS3.monitoring|DasS3|s3fsops|fileclose|s3op_max_time

Row           Timestamp s3op_count s3op_size s3op_error_count s3op_avg_time s3op_min_time
s3op_max_time
  1 2023-04-11-07:24:00          0         0                0             0
0             0
  2 2023-04-11-07:25:00          3         0                0            34
29            45
  3 2023-04-11-07:26:00          0         0                0             0
0             0
```

Sample query:

```
# oc -n ibm-spectrum-scale rsh -c gpfs master2 mmperfmon query
s3op_count,s3op_size,s3op_error_count,s3op_avg_time,s3op_min_time,s3op_max_time --filter
"operation=rename" -b 60 -n 3
```

Query output:

```
Legend:
 1:     dasS3.monitoring|DasS3|s3fsops|rename|s3op_count
 2:     dasS3.monitoring|DasS3|s3fsops|rename|s3op_size
 3:     dasS3.monitoring|DasS3|s3fsops|rename|s3op_error_count
 4:     dasS3.monitoring|DasS3|s3fsops|rename|s3op_avg_time
 5:     dasS3.monitoring|DasS3|s3fsops|rename|s3op_min_time
 6:     dasS3.monitoring|DasS3|s3fsops|rename|s3op_max_time

Row           Timestamp s3op_count s3op_size s3op_error_count s3op_avg_time s3op_min_time
s3op_max_time
  1 2023-04-11-07:24:00          0         0                0             0
0             0
  2 2023-04-11-07:25:00          3         0                0           563
314           881
  3 2023-04-11-07:26:00          0         0                0             0
0             0
```

Sample query:

```
# oc -n ibm-spectrum-scale rsh -c gpfs master2 mmperfmon query
s3op_count,s3op_size,s3op_error_count,s3op_avg_time,s3op_min_time,s3op_max_time --filter
"operation=mkdir" -b 60 -n 3
```

Query output:

```
Legend:
 1:     dasS3.monitoring|DasS3|s3fsops|mkdir|s3op_count
 2:     dasS3.monitoring|DasS3|s3fsops|mkdir|s3op_size
 3:     dasS3.monitoring|DasS3|s3fsops|mkdir|s3op_error_count
 4:     dasS3.monitoring|DasS3|s3fsops|mkdir|s3op_avg_time
 5:     dasS3.monitoring|DasS3|s3fsops|mkdir|s3op_min_time
 6:     dasS3.monitoring|DasS3|s3fsops|mkdir|s3op_max_time

Row           Timestamp s3op_count s3op_size s3op_error_count s3op_avg_time s3op_min_time
s3op_max_time
  2 2023-04-11-07:24:00          0         0                0             0
0             0
  3 2023-04-11-07:25:00         24         0               22           271
15          6083
  4 2023-04-11-07:26:00          0         0                0             0
0             0
```

list_objects

- S3 operations

  To collect the list object S3 operations statistics using the **mmperfmon** query, issue the command as follows:

```
# oc -n ibm-spectrum-scale rsh -c gpfs master2 mmperfmon query
s3op_count,s3op_size,s3op_error_count,s3op_avg_time,s3op_min_time,s3op_max_time --filter
"operation=list_objects" -b 60 -n 3
```

  Query output:

```
Legend:
 1:     dasS3.monitoring|DasS3|s3ops|list_objects|s3op_count
 2:     dasS3.monitoring|DasS3|s3ops|list_objects|s3op_size
 3:     dasS3.monitoring|DasS3|s3ops|list_objects|s3op_error_count
 4:     dasS3.monitoring|DasS3|s3ops|list_objects|s3op_avg_time
 5:     dasS3.monitoring|DasS3|s3ops|list_objects|s3op_min_time
 6:     dasS3.monitoring|DasS3|s3ops|list_objects|s3op_max_time

Row           Timestamp s3op_count s3op_size s3op_error_count s3op_avg_time s3op_min_time
s3op_max_time
  1 2023-04-11-10:00:00          0         0                0             0
0             0
  2 2023-04-11-10:01:00          1         0                0             4
4             4
  3 2023-04-11-10:02:00          0         0                0             0
0             0
```

- File system operations

  To collect the list objects file system operations statistics using the **mmperfmon** query, issue the command as follows:

```
# oc -n ibm-spectrum-scale rsh -c gpfs master2 mmperfmon query
s3op_count,s3op_size,s3op_error_count,s3op_avg_time,s3op_min_time,s3op_max_time --filter
"operation=stat" -b 60 -n 3
```

  Query output:

```
Legend:
 1:     dasS3.monitoring|DasS3|s3fsops|stat|s3op_count
 2:     dasS3.monitoring|DasS3|s3fsops|stat|s3op_size
 3:     dasS3.monitoring|DasS3|s3fsops|stat|s3op_error_count
 4:     dasS3.monitoring|DasS3|s3fsops|stat|s3op_avg_time
 5:     dasS3.monitoring|DasS3|s3fsops|stat|s3op_min_time
 6:     dasS3.monitoring|DasS3|s3fsops|stat|s3op_max_time

Row           Timestamp s3op_count s3op_size s3op_error_count s3op_avg_time s3op_min_time
s3op_max_time
  1 2023-04-11-10:00:00          0         0                0             0
0             0
  2 2023-04-11-10:01:00          9         0                0            11
2            45
  3 2023-04-11-10:02:00          0         0                0             0
0             0
```

  Sample query:

```
# oc -n ibm-spectrum-scale rsh -c gpfs master2 mmperfmon query
s3op_count,s3op_size,s3op_error_count,s3op_avg_time,s3op_min_time,s3op_max_time --filter
"operation=realpath" -b 60 -n 3
```

  Query output:

```
Legend:
 1:     dasS3.monitoring|DasS3|s3fsops|realpath|s3op_count
 2:     dasS3.monitoring|DasS3|s3fsops|realpath|s3op_size
 3:     dasS3.monitoring|DasS3|s3fsops|realpath|s3op_error_count
 4:     dasS3.monitoring|DasS3|s3fsops|realpath|s3op_avg_time
 5:     dasS3.monitoring|DasS3|s3fsops|realpath|s3op_min_time
 6:     dasS3.monitoring|DasS3|s3fsops|realpath|s3op_max_time

Row           Timestamp s3op_count s3op_size s3op_error_count s3op_avg_time s3op_min_time
```

```
s3op_max_time
  1 2023-04-11-10:00:00          0         0                0             0
0             0
  2 2023-04-11-10:01:00          7         0                0            20
12            56
  3 2023-04-11-10:02:00          0         0                0             0
0             0
```

Sample query:

```
# oc -n ibm-spectrum-scale rsh -c gpfs master2 mmperfmon query
s3op_count,s3op_size,s3op_error_count,s3op_avg_time,s3op_min_time,s3op_max_time --filter
"operation=checkaccess" -b 60 -n 3
```

Query output:

```
Legend:
 1:     dasS3.monitoring|DasS3|s3fsops|checkaccess|s3op_count
 2:     dasS3.monitoring|DasS3|s3fsops|checkaccess|s3op_size
 3:     dasS3.monitoring|DasS3|s3fsops|checkaccess|s3op_error_count
 4:     dasS3.monitoring|DasS3|s3fsops|checkaccess|s3op_avg_time
 5:     dasS3.monitoring|DasS3|s3fsops|checkaccess|s3op_min_time
 6:     dasS3.monitoring|DasS3|s3fsops|checkaccess|s3op_max_time

Row           Timestamp s3op_count s3op_size s3op_error_count s3op_avg_time s3op_min_time
s3op_max_time
  1 2023-04-11-10:00:00          0         0                0             0
0             0
  2 2023-04-11-10:01:00          1         0                0            49
49            49
  3 2023-04-11-10:02:00          0         0                0             0
0             0
```

Sample query:

```
# oc -n ibm-spectrum-scale rsh -c gpfs master2 mmperfmon query
s3op_count,s3op_size,s3op_error_count,s3op_avg_time,s3op_min_time,s3op_max_time --filter
"operation=readdir" -b 60 -n 3
```

Query output:

```
Legend:
 1:     dasS3.monitoring|DasS3|s3fsops|readdir|s3op_count
 2:     dasS3.monitoring|DasS3|s3fsops|readdir|s3op_size
 3:     dasS3.monitoring|DasS3|s3fsops|readdir|s3op_error_count
 4:     dasS3.monitoring|DasS3|s3fsops|readdir|s3op_avg_time
 5:     dasS3.monitoring|DasS3|s3fsops|readdir|s3op_min_time
 6:     dasS3.monitoring|DasS3|s3fsops|readdir|s3op_max_time

Row           Timestamp s3op_count s3op_size s3op_error_count s3op_avg_time s3op_min_time
s3op_max_time
  1 2023-04-11-10:00:00          0         0                0             0
0             0
  2 2023-04-11-10:01:00          1         0                0            68
68            68
  3 2023-04-11-10:02:00          0         0                0             0
0             0
```

delete_objects

- S3 operations

  To collect the delete object S3 operations statistics using the **mmperfmon** query, issue the command as follows:

```
# oc -n ibm-spectrum-scale rsh -c gpfs master2 mmperfmon query
s3op_count,s3op_size,s3op_error_count,s3op_avg_time,s3op_min_time,s3op_max_time --filter
"operation=delete_object" -b 60 -n 3
```

Query output:

```
Legend:
 1:     dasS3.monitoring|DasS3|s3ops|delete_object|s3op_count
 2:     dasS3.monitoring|DasS3|s3ops|delete_object|s3op_size
```

```
 3:     dasS3.monitoring|DasS3|s3ops|delete_object|s3op_error_count
 4:     dasS3.monitoring|DasS3|s3ops|delete_object|s3op_avg_time
 5:     dasS3.monitoring|DasS3|s3ops|delete_object|s3op_min_time
 6:     dasS3.monitoring|DasS3|s3ops|delete_object|s3op_max_time

Row           Timestamp s3op_count s3op_size s3op_error_count s3op_avg_time s3op_min_time 
s3op_max_time
  1 2023-04-11-08:52:00          0         0                0             0
0             0
  2 2023-04-11-08:53:00          2         0                0             2
2             2
  3 2023-04-11-08:54:00          0         0                0             0
0             0
```

- File system operations

  To collect the delete object file system operations statistics using the **mmperfmon** query, issue the command as follows:

```
# oc -n ibm-spectrum-scale rsh -c gpfs master2 mmperfmon query
s3op_count,s3op_size,s3op_error_count,s3op_avg_time,s3op_min_time,s3op_max_time --filter
"operation=stat" -b 60 -n 3
```

  Query output:

```
Legend:
 1:     dasS3.monitoring|DasS3|s3fsops|stat|s3op_count
 2:     dasS3.monitoring|DasS3|s3fsops|stat|s3op_size
 3:     dasS3.monitoring|DasS3|s3fsops|stat|s3op_error_count
 4:     dasS3.monitoring|DasS3|s3fsops|stat|s3op_avg_time
 5:     dasS3.monitoring|DasS3|s3fsops|stat|s3op_min_time
 6:     dasS3.monitoring|DasS3|s3fsops|stat|s3op_max_time

Row           Timestamp s3op_count s3op_size s3op_error_count s3op_avg_time s3op_min_time 
s3op_max_time
  1 2023-04-11-08:52:00          0         0                0             0
0             0
  2 2023-04-11-08:53:00          2         0                0            30
25            36
  3 2023-04-11-08:54:00          0         0                0             0
0             0
```

  Sample query:

```
# oc -n ibm-spectrum-scale rsh -c gpfs master2 mmperfmon query
s3op_count,s3op_size,s3op_error_count,s3op_avg_time,s3op_min_time,s3op_max_time --filter
"operation=realpath" -b 60 -n 3
```

  Query output:

```
Legend:
 1:     dasS3.monitoring|DasS3|s3fsops|realpath|s3op_count
 2:     dasS3.monitoring|DasS3|s3fsops|realpath|s3op_size
 3:     dasS3.monitoring|DasS3|s3fsops|realpath|s3op_error_count
 4:     dasS3.monitoring|DasS3|s3fsops|realpath|s3op_avg_time
 5:     dasS3.monitoring|DasS3|s3fsops|realpath|s3op_min_time
 6:     dasS3.monitoring|DasS3|s3fsops|realpath|s3op_max_time

Row           Timestamp s3op_count s3op_size s3op_error_count s3op_avg_time s3op_min_time 
s3op_max_time
  1 2023-04-11-08:52:00          0         0                0             0
0             0
  2 2023-04-11-08:53:00          2         0                0            81
26           136
  3 2023-04-11-08:54:00          0         0                0             0
0             0
```

  Sample query:

```
# oc -n ibm-spectrum-scale rsh -c gpfs master2 mmperfmon query
s3op_count,s3op_size,s3op_error_count,s3op_avg_time,s3op_min_time,s3op_max_time --filter
"operation=unlink" -b 60 -n 3
```

  Query output:

```
Legend:
 1:     dasS3.monitoring|DasS3|s3fsops|unlink|s3op_count
 2:     dasS3.monitoring|DasS3|s3fsops|unlink|s3op_size
 3:     dasS3.monitoring|DasS3|s3fsops|unlink|s3op_error_count
 4:     dasS3.monitoring|DasS3|s3fsops|unlink|s3op_avg_time
 5:     dasS3.monitoring|DasS3|s3fsops|unlink|s3op_min_time
 6:     dasS3.monitoring|DasS3|s3fsops|unlink|s3op_max_time

