IBM Storage Scale
5.2.0

Erasure Code Edition Guide

SC28-3493-00

Note

Before using this information and the product it supports, read the information in "Notices" on page 119.

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

- IBM Storage Scale Erasure Code Edition ordered through Passport Advantage® (product number 5737-J34)

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

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

Contents

Tables

About this information

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

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

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

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

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

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

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

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

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

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

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

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

Table 1. IBM Storage Scale library information units

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

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	

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

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

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

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

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 • 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*

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

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

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

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

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

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

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

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

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

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

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

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

Information unit	Type of information	Intended users
IBM Storage Scale: Big Data and Analytics Guide	Cloudera 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)

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 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 xxix describes the typographic conventions used in this information. UNIX file name conventions are used throughout this information.

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

Table 2. Conventions

Convention	Usage
bold	Bold words or characters represent system elements that you must use literally, such as commands, flags, values, and selected menu options. Depending on the context, **bold** typeface sometimes represents path names, directories, or file names.
bold underlined	**bold underlined** keywords are defaults. These take effect if you do not specify a different keyword.
`constant width`	Examples and information that the system displays appear in `constant-width` typeface. Depending on the context, `constant-width` typeface sometimes represents path names, directories, or file names.
italic	*Italic* words or characters represent variable values that you must supply. *Italics* are also used for information unit titles, for the first use of a glossary term, and for general emphasis in text.
<*key*>	Angle brackets (less-than and greater-than) enclose the name of a key on the keyboard. For example, `<Enter>` refers to the key on your terminal or workstation that is labeled with the word *Enter*.

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

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

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

How to send your comments

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

`mhvrcfs@us.ibm.com`

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

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

`scale@us.ibm.com`

Chapter 1. Summary of changes

This topic summarizes changes to the 5.2.0 version of IBM Storage Scale Erasure Code Edition.

- IBM Storage Scale Erasure Code Edition supports up to maximum 256 storage nodes in an IBM Storage Scale cluster.
- IBM Storage Scale Erasure Code Edition supports up to maximum of 700 drives per recovery group whereas an individual declustered array (DA) can only contain up to 512 pdisks.
- IBM Storage Scale Erasure Code Edition installation toolkit supports creation of vdisk sets and file systems with multi declustered arrays (DAs).

Chapter 2. Introduction to IBM Storage Scale Erasure Code Edition

IBM Storage Scale Erasure Code Edition provides IBM Storage Scale RAID as software, allowing customers to create IBM Storage Scale clusters that use scale-out storage on any hardware that meets the minimum hardware requirements.

All of the benefits of IBM Storage Scale and IBM Storage Scale RAID can be realized by using your own commodity hardware.

For example, IBM Storage Scale Erasure Code Edition provides:

- Reed-Solomon highly fault-tolerant declustered Erasure Coding, protecting against individual drive failures and node failures.
- Disk Hospital to identify issues before they become disasters.
- End-to-end checksum to identify and correct errors that are introduced by network and/or media.

IBM Storage Scale Erasure Code Edition uses the same software and most of the same concepts that are used in the Elastic Storage Server (ESS). Elastic Storage Server (ESS) is a solution that consists of two I/O (storage) servers and between one and several JBOD disk enclosures, with each storage device (pdisk) attached to both servers. Elastic Storage Server (ESS) has two recovery groups (RGs). Each RG takes half of each enclosure among all enclosures. Under normal conditions, each I/O server supports one of the two RGs. If either I/O server fails, the remaining I/O server takes over and supports both RGs.

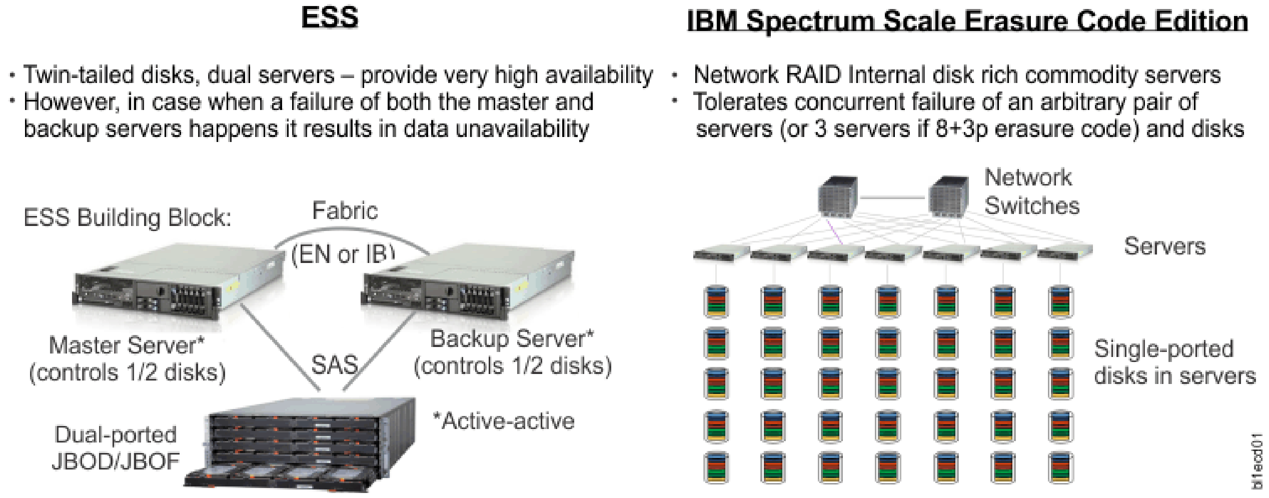

ESS
• Twin-tailed disks, dual servers – provide very high availability
• However, in case when a failure of both the master and backup servers happens it results in data unavailability
ESS Building Block:
Fabric
(EN or IB)
Master Server*
(controls 1/2 disks)
SAS
Backup Server*
(controls 1/2 disks)
Dual-ported JBOD/JBOF
*Active-active
IBM Spectrum Scale Erasure Code Edition
• Network RAID Internal disk rich commodity servers
• Tolerates concurrent failure of an arbitrary pair of servers (or 3 servers if 8+3p erasure code) and disks
Network Switches
Servers
Single-ported disks in servers
bl1ecd01

Figure 1. IBM Storage Scale Erasure Code Edition architecture

IBM Storage Scale Erasure Code Edition can have one or more recovery groups in a cluster and each storage server belongs to only one RG. All of the storage servers in a recovery group must have a matching configuration, including identical CPU, memory, network, and storage device configurations. The storage devices (pdisks) are directly attached to only one storage server. Each storage server typically serves two log groups and each log group manages one half of the virtual disks (vdisk NSDs) assigned to a server. If a storage server fails, the log groups and vdisk NSDs are distributed to the remaining storage servers. Any storage server failure causes the remaining storage servers to serve at most one more log group.

In both Elastic Storage Server (ESS) and IBM Storage Scale Erasure Code Edition, the placement of data is topology aware by using a failure domain hierarchy of rack, node, enclosure, and storage device (pdisk). The RAID code makes placement decisions to maximize fault tolerance, depending on the RAID level you choose. IBM Storage Scale Erasure Code Edition supports the following erasure codes and replication levels: 8+2p, 8+3p, 4+2p, 4+3p, 3WayReplication, and 4WayReplication.

With IBM Storage Scale Erasure Code Edition, it is possible for either IBM Storage Scale Cluster Export Services with protocol software or customer applications to run directly on the storage servers if sufficient hardware resources are available. Customer applications must run in a constrained environment by using Linux cgroups or Docker containers. For protocol workloads with high-performance requirements, the Cluster Export Services must run on separate nodes.

In both Elastic Storage Server (ESS) and IBM Storage Scale Erasure Code Edition, the IBM Storage Scale file system, and file system features are independent of the storage configuration. A file system can be composed of NSDs provided by more than one recovery group, and the recovery groups can be from Elastic Storage Server (ESS) or IBM Storage Scale Erasure Code Edition or a combination of both. All of the IBM Storage Scale file system features can be used in a cluster with IBM Storage Scale Erasure Code Edition storage servers, but there are strict guidelines as to where the various components might run.

For an overview of IBM Storage Scale RAID, see the Introducing IBM Storage Scale RAID topic in the *IBM Storage Scale RAID: Administration*.

Minimum hardware requirements

At a high level, you must have a limited number of storage servers per recovery group, and each server must be an x86 server that runs Red Hat® Enterprise Linux. The storage configuration must be identical for all storage servers. The supported storage types are SAS-attached HDD or SSD drives that use specified LSI adapters, or enterprise-class NVMe drives. Each storage server must have at least one SSD or NVMe drive, this is used for a fast write cache and user data storage. When you set up IBM Storage Scale Erasure Code Edition on an IBM Z® server, the minimum supported OS version is Red Hat Enterprise Linux version 8.1. Only enterprise-class NVMe drives are the supported storage types in this environment. For more information about hardware requirements, see “Minimum hardware requirements and precheck” on page 9.

Maximum storage nodes in a cluster

There can be up to 256 IBM Storage Scale Erasure Code Edition storage nodes in a IBM Storage Scale cluster. For example, 8 RGs with 32 nodes each or 16 RGs with 16 nodes each, or some other combination that results in no more than 256 total storage nodes.

Network configurations

The network can be either Ethernet or InfiniBand for x86_64 architecture, and must be at least 25 Gbps bandwidth, with an average latency of 1.0 msec or less between any two storage nodes. It is recommended to have a determined based on network for storage server traffic. In most cases, the overall storage performance is dictated by network bandwidth and latency. Your performance requirements must be carefully considered when you select the network hardware and the network architecture for your IBM Storage Scale Erasure Code Edition cluster. For more information about networking requirements, see “Network requirements and precheck” on page 15.

Administration and maintenance procedures

IBM Storage Scale Erasure Code Edition administration and maintenance procedures are similar to Elastic Storage Server (ESS), but not identical. With IBM Storage Scale Erasure Code Edition, the customer is responsible for managing the storage server hardware and software. For example, the customer is responsible for updating any firmware and the operating system, including security updates, when needed. Most of the IBM Storage Scale RAID maintenance commands are accomplished by using the **`mmvdisk`** command. For more information about IBM Storage Scale Erasure Code Edition admin and maintenance procedures, see Chapter 12, “Administering IBM Storage Scale Erasure Code Edition,” on page 87.

Health monitoring and problem determination

IBM Storage Scale Erasure Code Edition health monitoring and problem determination procedures rely on IBM Storage Scale **mmhealth** capabilities and IBM Storage Scale RAID troubleshooting guidelines. For more information, see Chapter 13, “Troubleshooting,” on page 109.

Understanding IBM Storage Scale Erasure Code Edition fault tolerance

IBM Storage Scale RAID uses erasure codes, which are selected by the user for protecting data and metadata. The selected erasure code, available disk space, and current disk hardware configuration play a role regarding levels of failures can be survived. IBM Storage Scale RAID has a placement algorithm for distributing strips of the erasure code. The placement algorithm is aware of the hardware groupings of disks, for example, storage nodes that are present in IBM Storage Scale Erasure Code Edition system. It attempts to segregate individual strips of an erasure code stripe across as many groups as possible, which allows survival of larger units of concurrent disk failures.

For example, if IBM Storage Scale Erasure Code Edition hardware configuration includes six storage nodes and a vdisk has been created with 4+2p erasure code, there are 6 strips (4 data and 2 parity) for each block, and each strip of the vdisk's stripe can be placed on a separate node. The vdisk can tolerate two random disk failures without data loss. Furthermore, if two random complete storage nodes (potentially several or tens of disks on each node) are failed, the surviving erasure code strips on other nodes would ensure no data loss. It can also tolerate one random disk failure and one random node failure. The following table shows the various number of storage nodes, erasure codes, number of strips per node, and fault tolerance level for that combination of nodes and erasure code.

Note:

- Only 3 node and 4 node with 3way and 4way replicated RAID codes are demonstrated in the below table since all other node numbers have a similar layout and the same fault tolerance as 4 node with the RAID codes.
- The real vdisk fault tolerance with different node numbers and RAID code may also be limited by Recovery Group Descriptor (RGD) placement other than the RAID code itself. For actual fault tolerance and recommendations, see “Recommendations” on page 18.

Table 3. Erasure Code layout and tolerances for various RAID codes on different number of nodes

Nodes	Code	RAID Layout (strips of a stripe per node)	RAID Tolerance (N Nodes, D Disks)*
3	3WayReplication	1,1,1	2N, N+D, 2D
3	4WayReplication	2,1,1	2N, N+D, 3D
3	4+2p	2,2,2	1N, 2D
3	4+3p	3,2,2	1N, 3D
4	3WayReplication	1,1,1,0	2N, N+D, 2D
4	4WayReplication	1,1,1,1	3N, 2N+D, N+2D, 3D
4	4+2p	2,2,1,1	N, 2D
4	4+3p	2,2,2,1	N+D, 3D
4	8+2p	3,3,2,2	2D
4	8+3p	3,3,3,2	N, 3D
5	4+2p	2,1,1,1,1	N, 2D
5	4+3p	2,2,1,1,1	N+D, 3D

Table 3. Erasure Code layout and tolerances for various RAID codes on different number of nodes (continued)

Nodes	Code	RAID Layout (strips of a stripe per node)	RAID Tolerance (N Nodes, D Disks)*
5	8+2p	2,2,2,2,2	N, 2D
5	8+3p	3,2,2,2,2	N, 3D
6	4+2p	1,1,1,1,1,1	2N, N+D, 2D
6	4+3p	2,1,1,1,1,1	2N, N+D, 3D
6	8+2p	2,2,2,2,1,1	N, 2D
6	8+3p	2,2,2,2,2,1	N+D, 3D
7	4+2p	1,1,1,1,1,1,0	2N, N+D, 2D
7	4+3p	1,1,1,1,1,1,1	3N, 2N+D, N+2D, 3D
7	8+2p	2,2,2,1,1,1,1	N, 2D
7	8+3p	2,2,2,2,1,1,1	N+D, 3D
8	4+2p	1,1,1,1,1,1,0,0	2N, N+D, 2D
8	4+3p	1,1,1,1,1,1,1,0	2N+D, N+2D, 3D
8	8+2p	2,2,1,1,1,1,1,1	N, 2D
8	8+3p	2,2,2,1,1,1,1,1	N+D, 3D
9	4+2p	1,1,1,1,1,1,0,0,0	2N, N+D, 2D
9	4+3p	1,1,1,1,1,1,1,0,0	3N, 2N+D, N+2D, 3D
9	8+2p	2,1,1,1,1,1,1,1,1	N, 2D
9	8+3p	2,2,1,1,1,1,1,1,1	N+D, 3D
10	4+2p	1,1,1,1,1,1,0,0,0,0	2N, N+D, 2D
10	4+3p	1,1,1,1,1,1,1,0,0,0	3N, 2N+D, N+2D, 3D
10	8+2p	1,1,1,1,1,1,1,1,1,1	2N, N+D, 2D
10	8+3p	2,1,1,1,1,1,1,1,1,1	2N, N+D, 3D
11	4+2p	1,1,1,1,1,1,0,0,0,0,0	2N, N+D, 2D
11	4+3p	1,1,1,1,1,1,1,0,0,0,0	3N, 2N+D, N+2D, 3D
11	8+2p	1,1,1,1,1,1,1,1,1,1,0	2N, N+D, 2D
11	8+3p	1,1,1,1,1,1,1,1,1,1,1	3N, 2N+D, N+2D, 3D

IBM Storage Scale Erasure Code Edition discovers the disk hardware groups and their current status automatically and uses this information to rebuild or rebalance the erasure code strips. If the disk hardware configuration changes, for example, if new disks or storage nodes are added to the recovery group, IBM Storage Scale RAID recognizes the change automatically and performs a rebalancing operation in the background. Additionally, a rebuild operation of hardware failure is also cognizant of the hardware groupings. So failed erasure code strips are rebuilt in a manner that is aware of the current disk hardware grouping.

When you plan IBM Storage Scale Erasure Code Edition fault tolerance or maintain IBM Storage Scale Erasure Code Edition system, it is important to understand the hardware failures. The hardware failures might be complete disk failures and latent sector errors. The former is more noticeable while the latter is hidden and easier to be overlooked.

- Complete disk failures can be detected in any I/O to the disk and impact the whole disk. The outage of storage node, storage adapter, or backplane failures, SAS cable problems, or disk internal faults can lead to complete disk failures. It could be a permanent failure, for example, a dead disk that cannot be read or written due to disk internal faults. It might also be a transient failure, for example, the storage node might come back to service soon with the disks. When it happens, the whole disk is taken out for service. All data are rebuilt to restore better fault tolerance if the disk cannot provide service anymore or come back to service within a period.
- Latent sector errors are the ones that go undetected until the corresponding disk sectors are accessed. IBM Storage Scale RAID is designed with comprehensive end-to-end data integrity protection and validation to catch and fix such errors. It provides an automatic scrub process in the background. Each block is read and examined within a period, for example, every 14 days. It is fixed if latent sector errors are detected. There is a low possibility for the latest sector errors to occur before the next scrub.

It is highly recommended the fault tolerance is planned with at least one node and one disk to well prepare for both types of failures. For example, a vdisk has been created with one node and one disk fault tolerance. When a node goes down or in maintenance, there is still one disk fault tolerance for latent sector errors. Otherwise, some block might exceed the fault tolerance when both types of failures happen at the same time. Users must also be aware that they cannot always take down the same number of nodes and disks as the fault tolerance and expect IBM Storage Scale Erasure Code Edition can still function normally. For example, another vdisk has been created with two nodes fault tolerance. It does not mean that users can always take down two nodes safely and unconditionally due to the potential latent sector errors. The vdisk tolerates two nodes (or one node and one disk, or two disks) failures. However, the failures might be either complete disk failures or latent sector errors.

IBM Storage Scale Erasure Code Edition limitations

This topic describes the known limitations of IBM Storage Scale Erasure Code Edition.

General limitations

- The installation toolkit does not support installing mixed Elastic Storage Server (ESS) and IBM Storage Scale Erasure Code Edition in the same cluster. If you need this configuration, see Chapter 10, “Incorporating IBM Storage Scale Erasure Code Edition in an IBM Elastic Storage Server (ESS) cluster,” on page 71 and Chapter 11, “Incorporating IBM Elastic Storage Server (ESS) building block in an IBM Storage Scale Erasure Code Edition cluster,” on page 83. For information about installation toolkit limitations in an IBM Storage Scale Erasure Code Edition environment, see “Installation toolkit-related limitations” on page 28.
- Rack-level fault tolerance is not fully supported. This fault tolerance can be achieved by spreading servers evenly between the racks. However, it is recommended that there must be no more than two storage servers per rack for an N+3P erasure code, and one storage server per rack for N+2P Erasure Code.
- When you use NVMe drives that are hot-swappable, you must create an EDF file to specify the drives and their slot numbers. For more information, see “Mapping NVMe disk slot location” on page 35. However, users are responsible for defining the correct mapping.
- Users must check the physical SAS disk location with IBM Storage Scale Erasure Code Edition commands, and remap the disk slot location when the command output mismatches with the physical locations. For more information, see “Mapping LMR disk location” on page 36. However, users are responsible for checking and setting the correct slot locations.
- Configuration with single server per compute chassis is supported. Configurations with 2 or more servers that are packaged together in the same physical unit are not supported.
- Attaching storage that is not associated with IBM Storage Scale Erasure Code Edition (for example, SAN-attached disks) to recovery group server nodes is not supported.

Configuration limitations

All limitations of the IBM Storage Scale apply, notably:

- All nodes in the RG must be configured the same (memory, drives, CPU, and network).
- Supported erasure codes are 4+2P, 4+3P, 8+2P, 8+3P, 3WayReplication, and 4WayReplication.
- All nodes/HBAs/drives in an RG must have consistent firmware levels, and be at a level that is supported by the hardware provided. For more information, see "Hardware checklist" on page 12.

For information about the hardware requirements, see "Minimum hardware requirements and precheck" on page 9.

Chapter 3. Planning for IBM Storage Scale Erasure Code Edition

This topic describes information on various activities that must be planned for the effective usage of IBM Storage Scale Erasure Code Edition in an enterprise.

IBM Storage Scale Erasure Code Edition hardware requirements

This document describes the requirements for storage hardware, including network requirements that can be used with IBM Storage Scale Erasure Code Edition.

IBM Storage Scale Erasure Code Edition provides the best performance that a hardware platform can provide. Hardware requirements are also defined by the performance requirements of the customer's use case. For example, the minimum network requirement of 25 Gbps might work for some use cases, but 100 Gbps Ethernet or InfiniBand might be needed to achieve performance goals for high-performance workloads.

In IBM Storage Scale Erasure Code Edition, the customer's responsibility is to manage the operating system, firmware, and device driver software on each server. This guide is meant to be a starting point in system sizing and not a substitute for performance engineering and tuning for each customer environment and use case.

Minimum hardware requirements and precheck

This topic describes the minimum requirements for IBM Storage Scale Erasure Code Edition.

These hardware requirements are for the base operating system and the IBM Storage Scale Erasure Code Edition storage functions. Extra resources are needed when you run the IBM Storage Scale protocol software or other workloads on the IBM Storage Scale Erasure Code Edition storage servers, or to achieve specific performance goals.

There is a limit on the number listed in the following table of IBM Storage Scale Erasure Code Edition storage nodes in an IBM Storage Scale Erasure Code Edition recovery group and a cluster. These nodes can be configured as several recovery groups with total storage nodes in a cluster. Every server in a recovery group must have the same configuration regarding CPU, memory, and storage.

Note:

- In a x86_64 environment, only a bare metal server is allowed as IBM Storage Scale Erasure Code Edition storage server.
- Drives with hardware compression enabled are not supported.
- Drives must have unique worldwide name (WWN).
- Drives with volatile cache enabled are not supported. For more information, see "Volatile write cache detection" on page 99.
- SED capable drives are not allowed if they are enrolled, or if they require a key after power-on to use.
- Disk drives in expansion enclosures are not allowed.
- Drives must be hot-swappable and can be replaced independently without having to shut down the storage server.

Note: You can use the `ece_os_readiness` open-source tool to check that your planned IBM Storage Scale Erasure Code Edition servers meet the minimum hardware requirements. This tool is available on the IBM Storage Scale Tools GitHub (https://github.com/IBM/SpectrumScaleTools). Contact IBM for further details.

Table 4. IBM Storage Scale Erasure Code Edition hardware requirements for each x86_64 storage server

Hardware	**Description**
CPU architecture	Single or dual socket Intel or AMD x86_64-bit processor with a total of 16 or more processor cores. AMD CPU requires EPYC Rome and newer generations. **Note:** Dual socket AMD CPU system might need tuning NUMA per socket (NPS®) for better performance.
Memory	64 GB or more for configurations up to 64 drives per node. • For NVMe configurations, you can use all available memory DIMM sockets to get optimal performance. • For server configurations with more than 64 drives per node, contact IBM support for memory requirements.
Server packaging	Single server per enclosure. Multi-node server packaging with common hardware components that provide a single point of failure across servers is not supported currently.
Operating System	See IBM Storage Scale FAQs for details of supported versions.
Drives per storage node	A maximum of 64 drives per storage node is supported.
Drives per recovery group	A maximum of 700 drives per recovery group is supported. However, an individual declustered array (DA) can contain up to 512 pdisks only. At least one declustered array (DA) must contain 12 or more drives and every DA must have four or more drives. **Note:** A DA is a subset of the physical disks within a recovery group that matches in size and speed. A recovery group might contain multiple DAs, which are unique. That is, a pdisk must belong to exactly one DA. The minimum DA size is met by each node that contributes a uniform number of disks.
Nodes per recovery group	A minimum of 3 and maximum of 32 nodes per recovery group is supported.
Storage nodes per cluster	A maximum of 256 IBM Storage Scale Erasure Code Edition storage nodes per cluster is supported.
System drive	A physical drive is needed for each server's system disk. The storage should be RAID1 protected and the capacity should be 100GB or more.

Table 4. IBM Storage Scale Erasure Code Edition hardware requirements for each x86_64 storage server (continued)

Hardware	**Description**
SAS Data Drives	SAS or NL-SAS HDD or SSDs in JBOD mode and these drives are connected to the supported SAS host bus adapters. SATA drives and shingled magnetic recording (SMR) drives are not supported as data drives currently.
NVMe Data Drives	Enterprise class NVMe drives with U.2 form factor and connected to PCIe buses directly or by PCIe switch. NVMe drives that are connected to SAS host bus adapters are not supported as data drives currently.
Fast Drive Requirement	At least one SSD or NVMe drive is needed in each server for IBM Storage Scale Erasure Code Edition logging. The total space of fast drive required on each node is at least 500G.
Network Adapter	Mellanox ConnectX-4, ConnectX-5, ConnectX-6, or ConnectX-7 (Ethernet or InfiniBand).
Network Bandwidth	25 Gbps or more between storage nodes. Higher bandwidth might be needed depending on your workload requirements.
Network Latency	Average latency must be less than 1 msec between any storage nodes.
Network Topology	Use a dedicated storage network to achieve maximum performance for your workload. For other workloads, use a separate network. **Note:** RoCE is supported on lossless network only.

Table 4. IBM Storage Scale Erasure Code Edition hardware requirements for each x86_64 storage server (continued)

Hardware	**Description**
SAS Storage Adapters/Controllers	LSI cards support JBOD mode can be detected and managed by StorCLI utility. You can use the following IBM verified models: • 12 Gb/s RAID Controller models: SAS3008, SAS3108, SAS3408, SAS3508, or SAS3516 • 12 Gb/s Fusion-MPT Host Bus Adapter models: SAS3008, SAS3408, SAS3416, or SAS3616 Dell PowerEdge server with 12Gb/s PowerEdge RAID Controller: PERC H730P Mini, PERC H745 Front SAS, and PERC H755 Front SAS can be detected and managed by PercCLI utility. **Note:** • The StorCLI utility for LSI or PercCLI utility for DELL is a prerequisite for managing these cards. Do not use mixed card types in one IBM Storage Scale Erasure Code Edition recovery group because they might cause performance issues. • The JBOD connection mode is needed for drives that are used for IBM Storage Scale Erasure Code Edition storage.

Hardware checklist

This topic describes the hardware checklists that must be completed before you install IBM Storage Scale Erasure Code Edition at your site.

You can use the `ece_os_readiness` open-source tool to check the defined requirements. This tool is available on IBM Storage Scale Tools GitHub repository (https://github.com/IBM/SpectrumScaleTools).

Disabling volatile write cache on IBM Storage Scale Erasure Code Edition drives

It is necessary that all drives managed by IBM Storage Scale Erasure Code Edition have their volatile write cache that is disabled.

Note: Not disabling the volatile write cache might result in data loss on server failure.

The procedure for disabling the volatile write cache varies between drive types. Contact IBM if you need assistance with the checklist.

- Following is an example of how to disable volatile write cache on a SCSI drive:

  ```
  sdparm --set WCE=0  --save <device>
  ```

- To verify the change:

  ```
  sdparm --get WCE /dev/<device>
  /dev/sda: HGST      HUH721010AL4204   C384
  WCE         0  [cha: y, def:  1, sav:  0]  ----> sav is 0 for it persists across power cycles
  ```

  Note: This example is for SCSI drives only.

- Following is an example of how to query WCE for NVMe devices:

  To show current/default/saved setting (it must be 0 IN ALL three cases for IBM Storage Scale Erasure Code Edition):

  ```
  # nvme get-feature -f 0x6 /dev/nvme0 -n 0 -s 0
  get-feature:0x6 (Volatile Write Cache), Current value:00000000

  # nvme get-feature -f 0x6 /dev/nvme0 -n 0 -s 1
  get-feature:0x6 (Volatile Write Cache), Default value:00000000

  # nvme get-feature -f 0x6 /dev/nvme0 -n 0 -s 2
  get-feature:0x6 (Volatile Write Cache), Saved value:00000000
  ```

  If your NVMe devices have *Volatile Write Cache* enabled, it can be disabled by using the following command:

  ```
  # nvme set-feature -f 0x6 /dev/nvme0 -v 0 -s 0
  set-feature:06 (Volatile Write Cache), value:00000000
  ```

  Every device does not support saving this setting. If you see the following output when this feature is set, you need to disable write cache with a *udev* rule or some other mechanism that is automatically applied following a node restart.

  ```
  # nvme set-feature -f 0x6 /dev/nvme0 -v 0 -s
  NVMe Status:FEATURE_NOT_SAVEABLE(210d)
  ```

  If the command reports the following error message, it means that the NVMe device does not support volatile write cache. Contact the hardware vendor for further details.

  ```
  # nvme get-feature -f 0x6 /dev/nvme0 -n 0 -s 0
  NVMe Status:INVALID_FIELD: A reserved coded value or an unsupported value in a defined
  field(4002)
  ```

  Contact IBM Support if you have questions about this procedure.

Verifying that SAS drives are in JBOD mode

Note: For Dell PERC RAID controller, **perccli** command is a substitute of **storcli** command.

- To verify that the disks are in the JBOD mode, issue the following command:

  ```
  /opt/MegaRAID/storcli/storcli64 /call show
  ```

  The system displays an output similar to the following example:

  ```
  PD LIST :
  =======

  -----------------------------------------------------------------------------------------------------
  EID:Slt DID State DG       Size Intf Med SED PI SeSz Model                                   Sp Type
  -----------------------------------------------------------------------------------------------------
  134:0    23 JBOD  -  446.102 GB SATA SSD N   N  512B MTFDDAK480TCC-1AR1ZA 01GT749D7A09326LEN U  -
  134:1    19 JBOD  -  446.102 GB SATA SSD N   N  512B MTFDDAK480TCC-1AR1ZA 01GT749D7A09326LEN U  -
  134:2    21 JBOD  -  446.102 GB SATA SSD N   N  512B MTFDDAK480TCC-1AR1ZA 01GT749D7A09326LEN U  -
  134:3    22 JBOD  -  446.102 GB SATA SSD N   N  512B MTFDDAK480TCC-1AR1ZA 01GT749D7A09326LEN U  -
  134:4    20 Onln  0  557.861 GB SAS  HDD N   N  512B ST600MM0009                             U  -
  134:5    17 JBOD  -  557.861 GB SAS  HDD N   N  512B ST600MM0009                             U  -
  134:6    18 JBOD  -  557.861 GB SAS  HDD N   N  512B ST600MM0009                             U  -
  134:7    16 JBOD  -  557.861 GB SAS  HDD N   N  512B ST600MM0009                             U  -
  -----------------------------------------------------------------------------------------------------
  ```

IBM Storage Scale Erasure Code Edition required NVMe drive format

Note: Ensure that NVMe drives used for IBM Storage Scale Erasure Code Edition are newly formatted. NVMe drives that are populated with data or metadata might degrade performance.

NVMe drives used by IBM Storage Scale Erasure Code Edition must be formatted with metadata size of zero, and protection information disabled. All NVMe drives in the same declustered array must be formatted with the same Logical block address (LBA) size.

To see the format that is in use for NVMe drives, use the **`nvme list`** command. In this example, `nvme0n1` is formatted with 4-KiB logical block size and 0-byte metadata, while `nvme1n1` is formatted with 8-bytes metadata size.

```
# nvme list
 Node             SN                   Model               Namespace   Usage               Format         FW Rev
 -------------- ------------------- ---------------------  ----------  ------------------  -------------  -------------
 /dev/nvme0n1   CVFT7155000D1P6NGN  INTEL SSDPEDMD016T4L    1           1.60 TB / 1.60 TB   4 KiB + 0 B   8DV1LP13
 /dev/nvme1n1   CVFT715500171P6NGN  INTEL SSDPEDMD016T4L    1           1.60 TB / 1.60 TB   4 KiB + 8 B   8DV1LP13
```

To see the available formats for an NVMe drive (and all drives of that particular type), use the `nvme id-ns` command that specifies the drive path.

```
# nvme id-ns /dev/nvme1n1
 NVME Identify Namespace 1:
 nsze    : 0x1749a956
 ncap    : 0x1749a956
 nuse    : 0x1749a956
 nsfeat  : 0
 nlbaf   : 6
 flbas   : 0x14
 mc      : 0x1
 dpc     : 0x11
 dps     : 0
 nmic    : 0
 rescap  : 0
 fpi     : 0
 dlfeat  : 0
 nawun   : 0
 nawupf  : 0
 nacwu   : 0
 nabsn   : 0
 nabo    : 0
 nabspf  : 0
 noiob   : 0
 nvmcap  : 0
 nvmsetid: 0
 endgid  : 0
 nguid   : 00000000000000000000000000000000
 eui64   : 0000000000000000
 lbaf 0  : ms:0 lbads:9 rp:0x2
 lbaf 1  : ms:8 lbads:9 rp:0x2
 lbaf 2  : ms:16 lbads:9 rp:0x2
 lbaf 3  : ms:0 lbads:12 rp:0
 lbaf 4  : ms:8 lbads:12 rp:0 (in use)
 lbaf 5  : ms:64 lbads:12 rp:0
 lbaf 6  : ms:128 lbads:12 rp:0
```

The entries at the last of the output indicate the available LBA formats (LBAF 0 - 6 in this example). For IBM Storage Scale Erasure Code Edition use a format with metadata size of zero (ms:0). Use a format with relative performance of 0 (rp:0) for best performance.

This example shows the `nvme0n1` is formatted with a metadata size of 8, so it needs to be reformatted for use with IBM Storage Scale Erasure Code Edition. LBA format 3 has zero metadata size, and has *rp* of zero. Use the following command to format the NVMe drive with this format.

```
# nvme format /dev/nvme1n1 --lbaf=3
         Success formatting namespace:1
```

Now, all the NVMe drives have metadata size of zero.

```
# nvme list
 Node              SN                    Model                   Namespace  Usage               Format         FW Rev
 ---------------- -------------------- ------------------------- --------- ------------------- --------------- --------
 /dev/nvme0n1      CVFT7155000D1P6NGN    INTEL SSDPEDMD016T4L     1         1.60 TB / 1.60 TB   4 KiB + 0 B    8DV1LP13
 /dev/nvme1n1      CVFT715500171P6NGN    INTEL SSDPEDMD016T4L     1         1.60 TB / 1.60 TB   4 KiB + 0 B    8DV1LP13
```

Note: For all SCSI and NVMe drives that support volatile write cache, *udev* rules must be created that disable volatile cache for these drives. These rules simplify disk replacement by ensuring that the write cache is disabled automatically before you add them into the recovery group. It also ensures that drives are persistently in the correct state across storage node restarts.

Selecting physical disks for TRIM

You must choose the physical disks with the appropriate alignment and TRIM granularity. To understand the disk capabilities, run the following command: **lsblk --discard**.

You must ensure the following factors when you run this command.

- The alignment (DISC-ALN) is either 0 or less than or equal to the logical block size of the device.
- The discard granularity (DISC-GRAN) is less than or equal to the logical block size of the device.

A sample command is shown:

```
[root@node01 ~]# lsblk --discard /dev/nvme0n1
NAME        DISC-ALN DISC-GRAN DISC-MAX DISC-ZERO
nvme0n1          512      512B       2T         0
```

Note:

Before TRIM is enabled in production, some requirements must be met. See Support for TRIM procedures to know about the requirements.

Operating system and drive firmware levels

All servers must have the same level of operating system software that is installed, and must have the same levels of drive and adapter firmware. Some of these servers can be verified by using **mmlsfirmware** command after your system is configured, but some of the servers are left to the customers to manage.

Using a VMware virtual machine as a storage node

Note:

- IBM Storage Scale Erasure Code Edition supports a constrained VMWare virtual machine. For more information, see Deploying IBM Storage Scale Erasure Code Edition on a VMware virtual machine.
- Use any other virtual machine configurations as the storage node only for testing purposes.

To use a virtual machine as a storage node for testing purpose, check the following conditions:

- Disk drives must be presented as SCSI pass-through devices in a virtual machine.
- Each drive that is used in Recovery Group must be assigned with a unique WWID in the cluster. You can check this WWID by using the **ls -l /dev/disk/by-id or lsscsi -i** command on the virtual machine.
- Run the hardware precheck tool to verify the virtual machine configuration. For systems planned to be used for test and evaluation, you can ignore error messages that are related to the virtualized configuration.
- The memory needed for a virtual machine to serve an IBM Storage Scale Erasure Code Edition recovery group depends on the total disk number and capacity. The pagepool setting requires at least no less than 10 GB.

Network requirements and precheck

This topic describes the networking requirements that must be met before you use IBM Storage Scale Erasure Code Edition.

In IBM Storage Scale Erasure Code Edition configuration, network bandwidth is used by the client workload and the backend erasure code traffic between nodes. For read I/O, every 1.0 Gbps of usable bandwidth requires 2.0 Gbps of total bandwidth. For write, the overhead depends on the selected erasure code. When you write with 8+3P, each 1.0 Gbps of usable bandwidth requires 2.4 Gbps of total bandwidth. This factor is 2.25 for 8+2P, 2.5 for 4+2P, and 2.75 for 4+3P.

Other network considerations and requirements are as follows:

- Linux bonding is supported on mode 1 (active-backup) for Ethernet and Remote Direct Memory Access (RDMA) and mode 4 (IEEE 802.3ad) on Ethernet only. For mode 4, any xmit_hash_policy is supported. However, use layer 3+4.
- Jumbo frames of 9000 MTU (on Ethernet) or higher (on RDMA) can be used.
- When you use Cluster Export Services (CES) protocol software with IBM Storage Scale Erasure Code Edition, a dedicated network for CES protocol traffic is needed.

Network Key Performance Indicators are listed as follows:

- The average Internet Control Message Protocol (ICMP) latency between any two storage nodes must be 1 msec or less.
- The maximum ICMP latency between any two storage nodes must be 2 msec or less.
- The standard deviation must be 0.333 msec or less on the ICMP latency measurements.
- The minimum throughput test of 2000 MB/sec with one client and all the other nodes as server for read test. Note that this is a specific test and not a performance estimator.
- The difference between the maximum and minimum throughput values cannot be more than 20%.
- The ICMP latency metrics must be collected over an extended period, at least 500 seconds for each measurement.
- The throughput metrics must be collected over an extended period, at least 1200 seconds for each measurement.

Note: You can use the `ece_network_readiness` open-source tool to check the defined KPI. This tool is available on the IBM Storage Scale Tools GitHub (https://github.com/IBM/SpectrumScaleTools). Contact IBM for further details.

Disk requirements and precheck

This topic describes the disks performance requirements that must be met before you use IBM Storage Scale Erasure Code Edition.

In the IBM Storage Scale configuration, the network performance and the disk performance are the key factors for file system performance. I/O requests from file system require disks to provide service in parallel according to the erasure code applied in the declustered array of the recovery group. For example, a full block that is read with 8+3p erasure code would ask eight disks to provide data in the declustered array, a full block write with 8+3p erasure code writes data to 11 disks in the declustered array, and all data stripes are scattered across all disks between nodes in the declustered array.

IBM Storage Scale supports NVMe, SSD, and HDD type disks, and all disks in one declustered array must have the same type to provide the same rate of throughput.

Disk Key Performance Indicators are listed as follows (based on the random read performance of 128 K raw device logical block size):

- The minimum IOPS performance on NVMe drive must be greater than 10000, and the average IOPS performance on NVMe drive must be greater than 15000.
- The max latency of NVMe drive must be less than 20 msec, and the average latency of NVMe drive must be less than 1.5 msec.
- The IOPS performance on SSD drive must be greater than 800, and the average IOPS performance on SSD drive must be greater than 1200.
- The max latency of SSD drive must be less than 100 msec, and the average latency of SSD drive must be less than 20 msec.
- The IOPS performance on HDD drive must be greater than 55, and the average IOPS performance on HDD drive must be greater than 110.
- The max latency of HDD drive must be less than 1500 msec, and the average latency of HDD drive must be less than 150 msec.

- The performance on drives of same type must not have more than 10% difference.

Note: Do not test write performance if you have valuable data on the disk. It overwrites the existing data on the disk.

You can use the `ece_storage_readiness` open-source tool to check the defined KPI. This tool is available on the IBM Storage Scale Tools GitHub (https://github.com/IBM/SpectrumScaleTools). Contact IBM for further details.

Run the `ece_os_readiness` tool to ensure that the system has met the requirements before the disk performance testing. `ece_storage_readiness` tool tests read performance by default. Be careful to use the `-w` or `--random-write` option, which test writes performance. This option overwrites the existing data on the test drives.

Planning for erasure code selection

This topic describes the various erasure codes and the factors that need to be considered while you select an erasure code.

Minimizing the risk of data loss due to multiple failures and minimizing disk rebuilds can be done by using 4+3P or 8+3P encoding, at the expense of extra storage overhead.

IBM Storage Scale Erasure Code Edition supports four different erasure codes: 4+2P, 4+3P, 8+2P, and 8+3P in addition to 3WayRepliction and 4WayReplication. Choosing an erasure code involves considering several factors. Examine some of them as follows.

Data protection and storage utilization

Minimizing the risk of data loss due to multiple failures and minimizing disk rebuilds can be done by using 4+3P or 8+3P encoding, at the expense of additional storage overhead. The following table shows the approximate percentage of total capacity that is usable by the file system, excluding user-configurable spare space and IBM Storage Scale RAID metadata. Contact IBM Support if you require any more exact estimate of usable space for your selected configuration.

Table 5. Capacity usable by file system

Protection Type	Usable capacity
4WayReplication	25%
3WayReplication	33%
4+3P	57%
4+2P	67%
8+3P	73%
8+2P	80%

RAID rebuild and Spare space

IBM Storage Scale RAID performs intelligent rebuilds based on the number of failures to a vdisk. For example, with 8+2P protection, if one failure occurs, IBM Storage Scale RAID begins rebuilding the missing data or parity strip that was lost on the failed disk or node. Since data is still protected, this rebuild process occurs in the background and has little effect on the file system performance. If a second failure occurs, IBM Storage Scale RAID recognizes that another failure would result in data loss. It then begins a critical rebuild to restore data protection. This critical rebuild phase results in performance degradation until at least one level of protection can be restored.

The number of reserved spare disk spaces in Declustered Array also impacts fault tolerance in disk failure. For example, in a Recovery Group with 10 nodes, the vdisks with 8+2p erasure code and reserved 5 spare disk space in the Declustered Array. Each node holds only one strip for each data stripe. With

2 parity strips redundancy, the vdisks have 2 node fault tolerance. When one disk failure happens in the Declustered Array, IBM Storage Scale Erasure Code Edition uses spare space to rebuild the data. For 5 spare disk space evenly distributed on 10 nodes, each node has equivalent 1/2 disk space that is reserved for rebuilding. IBM Storage Scale Erasure Code Edition rebuild process will try to keep the best fault tolerance, so data will be rebuilt to the node which has the failed disk first to keep 2 nodes fault tolerance. But given the spare space on the node isn't enough to keep all data of the failed disk, other nodes' spare space will be used to rebuild data which means the fault tolerance will degrade to 1 node after rebuilding the data. IBM Storage Scale Erasure Code Edition still has one node fault tolerance even if one node has multiple (less than 5) disk failures since the rebuild procedure distributes some of the data among the other 9 nodes' spare space and keeps each node with a maximum of 2 strips from each data stripe. The disadvantage is that the Recovery Group will have no data redundancy if one additional node failure happens in consequence. Therefore, it is recommended to plan a Recovery Group with node number and spare disk space in considering for the best failure tolerance in a healthy state and keeping Declustered Array with more than 1 node + 1 pdisk failure tolerance in the disk failure case. For this case, 11 or more nodes with 8+2p is recommended.

To protect data from disk failure, see the NOTE in "Recommendations" on page 18.

Nodes in a recovery group

The number of nodes in a recovery group can also impact erasure code selection. If you consider a 4-node recovery group with 4+2P protection, each node contains 1 piece of data. In addition, for each stripe, 2 nodes contain 1 piece of parity data. A failure of a node that contains both parity and data results in a double failure for that stripe of data, which causes that stripe to be critical and results in performance degradation during the critical rebuild phase. However, in a 6-node recovery group, with the same 4+2P protection, a single node failure only results in 1 failure to the RAID array. Considering disk failures and spare disk space for rebuild, a 7 or more-node recovery group is recommended for 4+2P erasure code, it can keep fault tolerance after rebuilding the data to spare space in case of disk failure. For more information, see "RAID rebuild and Spare space" on page 17.

Recommendations

This topic describes recommendations on what block sizes to be used with each erasure code and how many node failures can be tolerated based on the recovery group size.

The following table shows how many node and disk failures can be tolerated with different RAID protections and node numbers.

Note:

To protect data from disk failure, all failure tolerances that are marked with **#** in the following table need to be paid attention to for the spare disk space other than the erasure code. You can change the number of spare disk space to the same or bigger number than the node number before creating vdisks. For example, for 6 nodes with 4+2p erasure code, you can change all DA's spare disk space to 6 before creating vdisks. (This can only hold one disk failure without downgrading the fault tolerance below 1 node + 1 pdisk after rebuilding and you need to replace the disks as quickly as possible if more than one disk fails.)

Use the recommended Erasure code with the nodes number without **#** marked if you do not have enough spare disk space.

You must replace any failed disk as quickly as possible if you cannot meet the requirements. For more information, see "RAID rebuild and Spare space" on page 17.

Note: All failure tolerances that are marked with ***** are limited by recovery group descriptors rather than by the RAID code.

Table 6. Node and disk failures that can be tolerated based on RAID codes and node numbers

Number of nodes	3WayReplication	4WayReplication	4+2P	4+3P	8+2P	8+3P
3	1 Node + 1 Device *	1 Node + 1 Device *	Not recommended 1 Node	Not recommended 1 Node	Not recommended 2 Devices	Not recommended 3 Devices
4	1 Node + 1 Device *	1 Node + 1 Device *	Not recommended 1 Node	1 Node + 1 Device #	Not recommended 2 Devices	Not recommended 1 Node
5	2 Nodes	2 Nodes *	Not recommended 1 Node	1 Node + 1 Device	Not recommended 1 Node	Not recommended 1 Node
6	2 Nodes	2 Nodes *	2 Nodes #	2 Nodes	Not Recommended 1 Node	1 Node + 1 Device #
7	2 Nodes	2 Nodes *	2 Nodes	2 Nodes*	Not Recommended 1 Node	1 Node + 1 Device
8	2 Nodes	2 Nodes *	2 Nodes	2 Nodes*	Not Recommended 1 Node	1 Node + 1 Device
9	2 Nodes	3 Nodes	2 Nodes	3 Nodes	Not Recommended 1 Node	1 Node + 1 Device
10	2 Nodes	3 Nodes	2 Nodes	3 Nodes	2 Nodes #	2 Nodes
11+	2 Nodes	3 Nodes	2 Nodes	3 Nodes	2 Nodes	3 Nodes

There are limits on what block sizes can be used with each RAID Code. The following table provides information about the limits:

Table 7. Limits on block sizes to be used with RAID Code

Block size	3WayReplication	4WayReplication	4+2P	4+3P	8+2P	8+3P
256 KiB	Supported	Supported	Not supported	Not supported	Not supported	Not supported
512 KiB	Supported	Supported	Supported	Supported	Supported	Supported
1 MiB	Supported	Supported	Supported	Supported	Supported	Supported

Table 7. Limits on block sizes to be used with RAID Code (continued)

Block size	3WayReplication	4WayReplication	4+2P	4+3P	8+2P	8+3P
2 MiB	Supported	Supported	Supported	Supported	Supported	Supported
4 MiB	Not supported	Not supported	Supported	Supported	Supported	Supported
8 MiB	Not supported	Not supported	Supported	Supported	Supported	Supported
16 MiB	Not supported	Not supported	Not supported	Not supported	Supported	Supported

The following considerations are required for choosing block sizes to be used depending on the device media type:

- SSD (NVMe or SAS) drives are better to be set with smaller block size for small I/O workloads.
- HDD drives are better to be set with larger block size for large I/O workloads.

Even though the number of failures that can be tolerated in a smaller recovery group is the same as the number of failures in a larger recovery group, the amount of critical data that must be rebuilt for each failure is less for a larger recovery group. For example, with an 8+3P array on an 11-node recovery group, three-node failures would impact all of the data in the file system. On a 30-node recovery group, three node failures would impact only about 10% of the data on the file system (assuming all disks are the same size). The critical rebuild would complete more quickly because the rebuild work is distributed across a larger number of remaining nodes.

When you plan the erasure code type, also consider future expansion of the cluster and storage utilization. Erasure codes for vdisks cannot be changed after the vdisk is created, and larger stripe widths have better storage utilization. A 4+3P code uses 57% of total capacity for usable data, while an 8+3P code uses 73% of the total capacity for usable data. So, rather than creating a 9-node cluster with 4+3P and expanding it in the future, a 12 or more-node cluster by using 8+3P might be more cost-effective. In some cases, using a non-recommended erasure code might be tolerable if there are plans to increase the cluster size.

Planning for node roles

When you configure an IBM Storage Scale Erasure Code Edition system, it is important to account both for workload and roles of various nodes.

Each cluster requires manager nodes and quorum nodes. Each recovery group requires a recovery group master. The IBM Storage Scale installation toolkit helps to configure the quorum and the manager node roles.

In addition, more IBM Storage Scale features require more node types:

- CES services require CES nodes, which can also be part of an IBM Storage Scale Erasure Code Edition recovery group.
- AFM gateway nodes, which cannot be a part of a recovery group.
- Transparent cloud tiering (TCT) nodes, which cannot be a part of a recovery group.
- GUI nodes, which cannot be a part of a recovery group.
- TSM backup nodes, which cannot be a part of a recovery group.
- Other (non- IBM Storage Scale Erasure Code Edition) storage types, which cannot be a part of a recovery group.

Before you install IBM Storage Scale Erasure Code Edition, a basic network test must be passed. A freely available open-sourced tool is provided with no warranty and official support from IBM to help you achieve running the test. Any network that does not run or pass the test must be considered as not suited to install IBM Storage Scale Erasure Code Edition. For more information, see "Network requirements and precheck" on page 15.

When you plan a system, it is best to determine the minimum requirements for IBM Storage Scale RAID to get the performance and capacity needed. Then, add additional hardware as needed to meet your functional requirements with hardware for the various node roles and applications.

As nodes take on more roles, the performance of applications that run on that node might be affected by the operations of those roles. File system and CPU-intensive tasks might run slower on a node that is running as a recovery group master and file system manager than on other nodes in the cluster. There are two strategies to consider when you distribute node roles and workload across a cluster:

- A small subset of these nodes might be used to act in several of these roles. For example, you might choose three nodes to act as file system managers, recovery group masters, and quorum. Other cluster applications can then avoid these three nodes entirely determining when to run, as these nodes might be more heavily used.
- Distribute the roles of file system managers and recovery group masters to different nodes across the cluster. In this way, you can use any node in the cluster to run applications, with the expectation that they can be slightly impacted.

The installation toolkit assists with node role selection and configuration during system installation.

Recovery group master

When a recovery group is defined in IBM Storage Scale RAID, a server is chosen to be the recovery group master. The node performing this role is automatically chosen by the system. The RG master can be used for other tasks in the cluster.

Quorum nodes

IBM Storage Scale uses a cluster mechanism that is called quorum to maintain data consistency when a node fails.

Quorum operates on a simple majority rule, meaning that a majority of quorum nodes in the cluster must be accessible before any node in the cluster can access a file system. This keeps any nodes that are cut off from the cluster (by a network failure for example) from writing data to the file system. When nodes fail, quorum must be maintained in order for the cluster to remain online. If quorum is not maintained, IBM Storage Scale file systems unmount across the cluster until a quorum is reestablished, at which point file system recovery occurs. For this reason, it is important that the set of quorum nodes be carefully considered.

IBM Storage Scale can use one of the following two methods for determining quorum:

- Node quorum

  Node quorum is the default quorum algorithm for IBM Storage Scale. Quorum is defined as one plus half of the explicitly defined quorum nodes in the IBM Storage Scale cluster. There are no default quorum nodes; you must specify which nodes have this role.
- Node quorum with tiebreaker disks

  Tiebreaker disks can be used in shared storage configurations in order to preserve quorum. Because clusters that run IBM Storage Scale Erasure Code Edition do not typically use shared storage, only quorum nodes need to be configured based on the number of recovery groups and IBM Storage Scale Erasure Code Edition nodes that are configured in the cluster. It is best to configure an odd number of quorum nodes, with 3, 5, or 7 nodes being the typical numbers used. Suppose a cluster spans multiple failure domains, such as racks, power domains, or network domains. In that case, it is best to allocate quorum nodes from each failure domain to maintain availability. The number of quorum nodes with the Erasure Code selection determines the maximum number of nodes that can simultaneously fail in the cluster.

  It is best to allocate quorum nodes as nodes that do not require frequent reboots or downtime. If possible, choose nodes that do not run intensive compute or network loads, as these might impact the quorum messages. This becomes more important as clusters grow larger in size, as the number of quorum messages increase. Finally, quorum nodes are used to maintain critical configuration data, which is stored on the operating system disk in the /var file system. In order to preserve access to this data, it is best to ensure that any workloads on the quorum node do not overly stress the disk that the /var file system resides on. Also, note that /var file system must be on persistent local storage for each quorum node.

Manager nodes

When defining an IBM Storage Scale cluster, we define one or more manager nodes. Manager nodes are used for various internal tasks.

For each file system, one manager node is designated as a file system manager. This node is responsible for providing certain tasks, such as file system configuration changes, quota management, and free space management. In addition, manager nodes are responsible for token management throughout the cluster. Due to the extra load on manager nodes, it is generally recommended to not run tasks on a manager node that are time sensitive, that require real-time response, or that might excessively use the system CPU or cluster network. Any tasks that might slow the IBM Storage Scale file system daemon affect the overall response of the file system throughout the cluster.

For large clusters of 100 or more nodes, or clusters where the maxFilesToCache parameter is modified from the default, it is necessary to consider the memory use on manager nodes for token management. Tokens are used in order to maintain locks and consistency when files are opened in the cluster. The number of tokens in use depends on the number of files that each node might open or cached and the number of nodes in the cluster. For large clusters (generally 512 nodes or more), it might be beneficial to have dedicated nodes responsible for the manager role.

To determine the overall token memory used in a system, an approximation is to examine the maxFilesToCache (default 4000) and maxStatCache (default 1000) for all nodes. Each token uses approximately 512 bytes of memory on a token manager node. For example, a 20-node cluster of default values use (4000 + 1000) tokens * 20 nodes * 512 bytes/token = approx. 49 MB of memory. This memory is distributed across all manager nodes, as all manager nodes share the role of token management. If there are four manager nodes in the above example, each manager node is responsible for just over 12 MB of tokens. For fault tolerance, it is best to leave room for a manager node to go down, so we can assume just over 16 MB of memory required.

For default values, the token memory is not considered small or mid-sized clusters with default values. However, it can be beneficial to increase the maxFilesToCache on nodes to hundreds of thousands or even millions of files in some cases. In these cases, it is important to calculate the additional memory requirement and ensure that any nodes have enough memory beyond the IBM Storage Scale Erasure Code Edition requirements to perform token management tasks.

You can have uniform workload on each IBM Storage Scale Erasure Code Edition storage node, to the degree possible. For this reason, either all nodes in the recovery group can be manager nodes or none of the nodes can be manager nodes. In storage clusters that are composed of only IBM Storage Scale Erasure Code Edition storage nodes, you can set all nodes as manager nodes. In a large cluster or a cluster with more than one IBM Storage Scale Erasure Code Edition recovery group, you can set the manager nodes in one recovery group, several recovery groups together, or on separate non-recovery group nodes. The default limit of maximum token manager nodes is 128, and the limit must not be changed.

CES nodes

Cluster Export Services (CES) is used to provide SMB, NFS, or Object access to data in the IBM Storage Scale file system.

For environments with high-performance requirements, separate CES nodes are necessary. In these environments, it is suggested that a CES node runs no other workload other than the export services. For more information about the memory and CPU requirements for CES nodes, see IBM Storage Scale FAQs.

Finally, the network that is used for accessing the nodes through CES protocols must run on a different physical adapter and network than the network used for IBM Storage Scale Erasure Code Edition traffic. Typically, this means that a CES node has at least two adapters, one for node-to-node access for IBM

Storage Scale, and one for CES protocol access. This recommendation helps ensure that CES protocol traffic does not interfere with the IBM Storage Scale traffic, which results in better overall performance and improved cluster stability.

NSD server nodes

In some cases, a cluster might contain both IBM Storage Scale RAID storage, and other storage subsystems, such as IBM V5000, V7000, or other storage arrays. This storage can be made available for separate file systems or to tier data from a single file system.

In this case, a number of servers, typically at least two, are attached to the external storage system that uses Fibre Channel or a similar interconnect. These then serve NSDs to the rest of the cluster. It is mandatory that any server that provides NSDs to the rest of the cluster be dedicated servers, separate from the servers providing storage for IBM Storage Scale RAID. These servers typically must not run any applications. If applications are run on these servers, then they must not be time critical, as the demands of servicing disk requests might conflict with these applications. The connectivity of these servers must be sufficient to meet the requirements of the attached disk. Ensure that CPU and network bandwidth are capable of driving the attached disk system sufficiently.

Default helper node

Certain IBM Storage Scale commands that can generate a significant amount of IO, such as file system restripes, adding disks, or policy scans, use helper nodes to run faster.

These nodes can be specified by using the '-N' flag to the command or by using the *defaultHelperNode* configuration value. Some commands, such as **mmapplypolicy**, might use more memory or CPU resources while you run to sort file lists. Other commands, such as **mmrestripefs**, or **mmdelsnapshot**, might generate a significant amount of IO to move data and update metadata structures. When you specify helper nodes, it is best to ensure that these nodes have sufficient memory, idle CPU, and network to handle these requests. It might be necessary to schedule these commands for a time when the nodes or cluster are not heavily used as well.

Commands that use helper nodes include: **mmadddisk**, **mmapplypolicy**, **mmbackup**, **mmchdisk**, **mmcheckquota**, **mmdefragfs**, **mmdeldisk**, **mmdelsnapshot**, **mmfileid**, **mmfsck**, **mmimgbackup**, **mmimgrestore**, **mmrestorefs**, **mmrestripefs**, and **mmrpldisk**. Helper nodes typically must be separated from the servers that provide storage for IBM Storage Scale RAID.

AFM gateway node

On AFM cache clusters, AFM uses gateway nodes in order to connect to the home system. Each AFM-enabled fileset uses a designated primary gateway node in order to connect to home and fail over to other gateway nodes as necessary.

AFM gateway nodes might generate a large amount of network traffic between themselves and the home system to fetch and synchronize files. The bandwidth and latency on this network can directly impact file operations on AFM-enabled filesets. In order to ensure best performance and cluster stability, it is best to have AFM traffic use a different physical adapter than the IBM Storage Scale cluster network. It is best to use designated gateway nodes that are not used for other application workloads. AFM uses additional node memory and cache entries on gateway nodes, so applications that run on these nodes compete for cache usage, which slows both the application and AFM operations. AFM gateway nodes are required to be separate from the servers that provide storage for IBM Storage Scale RAID.

IBM Storage Protect backup node

This topic describes how IBM Storage Scale is integrated with IBM Storage Protect.

IBM Storage Scale can integrate with IBM Storage Protect in one of two ways. IBM Storage Scale can be used as a backup pool for IBM Storage Protect. In this use, external clients use IBM Storage Protect to back up their data to the file system. Alternatively, IBM Storage Protect can also be used to back up the IBM Storage Scale file system itself. When you use IBM Storage Scale as a backup target, one or more nodes run the IBM Storage Protect server. This server is contacted by other clients in order to back up. The IBM Storage Scale server must communicate to external clients via a separate network used for internal cluster traffic, due to the bandwidth requirements on this server.

IBM Storage Scale can also integrate with IBM Storage Protect in order to back up the IBM Storage Scale file system. One or more nodes in the cluster can run the IBM Storage Protect agents, which transfer data to an IBM Storage Protect server. Other backup platforms also might use a similar agent to scan and migrate data on a file system.

Backup nodes can become heavily used during the backup window, when data is scanned and transferred to the backup provider. It is best to use a separate network on these nodes for communication with the backup server. It is also best to not run any other applications on these nodes, especially during the backup window itself.

IBM Storage Protect uses the IBM Storage Scale policy engine to scan for changed files. This scan can run across multiple nodes in the cluster, other than just the node that runs the backup agent. For more information about guidance on helper nodes during a policy scan, see the "Default helper node" on page 23 section.

Both nodes used to run the IBM Storage Protect server, as well as nodes that run the client are required to be separate from the servers that provide storage for IBM Storage Scale RAID.

Transparent cloud tiering nodes

Transparent cloud tiering might use 1-4 gateway nodes per file system to communicate to a cloud provider.

These nodes are used to transfer files to and from the cloud provider. During large file migrations, or if users need to recall files, these nodes might be used heavily for file transfer. It is best to communicate to the cloud provider on a different physical network than the network used for internal cluster communications. On heavily used clusters, Transparent cloud tiering might impact any other applications that run on these nodes. Transparent cloud tiering gateway nodes are necessary to be separate from the servers that provide storage for IBM Storage Scale RAID.

IBM Storage Scale Management Interface Node

IBM Storage Scale Management Interface supports both GUI and RESTful API access to an IBM Storage Scale cluster.

IBM Storage Scale Management Interface can run on 1 or more dedicated nodes within the cluster. These nodes run processes and databases to monitor the cluster. The GUI consumes extra memory as well as internal hard drive space for state databases. The GUI node might also run scheduled tasks to monitor the health and utilization of the cluster. It is best to not run any compute or memory-intensive applications on the GUI node, as the GUI might impact the performance of these applications. In many cases, the nodes that run the management interface are also used as the call home server and the performance monitoring collector. Management interface nodes are required to be separate from the servers that provide storage for IBM Storage Scale RAID.

IBM Storage Scale call home nodes

IBM Storage Scale call home is used to send diagnostic data to IBM.

Nodes are arranged into call home servers, which are responsible for collecting all of the data within a call home group and sending the data to IBM. Large clusters might consist of several groups. Use call home whenever possible to assist in gathering data for support.

Call home servers are required to be separate from severs providing storage for IBM Storage Scale RAID. In the case of small clusters of 32 nodes or less, the call home server might be the same as the management interface node. In larger clusters, additional call home servers might be required. For more information on sizing call home requirements, see the *Understanding call home* topic in the *IBM Storage Scale: Administration Guide*.

Performance monitoring

IBM Storage Scale performance monitoring divides nodes into collector and sensor nodes. Sensors run on all nodes that you want to collect performance data from. Collectors run on few nodes and are used to aggregate all of the sensor data into a single view. Sensors can run on all nodes, including nodes that provide storage for IBM Storage Scale Erasure Code Edition. Collectors must be run on nodes that do not provide IBM Storage Scale Erasure Code Edition storage. Typically, the same nodes that are used as management interface nodes would be used as collector nodes. On clusters with hundreds of nodes, multiple collectors might be required in order to aggregate data across the cluster. It is not recommended to run real-time or time-sensitive tasks on collector nodes.

File audit logging and watch folders

File Audit Logging (FAL) and Watch Folders use message queues in order to monitor file access on the cluster.

FAL producers create messages when certain file operations are performed (for example, file writes, reads, and so on). FAL consumers read these messages and perform required actions, such as writing to audit logs. All nodes, including the nodes providing IBM Storage Scale Erasure Code Edition storage may be producers, in order to provide complete access logging. Consumers must be on nodes that do not provide storage to IBM Storage Scale Erasure Code Edition cluster, due to the additional load on the system caused by monitoring usage on the cluster. In addition, consumer nodes should not run real-time or time-sensitive applications.

Other IBM Storage Scale features

IBM Storage Scale offers caching features such as local read-only cache (LROC) and high-availability write cache (HAWC) can provide additional high-speed caching to speed up certain applications.

LROC and HAWC can be used on file systems that contain storage provided by IBM Storage Scale Erasure Code Edition. However, LROC and HAWC devices cannot be installed directly on nodes providing IBM Storage Scale Erasure Code Edition storage. Client nodes that are not part of the IBM Storage Scale Erasure Code Edition recovery group can use these devices.

Planning for recovery group space and scale up

IBM Storage Scale Erasure Code Edition manages all disks in the recovery group. The recovery group puts different type of disks into the declustered array. The declustered array space is the base for creating vdisks.

IBM Storage Scale Erasure Code Edition supports the scale-up of declustered array space by adding new disk in the server. When you plan for IBM Storage Scale Erasure Code Edition recovery group, it is needed to consider the initial disk type and the disk number on each server node. The method for expanding disk slots needs to be planned.

When you add new disks into the declustered array of recovery group, ensure to meet the following requirements:

- The new disks that are inserted into the empty slots of the server for expanding declustered array space must be the same type of the existing disks in the declustered array.
- All nodes in the recovery group must have the same number of new disks that are added for the declustered array.
- New disks might be added into multiple declustered arrays at the same time.
- It is recommended to add the same number of disks as the existing disks in the declustered array to double the declustered array space.

  For example, start from 1/2 populated (20 disks per node) for high storage density servers (40 disk slots per node), then add another 1/2 populated in scale up.

You can start from 1/4 populated (10 disks per node), then add another 1/4 populated in each of the scale up until the server is fully populated.

Chapter 4. Installing IBM Storage Scale Erasure Code Edition

You can install IBM Storage Scale Erasure Code Edition by using the installation toolkit.

IBM Storage Scale Erasure Code Edition installation prerequisites

IBM Storage Scale Erasure Code Edition requires several software packages in addition to the base operating system.

Before you install IBM Storage Scale Erasure Code Edition, your network must pass the latency network KPIs for Ethernet networks to support RDMA network on x86_64.

Note: In IBM Storage Scale Erasure Code Edition, customers are required to meet the following network and storage KPI metrics before an installation is completed. For more information, see "Network requirements and precheck" on page 15 and "Disk requirements and precheck" on page 16. Also, you must verify that the hardware that is planned for IBM Storage Scale Erasure Code Edition storage servers meets the minimum requirements. For more information, see "Minimum hardware requirements and precheck" on page 9. The installation toolkit would also verify that your hardware meets minimum requirements, but it is useful to execute this tool before you begin the installation.

The following RPMs are required to be installed[1]:

- `sg3_utils`
- `nvme-cli`
- `storcli` (if you use SAS drives with LSI HBA)
- `perccli` (if you use SAS drives with DELL RAID controller)
- `dmidecode` (x86_64 only)
- `PyYAML`
- `Sqlite`

Note: To use IBM Storage Scale Erasure Code Edition 5.1.0 and later releases on RHEL 7.*, install PyYAML for python3 by using one of the following two methods:

- **Method 1**: Using the pip3 command:

  ```
  pip3 install pyyaml
  ```

- **Method 2**: Download the package and install it:

  1. Download the PyYAML package from http://pyyaml.org/download/pyyaml/PyYAML-5.3.1.tar.gz.
  2. Extract the package in a directory and issue the following command from that directory.

     ```
     python3 setup.py --without-libyaml install
     ```

Furthermore, it is important to ensure that you have the latest version of Mellanox OFED installed on each x86_64 node. Likewise, the driver versions must be maintained at a consistent level across all nodes.

Note: All IBM Storage Scale cluster software and configuration prerequisites must also be satisfied. For more information, see *Installation prerequisites* in *IBM Storage Scale: Concepts, Planning, and Installation Guide*.

IBM Storage Scale Erasure Code Edition precheck

IBM Storage Scale Erasure Code Edition precheck is integrated with the installation toolkit installation, deployment, or upgrade precheck. For IBM Storage Scale Erasure Code Edition, the precheck includes the following on all scale-out nodes:

- Check whether the CPU requirements are met.
- Check whether the memory requirements are met.
- Check whether the OS is supported.
- Check whether the networking requirements that include the required NIC and SAS adapters are met.
- Check whether the required syscall parameters are set correctly.

Installation toolkit-related prerequisites

- Ensure that networking is set up in one of the following ways.
  - DNS is configured such that all hostnames, either short or long, are resolvable.
  - All hostnames are resolvable in the `/etc/hosts` file. The host entries in the `/etc/hosts` file must be in the following order:

    ```
    <IP address> <Fully qualified domain name> <Short name>
    ```
- Passwordless SSH must be set up using the FQDN and the short name of the node.

For more information, see *Preparing to use the installation toolkit* in *IBM Storage Scale: Concepts, Planning, and Installation Guide*.

Installation toolkit-related limitations

- The installation toolkit is not supported in a sudo wrapper environment. Therefore, sudo wrappers cannot be used for installation, deployment, or upgrade of IBM Storage Scale Erasure Code Edition. After installation, deployment, or upgrade, you can use sudo wrappers for administration tasks in an IBM Storage Scale Erasure Code Edition environment.

  For more information, see *Limitations of the installation toolkit* in *IBM Storage Scale: Concepts, Planning, and Installation Guide*.
- The installation toolkit does not support advanced parameters of vdisk sets and file systems that can be specified by using the **mmvdisk** command. After recovery groups are created, you can use the **mmvdisk** command to create vdisk sets and file systems with the advanced configuration parameters. Thereafter, you can use the installation toolkit deployment operation for protocol deployment.
- The installation toolkit can be used to add new nodes to an existing IBM Storage Scale Erasure Code Edition cluster. But, the installation toolkit cannot be used for adding the new node into the existing recovery group. Use the **mmvdisk** command instead for adding new nodes into the existing recovery group.
- The installation toolkit cannot accept multiple recovery groups as an argument while you define a vdisk set. If you want to specify more than one recovery group with the vdisk set, use the **mmvdisk** command after the installation phase is completed.
- The installation toolkit does not support the creation of hybrid clusters (IBM Storage Scale + ESS + IBM Storage Scale Erasure Code Edition).
- The installation toolkit cannot be used to configure the recovery group servers. If you want to change the default settings for pagepool, vdisk space, or map memory, use the **mmvdisk** command after the recovery groups are created.

IBM Storage Scale Erasure Code Edition installation overview

The installation of IBM Storage Scale Erasure Code Edition by using the installation toolkit occurs in these phases.

Phase 1: Network and hardware precheck

1. Download or clone the following `ece_os_readiness`, `ece_network_readiness`, and `ece_storage_readiness` precheck tools on one of the nodes that are planned for your IBM Storage Scale Erasure Code Edition storage configuration.

   - These tools are available on the IBM Storage Scale Tools GitHub, https://github.com/IBM/SpectrumScaleTools

2. Run the `ece_os_readiness` precheck tool on at least one of your IBM Storage Scale Erasure Code Edition storage nodes for each recovery group. Review the `README.md` file carefully for prerequisites and execution procedures.
3. Run the `ece_network_readiness` and `ece_storage_readiness` precheck tools that include each IBM Storage Scale Erasure Code Edition storage node. Review the `README.md` file carefully for prerequisites and execution procedures.

Phase 2: Cluster definition

By using the **`./spectrumscale`** command, the following steps are done:

1. Installer node is defined by the user.
2. Setup type is specified as `ece` by the user.
3. Scale-out nodes and other node designations are done by the user.

   Other types of nodes that can be designated include protocol, GUI, call home, and file audit logging. If you are planning to use GUI, call home, performance monitoring, or file audit logging, you must add a client node for each of these functions.

   Note: When you are adding a node in an existing cluster, the installation toolkit adds only the node in the existing cluster with the client or the server license. You must use the **`mmvdisk`** command to manually add the node into the existing node class.

4. Recovery group is defined by the user.

   Note: Recovery group definition can be done after the first installation run is done in which the package is installed and the cluster is created. With the package installed, the user can use the IBM Storage Scale Erasure Code Edition slot-mapping tool to create disk slot location configurations on IBM Storage Scale Erasure Code Edition storage servers. Then, the user can define the recovery group and run installation phase again to create the recovery group.

5. Vdisk set is defined by the user. [Vdisk set definition can be done after the installation phase]

   Note:

   - The installation toolkit can be used to add new nodes to an existing IBM Storage Scale Erasure Code Edition cluster. But, the installation toolkit cannot be used for adding the new node into the existing recovery group. Use the **`mmvdisk`** command instead for adding new nodes into the existing recovery group.
   - The installation toolkit cannot accept multiple recovery groups as an argument while you define a vdisk set. If you want to specify more than one recovery group with the vdisk set, use the **`mmvdisk`** command after the installation phase is completed.
   - Depending on the cluster configuration, there might be cases where the settings for pagepool, vdisk space, or map memory of the recovery group servers need to be changed to fulfill the requirements for the vdisk set definition. The installation toolkit cannot be used to configure the recovery group servers. Instead, use the **`mmvdisk`** command to change the recovery group server configuration after the recovery groups are created.

6. File system is defined by the user. [File system definition can be done after the installation phase]

Phase 3: Installation

This phase starts upon issuing the **`./spectrumscale install`** command.

1. IBM Storage Scale Erasure Code Edition packages that include IBM Storage Scale Erasure Code Edition license package are installed.
2. IBM Storage Scale Erasure Code Edition cluster is created.
3. Quorum and manager nodes are configured.
4. Server and client licenses are applied.
5. Node class is created.
6. Recovery group is created.
7. Vdisk sets are created.
8. File systems are created.