Row           Timestamp s3op_count s3op_size s3op_error_count s3op_avg_time s3op_min_time 
s3op_max_time
  1 2023-04-11-08:52:00          0         0                0             0
0             0
  2 2023-04-11-08:53:00          2         0                0           757
440          1074
  3 2023-04-11-08:54:00          0         0                0             0
0             0
```

head_object and read_object

- I/O operations

  To collect the I/O operations statistics using the **mmperfmon** query, issue the command as follows:

```
# oc -n ibm-spectrum-scale rsh -c gpfs master2 mmperfmon query 
s3op_count,s3op_size,s3op_error_count,s3op_avg_time,s3op_min_time,s3op_max_time --filter 
"operation=read" -b 60 -n 3
```

  Query output:

```
Legend:
 1:     dasS3.monitoring|DasS3|s3io|read|s3op_count
 2:     dasS3.monitoring|DasS3|s3io|read|s3op_size
 3:     dasS3.monitoring|DasS3|s3io|read|s3op_error_count
 4:     dasS3.monitoring|DasS3|s3io|read|s3op_avg_time
 5:     dasS3.monitoring|DasS3|s3io|read|s3op_min_time
 6:     dasS3.monitoring|DasS3|s3io|read|s3op_max_time

Row           Timestamp s3op_count s3op_size s3op_error_count s3op_avg_time s3op_min_time 
s3op_max_time
  2 2023-04-11-09:31:00          0         0                0             0
0             0
  3 2023-04-11-09:32:00          8  67108864                0             0
0             0
  4 2023-04-11-09:33:00          0         0                0             0
0             0
```

- S3 operations

  To collect the S3 operations statistics using the **mmperfmon** query, issue the command as follows:

```
# oc -n ibm-spectrum-scale rsh -c gpfs master2 mmperfmon query 
s3op_count,s3op_size,s3op_error_count,s3op_avg_time,s3op_min_time,s3op_max_time --filter 
"operation=head_object" -b 60 -n 3
```

  Query output:

```
Legend:
 1:     dasS3.monitoring|DasS3|s3ops|head_object|s3op_count
 2:     dasS3.monitoring|DasS3|s3ops|head_object|s3op_size
 3:     dasS3.monitoring|DasS3|s3ops|head_object|s3op_error_count
 4:     dasS3.monitoring|DasS3|s3ops|head_object|s3op_avg_time
 5:     dasS3.monitoring|DasS3|s3ops|head_object|s3op_min_time
 6:     dasS3.monitoring|DasS3|s3ops|head_object|s3op_max_time

Row           Timestamp s3op_count s3op_size s3op_error_count s3op_avg_time s3op_min_time 
s3op_max_time
  1 2023-04-11-09:31:00          0         0                0             0
0             0
  2 2023-04-11-09:32:00          9         0                0            11
2            25
  3 2023-04-11-09:33:00          0         0                0             0
0             0
```

  Sample query:

```
# oc -n ibm-spectrum-scale rsh -c gpfs master2 mmperfmon query
s3op_count,s3op_size,s3op_error_count,s3op_avg_time,s3op_min_time,s3op_max_time --filter
"operation=read_object" -b 60 -n 3
```

Query output:

```
Legend:
 1:     dasS3.monitoring|DasS3|s3ops|read_object|s3op_count
 2:     dasS3.monitoring|DasS3|s3ops|read_object|s3op_size
 3:     dasS3.monitoring|DasS3|s3ops|read_object|s3op_error_count
 4:     dasS3.monitoring|DasS3|s3ops|read_object|s3op_avg_time
 5:     dasS3.monitoring|DasS3|s3ops|read_object|s3op_min_time
 6:     dasS3.monitoring|DasS3|s3ops|read_object|s3op_max_time

Row           Timestamp s3op_count s3op_size s3op_error_count s3op_avg_time s3op_min_time
s3op_max_time
  1 2023-04-11-09:31:00          0         0                0             0
0             0
  2 2023-04-11-09:32:00          8         0                0           174
155           207
  3 2023-04-11-09:33:00          0         0                0             0
0             0
```

- File system operations

  To collect the file system operations statistics using the **mmperfmon** query, issue the command as follows:

```
# oc -n ibm-spectrum-scale rsh -c gpfs master2 mmperfmon query
s3op_count,s3op_size,s3op_error_count,s3op_avg_time,s3op_min_time,s3op_max_time --filter
"operation=realpath" -b 60 -n 3
```

Query output:

```
Legend:
 1:     dasS3.monitoring|DasS3|s3fsops|realpath|s3op_count
 2:     dasS3.monitoring|DasS3|s3fsops|realpath|s3op_size
 3:     dasS3.monitoring|DasS3|s3fsops|realpath|s3op_error_count
 4:     dasS3.monitoring|DasS3|s3fsops|realpath|s3op_avg_time
 5:     dasS3.monitoring|DasS3|s3fsops|realpath|s3op_min_time
 6:     dasS3.monitoring|DasS3|s3fsops|realpath|s3op_max_time

Row           Timestamp s3op_count s3op_size s3op_error_count s3op_avg_time s3op_min_time
s3op_max_time
  1 2023-04-11-09:31:00          0         0                0             0
0             0
  2 2023-04-11-09:32:00         17         0                0            39
13            84
  3 2023-04-11-09:33:00          0         0                0             0
0             0
```

Sample query:

```
# oc -n ibm-spectrum-scale rsh -c gpfs master2 mmperfmon query
s3op_count,s3op_size,s3op_error_count,s3op_avg_time,s3op_min_time,s3op_max_time --filter
"operation=stat" -b 60 -n 3
```

Query output:

```
Legend:
 1:     dasS3.monitoring|DasS3|s3fsops|stat|s3op_count
 2:     dasS3.monitoring|DasS3|s3fsops|stat|s3op_size
 3:     dasS3.monitoring|DasS3|s3fsops|stat|s3op_error_count
 4:     dasS3.monitoring|DasS3|s3fsops|stat|s3op_avg_time
 5:     dasS3.monitoring|DasS3|s3fsops|stat|s3op_min_time
 6:     dasS3.monitoring|DasS3|s3fsops|stat|s3op_max_time

Row           Timestamp s3op_count s3op_size s3op_error_count s3op_avg_time s3op_min_time
s3op_max_time
  1 2023-04-11-09:31:00          0         0                0             0
0             0
  2 2023-04-11-09:32:00         17         0                0            15
2            40
  3 2023-04-11-09:33:00          0         0                0             0
0             0
```

Sample query:

```
# oc -n ibm-spectrum-scale rsh -c gpfs master2 mmperfmon query
s3op_count,s3op_size,s3op_error_count,s3op_avg_time,s3op_min_time,s3op_max_time --filter
"operation=fileopen" -b 60 -n 3
```

Query output:

```
Legend:
 1:     dasS3.monitoring|DasS3|s3fsops|fileopen|s3op_count
 2:     dasS3.monitoring|DasS3|s3fsops|fileopen|s3op_size
 3:     dasS3.monitoring|DasS3|s3fsops|fileopen|s3op_error_count
 4:     dasS3.monitoring|DasS3|s3fsops|fileopen|s3op_avg_time
 5:     dasS3.monitoring|DasS3|s3fsops|fileopen|s3op_min_time
 6:     dasS3.monitoring|DasS3|s3fsops|fileopen|s3op_max_time

Row           Timestamp s3op_count s3op_size s3op_error_count s3op_avg_time s3op_min_time
s3op_max_time
  1 2023-04-11-09:31:00          0         0                0             0
0             0
  2 2023-04-11-09:32:00         17         0                0            35
6            84
  3 2023-04-11-09:33:00          0         0                0             0
0             0
```

Sample query:

```
# oc -n ibm-spectrum-scale rsh -c gpfs master2 mmperfmon query
s3op_count,s3op_size,s3op_error_count,s3op_avg_time,s3op_min_time,s3op_max_time --filter
"operation=filestat" -b 60 -n 3
```

Query output:

```
Legend:
 1:     dasS3.monitoring|DasS3|s3fsops|filestat|s3op_count
 2:     dasS3.monitoring|DasS3|s3fsops|filestat|s3op_size
 3:     dasS3.monitoring|DasS3|s3fsops|filestat|s3op_error_count
 4:     dasS3.monitoring|DasS3|s3fsops|filestat|s3op_avg_time
 5:     dasS3.monitoring|DasS3|s3fsops|filestat|s3op_min_time
 6:     dasS3.monitoring|DasS3|s3fsops|filestat|s3op_max_time

Row           Timestamp s3op_count s3op_size s3op_error_count s3op_avg_time s3op_min_time
s3op_max_time
  1 2023-04-11-09:31:00          0         0                0             0
0             0
  2 2023-04-11-09:32:00          9         0                0             3
0             8
  3 2023-04-11-09:33:00          0         0                0             0
0             0
```

Sample query:

```
# oc -n ibm-spectrum-scale rsh -c gpfs master2 mmperfmon query
s3op_count,s3op_size,s3op_error_count,s3op_avg_time,s3op_min_time,s3op_max_time --filter
"operation=filegetxattr" -b 60 -n 3
```

Query output:

```
Legend:
 1:     dasS3.monitoring|DasS3|s3fsops|filegetxattr|s3op_count
 2:     dasS3.monitoring|DasS3|s3fsops|filegetxattr|s3op_size
 3:     dasS3.monitoring|DasS3|s3fsops|filegetxattr|s3op_error_count
 4:     dasS3.monitoring|DasS3|s3fsops|filegetxattr|s3op_avg_time
 5:     dasS3.monitoring|DasS3|s3fsops|filegetxattr|s3op_min_time
 6:     dasS3.monitoring|DasS3|s3fsops|filegetxattr|s3op_max_time

Row           Timestamp s3op_count s3op_size s3op_error_count s3op_avg_time s3op_min_time
s3op_max_time
  1 2023-04-11-09:31:00          0         0                0             0
0             0
  2 2023-04-11-09:32:00          9         0                0            13
2            55
  3 2023-04-11-09:33:00          0         0                0             0
0             0
```

Sample query:

```
# oc -n ibm-spectrum-scale rsh -c gpfs master2 mmperfmon query
s3op_count,s3op_size,s3op_error_count,s3op_avg_time,s3op_min_time,s3op_max_time --filter
"operation=fileclose" -b 60 -n 3
```

Query output:

```
Legend:
 1:     dasS3.monitoring|DasS3|s3fsops|fileclose|s3op_count
 2:     dasS3.monitoring|DasS3|s3fsops|fileclose|s3op_size
 3:     dasS3.monitoring|DasS3|s3fsops|fileclose|s3op_error_count
 4:     dasS3.monitoring|DasS3|s3fsops|fileclose|s3op_avg_time
 5:     dasS3.monitoring|DasS3|s3fsops|fileclose|s3op_min_time
 6:     dasS3.monitoring|DasS3|s3fsops|fileclose|s3op_max_time

Row           Timestamp s3op_count s3op_size s3op_error_count s3op_avg_time s3op_min_time
s3op_max_time
  1 2023-04-11-09:31:00          0         0                0             0
0             0
  2 2023-04-11-09:32:00         17         0                0            30
2           130
  3 2023-04-11-09:33:00          0         0                0             0
0             0
```

Sample query:

```
# oc -n ibm-spectrum-scale rsh -c gpfs master2 mmperfmon query
s3op_count,s3op_size,s3op_error_count,s3op_avg_time,s3op_min_time,s3op_max_time --filter
"operation=fileread" -b 60 -n 3
```

Query output:

```
Legend:
 1:     dasS3.monitoring|DasS3|s3fsops|fileread|s3op_count
 2:     dasS3.monitoring|DasS3|s3fsops|fileread|s3op_size
 3:     dasS3.monitoring|DasS3|s3fsops|fileread|s3op_error_count
 4:     dasS3.monitoring|DasS3|s3fsops|fileread|s3op_avg_time
 5:     dasS3.monitoring|DasS3|s3fsops|fileread|s3op_min_time
 6:     dasS3.monitoring|DasS3|s3fsops|fileread|s3op_max_time

Row           Timestamp s3op_count s3op_size s3op_error_count s3op_avg_time s3op_min_time
s3op_max_time
  1 2023-04-11-09:31:00          0         0                0             0
0             0
  2 2023-04-11-09:32:00          8         0                0          1866
1385          2263
  3 2023-04-11-09:33:00          0         0                0             0
0             0
```

initiate_multipart, upload_part, and complete_object_upload

- I/O operations

  To collect the I/O operations statistics using the **mmperfmon** query, issue the command as follows:

```
# oc -n ibm-spectrum-scale rsh -c gpfs master2 mmperfmon query
s3op_count,s3op_size,s3op_error_count,s3op_avg_time,s3op_min_time,s3op_max_time --filter
"operation=write" -b 60 -n 3
```

Query output:

```
Legend:
 1:     dasS3.monitoring|DasS3|s3io|write|s3op_count
 2:     dasS3.monitoring|DasS3|s3io|write|s3op_size
 3:     dasS3.monitoring|DasS3|s3io|write|s3op_error_count
 4:     dasS3.monitoring|DasS3|s3io|write|s3op_avg_time
 5:     dasS3.monitoring|DasS3|s3io|write|s3op_min_time
 6:     dasS3.monitoring|DasS3|s3io|write|s3op_max_time

Row           Timestamp s3op_count s3op_size s3op_error_count s3op_avg_time s3op_min_time
s3op_max_time
  1 2023-04-11-09:01:00          0         0                0             0
0             0
```