Note: During the installation, support packages are also installed. These support packages include supported disk topologies and starting udev rules for each node. There is a rule file that is placed here: `/etc/udev/rules.d/99-ibm-scaleout.rules`. These rules have these settings and they are meant to be a good starting point for a typical hardware configuration. You might need to adjust these settings for your hardware configuration:

```
#
# IBM Storage Scale RAID (GNR) block device attributes for
# Erasure Code Edition (ECE) storage-rich servers.
#
# These are least common denominator settings. It is likely
# that specific installations can increase especially the
# max_sectors_kb for GNR pdisks.
#
# After initial ECE installation and after any change to the
# contents of these rules, run
#       udevadm trigger --subsystem-match=block
# and inspect /var/log/messages for unexpected udev entries.
# Subsequent reboots and block device replacement will
# automatically invoke these rules as "add|change" events.
#
# -------------------------------------------------------------------------
#
# Identify the boot SCSI disk by the presence of a SWAP partition.
# Set boot disk nr_requests and queue_depth to reasonable values.
#
ACTION=="add|change", SUBSYSTEM=="block",
KERNEL=="sd*[^0-9]", PROGRAM="/usr/bin/lsblk -rno
FSTYPE,MOUNTPOINT,NAME /dev/%k", RESULT=="*SWAP*",
ATTR{queue/nr_requests}="128", ATTR{device/queue_depth}="64"
#
# Identify eligible GNR SCSI pdisks by the absence of a SWAP partition.
# Set preferred GNR attributes. The only attribute that should possibly
# be changed is max_sectors_kb, up to a value of 8192, depending on
# what the SCSI driver and disks support.
#
ACTION=="add|change", SUBSYSTEM=="block",
KERNEL=="sd*[^0-9]", PROGRAM="/usr/bin/lsblk -rno
FSTYPE,MOUNTPOINT,NAME /dev/%k",
RESULT!="*SWAP*", ATTR{queue/scheduler}="deadline",
ATTR{queue/nr_requests}="256", ATTR{device/queue_depth}="31",
ATTR{queue/max_sectors_kb}="1024", ATTR{queue/read_ahead_kb}="0",
ATTR{queue/rq_affinity}="2"
#
# Identify eligible GNR NVMe pdisks by the absence of a MOUNTPOINT.
# Set preferred GNR attributes. The only attribute that should possibly
# be changed is max_sectors_kb, up to a value of 8192, depending on
# what the NVMe driver and devices support.
#
ACTION=="add|change", SUBSYSTEM=="block",
KERNEL=="nvme*", KERNEL!="nvme*p[0-9]",
PROGRAM="/usr/bin/lsblk -rno
FSTYPE,MOUNTPOINT,NAME /dev/%k", RESULT!="*/*",
ATTR{queue/scheduler}="none", ATTR{queue/nr_requests}="256",
ATTR{queue/max_sectors_kb}="128",
ATTR{queue/read_ahead_kb}="0",
ATTR{queue/rq_affinity}="2"
```

Note: If you are planning to deploy protocols in the IBM Storage Scale Erasure Code Edition cluster, you must define a CES shared root file system before you initiate the installation toolkit deployment phase by using the following command.

```
./spectrumscale config protocols -f FileSystem -m MountPoint
```

Quorum or manager node rules in IBM Storage Scale Erasure Code Edition

- In case of a single recovery group, the following quorum node rules apply.
  - When the number of scale-out nodes is 4, the number of quorum nodes is set to 3.
  - When the number of scale-out nodes is 5 or 6, the number of quorum nodes is set to 5.
  - When the number of scale-out nodes is 7 or more, the number of quorum nodes is set to 7.
- If the number of recovery groups is more than 1 and less than or equal to 7, 7 quorum nodes are distributed across recovery groups in a round robin manner.
- If the number of recovery groups is more than 7, 7 recovery groups are selected as quorum holders.
- If there is no recovery group or quorum node that is defined in the cluster configuration, the installation toolkit displays the following message.

  ```
  "You have not defined any recovery group in the cluster configuration.
  Installer will automatically define the quorum configuration. Do you want to continue"
  ```

  If you specify `yes` then the quorum nodes are distributed according to the single recovery group rule.
- If you are adding a new recovery group in an existing cluster or if you want to add a new node into the existing node class, the existing quorum configuration is not modified by the installation toolkit.
- For an existing cluster, if you want to have quorum on a different node or a different recovery group then you must use an IBM Storage Scale command such as **mmchnode** to change this configuration.

Installing IBM Storage Scale Erasure Code Edition by using the installation toolkit

IBM Storage Scale Erasure Code Edition is available in a separate installation package and you install it by using the installation toolkit.

Use the following steps to install IBM Storage Scale Erasure Code Edition.

1. Download IBM Storage Scale Erasure Code Edition self-extracting package from the IBM Storage Scale page on Fix Central.
2. Extract the installation package.

   ```
   # ./Spectrum_Scale_Erasure_Code-5.x.y.z-x86_64-Linux-install --textonly
   ```

   The installation toolkit gets extracted to the `/usr/lpp/mmfs/5.x.y.z/ansible-toolkit/` directory.
3. Change the directory to where the installation toolkit is extracted.

   ```
   cd /usr/lpp/mmfs/5.x.y.z/ansible-toolkit/
   ```

4. Specify the installer node and the setup type in the cluster definition file.

   The setup type must be `ece` for IBM Storage Scale Erasure Code Edition.

   ```
   ./spectrumscale setup -s InstallerNodeIP -st ece
   ```

5. Add scale-out nodes for IBM Storage Scale Erasure Code Edition in the cluster definition file.

   ```
   ./spectrumscale node add NodeName -so
   ```

Specify any other node designations in the cluster definition file. You can use the following command to change the cluster name:

```
./spectrumscale config gpfs -c ece_cluster
```

Note: For environments with high-performance requirements, IBM Storage Scale Erasure Code Edition storage nodes must not be assigned file audit logging, call home, or protocol node roles.

You can use the following command to display the list of nodes that are specified in the cluster definition file and the respective node designations.

```
./spectrumscale node list
```

A sample output is as follows:

```
[ INFO  ] List of nodes in current configuration:
[ INFO  ] [Installer Node]
[ INFO  ] 198.51.100.15
[ INFO  ]
[ INFO  ] [Cluster Details]
[ INFO  ] Name: ece_cluster
[ INFO  ] Setup Type: Erasure Code Edition
[ INFO  ]
[ INFO  ] [Extended Features]
[ INFO  ] File Audit logging     : Disabled
[ INFO  ] Watch folder           : Disabled
[ INFO  ] Management GUI         : Enabled
[ INFO  ] Performance Monitoring : Enabled
[ INFO  ] Callhome               : Disabled
[ INFO  ]
[ INFO  ] GPFS              Admin  Quorum  Manager   NSD   Protocol   GUI    Perf Mon
Scale-out   OS   Arch
[ INFO  ] Node               Node   Node     Node   Server   Node    Server Collector
Node
[ INFO  ] node1.example.com           X       X
X     rhel7  x86_64
[ INFO  ] node2.example.com           X       X
X     rhel7  x86_64
[ INFO  ] node3.example.com   X                                        X
X               rhel7  x86_64
[ INFO  ] node4.example.com           X       X
X     rhel7  x86_64
[ INFO  ] node5.example.com           X       X
X     rhel7  x86_64
[ INFO  ] node6.example.com           X       X
X     rhel7  x86_64
[ INFO  ] node7.example.com
X                                 rhel7  x86_64
[ INFO  ] node8.example.com
X                                 rhel7  x86_64

[ INFO  ]
[ INFO  ] [Export IP address]
[ INFO  ] [Export IP address]
[ INFO  ] 198.51.100.11 (pool)
[ INFO  ] 198.51.100.12 (pool)
```

6. Perform environment prechecks before you issue the installation toolkit installation command.

```
./spectrumscale install -pr
```

Note: You can skip the following installation and slot location configuration steps, if:

- IBM Storage Scale Erasure Code Edition storage server already has the correct slot location configuration files.

When you continue with the recovery group definition step, the cluster and the recovery group are created together after defining the recovery group.

7. Perform the installation toolkit installation procedure.

```
./spectrumscale install
```

Now the GPFS cluster is created.

8. Map the slot location configuration on IBM Storage Scale Erasure Code Edition storage servers by using the slot-mapping tool.

   For more information, see “Setting up IBM Storage Scale Erasure Code Edition for disk slot location” on page 34.

   Note: If you want to create file systems with the Recovery group manually or you want to work around the installation toolkit limitation, see “ IBM Storage Scale Erasure Code Edition configurations” on page 41 to use the **mmvdisk** command directly to finish the deployment.
9. Define the recovery group for IBM Storage Scale Erasure Code Edition in the cluster definition file.

   ```
   ./spectrumscale recoverygroup define -N Node1,Node2,...,NodeN
   ```
10. Do the installation toolkit installation procedure again to create recovery group.

    ```
    ./spectrumscale install
    ```

    Note:

    After this step, a Recovery Group should be created.

    Before creating vdisks, you must check or adjust the Declustered Array spare disk space following the “Recommendations” on page 18.

    If there is any problem in creating recovery group or you want to work around the installation toolkit limitation on vdisk and file system, see “ IBM Storage Scale Erasure Code Edition configurations” on page 41 to use the **mmvdisk** command directly to finish the deployment.
11. Define vdisk sets for IBM Storage Scale Erasure Code Edition in the cluster definition file.

    a. Check declustered array information.

    ```
    ./spectrumscale recoverygroup list
    ```

    Sample output is as follows:

    ```
    [ INFO  ] Name nodeclass Server                        DA_Name:FreeCapacity:Type
    [ INFO  ] rg_1 nc_1      node1,ndoe2,node4,node5,node6 DA1:9757G:NVMe,DA2:8829G:HDD
    ```

    b. Define vdisk sets. The following example defines one vdiskset in DA1 of NVMe for file system metadata, one vdiskset in DA2 of HDD for file system data.

    ```
    ./spectrumscale vdiskset define -vs nvme_meta -rg rg_1 -code 4WayReplication -bs 2M -ss
    50% -da DA1 -nsd-usage metadataOnly -storage-pool system

    ./spectrumscale vdiskset define -vs hdd_data -rg rg_1 -code 4+3P -bs 8M -ss 80% -da DA2
    -nsd-usage dataOnly -storage-pool datapool
    ```

    c. Check vdiskset definition.

    ```
    ./spectrumscale vdiskset list
    ```

    Sample output is as follows:

    ```
    [ INFO  ] name      recoverygroup blocksize setsize  RaidCode nsdUsage     poolName daName
    [ INFO  ] hdd_data  rg_1            8M      80%      4+3P   dataOnly     datapool    DA2
    [ INFO  ] nvme_meta rg_1            2M      50%    4WayReplication metadataOnly   system    DA1
    ```
12. Define the file system for IBM Storage Scale Erasure Code Edition in the cluster definition file.

    ```
    ./spectrumscale filesystem define -fs gpfs1 -vs hdd_data,nvme_meta --mmcrfs '-T /gpfs1'
    ```
13. Do environment prechecks before you run the installation toolkit installation procedure.

    ```
    ./spectrumscale install -pr
    ```
14. Do the installation toolkit installation procedure again to create the vdisk sets and file systems.

    ```
    ./spectrumscale install
    ```

For troubleshooting, see *Installation and configuration issues* topic in the *IBM Storage Scale: Problem Determination Guide*.

Setting up IBM Storage Scale Erasure Code Edition for disk slot location

IBM Storage Scale Erasure Code Edition includes the support for NVMe disks, spinning SAS disks, and SAS SSDs. To support disk replacement for these drives, the configuration mapping files that describe a server's disk layout and capabilities must be generated.

The **`/usr/lpp/mmfs/bin/ecedrivemapping`** command generates mapping files for the corresponding drives with the information that is provided by the user. This command identifies the disks within the server, requests for the appropriate slot location for each disk, and generates a specific mapping file.

```
ecedrivemapping --help
ecedrivemapping [-h] [--mode {nvme,lmr}]
                [--slotrange {0-MAX_SLOT} {0-MAX_SLOT}] [--report]
                [--force] [--nvmeversion {1,2}][--lmrversion {1,2}]
```

- The `--mode` argument has two specifications: `nvme` and `lmr`. The `nvme` mode handles the mapping of NVMe drives, while the `lmr` (LSI megaraid) mode handles the mapping of spinning SAS drives and SAS SSDs.
- The `--slotrange` argument provides a proper indicator for what slots the drives must be mapped too. The command fails if a slot is not within the specified range or if more disks identified than the slot range capable of handling. If there are more slots than disks identified, the `--force` argument must be specified, otherwise the command fails.
- The `--report` argument provides a quick overview of current map file. The `--mode` argument must be specified to determine which mapping file to be summarized.
- The `--force` argument must be provided if the slot range given contains more slots than disks identified.

  For example, the slot range 0-10 is provided, but slots 5-6 are not used. The `--force` argument must be provided for proper execution of the mapping procedure.

  Note: NVMe disks mapping must have a slot range that matches the disk identified. All NVMe disks must be adjacent to each other in the slots. The `–force` option should not be used for NVMe drives.
- The `--nvmeversion` argument determines the version of the mapping file for drives that are handled by the `nvme` mode. Default to the highest version when possible. Currently, it supports only versions 1 and 2.
- The `--lmrversion` argument determines the version of the mapping file for drives that are handled by the `lmr` mode. Default is the highest version. Currently, it supports only versions 1 and 2.

Running the **`ecedrivemapping`** command identifies disks, requests slot locations, and generates map files for both NVMe drives, spinning SAS drives, and SAS SSDs.

The **`ecedrivemapping`** command allows for specific drives to be mapped by specifying the `--mode` argument. Specify the `--mode` argument only if you do not have to map the drives in the other mode or if you need to view a summary of the map file.

Note:

You must make sure there are no duplicated slot locations between the modes.

Different hardware configurations on servers can introduce a mismatch in the disk slot locations. A mismatch in the disk slot locations would lead to issues with replacing the disks. It is suggested to generate mapping files of disks for each server.

You must verify the slot location on each server when copying mapping config files between servers.

Mapping NVMe disk slot location

IBM Storage Scale Erasure Code Edition requires additional configuration for use with NVMe drives.

IBM Storage Scale Erasure Code Edition brings enclosure-like management services to direct attached storage disks, allowing users to identify and replace disks without compromising system availability or integrity. IBM Storage Scale Erasure Code Edition ships with support for NVMe disks with a U.2 form factor. The U.2 form factor allows system administrators to replace NVMe disks as if they were regular HDD or SSD drives. Drive LED control is not supported currently, but replacement operations will work with their slot location. This means that NVMe drives might be replaced, but the replacement process would not trigger any identification or replace lights on the drive. For more information on disk replacement procedure, see "Physical disk procedures" on page 87.

Note: For supporting NVMe drives with LED control of showing the disk state, an RPQ is required for IBM to review. It is highly hardware-related and only some types of servers are tested. Ask your sales representative to contact IBM Storage Scale development about the RPQ or SCORE process.

To support disk replacement for NVMe drives on IBM Storage Scale Erasure Code Edition, users need to define a pseudo enclosure describing a server's disk layout and capabilities.

Creating an Enclosure Descriptor File (EDF)

U.2 NVMe drives reside in a pseudo enclosure within their server node. This pseudo enclosure is defined by using a plain-text EDF. The EDF describes the structure and layout of the storage components within the enclosure, as well as the capabilities of these components.

The EDF also contains a structure that is known as a "bay_map", which describes a mapping from the server's external drive slots to hardware address. A given server node's slot to hardware address mapping might vary depending on its vendor and its internal cabling. Therefore, this mapping is crucial to ensure that disk replacement operations select the correct disk. It is recommended to use the same server node hardware across an IBM Storage Scale Erasure Code Edition recovery group, as this ensures a uniform NVMe drive mapping and allows a single EDF to be deployed on all nodes without additional configuration. Otherwise, a separate EDF must be created on each node.

Note:

- NVMe drives might be organized into exclusive namespaces on a single controller or shared namespaces across multiple controllers. For use with IBM Storage Scale Erasure Code Edition, NVMe drives must be configured such that there is a single namespace on each controller.
- NVMe drive slot mapping must be done before the recovery group creation. IBM Storage Scale Erasure Code Edition supports doing the mapping and the remapping after the recovery group is created. The `ecedrivemapping` tool only reads data from NVMe drives. Do not write data on NVMe drives after the recovery group is created. The `*.edf` files are located in `/usr/lpp/mmfs/data/gems` directory.
- To define NVMe drive mapping, you must first select a server and populate all NVMe-capable slots with NVMe drives. After the mapping process is completed, the extra drives can be returned to the spare inventory or to other servers. This can be done once for each collection of servers with the same disk topology. IBM Storage Scale Erasure Code Edition does not support mapping additional NVMe server slots after this initial NVMe drive mapping is completed.

Mapping NVMe disks

The creation of a properly formatted and named EDF with a correct bay_map is produced by the **`ecedrivemapping`** command.

```
ecedrivemapping [-h] [--mode {nvme,lmr}]
                    [--slotrange {0-MAX_SLOT} {0-MAX_SLOT}] [--report]
                    [--force] [--version {1,2}]
```

For example, to create an EDF that describes NVMe-capable server slots 16-18, issue the following command.

```
# ecedrivemapping  --mode nvme --slotrange 16 18
```

Note:

- If the `--slotrange` argument is not specified, the slot range is immediately requested upon execution of the **ecedrivemapping** command.
- If the `--mode` argument is not specified, the slot range is applied to NVMe disks and lmr disks.

For each NVMe block device found in `/proc/partitions`, the tool flashes that block device's activity light by using a read workload. This prompts the user to enter the corresponding slot for the flashing disk.

```
>>> 3 Nvme drives were detected
Now blinking path /dev/nvme0n1
>>> Enter the slot number: 18
Now blinking path /dev/nvme1n1
>>> Enter the slot number: 17
Now blinking path /dev/nvme2n1
>>> Enter the slot number: 16
```

In this example, slots 16-18 represent all NVMe-capable drive slots on the server. The tool fails if it detects that you are trying to map more slots than the actual number of NVMe-capable drive slots. The EDF is written to `/usr/lpp/mmfs/data/gems/`, and it must be copied to all nodes with the same NVMe drive topology.

Verifying the Enclosure Descriptor File

Note: Ensure that you do not use any command to corrupt the data on the disk if the recovery group is already created.

The **ecedrivemapping** command can be used to summarize the current enclosure descriptor file as follows:

```
# ecedrivemapping --mode nvme –report
```

To check the slot to bus mapping from the report above, do the following steps:

1. Check the slot location with:

   ```
   # tslsenclslot -ad| mmyfields -s slot SlotHandle LocationCode Devices| grep gems | awk
   '{print "location: "$2" device: "$3}'
   location: SERIALNUMBER-SLOTX device: /dev/nvmeAn1
   location: SERIALNUMBER-SLOTY device: /dev/nvmeCn1
   ...
   ```

   This command must print the location codes of the correctly configured slots.

   The serial number `SERIALNUMBER` can be checked by the following command:

   ```
   # dmidecode -s system-serial-number
   ```

   The SLOTX is the physical disk location on the server defined by mapping tool.

2. For each disk controller, issue a **dd** read command to its corresponding block device:

   ```
   # dd if=/dev/nvmeAn1 of=/dev/null bs=1M count=10000 skip=1000
   ```

   Because the drives must be formatted with a single namespace, controller `nvmeA` corresponds to block device `/dev/nvmeAn1`. From the above report, if you ran this command on `/dev/nvmeAn1`, you must expect to see the activity light in slot SLOTX light up.

Mapping LMR disk location

IBM Storage Scale Erasure Code Edition uses **storcli** (LSI HBA) or **perccli** (DELL HBA) command to mapping SAS disk slot locations.

The **ecedrivemapping** command assists the user in generating a file that contains a server's disk layout.

Before you start, ensure that the following conditions are met:

- The server that you designate for mapping drives is populated with the lmr disks you want to map.
- The **storcli** (LSI HBA) or **perccli** (DELL HBA) command is installed on your selected server node.

Note: For Dell PERC RAID controller, **perccli** command is a substitute of **storcli** command.

Mapping lmr disks

The creation of a properly formatted `slotmap.yaml` file is produced by the **ecedrivemapping** command.

```
ecedrivemapping [-h] [--mode {nvme,lmr}]
                              [--slotrange {0-MAX_SLOT} {0-MAX_SLOT}] [--report]
                              [--force] [--version {1,2}]
```

For example, to create a `slotmap.yaml` file for lmr disks in slots 5-7, run the following command:

```
# ecedrivemapping  --mode lmr --slotrange 5 7
```

Note:

- If the `--slotrange` argument is not specified, the slot range is immediately requested upon execution of the **ecedrivemapping** command.
- If the `--mode` argument is not specified, the slot range is applied to NVMe and lmr disks.

For each lmr disk found, the identifier LED flashes to display the physical slot location of the lmr disk.

```
>>> 3 Lmr drives were detected
Now blinking path /c0/e134/s5
>>> Enter the slot number: 5
Now blinking path /c0/e134/s6
>>> Enter the slot number: 6
Now blinking path /c0/e134/s7
>>> Enter the slot number: 7
```

The slots 5-7 represent the physical slot locations that the corresponding lmr disks are mapped too. A `slotmap.yaml` file is generated and written to `/usr/lpp/mmfs/data/gems/`. If a `slotmap.yaml` file exists, it is saved as a backup file while the newly generated `slotmap.yaml` file is used.

Verifying slotmap.yaml file

The **ecedrivemapping** command can be used to summarize the current `slotmap.yaml` file as follows:

```
# ecedrivemapping --mode lmr --report
Displaying current storcli-slot to machine-slot map file:
------------------------------------------
Controller: 0
------------------------------------------
  storcli-slot : 5 => machine-slot : 5
  storcli-slot : 6 => machine-slot : 6
  storcli-slot : 7 => machine-slot : 7
```

The storcli-slot identifier contains the disks that are reported by the **storcli** command. The machine-slot identifier contains physical slot locations where the disks belong.

For example, the line:

```
storcli-slot : 5 => machine-slot : 5
```

means the disk in slot 5 reported by the **storcli** command is mapped to the physical disk slot location 5.

You can check the `slotmap.yaml` file in the `/usr/lpp/mmfs/data/gems/` directory to see whether it matches the following summary output:

```
# cat /usr/lpp/mmfs/data/gems/slotmap.yaml
controllers:
    - controller: 0
      eids: 134
      - eidx:
        slots:
        - {storcli-slot: 5,  machine-slot: 5}
        - {storcli-slot: 6,  machine-slot: 6}
        - {storcli-slot: 7,  machine-slot: 7}
```

The storcli-slots are properly mapped to the machine-slots and are categorized under controller 0. The “eids” represents the enclosure ID number that is verified by the **`storcli`** command as follows:

Use the **`tslsenclslot -ad`** command for further verification.

```
# tslsenclslot -ad | mmyfields LocationCode Devices LogicalUnits DiskSerial
J1005749-5 /dev/sda naa.5000C500B8620CCF WFJ0GBST0000E843NRY0
J1005749-6 /dev/sdc naa.5000C500B862FAFB WFJ0GWSD0000J746RSZ9
J1005749-7 /dev/sdb naa.5000C500B8632787 WFJ0GVQA0000E8447RZ5
```

The output verifies that you have three disks that belong to slots 5-7 with corresponding device names and serial numbers.

The serial numbers of the three disks can be checked with the **`storcli`** command that displays a standard output as follows.

```
# /opt/MegaRAID/storcli/storcli64 /call/eall/sall show all j | grep -E "SN|Detailed Information"
"Drive /c0/e134/s5 - Detailed Information" : {
                "SN" : "WFJ0GBST0000E843NRY0",
"Drive /c0/e134/s6 - Detailed Information" : {
                "SN" : "WFJ0GWSD0000J746RSZ9",
"Drive /c0/e134/s7 - Detailed Information" : {
                "SN" : "WFJ0GVQA0000E8447RZ5"
```

“Drive /cX/eY/sN” displays the controller (cX), enclosure ID (eY), and slot number (sN). You can verify the serial numbers of slots 5-7 match with the outputs of the **`tslsenclslot`** and **`storcli`** commands.

Manual authentication of disks and their physical slot locations can be achieved with **`storcli start locate`** and **`strocli stop locate`** commands.

```
# /opt/MegaRAID/storcli/storcli64 /c0/e134/s5 start locate
CLI Version = 007.0504.0000.0000 Nov 22, 2017
Operating system = Linux 3.10.0-1127.el7.x86_64
Controller = 0
Status = Success
Description = Start Drive Locate Succeeded.
```

```
# /opt/MegaRAID/storcli/storcli64 /c0/e134/s5 stop locate
CLI Version = 007.0504.0000.0000 Nov 22, 2017
Operating system = Linux 3.10.0-1127.el7.x86_64
Controller = 0
Status = Success
Description = Stop Drive Locate Succeeded.
```

The **`storcli locate`** command is used to turn the identifier light of a disk on or off. You can turn on the identifier light of a specific disk with an execution of the “/cX/eY/sN” **`start locate`**, where cX, eY, and sN represents the controller, enclosure ID, and slot number. After you verify the physical slot location of the disk, turn off the identifier light with “/cX/eY/sN” **`stop locate`**.

Chapter 5. Uninstalling IBM Storage Scale Erasure Code Edition

IBM Storage Scale Erasure Code Edition maintains a number of files that contain configuration and file system-related data. Because these files are critical for the proper functioning of IBM Storage Scale Erasure Code Edition and that they must be preserved across releases, they are not automatically removed when you uninstall IBM Storage Scale Erasure Code Edition.

Follow these steps if you do not intend to use IBM Storage Scale Erasure Code Edition on any of the nodes in your cluster and you want to remove all traces of IBM Storage Scale Erasure Code Edition.

⚠ **Attention:** After following these steps and manually removing the configuration and file system-related information, you will permanently lose access to all data of your current IBM Storage Scale Erasure Code Edition cluster.

1. List all GPFS file systems that are mounted in the cluster by issuing the following command.

   ```
   mmlsmount all -L
   ```

2. If there are file systems that are mounted, unmount all GPFS file systems in the cluster by issuing the following command.

   ```
   mmumount all -a
   ```

3. Verify whether all GPFS file systems in the cluster are unmounted by issuing the following command.

   ```
   mmlsmount all -L
   ```

4. Remove GPFS file systems by issuing the following command for each file system in the cluster.

   ```
   mmvdisk filesystem delete
   ```

5. Verify that all GPFS file systems are removed by issuing the following command.

   ```
   mmvdisk filesystem list
   ```

6. Remove vdisk sets by issuing the following command for each vdisk set in the cluster.

   ```
   mmvdisk vdiskset delete
   ```

7. Undefine vdisk sets by issuing the following command for each vdisk set in the cluster.

   ```
   mmvdisk vdiskset undefine
   ```

8. Verify that all vdisk sets are removed and undefined by issuing the following command.

   ```
   mmvdisk vdiskset list
   ```

9. Remove recovery groups by issuing the following command for each recovery group in the cluster.

   ```
   mmvdisk recoverygroup delete
   ```

10. Verify that all recovery groups are removed by issuing the following command.

    ```
    mmvdisk recoverygroup list
    ```

11. Unconfigure all node classes by issuing the following command for each node class in the cluster.

    ```
    mmvdisk server unconfigure
    ```

12. Delete all node classes by issuing the following command for each node class in the cluster.

    ```
    mmvdisk nodeclass delete
    ```

13. Verify that all node classes are removed by issuing the following command.

    ```
    mmvdisk nodeclass list
    ```

14. Shut down GPFS on all nodes in the cluster by issuing the following command.

    ```
    mmshutdown -a
    ```

15. Uninstall IBM Storage Scale Erasure Code Edition packages by issuing the following commands on each node.

    a. List IBM Storage Scale Erasure Code Edition packages and check the list has all IBM Storage Scale Erasure Code Edition installed packages by using the following command:

    ```
    rpm -qa|grep "^gpfs"
    ```

    b. If the list is correct, remove the packages by using the following command:

    ```
    rpm -qa|grep "^gpfs"|xargs rpm -e
    ```

16. Remove the `/var/mmfs` and `/usr/lpp/mmfs` directories on each node in the cluster.
17. Remove all files whose names start with mm from the `/var/adm/ras` directory on each node in the cluster.
18. Remove the `/tmp/mmfs` directory and its content on each node, if present.

Note: For information on uninstalling components such as GPFS clients, performance monitoring, management GUI, and Cloud services, see the *Steps to permanently uninstall GPFS* topic in the *IBM Storage Scale: Concepts, Planning, and Installation Guide*.

Chapter 6. Creating an IBM Storage Scale Erasure Code Edition storage environment

This topic describes the procedure for creating IBM Storage Scale Erasure Code Edition storage environment.

Cluster creation

This topic describes the procedure for creating an IBM Storage Scale Erasure Code Edition cluster.

Install IBM Storage Scale Erasure Code Edition on all cluster nodes, and create an IBM Storage Scale cluster by using either the IBM Storage Scale installation toolkit or manual procedures that are documented in the *Steps for establishing and starting your IBM Storage Scale cluster* topic in the *IBM Storage Scale: Concepts, Planning, and Installation Guide*.

Assign quorum nodes, cluster manager roles, and other roles as described in the "Planning for node roles" on page 20.

Use the **`mmnetverify connectivity all`** option in the *mmnetverify* command in the *IBM Storage Scale: Command and Programming Reference Guide* to ensure that your network is configured for use by IBM Storage Scale.

IBM Storage Scale Erasure Code Edition configurations

If you are using the installation toolkit, your initial recovery group, vdisk sets, and file systems are created. In that case, the commands that are shown here might be used to add additional recovery groups to your environment.

Note: Before you try to create recovery groups, ensure that the servers that are used to create recovery groups meet the mandatory requirements. For more information, see " IBM Storage Scale Erasure Code Edition hardware requirements" on page 9.

There are six steps to configuring IBM Storage Scale Erasure Code Edition. For details on each command and the supported arguments. see the mmvdisk command topic in the *IBM Storage Scale RAID: Administration*.

1. Create a node class that contains a set of identical storage servers that belong to a single recovery group.

   ```
   mmvdisk nc create --nc <nodeclass-name>  -N <node-list>
   ```

2. To maintain quorum availability in the IBM Storage Scale cluster, exercise caution when you recycle nodes. The example below uses "--recycle one" so that nodes are recycled one at a time.

   ```
   mmvdisk server configure --nc <nodeclass-name>  --recycle one
   ```

   Note: The `--recycle one` option restarts GPFS to enable new configuration one by one. Be careful when you use the `--recycle all` option. When you use this option, the **`mmvdisk`** command asks a confirmation on the console. If you provide "yes", the command restarts all the nodes at the same time, which might cause the cluster to lose quorum or file system availability.

3. Create a recovery group:

   ```
   mmvdisk rg create --rg <rg-name>  --nc <nodeclass-name>
   ```

   Note:

After Recovery Group is created and before creating the vdisks, use the following command to check the spare disk space and follow the "Recommendations" on page 18:

```
mmvdisk rg list --da
```

You can adjust the spare disk space by using the following command if needed:

```
mmvdisk rg change --rg <rg-name> --da <da-name> --spare-pdisks N
```

4. Define one or more vdisk sets:

   ```
   mmvdisk vs define --vdisk-set <vs-name>  --rg <rg-name> --code <erasure-code>
    --block-size <bsize> --set-size <set-size>
   ```

5. Create the vdisk sets that you defined:

   ```
   mmvdisk vs create --vs <vs-name>
   ```

6. Create and mount the file system:

   ```
   mmvdisk filesystem create --file-system <fs-name> --vs <vs-name>
   mmmount <fs-name> -N <nodes-to-mount-on>
   ```

Chapter 7. Using IBM Storage Scale Erasure Code Edition for data mirroring and replication

The secondary replica with synchronous mirroring by using GPFS replication can be set up by using IBM Storage Scale Erasure Code Edition. IBM Storage Scale Erasure Code Edition starts to support this feature from 5.0.5.2 release.

For more information on basic concept of synchronous mirroring with GPFS replication, see *Data mirroring and replication* topic in the *IBM Storage Scale: Administration Guide*.

In a configuration that uses GPFS replication, a single GPFS cluster is defined over three geographically separate sites. This GPFS cluster consists of two production sites and a tiebreaker site. Each production site has a set of IBM Storage Scale Erasure Code Edition storage nodes to create recovery groups.

In each IBM Storage Scale Erasure Code Edition recovery group, vdisks created are assigned to one disk failure group. The GPFS file systems that are created on these vdisks have two disk failure groups to hold file system data and metadata. Specifying file system replication factor of 2 for both data and metadata maintains two replicated file system blocks in each disk failure group. These replicated file system blocks provide a level of data redundancy that allows each site to continue to operate independently when the other site fails.

IBM Storage Scale Erasure Code Edition has two configurations that need to be adjusted for the mirroring and replication setting:

- **nsdRAIDReadRGDescriptorTimeout**: The default value is 300 seconds. It defines how long the recovery group tries to recover the root log group in each attempt.
- **nsdRAIDMaxRecoveryRetries**: The default value is 1000 times. It defines how many times the recovery group tries to recover before the vdisk failure is reported.

The suggested values for IBM Storage Scale Erasure Code Edition configured with mirroring and replication are:

- **nsdRAIDReadRGDescriptorTimeout**: 15 - 60
- **nsdRAIDMaxRecoveryRetries**: 3 - 5

Installing a typical IBM Storage Scale Erasure Code Edition cluster of synchronous mirroring by using GPFS replication

Use the following steps to install a typical IBM Storage Scale Erasure Code Edition cluster of synchronous mirroring by using GPFS replication.

In the following example, there are three geographically separated sites: Site A, Site B, and Site C.

- Site A has a set of storage nodes (nodeA01, nodeA02, nodeA03, and nodeA04) for creating IBM Storage Scale Erasure Code Edition recovery group A.
- Site B has another set of storage nodes (nodeB01, nodeB02, nodeB03, and nodeB04) for creating IBM Storage Scale Erasure Code Edition recovery group B.
- Site C has one tiebreaker node (nodeC) as the quorum node and a general NSD disk to hold file system quorum descriptor.

1. Download IBM Storage Scale Erasure Code Edition self-extracting package from the IBM Storage Scale page on Fix Central.
2. Extract the installation package.

```
# ./Spectrum_Scale_Erasure_Code-5.x.y.z-x86_64-Linux-install --text-only
```

The installation toolkit gets extracted to the `/usr/lpp/mmfs/5.x.y.z/ansible-toolkit/` directory.

3. Change the working directory to where the installation toolkit is extracted.

```
cd /usr/lpp/mmfs/5.x.y.z/ansible-toolkit/
```

4. Specify the installer node and the setup type in the cluster definition file.

   The setup type must be "`ece`" for IBM Storage Scale Erasure Code Edition.

```
./spectrumscale setup -s InstallerNodeIP -st ece
```

5. Add nodeC as the quorum node for IBM Storage Scale Erasure Code Edition in the cluster definition file.

```
./spectrumscale node add nodeC -q -g -a
```

6. Add nodeA01, nodeA02, nodeB01, and nodeB02 as the scale-out and quorum nodes for IBM Storage Scale Erasure Code Edition in the cluster definition file.

```
./spectrumscale node add NodeName -so -q
```

7. Add nodeA03, nodeA04, nodeB03, and nodeB04 as the scale-out node for IBM Storage Scale Erasure Code Edition in the cluster definition file.

```
./spectrumscale node add NodeName -so
```

8. Check the list of nodes that are specified in the cluster definition file and the respective node designations.

```
./spectrumscale node list
```

   A sample output is as follows:

```
[ INFO  ] List of nodes in current configuration:
[ INFO  ] [Installer Node]
[ INFO  ] 192.168.0.1
[ INFO  ]
[ INFO  ] [Cluster Details]
[ INFO  ] No cluster name configured
[ INFO  ] Setup Type: Erasure Code Edition
[ INFO  ]
[ INFO  ] [Extended Features]
[ INFO  ] File Audit Logging : Disabled
[ INFO  ] Watch Folder    : Disabled
[ INFO  ] Management GUI    : Enabled
[ INFO  ] Performance Monitoring    : Enabled
[ INFO  ] Callhome    : Disabled
[ INFO  ]
[ INFO  ] [Extended Features]
[ INFO  ] File Audit logging     : Enabled
[ INFO  ] Watch folder           : Disabled
[ INFO  ] Management GUI         : Disabled
[ INFO  ] Performance Monitoring : Disabled
[ INFO  ] Callhome               : Enabled
[ INFO  ]
[ INFO  ] GPFS                                 Admin  Quorum  Manager   NSD   Protocol
GUI   Callhome Scale-out   OS   Arch
[ INFO  ] Node                                  Node   Node     Node   Server   Node
Server  Server    Node
[ INFO  ] nodeC                                  X       X
X                      rhel7  x86_64
[ INFO  ] nodeA01
X                                                 X     rhel7  x86_64
[ INFO  ] nodeA01
X                                                 X     rhel7  x86_64
[ INFO  ]
nodeA03
     X     rhel7  x86_64
[ INFO  ]
nodeA04
     X     rhel7  x86_64
[ INFO  ] nodeB01
```

```
X                                                 X     rhel7  x86_64
[ INFO  ] nodeB02
X                                                 X     rhel7  x86_64
[ INFO  ]
nodeB03
     X     rhel7  x86_64
[ INFO  ]
nodeB04
     X     rhel7  x86_64
[ INFO  ]
[ INFO  ] [Export IP address]
[ INFO  ] No export IP addresses configured
```

9. Define the recovery group A of site A for IBM Storage Scale Erasure Code Edition in the cluster definition file.

   ```
   ./spectrumscale recoverygroup define -rg rgA -nc ncA -N nodeA01,nodeA02,nodeA03,nodeA04
   ```

10. Define the recovery group B of site B for IBM Storage Scale Erasure Code Edition in the cluster definition file.

    ```
    ./spectrumscale recoverygroup define -rg rgB -nc ncB -N nodeB01,nodeB02,nodeB03,nodeB04
    ```

11. Perform environment prechecks before you issue the installation toolkit installation command.

    ```
    ./spectrumscale install -pr
    ```

12. Perform the installation toolkit installation procedure.

    ```
    ./spectrumscale install
    ```

13. Check the state of the cluster and of the recovery group. All nodes, rgA, and rgB must be in the active state.

    ```
    mmgetstate -a
    mmvdisk rg list
    ```

14. Create vdisksets for the file system.

    Two declustered arrays of SSD and HDD disks are used in this example.

    ```
    mmvdisk rg list --da
    ```

    A sample output is as follows:

    ```
     declustered       needs                      capacity              pdisks
    recovery group     array     service  type  total raw  free raw  free%  total  spare
    background task
    --------------  -----------  -------  ----  ---------  --------  -----  -----  -----
    ---------------
    rgA             DA1          no       HDD      24 TiB    24 TiB   100%     48      3
    inactive 0%
    rgA             DA2          no       SSD    5074 GiB  5074 GiB   100%     16      2  scrub
    4%
    rgB             DA1          no       HDD      24 TiB    24 TiB   100%     48      3
    inactive 0%
    rgB             DA2          no       SSD    5074 GiB  5074 GiB   100%     16      2  scrub
    4%
    ```

    a) Define the vdiskset for declustered array, SSD for file system metadata, and HDD for file system data, by using the following commands:

    ```
    mmvdisk vdiskset define --vs SSD01A --rg rgA --code 8+2p --bs 2M --da DA2 --nsd-usage
    metadataonly --sp system --set-size 90%
    mmvdisk vdiskset define --vs SSD01B --copy SSD01A --rg rgB
    mmvdisk vdiskset define --vs HDD01A --rg rgA --code 8+2p --bs 8M --da DA1 --nsd-usage
    dataonly --sp data --set-size 90%
    mmvdisk vdiskset define --vs HDD01B --copy HDD01A --rg rgB
    ```

    b) Create the vdisks.

```
mmvdisk vdiskset create --vs all
```

15. Create a file system with two replicas of metadata and data block.

    vdisk in rgA of node class ncA has failure group number 1, vdisk in rgB of node class ncB has failure group number 2.

    ```
    mmvdisk filesystem create --fs gpfs1 --vs SSD01A,SSD01B,HDD01A,HDD01B --fg ncA=1,ncB=2 --
    mmcrfs -T /gpfs1 -r 2 -m 2
    ```

16. Add disk of nodeC to the file system as the `descOnly` disk.

    a) Create an NSD stanza file for the disk, it has the failure group number 3.

    ```
    # cat diskC.stanza
    %nsd:
     nsd=diskDescNsd
     device=/dev/sdo
     servers=NodeC
     usage=descOnly
     failureGroup=3
     pool=system
    ```

    b) Create the NSD.

    ```
    mmcrnsd -F diskC.stanza
    ```

    c) Add this NSD into the file system.

    ```
    mmadddisk gpfs1 -F diskC.stanza
    ```

17. To avoid unexpected mounts on nodeC, create the following empty file on nodeC.

    ```
    touch /var/mmfs/etc/ignoreAnyMount.gpfs1
    ```

18. Mount the file system.

    ```
    mmmount all -a
    ```

19. Change the configurations as follows:

    ```
    # mmchconfig nsdRAIDMaxRecoveryRetries=4 -i
    # mmchconfig nsdRAIDReadRGDescriptorTimeout=60 -i
    ```

Bringing back the recovery group

When the recovery group fails after the maximum retry limit, it is not automatically recovered. In such a scenario, you must manually recover the RG after you analyze and repair the factors that caused the failure. For failover and failback steps after a disaster, see the *Steps to take after a disaster when using IBM Storage Scale replication* topic in the *IBM Storage Scale: Administration Guide*.

Use the following steps to check and bring back the RG when the system is restored after a failure.

1. Show the current recovery group state:

   ```
   # mmvdisk rg list
   recovery group  active   current or master server              service  vdisks  remarks
   --------------  -------  ------------------------------------  -------  ------  -------
   rgA             yes      nodeA01                               no           16
   rgB             no       -                                     unknown      16
   ```

2. Manually bring the recovery group back to the active state:

   ```
   # mmvdisk rg change --rg rgB --restart
   ```

The system displays an output similar to this:

```
mmvdisk: Waiting up to 5 minutes for recovery group 'rgB' to restart.
 node
number  server                                active   remarks
------  ------------------------------------  -------  -------
     4  nodeB01                               yes      serving rgB: LG001, LG005
     5  nodeB02                               yes      serving rgB: root, LG002, LG006
     8  nodeB03                               yes      serving rgB: LG004, LG008
     9  nodeB04                               yes      serving rgB: LG003, LG007
mmvdisk: Recovery group 'rgB' has been restarted.
```

Chapter 8. Deploying IBM Storage Scale Erasure Code Edition on VMware infrastructure

You can deploy IBM Storage Scale Erasure Code Edition on a VMware virtual machine.

VMware virtual machine requirements and limitations

This topic describes the requirements for storage hardware and software that can be used with IBM Storage Scale Erasure Code Edition on VMware virtual machine.

Hardware requirements and limitations

This section describes the minimum requirements and limitations to run IBM Storage Scale Erasure Code Edition on a VMware virtual machine.

For more information on minimum hardware requirements to run IBM Storage Scale Erasure Code Edition, see Minimum hardware requirements and precheck.

IBM Storage Scale Erasure Code Edition has some additional requirements on CPU, disk drive, network, and storage adapter.

Table 8. IBM Storage Scale Erasure Code Edition hardware requirements for each VMware virtual machine

Hardware	Description
CPU	16 VMware virtual CPU
SAS Data Drives	Physical drives must follow general hardware requirements and use Raw Device Mapping (RDM) with physical compatibility mode for VM local disk drive
NVMe Data Drives	Physical drives must follow general hardware requirements and use pass through enabled device as VM local PCIe NVMe device
Network Adapter	Physical network cards must follow general hardware requirements and use SR-IOV enabled device as VM network adapter
SAS Storage Adapters/Controllers	Physical SAS card must follow general hardware requirements and use VMware supported storage adapters

Software requirements

This section describes the recommended software version to run IBM Storage Scale Erasure Code Edition on VMware virtual machine.

Table 9. Recommended software version

Software	Version
VMware ESXi	VMware ESXi 7 (7.0.3 or later)
VMware vSphere Client	VMware vSphere 7 (7.0.3 or later)
Operator System	RHEL 8.2 or later
Storage Scale	5.1.4.0 or later

Limitations

This section describes the limitations.

- Each ESXi host can run only one ECE storage server VM. Deploying more than one ECE storage virtual machine on one physical server will cause data integrity issues in the event of physical server failure.
- Migrating ECE storage server VM from one ESXi host to another host is not supported.

VMware deployment architecture

This topic describes the deployment architecture of running IBM Storage Scale Erasure Code Edition on VMware.

The following figure shows the VMware deployment architecture:

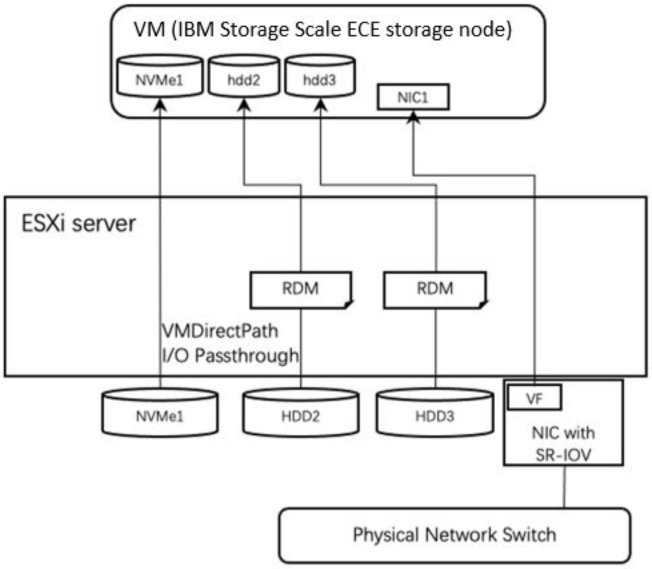


Figure 2. IBM Storage Scale Erasure Code Edition on the VMware deployment architecture

In the IBM Storage Scale Erasure Code Edition on the VMware deployment architecture, each VMware ESXi server provides a VM to run as the IBM Storage Scale ECE storage node. VMware ESXi server local HDD and NVMe disk drives are configured to the VM for IBM Storage Scale ECE. Virtual function from Single Root I/O Virtualization (SR-IOV) enabled network device is configured to the ECE storage node for the ECE network connection.

Note: Physical network used as an ECE storage interface should not be used for other purpose. Any other network access requirements need to be set up on other network interface.

Installing IBM Storage Scale Erasure Code Edition on VMware virtual machine

This topic describes the procedure for installing IBM Storage Scale Erasure Code Edition on VMware virtual machine and provisioning a new virtual machine.

Secure boot

Note: IBM Storage Scale does not support secure boot.

To disable the secure boot for the ECE storage node VM, select the **VM Option** > **Boot Option**, and deselect **Secure boot Option**.

CPU and memory

For CPU and memory requirements, see “VMware virtual machine requirements and limitations” on page 49.

NVMe disk

IBM Storage Scale Erasure Code Edition on VMware supports use of `VMDirectPath` I/O passthrough to configure NVMe disk drive to IBM Storage Scale Erasure Code Edition storage node VM.

Note:

Before enabling the passthrough feature for an NVMe storage adapter, add a hardware label for each NVMe device for IBM Storage Scale Erasure Code Edition.

Before configuring NVMe disk drive to IBM Storage Scale ECE, see “Hardware checklist” on page 12 to prepare NVMe disk drives, such as disabling volatile cache on NVMe drives, formatting NVMe drives in proper format and so on.

Configuring an NVMe disk for IBM Storage Scale Erasure Code Edition

Use the following steps to configure an NVMe disk for IBM Storage Scale Erasure Code Edition:

1. Get NVMe device name.

   In the **Storage Adapters** page, select the NVMe storage adapter, in **Device** property page, get the Name of the storage adapter device.

   Use the **`esxcli`** command to get detailed information of this NVMe device and property.

   ```
   [root@localhost:~] esxcli hardware pci list
   0000:da:00.0
   Address: 0000:da:00.0
      Segment: 0x0000
      Bus: 0xda
      Slot: 0x00
      Function: 0x0
      VMkernel Name: vmhba3
      Vendor Name: Samsung Electronics Co Ltd
      Device Name: Thinksystem U.2 PM983 NVMe SSD
      Configured Owner: VMkernel
      Current Owner: VMkernel
      Vendor ID: 0x144d
      Device ID: 0xa808
      SubVendor ID: 0x1d49
      SubDevice ID: 0x403b
      Device Class: 0x0108
      Device Class Name: Non-Volatile memory controller
      Programming Interface: 0x02
      Revision ID: 0x00
      Interrupt Line: 0x0b
      IRQ: 255
      Interrupt Vector: 0x00
      PCI Pin: 0x00
      Spawned Bus: 0x00
      Flags: 0x3001
      Module ID: 90
      Module Name: nvme_pcie
      Chassis: 0
      Physical Slot: 73
      Slot Description: Bay 9
      Device Layer Bus Address: s00000049.00
      Passthru Capable: true
      Parent Device: PCI 0:215:2:0
      Dependent Device: PCI 0:218:0:0
      Reset Method: Function reset
      FPT Sharable: true
      NUMA Node: 1
      Extended Device ID: 0
      Extended Device Name:
   [root@localhost:~] esxcli nvme device namespace list -A vmhba3
   Namespace ID  Status  Device Name
   ------------  ------  -----------
   ```

```
           1  Active  eui.343731304d70036700253845000000001
[root@localhost:~]
```

In above example, for the NVMe storage adapter *vmhba3*, the PCI address is *00:DA:00* and the NVMe device name of its namespace is *eui.343731304d70036700253845000000001*.

Use the ID in the device name *343731304d70036700253845000000001* as the hardware label for this NVMe disk.

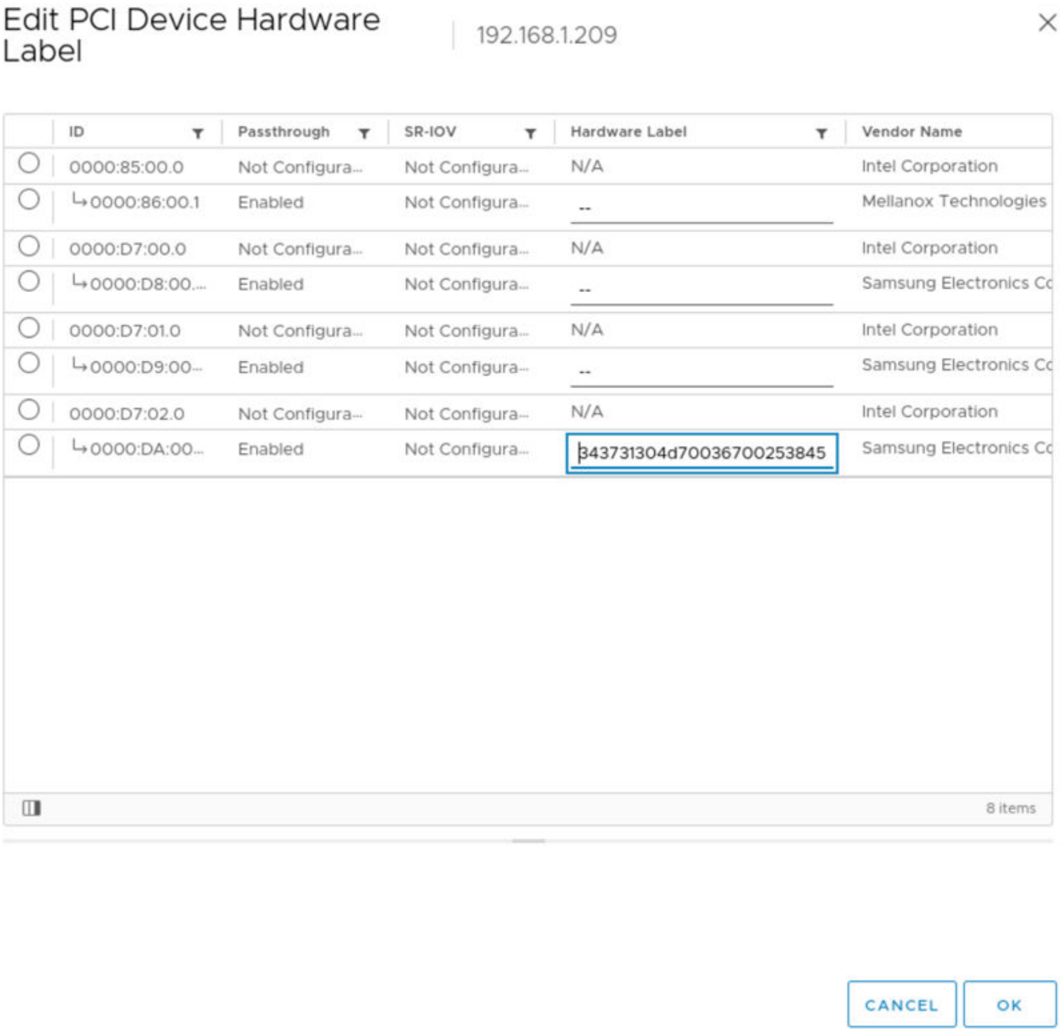
2. Toggle passthrough for the NVMe disk.

   Click the **Configure** tab, on the **PCI Devices** page, click **ALL PCI DEVICE**, and use the PCI address obtained from step 1 to identify the PCI device from the ID column. Select the PCI device and click **TOGGLE PASSTHROUGH** button to enable passthrough feature of this NVMe disk.

   Note: After the passthrough feature has been enabled for this NVMe disk, it will be listed in the **PASSTHROUGH ENABLED DEVICES** list.

3. Set the hardware label.

   Use the PCI address obtained from step 1 to identify the PCI device from the **ID** column. Select the PCI device and click the **HARDWARE LABEL** button and set the hardware label defined in step 1.



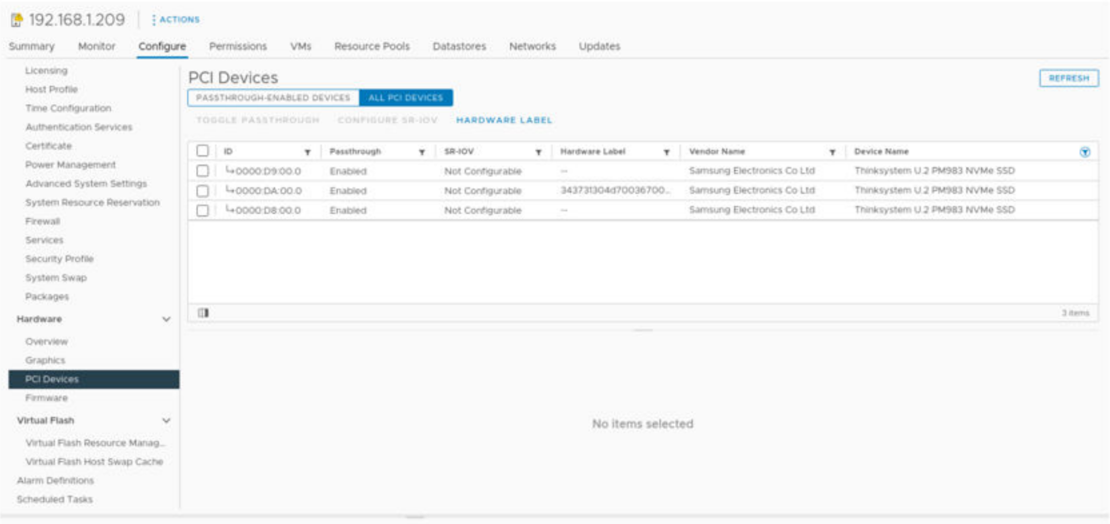
Edit PCI Device Hardware Label | 192.168.1.209

	ID	Passthrough	SR-IOV	Hardware Label	Vendor Name
○	0000:85:00.0	Not Configura...	Not Configura...	N/A	Intel Corporation
○	↳0000:86:00.1	Enabled	Not Configura...	--	Mellanox Technologies
○	0000:D7:00.0	Not Configura...	Not Configura...	N/A	Intel Corporation
○	↳0000:D8:00...	Enabled	Not Configura...	--	Samsung Electronics Co
○	0000:D7:01.0	Not Configura...	Not Configura...	N/A	Intel Corporation
○	↳0000:D9:00...	Enabled	Not Configura...	--	Samsung Electronics Co
○	0000:D7:02.0	Not Configura...	Not Configura...	N/A	Intel Corporation
○	↳0000:DA:00...	Enabled	Not Configura...	343731304d70036700253845	Samsung Electronics Co

8 items

Figure 3. Set hardware label for NVMe device

After the hardware label is set, it will be listed in the **Hardware Label** column in the NVMe PCI device.



PCI Devices

ID	Passthrough	SR-IOV	Hardware Label	Vendor Name	Device Name
0000:D9:00.0	Enabled	Not Configurable	--	Samsung Electronics Co Ltd	Thinksystem U.2 PM983 NVMe SSD
0000:DA:00.0	Enabled	Not Configurable	343731304d70036700...	Samsung Electronics Co Ltd	Thinksystem U.2 PM983 NVMe SSD
0000:D8:00.0	Enabled	Not Configurable	--	Samsung Electronics Co Ltd	Thinksystem U.2 PM983 NVMe SSD

3 items

No items selected

Figure 4. NVMe device with a hardware label

Adding a PCI passthrough NVMe disk into VM

In VM config, select **Edit Settings**, and then **Add New Device**, select **PCI Device**, add NVMe PCI device into VM. For more details, see: https://docs.vmware.com/en/VMware-vSphere/7.0/com.vmware.vsphere.vm_admin.doc/GUID-5B3CAB26-5D06-4A99-92A0-3A04C69CE64B.html.

Note:

Features such as hot adding and removal of virtual device are not available for PCI device passthrough. For more details, see: https://docs.vmware.com/en/VMware-vSphere/7.0/com.vmware.vsphere.networking.doc/GUID-BF2770C3-39ED-4BC5-A8EF-77D55EFE924C.html.

IBM Storage Scale ECE needs special procedure for replacing the failed NVMe disk which is described in "Replacing an NVMe disk" on page 56.

HDD disk

IBM Storage Scale Erasure Code Edition on VMware supports use of Raw Device Mapping (RDM) to add an HDD or SSD disk drive to the VM.

See *Add an RDM Disk to a Virtual Machine* page in https://docs.vmware.com/en/VMware-vSphere/7.0/com.vmware.vsphere.vm_admin.doc/GUID-4236E44E-E11F-4EDD-8CC0-12BA664BB811.html to add HDD device to IBM Storage Scale ECE storage node VM.

Use the physical compatibility mode to add an RDM HDD disk to the ECE storage node VM.

Note: Current IBM Storage Scale Erasure Code Edition does not support use of a hard drive created from the VMware datastore.

Network

IBM Storage Scale Erasure Code Edition supports to use virtual function of SR-IOV enabled physical network adapter.

Storage Scale ECE uses NIC1 for network connection as shown in Figure 2 on page 50.

Note:

It is not supported to use virtual network interface connected to VMware virtual switch for Storage Scale ECE network connection.

See https://docs.vmware.com/en/VMware-vSphere/7.0/com.vmware.vsphere.networking.doc/GUID-BEA364D2-7A9D-44E6-BA2D-92A07516B6B9.html to enable SR-IOV on a host physical adapter.

See https://docs.vmware.com/en/VMware-vSphere/7.0/com.vmware.vsphere.networking.doc/GUID-EE03DC6F-32CA-42EF-98FC-12FDE06C0BE0.html to configure a virtual machine to use a virtual function of SR-IOV enabled network adapter.

Precheck on the VMware virtual machine

When the IBM Storage Scale VMs are ready, you can download and run the IBM Storage Scale ECE precheck tool from the repository: https://github.com/IBM/SpectrumScaleTools/.

Installing an ECE cluster by using the IBM installation toolkit on the VMware virtual machines

See "Installing IBM Storage Scale Erasure Code Edition by using the installation toolkit" on page 31 to create an IBM Storage Scale ECE cluster by using the IBM Storage Scale installation toolkit.

Note:

Disk slot mapping is not required in VMware Virtual Machine. Skip Step 8 in "Installing IBM Storage Scale Erasure Code Edition by using the installation toolkit" on page 31.

After finishing step 10, if there are multiple Declustered Array (DA) created in the recovery group, current installation toolkit will not create vdiskset and file system properly. Ignore steps 11 through 14 in "Installing IBM Storage Scale Erasure Code Edition by using the installation toolkit" on page 31. See " IBM Storage Scale Erasure Code Edition configurations" on page 41 to use the **mmvdisk** command to create vdiskset and file system.

Administering ECE on a VMware virtual machine cluster

This topic describes the procedure to administer ECE on a VMware virtual machine.

Managing physical disks

Note:

- IBM Storage Scale ECE on VMware does not depend on a disk slot map to identify a drive location.
- To replace a broken disk, it is not necessary to insert the new disk into the same slot as the broken one.

Replacing HDD

Use the following steps to replace HDD:

1. Identify the disks with problems. Issue the following command to check the current disks that have a problem:

```
[root@ece09vm1 ~]# mmvdisk pdisk list --rg all --not-ok

                              declustered
recovery group  pdisk            array     paths  capacity  free space  FRU (type)
state
--------------  ------------  -----------  -----  --------  ----------  ---------------
-----
rg_1            n001p008      DA1              0  7452 GiB    6772 GiB  00YK034
missing/draining

[root@ece09vm1 ~]#
```

   You can get more information of this disk.

```
[root@ece09vm1 ~]# mmvdisk pdisk list --rg rg_1 --pdisk n001p008 -L
pdisk:
   replacementPriority = 1000
   name = "n001p008"
   device = ""
   recoveryGroup = "rg_1"
   declusteredArray = "DA1"
   state = "missing/drained"
   internalState = 00048.1c0
   capacity  = 8001524072448
   freeSpace = 7992934137856
   fru = "00YK034"
   location = ""
   WWN = "naa.50000399587B6395"
   server = "ece11vm1-hs.gpfs.net"
   reads = 0
   writes = 302
   bytesReadInGiB = 0.000
   bytesWrittenInGiB = 0.143
   IOErrors = 0
   IOTimeouts = 0
   mediaErrors = 0
   checksumErrors = 0
   pathErrors = 0
   relativePerformance = 1.000
   bitErrorRate = 0.000e+00
   rgIndex = 60
   userLocation = ""
   hardware = "LENOVO MG06SCA800E TG64 49P0A001FK0F"
   hardwareType = Rotating 7200
   nPaths = 0 active 0 total
   sedSupported =


[root@ece09vm1 ~]#
```

Note: WWN is the unique ID to identify this physical disk. You can get help from your VMware admin to replace this physical disk.

2. Add a new RDM HDD into this VM.

   You can use the **mmgetpdisktopology** command to identify a new device.

```
[root@ece09vm1 ~]# mmgetpdisktopology
name:scsitype:device:nsdid:wwid:location:hctl:altdev:vendor:product:firmware:serial:fru:size:
adapters:expanders:enclosure:portcards:slot:sasaddrs:sesdev:
nvme1n1:0:/dev/nvme1n1:0B64A8C063F8223F|n001p009|rg_1|0A64A8C063F8220C|DA2||0||
9.0:eui.343731304BB00229002538460000000l:0000%3a1b%3a00.0:[N.1.0.1]:/dev/
nvme1:144D:MZQLB1T9HAJR-000V3:    CK32:S471NF0KB00229::1920383410176::::::
nvme0n1:0:/dev/nvme0n1:0B64A8C063F82240|n001p010|rg_1|0A64A8C063F8220C|DA2||0||
9.0:eui.343731304D700376002538450000000l:0000%3a04%3a00.0:[N.0.0.1]:/dev/
nvme0:144D:MZQLB1T9HAJR-000V3:    CK32:S471NE0M700376::1920383410176::::::
sdd:0:/dev/sdd:0B64A8C063F8220E|n001p001|rg_1|0A64A8C063F8220C|DA1||7200||
9.0:naa.500003995849AFE1::[1.0.1.0]:/dev/
sg3:LENOVO:MG06SCA800E:TG64:49F0A004FK0F:00YK034:8001563222016::500062b204c6334e.000000000000
0000::::5000
03995849afe2.500003995849afe3:
sdh:0:/dev/sdh:0B64A8C063F8220F|n001p002|rg_1|0A64A8C063F8220C|DA1||7200||
9.0:naa.500003995849AFF5::[1.0.5.0]:/dev/
sg7:LENOVO:MG06SCA800E:TG64:49F0A005FK0F:00YK034:8001563222016::500062b204c63348.000000000000
0000::::500003995849aff6.500003995849aff7:
sdf:0:/dev/sdf:0B64A8C063F82210|n001p003|rg_1|0A64A8C063F8220C|DA1||7200||
9.0:naa.500003995849B0B9::[1.0.3.0]:/dev/
sg5:LENOVO:MG06SCA800E:TG64:49F0A006FK0F:00YK034:8001563222016::500062b204c6334b.000000000000
0000::::500003995849b0ba.500003995849b0bb:
sdc:0:/dev/sdc:0B64A8C063F82211|n001p004|rg_1|0A64A8C063F8220C|DA1||7200||
9.0:naa.500003995863233D::[1.0.0.0]:/dev/
sg2:LENOVO:MG06SCA800E:TG64:49K0A024FK0F:00YK034:8001563222016::500062b204c63349.000000000000
0000::::500003995863233e.500003995863233f:
sde:0:/dev/sde:0B64A8C063F82212|n001p005|rg_1|0A64A8C063F8220C|DA1||7200||
9.0:naa.50000399586B104D::[1.0.2.0]:/dev/
sg4:LENOVO:MG06SCA800E:TG64:49M0A019FK0F:00YK034:8001563222016::500062b204c63342.000000000000
0000::::50000399586b104e.50000399586b104f:
sdi:0:/dev/sdi:0B64A8C063F82213|n001p006|rg_1|0A64A8C063F8220C|DA1||7200||
9.0:naa.50000399587A1D89::[1.0.6.0]:/dev/
sg8:LENOVO:MG06SCA800E:TG65:49X0A00EFK0F:00YK034:8001563222016::500062b204c63341.000000000000
0000::::50000399587a1d8a.50000399587a1d8b:
sdg:0:/dev/sdg:0B64A8C063F82214|n001p007|rg_1|0A64A8C063F8220C|DA1||7200||
9.0:naa.50000399587B63C5::[1.0.4.0]:/dev/
sg6:LENOVO:MG06SCA800E:TG64:49P0A002FK0F:00YK034:8001563222016::500062b204c6334d.000000000000
0000::::50000399587b63c6.50000399587b63c7:
sdb:0:/dev/sdb:||||||7200||9.0:naa.50000399588024CD::[0.0.1.0]:/dev/
sg1:LENOVO:MG06SCA800E:TG64:49P0A00GFK0F:00YK034:8001563222016::500062b204c6334c.000000000000
0000::::50000399588024ce.50000399588024cf:
host:-1:ece09vm1:5.1.0.0::1:x86_64:ece09vm1-hs.gpfs.net::VMware7%2c1::VMware-42 0c 08 dd 66
36 5d f5-c5 af 1c 40 ec 59 5d
11:::S0000%3a00%3a0f.0ASlotSSlotA0000%3a00%3a11.0S0000%3a00%3a12.0ASlotSSlotA0000%3a00%3a14.0
```

```
:::::
name:scsitype:device:nsdid:wwid:location:hctl:altdev:vendor:product:firmware:serial:fru:size:
adapters:expanders:enclosure:portcards:slot:sasaddrs:sesdev:
[root@ece09vm1 ~]#
```

In the above example, there is no *nsdid* created for /dev/sdb which is the new added disk.

Note: The new disk must have the same type and capacity as the disk replaced.

3. Prepare a stanza file with the following format.

```
# cat pdisk.stz
%pdisk: pdiskName=n001p008 device=//ece09vm1-hs/dev/sdb da=DA1
```

- **N001p008** is the pdisk name identified from step 1.
- **ece09vm1-hs** is the node name of the pdisk.
- **/dev/sdb** is the new device name.
- **DA1** is the disk array of the pdisk identified in step 1.

4. Use the new device to replace the broken disk.

```
[root@ece09vm1 ~]# mmaddpdisk rg_1 -F pdisk.stz --replace --class-mode vdisk
The following pdisks will be formatted on node ece11vm1.gpfs.net:
    //ece09vm1-hs/dev/sdb
[root@ece09vm1 ~]#
```

5. Confirm that all pdisks are OK.

```

[root@ece09vm1 ~]# mmvdisk pdisk list --rg all --not-ok
mmvdisk: All pdisks are ok.
[root@ece09vm1 ~]#
```

Replacing an NVMe disk

Use the following steps to replace NVMe disk:

1. Identify the disks with problems. Issue the following command to check the current disks that have a problem:

```
[root@ece09vm1 ~]# mmvdisk pdisk list --rg all --not-ok

                              declustered
recovery group  pdisk            array     paths  capacity  free space  FRU (type)
state
--------------  ------------  -----------  -----  --------  ----------  ---------------
-----
rg_1            n001p010      DA2              0  1788 GiB    1784 GiB  MZQLB1T9HAJR-00
missing/drained
[root@ece09vm1 ~]#
```

You can get more information of this pdisk.

```
[root@ece09vm1 ~]# mmvdisk pdisk list --rg rg_1 --pdisk n001p010 -L
pdisk:
   replacementPriority = 1000
   name = "n001p010"
   device = ""
   recoveryGroup = "rg_1"
   declusteredArray = "DA2"
   state = "missing/drained"
   internalState = 00048.1c0
   capacity  = 1919850381312
   freeSpace = 1915555414016
   fru = "MZQLB1T9HAJR-00"
   location = ""
   WWN = "eui.343731304D7003760025384500000001"
   server = "ece11vm1-hs.gpfs.net"
   reads = 33100
   writes = 5295955
   bytesReadInGiB = 64.648
   bytesWrittenInGiB = 284.034
   IOErrors = 0
   IOTimeouts = 0
   mediaErrors = 0
   checksumErrors = 0
   pathErrors = 2
   relativePerformance = 1.000
   bitErrorRate = 0.000e+00
   rgIndex = 49
   userLocation = "Enclosure VMware420c08dd66365df5c5af1c40ec595d11 Drive 10"
   hardware = "144D MZQLB1T9HAJR-000V3     CK32 S471NE0M700376"
   hardwareType = NVMe
   nPaths = 0 active 0 total
   sedSupported =


[root@ece09vm1 ~]#
```

The WWN listed here is the NVMe namespace guid. In section "Configuring an NVMe disk for IBM Storage Scale Erasure Code Edition" on page 51, the guid has been set to the PCI device "hardware label". You can use this guid to get help from VMware admin to identify this NVMe disk and replace it.

2. Add a new NVMe HDD device into VM.

   See "Configuring an NVMe disk for IBM Storage Scale Erasure Code Edition" on page 51 to configure and add a new NVMe device into VM.

   Note: Due to VMware limitation, you have to shut down the VM to add the new NVMe device into VM. Follow steps 3 through 5 in "Manual online upgrade of IBM Storage Scale Erasure Code Edition" on page 65 to shut down the IBM Storage Scale ECE on the VM before shutting down the VM.

   After the NVMe device has been attached to the VM, Start the VM, and then follow step 7 in "Manual online upgrade of IBM Storage Scale Erasure Code Edition" on page 65 to start up GPFS.

3. Issue the **mmgetpdisktopology** command to identify the new added disk to the NVMe device.

```
[root@ece09vm1 ~]# mmgetpdisktopology
name:scsitype:device:nsdid:wwid:location:hctl:altdev:vendor:product:firmware:serial:fru:size:
adapters:expanders:enclosure:portcards:slot:sasaddrs:sesdev:
nvme1n1:0:/dev/nvme1n1:0B64A8C063F8223F|n001p009|rg_1|0A64A8C063F8220C|DA2||0||
9.0:eui.343731304BB00229002538460000001:0000%3a1b%3a00.0:[N.1.0.1]:/dev/
nvme1:144D:MZQLB1T9HAJR-000V3:    CK32:S471NF0KB00229::1920383410176::::::
nvme0n1:0:/dev/nvme0n1:||||||0||9.0:eui.343731304D70036700253845000000001:0000%3a04%3a00.0:
[N.0.0.1]:/dev/nvme0:144D:MZQLB1T9HAJR-000V3:    CK32:S471NE0M700367::1920383410176::::::
sdd:0:/dev/sdd:0B64A8C063F8220E|n001p001|rg_1|0A64A8C063F8220C|DA1||
7200||:naa.500003995849AFE1::[1.0.1.0]:/dev/
sg3:LENOVO:MG06SCA800E:TG64:49F0A004FK0F:00YK034:8001563222016::500062b204c6334e.000000000000
0000::::500003995849afe2.500003995849afe3:
sdh:0:/dev/sdh:0B64A8C063F8220F|n001p002|rg_1|0A64A8C063F8220C|DA1||
7200||:naa.500003995849AFF5::[1.0.5.0]:/dev/
sg7:LENOVO:MG06SCA800E:TG64:49F0A005FK0F:00YK034:8001563222016::500062b204c63348.000000000000
0000::::500003995849aff6.500003995849aff7:
sdf:0:/dev/sdf:0B64A8C063F82210|n001p003|rg_1|0A64A8C063F8220C|DA1||
7200||:naa.500003995849B0B9::[1.0.3.0]:/dev/
sg5:LENOVO:MG06SCA800E:TG64:49F0A006FK0F:00YK034:8001563222016::500062b204c6334b.000000000000
0000::::500003995849b0ba.500003995849b0bb:
sdc:0:/dev/sdc:0B64A8C063F82211|n001p004|rg_1|0A64A8C063F8220C|DA1||
7200||:naa.500003995863233D::[1.0.0.0]:/dev/
sg2:LENOVO:MG06SCA800E:TG64:49K0A024FK0F:00YK034:8001563222016::500062b204c63349.000000000000
0000::::500003995863233e.500003995863233f:
sde:0:/dev/sde:0B64A8C063F82212|n001p005|rg_1|0A64A8C063F8220C|DA1||
7200||:naa.50000399586B104D::[1.0.2.0]:/dev/
sg4:LENOVO:MG06SCA800E:TG64:49M0A019FK0F:00YK034:8001563222016::500062b204c63342.000000000000
0000::::50000399586b104e.50000399586b104f:
sdi:0:/dev/sdi:0B64A8C063F82213|n001p006|rg_1|0A64A8C063F8220C|DA1||
7200||:naa.50000399587A1D89::[1.0.6.0]:/dev/
sg8:LENOVO:MG06SCA800E:TG65:49X0A00EFK0F:00YK034:8001563222016::500062b204c63341.000000000000
0000::::50000399587a1d8a.50000399587a1d8b:
sdg:0:/dev/sdg:0B64A8C063F82214|n001p007|rg_1|0A64A8C063F8220C|DA1||
7200||:naa.50000399587B63C5::[1.0.4.0]:/dev/
sg6:LENOVO:MG06SCA800E:TG64:49P0A002FK0F:00YK034:8001563222016::500062b204c6334d.000000000000
0000::::50000399587b63c6.50000399587b63c7:
sdb:0:/dev/sdb:0B64A8C063FC4F7F|n001p008|rg_1|0A64A8C063F8220C|DA1||
7200||:naa.50000399588024CD::[0.0.1.0]:/dev/
sg1:LENOVO:MG06SCA800E:TG64:49P0A00GFK0F:00YK034:8001563222016::500062b204c6334c.000000000000
0000::::50000399588024ce.50000399588024cf:
host:-1:ece09vm1:5.1.0.0::1:x86_64:ece09vm1-hs.gpfs.net::VMware7%2c1::VMware-42 0c 08 dd 66
36 5d f5-c5 af 1c 40 ec 59 5d
11:::S0000%3a00%3a0f.0ASlotSSlotA0000%3a00%3a11.0S0000%3a00%3a12.0ASlotSSlotA0000%3a00%3a14.0
```

```
:::::
name:scsitype:device:nsdid:wwid:location:hctl:altdev:vendor:product:firmware:serial:fru:size:
adapters:expanders:enclosure:portcards:slot:sasaddrs:sesdev:
[root@ece09vm1 ~]#
```

In above example, there is no *nsdid* in /dev/nvme0n1 which is the new added NVMe disk.

Note: The new added disk must have the same format and size as the disk replaced.