```
  2 2023-04-11-09:02:00          8  67108864                0             0
0             0
  3 2023-04-11-09:03:00          0         0                0             0
0             0
```

- S3 operations

  To collect the S3 operations statistics using the **mmperfmon** query, issue the command as follows:

```
# oc -n ibm-spectrum-scale rsh -c gpfs master2 mmperfmon query
s3op_count,s3op_size,s3op_error_count,s3op_avg_time,s3op_min_time,s3op_max_time --filter
"operation=initiate_multipart" -b 60 -n 3
```

  Query output:

```
Legend:
 1:     dasS3.monitoring|DasS3|s3ops|initiate_multipart|s3op_count
 2:     dasS3.monitoring|DasS3|s3ops|initiate_multipart|s3op_size
 3:     dasS3.monitoring|DasS3|s3ops|initiate_multipart|s3op_error_count
 4:     dasS3.monitoring|DasS3|s3ops|initiate_multipart|s3op_avg_time
 5:     dasS3.monitoring|DasS3|s3ops|initiate_multipart|s3op_min_time
 6:     dasS3.monitoring|DasS3|s3ops|initiate_multipart|s3op_max_time

Row           Timestamp s3op_count s3op_size s3op_error_count s3op_avg_time s3op_min_time
s3op_max_time
  1 2023-04-11-09:01:00          0         0                0             0
0             0
  2 2023-04-11-09:02:00          1         0                0            14
14            14
  3 2023-04-11-09:03:00          0         0                0             0
0             0
```

  Sample query:

```
# oc -n ibm-spectrum-scale rsh -c gpfs master2 mmperfmon query
s3op_count,s3op_size,s3op_error_count,s3op_avg_time,s3op_min_time,s3op_max_time --filter
"operation=upload_part" -b 60 -n 3
```

  Query output:

```
Legend:
 1:     dasS3.monitoring|DasS3|s3ops|upload_part|s3op_count
 2:     dasS3.monitoring|DasS3|s3ops|upload_part|s3op_size
 3:     dasS3.monitoring|DasS3|s3ops|upload_part|s3op_error_count
 4:     dasS3.monitoring|DasS3|s3ops|upload_part|s3op_avg_time
 5:     dasS3.monitoring|DasS3|s3ops|upload_part|s3op_min_time
 6:     dasS3.monitoring|DasS3|s3ops|upload_part|s3op_max_time

Row           Timestamp s3op_count s3op_size s3op_error_count s3op_avg_time s3op_min_time
s3op_max_time
  1 2023-04-11-09:01:00          0         0                0             0
0             0
  2 2023-04-11-09:02:00          8         0                0             0
0             1
  3 2023-04-11-09:03:00          0         0                0             0
0             0
```

  Sample query:

```
# oc -n ibm-spectrum-scale rsh -c gpfs master2 mmperfmon query
s3op_count,s3op_size,s3op_error_count,s3op_avg_time,s3op_min_time,s3op_max_time --filter
"operation=complete_object_upload" -b 60 -n 3
```

  Query output:

```
Legend:
 1:     dasS3.monitoring|DasS3|s3ops|complete_object_upload|s3op_count
 2:     dasS3.monitoring|DasS3|s3ops|complete_object_upload|s3op_size
 3:     dasS3.monitoring|DasS3|s3ops|complete_object_upload|s3op_error_count
 4:     dasS3.monitoring|DasS3|s3ops|complete_object_upload|s3op_avg_time
 5:     dasS3.monitoring|DasS3|s3ops|complete_object_upload|s3op_min_time
 6:     dasS3.monitoring|DasS3|s3ops|complete_object_upload|s3op_max_time

Row           Timestamp s3op_count s3op_size s3op_error_count s3op_avg_time s3op_min_time
s3op_max_time
  1 2023-04-11-09:01:00          0         0                0             0
```

```
0             0
  2 2023-04-11-09:02:00          1         0                0            92
92            92
  3 2023-04-11-09:03:00          0         0                0             0
0             0
```

Chapter 8. Troubleshooting

Use the following information to review known issues and potential workarounds.

Common issues

The issues that you might encounter due to problems in deployment or configuration are as follows:

- "The mmdas command does not work as expected" on page 115
- "The mmdas command cannot create account or export" on page 116

The mmdas command does not work as expected

The **mmdas** command might fail with the following error message.

```
Something went wrong while processing the request.
```

For example,

```
mmdas account list

Something went wrong while processing the request.
Check 'ibm-spectrum-scale-das-endpoint' pod logs in 'ibm-spectrum-scale-das' namespace for more
details
```

This issue occurs if the `das-gui-user` secret is configured incorrectly in the `ibm-spectrum-scale-das` namespace.

Workaround 1

Verify that the secret is configured by using the credentials of the IBM Storage Scale GUI or REST API user that you created in "Example configuration of IBM Storage Scale DAS" on page 37.

To verify, you can view or edit the `das-gui-user` secret and make sure that the username and password have correct base64 encoded values.

```
oc edit secret das-gui-user -n ibm-spectrum-scale-das
```

A sample output is as follows:

```
apiVersion: v1
data:
  password: UGFzc3cwcmQ=
  username: czMtYWRtaW4=
kind: Secret
metadata:
  creationTimestamp: "2021-12-09T13:28:19Z"
  name: das-gui-user
  namespace: ibm-spectrum-scale-das
  resourceVersion: "19127763"
  uid: 07fdbe45-1cdf-4b74-bd17-9220050a5238
type: Opaque
```

Update the credentials if needed and save this change.

If the issue persists, do the following:

Workaround 2

Make sure that the GUI pods in the `ibm-spectrum-scale` namespace are restarted and that they enter the `Running` state.

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

A sample output is as follows:

```
NAME                                READY   STATUS    RESTARTS   AGE
ibm-spectrum-scale-gui-0            4/4     Running   0          87s
ibm-spectrum-scale-gui-1            4/4     Running   0          2m31s
ibm-spectrum-scale-pmcollector-0    2/2     Running   2          28d
ibm-spectrum-scale-pmcollector-1    2/2     Running   2          28d
worker0                             2/2     Running   2          28d
worker1                             2/2     Running   2          28d
```

Doing this, re-establishes the required roles and role bindings for IBM Storage Scale GUI pods to access the required services and resources in the `ibm-spectrum-scale-das` namespace. Thereafter, the **mmdas** command should work as expected.

The mmdas command cannot create account or export

When you create an account by using the **mmdas** command, and it displays the following error message:

```
mmdas account create s3user5004@fvt.com --uid 5004 --gid 5000 --newBucketsPath /mnt/remote-
sample/s3user5004-dir
"this.begin() must be called before sending queries on this transaction'
```

Workaround

This is due to the `noobaa-db` pod being in the `Init` state and not in `Running` state. Retry the `account/export create` command when the pod moves to the `Running` state.

Known issues

The known issues in IBM Storage Scale DAS 5.1.9.1 release and possible workarounds are as follows:

- “S3 service creation fails with the error "Something went wrong while processing the request."” on page 117
- “I/O gets interrupted if the node running the noobaa-core and noobaa-db pods goes down” on page 117
- “I/O gets interrupted due to IBM Storage Scale container native update” on page 118
- “Unable to create new accounts or exports during noobaa-db pod migration” on page 118
- “mmdas commands might fail with could not open file "global/pg_filenode.map"” on page 118
- “Changing scaleFactor might result in I/O failure” on page 118
- “Account creation fails with the EOF message” on page 119
- “Export creation fails with the INVALID_READ_RESOURCES error” on page 119
- “S3 service instance is in the FAILED state upon its creation” on page 119
- “Account names that contain special characters trigger error” on page 120
- “Slow reader applications might lose S3 access to data” on page 120
- “IBM Storage Scale DAS does not verify MD5 checksums, in case MD5 based Etags are disabled” on page 120
- “IBM Storage Scale DAS does not properly fail-over the IP address” on page 120
- “Performance degrade of S3 applications while connecting to more than one data access node” on page 121
- “Uneven distribution of NooBaa endpoint pods” on page 121
- “When noobaa-core and noobaa-db pod running node is made down” on page 121
- “Warp workload fails occasionally with “The specified key does not exist” error” on page 121
- “S3 service update with some combinational flags is not honored” on page 122
- “mmdas command fails with the error "Something went wrong while processing the request"” on page 123
- “Performance degradation for read of small objects” on page 124
- “IBM Storage Scale DAS 5.1.9.1 pods run into CrashLoopBackOff error or mmdas command fails on fresh install/upgrade of IBM Storage Scale DAS” on page 124

- “ODF 4.13.x upgrade might show the CreateContainerConfigError state of the rook-ceph-operator pod in the openshift-storage namespace” on page 124

S3 service creation fails with the error "Something went wrong while processing the request."

Once the IBM Storage Scale DAS is deployed, when you use the **mmdas** command to create the S3 service, the command might fail.

For example,

```
mmdas service create s3 --acceptLicense --ipRange 192.0.2.13-192.0.2.15
```

```
Something went wrong while processing the request.
Check 'ibm-spectrum-scale-das-endpoint' pod logs in 'ibm-spectrum-scale-das' namespace for more
details
```

Try using the IBM Storage Scale DAS REST API to check if there is an issue with the REST API interface as well:

```
curl -k -u s3-admin -X GET -H "accept: application/json" https://<ibm-spectrumscale_host>/
scalemgmt/v2/das/services
Enter host password for user 's3-admin':

Error 401: SRVE0295E: Error reported: 401
```

If there is an error when you use the IBM Storage Scale DAS REST API as well, check if the IBM Storage Scale GUI REST API is working fine:

```
curl -kv -u 's3-admin' https://<ibm-spectrumscale_host>/scalemgmt/v2/filesystems
Trying x.x.x.x
TCP_NODELAY set
Connected to <ibm-spectrumscale_host> port 443 (#0)
..

Error 401: SRVE0295E: Error reported: 401
```

If using the IBM Storage Scale REST API also results in an error, it indicates that there might be an issue with the user authentication. The user 's3-admin' created for IBM Storage Scale DAS might be deleted or its password might have expired. If that is the case, resolve the issue and then retry.

Otherwise, there might be an issue with the IBM Storage Scale GUI pod.

Workaround

1. Restart the GUI pods in the IBM Storage Scale namespace by issuing the following command:

   ```
   oc delete pod <gui-0> <gui-1>
   ```

2. After the new GUI pods are up and running, check if the REST API interface to access IBM Storage Scale `filesystems` or `das/services` is working fine.

If the REST API is working, the **mmdas** command should also work as expected.

Note: This issue can also occur while running the **`mmdas service list`** command. If you see the error message, apply the same workaround.

I/O gets interrupted if the node running the noobaa-core and noobaa-db pods goes down

If the `noobaa-core` and `noobaa-db` pods are running on the same node and that node goes down, I/O might get interrupted.

Note: Endpoint refers to NooBaa endpoints.

This issue occurs because it takes approximately 6 minutes for the noobaa-db pod to come online. During this time, the noobaa-core pod cannot communicate with the noobaa-db pod, which cause the I/O interruption.

Workaround

Use the **oc get pods** command on the openshift-storage namespace to check the state of the noobaa-db pod. Once the state of the noobaa-db pod changes to Running, I/O resumes.

I/O gets interrupted due to IBM Storage Scale container native update

The IBM Storage Scale container native update reboots each node. Due to the duration of each reboot, this concurrent update can take around 20 to 45 minutes. Administrators should plan for intermittent I/O outage for this duration.

Workaround

This is currently a limitation in IBM Storage Scale DAS.

Unable to create new accounts or exports during noobaa-db pod migration

If the node on which the noobaa-db pod is running is shutdown, new accounts or exports cannot be created for some time.

This issue occurs because it takes approximately 6 minutes for the noobaa-db pod to be migrated to another node. During this time, you cannot create new accounts or exports.

Workaround

Use the **oc get pods** command on the openshift-storage namespace to check the state of the noobaa-db pod. Once the state of the noobaa-db pod changes to Running, you can create new accounts or exports.

mmdas commands might fail with could not open file "global/pg_filenode.map"

Commands such as **mmdas account list** and **mmdas export list** might fail with the following error message:

```
could not open file "global/pg_filenode.map": Permission denied
```

This error occurs when one of the node's interfaces goes down and the NooBaa database pods were running on that node.

Workaround

Start the interface by applying the network policy with the **nmstate** command. For more information, see Updating node network configuration in Red Hat OpenShift Container Platform documentation.

Tip: You can use **oc get nncp** or **oc get nnce** to verify if the network policy is configured.

Changing scaleFactor might result in I/O failure

If you change the scale factor of the S3 service during active I/O, I/O failures might occur.

For example, consider a scenario in which the S3 service was initially created with a scaleFactor of 2. If you reduce the scaleFactor to 1 during active I/O, you might encounter I/O failures.

- These failures occur because when you change the scaleFactor to 1, Kubernetes initiates a cleanup as the number of endpoints need to be reduced.
- This cleanup results in skewed distribution of endpoints between the nodes such that on some nodes the number of endpoints might be high while on other nodes the number of endpoints might reduce to 0. This unbalanced configuration might lead to I/O failures.

Workaround

To avoid this unbalanced configuration, plan and configure the `scaleFactor` at the time of S3 service creation according to your requirements to ensure that the distribution of endpoints does not become skewed.

If you must change the `scaleFactor`, plan it during a maintenance window when there is no active I/O.

Account creation fails with the EOF message

Account creation by using the **mmdas account create** command might fail with the EOF message.