4. Prepare a stanza file with the following format:

```
# cat pdisk1.stz
%pdisk: pdiskName=n001p010 device=//ece09vm1-hs/dev/nvme0n1 da=DA2
```

- **n001p010** is the pdisk name identified from step 1.
- **ece09vm1-hs** is the node name of the pdisk.
- **/dev/ nvme0n1** is the new device name.
- **DA2** is the disk array of the pdisk identified in step 1.

5. Use the new device to replace the broken disk.

```
[root@ece09vm1 ~]# mmaddpdisk rg_1 -F pdisk1.stz --replace --class-mode vdisk

The following pdisks will be formatted on node ece11vm1.gpfs.net:
    //ece09vm1-hs/dev/nvme0n1
[root@ece09vm1 ~]
```

6. Confirm that all pdisks are OK.

```
[root@ece09vm1 ~]# mmvdisk pdisk list --rg all --not-ok
mmvdisk: All pdisks are ok.
[root@ece09vm1 ~]#
```

Adding new disks into the ECE storage server

See “NVMe disk” on page 51 and “HDD disk” on page 53 to add new disk(s) into all ECE storage servers and then see “Adding new disks in the declustered array of the recovery group” on page 104 to add new disk in the declustered array of the recovery group.

Managing I/O nodes

Note: Disk slot mapping is not required on a VMware virtual machine.

Replacing an I/O node with a new node and disks

Because you need to prepare a new ECE storage node and disks, which means you need to find a new ESXi server to provide a new VM to run this as ECE storage node.

See “Installing IBM Storage Scale Erasure Code Edition on VMware virtual machine” on page 50 to provide a new VM with the same configuration with storage server node in the ECE recovery group.

See Replacing an I/O node with a new node and disks to replace an I/O node with the new provisioned node.

Replacing a broken I/O node with moving disks into a new node

Use the steps to replace a broken I/O node with moving disks into a new node:

1. Shut down the broken VM to make sure that the VM is not pingable.
2. Detach the NVMe and HDD disks for IBM Storage Scale ECE from the broken VM and add them into a new provisioned VM.

   See “Installing IBM Storage Scale Erasure Code Edition on VMware virtual machine” on page 50 to provision a new VM.

Note: The NVMe and HDD disks in the new provisioned VM are the ECE disks that are moved to the server. Ensure that the data on the disks is intact.

3. Replace the broken I/O node with the new provisioned VM. See Replacing broken I/O nodes with moving disks to new nodes for more details.

Adding a new node into an ECE recovery group

See "Installing IBM Storage Scale Erasure Code Edition on VMware virtual machine" on page 50 to provision a new VM and then see Adding new nodes to add new I/O nodes into an ECE recovery group.

Chapter 9. Upgrading IBM Storage Scale Erasure Code Edition

You can upgrade to a newer available version of IBM Storage Scale Erasure Code Edition by using the installation toolkit or by using manual steps.

Use one of the following available upgrade options depending on your requirements:

- Use the installation toolkit to do an online upgrade of your IBM Storage Scale Erasure Code Edition cluster. For more information, see "Online upgrade of IBM Storage Scale Erasure Code Edition by using the installation toolkit" on page 61.
- Use the installation toolkit to do an offline upgrade of your IBM Storage Scale Erasure Code Edition cluster. For more information, see "Offline upgrade of IBM Storage Scale Erasure Code Edition by using the installation toolkit" on page 63.
- Use the manual procedure to do an online upgrade of your IBM Storage Scale Erasure Code Edition cluster. For more information, see "Manual online upgrade of IBM Storage Scale Erasure Code Edition" on page 65.

You can verify and update the configurations after upgrading your IBM Storage Scale Erasure Code Edition cluster. For more information, see "Verifying and updating the configurations" on page 69.

Online upgrade of IBM Storage Scale Erasure Code Edition by using the installation toolkit

You can upgrade to a newer available version of IBM Storage Scale Erasure Code Edition by using the installation toolkit. For upgrading from version 5.0.4.3 or later to version 5.0.5 or later, you can use the installation toolkit for online upgrade.

For upgrading from a version earlier than 5.0.4.3 to a later version (including version 5.0.4.3 to 5.0.4.4), you can use the installation toolkit only for offline upgrade. For more information, see "Offline upgrade of IBM Storage Scale Erasure Code Edition by using the installation toolkit" on page 63.

Note: Online upgrade is only allowed when all vdisks of the recovery group have disk tolerance more than "`1 node + 1 pdisk`". Check the disk group fault tolerance by using the following command:

```
mmvdisk rg list --rg rg_name --fault-tolerance
```

1. Download IBM Storage Scale Erasure Code Edition self-extracting package from the IBM Storage Scale page on Fix Central.

   The name of IBM Storage Scale Erasure Code Edition self-extracting installation package is similar to `Spectrum_Scale_Erasure_Code-5.x.y.z-x86_64-Linux-install`.
2. Extract the installation package.

   ```
   # ./Spectrum_Scale_Erasure_Code-5.x.y.z-x86_64-Linux-install --textonly
   ```

   The installation toolkit gets extracted to the `/usr/lpp/mmfs/5.x.y.z/ansible-toolkit/` directory.

   To verify that the extracted package is of IBM Storage Scale Erasure Code Edition, go to the `/usr/lpp/mmfs/5.x.y.z/gpfs_rpms` directory and check for `gpfs.gnr*` packages.
3. Change the directory to where the installation toolkit is extracted.

   ```
   cd /usr/lpp/mmfs/5.x.y.z/ansible-toolkit/
   ```
4. Specify the installer node and the setup type in the cluster definition file.

The setup type must be ece for IBM Storage Scale Erasure Code Edition.

```
./spectrumscale setup -s InstallerNodeIP -st ece
```

5. Run the config populate command to populate the cluster definition file with the current cluster configuration.

   ```
   ./spectrumscale config populate -N ScaleOutNodeIP
   ```

   Note: If the config populate command does not work, you can still use the installation toolkit to populate the cluster configuration by using manual commands such as **./spectrumscale node add**.

   Verify the populated configuration of the cluster that is to be upgraded.

   ```
   ./spectrumscale node list
   ```

   You can exclude nodes from the current upgrade process by using the following command.

   ```
    ./spectrumscale upgrade config exclude -N NodeName
   ```

   You can list the excluded nodes by using the following command.

   ```
   ./spectrumscale upgrade config list
   ```

6. Perform the installation toolkit upgrade precheck before the online upgrade.

   ```
   ./spectrumscale upgrade precheck
   ```

   ```
   [ INFO  ] Logging to file: /usr/lpp/mmfs/5.1.y.z/ansible-toolkit/logs/UPGRADE-PRECHECK-dd-
   mm-yyyy_hh:mm:ss.log
   After a node is upgraded in an ECE setup, certain rebuild and rebalance tasks need to be
   performed.
   The default timeout for these tasks is infinite. During the upgrade precheck, you can
   specify a timeout
   for these rebuild and rebalance tasks depending on your environment.
   Do you want to continue with default timeout [Y/n]: n
   Please specify a timeout value in minutes : 30
   ```

   Note: It is recommended to specify a timeout value for scenarios in which an error occurs when the cluster is upgrading.

If you are running applications on scale-out nodes that directly access the GPFS file system in IBM Storage Scale Erasure Code Edition cluster, you need to stop the applications or migrate the applications to other nodes before you upgrade these nodes. Alternatively, you must turn on the interactive mode. Then, the installation toolkit prompts you to stop the applications before you upgrade a node.

Note: Remote file system access is not affected by online upgrade.

7. Change the installation toolkit to interactive mode when scale-out nodes are being used as local file system access nodes.

   ```
   ./spectrumscale upgrade config workloadprompt -N all
   ```

   Note:

   - Without turning on the interactive mode, the installation toolkit shuts down GPFS even if applications are running on the node that is being upgraded.
   - When the installation toolkit upgrades a protocol node, even if the interactive mode is enabled, no prompts are displayed. The installation toolkit suspends protocol services before you upgrade and after upgrading it resumes protocol services, and exports the file systems again on the node that is being upgraded. To check the consistency of the exported file systems, you can use the **showmount -e** command before and after the upgrade.

8. Perform the installation toolkit upgrade.

```
./spectrumscale upgrade run
```

```
The upgrade process is divided into several phases. If the nodes being upgraded include
both protocol (CES)
 and non-protocol nodes, then protocol nodes and non-protocol nodes are upgraded in
separate phases.
If the nodes being upgraded are all of the same type, either protocol or non-protocol, then
only the
upgrade phases applicable to that node type will be performed.
The upgrade process may cause a brief outage of Object, SMB, NFS, HDFS and Performance
Monitoring components.
Do you really want to begin upgrading? [y/N]: y
[ INFO  ] Logging to file: /usr/lpp/mmfs/5.1.y.z/ansible-toolkit/logs/UPGRADE-dd-mm-
yyyy_hh:mm:ss.log
After a node is upgraded in an ECE setup, certain rebuild and rebalance tasks need to be
performed.
The default timeout for these tasks is infinite. During the upgrade precheck, you can
specify a timeout
for these rebuild and rebalance tasks depending on your environment.
Do you want to continue with default timeout [Y/n]: n
Please specify a timeout value in minutes : 30
```

You can access the installation toolkit upgrade logs from the `/usr/lpp/mmfs/5.x.y.z/ansible-toolkit/logs` directory.

9. If you are using customized udev rules on your storage nodes, you need to reapply those changes to the new udev rules. The previous rules are saved in the `/etc/udev/rules.d/` directory as part of the upgrade. After you apply the changes, activate the changes by using the **udevadm** command.
10. After the upgrade process is done, complete the upgrade to the new code level to take advantage of the new functions. For more information, see *Completing the upgrade to a new level of IBM Storage Scale* in *IBM Storage Scale: Concepts, Planning, and Installation Guide*.
11. After successfully upgrading all ECE nodes to the new release, you must verify and update the configurations on the ECE server with the new release settings. For the detailed steps, see "Verifying and updating the configurations" on page 69.

After the upgrade is completed, you can use the installation toolkit for tasks such as adding new nodes, adding NSDs, creating more file systems, adding management GUI nodes, and adding protocol nodes. For more information, see *Performing additional tasks using the installation toolkit* in *IBM Storage Scale: Concepts, Planning, and Installation Guide*.

Offline upgrade of IBM Storage Scale Erasure Code Edition by using the installation toolkit

You can upgrade to a newer available version of IBM Storage Scale Erasure Code Edition by using the installation toolkit. For upgrading from a version earlier than 5.0.4.3 to a later version (including from version 5.0.4.3. to 5.0.4.4), you can use the installation toolkit only for offline upgrade. For upgrading from version 5.0.4.3 or later to version 5.0.5 or later, you can use the installation toolkit for online upgrade. For more information, see "Online upgrade of IBM Storage Scale Erasure Code Edition by using the installation toolkit" on page 61.

1. Download IBM Storage Scale Erasure Code Edition self-extracting package from the IBM Storage Scale page on Fix Central.

   The name of IBM Storage Scale Erasure Code Edition self-extracting installation package is similar to `Spectrum_Scale_Erasure_Code-5.x.y.z-x86_64-Linux-install`.
2. Extract the installation package.

   ```
   # ./Spectrum_Scale_Erasure_Code-5.x.y.z-x86_64-Linux-install --silent --textonly
   ```

   ```
   # ./Spectrum_Scale_Erasure_Code-5.x.y.z-x86_64-Linux-install --textonly
   ```

   The installation toolkit gets extracted to the `/usr/lpp/mmfs/5.x.y.z/ansible-toolkit/` directory.

To verify that the extracted package is of IBM Storage Scale Erasure Code Edition, go to the `/usr/lpp/mmfs/5.x.y.z/gpfs_rpms` directory and check for `gpfs.gnr*` packages.

3. Change the directory to where the installation toolkit is extracted.

   ```
   cd /usr/lpp/mmfs/5.x.y.z/ansible-toolkit
   ```

4. Specify the installer node and the setup type in the cluster definition file.

   The setup type must be `ece` for IBM Storage Scale Erasure Code Edition.

   ```
   ./spectrumscale setup -s InstallerNodeIP -st ece
   ```

5. Run the config populate command to populate the cluster definition file with the current cluster configuration.

   ```
   ./spectrumscale config populate -N ScaleOutNodeIP
   ```

   Note: If the config populate command does not work, you can still use the installation toolkit to populate the cluster configuration by using manual commands such as **`./spectrumscale node add`**.

6. Stop the workloads that are running on the nodes that you are upgrading.
7. If there are protocol nodes in the cluster, suspend Cluster Export Services (CES) on the protocol nodes and stop protocol services.

   ```
   mmces node suspend -N ProtocolNodeList --stop
   ```

   ProtocolNodeList is a list of all protocol nodes in the cluster.

8. Shut down GPFS on all nodes in the cluster.

   ```
   mmshutdown -a
   ```

9. Designate all nodes in the cluster as offline in the installation toolkit upgrade configuration.

   ```
   ./spectrumscale upgrade config offline -N NodeList
   ```

   NodeList is a list of all nodes in the cluster.

   You can exclude nodes from the current upgrade process by using the following command:

   ```
   ./spectrumscale upgrade config exclude -N NodeName
   ```

Perform the installation toolkit upgrade precheck and upgrade operations to upgrade IBM Storage Scale Erasure Code Edition cluster after you run the config populate command.

10. Perform the installation toolkit upgrade precheck before the installation toolkit upgrade.

    ```
    ./spectrumscale upgrade precheck
    ```

11. Perform the installation toolkit upgrade.

    ```
    ./spectrumscale upgrade run
    ```

    You can access the installation toolkit upgrade logs from the `/usr/lpp/mmfs/5.x.y.z/ansible-toolkit/logs` directory.

12. If you are using customized udev rules on your storage nodes, you need to reapply those changes to the new udev rules. The previous rules are saved in the `/etc/udev/rules.d/` directory as part of the upgrade. After you apply your changes, activate the changes with the **udevadm** command.
13. Start GPFS on all nodes in the cluster.

    ```
    mmstartup -a
    ```

14. If there are protocol nodes in the cluster, resume CES on the protocol nodes and start protocol services.

```
mmces node resume -N ProtocolNodeList --start
```

ProtocolNodeList is a list of all protocol nodes in the cluster.

15. After the upgrade process is done, complete the upgrade to the new code level to take advantage of the new functions. For more information, see *Completing the upgrade to a new level of IBM Storage Scale* in *IBM Storage Scale: Concepts, Planning, and Installation Guide*.
16. After successfully upgrading all ECE nodes to the new release, you must verify and update the configurations on the ECE server with the new release settings. For the detailed steps, see "Verifying and updating the configurations" on page 69.

After the upgrade is completed, you can use the installation toolkit for tasks such as adding new nodes, adding NSDs, creating more file systems, adding management GUI nodes, and adding protocol nodes. For more information, see *Performing additional tasks using the installation toolkit* in *IBM Storage Scale: Concepts, Planning, and Installation Guide*.

Manual online upgrade of IBM Storage Scale Erasure Code Edition

You can upgrade to a newer available version of IBM Storage Scale Erasure Code Edition by using the manual online upgrade procedure.

Before you begin:

- The versions of firmware drivers and operating system on each node must meet the requirements of IBM Storage Scale Erasure Code Edition version that you are planning to upgrade to. For upgrading firmware or OS on the nodes, see the respective vendor documentation.
- It is recommended to plan the upgrade when IBM Storage Scale Erasure Code Edition cluster is running a light workload.
- IBM Storage Scale Erasure Code Edition allows nodes in mixed old and new versions in the cluster. The administrator can divide the whole upgrade plan into several upgrade windows.

About fault tolerance: Fault tolerance is important in the entire upgrade progress and the administrator must monitor it from the beginning to the end of the upgrade. Fault tolerance can get affected due to an offline node or due to a pdisk failure and it automatically recovers after failures are repaired. The administrator must check the fault tolerance in each node during the upgrade because a node is offline from IBM Storage Scale Erasure Code Edition cluster when it is being upgraded. It is recommended to recover the fault tolerance to the best possible configuration at the beginning of upgrade of each node. For more information, see "Understanding IBM Storage Scale Erasure Code Edition fault tolerance" on page 5.

Special scenarios: If the cluster has multiple recovery groups, the administrator can speed up the upgrading process by upgrading multiple nodes in different recovery groups at one time.

Typically, node quorum is sufficient for upgrading. However, there are some scenarios when there might be a risk of losing quorum. For example, if there are three recovery groups in IBM Storage Scale Erasure Code Edition cluster and if you upgrade one quorum node in each recovery group, it results in three offline quorum nodes at a time. The administrator must be aware of the risk of losing quorum during each upgrade process. For more information, see *Node quorum* in *IBM Storage Scale: Concepts, Planning, and Installation Guide*.

1. Prepare for upgrading as follows.
   a) Download IBM Storage Scale Erasure Code Edition self-extracting package from the IBM Storage Scale page on Fix Central.

      The name of IBM Storage Scale Erasure Code Edition self-extracting installation package is similar to `Spectrum_Scale_Erasure_Code-5.1.y.z-x86_64-Linux-install`.

   b) Extract the installation package.

      ```
      ./Spectrum_Scale_Erasure_Code-5.1.y.z-x86_64-Linux-install --silent --text-only
      ```

If no directory is specified, the self-extracting package extracts the GPFS RPMs to the `/usr/lpp/mmfs/5.1.y.z/gpfs_rpms` directory. Copy the `gpfs_rpms` directory to all nodes that are to be upgraded in IBM Storage Scale Erasure Code Edition cluster.

c) Perform a health check of all nodes in the cluster.

```
mmdsh -N all 'mmhealth node show'
mmhealth cluster show
```

If IBM Storage Scale Erasure Code Edition nodes in the cluster are not healthy, resolve the issues before you proceed with the upgrade. If you are unable to resolve the issues, Contact IBM Storage Scale support to assess the upgrade risk.

d) Save the initial IBM Storage Scale Erasure Code Edition cluster configuration before doing any changes.

```
mmfsadm dump config > ./cluster_config_before_upgrade.txt
```

e) Save the initial mount map.

```
mmlsmount all -L > mount_map_before_upgrade.txt
```

Customer might mount several file systems with the auto-mount method. However, it is possible that some auto-mount points are manually unmounted. Or, some new mount points are mounted by mistake, but they are not mounted before even if they are in the auto-mount list.

During the upgrade progress, the mount configuration must remain unchanged before and after upgrading.

Note: The following steps describe an example scenario wherein one node of a single recovery group is being upgraded. These steps must be done iteratively until all scale out nodes in the cluster are upgraded.

Online upgrade allows running a light workload on the cluster during the upgrade window.

2. Check the current state of the cluster to ensure that it is ready for the upgrade.

a) Monitor the pdisk status.

```
mmvdisk pdisk list --recovery-group all --not-ok
```

The expected output for this command is `mmvdisk: All pdisks are ok.` If the output is not similar, repair all pdisk failures before proceeding with the upgrade. If you are unable to resolve the issues, contact IBM Storage Scale support to assess the upgrade risk.

b) View the recovery group fault tolerance information.

```
mmvdisk recoverygroup list --recovery-group rgName --fault-tolerance
```

The recovery group on which the upgrade procedure is being run must have a minimum fault tolerance of at least one node + one pdisk failure. However, a fault tolerance of 2-node is recommended. The administrator must ensure that the recovery group has the best possible fault tolerance. If the minimum fault tolerance cannot be satisfied, over a significant part of the upgrade window, stop upgrading and contact IBM Storage Scale support.

3. Suspend or stop workloads on the node that is being upgraded.

a) If the node that is being upgraded is also running protocols in the cluster, suspend Cluster Export Services (CES) on the protocol node and stop protocol services.

```
mmces node suspend -N nodeName --stop
```

For information on upgrading protocol nodes, see *Upgrading IBM Storage Scale protocol nodes* in *IBM Storage Scale: Concepts, Planning, and Installation Guide*.

b) Stop or migrate the workloads off the node if they are running on the locally mounted GPFS file system.

c) Unmount all the file systems including user file systems and the CES shared root file system.

```
mmumount Device -N nodeName
```

You can use the following command to view the mount information.

```
mmlsmount all -L
```

To minimize upgrade time, the administrator must avoid upgrading nodes that are assigned with important roles. The administrator must migrate roles to different nodes that are not going to be upgraded or that are already upgraded.

4. Change the node roles as follows:

   a) Migrate the cluster manager.

      i) View the current cluster manager.

      ```
      mmlsmgr -c
      ```

      ii) If the node that is to be upgraded is listed as a cluster manager, migrate the cluster manager role to another node.

      ```
      mmchmgr -c nodeName
      ```

   b) Migrate the file system manager for all file systems.

      i) For each file system, run the following command to view the current file system manager.

      ```
      mmlsmgr filesystemName
      ```

      ii) If the node that is to be upgraded is listed as a file system manager, migrate the file system manager role to another node.

      ```
      mmchmgr filesystemName nodeName
      ```

   c) Use one of the following set of steps depending on the version that you are upgrading from.

      Note: If you are running an IBM Storage Scale Erasure Code Edition version earlier than 5.0.4.3, use the **tsrecgroupserver** command to migrate log groups to other nodes before shutting down GPFS on the node. If you are running version 5.0.4.3 or later, use the **mmvdisk** command to suspend the node.

      - If upgrading from a version earlier than 5.0.4.3, migrate the log groups.

        i) View the balanced log groups.

        ```
        mmvdisk server list --recovery-group rgName
        ```

        Typically, log groups are balanced across each scale out node.

        ii) For the node that is to be upgraded, move all the log groups that are residing on it to different nodes as follows.

           a) View the log groups that are residing on the node.

           ```
           mmvdisk server list --recovery-group rgName
           ```

           b) Move these log groups to different nodes.

           ```
           tsrecgroupserver rgName -f -l loggroupName nodeName
           ```

        For example, the output of the list command generates the following output:

        ```
        1  node1               yes      serving rg01: root, LG002, LG006
        ```

        According to the output, run the following commands to move all 3 log groups to three different nodes in the same recovery group (node2, node3, and node4 must be in the same recovery group).

```
# tsrecgroupserver rgName -f -l root node2
# tsrecgroupserver rgName -f -l LG002 node3
# tsrecgroupserver rgName -f -l LG006 node4
```

- If upgrading from version 5.0.4.3 or later, suspend the node.

  i) Use the **mmvdisk** command to suspend the node.

  ```
  mmvdisk rg change --recovery-group rgName --suspend -N nodeName --window minutes
  ```

  This command enables defer rebuild in the node upgrading time. You need to estimate the upgrade process duration for this node and specify the time in minutes with the `--window` option. You only need to run this command for the node that is being upgraded which is an I/O server in IBM Storage Scale Erasure Code Edition cluster.

  Note: If you are suspending a node, you must resume that node while starting up GPFS after the upgrade.

5. Shut down GPFS on the node.

   ```
   mmshutdown -N nodeName
   ```

6. Upgrade IBM Storage Scale Erasure Code Edition software.

   a) Change the directory to where the RPMs are located.

   ```
   cd /usr/lpp/mmfs/5.1.y.z/gpfs_rpms
   ```

   b) Upgrade the RPMs.

   ```
   rpm -Uvh --force --nodeps gpfs.base*.rpm gpfs.gpl*.rpm gpfs.crypto*.rpm
   gpfs.adv*.rpm gpfs.gskit*.rpm gpfs.msg*.rpm gpfs.gnr*.rpm
   gpfs.docs*.rpm gpfs.license.*.rpm
   ```

   c) Check the version to ensure that the updated version of RPMs is installed on the node.

   ```
   rpm -qa | grep gpfs
   ```

   d) Rebuild the GPFS portability layer (GPL).

   ```
   mmbuildgpl
   ```

   e) If you are using customized udev rules on your storage nodes, you need to reapply those changes to the new udev rules. The previous rules are saved in the `/etc/udev/rules.d/` directory as part of the upgrade. After applying your changes, activate the changes with the **udevadm** command.

7. Start GPFS on the node.

   ```
   mmstartup -N nodeName
   ```

   If you used **mmvdisk suspend** command to suspend this node earlier in the procedure, use the following command to resume the node to disable defer rebuild.

   ```
   mmvdisk rg change --recovery-group rgName --resume -N nodeName
   ```

   You can use the **mmgetstate -a** and the **mmlsrecoverygroup rgName -L --pdisk** commands to check the status of all nodes in the cluster.

8. Resume or restart workloads on the node that is being upgraded.

   a) Mount the file systems that were unmounted earlier.

   Remount all file systems according to the original mount map. If the file systems are set to auto mount, check if those file systems are mounted as saved in the `mount_map_before_upgrade.text` file in an earlier step.

b) If protocol services were stopped on the node, resume CES on the protocol node and start protocol services.

```
mmces node resume -N nodeName --start
```

c) If the workloads were earlier running on a locally mounted file system, restart or migrate the workload back on the upgraded node.

Repeat steps 2 - 8 on all the nodes one by one until all scale out nodes in the cluster are upgraded to the new IBM Storage Scale Erasure Code Edition version.

9. After the upgrade process is done, complete the upgrade to the new code level to take advantage of the new functions. For more information, see *Completing the upgrade to a new level of IBM Storage Scale* in *IBM Storage Scale: Concepts, Planning, and Installation Guide*.
10. After successfully upgrading all ECE nodes to the new release, you must verify and update the configurations on the ECE server with the new release settings. For the detailed steps, see “Verifying and updating the configurations” on page 69.

Verifying and updating the configurations

You can use the steps to verify and update the configurations after upgrading your IBM Storage Scale Erasure Code Edition cluster.

Use the following steps to verify and update the configurations of the new IBM Storage Scale Erasure Code Edition release:

1. Verify the different configurations between new and old release.

   ```
   mmvdisk server configure --nc nc_1 --verify
   ```

   Note:

   In the command output, you must check whether the expected value is same as the configured value.

   If the expected value is not set or changed after upgrade, you must run the next step to update the configurations.

2. Update the configurations changed in new release.

   ```
   mmvdisk server configure --update --nc nc_1 --recycle one
   ```

   Note: This command updates the configurations on each node by restarting IBM Storage Scale one by one. You must verify the configuration again after this command.

Chapter 10. Incorporating IBM Storage Scale Erasure Code Edition in an IBM Elastic Storage Server (ESS) cluster

Use these procedures to incorporate IBM Storage Scale Erasure Code Edition in an Elastic Storage Server (ESS) cluster.

Ensure that the following prerequisites are met before you install IBM Storage Scale Erasure Code Edition in an Elastic Storage Server (ESS) cluster.

Note: An existing IBM Storage Scale Erasure Code Edition setup is not supported for this integration procedure. Before performing the procedure, the installation prerequisites must be met on IBM Storage Scale Erasure Code Edition candidate nodes.

- ESS version is 5.3.4 or later.
- IBM Storage Scale Erasure Code Edition version is 5.0.3.1 or later.

  Note: IBM Storage Scale Erasure Code Edition version that you use in this procedure must match the version that is already running on the system. You cannot integrate IBM Storage Scale Erasure Code Edition into an Elastic Storage Server (ESS) cluster and upgrade IBM Storage Scale Erasure Code Edition at the same time.
- Minimum hardware requirements are met.
- All typical IBM Storage Scale and ESS prerequisites such as passwordless SSH, minimum OS levels, python, sg3_utils, and pciutils software requirements are met.

  The IBM Storage Scale installation toolkit can help identify any missing prerequisites.
- Network performance minimum requirements are met.
- General understanding of how the IBM Storage Scale installation toolkit process works.
- Possible protocol architecture conflicts are mitigated.

The installation of IBM Storage Scale Erasure Code Edition in an Elastic Storage Server (ESS) cluster comprises four phases.

1. Phase 1: Convert ESS into mmvdisk management.
2. Phase 2: Add nodes to the ESS cluster using the installation toolkit.
3. Phase 3: Prepare the IBM Storage Scale Erasure Code Edition cluster using the installation toolkit.
4. Phase 4: Complete the configuration with mmvdisk commands.

Converting Elastic Storage Server (ESS) to mmvdisk management

In the 1st phase of incorporating IBM Storage Scale Erasure Code Edition in an ESS cluster, check if ESS is **mmvdisk** managed, and if required convert ESS to **mmvdisk** managed.

1. Check if ESS is **mmvdisk** managed.

   If the cluster is not **mmvdisk** managed, the `remarks` column in the output contains `non-mmvdisk`.

   ```
   # mmvdisk server list

   node
   number  server                            node class  recovery groups             remarks
   ------  --------------------------------  ----------  -------------------------  -------
        1  gssio1-ib.example.com                -           rg_gssio1-ib, rg_gssio2-ib  non-
   mmvdisk
        2  gssio2-ib.example.com                -           rg_gssio1-ib, rg_gssio2-ib  non-
   mmvdisk

   # mmvdisk rg list
   ```

```
                                                            needs    user
recovery group  active   current or master server          service  vdisks  remarks
--------------  -------  --------------------------------  -------  ------  -------
rg_gssio1-ib    yes      gssio1-ib.example.com                no            1  non-mmvdisk
rg_gssio2-ib    yes      gssio2-ib.example.com                no            1  non-mmvdisk
```

2. Convert ESS to **mmvdisk** managed.

```
# mmvdisk recoverygroup convert --recovery-group rg_gssio1-ib,rg_gssio2-ib --node-class
ess_nc1
```

Note: The node class name, which is specified with **--node-class**, must not be in use and the node class is created as an mmvdisk node class that contains the two servers for the recovery group pair.

A sample output is as follows.

```
mmvdisk: This command will permanently change the GNR configuration
mmvdisk: attributes and disable the legacy GNR command set for the
mmvdisk: servers and recovery groups involved, and their subsequent
mmvdisk: administration must be performed with the mmvdisk command.

mmvdisk: Do you wish to continue (yes or no)? yes

mmvdisk: Converting recovery groups 'rg_gssio1-ib' and 'rg_gssio2-ib'.
mmvdisk: Creating node class 'ess_nc1'.
mmvdisk: Adding 'gssio1-ib' to node class 'ess_nc1'.
mmvdisk: Adding 'gssio2-ib' to node class 'ess_nc1'.
mmvdisk: Associating recovery group 'rg_gssio1-ib' with node class 'ess_nc1'.
mmvdisk: Associating recovery group 'rg_gssio2-ib' with node class 'ess_nc1'.
mmvdisk: Recording pre-conversion cluster configuration
mmvdisk: in /var/mmfs/tmp/mmvdisk.convert.rg_gssio1-ib.rg_gssio2-ib.before.m07
mmvdisk: Updating server configuration attributes.
mmvdisk: Checking resources for specified nodes.
mmvdisk: Setting configuration for node class 'ess_nc1'.
mmvdisk: Defining vdisk set 'VS001_essFS' with recovery group
mmvdisk: 'rg_gssio1-ib' (vdisk 'rg_gssio1_ib_DA1_DataAndMetaData_16M_2p_1').
mmvdisk: Defining vdisk set 'VS002_essFS' with recovery group
mmvdisk: 'rg_gssio2-ib' (vdisk 'rg_gssio2_ib_DA1_DataAndMetaData_16M_2p_1').
mmvdisk: Committing cluster configuration changes.
mmvdisk: Recording post-conversion cluster configuration in
mmvdisk: /var/mmfs/tmp/mmvdisk.convert.rg_gssio1-ib.rg_gssio2-ib.after.m07
mmvdisk: For configuration changes to take effect, GPFS should be restarted
mmvdisk: on node class 'ess_nc1'.
```

3. Restart GPFS.

```
# mmshutdown -a
# mmstartup -a
```

4. Verify the GPFS state.

```
# mmgetstate -a
```

5. View the ESS cluster after it is converted to **mmvdisk** managed.

```
# mmvdisk server list


node
number  server                 node class  recovery groups            remarks
------  ---------------------  ----------  -------------------------  -------
   1    gssio1-ib.example.com  ess_nc1     rg_gssio1-ib, rg_gssio2-ib
   2    gssio2-ib.example.com  ess_nc1     rg_gssio1-ib, rg_gssio2-ib

# mmvdisk rg list


                                                            needs    user
recovery group  active   current or master server          service  vdisks  remarks
--------------  -------  --------------------------------  -------  ------  -------
rg_gssio1-ib    yes      gssio1-ib.example.com                no       1
rg_gssio2-ib    yes      gssio2-ib.example.com                no       1

# mmvdisk vdisk list --vdisk-set all
```

```
                                                                       declustered array,
RAID code,
vdisk               vdisk set            file system    recovery group          block
size           remarks
------------------  -------------------  -------------  --------------
---------------------------  -------
rg_gssio1_ib_DA1_DataAndMetaData_16M_2p_1  VS001_essFS  essFS  rg_gssio1-ib  DA1, 8+2p, 16
MiB
rg_gssio2_ib_DA1_DataAndMetaData_16M_2p_1  VS002_essFS  essFS  rg_gssio2-ib  DA1, 8+2p, 16
MiB
```

Adding nodes to the Elastic Storage Server (ESS) cluster using the installation toolkit

In the 2nd phase of incorporating IBM Storage Scale Erasure Code Edition in an ESS cluster, use the installation toolkit to create a generic cluster definition file that will be used to install Erasure Code Edition candidate nodes in the ESS cluster as generic IBM Storage Scale nodes.

Note: The steps in this phase need to be done on Erasure Code Edition candidate nodes, not on the ESS nodes.

1. From IBM Fix Central, download the IBM Storage Scale Advanced Edition 5.x.y.z installation package. You must download this package to the node that you plan to use as your installer node for the IBM Storage Scale Advanced Edition installation and the subsequent IBM Storage Scale Erasure Code Edition installation. Also, use a node that you plan to add in the existing ESS cluster.
2. Extract the IBM Storage Scale Advanced Edition 5.x.y.z installation package to the default directory or a directory of your choice on the node that you plan to use as the installer node.

   ```
   /DirectoryPathToDownloadedCode/Spectrum_Scale_Advanced-5.x.y.z-x86_64-Linux-install
   ```

3. Change the directory to the default directory for the installation toolkit.

   ```
   # cd /usr/lpp/mmfs/5.x.y.z/ansible-toolkit
   ```

4. Set up the installer node and the setup type as ess.

   In this command example, 198.51.100.1 is the IP address of the scale-out node that is planned to be designated as the installer node.

   ```
   # ./spectrumscale setup -s 198.51.100.1 -st ess

   [ INFO ] Installing prerequisites for install node
   [ INFO  ] Found existing Ansible installation on system.
   [ INFO ] Install Toolkit setup type is set to ESS. This mode will allow the EMS node to
   execute Install Toolkit commands.
   [ INFO ] Your Ansible control node has been configured to use the IP 198.51.100.1 to
   communicate
   with other nodes.
   [ INFO ] Port 10080 will be used for package distribution.
   [ INFO ] SUCCESS
   [ INFO ] Tip : Designate an EMS node as admin node: ./spectrumscale node add <node> -a
   [ INFO ] Tip : After designating an EMS node, add nodes for the toolkit to act upon:
   ./spectrumscale node add <node> -p -n
   [ INFO ] Tip : After designating the EMS node, if you want to populate the cluster definition
   file with the current configuration, you can run: ./spectrumscale config populate -N
   <ems_node>
   ```

5. Add the existing EMS node to the cluster definition as admin, quorum, and EMS nodes.

   ```
   # ./spectrumscale node add ess.example.com -a -q -e
   ```

   ```

   [ INFO ] Adding node ess.example.com as a GPFS node.
   [ INFO ] Adding node ess.example.com as a quorum node.
   [ INFO ] Setting ess.example.com as an admin node.
   [ INFO ] Configuration updated.
   [ INFO ] Setting ess.example.com as an ESS node.
   [ INFO ] Configuration updated.
   ```

```
# ./spectrumscale node list

[ INFO  ] List of nodes in current configuration:
[ INFO  ] [Installer Node]
[ INFO  ] 198.51.100.1
[ INFO  ]
[ INFO  ] [Cluster Details]
[ INFO  ] Name: scalecluster.example.com
[ INFO  ] Setup Type: ESS
[ INFO  ]
[ INFO  ] [Extended Features]
[ INFO  ] File Audit logging     : Disabled
[ INFO  ] Watch folder           : Disabled
[ INFO  ] Management GUI         : Enabled
[ INFO  ] Performance Monitoring : Disabled
[ INFO  ] Callhome               : Disabled
[ INFO  ]
[ INFO  ] GPFS                Admin  Quorum  Manager   NSD   Protocol   GUI    Perf Mon
EMS   OS   Arch
[ INFO  ] Node                Node   Node     Node   Server   Node    Server
Collector
[ INFO  ] ess.example.com     X       X
X   rhel7  ppc64le
[ INFO  ]
[ INFO  ] [Export IP address]
[ INFO  ] No export IP addresses configured
```

6. Add IBM Storage Scale Erasure Code Edition candidate nodes generically.

```
# ./spectrumscale node add 198.51.100.1
[ INFO ] Adding node node1.example.com as a GPFS node.
# ./spectrumscale node add 198.51.100.2
[ INFO ] Adding node node2.example.com as a GPFS node.
# ./spectrumscale node add 198.51.100.3
[ INFO ] Adding node node3.example.com as a GPFS node.
# ./spectrumscale node add 198.51.100.4
[ INFO ] Adding node node4.example.com as a GPFS node.
# ./spectrumscale node add 198.51.100.5
[ INFO ] Adding node node5.example.com as a GPFS node.
# ./spectrumscale node add 198.51.100.6
[ INFO ] Adding node node6.example.com as a GPFS node.
```

Verify the node details.

```
# ./spectrumscale node list

[ INFO  ] List of nodes in current configuration:
[ INFO  ] [Installer Node]
[ INFO  ] 198.51.100.1
[ INFO  ]
[ INFO  ] [Cluster Details]
[ INFO  ] Name: scalecluster.example.com
[ INFO  ] Setup Type: ESS
[ INFO  ]
[ INFO  ] [Extended Features]
[ INFO  ] File Audit logging     : Disabled
[ INFO  ] Watch folder           : Disabled
[ INFO  ] Management GUI         : Enabled
[ INFO  ] Performance Monitoring : Enabled
[ INFO  ] Callhome               : Disabled
[ INFO  ]
[ INFO  ] GPFS                Admin  Quorum  Manager   NSD   Protocol   GUI    Perf Mon
EMS   OS   Arch
[ INFO  ] Node                Node   Node     Node   Server   Node    Server
Collector
[ INFO  ] ess.example.com     X       X                                                   X
rhel7  ppc64le
[ INFO  ] node1.example.com
rhel7  x86_64
[ INFO  ] node2.example.com
rhel7  x86_64
[ INFO  ] node3.example.com
rhel7  x86_64
[ INFO  ] node4.example.com
rhel7  x86_64
[ INFO  ] node5.example.com
rhel7  x86_64
[ INFO  ] node6.example.com
rhel7  x86_64
```

```
[ INFO  ] [Export IP address]
[ INFO  ] No export IP addresses configured
```

7. Perform an installation precheck by using the installation toolkit.

```
# ./spectrumscale install -pr


[ INFO ] Logging to file: /usr/lpp/mmfs/5.x.y.z/ansible-toolkit/logs/INSTALL-PRECHECK-02-02-
2021_13:17:42.log
[ INFO ] Validating configuration
[ WARN ] No NSD servers specified. The install toolkit will continue without creating any
NSDs. If you still want to continue, please ignore this warning. Otherwise, for information
on adding a node as an NSD server, see:
'http://www.ibm.com/support/knowledgecenter/STXKQY_5.0.3/com.ibm.spectrum.scale.v
5r03.doc/bl1ins_configuringgpfs.htm'
[ INFO ] Performing GPFS checks.
[ INFO ] Running environment checks
[ WARN ] No manager nodes specified. Assuming managers already configured on ESS.gpfs.net
…
[ INFO ] Checking pre-requisites for portability layer.
[ INFO ] GPFS precheck OK
[ INFO ] Performing Performance Monitoring checks.
[ INFO ] Running environment checks for Performance Monitoring
[ INFO ] Performing FILE AUDIT LOGGING checks.
[ INFO ] Running environment checks for file Audit logging
[ INFO ] Network check from admin node node1.example.com to all other nodes
in the cluster passed
[ WARN ] Ephemeral port range is not set. Please set valid ephemeral port range using the
command ./spectrumscale config gpfs --ephemeral_port_range . You may set the default
values as 60000-61000
[ INFO ] The install toolkit will not configure call home as it is disabled. To enable call
home,
use the following CLI command: ./spectrumscale callhome enable
[ INFO ] Pre-check successful for install.
[ INFO ] Tip : ./spectrumscale install
```

8. Install the nodes that are defined in the cluster definition by using the installation toolkit.

```
# ./spectrumscale install


[ INFO ] Logging to file: /usr/lpp/mmfs/5.x.y.z/ansible-toolkit/logs/INSTALL-02-02-
2021_18:18:29.log
[ INFO ] Validating configuration
[ WARN ] No NSD servers specified. The install toolkit will continue without creating any
NSDs. If you still want to continue, please ignore this warning. Otherwise, for information
on adding a node as an NSD server, see:
'http://www.ibm.com/support/knowledgecenter/STXKQY_5.0.3/com.ibm.spectrum.scale.v
5r03.doc/bl1ins_configuringgpfs.htm'
[ INFO ] Running pre-install checks
[ INFO ] Running environment checks
[ INFO ] The following nodes will be added to cluster scalecluster.example.com: node1-
.example.com, node2.example.com, node3.example.com, node4.example.com, node5.example.com,
node6.example.com, ess.example.com,
[ WARN ] No manager nodes specified. Assuming managers already configured on
ESS.gpfs.net.
…
…
…
…
[ INFO ] Checking for a successful install
[ INFO ] Checking state of GPFS
[ INFO ] GPFS callhome has been successfully installed. To configure callhome run
'mmcallhome -h' on one of your nodes.
[ INFO ] Checking state of GPFS on all nodes
[ INFO ] GPFS active on all nodes
[ INFO ] GPFS ACTIVE
[ INFO ] Checking state of Performance Monitoring
[ INFO ] Running Performance Monitoring post-install checks
[ WARN ] Historical performance data is still kept on: node1.example.com in the
'/opt/IBM/zimon/data' directory. For documentation on migrating the data to the new
Performance Monitoring collectors: refer to the IBM Spectrum Scale Knowledge Center.
[ INFO ] pmcollector running on all nodes
[ INFO ] pmsensors running on all nodes
[ INFO ] Performance Monitoring ACTIVE
[ INFO ] SUCCESS
[ INFO ] All services running
[ INFO ] StanzaFile and NodeDesc file for NSD, filesystem, and cluster setup have been
saved to /usr/lpp/mmfs folder on node: ess.example.com
```

```
[ INFO ] Installation successful. 7 GPFS nodes active in cluster scalecluster.example.com.
Completed in 6
minutes 6 seconds.
[ INFO ] Tip :If all node designations and any required protocol configurations are complete,
proceed to check the deploy configuration:./spectrumscale deploy --precheck
```

9. Verify that the installation completed successfully by issuing the following command.

   ```
   # ./spectrumscale install -po


   [ INFO ] Logging to file: /usr/lpp/mmfs/5.x.x.x/installer/logs/INSTALL-POSTCHECK-06-08-
   2019_13:25:31.log
   [ WARN ] No NSD servers specified. The install toolkit will continue without creating any
   NSDs. If you still want to continue, please ignore this warning. Otherwise, for information
   on adding a node as an NSD server, see:
   'http://www.ibm.com/support/knowledgecenter/STXKQY_5.0.3/com.ibm.spectrum.scale.v
   5r03.doc/bl1ins_configuringgpfs.htm'
   [ INFO ] Checking state of GPFS
   [ INFO ] GPFS callhome has been successfully installed. To configure callhome run
   'mmcallhome -h' on one of your nodes.
   [ INFO ] Checking state of GPFS on all nodes
   [ INFO ] GPFS active on all nodes
   [ INFO ] GPFS ACTIVE
   [ INFO ] Checking state of Performance Monitoring
   [ INFO ] Running Performance Monitoring post-install checks
   [ WARN ] Historical performance data is still kept on: ess.example.com in the
   '/opt/IBM/zimon/data' directory. For documentation on migrating the data to the new
   Performance Monitoring collectors: refer to the IBM Spectrum Scale Knowledge Center.
   [ INFO ] pmcollector running on all nodes
   [ INFO ] pmsensors running on all nodes
   ```

Before you proceed to the next phase, remove the Advanced Edition license package from IBM Storage Scale Erasure Code Edition candidate nodes by using the following command.

```
mmdsh -N ListofECECandidateNodes "rpm -e gpfs.license.adv"
```

Preparing the IBM Storage Scale Erasure Code Edition cluster using the installation toolkit

In the 3rd phase of incorporating IBM Storage Scale Erasure Code Edition in an ESS cluster, use the installation toolkit to create a new cluster definition file that would be used to create an unconfigured IBM Storage Scale Erasure Code Edition cluster within the ESS cluster.

Note: The steps in this phase need to be done on IBM Storage Scale Erasure Code Edition candidate nodes, not on the ESS nodes.

1. From IBM Fix Central, download IBM Storage Scale Erasure Code Edition 5.x.y.z installation package on your installer node.
2. Extract IBM Storage Scale Erasure Code Edition 5.x.y.z installation package to a directory on the installer node that is different from the installer directory that you used for the initial installation and deployment in phase 2. For example, `/usr/lpp/mmfs/5.x.y.z_ECE_New/`.

   ```
   /DirectoryPathToDownloadedCode/Spectrum_Scale_Erasure_Code-5.x.y.z-x86_64-Linux-install
   --dir /usr/lpp/mmfs/5.x.y.z_ECE_New/
   ```

3. Change the directory to the new directory in which the package was extracted.

   ```
   # cd /usr/lpp/mmfs/5.x.y.z_ECE_New/ansible-toolkit
   ```

4. Change the setup type of the installer node to `ece`.

   In this command example, 198.51.100.1 is the IP address of the scale-out node that is designated as the installer node.

   ```
   # ./spectrumscale setup -s 198.51.100.1 -st ece

   [ INFO ] Installing prerequisites for install node
   [ INFO  ] Found existing Ansible installation on system.
   ```

```
[ INFO ] Install Toolkit setup type is set to ECE (Erasure Code Edition).
[ INFO ] Your Ansible control node has been configured to use the IP 198.51.100.1 to
communicate
with other nodes.
[ INFO ] Port 10080 will be used for package distribution.
[ INFO ] SUCCESS
[ INFO ] Tip : Designate scale out, protocol and admin nodes in your environment to use
during install:./spectrumscale node add <node> -p -a -so
```

Verify the node details.

```
# ./spectrumscale node list

[ INFO ] List of nodes in current configuration:
[ INFO ] [Installer Node]
[ INFO ] 198.51.100.1
[ INFO ]
[ INFO ] [Cluster Details]
[ INFO ] No cluster name configured
[ INFO ] Setup Type: Spectrum Scale
[ INFO ]
[ INFO ] [Extended Features]
[ INFO ] File Audit logging : Disabled
[ INFO ] Watch folder : Disabled
[ INFO ] Management GUI : Disabled
[ INFO ] Performance Monitoring : Disabled
[ INFO ] Callhome : Enabled
[ INFO ]
[ INFO ] No nodes configured. Use 'spectrumscale node add' to add nodes.
[ INFO ] If a cluster already exists use 'spectrumscale config populate -N node_in_cluster'
to
sync toolkit with existing cluster.
```

5. Add the same IBM Storage Scale Erasure Code Edition candidate nodes and any other nodes that you added previously for functions such as file audit logging to the cluster.

   Note:

   - Designate one of IBM Storage Scale Erasure Code Edition candidate nodes as an admin node.
   - Do not add the EMS node in this part of the configuration.

```
# ./spectrumscale node add 198.51.100.1 -a -so

[ INFO ] Setting node1.example.com as an admin node.
[ INFO  ] Setting node1.example.com as a scale-out node.
[ INFO  ] Configuration updated.

# ./spectrumscale node add 198.51.100.2 -so

[ INFO  ] Setting node2.example.com as a scale-out node.
[ INFO  ] Configuration updated

# ./spectrumscale node add 198.51.100.3 -so

[ INFO  ] Setting node3.example.com as a scale-out node.
[ INFO  ] Configuration updated.

# ./spectrumscale node add 198.51.100.4 -so

[ INFO  ] Setting node4.example.com as a scale-out node.
[ INFO  ] Configuration updated.


# ./spectrumscale node add 198.51.100.5 -so

[ INFO  ] Setting node5.example.com as a scale-out node.
[ INFO  ] Configuration updated.

# ./spectrumscale node add 198.51.100.6 -so

[ INFO  ] Setting node6.example.com as a scale-out node.
[ INFO  ] Configuration updated.
```

Verify the node details.

```
# ./spectrumscale node list
```

```
[ INFO ] List of nodes in current configuration:
[ INFO ] [Installer Node]
[ INFO ] 198.51.100.1
[ INFO ]
[ INFO ] [Cluster Details]
[ INFO ] No cluster name configured
[ INFO ] Setup Type: Erasure Code Edition
[ INFO ]
[ INFO ] [Extended Features]
[ INFO ] File Audit logging : Disabled
[ INFO ] Watch folder : Disabled
[ INFO ] Management GUI : Disabled
[ INFO ] Performance Monitoring : Disabled
[ INFO ] Callhome : Enabled
[ INFO ]
[ INFO  ] GPFS               Admin  Quorum  Manager   NSD   Protocol  Callhome   Scale-out
OS   Arch
[ INFO  ] Node                Node   Node     Node   Server   Node    Server
Node
[ INFO  ] node1.example.com    X                                                   X
rhel7  x86_64
[ INFO  ] node2.example.com                                                        X
rhel7  x86_64
[ INFO  ] node3.example.com                                                        X
rhel7  x86_64
[ INFO  ] node4.example.com                                                        X
rhel7  x86_64
[ INFO  ] node5.example.com                                                        X
rhel7  x86_64
[ INFO  ] node6.example.com                                                        X
rhel7  x86_64

[ INFO ]
[ INFO ] [Export IP address]
[ INFO ] No export IP addresses configured
```

6. Do an installation pre-check by using the installation toolkit.

```
# ./spectrumscale install -pr
```

7. Install the defined IBM Storage Scale Erasure Code Edition configuration by using the installation toolkit.

```
# ./spectrumscale install
```

Completing the IBM Storage Scale Erasure Code Edition configuration with mmvdisk commands

In the fourth phase of incorporating IBM Storage Scale Erasure Code Edition in an ESS cluster, use **mmvdisk** commands from any IBM Storage Scale Erasure Code Edition **mmvdisk** enabled node in the cluster to complete the configuration of IBM Storage Scale Erasure Code Edition cluster.

1. Create IBM Storage Scale Erasure Code Edition node class from the candidate scale-out nodes that you deployed earlier.

```

# mmvdisk nc create --node-class ece_nc1 -N node1,node2,node3,node4,node5,node6

mmvdisk: Node class 'ece_nc1' created.
```

2. Configure IBM Storage Scale Erasure Code Edition node class and restart GPFS.

```
# mmvdisk server configure --node-class ece_nc1 --recycle one

mmvdisk: Checking resources for specified nodes.
mmvdisk: Node class 'ece_nc1' has a scale-out recovery group disk topology.
mmvdisk: Using 'default.scale-out' RG configuration for topology 'ECE 2 HDD'.
mmvdisk: Setting configuration for node class 'ece_nc1'.
mmvdisk: Node class 'ece_nc1' is now configured to be recovery group servers.
mmvdisk: Restarting GPFS daemon on node 'node1'.
mmvdisk: Restarting GPFS daemon on node 'node2'.
mmvdisk: Restarting GPFS daemon on node 'node4'.
mmvdisk: Restarting GPFS daemon on node 'node3'.
```

```
mmvdisk: Restarting GPFS daemon on node 'node6'.
mmvdisk: Restarting GPFS daemon on node 'node5'.
```

Note: The `--recycle one` option restarts GPFS to enable new configuration one by one. Be careful when you use the `--recycle all` option. When you use this option, the **mmvdisk** command asks the following confirmation on the console:

```
# mmvdisk server configure --update --nc nc_1 --recycle all

mmvdisk: This command will shutdown GPFS on multiple nodes at the same time.
mmvdisk: It is possible to lose quorum and cluster availability.
mmvdisk: It is possible to lose file system or recovery group availability.

mmvdisk: Do you wish to continue (yes or no)?
```

If you provide "yes", the command restarts all the nodes at the same time, which might cause the cluster to lose quorum or file system availability.

Verify the node class details.

```
# mmvdisk nc list

node class recovery groups
---------- --------------------------
ece_nc1 -
ess_nc1 rg_gssio1-ib, rg_gssio2-ib
```

3. Configure and create the recovery group.

```
# mmvdisk rg create --rg ece_rg1 --nc ece_nc1

mmvdisk: Checking node class configuration.
mmvdisk: Checking daemon status on node 'node1.example.com'.
mmvdisk: Checking daemon status on node 'node4.example.com'.
mmvdisk: Checking daemon status on node 'node5.example.com'.
mmvdisk: Checking daemon status on node 'node6.example.com'.
mmvdisk: Checking daemon status on node 'node3.example.com'.
mmvdisk: Checking daemon status on node 'node2.example.com'.
mmvdisk: Analyzing disk topology for node 'node1.example.com'.
mmvdisk: Analyzing disk topology for node 'node4.example.com'.
mmvdisk: Analyzing disk topology for node 'node5.example.com'.
mmvdisk: Analyzing disk topology for node 'node6.example.com'.
mmvdisk: Analyzing disk topology for node 'node3.example.com'.
mmvdisk: Analyzing disk topology for node 'node2.example.com'.
mmvdisk: Creating recovery group 'ece_rg1'.
mmvdisk: Formatting log vdisks for recovery group.
mmvdisk: (mmcrvdisk) [I] Processing vdisk RG003ROOTLOGHOME
mmvdisk: (mmcrvdisk) [I] Processing vdisk RG003LG001LOGHOME
mmvdisk: (mmcrvdisk) [I] Processing vdisk RG003LG002LOGHOME
mmvdisk: (mmcrvdisk) [I] Processing vdisk RG003LG003LOGHOME
mmvdisk: (mmcrvdisk) [I] Processing vdisk RG003LG004LOGHOME
mmvdisk: (mmcrvdisk) [I] Processing vdisk RG003LG005LOGHOME
mmvdisk: (mmcrvdisk) [I] Processing vdisk RG003LG006LOGHOME
mmvdisk: (mmcrvdisk) [I] Processing vdisk RG003LG007LOGHOME
mmvdisk: (mmcrvdisk) [I] Processing vdisk RG003LG008LOGHOME
mmvdisk: (mmcrvdisk) [I] Processing vdisk RG003LG009LOGHOME
mmvdisk: (mmcrvdisk) [I] Processing vdisk RG003LG010LOGHOME
mmvdisk: (mmcrvdisk) [I] Processing vdisk RG003LG011LOGHOME
mmvdisk: (mmcrvdisk) [I] Processing vdisk RG003LG012LOGHOME
mmvdisk: Created recovery group 'ece_rg1'.
```

Verify the recovery group details.

```
# mmvdisk rg list


                                                needs   user
recovery group active current or master server service vdisks remarks
-------------- ------ ------------------------ ------- ------ ------
ece_rg1         yes    node1.example.com       no       0
rg_gssio1-ib    yes    gssio1-ib.example.com   no       1
rg_gssio2-ib    yes    gssio2-ib.example.com   no       1
```

4. Define the vdisk sets with the desired parameters.

- In this command example, IBM Storage Scale Erasure Code Edition vdisk set is defined as a dataOnly storage pool that is separate from the existing ESS pool. The ESS pool in this case is the system pool and it is defined as dataAndMetadata.
- Make sure you use the same block size (16 M in this case) as the existing ESS file system if you are merging this vdisk set into that file system.

```
# mmvdisk vs define --vs ece_vs1 --rg ece_rg1 --code 8+2p --block-size 16M
  --set-size 80% --storage-pool ece_pool_1 --nsd-usage dataOnly

mmvdisk: Vdisk set 'ece_vs1' has been defined.
mmvdisk: Recovery group 'ece_rg1' has been defined in vdisk set 'ece_vs1'.

                     member vdisks
vdisk set       count   size   raw size  created  file system and attributes
--------------  ----- -------- --------  -------  --------------------------
ece_vs1          12   62 GiB   80 GiB    no       -, DA1, 8+2p, 16 MiB, dataOnly, ece_pool_1

                declustered                 capacity            all vdisk sets defined
recovery group     array     type  total raw  free raw  free%  in the declustered array
--------------  -----------  ----  ---------  --------  -----  ------------------------
ece_rg1             DA1      HDD    1213 GiB   253 GiB    20%  ece_vs1

              vdisk set map       memory per server
node class  available  required  required per vdisk set
----------  ---------  --------  ----------------------
ece_nc1      8996 MiB   390 MiB  ece_vs1 (2304 KiB)
```

5. Create vdisks, NSDs, and the vdisk set from the defined storage.

```
# mmvdisk vs create --vs ece_vs1

mmvdisk: 12 vdisks and 12 NSDs will be created in vdisk set 'ece_vs1'.
mmvdisk: (mmcrvdisk) [I] Processing vdisk RG003LG001VS003
mmvdisk: (mmcrvdisk) [I] Processing vdisk RG003LG002VS003
mmvdisk: (mmcrvdisk) [I] Processing vdisk RG003LG003VS003
mmvdisk: (mmcrvdisk) [I] Processing vdisk RG003LG004VS003
mmvdisk: (mmcrvdisk) [I] Processing vdisk RG003LG005VS003
mmvdisk: (mmcrvdisk) [I] Processing vdisk RG003LG006VS003
mmvdisk: (mmcrvdisk) [I] Processing vdisk RG003LG007VS003
mmvdisk: (mmcrvdisk) [I] Processing vdisk RG003LG008VS003
mmvdisk: (mmcrvdisk) [I] Processing vdisk RG003LG009VS003
mmvdisk: (mmcrvdisk) [I] Processing vdisk RG003LG010VS003
mmvdisk: (mmcrvdisk) [I] Processing vdisk RG003LG011VS003
mmvdisk: (mmcrvdisk) [I] Processing vdisk RG003LG012VS003
mmvdisk: Created all vdisks in vdisk set 'ece_vs1'.
mmvdisk: (mmcrnsd) Processing disk RG003LG001VS003
mmvdisk: (mmcrnsd) Processing disk RG003LG002VS003
mmvdisk: (mmcrnsd) Processing disk RG003LG003VS003
mmvdisk: (mmcrnsd) Processing disk RG003LG004VS003
mmvdisk: (mmcrnsd) Processing disk RG003LG005VS003
mmvdisk: (mmcrnsd) Processing disk RG003LG006VS003
mmvdisk: (mmcrnsd) Processing disk RG003LG007VS003
mmvdisk: (mmcrnsd) Processing disk RG003LG008VS003
mmvdisk: (mmcrnsd) Processing disk RG003LG009VS003
mmvdisk: (mmcrnsd) Processing disk RG003LG010VS003
mmvdisk: (mmcrnsd) Processing disk RG003LG011VS003
mmvdisk: (mmcrnsd) Processing disk RG003LG012VS003
mmvdisk: Created all NSDs in vdisk set 'ece_vs1'.
```

6. From any **mmvdisk** enabled node in the cluster, add the new vdisk set to the existing file system.

```
# mmvdisk fs add --fs ecefs1 --vs ece_vs1

mmvdisk: Creating file system 'ecefs1'.
mmvdisk: The following disks of ecefs1 will be formatted on node gssio2.example.com:
mmvdisk: RG003LG001VS003: size 64000 MB
mmvdisk: RG003LG002VS003: size 64000 MB
mmvdisk: RG003LG003VS003: size 64000 MB
mmvdisk: RG003LG004VS003: size 64000 MB
mmvdisk: RG003LG005VS003: size 64000 MB
mmvdisk: RG003LG006VS003: size 64000 MB
mmvdisk: RG003LG007VS003: size 64000 MB
mmvdisk: RG003LG008VS003: size 64000 MB
mmvdisk: RG003LG009VS003: size 64000 MB
mmvdisk: RG003LG010VS003: size 64000 MB
mmvdisk: RG003LG011VS003: size 64000 MB
```

```
mmvdisk: RG003LG012VS003: size 64000 MB
mmvdisk: Extending Allocation Map
mmvdisk: Creating Allocation Map for storage pool ece_pool_1
mmvdisk: Flushing Allocation Map for storage pool ece_pool_1
mmvdisk: Disks up to size 966.97 GB can be added to storage pool ece_pool_1.
mmvdisk: Checking Allocation Map for storage pool ece_pool_1
mmvdisk: Completed adding disks to file system ecefs1.
```

7. Verify the following entities from any **mmvdisk** enabled node.

   - File system details:

     ```
     # mmvdisk fs list


     file system     vdisk sets
     -----------     ----------
     ecefs1          VS001_essFS, VS002_essFS, ece_vs1
     ```

     Storage pools in the file system

     ```
     # mmlspool ecefs1

     Storage pools in file system at '/gpfs/ecefs1':
     Name        Id     BlkSize Data Meta Total Data in (KB) Free Data in (KB) Total Meta in
     (KB) Free Meta in (KB)
     system      0      16 MB   yes yes 12501204992 12496994304 (100%) 12501204992 12497076224
     (100%)
     ece_pool_1  65537  16 MB   yes no  786432000 785252352 (100%) 0
     0 ( 0%)
     ```

   - Recovery groups:

     ```
     # mmvdisk rg list

                                                      needs    user
     recovery group  active   current or master server           service  vdisks  remarks
     --------------  -------  ---------------------------------  -------  ------  -------
     ece_rg1         yes      node1.example.com                  no           12
     rg_gssio1-ib    yes      gssio1-ib.example.com              no            1
     rg_gssio2-ib    yes      gssio2-ib.example.com              no            1
     ```

   - pdisks for the new recovery group ece_rg1:

     ```
     # mmvdisk pdisk list --rg ece_rg1

                          declustered
     recovery group  pdisk            array     paths  capacity  free space  FRU (type)
     state
     --------------  ------------  -----------  -----  --------  ----------  ---------------
     -----
     ece_rg1         n013p001      DA1          1      136 GiB   44 GiB      42D0623
     ok
     ece_rg1         n013p002      DA1          1      136 GiB   44 GiB      42D0422
     ok
     ece_rg1         n014p001      DA1          1      136 GiB   44 GiB      42D0623
     ok
     ece_rg1         n014p002      DA1          1      136 GiB   44 GiB      42D0422
     ok
     ece_rg1         n015p001      DA1          1      136 GiB   44 GiB      42D0623
     ok
     ece_rg1         n015p002      DA1          1      136 GiB   44 GiB      42D0422
     ok
     ece_rg1         n016p001      DA1          1      136 GiB   44 GiB      42D0623
     ok
     ece_rg1         n016p002      DA1          1      136 GiB   44 GiB      42D0422
     ok
     ece_rg1         n017p001      DA1          1      136 GiB   44 GiB      42D0623
     ok
     ece_rg1         n017p002      DA1          1      136 GiB   44 GiB      42D0422
     ok
     ece_rg1         n018p001      DA1          1      136 GiB   44 GiB      22R6802
     ok
     ece_rg1         n018p002      DA1          1      136 GiB   44 GiB      42D0422
     ok
     ```

Chapter 11. Incorporating IBM Elastic Storage Server (ESS) building block in an IBM Storage Scale Erasure Code Edition cluster

Use the following steps to add a newly shipped ESS building block into an existing ECE cluster.

Ensure that the following prerequisites are met before you install Elastic Storage Server (ESS) building block in an IBM Storage Scale Erasure Code Edition cluster.

Note: An existing Elastic Storage Server (ESS) setup is not supported for this integration procedure. Before performing the procedure, the installation prerequisites must be met on Elastic Storage Server (ESS) candidate building block.

- ESS version is 6.1.5.0 or later.
- IBM Storage Scale Erasure Code Edition version is 5.1.5.1 or later.

  Note: Elastic Storage Server (ESS) version that you use in this procedure must match the version that is already running on the IBM Storage Scale Erasure Code Edition.
- Configure Elastic Storage Server (ESS) network according to requirements, such as hardware management network, software management network, and Daemon network.
- Elastic Storage Server (ESS) Daemon network must connect to IBM Storage Scale Erasure Code Edition cluster daemon network and Network performance requirements are met.

The installation of Elastic Storage Server (ESS) in an IBM Storage Scale Erasure Code Edition cluster comprises four phases.

1. Phase 1: "Prepare the ESS building block" on page 83.
2. Phase 2: "Install the ESS software and complete the network configuration" on page 83.
3. Phase 3: "Adding the ESS building block nodes into the ECE cluster" on page 84.
4. Phase 4: "Completing the configuration with mmvdisk command" on page 85.

Prepare the ESS building block

In the first phase of incorporating Elastic Storage Server (ESS) building block in IBM Storage Scale Erasure Code Edition cluster, you must prepare the ESS building block.

Follow *Hardware Planning and Installation Guide* in the ESS document for all SSR works. After this step, all hardware must be installed with network connected.

Note:

Make sure that the daemon network is correctly connected with ECE cluster.

Since the EMS node of the ESS system cannot manage ECE nodes, other network connections are not required.

Install the ESS software and complete the network configuration

In the second phase of incorporating Elastic Storage Server (ESS) building block in IBM Storage Scale Erasure Code Edition cluster, you must install the ESS software and complete the network configuration.

Follow *ESS Deployment Guide* in the ESS document to install and update the software on the ESS building block that includes the EMS node.

- Follow *ESS common installation instructions* to load the configuration and check the building block health state.

- Follow *ESS new deployment instructions* to upgrade the EMS and I/O nodes and creating networks. Stop the steps before creating ESS cluster.

  Note: Do not create ESS cluster as the EMS and I/O nodes of ESS building block will be added into ECE cluster.

Note: Make sure all nodes' IP addresses are correctly resolved by hostname, which is used as the daemon network. Configure the SSH passwordless access for all nodes, including ECE cluster nodes, ESS EMS, and I/O nodes.

Adding the ESS building block nodes into the ECE cluster

In the third phase of incorporating Elastic Storage Server (ESS) building block in IBM Storage Scale Erasure Code Edition cluster, you must add ESS EMS and I/O nodes into the ECE cluster.

1. Add ESS EMS node to the existing ECE cluster by running the following command from ESS EMS node.

   ```
   essaddnode -N ems_node_name --suffix=-hs --accept-license --no-fw-update --cluster-node
   ece_node_name --nodetype ems
   ```

   The following example shows how to run the command and what the output will look like:

   ```
   essaddnode -N ems --suffix=-hs --accept-license --no-fw-update --cluster-node ece1-ib --
   nodetype ems

   [INFO]  Starting add node...
   [INFO]  Nodeclass of target node: ems
   [INFO]  Getting list of quorum nodes.
   [INFO]  Number of quorum nodes in the cluster: 5
   [INFO]  Adding node(s) to cluster...
   [OK]    mmaddnode: Command successfully completed
   mmaddnode: Propagating the cluster configuration data to all affected nodes.

   [INFO]  Checking if nodeclass ems exists...
   [INFO]  Nodeclass ems does not exist.
   [INFO]  Adding node to nodeclass ems...
   [OK]
   [OK]    mmcrnodeclass: Propagating the cluster configuration data to all
     affected nodes.  This is an asynchronous process.

   [INFO]  Starting nodes...
   [INFO]  Active added node(s) found: 1
   [OK]    Node addition complete
   ```

2. Add ESS I/O nodes to the existing ECE cluster by running the following command from ESS EMS node.

   ```
   essaddnode -N essio1_node_name,essio2_node_name --suffix=-hs --accept-license --no-fw-update
   --cluster-node ece_node_name --nodetype ess_type
   ```

   The following example shows how to run the command and what the output will look like:

   ```
   essaddnode -N essio1,essio2 --suffix=-hs --accept-license --no-fw-update --cluster-node ece1-
   ib --nodetype ess3500

   [INFO]  Starting add node...
   [INFO]  Nodeclass of target node: ess_x86_64
   [INFO]  Getting list of quorum nodes.
   [INFO]  Number of quorum nodes in the cluster: 6
   [INFO]  Adding node(s) to cluster...
   [OK]    mmaddnode: Command successfully completed
   mmaddnode: Propagating the cluster configuration data to all affected nodes.
   mmaddnode: Propagating the cluster configuration data to all
     affected nodes.  This is an asynchronous process.

   [INFO]  Checking if nodeclass ess_x86_64 exists...
   [INFO]  Nodeclass ess_x86_64 does not exist.
   [INFO]  Adding node to nodeclass ess_x86_64...
   [OK]
   [OK]    mmcrnodeclass: Propagating the cluster configuration data to all
     affected nodes.  This is an asynchronous process.

   [INFO]  Starting nodes...
   ```

```
[INFO]  Active added node(s) found: 2
[OK]    Node addition complete
```

3. Check that the ECE cluster has the ESS nodes with **mmlscluster** command.

Completing the configuration with mmvdisk command

In the fourth phase of incorporating Elastic Storage Server (ESS) building block in IBM Storage Scale Erasure Code Edition cluster, you must complete the configuration with the **mmvdisk** command.

1. Make sure that all nodes are in active state.

   ```
   mmgetstate -a
   ```

2. Check for the node class name, which includes only ESS I/O nodes.

   ```
   mmlsnodeclass
   ```

3. Create recovery groups by submitting the following command from the EMS deploy container.

   ```
   essrun -N ESS_IO_node_class filesystem --rg-only --suffix=-hs
   ```

   Note:

   You can also use **mmvdisk** command directly to create the Recovery group on ESS I/O nodes.
4. Check the current RGs state. More than one Recovery Group will be listed including the ESS recovery group created.

   ```
   mmvdisk rg list
   ```

5. Create vdisks of ESS recovery group with **mmvdisk** command on I/O nodes.

   In this command example, ESS vdisk set is defined as a `dataOnly` storage pool that is separate from the existing ECE pool. The ECE pool in this case is the system pool and it is defined as `dataAndMetadata`.

   Note: Make sure you use the same block size (4 M in this case) as the existing ECE file system when you are merging this vdisk set into that file system.

   ```
   mmvdisk vs define --vs vs_fs1_essio1_hs_essio2_hs --rg ess3500_essio1_hs_essio2_hs --code
   8+2p --bs 4M --ss 80% --nsd-usage dataOnly --sp ess_data_pool
   ```

   ```
   mmvdisk vs create vs_fs1_essio1_hs_essio2_hs
   ```

6. Add ESS vdisks into existing ECE filesystem.

   ```
   mmvdisk fs add --file-system fs1 --vdisk-set vs_fs1_essio1_hs_essio2_hs
   ```

   Note: If needed, a few more steps can be followed in the ESS document.

   - Configuring performance monitoring and starting the GUI.

     Note: If you used GUI in the ECE cluster, you can switch to the ESS GUI on the EMS node.
   - Setting up ESS call home.
   - Setting up time server for all nodes.

Chapter 12. Administering IBM Storage Scale Erasure Code Edition

Physical disk procedures

This topic describes the various procedures that you can perform for the maintenance of disks.

1. Identify the problem disks. Use the following command to check the current disks that have a problem:

   ```
   # mmvdisk pdisk list --rg all --not-ok
   ```

   ```

                                 declustered
   recovery group  pdisk            array     paths  capacity  free space  FRU (type)
   state
   --------------  ------------  -----------  -----  --------  ----------  ---------------
   -----
   rg_1            n002p001      DA1              0   894 GiB     890 GiB  PX04PMB096
   missing/drained
   rg_1            n005p002      DA1              0   894 GiB     890 GiB  PX04PMB096
   failing/replace
   ```

   Note: If you find the state of a disk as "missing", it usually does not mean that there is a problem with the disk drive. Therefore, the "missing" state might be because of a disk connection problem or a network problem of the node, and you need to find the root cause of the problem. For example, to re-seat the drive or bring back the node. If the missing disk is confirmed of drive hardware failure after checking the system, the disk can be replaced by completing step 2 after it is drained.

2. Perform the following steps to replace disks:

   - To identify the pdisk to be replaced within all recovery groups:

     ```
     mmvdisk pdisk list --rg all --replace
     ```

     The system displays the following output:

     ```

     recovery group  pdisk          priority  FRU (type)       location
     --------------  ------------   --------  ---------------  --------
     rg_1            n005p003          12.95  00YK014          Enclosure J1005744 Drive 6
     rg_1            n005p004          12.95  00YK014          Enclosure J1005744 Drive 7

     mmvdisk: A lower priority value means a higher need for replacement.
     ```

     Note:

     – If you replace a pdisk not on this list, you risk data loss.

     – If the number of disks need replacement is below the replacement threshold for its member declustered array, then those disks will not generate call home behavior.

     – It is recommended to set your replacement threshold to 1 if you want call home happening as earlier as possible when you have only one disk failing.

   - To set your replacement threshold to 1:

     ```
     mmvdisk rg change --rg RgName --da DaName --replace-threshold 1
     ```

   - To replace hot swappable disk devices:

     a. Issue the following command:

     ```
     mmvdisk pdisk replace --prepare --recovery-group RgName --pdisk PdiskName
     ```

The system displays an output as follows:

```
mmvdisk: Suspending pdisk n005p003 of RG rg_1 in location J1005744-6.
mmvdisk: Location J1005744-6 is Enclosure J1005744 Drive 6.
mmvdisk: Carrier released.
mmvdisk:
mmvdisk:    - Remove carrier.
mmvdisk:    - Replace disk in location J1005744-6 with type '00YK014'.
mmvdisk:    - Reinsert carrier.
mmvdisk:    - Issue the following command:
mmvdisk:
mmvdisk:   mmvdisk pdisk replace --recovery-group rg_1 --pdisk 'n005p003'
```

b. Go to the node to replace a new disk for the `pdisk` according to the slot location.

c. Issue the following command:

```
mmvdisk pdisk replace --recovery-group RgName --pdisk PdiskName
```

The system displays an output as follows:

```

mmvdisk:
mmvdisk: mmchcarrier : [I] Preparing a new pdisk for use may take many minutes.
mmvdisk:
mmvdisk: The following pdisks will be formatted on node HostName:
mmvdisk:     // HostName /dev/DevName
mmvdisk: Pdisk PdiskName of RG RgName successfully replaced.
mmvdisk: Resuming pdisk PdiskName#nnn of RG RgName.
mmvdisk: Carrier resumed.
```

Note: After you replace a new pdisk in the slot, ensure to check and disable the volatile write cache on the new pdisk. For more information, see "Volatile write cache detection" on page 99.

Export and import of file systems based on ECE recovery groups

IBM Storage Scale Erasure Code Edition offers support for exporting and importing file systems based on vdisks within recovery groups. This feature can be used to move one or more file systems from one cluster to another, or to temporarily remove file systems from a cluster and restore them later.