```
mmdas account create s3user1@example.com --gid 9999 --uid 8003 --newBucketsPath /mnt/fs_s3user1/
exmp1

EOF
```

Workaround

Retry creating the account by using the **mmdas account create** command:

```
mmdas account list

No Accounts Available

mmdas account create s3user1@example.com --gid 9999 --uid 8003 --newBucketsPath /mnt/
fs_s3user1/exmp1


Account created successfully, below are the secret and access keys
 Secret Key                                        Access Key
 ----------                                        -----------
 09PSsA/4zxV92X/Da30D7seOzaW4AXn7dps40Azh          w2g9l8NthQDWTIxAIG28

mmdas account list

 Name                   UID     GID     New buckets path
 ----                   ---     ---     ----------------
 s3user1@example.com    8003    9999    /mnt/fs_s3user1/exmp1
```

Export creation fails with the `INVALID_READ_RESOURCES` error

S3 export creation might fail with the following error message:

```
"message": "INVALID_READ_RESOURCES"
```

This error is triggered if the NooBaa namespace store is in the `Rejected` phase. This namespace store is created for the IBM Storage Scale data backend and it is configured with the S3 service.

Workaround

Before you create exports, use the following command to ensure that the NooBaa namespace store is not in the `Rejected` phase.

```
oc get namespacestore -n openshift-storage
```

If the namespace is in the `Rejected` state, the customer should do some checks, such as:

- Basic file system mount check
- Ensure that CNSA and CSI pods are working
- Ensure PVC is bound
- Check the IBM Storage Scale DAS operator logs and make sure that service creation is logged

S3 service instance is in the `FAILED` state upon its creation

The S3 service instance might be in the `FAILED` state after its creation.

Workaround

If the S3 service instance is in the `FAILED` state, refer to the IBM Storage Scale DAS operator logs to determine the cause and then take appropriate action to resolve the issue.

This workaround is applicable for many pods and services.

Account names that contain special characters trigger error

You cannot use special characters in account names except @. For example,

```
user12#
```

Account names that contain special characters are not supported.

Workaround

Do not use special characters in account name.

Slow reader applications might lose S3 access to data

Applications that request IBM Storage Scale DAS to deliver data through read access and consume the delivered data very slowly, might lose S3 access to data. For such workloads, when a slow reader disconnects without draining the requested data first, the endpoint might fail to clean up its internal state. This accumulates and eventually causes all applications to lose S3 access to data. The only known workload which causes this issue is to run COSBench with the `hashCheck=true` option.

Workaround

- To resolve this issue, restart the NooBaa endpoint pods.
- There is no data loss or data corruption.

IBM Storage Scale DAS does not verify MD5 checksums, in case MD5 based Etags are disabled

IBM Storage Scale DAS does not verify MD5 checksums sent by clients using the optional `Content-MD5` header of HTTP requests, in case MD5 based Etags are disabled.

Workaround

Customers who desire that `Content-MD5` headers get validated, must enable the generation of MD5 based Etags by enabling via the S3 service.

IBM Storage Scale DAS does not properly fail-over the IP address

When a Data Access Node loses the high-speed network, then IBM Storage Scale DAS does not properly fail-over the IP address to one of the two other Data Access Nodes.

Workaround

To resolve this issue, shutdown the Red Hat OpenShift node to get all IP addresses moved to the other nodes. Then resolve the network issue and restart the Red Hat OpenShift node.

The IBM Storage Scale file system must have sufficient space while writing S3 objects

When writing S3 objects, ensure that the IBM Storage Scale file system has sufficient space because IBM Storage Scale DAS creates temporary files to process incoming data. For instance, writing a 30 GB object requires up to additional 30 GB temporary space in the file system, until the upload request is completed.

Workaround

This is a prerequisite of IBM Storage Scale DAS for writing S3 objects.

Performance degrade of S3 applications while connecting to more than one data access node

The performance of S3 applications may degrade in case that they connect to more than one IBM Storage Scale data access node and write objects that are stored in the same directory as of the underlying IBM Storage Scale file system.

Workaround

Ensure that such workloads use the same IP address for S3 access, so that this workload is handled from a single data access node.

Uneven distribution of NooBaa endpoint pods

The scaling factor determines the number of NooBaa endpoint pods which run on each data access node. The NooBaa endpoint pods shall be evenly distributed. For instance, with a scaling factor of four, each data access node should run four NooBaa endpoint pods. The decrease of the scaling factor like, reducing the scaling factor from four to three and certain infrastructure issues can lead to an uneven distribution of NooBaa endpoint pods. IBM Storage Scale DAS tries to correct this by terminating imbalanced NooBaa endpoint pods and directing the Kubernetes scheduler where to start new NooBaa endpoint pods. However, this correction is not always successful, at least one `noobaa-endpoint` runs on each DAN node either by scaling up or down.

Workaround

This is currently a limitation in IBM Storage Scale DAS.

When noobaa-core and noobaa-db pod running node is made down

As per the current design, noobaa-db pod would take few minutes (around 6+ minutes) to get into the `Running` state as it is moved to other node. In the interim, there is a possibility of I/O loss, which is expected as the Object Interface is not in healthy state. Once noobaa-db get into the `Running` state and the connection establishes between the two (that is, noobaa-core and noobaa-db) the I/O will be able to continue and new I/O requests will be serviced.

Workaround

This is currently a limitation in IBM Storage Scale DAS.

Warp workload fails occasionally with “The specified key does not exist” error

Warp I/O workload run into an error occasionally with the "The specified key does not exist" message.

Warp version:

```
warp --version
warp version 0.5.5 - 1baadbc
```

Monitor NooBaa endpoint logs to check whether the highlighted error is displayed.

When warp starts failing, the following error is observed in the NooBaa endpoint logs:

```
Sep-26 6:32:07.896 [Endpoint/14] [ERROR] CONSOLE:: RPC._on_request: ERROR srv
object_api.update_endpoint_stats reqid 19524@fcall://fcall(7om8vqvf) connid fcall://
fcall(7om8vqvf) AssertionError [ERR_ASSERTION]: _id must be unique. found 2 rows with
_id=undefined in table bucketstats
Sep-26 6:32:07.897 [Endpoint/14] [ERROR] core.rpc.rpc:: RPC._request: response ERROR srv
object_api.update_endpoint_stats reqid 19524@fcall://fcall(7om8vqvf) connid fcall://
fcall(7om8vqvf) params { namespace_stats: [ { io_stats: { read_count: 2199279, write_count:
929200, read_bytes: 55346668240896, write_bytes: 13374358598656, error_write_bytes: 0,
error_write_count: 0, error_read_bytes: 0, error_read_count: 0 }, namespace_resource_id:
'632d5b3674e74100298682d4' }, [length]: 1 ], bucket_counters: [ { bucket_name: SENSITIVE-
d11ed9bf0f42c55a, content_type: 'application/octet-stream', read_count: 1055154, write_count:
358804 }, { bucket_name: SENSITIVE-40584c364915f5f3, content_type: 'application/octet-stream',
read_count: 1144123, write_count: 374277 }, [length]: 2 ] } took [8.8+0.4=9.2] [RpcError: _id
must be unique. found 2 rows with _id=undefined in table bucketstats] { rpc_code: 'INTERNAL',
rpc_data: { retryable: true } }
Sep-26 6:32:07.897 [Endpoint/14] [ERROR] core.sdk.endpoint_stats_collector:: failed on
```

```
update_endpoint_stats. trigger_send_stats again [RpcError: _id must be unique. found 2 rows
with _id=undefined in table bucketstats] { rpc_code: 'INTERNAL', rpc_data: { retryable: true } }
Sep-26 6:32:37.907 [Endpoint/14] [ERROR] core.util.postgres_client:: updateOneWithClient failed
{ system: 632d5af574e74100298682c0, bucket: 632f441da43595b2582184de, content_type:
'application/octet-stream' } { '$set': { last_write: 1664173957897, last_read: 1664173957897,
system: 632d5af574e74100298682c0, bucket: 632f441da43595b2582184de, content_type: 'application/
octet-stream' }, '$inc': { writes: 358804, reads: 1055154 } } UPDATE bucketstats SET data =
jsonb_set(jsonb_set(jsonb_set(jsonb_set(jsonb_set(jsonb_set(jsonb_set(data,'{content_type}','"ap
plication/octet-
stream"'),'{bucket}','"632f441da43595b2582184de"'),'{system}','"632d5af574e74100298682c0"'),'{la
st_read}','1664173957897'::jsonb),'{last_write}','1664173957897'::jsonb),'{reads}',to_jsonb(COAL
ESCE(Cast(data->>'reads' as numeric),0)+1055154)),'{writes}',to_jsonb(COALESCE(Cast(data-
>>'writes' as numeric),0)+358804)) WHERE (data->>'system'='632d5af574e74100298682c0' and data-
>>'bucket'='632f441da43595b2582184de' and data->>'content_type'='application/octet-stream')
RETURNING _id, data AssertionError [ERR_ASSERTION]: _id must be unique. found 2 rows with
_id=undefined in table bucketstats
```

Workaround

1. Check noobaa-db pod in openshift-storage namespace by using the following commands:

   ```
   oc rsh noobaa-db-pg-0
   psql -U postgres
   \c nbcore
   ```

2. Identify the duplicate record by using the following query:

   ```
   SELECT data->>'bucket' as bucket,
               data->>'system' as system,
               jsonb_agg(jsonb_build_object('_id', _id)) as ids
               FROM bucketstats
               GROUP BY 1,2
               HAVING count(*) > 1;
   ```

   Check the record for which duplicate entries exist shown in the following example:

   ```
   nbcore=# select * from bucketstats where (data->>'system'='632431b4cab31d0029558440' and
   data->>'bucket'='63243a12cab31d0029558478' and data->>'content_type'='application/octet-
   stream');
              _id            |                                              data
   --------------------------
   +-----------------------------------------------------------------------------------------
   ------------------------------------------
   -----------------------------------------------------------------------------------------
   -----------------------------------------
    63243c108d5458000e5c5ea7 | {"_id": "63243c108d5458000e5c5ea7", "reads": 129634826905,
   "bucket": "63243a12cab31d0029558478", "system": "632431b4cab31d0029558
   440", "writes": 43169720959, "last_read": 1663676369913, "last_write": 1663676369913,
   "content_type": "application/octet-stream"}
    63243c10c781ba000e15953d | {"_id": "63243c10c781ba000e15953d", "reads": 129634807954,
   "bucket": "63243a12cab31d0029558478", "system": "632431b4cab31d0029558
   440", "writes": 43169713464, "last_read": 1663676369913, "last_write": 1663676369913,
   "content_type": "application/octet-stream"}
   (2 rows)
   ```

   The example shows two entries for a record, delete one of them as shown in the next step.

3. Delete the duplicate entry by using the following command:

   ```
   nbcore=# delete from bucketstats where (data->>'system'='632431b4cab31d0029558440' and
   data->>'bucket'='63243a12cab31d0029558478' and data->>'content_type'='application/octet-
   stream' and data->>'_id'='63243c108d5458000e5c5ea7');
   DELETE 1
   nbcore=#
   ```

4. Exit the noobaa-db pod shell.

S3 service update with some combinational flags is not honored

When S3 service is updated with the combination of flags enableMD5/disableMD5 and scaleFactor, then the scaleFactor flag is only honored. The enableMD5 flag value remains unchanged.

For example,

```
mmdas service update s3 --enableMD5 --scaleFactor 2
```

Workaround

Update the S3 service with `scaleFactor` and `enableMD5/disableMD5` flags individually one after another.

For example,

```
mmdas service update s3 --enableMD5
mmdas service update s3 --scaleFactor 2
```

mmdas command fails with the error "Something went wrong while processing the request"

After the IBM Storage Scale DAS deployment, when you run any **mmdas** command, the command might fail.

For example:

```
mmdas service list
Something went wrong while processing the request.
Check 'ibm-spectrum-scale-das-endpoint' pod logs in 'ibm-spectrum-scale-das' namespace for more
details
```

Try using the IBM Storage Scale DAS REST API to check if there is an issue with the REST API interface as well:

```
curl -k -u s3-admin -X GET -H "accept: application/json" https://<ibm-spectrumscale_host>/
scalemgmt/v2/das/services
Enter host password for user 's3-admin':
```

Sample output:

```

Error 403: SRVE0295E: Error reported: 403
```

`403` is forbidden `http` return code which refers to the multiple attempts with invalid password and user is locked.

Workaround

1. Remove s3 admin user from GUI pods in the IBM Storage Scale namespace and create new user, as shown in the following example:

   ```
   oc exec -c liberty ibm-spectrum-scale-gui-0 -n ibm-spectrum-scale -- /usr/lpp/
   mmfs/gui/cli/rmuser s3-admin
   EFSSG0021I The user s3-admin has been successfully removed.
   EFSSG1000I The command completed successfully.
   oc exec -c liberty ibm-spectrum-scale-gui-0 -n ibm-spectrum-scale -- /usr/lpp/
   mmfs/gui/cli/lsuser
   EFSSG0100I There are no values to return.
   oc exec -c liberty ibm-spectrum-scale-gui-0 -n ibm-spectrum-scale -- /usr/lpp/
   mmfs/gui/cli/mkuser s3-admin -p Passw0rd -g 'ProtocolAdmin'
   EFSSG0019I The user s3-admin has been successfully created.
   EFSSG1000I The command completed successfully.
   oc exec -c liberty ibm-spectrum-scale-gui-0 -n ibm-spectrum-scale -- /usr/lpp/
   mmfs/gui/cli/lsuser
   Name     Long name Password status Group names   Failed login attempts Disable Password
   Expiry Target Feedback Date
   s3-admin           active          ProtocolAdmin 0                     FALSE
   EFSSG1000I The command completed successfully.
   ```

2. Delete `das-gui-user` secret from IBM Storage Scale DAS namespace, then create new secret, as shown in the following example:

   ```
   oc delete secret das-gui-user
   oc -n ibm-spectrum-scale-das create secret generic das-gui-user --from-
   literal=username='s3-admin' --from-literal=password='Passw0rd'
   ```

Performance degradation for read of small objects

When using Red Hat OpenShift Data Foundation (ODF) 4.13 with IBM Storage Scale DAS 5.1.9, performance degradation may be observed when doing read of small objects (`size ~4k`). This issue is observed because of some changes made for NooBaa in Red Hat OpenShift Data Foundation (ODF) 4.13. A fix for this issue may be provided with newer versions of Red Hat OpenShift Data Foundation (ODF).

IBM Storage Scale DAS 5.1.9.1 pods run into `CrashLoopBackOff` error or `mmdas` command fails on fresh install/upgrade of IBM Storage Scale DAS

After fresh installation of IBM Storage Scale DAS 5.1.9.1, user may notice that the pods in `ibm-spectrum-scale-das` namespace are in `CrashLoopBackOff` error.

In case of upgrade to IBM Storage Scale DAS 5.1.9.1, user may notice one or both of the below issues:

- One or more pods in the `ibm-spectrum-scale-das` namespace are in the `CrashLoopBackOff` error.
- The **`mmdas`** command may hung or returns an error message shown as follows:

```
# mmdas service list
Something went wrong while processing the request.
Check 'ibm-spectrum-scale-das-endpoint' pod logs in 'ibm-spectrum-scale-das' namespace for
more details
```

Workaround

This issue might have been caused by network policy introduced in IBM Storage Scale DAS 5.1.9.1. To workaround this issue, perform the following steps:

1. Apply the latest IBM Storage Scale DAS manifest file from the IBM GitHub repository:

```
# oc apply -f https://raw.githubusercontent.com/IBM/ibm-spectrum-scale-container-native/
v5.1.9.1/generated/das/install.yaml
```

2. Check if there are network policies in the `ibm-spectrum-scale-das` namespace:

```
# oc get networkpolicy -n ibm-spectrum-scale-das
NAME                                      POD-SELECTOR   AGE
ibm-spectrum-scale-das-nwpolicy-egress    <none>         16s
ibm-spectrum-scale-das-nwpolicy-ingress   <none>         16s
```

Delete network policies if they are present:

```
#oc delete networkpolicy -n ibm-spectrum-scale-das ibm-spectrum-scale-das-nwpolicy-
egress ibm-spectrum-scale-das-nwpolicy-ingress
networkpolicy.networking.k8s.io "ibm-spectrum-scale-das-nwpolicy-egress" deleted
networkpolicy.networking.k8s.io "ibm-spectrum-scale-das-nwpolicy-ingress" deleted
```

3. Restart all the pods in the `ibm-spectrum-scale-das` namespace:

```
# oc delete pods --all -n ibm-spectrum-scale-das
```

ODF 4.13.x upgrade might show the CreateContainerConfigError state of the rook-ceph-operator pod in the openshift-storage namespace

After OCP is upgraded to 4.13.x, IBM Storage Scale DAS upgrades ODF from 4.13.x (that is, in 4.13.4), the `rook-ceph-operator` pod state is shown as the `CreateContainerConfigError` state. Red Hat is aware of this issue and has been working on this issue. For more information about this issue, see the issue (https://bugzilla.redhat.com/show_bug.cgi?id=2235571).

```
oc get pods -n openshift-storage |grep

rook-ceph-operator-6777844fcd-zw8b8 0/1 CreateContainerConfigError 0 63s 10.254.16.8
worker2.rkomandu-517-upgrade.cp.fyre.ibm.com
```

```
<none> <none>
```

Workaround

Issue the following command to fix this issue:

```
oc delete cm ocs-operator-config -n openshift-storage
```

The `rook-ceph-operator` pod state changes the `Running` state.

Note:

Check whether the Bugzilla fix is in Red Hat ODF errata for greater than 4.13.4. This issue did not occur when ODF is upgraded from 4.13.4 to 4.14.0 during the IBM Storage Scale DAS upgrade as the phase 2.

Chapter 9. Command reference (mmdas command)

The **mmdas** command manages IBM Storage Scale Data Access Services (DAS) service instances, accounts, and exports.

Synopsis

```
mmdas service create ServiceName --acceptLicense
                                 --ipRange IPAddressRange --scaleFactor ScaleFactor
                                 [ --scaleDataBackend FileSystemMountPoint ]
                                 [ --dbStorageClass CSIFilesetName ]
                                 [ --help ]
```

or

```
mmdas service delete ServiceName [ --help ]
```

or

```
mmdas service list [ ServiceName ] [ --output OutputFormat ] [ --help ]
```

or

```
mmdas service update ServiceName { --enable | --disable } --scaleFactor ScaleFactor
                                 { --enableMD5 | --disableMD5 } { --enableAutoHA | --
disableAutoHA } [ --help ]
```

or

```
mmdas account create AccountName [ --gid GroupID ] [ --newBucketsPath BucketsPath ]
                                 [ --uid UserID ] [ --help ]
```

or

```
mmdas account delete [ AccountName | UserID:GroupID ] [ --help ]
```

or

```
mmdas account list [ AccountName | UserID:GroupID ] [ --output OutputFormat ] [ --help ]
```

or

```
mmdas account update AccountName [ --newBucketsPath BucketsPath ]
                                 [ --resetKeys ][ --help ]
```

or

```
mmdas export create ExportName  [ --filesystemPath FileSystemPath ] [ --help ]
```

or

```
mmdas export delete ExportName [ --help ]
```

or

```
mmdas export list [ ExportName ] [ --output OutputFormat ] [ --help ]
```

Availability

Available on all IBM Storage Scale editions.

Description

Use the **mmdas** command to manage IBM Storage Scale Data Access Services (DAS) cluster. The **mmdas** command communicates with the IBM Storage Scale DAS REST interface for performing the management functions.

Prerequisite: Before you can use the **mmdas** command, you must complete the post deployment steps. For more information, see "Example configuration of IBM Storage Scale DAS" on page 37.

Parameters

service
Manages the IBM Storage Scale DAS instance with one of the following actions.

create
Creates an IBM Storage Scale DAS S3 service instance with the specified parameters.

ServiceName
Specifies the name of the service instance that you want to create. IBM Storage Scale DAS only supports `s3` as service instance name.

--acceptLicense
Accepts the IBM Storage Scale license. If you do not use this option, the license is not accepted.

--dbStorageClass *CSIFilesetName*
Optional. Specifies the name of the storage class to configure a database for the S3 service.

Note: The **dbStorageClass** parameter is optional. The IBM Storage Scale DAS operator selects the storage classes defined on the OCP cluster by using `spectrumscale.csi.ibm.com`, if there is only one such storage class. If there are more than one storage classes defined on the OCP cluster using `spectrumscale.csi.ibm.com` as the provisioner, the DAS operator cannot automatically select one of those to configure the S3 service with. In such a scenario, you need to specify which of those storage classes must be used to configure the S3 service.

--scaleDataBackend *FileSystemMountPoint*
Optional. Specifies the file system mount point that is to be enabled for the S3 service interface.

Note: IBM Storage Scale DAS only supports `scaleDataBackend` with the S3 service.

--ipRange *IPAddressRange*
Specifies the range of IP addresses that is to be used for the MetalLB configuration.

IPAddressRange has the following requirements:

- It must be in the format: `x.x.x.x-x.x.x.x`
- It must be in a sequence. For example, 192.0.2.11-192.0.2.15
- It must match the number of OCP nodes which are labeled for IBM Storage Scale usage; nodes that have the `scale=true` label.

--scaleFactor *ScaleFactor*
Specifies the number of IBM Storage Scale DAS labeled nodes on which the service endpoints can scale to. The default value is 1.

Note: Select a scale factor according to your requirements at the time of creating the service because the scale factor must not be changed during active I/O.

delete
Deletes the specified IBM Storage Scale DAS service instance.

ServiceName
Specifies the service instance that you want to delete. IBM Storage Scale DAS only supports `s3` as service instance name.

list

Displays the information for the specified IBM Storage Scale DAS service instance or all service instances.

ServiceName

Specifies the name of the service instance for which you want to display the information. IBM Storage Scale DAS only supports `s3` as service instance name.

--output *OutputFormat*

Optional. Specifies the format of the output. You can specify `json` to generate output in the JSON format. The default output format is text.

update

Enables or disables the IBM Storage Scale DAS service instance.

ServiceName

Specifies the name of the service instance. IBM Storage Scale DAS only supports `s3` as service instance name.

--disable

Disables the specified service instance.

--enable

Enables the specified service instance.

--disableMD5

Disables md5sum calculation for S3 objects at the S3 service level. The md5sum calculation is disabled by default.

--enableMD5

Enables md5sum calculation for S3 objects at the S3 service level.

--disableAutoHA

Disables automatic IP address failover and failback. Automatic IP address failover and failback is enabled at the time of the creation of the service instance.

--enableAutoHA

Enables automatic IP address failover and failback.

--scaleFactor *ScaleFactor*

Specifies the number of IBM Storage Scale DAS labeled nodes on which the service endpoints can scale to.

Note:

- You must not change `--scaleFactor` during active I/O, otherwise I/O failure might occur. Change the scale factor during a maintenance window when there is no active I/O. For more information, see "Changing scaleFactor might result in I/O failure" on page 118.
- You can set the `--scaleFactor` parameter only if the service is configured with `--ipRange` at the time of service creation.

account

Manages the IBM Storage Scale Data Access Services (DAS) S3 user accounts with one of the following actions:

create

Creates an IBM Storage Scale DAS S3 user account and generates the secret key and the access key for the S3 user account.

AccountName

Specifies the name of the S3 user account that you want to create.

--gid *GroupID*

Specifies the group ID that is associated with the S3 user account that you want to create.

--newBucketsPath *BucketsPath*

Optional. Specifies the file system absolute path, which acts as a base path for S3 buckets created using S3 API by this user.

Note: When you specify this parameter for creating an account, the specified path is not validated. If the specified path is not valid, an error occurs when you try to create an export. Administrators must specify the **newBucketsPath** to enable s3 accounts of end users to create exports using the S3 IO path. If **newBucketsPath** is not specified for an S3 account, by default, the S3 user cannot create new exports and gets the `AccessDenied` error while trying to create an export using the S3 IO path.

--uid *UserID*
Specifies the user ID that is associated with the S3 user account that you want to create.

delete
Deletes the specified IBM Storage Scale DAS S3 user account.

AccountName* | *UserID*:*GroupID
Specifies the account name or the group ID and the user ID of the S3 user account that you want delete.

list
Displays the IBM Storage Scale DAS S3 user account information for the specified account name or the group ID and the user ID or all user accounts.

AccountName* | *UserID*:*GroupID
Specifies the account name or the user ID and the group ID of the S3 user account for which you want to display the information.

Note: The access key and the secret key associated with an S3 user account are only displayed in the output if you specify an account name with this command.

--output *OutputFormat*
Specifies the format of the output. You can specify `json` to generate output in the JSON format. The default output format is text.

update
Updates the specified IBM Storage Scale DAS S3 user account.

AccountName
Specifies the name of the S3 user account that you want to update.

--newBucketsPath *BucketsPath*
Specifies the file system absolute path for creating new buckets for the S3 user account that you want to update.

--resetKeys
Resets the S3 user account access key and secret key.

export
Manages the IBM Storage Scale Data Access Services (DAS) S3 exports with one of the following actions.

create
Creates an IBM Storage Scale DAS S3 export access with the specified parameters.

ExportName
Specifies the name of the S3 export that you want to create. The name of the export must:

- consist of lower case alphanumeric characters, - (dash), or . (period)
- begin and end with an alphanumeric character
- have a length greater than or equal to 3 characters and less than or equal to 63 characters

--filesystemPath *FileSystemPath*
Specifies the absolute path that is to be exported.

delete
Deletes the S3 export associated with the specified IBM Storage Scale DAS export name.

ExportName
Specifies the name of the S3 export that you want to delete.

list
Displays the information for the specified IBM Storage Scale DAS S3 export or lists all IBM Storage Scale DAS S3 exports.

ExportName
Specifies the name of the S3 export for which you want to display the information.

--output *OutputFormat*
Specifies the format of the output. You can specify `json` to generate output in the JSON format. The default output format is text.

Exit status

0
Successful completion.

nonzero
A failure has occurred.

Security

You must have root authority to run the `mmdas` command.

Examples

- **`mmdas service`** examples:
  1. To create an IBM Storage Scale DAS S3 service instance and accept the IBM Storage Scale license with the IP address range and the scale factor specified, issue the following command:

     ```
     mmdas service create s3 --acceptLicense --ipRange "192.0.2.12-192.0.2.14" --scaleFactor 1
     ```

     A sample output is as follows:

     ```
     Create request for Spectrum Scale Data Access Service: 's3' is accepted
     ```

  2. To create an IBM Storage Scale DAS S3 service instance and accept the IBM Storage Scale license while specifying the CSI fileset for the S3 service database and the file system for the data backend for the S3 service, issue the following command:

     ```
     mmdas service create s3 --acceptLicense --ipRange "192.0.2.12-192.0.2.14" --scaleFactor 1
     --dbStorageClass ibm-spectrum-scale-csi-fileset --scaleDataBackend /mnt/fs1
     ```

     A sample output is as follows:

     ```
     Create request for Spectrum Scale Data Access Service: 's3' is accepted
     ```

  3. To list the information of IBM Storage Scale DAS service instances, issue the following command:

     ```
     mmdas service list
     ```

     A sample output is as follows:

     ```
     Name   Enable  Phase
     -----  -----   ------
     s3     true    Ready
     ```

     – The **Enable** column shows whether the S3 service instance is enabled or disabled.

     – The deployment phase of the service instance shown in the **Phase** column can be one of the following values:

       - **Ready:** The service instance is ready to be used for S3 account creation or export creation.
       - **Configuring:** The service instance configuration is in progress.

- **Connecting:** The service instance is trying to establish communication between the S3 endpoints and the S3 database.
- **Failed:** The service instance configuration has failed.

Attention: Once you issue the service creation command, for a brief period of time, the **Phase** column might be empty.

4. To list the detailed information for the IBM Storage Scale DAS S3 service instance, issue the following command:

```
mmdas service list s3
```

A sample output is as follows:

```
Name   AcceptLicense   DbStorageClass                 Enable  EnableMD5
-----  -----           ---------------                ------
s3     true            ibm-spectrum-scale-sample      true    true

ScaleDataBackend       Phase   S3Endpoints
-----                  -----   --------------
[/mnt/remote-sample]   Ready   [https://s3-endpoints.example.com https://192.0.2.12
https://192.0.2.13 https://192.0.2.14]

IpRange                EnableAutoHA    ScaleFactor
-----                  -----           --------------
192.0.2.12-192.0.2.14    true            1
```

5. To update the scale factor for an IBM Storage Scale DAS service instance, issue the following command:

```
mmdas service update s3 --scaleFactor 2
```

A sample output is as follows:

```
Update request for Spectrum Scale Data Access Service: 's3' is accepted
```

6. To enable md5sum calculation for S3 objects at the S3 service level, issue the following command:

```
mmdas service update s3 --enableMD5
```

A sample output is as follows:

```
Update request for Spectrum Scale Data Access Service: 's3' is accepted
```

7. To disable automatic IP address failover and failback, issue the following command:

```
mmdas service update s3 --disableAutoHA
```

A sample output is as follows.

```
Update request for Spectrum Scale Data Access Service: 's3' is accepted
```

8. To delete an IBM Storage Scale DAS service instance, issue the following command:

```
mmdas service delete s3
```

A sample output is as follows:

```
IBM Spectrum Scale DAS service s3 delete request accepted
```

- `mmdas account` examples:

1. To create an IBM Storage Scale DAS S3 user account, issue the following command:

```
mmdas account create s3user --gid 777 --uid 888 --newBucketsPath "mnt/fs1/fset1/
user1_buckets"
```

A sample output is as follows:

```
Account is created successfully. The secret and access keys are as follows.

Secret Key                        Access Key
----------                        -----------
q2F4l5tt8/8mFXt8YOroVrUPx80TW6dlrVYm/zGO    47al0MTOuj98WkgHWmti
```

2. To list the account information for all IBM Storage Scale DAS user accounts, issue the following command:

```
mmdas account list
```

A sample output is as follows:

```
Name       UID    GID    New buckets path
----       ---    ---    -----------------
s3user1    888    777    /mnt/fs1/fset1/user1_buckets/s3user1_buckets

s3user2    679    629    /mnt/fs1/fset1/user1_buckets/s3user2_buckets

s3user3    478    128    /mnt/fs1/fset1/user1_buckets/s3user3_buckets

s3user4    471    127    /mnt/fs1/fset1/user1_buckets/s3user4_buckets

s3user5    431    124    /mnt/fs1/fset1/user1_buckets/s3user5_buckets
```

3. To list the account information for a specified S3 user account in the JSON format, issue the following command:

```
mmdas account list s3user1 -o json
```

A sample output is as follows:


```
{"name":"s3user1","uid":888,"gid":777,"new_buckets_path":"/mnt/fs1/fset1/user1_buckets/
s3user1_buckets",
"access_key":"47al0MTOuj98WkgHWmti","secret_key":"q2F4l5tt8/8mFXt8YOroVrUPx80TW6dlrVYm/
zGO"}
```


4. To delete an IBM Storage Scale DAS S3 user account by specifying the account name, issue the following command:

```
mmdas account delete s3user1
```

A sample output is as follows:

```
Account is successfully deleted
```

Note: Before deleting the S3 user account, you must delete the associated exports.

5. To delete an IBM Storage Scale Data Access Services (DAS) S3 user account by specifying the group ID and user ID, issue the following command:

```
mmdas account delete 888:777
```

A sample output is as follows:

```
Account is successfully deleted
```

Note: Before deleting the S3 user account, you must delete the associated exports.

6. To update the bucket path and reset the access and secret keys for an IBM Storage Scale DAS S3 user account, issue the following command:

```
mmdas account update s3user2 --newBucketsPath "mnt/fs1/fset1/sharedBuckets" --resetKeys
```

A sample output is as follows:

```
Account is successfully updated
```

- **`mmdas export`** examples:

1. To create an IBM Storage Scale DAS S3 export, issue the following command:

   ```
   mmdas export create bucket2 --filesystemPath /mnt/fs1/fset1/bucket1
   ```

   A sample output is as follows:

   ```
   Export is successfully created
   ```

2. To list all IBM Storage Scale DAS S3 exports, issue the following command:

   ```
   mmdas export list
   ```

   A sample output is as follows:

   ```
   Name
   ------
   bucket2
   bucket2user1
   user1bucket1
   ```

3. To list the information of an IBM Storage Scale DAS S3 export, issue the following command:

   ```
   mmdas export list bucket2
   ```

   A sample output is as follows:

   ```
   Name       Filesystem Path
   -----      -----------------

   bucket2   /mnt/fs1/fset1/bucket1
   ```

4. To delete an IBM Storage Scale DAS S3 export, issue the following command:

   ```
   mmdas export delete bucket3
   ```

   A sample output is as follows:

   ```
   Export is successfully deleted
   ```

Location

`/usr/local/bin`

Chapter 10. Programming reference (REST APIs)

IBM Storage Scale Data Access Services (DAS) REST APIs are REST-style APIs that provide interoperability between a client and a server over a network. These APIs allow authenticated users to perform management tasks.

The following list shows the significant features of REST-style APIs:

- REST-style APIs are resource-based.
- REST-style APIs are stateless.
- REST-style APIs are client or server.
- REST-style APIs are cacheable.
- REST-style APIs are a layered system.

A representational state transfer (REST) system is a resource-based service system in which requests are made to the resource's universal resources identifier (URI). These requests start a response from the resource in the JSON format.

The operations that you can perform on the resources or a resource element are directed by the HTTP methods such as GET, POST, PUT, and DELETE and in some cases by the parameters of the HTTPS request. The following list provides the meanings of the basic HTTP methods that are used in the requests:

GET
: Reads a specific resource or a collection of resources and provides the details as the response.

PUT
: Updates a specific resource.

DELETE
: Removes or deletes a specific resource.

POST
: Creates a resource.

API endpoints

IBM Storage Scale Data Access Services (DAS) REST APIs include several API services for managing an IBM Storage Scale S3 object access cluster. It uses the HTTP protocol for sending and retrieving data and JSON formatted responses.

IBM Storage Scale DAS provides the following REST APIs:

- API for managing services
- API for managing accounts
- API for managing exports

The endpoints of each API have a characteristic basic syntax. In the following code blocks, *<ibm-spectrumscale_host>* is the host name or the IP address of the API server.

```
https://<ibm-spectrumscale_host>/scalemgmt/v2/das/<endpoint_ID>
```

Note: The variable *<ibm-spectrumscale_host>* in the request URL must be replaced with the route host. Obtain the route host by using the following command from a node that is configured to work with the Red Hat OpenShift Container Platform (OCP) cluster:

```
oc get route ibm-spectrum-scale-gui -n <IBM Storage Scale namespace> -o json | jq .spec.host
```

For example,

```
oc get route ibm-spectrum-scale-gui -n ibm-spectrum-scale -o json | jq .spec.host
```

A sample output is as follows:

```
"ibm-spectrum-scale-gui-ibm-spectrum-scale.example.com"
```

The supported endpoint IDs are:

- services
- accounts
- exports

To access a specific service, account, or export, use the name of the resource in the URL as follows:

```
https://<ibm-spectrumscale_host>/scalemgmt/v2/das/<endpoint_ID>/<resource_name>
```

For example:

```
curl -k -u "s3-admin:Passw0rd" https://<ibm-spectrumscale_host>/scalemgmt/v2/das/services/s3
```

or

```
curl -k -u "s3-admin:Passw0rd" https://<ibm-spectrumscale_host>/scalemgmt/v2/das/accounts/user1
```

Status codes

Each API request that is sent to the server returns a response that includes an HTTP status code and any requested information.

The following are some of the common HTTP status codes:

200 OK
The endpoint operation was successful.

201 Created
The endpoint operation was successful and resulted in the creation of a resource.

202 Accepted
The request is accepted for processing, but the processing is not yet completed. Asynchronous endpoints return this status code in the response to the original request.

204 No content (DELETE)
The endpoint operation was successful, but no content is returned in the response.

303 [interim response status]
The endpoint operation is in progress. Asynchronous endpoints return this status code in response to a request for status.

The following are some common HTTP status error codes:

400 Bad Request (format error in request data)

401 Unauthorized Request (Wrong credentials)

403 Forbidden

404 Not Found

500 Internal Server Error

503 Service Not Available

REST API authentication process

The REST API services require authentication with a user ID and a password.

You must create an IBM Storage Scale GUI user with the `ProtocolAdmin` role and use those credentials with Basic Auth to authenticate with the IBM Storage Scale REST APIs to access IBM Storage Scale DAS endpoints.

1. Create an IBM Storage Scale GUI or REST API user with the `ProtocolAdmin` role.

   ```
   oc exec -c liberty ibm-spectrum-scale-gui-0 -n ibm-spectrum-scale
   -- /usr/lpp/mmfs/gui/cli/mkuser s3-admin -p Passw0rd -g 'ProtocolAdmin'
   ```

   By default, a user's password is expired after 90 days. If the security policy of your organization permits, you can create a password without expatriation limit by issuing the following command:

   ```
   oc exec -c liberty ibm-spectrum-scale-gui-0 -n ibm-spectrum-scale
   -- /usr/lpp/mmfs/gui/cli/mkuser s3-admin -p Passw0rd -g 'ProtocolAdmin' -e 1
   ```

2. Use these user credentials to access the REST APIs for IBM Storage Scale DAS management.

   ```
   curl -k -u "s3-admin:Passw0rd" https://<ibm-spectrumscale-host>/scalemgmt/v2/das/
   <endpoint_ID>
   ```

DAS/services: POST

Creates an IBM Storage Scale Data Access Services (DAS) instance.

Availability

Available on all IBM Storage Scale editions.

Description

The `POST services` request creates a new IBM Storage Scale DAS service instance with the specified parameters.

Request URL

```
https://<ibm-spectrumscale_host>/scalemgmt/v2/das/services
```

Where

services
: Specifies services as the target of the operation.

Request headers

```
Content-Type: application/json
Authorization: Basic czMtYWRtaW46UGFzc3cwcmQ=
```

Request data

```
{
"name": "Supports only 's3' as name for the s3 service ",
"acceptLicense": "Accept the license for IBM Spectrum Scale DAS.",
"dbStorageClass": "(Optional) Name of the Storage Class to configure database for S3 service.",
"scaleDataBackend": [
"(Optional) Spectrum Scale filesystem mountpoint, which is to be enabled for S3 access."
],
"ipRange":"list of ip address range to use for metalib-config, ex: 10.10.10.13-10.10.10.15",
"scaleFactor":"scaleFactor(n) for service endpoints scaling upto (where n is the number of DAS
```