The **mmexportfs** and **mmimportfs** commands can be used to perform the following migration process:

- Export the current file system and recovery group configuration by using the **mmexportfs** command.
- Physically move all disk drives associated with a recovery group from the old servers within the previous cluster to the new servers in the new cluster.
- Use the **mmimportfs** command to import the file system and recovery group configuration into the new cluster.

Note: SED-enabled recovery groups do not support export and import of file systems.

Important considerations

- Before initiating the file system export or import process, ensure that both the file system and its associated recovery groups are in a healthy state.
- Verify that there are no ongoing rebuild or rebalancing operations, and confirm that all physical disks (pdisks) are in good state. Import process might not succeed if missing or damaged disks are detected in the new cluster.
- Before migrating the disk drives to the destination cluster, ensure that the servers in the destination cluster have the capability to manage the disk drives correctly. It is recommended to test the same type of drives on the new cluster servers to ensure compatibility and functionality.
- The destination cluster must maintain the same number of servers designated for the recovery groups as in the original configuration.
- Destination cluster must not have file systems or recovery groups with the same name as in the original one.

- Before migrating the disk drives, the system must be prepared with a mapping drives configuration file. For more information, see “Setting up IBM Storage Scale Erasure Code Edition for disk slot location” on page 34.

Exporting a file system and a recovery groups on an old ECE cluster

Use the **mmexportfs** command to export both file system and recovery group configurations.

For more information about the **mmexportfs** command, see the ***mmexportfs*** in the *IBM Storage Scale: Command and Programming Reference Guide* in the IBM Storage Scale documentation.

1. Check the recovery group and file system state.

   - Check the file system state.

     ```
     mmvdisk fs list --fs test_fs1
     ```

     A sample output is as follows:

     ```
                                     vdisk  with     list of       holds    holds  storage
     vdisk set       recovery group  count  trim  failure groups  metadata  data    pool
     --------------  --------------  -----  ----  --------------  --------  -----  -------
     VS1             rg1                 8     0            1, 2  yes       yes    system
     ```

   - Check the recovery state.

     ```
     mmvdisk rg list --rg rg1 --da --server --vdisk-set --pdisk
     ```

     A sample output is as follows:

     ```
     node
     number  server                            active   remarks
     ------  --------------------------------  -------  -------
          3  c72f3u25-eth0                     yes      serving rg1: LG002, LG006
          1  c72f3u27-eth0                     yes      serving rg1: root, LG004, LG008
          2  c72f3u29-eth0                     yes      serving rg1: LG001, LG005
          4  c72f3u31-eth0                     yes      serving rg1: LG003, LG007

     declustered   needs                         vdisks       pdisks           capacity
        array     service  type    BER    trim  user log  total spare rt  total raw free raw
     background task
     -----------  -------  ----  -------  ----  ---- ---  ----- ----- --  --------- --------
     ---------------
     DA1          no       HDD   enable   no       8   9     16     2  2     24 TiB   12 TiB
     scrub 14d (61%)

     mmvdisk: Total capacity is the raw space before any vdisk set definitions.
     mmvdisk: Free capacity is what remains for additional vdisk set definitions.

                   declustered      paths                               AU
     pdisk            array     active  total  capacity  free space  log size  state
     ------------  -----------  ------  -----  --------  ----------  --------  -----
     n001p001      DA1               1      1  1863 GiB    1017 GiB  3072 MiB  ok
     n001p002      DA1               1      1  1863 GiB    1017 GiB  3072 MiB  ok
     n001p003      DA1               1      1  1863 GiB    1017 GiB  3072 MiB  ok
     n001p004      DA1               1      1  1863 GiB    1017 GiB  3072 MiB  ok
     n002p001      DA1               1      1  1863 GiB    1024 GiB  3072 MiB  ok
     n002p002      DA1               1      1  1863 GiB    1016 GiB  3072 MiB  ok
     n002p003      DA1               1      1  1863 GiB    1017 GiB  3072 MiB  ok
     n002p004      DA1               1      1  1863 GiB    1017 GiB  3072 MiB  ok
     n003p001      DA1               1      1  1863 GiB    1018 GiB  3072 MiB  ok
     n003p002      DA1               1      1  1863 GiB    1017 GiB  3072 MiB  ok
     n003p003      DA1               1      1  1863 GiB    1017 GiB  3072 MiB  ok
     n003p004      DA1               1      1  1863 GiB    1017 GiB  3072 MiB  ok
     n004p001      DA1               1      1  1863 GiB    1017 GiB  3072 MiB  ok
     n004p002      DA1               1      1  1863 GiB    1018 GiB  3072 MiB  ok
     n004p003      DA1               1      1  1863 GiB    1017 GiB  3072 MiB  ok
     n004p004      DA1               1      1  1863 GiB    1017 GiB  3072 MiB  ok

                     declustered                 capacity            all vdisk sets defined
     recovery group     array     type  total raw  free raw  free%  in the declustered array
     --------------  -----------  ----  ---------  --------  -----  ------------------------
     rg1             DA1          HDD      24 TiB    12 TiB    50%  VS1

                       vdisk set map memory per server
     ```

```
node class  available  required  required per vdisk set
----------  ---------  --------  ----------------------
nc1            10 GiB  3633 MiB  VS1 (48 MiB)
```

2. Unmount the file system.

   ```
   # mmumount test_fs1 -a
   ```

   A sample output is as follows:

   ```
   mmumount: Unmounting file systems ...
   ```

3. Export the file system.

   ```
   # mmexportfs test_fs1 -o export
   ```

   A sample output is as follows:

   ```
   mmexportfs: Processing file system test_fs1 ...
   mmexportfs: Propagating the cluster configuration data to all affected nodes.  This is an
   asynchronous process.
   mmcommon pushSdr_async: mmsdrfs propagation started
   mmcommon pushSdr_async: mmsdrfs propagation completed; mmdsh rc=0
   ```

4. Save the export generated configuration file for later use in the new cluster. It is important to save the export file even after importing file system for later reference.
5. Move all drives to the destination servers of the new cluster.
   a) Remove drives from each server on the old cluster.
   b) Label each drive with a corresponding server ID to maintain clarity.
   c) Insert the drives into the new server, ensure that the drive to server relation is identical.
   d) Ensure that drives of different servers within a single server in the new cluster are not mixed.
   e) Ensure that all the drives were migrated. The **mmimportfs** command might fail if missing drives are found during the process.

Importing a file system and a recovery group on a new ECE cluster

Use the **mmimportfs** command to import desired file system(s) into the new cluster.

Prepare a cluster that imports a file system and a recovery group:

1. Migrate the drives from the old server to the new server on the destination cluster. Consult the hardware vendor for the safety migration procedure.
2. Prepare the drives mapping configuration file on the destination servers. For more information, see Setting up IBM Storage Scale Erasure Code Edition for disk slot location.
3. Issue the following command to create a new node class and node assignments to match the configuration of an original cluster:

   ```
   # mmvdisk nodeclass create --node-class ClassName -N Node[,Node...]
   ```

4. Issue the following command to configure servers of the node class before you import a file system and a recovery group:

   ```
   # mmvdisk server configure --node-class NodeClass --recycle one
   ```

See mmvdisk server command in *IBM Storage Scale RAID: Administration and Programming Reference* for more details.

For more information about the **mmimportfs** command, see the ***mmexportfs*** *command* in the *IBM Storage Scale: Command and Programming Reference Guide* in the IBM Storage Scale documentation.

1. Import a file system into a new cluster by using the **mmimportfs** command.

   ```
   # mmimportfs test_fs1 -i export
   ```

A sample output is as follows:

```
mmimportfs: Processing file system test_fs1 ...
mmimportfs: Processing disk RG001LG001VS001
mmimportfs: Processing disk RG001LG002VS001
mmimportfs: Processing disk RG001LG003VS001
mmimportfs: Processing disk RG001LG004VS001
mmimportfs: Processing disk RG001LG005VS001
mmimportfs: Processing disk RG001LG006VS001
mmimportfs: Processing disk RG001LG007VS001
mmimportfs: Processing disk RG001LG008VS001

mmimportfs: Validating pdisk consistency for Scale-out nodes

mmimportfs: Committing the changes ...

mmimportfs: The following file systems were successfully imported:
        test_fs1
mmimportfs: Propagating the cluster configuration data to all affected nodes.
mmimportfs: Activated recovery group rg1.
mmimportfs: Drives are inserted correctly on all nodes for RG rg1

mmimportfs: Updating RG Status ...
mmimportfs: Propagating the cluster configuration data to all affected nodes.
```

For example,

a. If the new cluster has nodes or node classes with different names from in the original, create a stanza file to specify the new names.

```
%rg: rgName=rg1
     servers=c72f3u54-eth0,c72f3u22-eth0,c72f3u11-eth0,c72f3u15-eth0
     nodeclass=nc2
```

b. Use the `-S` option with the **`mmimportfs`** command to make use of the stanza file.

```
# mmimportfs test_fs1 -i export -S stanzaFile
```

2. Check state of the imported file systems and recovery groups.

- Check the file systems status.

```
# mmvdisk fs list --fs test_fs1
```

A sample output is as follows:

```
                                vdisk  with     list of       holds    holds  storage
vdisk set       recovery group  count  trim  failure groups  metadata  data    pool
--------------  --------------  -----  ----  --------------  --------  -----  -------
VS1             rg1                 8     0            1, 2  yes       yes    system
```

- Check the recovery groups status.

```
mmvdisk rg list --rg rg1 --da --server --vdisk-set --pdisk
```

A sample output is as follows:

```
node
number  server                            active   remarks
------  --------------------------------  -------  -------
     3  c72f3u25-eth0                     yes      serving rg1: LG002, LG006
     1  c72f3u27-eth0                     yes      serving rg1: root, LG004, LG008
     2  c72f3u29-eth0                     yes      serving rg1: LG001, LG005
     4  c72f3u31-eth0                     yes      serving rg1: LG003, LG007

declustered   needs                         vdisks       pdisks           capacity
   array     service  type    BER    trim  user log  total spare rt  total raw free raw
background task
-----------  -------  ----  -------  ----  ---- ---  ----- ----- --  --------- --------
---------------
DA1          no       HDD   enable   no       8   9     16     2  2     24 TiB   12 TiB
scrub 14d (61%)

mmvdisk: Total capacity is the raw space before any vdisk set definitions.
mmvdisk: Free capacity is what remains for additional vdisk set definitions.
```

```
              declustered      paths                               AU
pdisk            array     active  total  capacity  free space  log size  state
------------  -----------  ------  -----  --------  ----------  --------  -----
n001p001      DA1               1      1  1863 GiB    1017 GiB  3072 MiB  ok
n001p002      DA1               1      1  1863 GiB    1017 GiB  3072 MiB  ok
n001p003      DA1               1      1  1863 GiB    1017 GiB  3072 MiB  ok
n001p004      DA1               1      1  1863 GiB    1017 GiB  3072 MiB  ok
n002p001      DA1               1      1  1863 GiB    1024 GiB  3072 MiB  ok
n002p002      DA1               1      1  1863 GiB    1016 GiB  3072 MiB  ok
n002p003      DA1               1      1  1863 GiB    1017 GiB  3072 MiB  ok
n002p004      DA1               1      1  1863 GiB    1017 GiB  3072 MiB  ok
n003p001      DA1               1      1  1863 GiB    1018 GiB  3072 MiB  ok
n003p002      DA1               1      1  1863 GiB    1017 GiB  3072 MiB  ok
n003p003      DA1               1      1  1863 GiB    1017 GiB  3072 MiB  ok
n003p004      DA1               1      1  1863 GiB    1017 GiB  3072 MiB  ok
n004p001      DA1               1      1  1863 GiB    1017 GiB  3072 MiB  ok
n004p002      DA1               1      1  1863 GiB    1018 GiB  3072 MiB  ok
n004p003      DA1               1      1  1863 GiB    1017 GiB  3072 MiB  ok
n004p004      DA1               1      1  1863 GiB    1017 GiB  3072 MiB  ok

                declustered                 capacity            all vdisk sets defined
recovery group     array     type  total raw  free raw  free%  in the declustered array
--------------  -----------  ----  ---------  --------  -----  ------------------------
rg1             DA1          HDD      24 TiB    12 TiB    50%  VS1

                  vdisk set map memory per server
node class  available  required  required per vdisk set
----------  ---------  --------  ----------------------
nc1            10 GiB  3633 MiB  VS1 (48 MiB)
```

3. Check pdisk installation in the new nodes.

   After importing a filesystem, it is recommended to check that the drives were migrated to the correct nodes. Use the **eceimportpdiskcheck validate** command to verify whether the disks are installed correctly.

   - An example of correctly installed pdisks

     ```
     # eceimportpdiskcheck validate
     eceimportpdiskcheck: Drives are inserted correctly on all nodes and recovery groups
     eceimportpdiskcheck: Done
     ```

   - An example of misplaced pdisks

     ```
     # eceimportpdiskcheck validate
     eceimportpdiskcheck: The following pdisks need renaming (rg:node:currentName:newName):
     eceimportpdiskcheck: rg1:c72f3u25-eth0:n001p001:n003p001
     eceimportpdiskcheck: rg1:c72f3u25-eth0:n001p002:n003p002
     eceimportpdiskcheck: rg1:c72f3u25-eth0:n001p004:n003p004
     eceimportpdiskcheck: rg1:c72f3u25-eth0:n001p003:n003p003
     eceimportpdiskcheck: rg1:c72f3u27-eth0:n003p002:n001p002
     eceimportpdiskcheck: rg1:c72f3u27-eth0:n003p004:n001p004
     eceimportpdiskcheck: rg1:c72f3u27-eth0:n003p003:n001p003
     eceimportpdiskcheck: rg1:c72f3u27-eth0:n003p001:n001p001
     ```

   The the **eceimportpdiskcheck validate** command shows a list of drives that need to be renamed if they were not installed correctly. To ensure system functionality, issue the **eceimportpdiskcheck update** command to properly rename the pdisks to match the node on which these pdisks are installed.

4. Check the imported pdisks.

   ```
   # eceimportpdiskcheck update
   eceimportpdiskcheck: The following pdisks need renaming (rg:node:currentName:newName):
   eceimportpdiskcheck: rg1:c72f3u25-eth0:n001p001:n003p001
   eceimportpdiskcheck: rg1:c72f3u25-eth0:n001p002:n003p002
   eceimportpdiskcheck: rg1:c72f3u25-eth0:n001p004:n003p004
   eceimportpdiskcheck: rg1:c72f3u25-eth0:n001p003:n003p003
   eceimportpdiskcheck: rg1:c72f3u27-eth0:n003p002:n001p002
   eceimportpdiskcheck: rg1:c72f3u27-eth0:n003p004:n001p004
   eceimportpdiskcheck: rg1:c72f3u27-eth0:n003p003:n001p003
   eceimportpdiskcheck: rg1:c72f3u27-eth0:n003p001:n001p001

   eceimportpdiskcheck: Updating scale-out pdisk names
   ```

```
eceimportpdiskcheck: Updating RG Status ...
eceimportpdiskcheck: Done
```

5. Verify the imported pdisks again.

   ```
   # eceimportpdiskcheck validate
   ```

   A sample output is as follows:

   ```
   eceimportpdiskcheck: Drives are inserted correctly on all nodes and recovery groups
   eceimportpdiskcheck: Done
   ```

   Important: The import steps may encounter failures if certain system state checks are not met. Review the command output for guidance on resolving any issues. If problems persist reach out to IBM support for assistance.

Virtual disk procedures

The **mmvdisk** command can be used to manage IBM Storage Scale Erasure Code Edition storage. The commands for listing individual or groups of virtual disks (vdisks), and for defining, creating, and deleting groups of virtual disks (vdisk sets) are available.

For more information, see the following topics:

- mmvdisk command in the *IBM Storage Scale RAID: Administration and Programming Reference*
- mmvdisk vdisk command in the *IBM Storage Scale RAID: Administration and Programming Reference*
- mmvdisk vdiskset command in the *IBM Storage Scale RAID: Administration and Programming Reference*

Node procedures

This topic describes various procedures that can be done on a node to accomplish various tasks.

Note: Any new hardware (especially disks) used for below section must be tested before put into production to make sure the hardware quality.

When you add a new node or replace a node, you need to prepare the following as the precondition for the new node to be operational:

- A homogeneous server is recommended. It must have the same CPU, memory, PCI speed, network speed, disk controller, disk number, and model. If the node has different configurations, make sure that this node does not introduce any performance bottlenecks to the cluster.
- Enclosure Descriptor File: If the server is homogeneous with other servers that include the drive mapping (which is what is recommended), the edf file (/usr/lpp/mmfs/data/gems/*edf) can be copied from the existing node to the new node. If the new server is not homogeneous with others, then new edf files must be created. For more information, see "Mapping NVMe disk slot location" on page 35.
- Setting the disks used for IBM Storage Scale Erasure Code Edition in JBOD mode, check the disk format, update firmware, and disable the disk writer cache. For more information, see "Hardware checklist" on page 12.
- SAS disk slot location: If the server is homogeneous with others that includes drive mapping and needing remapping the disk slot location, then the slot remapping file /usr/lpp/mmfs/data/gems/slotmap.yaml can be copied from an existing node to the new node. Otherwise, a new slotmap file must be created. For more information, see "Mapping LMR disk location" on page 36.
- Setting customized udev rules if required.
- Setting the systemctl settings if required.
- Follow the OS precheck tool Readme file to run the precheck tools after you prepare the node. For more information, see "Minimum hardware requirements and precheck" on page 9.

Adding new I/O nodes

Adding a new node by using the **mmvdisk** command:

1. Make sure that the node is a member of the IBM Storage Scale cluster and the state is active (if not, issue **mmaddnode** and **mmstartup**). Also, make sure that the node has the server license (if not, run **mmchlicense**).
2. Issue the **mmvdisk server list -N newnode --disk-topology** command to verify that the new node has the same disk topology as the other nodes in the recovery group to which the node is added.

   ```
   # mmvdisk server list  -N c72f4m5u15-ib0 --disk-topology -L
   ```

   The system displays the following output:

   ```
   GNR server: name c72f4m5u15-ib0 arch x86_64 model 7X06CTO1WW serial J100574A
   GNR enclosures found: internal
   Enclosure internal (internal, number 1):
   Enclosure internal sees 9 disks (6 SSDs, 3 HDDs)

   GNR server disk topology: ECE 6 SSD/NVMe and 3 HDD (match: 100/100)
   GNR configuration: 1 enclosure, 6 SSDs, 0 empty slots, 9 disks total, 0 NVRAM partitions
   ```

3. Issue the **mmvdisk server configure -N newnode --recycle one** command to configure the new node as IBM Storage Scale Erasure Code Edition server and restart the IBM Storage Scale daemon.

   ```
   # mmvdisk server configure -N c72f4m5u15-ib0 --recycle one
   ```

   ```
   mmvdisk: Checking resources for specified nodes.
   mmvdisk: Setting configuration for node 'c72f4m5u15-ib0'.
   mmvdisk: Node 'c72f4m5u15-ib0' has a scale-out recovery group disk topology.
   mmvdisk: Using 'default.scale-out' RG configuration for topology 'ECE 6 SSD/NVMe and 3 HDD'.
   mmvdisk: Node 'c72f4m5u15-ib0' is now configured to be a recovery group server.
   mmvdisk: Restarting GPFS daemon on node 'c72f4m5u15-ib0'.
   ```

4. Issue the **mmvdisk rg add --rg rg_name -N newnode** command to add the new node to the current recovery group. After that, all DAs must be in the rebalanced state. The **mmvdisk rg add --rg rg_name -N newnode** command adds a call-back script to monitor the rebalance process. When the rebalance is finished, the call-back runs **mmvdisk recoverygroup add --recovery-group rg_name --complete-node-add** command of the next step to finish the procedure for adding the node.

   ```
   # mmvdisk rg add --rg rg_1 -N c72f4m5u15-ib0
   ```

   ```
   mmvdisk: Checking daemon status on node 'c72f4m5u15-ib0'.
   mmvdisk: Checking resources for specified nodes.
   mmvdisk: Adding 'c72f4m5u15-ib0' to node class 'nc_1'.
   mmvdisk: Obtaining pdisk information for recovery group 'rg_1'.
   mmvdisk: Analyzing disk topology for node 'c72f4m5u13-ib0'.
   mmvdisk: Analyzing disk topology for node 'c72f4m5u19-ib0'.
   mmvdisk: Analyzing disk topology for node 'c72f4m5u17-ib0'.
   mmvdisk: Analyzing disk topology for node 'c72f4m5u21-ib0'.
   mmvdisk: Analyzing disk topology for node 'c72f4m5u11-ib0'.
   mmvdisk: Analyzing disk topology for node 'c72f4m5u15-ib0'.
   mmvdisk: Validating declustered arrays for recovery group 'rg_1'.
   mmvdisk: Updating server list for recovery group 'rg_1'.
   mmvdisk: Updating pdisk list for recovery group 'rg_1'.
   mmvdisk: Updating parameters for declustered array 'DA1'.
   mmvdisk: Updating parameters for declustered array 'DA2'.
   mmvdisk: Updating parameters for declustered array 'DA3'.
   mmvdisk: Node 'c72f4m5u15-ib0' added to recovery group 'rg_1'.
   mmvdisk: Log group and vdisk set operations for recovery group 'rg_1'
   mmvdisk: must be deferred until rebalance completes in all declustered arrays.
   mmvdisk: A callback 'RG001CompletNodeAdd' has been created to monitor the rebalance state.
   mmvdisk: Once rebalance completes in all declustered arrays,
   mmvdisk: log group and vdisk set will be created automatically.
   ```

   a. Check the DA status and rebalance progress by issuing the following command:

```
# mmvdisk rg list --rg rg_1 --da
```

The system displays the following output:

```
declustered   needs                vdisks       pdisks           capacity
   array     service  type  trim  user log  total spare rt  total raw free raw
background task
-----------  -------  ----  ----  ---- ---  ----- ----- --  --------- --------
---------------
DA1          no       NVMe  no      10   0     12     2  2   8869 GiB 1237 GiB  rebalance
(12%)
DA2          yes      HDD   no      10   0     18     2  2   8829 GiB 1089 GiB  rebalance
(88%)
DA3          no       SSD   no      10  11     24     3  2   9173 GiB  695 GiB  rebalance
(19%)

mmvdisk: Total capacity is the raw space before any vdisk set definitions.
mmvdisk: Free capacity is what remains for additional vdisk set definitions.

mmvdisk: Attention: Recovery group 'rg_1' has an incomplete node addition (c72f4m5u15-
ib0).
mmvdisk: callback 'RG001CompletNodeAdd' will perform the node addition after rebalance
completes
mmvdisk: in all declustered arrays of recovery group 'rg_1'.
```

b. Verify that the call-back is added by issuing the following command:

```
# mmlscallback RG001CompletNodeAdd
```

```
RG001CompletNodeAdd
        command       = /usr/lpp/mmfs/bin/mmvdisk
        sync          = false
        event         = imEventRebalance
        node          = c72f4m5u11-ib0,c72f4m5u13-ib0,c72f4m5u15-ib0,c72f4m5u17-
ib0,c72f4m5u19-ib0,c72f4m5u21-ib0
        parms         = recoverygroup add --recovery-group %rgName --complete-node-add --
callback RG001CompletNodeAdd
```

5. The call-back automatically runs the **mmvdisk recoverygroup add --recovery-group rg_name --complete-node-add** command to finish the adding node process after the rebalance is finished. This operation creates new log groups, new vdisks for all existing vdisk sets, NSDs, and adds the free NSDs to file systems if the vdisk sets belong to some file system.

   If the rebalance is ongoing, run the following command:

```
# mmvdisk recoverygroup add --recovery-group rg_1 --complete-node-add
```

```
mmvdisk: Verifying that the DAs in recovery group 'rg1' are idle.
mmvdisk: Declustered array 'DA1' is in task 'rebalance'.
mmvdisk: All DAs must be in task 'scrub' to complete node addition.
mmvdisk: Log group and vdisk set operations for recovery group 'rg1'
mmvdisk: must be deferred until rebalance completes in all declustered arrays.
mmvdisk: A callback 'RG001CompletNodeAdd' has been created to monitor the rebalance state.
mmvdisk: Once rebalance completes in all declustered arrays,
mmvdisk: log group and vdisk set will be created automatically.
mmvdisk: Command failed. Examine previous error messages to determine cause.
```

   Generally the **mmvdisk** command reports the same message if:

   - The rebalance is ongoing.
   - The call-back is not finished.

```
# mmvdisk rg list --rg rg_1 --da
```

   The system displays the following output:

```
declustered   needs                vdisks       pdisks           capacity
   array     service  type  trim  user log  total spare rt  total raw free raw  background
task
-----------  -------  ----  ----  ---- ---  ----- ----- --  --------- --------
---------------
```

```
DA1          no       NVMe  no      10   0     12     2  2   8869 GiB 1237 GiB  rebalance
(12%)
DA2          yes      HDD   no      10   0     18     2  2   8829 GiB 1089 GiB  rebalance
(88%)
DA3          no       SSD   no      10  11     24     3  2   9173 GiB  695 GiB  rebalance
(19%)

mmvdisk: Total capacity is the raw space before any vdisk set definitions.
mmvdisk: Free capacity is what remains for additional vdisk set definitions.

mmvdisk: Attention: Recovery group 'rg_1' has an incomplete node addition (c72f4m5u15-ib0).
mmvdisk: callback 'RG001CompletNodeAdd' will perform the node addition after rebalance
completes
mmvdisk: in all declustered arrays of recovery group 'rg_1'.
```

After the call-back is executed, the above **mmvdisk** command message will disappear.

6. Run the following command that would display an increased vdisks number.

```
# mmvdisk rg list --rg rg_1 --da
```

The system displays the following output:

```
declustered   needs                vdisks       pdisks           capacity
   array     service  type  trim  user log  total spare rt  total raw free raw  background
task
-----------  -------  ----  ----  ---- ---  ----- ----- --  --------- --------
---------------
DA1          no       NVMe  no      12   0     12     2  2   8869 GiB 1237 GiB  scrub 14d
(63%)
DA2          yes      HDD   no      12   0     18     2  2   8829 GiB 1089 GiB  scrub 14d
(63%)
DA3          no       SSD   no      12  13     24     3  2   9173 GiB  695 GiB  scrub 14d
(65%)

mmvdisk: Total capacity is the raw space before any vdisk set definitions.
mmvdisk: Free capacity is what remains for additional vdisk set definitions.
```

Replacing an I/O node with a new node and disks

In this scenario, a failed server is to be replaced with an entirely new server, including new drives.

1. Prepare a new node with the same disk topology as the node needs to be replaced. The server type, memory, and disks must be same.
2. Issue the **mmaddnode** command to add this node into the IBM Storage Scale, accept the license as the server, and issue the **mmstartup -N** command to bring up the IBM Storage Scale daemon.
3. Define the node as the same role as the old server, such as quorum, `fsmgr`, and so on.
4. Run the **mmvdisk server configure -N nodename** command to configure the node, then restart the daemon on this node.
5. Run the **mmvdisk rg replace** command to replace the existing node with a new node. In some cases, you might need to specify **--match** parameter if there are slight differences between your configuration and the standard topology definitions, for example **--match 90**.

```

mmvdisk rg replace --rg rg1 -N c72f4m5u01-ib0 --new-node c72f4m5u07-ib0
mmvdisk: Attempting to complete a previous replace command.
mmvdisk: Analyzing disk topology for node 'c72f4m5u01-ib0'.
mmvdisk: Analyzing disk topology for node 'c72f4m5u03-ib0'.
mmvdisk: Analyzing disk topology for node 'c72f4m5u05-ib0'.
mmvdisk: Analyzing disk topology for node 'c72f4m5u11-ib0'.
mmvdisk: Analyzing disk topology for node 'c72f4m5u09-ib0'.
mmvdisk: Analyzing disk topology for node 'c72f4m5u15-ib0'.
mmvdisk: Analyzing disk topology for node 'c72f4m5u13-ib0'.
mmvdisk: Analyzing disk topology for node 'c72f4m5u07-ib0'.
mmvdisk: Updating server list for recovery group 'rg1'.
mmvdisk: Updating pdisk list for recovery group 'rg1'.
mmvdisk: This could take a long time.
mmvdisk: The following pdisks will be formatted on node c72f4m5u01.gpfs.net:
mmvdisk:     //c72f4m5u07-ib0/dev/nvme1n1
mmvdisk:     //c72f4m5u07-ib0/dev/nvme0n1
mmvdisk:     //c72f4m5u07-ib0/dev/sda
```

```
mmvdisk:     //c72f4m5u07-ib0/dev/sdc
mmvdisk:     //c72f4m5u07-ib0/dev/sdb
mmvdisk:     //c72f4m5u07-ib0/dev/sde
mmvdisk:     //c72f4m5u07-ib0/dev/sdg
mmvdisk:     //c72f4m5u07-ib0/dev/sdf
mmvdisk:     //c72f4m5u07-ib0/dev/sdd
mmvdisk: Removing node 'c72f4m5u01-ib0' from node class 'r1'.
mmvdisk: Updating server list for recovery group 'rg1'.
```

6. Run the **`mmvdisk rg list`** command to make sure that the new node joins the node class, and that all related `pdisks` work fine. Also, make sure that the replaced node and the related `pdisks` are not in the RG anymore. Then, wait for some time to make sure all DAs are into the scrub state.
7. Now the node is replaced from RG successfully. Run **`mmshutdown -N and mmdelnode -N`** to delete the replaced node from the cluster (if you do not need the node in the cluster anymore).

Replacing broken I/O nodes with moving disks to new nodes

1. Make sure that the node is broken, not pingable, or cannot be logged in. You can pull the network cable on this broken node if you can physically access the node.
2. Prepare a new node that is of the same hardware as that of the broken node.
3. Install the same OS on it, check the time to sync with all other nodes in IBM Storage Scale Erasure Code Edition cluster, and then install the same IBM Storage Scale build on the new node.
4. Connect the new node to the switch, change the hostname and IP address of the node as that of the old node.
5. Pull the pdisks that the old node was using and insert them into the new node.
6. Make sure that all disks are visible on the new node and that none of the pdisks are broken. If the pdisks are broken, data in this disk never gets restored.
7. Make sure that the *ssh* and *scp* commands work on the new node. You must configure passwordless ssh and scp for root users.
8. Make sure that *ssh/scp* works between ALL nodes and the new node.
9. Issue the **`mmsdrrestore -p <node name> -R /usr/bin/scp`** command on the new node, where *<node name>* is one of the active nodes in the node class.

Maintenance of a node

This topic describes the procedures for taking off an IBM Storage Scale Erasure Code Edition node from service for maintenance.

For the general procedure for maintenance of an IBM Storage Scale Erasure Code Edition node, follow the steps that are described in "Manual online upgrade of IBM Storage Scale Erasure Code Edition" on page 65 and replace the step "6. Upgrade the IBM Storage Scale Erasure Code Edition software" with "Operations of node maintenance".

Manually online upgrade OS/driver for IBM Storage Scale Erasure Code Edition node

1. Follow the upgrading steps 1 through 5 as described in "Manual online upgrade of IBM Storage Scale Erasure Code Edition" on page 65.
2. Disable GPFS autostart by issuing the following command:

   ```
   mmchconfig autoload=no
   ```

3. Upgrade the OS or the driver.

   Note: If you have Mellanox OFED driver installed, uninstall it before the OS upgrade, and install the new driver according to the upgraded OS version. You might need to restart the node after the upgrade.
4. After the upgrade, with the system that runs the new kernel or driver, issue the following command to build the GPFS portability layer:

```
mmbuildgpl
```

5. Perform steps 7 through 9 as described in “Manual online upgrade of IBM Storage Scale Erasure Code Edition” on page 65.
6. After all the nodes are upgraded, change the autoload config back to "yes".

   ```
   mmchconfig autoload=yes
   ```

Replace SAS controller or adapter

This topic describes the procedures for replacing SAS controller card or adapter.

Perform the following steps:

1. Follow steps 1 through 5 as described in “Manual online upgrade of IBM Storage Scale Erasure Code Edition” on page 65.

   Note: The same type of card or adapter is recommended for replacement. Card or adapter that has a different speed might introduce performance issues.

   If the SAS controller need replacement is failed and caused the disks that belong to the controller are missing, the recovery group fault tolerance might be degraded. If the left fault tolerance is lower than one node + one pdisk, contact IBM support to make sure you can shut down the node online.
2. Disable GPFS autostart by issuing the following command:

   ```
   mmchconfig autoload=no
   ```
3. Shut down and power off the node.
4. Follow instructions of the server manufacturer to replace the SAS card.
5. Power® on and then start the node.
6. Check the disks with **storcli** or **perccli** command and set the disk in JBOD mode if needed.
7. Verify the disk slot location by following the steps in “Mapping LMR disk location” on page 36.
8. Check the disk write cache and disable it if needed. For more information, see “Hardware checklist” on page 12.
9. Run the OS precheck tool again and make sure that the test is passed.
10. Change the autoload config back to "yes".

    ```
    mmchconfig autoload=yes
    ```
11. Perform steps 7 through 9 as described in “Manual online upgrade of IBM Storage Scale Erasure Code Edition” on page 65.

Firmware updates

In IBM Storage Scale Erasure Code Edition, it is the customer’s responsibility to ensure that the firmware and operating system software are kept current. The following procedures are meant as a model, but exact procedures might vary depending on your hardware configuration.

- **HBA firmware update**:

  Follow the steps that are described in “Manual online upgrade of IBM Storage Scale Erasure Code Edition” on page 65 and replace the step "6. Upgrade the IBM Storage Scale Erasure Code Edition software" with "Operations of upgrading HBA firmware".

  Here is an example for upgrading firmware of Server RAID 5110e:

  ```
  #./ibm_fw_sraidmr_6gb-23.34.0-0023_linux_32-64.bin -s
  Running in 64 bit mode.
  ```

```
*************************************************
 LSI MR Update Utility for use with IBM hardware
       Version 1.39 - Release Date 8/11/11
*************************************************

This update is for the following controllers:
- ServeRAID M5120 - ServeRAID M5110e for System x3650 M4 - ServeRAID M5110
- ServeRAID M5016 - ServeRAID M5115 - IBM Flex System Storage Expansion Node
Found 1 ServeRAID M, MR or MegaRAID Controller(s)
Getting configuration for Controller 0.  Please wait....
Attempting to flash controller 0!
Updating Controller 0.  Please wait....
./MegaCLI -AdpFwFlash -f lsi2208.rom -a0 > result.out
Update of controller 0 completed successfully.
Successfully flashed controller 0!
You must reboot your system to complete the firmware update process.
You do not need to reboot your system immediately.
reboot the node gpfstest2
```

- **To update disk firmware on one node**:

  User can use the **`mmlsfirmware`** command to check the current disk firmware version.

  – Case 1: Upgrade disk firmware one by one

    If you have a firmware image, which allows you to upgrade disk firmware one by one:

    1. Issue the **`mmvdisk pdisk change --rg rgname --pdisk pdiskname -suspend`** command to suspend one `pdisk`.
    2. Issue the external tool to update disk firmware.
    3. Run the **`mmvdisk pdisk change --rg rgname --pdisk pdiskname -resume`** to resume the `pdisk`.
    4. Repeat step 1 - 3 to make sure all `pdisks` firmware gets updated.
    5. On the RG master node, issue the **`tschrecgroup --rg ALL --path-discovery enable`** command to trigger GNR load new firmware level for all `pdisks`.

  – Case 2: Upgrade disk firmware in batch

    If the firmware upgrade tool only supports update all `pdisk` firmware together instead of upgrading firmware one by one, we need to take the node out of service, run the tools, and then bring the node back to service.

    Follow the steps that are described in “Manual online upgrade of IBM Storage Scale Erasure Code Edition” on page 65 and replace the step "6. Upgrade the IBM Storage Scale Erasure Code Edition software" with "Operations of upgrading disk firmware".

Volatile write cache detection

IBM Storage Scale Erasure Code Edition now has the ability to test if volatile write caching mode is enabled on the physical disks.

Many SCSI and NVMe drives support a volatile write caching mode in which a drive reports success back from write operations as soon as data has been received into the drive's internal cache memory. IBM Storage Scale Erasure Code Edition cannot be used with drives operating in this mode because on power failure, the cached data is lost, causing already committed data to revert to an older version. This can lead to corruption of both the RAID and file system metadata, resulting in data integrity issues.

If IBM Storage Scale Erasure Code Edition detects a drive with volatile write caching mode that is enabled, it puts the pdisk into a new volatile write cache that is enabled (VWCE) state and drains all data from the drive. If IBM Storage Scale Erasure Code Edition detects many drives with volatile write caching enabled, it stops service of the recovery group and waits for volatile write caching mode to be disabled on the drives.

The volatile write cache detection feature is enabled for all new IBM Storage Scale Erasure Code Edition installations starting from version 5.0.4. On previous installations, the feature is disabled by default and must be manually enabled in order to take advantage of the check.

ECE introduces asynchronous detection of volatile write cache in version 5.1.6. In the previous release, ECE only checked the volatile write cache mode when rediscovering disk states like RG initialization and RG master failover, or ran the command to refresh pdisk information.

Check IBM Storage Scale Erasure Code Edition cluster configuration for VWCE

IBM Storage Scale Erasure Code Edition supports volatile write cache detection from version 5.0.4, upgrade from previous versions need to enable it.

Use the following commands to check the current IBM Storage Scale Erasure Code Edition configuration for VWCE detection:

1. `# mmdiag --config|grep nsdRAIDDiskCheckVWCE`

   If `nsdRAIDDiskCheckVWCE` is 1, it means enabled. If `nsdRAIDDiskCheckVWCE` is 0, it means disabled. Check all physical disks volatile write cache state before enabling it.
2. After making sure that all disks have disabled volatile write cache, use this command to enable it:

   ```
   # mmchconfig nsdRAIDDiskCheckVWCE=yes -i
   ```
3. Rediscover the disk state with this command:

   ```
   # mmvdisk rg change --recovery-group rg_name --refresh-pdisk-info
   ```

Creation of recovery group fails if volatile write cache mode is enabled on disk

Before you install IBM Storage Scale Erasure Code Edition and create a recovery group, run the `ece_os_readiness` tool first, it detects volatile write cache of disks and give you warning messages. When disks have volatile write cache mode that is enabled, creation of recovery group fails with error messages in the `/var/adm/ras/mmfs.log.latest` file.

Asynchronous detection of volatile write cache

IBM Storage Scale Erasure Code Edition introduces asynchronous detection of volatile write cache mode in version 5.1.6. The detection interval is controlled by configuration **`nsdRAIDDiskSMARTUpdateInterval`**, the default value is 24 hours. When Recovery Group is running, ECE will rediscover the disk volatile write cache mode every 24 hours. When pdisk state changes from missing to ok, ECE will also rediscover the pdisk volatile write cache mode. The pdisk will be put into VWCE mode and drain data from it if the volatile write cache enabled is detected.

Failure of disk replacement

For replacing failure disk, check and disable volatile write cache mode for the new physical disk. If volatile write cache mode is not disabled, replace command would fail.

Scale out IBM Storage Scale Erasure Code Edition by adding new node

Run the `ece_os_readiness` tool first on new node, and disable volatile write cache mode for each disk if needed before you add a node into an IBM Storage Scale Erasure Code Edition recovery group.

What to do if volatile write cache is detected

For instructions on how to disable volatile writer caching on SCSI and NVMe disks, see "Hardware checklist" on page 12.

Adding new recovery group into the existing IBM Storage Scale Erasure Code Edition cluster

The newly added servers must meet the IBM Storage Scale Erasure Code Edition hardware requirements.

For more information, see " IBM Storage Scale Erasure Code Edition hardware requirements" on page 9.

Note: Any new hardware (especially disks) used for below section must be tested before put into production to make sure the hardware quality.

Use the following steps to add a new recovery group into the existing IBM Storage Scale Erasure Code Edition cluster.

1. From IBM Fix Central, download the IBM Storage Scale Advanced Edition 5.x.y.z installation package. The version of this package must match with the version of IBM Storage Scale Erasure Code Edition cluster. You must download this package to the node that you plan to use as your installer node for the IBM Storage Scale Advanced Edition installation and the subsequent IBM Storage Scale Erasure Code Edition installation. Also, use a node that you plan to add in the existing IBM Storage Scale Erasure Code Edition cluster.
2. Extract the IBM Storage Scale Advanced Edition 5.x.y.z installation package to the default directory or a directory of your choice on the node that you plan to use as the installer node.

   ```
   /DirectoryPathToDownloadedCode/Spectrum_Scale_Advanced-5.x.y.z-x86_64-Linux-install --text-
   only
   ```

3. Change the directory to the default directory for the installation toolkit.

   ```
   # cd /usr/lpp/mmfs/5.x.y.z/ansible-toolkit/
   ```

4. Set up the installer node by using the following command:

   ```
   # ./spectrumscale setup -s 192.0.2.6 -st ece
   ```

   Note: In this command example, `192.0.2.6` is the IP address of the scale-out node that is planned to be designated as the installer node.
5. Issue the config populate command to populate the existing IBM Storage Scale Advanced Edition cluster configuration.

   ```
   # ./spectrumscale config populate ece-node2
   ```

   In this command example, `ece-node2` is the IBM Storage Scale Advanced Edition recovery group server.
6. Add the IBM Storage Scale Advanced Edition candidate nodes.

   ```
   # ./spectrumscale node add 192.0.2.6 -so
   # ./spectrumscale node add 192.0.2.7 -so
   # ./spectrumscale node add 192.0.2.8 -so
   # ./spectrumscale node add 192.0.2.9 -so
   ```

7. Verify the nodes.

   ```
   # ./spectrumscale node list

   [ INFO  ] List of nodes in current configuration:
   [ INFO  ] [Installer Node]
   [ INFO  ] 192.0.2.6
   [ INFO  ]
   [ INFO  ] [Cluster Details]
   [ INFO  ] Name: ece-node8
   [ INFO  ] Setup Type: Erasure Code Edition
   [ INFO  ]
   [ INFO  ] [Extended Features]
   [ INFO  ] File Audit logging     : Disabled
   [ INFO  ] Watch folder           : Disabled
   [ INFO  ] Management GUI         : Disabled
   [ INFO  ] Performance Monitoring : Enabled
   [ INFO  ] Callhome               : Disabled
   ```

```
[ INFO  ]
[ INFO  ] GPFS                                 Admin  Quorum  Manager   NSD   Protocol
Perf Mon  Scale-out   OS   Arch
[ INFO  ] Node                                  Node   Node     Node   Server   Node
Collector    Node
[ INFO  ] ece-node1                                      X
X                rhel7  x86_64
[ INFO  ] ece-node2                              X       X
X                                    X     rhel7  x86_64
[ INFO  ] ece-node3                                      X
X                                    X     rhel7  x86_64
[ INFO  ] ece-node4
X                                    X     rhel7  x86_64
[ INFO  ] ece-node5
X                                    X     rhel7  x86_64
[ INFO  ] ece-
node6
X     rhel7  x86_64
[ INFO  ] ece-
node7
X     rhel7  x86_64
[ INFO  ] ece-
node8
X     rhel7  x86_64
[ INFO  ] ece-
node9
X     rhel7  x86_64
[ INFO  ]
[ INFO  ] [Export IP address]
[ INFO  ] No export IP addresses configured
```

8. Perform an installation pre-check.

```
# ./spectrumscale install -pr
```

9. Run the installation procedure.

```
# ./spectrumscale install
```

10. If needed, change the quorum node after the new nodes are added in.

```
# mmchnode --nonquorum -N ece-node2
```

```
# mmchnode --quorum -N ece-node6
```

11. Verify the disk slot location of the new server. For more information, see “Mapping NVMe disk slot location” on page 35 and “Mapping LMR disk location” on page 36.
12. Define the new recovery group with the newly added nodes.

```
# ./spectrumscale recoverygroup define -N ece-node6,ece-node7,ece-node8,ece-node9
```

You can verify the defined recovery groups by running the following command:

```
# ./spectrumscale recoverygroup list
```

13. Run the installation procedure again.

```
# ./spectrumscale install
```

14. Check new recovery group information.

```
# mmvdisk rg list
```

15. Define a new vdiskset by using one of the following methods:

- Copy existing RG's vdiskset configuration to the new vdisks.

```
# mmvdisk vdiskset define --vs VS02 --copy VS01 --rg rg_2
```

In this command example, VS01 is the existing vdiskset.

- Define it specifically.

```
# mmvdisk vs define --vs VS02 --rg rg_2 --code 8+2p --bs 8M --da DA1 --set-size 90%
```

- Define it by specifying a new data pool.

```
# mmvdisk vs define --vs VS02 --rg rg_2 --code 8+2p --bs 8M --da DA1 --set-size 90%
--nsd-usage dataonly --sp data2
```

16. Create new vdisks.

    ```
    # mmvdisk vs create --vs all
    ```

17. For file system operations, do either of the following steps:

    - Create a file system.

      ```
      # mmvdisk fs create --fs gpfs2 --vs VS02
      ```

    - Add vdiskset into the file system.

      ```
      # mmvdisk fs add --fs gpfs1 --vs VS02
      ```

18. If needed, restripe the file system.

    Note: For more information about adding vdiskset into the existing file system, see *Modifying file system attributes*, *Restriping a GPFS file system*, and *Changing GPFS disk parameters* topics in the *IBM Storage Scale: Administration Guide*.

Support for TRIM procedures

IBM Storage Scale Erasure Code Edition supports the TRIM feature to enable space reclamation.

Supported Configuration for TRIM

IBM Storage Scale Erasure Code Edition supports manual reclamation of free space for NVMe-based NSDs in 5.0.5.1 or later.

IBM Storage Scale Erasure Code Edition supports automatic reclamation of free space for NVMe-based NSDs in 5.1.5 or later.

For more information, see the topic Managing TRIM support for storage space reclamation in the *IBM Storage Scale RAID: Administration and Programming Reference* Guide.

Note: Before TRIM is enabled in production, the following requirements must be met:

1. Disks that support TRIM need to meet the specified requirements. For more information, see “Selecting physical disks for TRIM” on page 15.
2. System stability must be verified by running TRIM with a mixed file system workload (including read, write, and free up file space). In the test, the pdisk’s state must be kept in OK state to sustain recovery group and file system service. Only the tested firmware version can be applied in the production system.
3. Automatic background reclaim is controlled with the config parameter **backgroundSpaceReclaimThreshold**, which may be set via the **mmchconfig** command. In Erasure Code Edition, this value may be set to 15 to enable the feature. Different backend storage devices respond to reclaim commands differently, and this may result in an impact on write performance. Before enabling Automatic Background Reclaim, it is recommended that you verify the performance of workloads that create and write to new files while deleting files at the same time. For more information, see the topic Automatic background TRIM in the *IBM Storage Scale RAID: Administration and Programming Reference* Guide.

Adding new disks in the declustered array of the recovery group

This topic describes the procedure for adding new disks in declustered array of recovery group. The newly added disks must meet the IBM Storage Scale Erasure Code Edition hardware requirements and considerations.

For more information, see "Planning for recovery group space and scale up" on page 25.

Note: Any new hardware (especially disks) used for below section must be tested before put into production to make sure the hardware quality.

Note: The following steps show the process of adding one disk per node in the recovery group. It is recommended to add at least two disks per node each time.

Use the following steps to add new disks in the declustered array of the recovery group.

1. Verify the current disk topology of the server by running the following command:

   ```
   # mmvdisk server list --nc nc_1 --disk-topology
   ```

   The system displays the following output:

   ```
    node                                       needs    matching
   number  server                            attention   metric   disk topology
   ------  --------------------------------  ---------  --------  -------------
        5  node5.ibm.com                     no          100/100  ECE 6 SSD/NVMe and 2 HDD
        1  node1.ibm.com                     no          100/100  ECE 6 SSD/NVMe and 2 HDD
        2  node2.ibm.com                     no          100/100  ECE 6 SSD/NVMe and 2 HDD
        3  node3.ibm.com                     no          100/100  ECE 6 SSD/NVMe and 2 HDD
        4  node4.ibm.com                     no          100/100  ECE 6 SSD/NVMe and 2 HDD
   ```

2. Check the details of the declustered array that needs to be expanded.

   In the following example, DA2 is the declustering array (DA) that would be expanded. DA2 currently has 10 disks.

   ```
   # mmvdisk rg list --da
   ```

   The system displays the following output:

   ```
                   declustered   needs                       capacity             pdisks
   recovery group     array     service  type  trim  total raw free raw free%  total spare
   background task
   --------------  -----------  -------  ----  ----  --------- -------- -----  ----- -----
   ---------------
   rg_1            DA1          no       NVMe  yes    7095 GiB  725 GiB   10%     10     2
   scrub (41%)
   rg_1            DA2          no       HDD   no     4414 GiB  459 GiB   10%     10     2
   scrub (41%)
   rg_1            DA3          no       SSD   no     7866 GiB  796 GiB   10%     20     2
   scrub (39%)

   mmvdisk: Total capacity is the raw space before any vdisk set definitions.
   mmvdisk: Free capacity is what remains for additional vdisk set definitions.
   ```

3. Insert new disks into the slots on each server.

   ```
   # mmvdisk server list --nc nc_1 --disk-topology
   ```

   The system displays the following output:

   ```
    node                                       needs    matching
   number  server                            attention   metric   disk topology
   ------  --------------------------------  ---------  --------  -------------
        5  node5.ibm.com                     no          100/100  ECE 6 SSD/NVMe and 3 HDD
        1  node1.ibm.com                     no          100/100  ECE 6 SSD/NVMe and 3 HDD
        2  node2.ibm.com                     no          100/100  ECE 6 SSD/NVMe and 3 HDD
        3  node3.ibm.com                     no          100/100  ECE 6 SSD/NVMe and 3 HDD
        4  node4.ibm.com                     no          100/100  ECE 6 SSD/NVMe and 3 HDD
   ```

4. Verify that the slot location of the newly inserted disks is correct. For more information, see "Mapping NVMe disk slot location" on page 35 and "Mapping LMR disk location" on page 36.
5. Add new disks in the declustered array by running the following command:

```
# mmvdisk rg resize --rg rg_1
```

```
mmvdisk: Obtaining pdisk information for recovery group 'rg_1'.
mmvdisk: Analyzing disk topology for node 'c72f4m5u19-ib0'.
mmvdisk: Analyzing disk topology for node 'c72f4m5u13-ib0'.
mmvdisk: Analyzing disk topology for node 'c72f4m5u21-ib0'.
mmvdisk: Analyzing disk topology for node 'c72f4m5u17-ib0'.
mmvdisk: Analyzing disk topology for node 'c72f4m5u11-ib0'.
mmvdisk: Validating existing pdisk locations for recovery group 'rg_1'.
mmvdisk: The resized server disk topology is 'ECE 6 SSD/NVMe and 3 HDD'.
mmvdisk: Validating declustered arrays for recovery group 'rg_1'.
mmvdisk: Adding new pdisks to recovery group 'rg_1'.
mmvdisk: Updating declustered array attributes for recovery group 'rg_1'.
mmvdisk: Successfully resized recovery group 'rg_1'.
```

6. Check the declustered array information.

   In this example, DA2 has 15 disks now.

```
# mmvdisk rg list --da
```

   The system displays the following output:

```
                declustered   needs                            capacity             pdisks
recovery group     array     service  type  trim  total raw free raw free%    total spare
background task
--------------  -----------  -------  ----  ----  --------- -------- -----  ----- -----
---------------
rg_1            DA1          no       NVMe  yes    7095 GiB  725 GiB   10%     10     2
scrub (46%)
rg_1            DA2          no       HDD   no     7174 GiB 3219 GiB   44%     15     2
scrub (46%)
rg_1            DA3          no       SSD   no     7866 GiB  796 GiB   10%     20     2
scrub (45%)

mmvdisk: Total capacity is the raw space before any vdisk set definitions.
mmvdisk: Free capacity is what remains for additional vdisk set definitions.
```

7. Use **mmvdisk** command to create vdisks and NSDs as required.

   For more information on the management of the recovery groups, see the topic Recovery group management in the *IBM Storage Scale RAID: Administration and Programming Reference* Guide.

Self-encrypting drive support

IBM Storage Scale Erasure Code Edition supports the migration of an existing recovery group to enable a self-encrypting drive (SED).

Support migration of a recovery group by enabling SED online

The existing recovery groups can be enabled for the SED support through a process called migration. The migration can be done on live system when a workload is in-progress.

Notes:

- A recovery group only with NVMe drives (PCIe attached) are supported.
- All the drives in a recovery group must be SED capable to support this feature.
- This feature with the drive hardware must be tested before the production.
- The MegaRaid Controller is not supported. The use of SED on drives, which are attached to MegaRAID HBA, cause unrecoverable data lost when a drive is power-recycled.
- Configuration of a remote key manager (RKM) server must be backed up, and restore it when the recovery is needed.

Enabling the SED online on a recovery group

The self-encrypting drive support can be enabled on existing recovery groups.

A remote key manager (RKM) server is needed. For requirements reference, see the mmkeyserv command in the *IBM Storage Scale: Command and Programming Reference Guide*.

Complete the following steps to enable the SED support on all drives of a recovery group.

1. By using the **mmkeyserv** command, configure the master encryption key (MEK) and find a Key ID and an RKM ID.

   a. Add a key server to a cluster.

   ```
   # mmkeyserv server add keyserver01
   ```

   b. Create a tenant in the key server.

   ```
   # mmkeyserv tenant add ece01 --server keyserver01
   ```

   c. Create a key client on the key server.

   ```
   # mmkeyserv client create cluster01 --server keyserver01
   ```

   d. Register the key client to the tenant with the RKM ID.

   ```
   # mmkeyserv client register cluster01 --tenant ece01 --rkm-id keyserver01_ece01
   ```

   e. Create an encryption key in the tenant (It returns a Key ID.).

   ```
   # mmkeyserv key create --server keyserver01 --tenant ece01
   ```

2. Ensure that all the drives of the recovery group are in the OK state.

   ```
   # mmvdisk pdisk list --rg rg1 --not-ok
   ```

3. Verify that all drives of recovery group are SED-capable by using the **mmvdisk sed verify** command.

   ```
   # mmvdisk sed verify --rg rg_name
   ```

4. Run the **mmvdisk sed enroll** command by passing the Key ID and RKM ID mentioned in step 1 to enroll the RG with the SED that is enabled.

   ```
   # mmvdisk sed enroll --recovery-group rg_name --rkmid keyserver01_ece01 --key-uuid KeyId
   ```

   The **sed enroll** command performs the following steps:

   a. Stores the new <KeyId:RKMid> value in the sedKeyId config variable of node class.

   b. Updates the MEK key from default MSID to the new key specified on all drives, and enable SED support on all drives.

5. Verify that the drives are enrolled.

   ```
   # mmvdisk sed list --recovery-group rg_name
   ```

   Notes:

   - If the enroll process is stopped or interrupted, the **sed enroll** command can be issued to resume the migration.
   - For replacing a failed disk with a new disk, scale-up of adding new disks into a recovery group or scale-out of adding new nodes into a recovery group, complete the following steps:

   a. To verify whether all the new drives are enrolled with an sedKey after the enroll process, issue the following command:

      ```
      # mmvdisk sed list --recovery-group rg_name
      ```

   b. Run the following command to enroll the drives if any drives are not enrolled:

      ```
      # mmvdisk sed enroll
      ```

   For more information, see the mmvdisk sed command in *IBM Storage Scale RAID: Administration and Programming Reference*.

Changing an SED key of a recovery group

For an SED-enabled recovery group, you can change the MEK key.

Complete the following steps to change the MEK key:

1. Create an encryption key in the tenant (It shows a Key ID).

   ```
   # mmkeyserv key create --server keyserver01 --tenant ece01
   ```

2. Ensure that all the drives of recovery groups are in the OK state.

   ```
   # mmvdisk pdisk list --rg rg_name --not-ok
   ```

3. Update the MEK key of a recovery group by using the new Key ID.

   ```
   # mmvdisk sed rekey --recovery-group RgName --rkmid keyserver01_ece01 --key-uuid New_KeyId
   ```

4. Verify the new Key ID with an RKM ID.

   ```
   # mmlsconfig sedKeyID
   ```

5. Verify that the drives are all unlocked correctly.

   ```
   # mmvdisk sed list --recovery-group rg_name
   ```

   Note: If the rekey process is stopped or interrupted, the **sed rekey** command can be issued to resume this process.

   For more information, see mmvdisk sed command in *IBM Storage Scale RAID: Administration and Programming Reference*.

Chapter 13. Troubleshooting

This topic describes the known issues and workarounds of IBM Storage Scale Erasure Code Edition.

Monitoring the overall health

This topic describes different methods to monitor and troubleshoot IBM Storage Scale Erasure Code Edition.

For monitoring:

- From GUI, see the *Monitoring system health using IBM Storage Scale GUI* topic in the *IBM Storage Scale: Administration Guide*.
- From command line, see the *Monitoring system health by using the mmhealth command* topic in the *IBM Storage Scale: Administration Guide*.
- For general IBM Storage Scale troubleshooting, see the Troubleshooting topic in the *IBM Storage Scale: Problem Determination Guide*.
- For IBM Storage Scale RAID troubleshooting best practices, see the *Best practices for troubleshooting* topic in the *IBM Storage Scale RAID: Administration*.

What to do if you see degraded performance over NSD protocol

This topic describes the issues relating to degraded performance over NSD protocol.

Compared degraded performance to what? Is there a repeatable test and a baseline to compare to? "It is slow" is not a valid measurable metric. You must have a baseline to compare to.

There are multiple tools to create that. The product includes `nsdperf`, but you can choose other available tools in the market such as `ior`, `iozone`, `bonnie++`.

Things to check:

- First, check the network end to end.
- Review any changes that are done to either the clients or servers (sysctl, software updates)
- Check OS resources on the client system (CPU, memory, swap in and out)
- Check OS resources on the server system.
- Look for **`mmhealth`** events.
- Look for SMART events (if applicable).
- Restart the client.

If you still see degraded performance compared to your baseline with the repeatable test, it is time to gather some information and contact IBM, as follows:

- Generate an IBM Storage Scale snap on IBM Storage Scale Erasure Code Edition cluster.
- Generate an IBM Storage Scale snap on the client cluster.

You can already contact IBM support with the above snaps. If you suspect any issues at the disks level, you must engage with the disk vendor tools. In addition, you might gather the following information and attach to the IBM case.

ICT (intercompletion time) data is a full I/O trace that gives size, seek distance, LBA, queue depth at time of completion, overall response time of the I/O and the completion time of this I/O relative to the previous or relative to the start of the I/O, whichever is later, for each `pdisk` I/O request. Things to look for would be the distribution of the ICT times, comparison of the response time to ICT time, and so on. And checking whether anomalies are specific to hardware domains or to particular ranges of time. This data can be useful to IBM support to help determine many different types of issues.

When you contact IBM support, compile the following data in addition to your baseline and the results that you obtain that differ from the baseline. Also, include an overview of the environment and the tools as versions used to create the baseline:

- Gather ICT debug data:
  - Create a directory to host the debugs. You can use NFS or separate disk, as it can generate a fair amount of data. In the following example, `/tmp/mmfs/ict` is used:

    ```
    # mkdir /tmp/mmfs/ict
    ```

  - Enable the gather of ICT data on IBM Storage Scale Erasure Code Edition node:

    ```
    #mmchconfig nsdRAIDICTLogDir=/tmp/mmfs/ict,nsdRAIDDetailedICTLogging=all -N NODE i
    ```

  - Once you re-created the performance degradation against the baseline, set the login back to default and tar the information to be sent to IBM:

    ```
    #  mmchconfig nsdRAIDICTLogDir=default,nsdRAIDDetailedICTLogging=default -N NODE -i
    ```

    ```
    # tar -czf ict.tgz -C /tmp/mmfs ict
    ```

  - Attach the compressed file to the IBM case.
- Unbalance of vdisk partition distribution:
  - Add the output of the following command from IBM Storage Scale Erasure Code Edition nodes to a text file and add it to the IBM case.

    ```
    # /usr/lpp/mmfs/bin/mmfsadm test vdisk vdDist 1
    ```

What to do if you see degraded performance over CES with NFS and/or SMB

This topic describes the procedure to troubleshoot any issues relating to degraded performance over CES with NFS or SMB.

Compared degraded performance to what? Is there a repeatable test and a baseline to compare to? "It is slow" is not a valid measurable metric. You should have a baseline to compare to. It is also important to identify if the issue is only reproducible on CES-served protocols instead of on NSD protocol. If it is also reproducible on the NSD protocol, see "What to do if you see degraded performance over NSD protocol" on page 109.

There are multiple tools to create that. The product includes `nsdperf`, but you can choose to use other tools available in the market such as `ior`, `iozone`, `bonnie++`.

Whichever tool that you choose when you deploy the system, use the same tool to compare against baseline. Mention the tool you used and the results of baseline and the current results when you contact IBM support.

Things to check:

- First and foremost, check the network end to end.
- Review any changes done to either the clients or servers (syscall).
- Check OS resources on the client system (CPU, memory, swap in and out).
- Check OS resources on the server system.
- Look for **mmhealth** events.
- Look for SMART events (if applicable).
- Reboot the client.

If you still see degraded performance compared to your baseline with the repeatable test, it is time to gather some information and contact IBM support with the following data.

For detailed information, see the *CES tracing and debug data collection* topic in the *IBM Storage Scale: Problem Determination Guide*.

- Generate an IBM Storage Scale snap on the CES cluster. Use `--performance` and `--protocol` with the protocol of interest (NFS or SMB or NFS, SMB).
- Gather protocol traces:
  - For SMB:
    - Start the traces from a CES node that serves SMB.

      ```
      #  mmprotocoltrace start smb -c <clientIP>
      ```

    - You can check the status of the trace as well as the output files with:

      ```
      # mmprotocoltrace status smb
      ```

    - Once the problem has been reproduced from the client, stop the traces and send the files to IBM:

      ```
      # mmprotocoltrace stop smb
      ```

  - For NFS:
    - NFS traces are obtained by changing the log level to `FULL_DEBUG`. Be aware that the change of log level will do a restart of the CES NFS daemons on all nodes and that generates a vast amount of data that might impact performance.

      ```
      # mmnfs config change LOG_LEVEL=FULL_DEBUG
      ```

    - When the issue has been reproduced from the client, gather a snap and restore to default (EVENT) log level. Be aware that the restore of the log level will trigger a restart of all CES NFS daemons.

      ```
      # gpfs.snap --protocol nfs
      ```

      ```
      # mmnfs config change LOG_LEVEL=EVENT
      ```

Monitoring NVMe Devices

You can monitor the health of any NVMe drives in your system using the **`mmlsnvmestatus`** command. You can monitor the status of all devices or a specific device, specified by serial number.

For each NVMe device, the **`mmlsnvmestatus`** command will identify any devices where the link status does not match the link capabilities (speed and width). Additionally, it will identify any devices where the device LBA format is not one of the designated “best” formats for that device.

This example shows the output of the command on a 4-server system:

```
mmlsnvmestatus all
                                             Optimal     Optimal      needs
 node       NVMe device    serial number     Link State  LBA Formats service
 ------     -----------    -------------     ----------  -----------  ------
 node1      /dev/nvme0     57L0A03LTZ5D      NO          YES          NO
 node1      /dev/nvme1     57L0A03KTZ5D      YES         YES          NO
 node2      /dev/nvme0     57M0A01GTZ5D      YES         NO           NO
 node2      /dev/nvme1     57M0A01JTZ5D      YES         YES          NO
 node3      /dev/nvme0     57M0A00UTZ5D      YES         YES          NO
 node3      /dev/nvme1     57M0A00KTZ5D      YES         YES          NO
 node4      /dev/nvme0     57M0A019TZ5D      YES         YES          NO
 node4      /dev/nvme1     57M0A00QTZ5D      YES         YES          NO
```

You can pass the --*not-ok* flag example to only return devices with Link State or LBA Format that is not optimal. For example:

```
mmlsnvmestatus all --not-ok
                                             Optimal     Optimal      needs
 node       NVMe device    serial number     Link State  LBA Formats service
 ------     -----------    -------------     ----------  -----------  ------
 node1      /dev/nvme0     57L0A03LTZ5D      NO          YES          NO
 node2      /dev/nvme0     57M0A01GTZ5D      YES         NO           NO
```

In this example, the NVMe device on node1 is shown to have "Optimal Link State" value of "NO". This is likely due to device not being seated properly in PCIe slot. You can see more details by comparing at the *LnkCap* and *LnkSta* output of **lspci** command for this device. The NVMe device on node1 is shown to have "Optimal LBA Formats" value of "NO". You can view the available format values and the current in use value with the **nvme id-ns** command for the NVMe device.

Monitoring the endurance of SSD Devices

You can monitor the endurance of the SSD drives in your system by using the **mmhealth** command.

An SSD or physical disk has a finite lifetime based on the number of drive writes per day. The SSD endurance is a number between 0 and 255. The **ssd-endurance-percentage** value indicates the percentage of life that is used by the drive. The value 0 indicates that full life remains, and 100 indicates that the drive is at or past its end of life. When the endurance number exceeds this threshold, the **mmhealth** command displays a ssd_endurance_warn warning with the specific physical disk name and the recovery group name information. The drive must be replaced when the value exceeds 100, and the state of its health is reported as DEGRADED by the **mmhealth** command.

Issue the following command to display the health status of the **NATIVE_RAID** component:

```
[root@client21 ~]# mmhealth node show NATIVE_RAID
```

If the endurance number exceeds 100, the system gives an output similar to the following:

```


Node name:      client21.sonasad.almaden.ibm.com

Component             Status        Status Change     Reasons
------------------------------------------------------------------------------------------------
----------------
NATIVE_RAID           DEGRADED      Now               ssd_endurance_warn(rg1/n001p013)
  ARRAY               HEALTHY       Now               -
  NVME                HEALTHY       1 hour ago        -
  PHYSICALDISK        DEGRADED      Now               ssd_endurance_warn(rg1/n001p013)
  RECOVERYGROUP       HEALTHY       Now               -
  VIRTUALDISK         HEALTHY       Now               -
```

You can replace the SSD physical disk to resolve this warning message. After the SSD is replaced, issue the **mmhealth** command as shown to check the health status of the SSD:

```
[root@client21 ~]# mmhealth node show NATIVE_RAID
```

After the issue is resolved the system gives an output similar to the following:

```
Node name:      client21.sonasad.almaden.ibm.com

Component             Status        Status Change     Reasons
--------------------------------------------------------------------
NATIVE_RAID           HEALTHY       Now               -
  ARRAY               HEALTHY       Now               -
  NVME                HEALTHY       1 hour ago        -
  PHYSICALDISK        HEALTHY       Now               -
  RECOVERYGROUP       HEALTHY       Now               -
  VIRTUALDISK         HEALTHY       Now               -
```

What to do if the disk is not in the recovery group after creation or adding node

This topic describes what needs to be done if the disk is not in the recovery group after creating the recovery group or adding a new node.

Use the following command on the server to check whether the drive is formatted. The formatted disk is not used for recovery group.

Note: Make sure that the disk you plan to add into IBM Storage Scale Erasure Code Edition recovery group is not used for other purposes.

```
# lsblk -ino NAME,TYPE,FSTYPE,MOUNTPOINT
```

The FSTYPE and `MOUNTPOINT` columns must be blank to be included in the recovery group.

What to do if the installation toolkit online upgrade process is broken with an error

This topic describes how to check the cluster state when the online upgrade process breaks with an error of the installation toolkit.

When you perform an online upgrade of IBM Storage Scale Erasure Code Edition cluster by using the installation toolkit, the toolkit temporarily suspends the node that needs to be upgraded. The toolkit resumes the node after it is upgraded. In some cases where cluster problems occur, the upgrade process breaks before the node resumes.

Perform the following steps to check the cluster state and resume the node manually if needed:

1. Check whether the recovery group has a suspended node.

   ```
   # mmvdisk rg list --not-ok
   ```

   A sample output is as follows.

   ```
   recovery group  remarks
   --------------  -------
   rg_1            server ece01-ib0 'down/suspended'
   ```

   ```
   # mmvdisk rg list --rg rg_1 --server
   ```

   The system displays an output similar to the following example:

   ```
    node
   number  server                            active   remarks
   ------  --------------------------------  -------  -------
        6  ece06-ib0                         yes      serving rg_1: LG009, LG012
        1  ece01-ib0                         no       configured, suspended
        2  ece02-ib0                         yes      serving rg_1: LG002, LG006, LG010
        3  ece03-ib0                         yes      serving rg_1: LG004, LG011
        4  ece04-ib0                         yes      serving rg_1: root, LG005, LG007
        5  ece05-ib0                         yes      serving rg_1: LG001, LG003, LG008
   ```

   In the previous example, the command output shows that "ece01-ib0" node is suspended.
2. Use the following command to bring back the suspended node:

   ```
   # mmvdisk rg change --rg rg_1 --resume -N ece01-ib0
   ```

3. Check the state of the recovery group again.

   ```
   # mmvdisk rg list --not-ok
   mmvdisk: All recovery groups are ok.
   ```

What to do if creating a recovery group or adding a node command fails

This topic describes the steps that you need to take when the command for creating a recovery group or adding a node fails.

Note: The following error messages on checking slot location string are supported from 5.0.5.4 release.

When you use the installation toolkit or **mmvdisk** command to create a recovery group or add a node to the recovery group, you might get one of the following error messages:

- **Error message**: `mmvdisk: Recovery group descriptor for pdisk n014p013 of recovery group rg_2 could not be written because volatile write caching is enabled on this drive.`

  Interpretation: This error message indicates that volatile write cache is enabled on the drive.

  What action needs to be taken: Check the write cache of the drives on each node and create a recovery group again after you disable the volatile write cache. For more information, see "Hardware checklist" on page 12.

- **Error messages**: `mmvdisk: Slot location is missing from pdisk n003p013 device(s) //client23-ib0/dev/nvme0n1 of declustered array DA2 in recovery group rg1 with hardware type NVMe.`

  Interpretation: This error message indicates that the slot location is missing for NVMe drive.

  What action needs to be taken: Check the NVMe drive slot location with the **tslsenclslot** command and make any corrections if needed. For more information, see "Mapping NVMe disk slot location" on page 35.

- **Error messages**: `mvdisk: Slot location is missing from pdisk n015p012 device(s) //client28-ib0/dev/sds of declustered array DA1 in recovery group rg_2 with hardware type Rotating 10500`

  Interpretation: This error message indicates that the slot location is missing for SAS drive.

  What action needs to be taken: Check for any missing or duplicated slot location string with the **tslsenclslot** command and make any corrections if needed. For more information, see "Mapping LMR disk location" on page 36.

  Note: If the disk is not listed in the **tslsenclslot -a** command, proceed as follows:

  - Check whether all required software applications are installed. For more information, see " IBM Storage Scale Erasure Code Edition installation prerequisites" on page 27.
  - Check whether the disk is in JBOD mode. For more information, see "Hardware checklist" on page 12.

What to do if a recovery group stops service when a disk hangs because of hardware failure

The topic describes the steps that need to be taken if a recovery group stops service. A recovery group stops service when a disk hangs because of hardware failure.

There are two types of requests that are sent to a disk. One is a general I/O read or write request and the other is a pass-through query request.

IBM Storage Scale has two configuration parameters that control the response when a disk hang problem occurs. The **panicOnIOHang** parameter value is set to *yes* on storage servers by default. When a disk request hangs in the kernel longer than the defined time (300 seconds by default) of the **ioHangDetectorTimeout** parameter, IBM Storage Scale reboots the node automatically.

When the request to disk hangs, you can see long waiters as follows:

```
# mmdiag --waiters|grep NSPDServerIOWorkerThread
Waiting 159.2446 sec since 2022-04-21_06:20:57, monitored, thread 64855
NSPDServerIOWorkerThread: for I/O completion on disk sda
```

or

```
# mmdiag --waiters|grep DiscoverAndOpenNSPDThread
 Waiting 71.8359 sec since 2022-04-22_01:57:17, monitored, thread 257458
DiscoverAndOpenNSPDThread: for read SCSI world-wide name on disk /dev/sdh
```

Note: SCSI waiters might also appear on NVMe storage as SCSI-to-NVMe emulation method in GNR.

When the request hang is detected and IBM Storage Scale reboots the node, the `mmfs log` will have the following messages:

```
# cat /var/adm/ras/mmfs.log.previous |grep "Kernel I/O"
2022-04-21_06:23:38.892-0400: [E] Kernel I/O hang detected on /dev/sde: write sector 438921496
length 8 pending 305 seconds
```

or

```
# cat /var/adm/ras/mmfs.log.previous |grep "Kernel SCSI I/O"
2022-04-22_02:00:50.290-0400: [E] Kernel SCSI I/O hang detected on /dev/nvme1n1, reason: 'get
the port addresses', pending 312 seconds
```

The `vmcore-dmesg` from the crash can also be used to check the reboot reason if `vmcore` is generated:

```
# cat vmcore-dmesg.txt |grep -i "kernel panic"
[33801.657931] <5>kp 20759: cxiPanic: Forcing kernel panic to clear hung I/O
[33801.657934] Kernel panic - not syncing: cxiPanic: Forcing kernel panic to clear hung I/O
```

If the **panicOnIOHang** parameter is set to *no*, ECE will not reboot the node; it will call the user exit callback instead. The user exit callback event **diskIOHang** can be used to monitor the issue and perform the user-defined operations.

Examples

1. Create an executable script file:

   ```
   # cat /home/iohang
   #!/bin/bash
   # Adding user-defined operations here
   echo $@ > /tmp/iohang.out.`date "+%Y-%m-%d_%H_%M_%S"`
   # chmod +x /home/iohang
   ```

2. Registers user-defined command to the callback event **diskIOHang**.

   ```
   # mmaddcallback iohang --command=/home/iohang --event diskIOHang --parms "%diskName %reason"
   ```

3. When disk request hangs longer than **ioHangDetectorTimeout** parameter defined, the user exit callback will be triggered. For this example, the following file is generated:

   ```
   # cat /tmp/iohang.out.2022-04-21_06_31_42
   /dev/sda Block I/O
   ```

   or

   ```
   # cat /tmp/iohang.out.2022-04-21_02_00_14
   /dev/sdg execute SCSI command Read (10)(0x28)
   ```

   Note: The **diskName** parameter returned by callback is the device name which has the request hangs. The **reason** parameter is the request type sent to the device.

Table 10. The ***reason*** *parameter callback returned to the user executable script can have the listed types*

Parameter	**Types**
Reason	Block I/O
	read SCSI world-wide name
	get the port addresses
	get the medium rotation rate
	standard inquiry:get vendor/product information
	execute SCSI command %s(0x%02x) For example: execute SCSI command Inquiry (0x12) execute SCSI command Test Unit Ready (0x00) execute SCSI command Read (10)(0x28)

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

Product Number: 5737-F34

SC28-3493-00