```
labeled nodes) (default 1)"
}}
```

"name": "Supports only 's3' as name for the s3 service"
The name of the IBM Storage Scale DAS service instance. IBM Storage Scale DAS only supports s3 as service instance name.

"acceptLicense": "Accept the license for IBM Storage Scale DAS."
Specifies whether you accept the IBM Storage Scale DAS license. Specify `true` or `false`.

"dbStorageClass": "(Optional) Name of the Storage Class to configure database for S3 service."
Optional. Specifies the storage class that is used to configure a database for the S3 service.

Note: The **`dbStorageClass`** parameter is optional. The IBM Storage Scale DAS operator selects the storage classes defined on the OCP cluster by using `spectrumscale.csi.ibm.com`, if there is only one such storage class. If there are more than one storage classes defined on the OCP cluster using `spectrumscale.csi.ibm.com` as the provisioner, the DAS operator cannot automatically select one of those to configure the S3 service with. In such a scenario, you need to specify which of those storage classes must be used to configure the S3 service.

"scaleDataBackend": [(Optional) IBM Storage Scale filesystem mountpoint, which is to be enabled for S3 access."]
Optional. Specifies the file system mount point that is to be enabled for S3 service interface.

"ipRange":"list of ip address range to use for metalib-config, for example: 10.10.10.13-10.10.10.15"
Specifies the range of IP addresses that is to be used for the MetalLB configuration.

"scaleFactor":"scaleFactor(n) for service endpoints scaling upto (where n is the number of DAS labeled nodes) (default 1)"
Specifies the number of DAS labeled nodes on which the service endpoints can scale to.

Note:

- Only the name and the `acceptLicense` fields are mandatory.
- The IBM Storage Scale DAS operator discovers the values for `dbStorageClass` and `scaleDataBackend` fields automatically.
- `ipRange` must be set only at the service creation time. You cannot update it with the service update operation (PUT). If you want to set up the S3 service access with more than one IP addresses, set this field to a valid IP address range. Only IPV4 IP address range is supported.

  `ipRange` has the following requirements:

  – It must be in the format: `x.x.x.x-x.x.x.x`
  – It must be in a sequence. For example, 192.0.2.11-192.0.2.15
  – It must match the number of OCP nodes which are labeled for IBM Storage Scale usage; nodes that have the `scale=true` label.

Response data

No response data

Examples

The following example shows how to create an IBM Storage Scale DAS service instance.

1. Submit the request:

```
curl -k -X POST -H "Content-Type: application/json" -H "Authorization: Basic
czMtYWRtaW46UGFzc3cwcmQ=" http://<ibm-spectrumscale_host>/scalemgmt/v2/das/services/
-d '{"name":"s3","enable":true,"acceptLicense":true,
"ipRange":"192.0.2.12-192.0.2.14",
"scaleFactor":"1"}'
```

2. An example response is as follows:

```
{"message":"Create request for Spectrum Scale Data Access Service: 's3' is accepted"}
```

DAS/services: GET

Lists the information for the specified IBM Storage Scale Data Access Services (DAS) instance.

Availability

Available on all IBM Storage Scale editions.

Description

The GET `services` request lists the information for the specified IBM Storage Scale DAS instance.

Request URL

```
https://<ibm-spectrumscale_host>/scalemgmt/v2/das/services/ServiceName
```

or

```
https://<ibm-spectrumscale_host>/scalemgmt/v2/das/services
```

Where

services
Specifies the services as the target of the operation.

ServiceName
The service name for which you want to list the information. IBM Storage Scale DAS only supports s3 as service instance name.

Request headers

```
Content-Type: application/json
Authorization: Basic czMtYWRtaW46UGFzc3cwcmQ=
```

Request data

No request data.

Response data (List services)

```
{
        "name": " Supports only 's3' as name for the s3 service",
        "enable": "s3 service is enabled/disabled:  true or false",
        "phase": "s3 service deployment phase, ex: ready, configuring, failed",
}
```

"name": " Supports only 's3' as name for the s3 service"
The name of the IBM Storage Scale DAS service instance.

"enable": "s3 service is enabled/disabled: true or false"
Specifies whether the S3 service instance is enabled or disabled.

"phase": "s3 service deployment phase, for example: ready, configuring, failed"
The s3 service deployment phase.

Response data (List by service name)

```
{
        "acceptLicense": "Accept License for IBM Spectrum Scale Data Access Services Edition,
true or false",
        "dbStorageClass": " Storage Class to be used to configure PVC for the S3 Service
Database, ex: ibm-spectrum-scale-csi-fileset",
        "enable": "s3 service is enabled/disabled:  true or false",
        "enableAutoHA" "Enables automatic IP address failover and failback",
        "enableMD5": "Enables md5sum calculation for S3 objects at the S3 service level",
        "ipRange": "List of ip address range to use for metalib-config, ex:
10.10.10.13-10.10.10.15",
        "name": " Supports only 's3' as name for the s3 service ",
        "phase": " s3 service deployment phase, ex: ready, configuring, failed ",
        "s3Endpoints": [
            "S3 service Endpoints for Data Access, ex: "https://10.10.10.13", "https://
10.10.10.14", "https://10.10.10.15"],
        "scaleDataBackend": [
            "Name of File system act as data backend for access using the s3 service interface,
ex: /mnt/fs1"
        ],
        "scaleFactor":"scaleFactor(n) for service endpoints scaling upto (where n is the number
of DAS labeled nodes) (default 1)"
}
```

"acceptLicense": "Accept License for IBM Storage Scale Data Access Services (DAS) Edition, true or false"
: Specifies whether you accept the license. Specify `true` or `false`.

"dbStorageClass": "Storage Class to be used to configure PVC for the S3 Service Database, for example: ibm-spectrum-scale-csi-fileset"
: Specifies the storage class that is used to configure a PVC for the S3 service database.

"enable": "s3 service is enabled/disabled: true or false"
: Specifies whether the S3 service instance is enabled or disabled upon creation.

"enableAutoHA" "Enables automatic IP address failover and failback",
: Specifies whether the automatic IP address failover and failback is enabled or disabled.

"enableMD5": "Enables md5sum calculation for S3 objects at the S3 service level"
: Specifies whether the md5sum calculation is enabled or disabled. This parameter is disabled by default.

"ipRange": "List of ip address range to use for metalib-config, for example: 10.10.10.13-10.10.10.15"
: Specifies the range of IP addresses that is to be used for the MetalLB configuration.

"name": " Supports only 's3' as name for the s3 service "
: The name of the IBM Storage Scale DAS service instance. IBM Storage Scale DAS only supports s3 as service name.

"phase": "s3 service deployment phase, for example: ready, configuring, failed"
: The s3 service deployment phase.

"s3Endpoints": ["S3 service Endpoints for Data Access, for example: "https://10.10.10.13", "https://10.10.10.14", "https://10.10.10.15"]
: Specifies the S3 service endpoints for data access.

Note: If the IP address range is configured, the S3 service can be accessed over those IP addresses through https://*IPAddress1*, https://*IPAddress2*, and so on. For example, if the IP address range is set to 192.0.2.10-192.0.2.12, the S3 service can be accessed through `https://192.0.2.10`, `https://192.0.2.11`, and `https://192.0.2.12`.

You can configure a DNS with the S3 application nodes resolving a domain name such as `s3-endpoints.example.com` to these IP addresses. Thereafter, this domain name can be used in the URL to access the data over S3 protocol through `https://s3-endpoints.example.com`.

"scaleDataBackend": ["Name of File system act as data backend for access using the S3 service interface, for example: /mnt/fs1"]
: Specifies the name of the file system that acts as the data backend for access using the S3 service interface.

"scaleFactor": "scaleFactor(n) for service endpoints scaling upto (where n is the number of DAS labeled nodes) (default 1)"
: Specifies the number of IBM Storage Scale DAS labeled nodes on which the service endpoints can scale to.

Examples

The following example shows how to list all services.

1. Submit the request:

```
curl -k -X GET -H "accept: application/json" https://<ibm-spectrumscale_host>/
scalemgmt/v2/das/services
```

2. An example response is as follows:

```
[{"name":"s3","enable":true,"phase":"Ready"}]
```

The following example shows how to list the information for the specified service.

1. Submit the request:

```
curl -k -X GET -H "accept: application/json" https://<ibm-spectrumscale_host>/
scalemgmt/v2/das/services/s3
```

2. An example response is as follows:

```
{
"acceptLicense" : true,
"dbStorageClass" : "ibm-spectrum-scale-sample",
"enable" : true,
"enableAutoHA" : false,
"enableMD5" : false,
"ipRange" : "192.0.2.12-192.0.2.14",
"name" : "s3",
"phase" : "Ready",
"s3Endpoints" : [ "https://192.0.2.12", "https://192.0.2.13", "https://192.0.2.14" ],
"scaleDataBackend" : [ "/mnt/remote-sample" ],
"scaleFactor" : 1
}
```

DAS/services: DELETE

Deletes the specified IBM Storage Scale Data Access Services (DAS) instance.

Availability

Available on all IBM Storage Scale editions.

Description

The DELETE `services` request deletes the user account for the specified IBM Storage Scale DAS user account name or the specified user ID and group ID.

Request URL

```
https://<ibm-spectrumscale_host>/scalemgmt/v2/das/services/ServiceName
```

Where

services
: Specifies services as the target of the operation.

ServiceName
The name of the service instance that you want to delete. IBM Storage Scale DAS only supports s3 as service instance name.

Request headers

```
Content-Type: application/json
Authorization: Basic czMtYWRtaW46UGFzc3cwcmQ=
```

Request data

No request data.

Response data

No response data.

Examples

The following example shows how to delete the user account associated with the specified account name.

1. Submit the request:

```
curl -k -X DELETE -H "Authorization: Basic czMtYWRtaW46UGFzc3cwcmQ=" https://<ibm-
spectrumscale_host>/scalemgmt/v2/das/services/s3
```

2. An example response is as follows:

```
{"message":"IBM Spectrum Scale DAS service s3 delete request is accepted"}
```

DAS/services: PUT

Updates the IBM Storage Scale Data Access Services (DAS) instance.

Availability

Available on all IBM Storage Scale editions.

Description

The PUT `services` request updates an existing IBM Storage Scale DAS S3 user account with the specified parameters.

Request URL

```
https://<ibm-spectrumscale_host>/scalemgmt/v2/das/services
```

Where

services
Specifies services as the target of the operation.

Request headers

```
Content-Type: application/json
Authorization: Basic czMtYWRtaW46UGFzc3cwcmQ=
```

Request data

```
{
        "name": "Supports only 's3' as name for the s3 service",
        "enable": "s3 service is enabled/disabled:  true or false"
        "scaleFactor":"scalefactor(n) for noobaa endpoints scaling upto
        (n* number of HPO labled nodes)"
        "enableMD5":"MD5sum is enabled/disabled: true or false"
        "enableAutoHA":"Automatic IP fail-over/fail-back is enabled/disabled: true or false"
}
```

"name": " Supports only 's3' as name for the s3 service "
: The name of the IBM Storage Scale DAS service instance. In IBM Storage Scale DAS, only s3 is supported.

"enable": "s3 service is enabled/disabled: true or false"
: Specifies whether the S3 service instance is enabled or disabled upon creation.

"scaleFactor":"scalefactor(n) for noobaa endpoints scaling upto(n* number of HPO labled nodes)"
: Specifies the number of IBM Storage Scale DAS labeled nodes on which the service endpoints can scale to.

  Note: The `scaleFactor` parameter can be set only if the service is configured with `ipRange` at the time of creation (POST).

"enableMD5":"MD5sum is enabled/disabled: true or false"
: Enables or disables md5sum calculation for S3 object at S3 service level; true or false.

"enableAutoHA":"Automatic IP fail-over/fail-back is enabled/disabled: true or false"
: Enables or disables automatic IP address failover and failback; true or false

Response data

No response data.

Examples

The following example shows how to update the user account information.

1. Submit the request:

```
curl -k -X PUT -H "Content-Type: application/json" -H "Authorization: Basic
czMtYWRtaW46UGFzc3cwcmQ=" https://<ibm-spectrumscale_host>/scalemgmt/v2/das/services/
-d '{"name":"s3","enableMD5":true,"enableAutoHA":false,"scaleFactor":2}'
```

2. An example response is as follows:

```
{"message":"Update request for Spectrum Scale Data Access Service: 's3' is accepted"}
```

DAS/accounts: POST

Creates an IBM Storage Scale Data Access Services (DAS) S3 user account.

Availability

Available on all IBM Storage Scale editions.

Description

The `POST accounts` request creates a new IBM Storage Scale DAS S3 user account with the specified parameters.

Request URL

```
https://<ibm-spectrumscale_host>/scalemgmt/v2/das/accounts
```

Where

accounts
Specifies accounts as the target of the operation.

Request headers

```
Content-Type: application/json
Authorization: Basic czMtYWRtaW46UGFzc3cwcmQ=
```

Request data

```
{
  "name": "Account name",
  "uid": "UID associated with the user account",
  "gid": "ID associated with the user account",
  "newBucketsPath": "Filesystem absolute path which will be used as base path for creating new
buckets for this account"
}
```

"name": "Account name"
The name of the S3 user account that you want to create.

"uid": "UID associated with the user account"
The user ID of the new S3 user account that you want to create.

"gid": "ID associated with the user account"
The group ID of the new S3 user account that you want to create.

"newBucketsPath": "Filesystem absolute path which will be used as base path for creating new buckets for this account"
The file system absolute path that is used as the base path for creating new buckets for the S3 user account.

Note: When you specify this parameter for creating an account, the specified path is not validated. If the specified path is not valid, an error occurs when you try to create an export. Administrators must specify the **newBucketsPath** to enable s3 accounts of end users to create exports using the S3 IO path. If **newBucketsPath** is not specified for an S3 account, by default, the S3 user cannot create new exports and gets the AccessDenied error while trying to create an export using the S3 IO path.

Response data

```
{
  "access_key": "s3 access key",
  "secret_key": "s3 secret key"
}
```

"access_key": "s3 access key"
The access key for the account that is created.

"secret_key": "s3 secret key"
The secret key for the account that is created.

Examples

The following example shows how to create a new user account.

1. Submit the request:

```
curl -k -X POST -H "Content-Type: application/json" -H "Authorization: Basic
czMtYWRtaW46UGFzc3cwcmQ=" https://<ibm-spectrumscale_host>/scalemgmt/v2/das/accounts/
-d '{"name":"s3user","uid":5001,"gid":500,"newBucketsPath":"/mnt/fs1/fset1/s3user_bucket1"}'
```

2. An example response is as follows:


```
{"access_key":"UTnMjG1MUTMyXug8U6aT","secret_key":"PfaJm8ueu+4NrlgF8HI4Y8HrpZOElVJg8kVb0Fp+"}
```


DAS/accounts: GET

Lists the information for the specified IBM Storage Scale Data Access Services (DAS) user account.

Availability

Available on all IBM Storage Scale editions.

Description

The GET `accounts` request lists the information for the specified IBM Storage Scale DAS user account name or user ID and group ID.

Request URL

```
https://<ibm-spectrumscale_host>/scalemgmt/v2/das/accounts/UserName
```

or

```
https://<ibm-spectrumscale_host>/scalemgmt/v2/das/accounts?uid=UserID&gid=GroupID
```

Where

accounts
Specifies accounts as the target of the operation.

UserName
The account name for which you want to list the information.

uid=*UserID*&gid=*GroupID*
The user ID and the group ID of the account for which you want to list the information.

Request headers

```
Content-Type: application/json
Authorization: Basic czMtYWRtaW46UGFzc3cwcmQ=
```

Request data

No request data.

Response data

```
{
  "name": "Account name",
  "uid": "UID associated with the user account",
  "gid": "GID associated with the user account",
  "access_key": "s3 access key",
  "secret_key": "s3 secret key",
  "newBucketsPath": "Filesystem absolute path which will be used as base path for creating new
buckets for this account"
}
```

"name": "Account name"
The name of the specified S3 user account.

"uid": "UID associated with the user account"
The user ID of the specified S3 user account.

"gid": "ID associated with the user account"
The group ID of the specified S3 user account.

"access_key": "s3 access key"
The access key for the S3 user account.

Note: The access key associated with an S3 user account is only displayed in the output if you specify an account name with this API request.

"secret_key": "s3 secret key"
The secret key for the S3 user account.

Note: The secret key associated with an S3 user account is only displayed in the output if you specify an account name with this API request.

"newBucketsPath": "Filesystem absolute path which will be used as base path for creating new buckets for this account"
The file system absolute path that is used as the base path for creating new buckets for the S3 user account.

Examples

The following example shows how to list all S3 user accounts.

1. Submit the request:

```
curl -k -X GET -H "Authorization: Basic czMtYWRtaW46UGFzc3cwcmQ=" -H "accept: application/
json" https://<ibm-spectrumscale_host>/scalemgmt/v2/das/accounts
```

2. An example response is as follows:

```
[
{
    "gid": 52,
    "name": "s3user1",
    "newBucketsPath": "/mnt/fs1/fset1/s3user1_bucket1",
    "uid": 51
  },
  {
    "gid": 101,
    "name": "s3user2",
    "newBucketsPath": "/mnt/fs1/fset1/s3user2_bucket1",
    "uid": 1003
  },
  {
    "gid": 101,
    "name": "s3user3",
    "newBucketsPath": "/mnt/fs1/fset1/s3user3_bucket1",
    "uid": 1001
  },
  {
    "gid": 101,
    "name": "s3user4",
    "newBucketsPath": "/mnt/fs1/fset1/s3user4_bucket1",
    "uid": 1001
  }
]
```

The following example shows how to list the information for the specified account name.

1. Submit the request:

```
curl -k -X GET -H "Authorization: Basic czMtYWRtaW46UGFzc3cwcmQ=" -H "accept: application/
json" https://<ibm-spectrumscale_host>/scalemgmt/v2/das/accounts/s3user
```

2. An example response is as follows:

```
{"name": "s3user",
"uid": 5001,
"gid": 500,
"newBucketsPath": "/mnt/fs1/fset1/s3user_bucket1",
"access_key": "UTnMjG1MUTMyXug8U6aT",
"secret_key": "PfaJm8ueu+4NrlgF8HI4Y8HrpZOElVJg8kVb0Fp+"}
```

DAS/accounts: DELETE

Deletes the specified IBM Storage Scale Data Access Services (DAS) user account.

Availability

Available on all IBM Storage Scale editions.

Description

The `DELETE accounts` request deletes the user account for the specified IBM Storage Scale DAS user account name or the specified user ID and group ID.

Request URL

```
https://<ibm-spectrumscale_host>/scalemgmt/v2/das/accounts/UserName
```

or

```
https://<ibm-spectrumscale_host>/scalemgmt/v2/das/accounts?uid=UserID&gid=GroupID
```

Where

accounts
: Specifies accounts as the target of the operation.

UserName
: The account name for the account that you want to delete.

uid=*UserID*&gid=*GroupID*
: The user ID and the group ID for the account that you want to delete.

Request headers

```
Content-Type: application/json
Authorization: Basic czMtYWRtaW46UGFzc3cwcmQ=
```

Request data

No request data.

Response data

No response data.

Examples

The following example shows how to delete the user account associated with the specified account name.

1. Submit the request:

```
curl -k -X DELETE -H "Authorization: Basic czMtYWRtaW46UGFzc3cwcmQ=" https://<ibm-
spectrumscale_host>/scalemgmt/v2/das/accounts/s3user
```

DAS/accounts: PUT

Updates an IBM Storage Scale Data Access Services (DAS) S3 user account.

Availability

Available on all IBM Storage Scale editions.

Description

The PUT `accounts` request updates an existing IBM Storage Scale DAS S3 user account with the specified parameters.

Request URL

```
https://<ibm-spectrumscale_host>/scalemgmt/v2/das/accounts
```

Where

accounts
: Specifies accounts as the target of the operation.

Request headers

```
Content-Type: application/json
Authorization: Basic czMtYWRtaW46UGFzc3cwcmQ=
```

Request data

```
{
 "name" : "accountname",
 "newBucketsPath": "Filesystem absolute path which will be used as base path for creating new
buckets for this account",
 "resetKeys": "Reset access key and secret key for the given S3 user account"
}
```

"name": "Account name"
: The name of the account that you want to update.

"newBucketsPath": "Filesystem absolute path which will be used as base path for creating new buckets for this account"
: The file system absolute path that is used as the base path for creating new buckets for the S3 user account.

"resetKeys": "Reset access key and secret key for the given S3 user account"
: Resets the access key and the secret key for the specified S3 user account.

Response data

No response data.

Examples

The following example shows how to update the user account information.

1. Submit the request:

```
curl -k -X PUT -H "Content-Type: application/json" -H "Authorization: Basic
czMtYWRtaW46UGFzc3cwcmQ=" https://<ibm-spectrumscale_host>/scalemgmt/v2/das/accounts/
-d '{"name":"s3user", "newBucketsPath":"/mnt/fs1/fset1/s3user_bucket2","resetKeys":true}'
```

DAS/exports: POST

Creates an IBM Storage Scale Data Access Services (DAS) S3 export.

Availability

Available on all IBM Storage Scale editions.

Description

The POST `exports` request creates a new IBM Storage Scale DAS S3 export with the specified parameters. The specified IBM Storage Scale file system path is exported with the specified export name for S3 access.

Request URL

```
https://<ibm-spectrumscale_host>/scalemgmt/v2/das/exports
```

Where

exports
Specifies exports as the target of the operation.

Request headers

```
Content-Type: application/json
Authorization: Basic czMtYWRtaW46UGFzc3cwcmQ=
```

Request data

```
{
  "name" : "name of export",
  "filesystemPath" : "Filesystem absolute path to be exported"
}
```

"name" : "name of export"
The name of the export that you want to create. The name of the export must meet the following criteria:

- consist of lower case alphanumeric characters, - (dash), or . (period)
- begin and end with an alphanumeric character
- have a length greater than or equal to 3 characters and less than or equal to 63 characters

"filesystemPath" : "Filesystem absolute path to be exported"
The file system absolute path assigned for the S3 export.

Response data

No response data.

Examples

The following example shows how to create a new S3 export.

1. Submit the request:

```
curl -k -X POST -H "Content-Type: application/json" -H "Authorization: Basic
czMtYWRtaW46UGFzc3cwcmQ=" https://<ibm-spectrumscale_host>/scalemgmt/v2/das/exports
-d '{ "name" : "s3project", "filesystemPath": "/mnt/fs1/fset1/s3user_bucket3"}'
```

DAS/exports: GET

Lists the information for the specified IBM Storage Scale Data Access Services (DAS) S3 exports or lists all IBM Storage Scale Data Access Services (DAS) S3 exports.

Availability

Available on all IBM Storage Scale editions.

Description

The GET `exports` request lists the information for the specified IBM Storage Scale DAS S3 export or it lists all S3 exports if the export name is not specified.

Request URL

```
https://<ibm-spectrumscale_host>/scalemgmt/v2/das/exports/ExportName
```

or

```
https://<ibm-spectrumscale_host>/scalemgmt/v2/das/exports
```

Where

exports
: Specifies exports as the target of the operation.

ExportName
: The export name for which you want to list the information.

Request headers

```
Content-Type: application/json
Authorization: Basic czMtYWRtaW46UGFzc3cwcmQ=
```

Request data

No request data.

Response data (List all exports)

```
{
        "name": "name of exports"
    }
```

"name": "name of exports"
: The names of the exports.

Response data (List by export name)

```
{
  "name" : "name of export",
  "filesystemPath" : "Filesystem absolute path to be exported"
}
```

"name" : "name of export"
: The name of the export that you want to create.

"filesystemPath" : "Filesystem absolute path to be exported"
: The file system absolute path assigned for the S3 export.

Examples

The following example shows how to list the exports.

1. Submit the request:

```
curl -k -X GET -H "Authorization: Basic czMtYWRtaW46UGFzc3cwcmQ=" -H "accept: application/
json" http://<ibm-spectrumscale_host>/scalemgmt/v2/das/exports
```

2. An example response is as follows:

```
[
{ "name" : "s3project"},
{ "name" : "s3project1"},
{ "name" : "s3project2"},
{ "name" : "s3project3"}
]
```

The following example shows how to list the information for the specified export name.

1. Submit the request:

```
curl -k -X GET -H "Authorization: Basic czMtYWRtaW46UGFzc3cwcmQ=" -H "accept: application/
json" https://<ibm-spectrumscale_host>/scalemgmt/v2/das/exports/s3project2
```

2. An example response is as follows:

```
{ "name" : "s3project2",
"filesystemPath": "/mnt/fs1/fset1/s3user_bucket4" }
```

DAS/exports: DELETE

Deletes the specified IBM Storage Scale Data Access Services (DAS) S3 export.

Availability

Available on all IBM Storage Scale editions.

Description

The `DELETE exports` request deletes the S3 export associated with the specified IBM Storage Scale DAS export name.

Request URL

```
https://<ibm-spectrumscale_host>/scalemgmt/v2/das/exports/ExportName
```

Where

exports
: Specifies exports as the target of the operation.

ExportName
: The name of the S3 export that you want to delete.

Request headers

```
Content-Type: application/json
Authorization: Basic czMtYWRtaW46UGFzc3cwcmQ=
```

Request data

No request data.

Response data

No response data.

Examples

The following example shows how to delete the export associated with the specified export name.

1. Submit the request:

```
curl -k -X DELETE -H "Authorization: Basic czMtYWRtaW46UGFzc3cwcmQ=" https://<ibm-
spectrumscale_host>/scalemgmt/v2/das/exports/s3project1
```

Accessibility features for IBM Storage Scale

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

Accessibility features

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

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

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

Keyboard navigation

This product uses standard Microsoft Windows navigation keys.

IBM and accessibility

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

Notices

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

COPYRIGHT LICENSE:

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

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

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

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

Trademarks

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

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

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

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

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

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

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

Terms and conditions for product documentation

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

IBM Privacy Policy

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

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

Applicability

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

Personal use

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

Commercial use

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

Rights

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

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

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

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

Glossary

This glossary provides terms and definitions for IBM Storage Scale.

The following cross-references are used in this glossary:

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

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

B

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

C

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

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

Cluster Export Services (CES) nodes
A subset of nodes configured within a cluster to provide a solution for exporting GPFS file systems by using the Network File System (NFS), Server Message Block (SMB), and 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 Storage Scale file system. A clustered watch folder can watch file system activity on a fileset, inode space, or an entire file system. Events are streamed to an external Kafka sink cluster in an easy-to-parse JSON format. For more information, see the *mmwatch command* in the *IBM Storage Scale: Command and Programming Reference Guide*.

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

D

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

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

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

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

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

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

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

E

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

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

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

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

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

F

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

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

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

FEK
: See *file encryption key*.

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

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

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

file clone
A writable snapshot of an individual file.

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

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

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

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

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

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

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

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

G

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

global snapshot
A snapshot of an entire GPFS file system.

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

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

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

I

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

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

independent fileset
: A fileset that has its own inode space.

indirect block
: A block containing pointers to other blocks.

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

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

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

J

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

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

K

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

M

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

MEK
: See *master encryption key*.

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

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

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

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

multi-tailed
: A disk connected to multiple nodes.

N

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

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

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

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

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

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

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

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

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

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

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

P

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

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

pool
: A group of resources with similar characteristics and attributes.

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

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

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

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

Q

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

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

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

R

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

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

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

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

RKM server
: See *remote key management server*.

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

S

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

T

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

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

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

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

twin-tailed
A disk connected to two nodes.

U

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

V

VFS
See *virtual file system*.

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

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

W

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

Index

SC27-9872-10