

Implementation Guide for IBM Elastic Storage System 3000

Brian Herr

Chiahong Chen

Farida Yaragatti

John Lewars

Jonathan Turner

Luis Bolinches

Olaf Weiser

Puneet Chaudhary

Ravindra Sure

Robert Guthrie

Stefan Roth

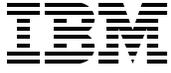
Todd M Tosseth

Vasfi Gucer

Wesley Jones



Storage



IBM Redbooks

**Implementation Guide for IBM Elastic Storage System
3000**

April 2020

Note: Before using this information and the product it supports, read the information in “Notices” on page v.

First Edition (April 2020)

This edition applies to IBM Elastic Storage Server 3000.

© Copyright International Business Machines Corporation 2020.

Note to U.S. Government Users Restricted Rights -- Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Notices	v
Trademarks	vi
Preface	vii
Authors	vii
Now you can become a published author, too	xi
Comments welcome	xi
Stay connected to IBM Redbooks	xii
Chapter 1. Introduction	1
1.1 IBM Spectrum Scale RAID	2
1.1.1 Product history	2
1.1.2 Distinguishing features	3
1.2 IBM Elastic Storage System (ESS)	4
1.3 IBM Elastic Storage System 3000	5
1.3.1 What is new in ESS 3000?	5
1.3.2 Value added	5
1.4 License considerations	6
Chapter 2. ESS 3000 architecture and overview	7
2.1 Platform	8
2.1.1 Canisters and servers	8
2.1.2 Peripheral Component Interconnect Express (PCIe)	10
2.1.3 NVMe (non-volatile memory express)	10
2.2 GUI enhancements	11
2.2.1 GUI users	11
2.2.2 System setup wizard	12
2.2.3 Using the GUI	15
2.2.4 Monitoring of ESS 3000 hardware	17
2.2.5 Storage	20
2.2.6 Replace broken disks	20
2.2.7 Health events	21
2.2.8 Event notification	22
2.2.9 Dashboards	25
2.2.10 More information	26
2.3 Software enhancements	26
2.3.1 Containerized deployment	26
2.3.2 Ansible	27
2.3.3 The mmvdisk command	27
2.3.4 The mmhealth command	27
2.4 RAS enhancements	31
2.4.1 Enclosure overview	32
2.4.2 Machine type model and warranty	33
2.4.3 Components: FRU versus CRU	33
2.4.4 RAS features	34
2.4.5 Maintenance and service procedures	34
2.4.6 Software related RAS enhancements	34
2.4.7 Integrated call home	35
2.4.8 Software call home	39

2.4.9 Performance	40
Chapter 3. Planning considerations	43
3.1 Planning	44
3.1.1 Technical and Delivery Assessment (TDA).....	44
3.1.2 Hardware remarks.....	44
3.2 Standalone environment.....	45
3.3 Mixed environment	46
3.3.1 Adding ESS 3000 to an existing ESS cluster	46
3.3.2 Scenario-1: Using ESS 3000 for metadata NSDs for the existing file system ...	50
3.3.3 Scenario-2: Using ESS 3000 to create a new file system.....	53
Chapter 4. Use cases	57
4.1 Metadata and High Speed Data Tiering	58
4.2 Database use cases	58
4.2.1 IBM Spectrum Scale for SAP HANA.....	59
4.2.2 Deploy ESS.....	59
4.2.3 Creating a file system	59
4.2.4 Preparing your clients (HANA nodes).....	60
4.2.5 Using the file systems.....	61
4.3 Artificial intelligence (AI) and machine learning (ML)	62
4.4 Other uses cases	62
4.4.1 IBM Spectrum Scale with big data and analytics solutions.....	62
4.4.2 Genomics Medicine workloads in IBM Spectrum Scale	63
4.4.3 IBM Cloud Object Store	63
Related publications	65
IBM Redbooks	65
Online resources	65
Help from IBM	66

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

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.

The performance data and client examples cited are presented for illustrative purposes only. Actual performance results may vary depending on specific configurations and operating conditions.

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.

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.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation, 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 <http://www.ibm.com/legal/copytrade.shtml>

The following terms are trademarks or registered trademarks of International Business Machines Corporation, and might also be trademarks or registered trademarks in other countries.

DB2®	IBM Spectrum®	pureScale®
IBM®	IBM Spectrum Storage™	Redbooks®
IBM Cloud™	POWER®	Redbooks (logo)  ®
IBM Elastic Storage®	POWER7®	Storwize®
IBM Research™	POWER8®	

The following terms are trademarks of other companies:

Intel, Intel logo, Intel Inside logo, and Intel Centrino logo are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

ITIL is a Registered Trade Mark of AXELOS Limited.

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.

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

Other company, product, or service names may be trademarks or service marks of others.

Preface

This IBM® Redbooks® publication introduces and describes the IBM Elastic Storage® Server 3000 (ESS 3000) as a scalable, high-performance data and file management solution. The solution is built on proven IBM Spectrum® Scale technology, formerly IBM General Parallel File System (IBM GPFS).

IBM Elastic Storage System 3000 is an all-Flash array platform. This storage platform uses NVMe-attached drives in ESS 3000 to provide significant performance improvements as compared to SAS-attached flash drives.

This book provides a technical overview of the ESS 3000 solution and helps you to plan the installation of the environment. We also explain the use cases where we believe it fits best.

Our goal is to position this book as the starting point document for customers that would use the ESS 3000 as part of their IBM Spectrum Scale setups.

This book is targeted toward technical professionals (consultants, technical support staff, IT Architects, and IT Specialists) who are responsible for delivering cost-effective storage solutions with ESS 3000.

Authors

This book was produced by a team of specialists from around the world working at IBM Redbooks, Poughkeepsie Center.



Brian Herr is a Software Engineer for IBM Spectrum Scale. He has been with IBM since 1983 working mostly in the area of High Performance Computing. He has been working on the ESS development team since 2008.



Chiahong Chen is an STSM in the ESS Platform team. He has 28 years of experience in IBM storage platform microcode development started from 3570/3590 tape drives and transitioned to DS6K/DS8K disk storage systems in 2002. Prior to joining ESS, he was part of the IBM storage CTO office for 3 years.



Farida Yaragatti is a Senior Software Engineer at IBM India. She has a BE, Electronics and Communication from Karnataka University, India and has 12 years of experience in Software testing field. She has been part of manual and automation testing for IBM Spectrum Scale and IBM ESS deployment as a Senior Tester. Farida has worked at IBM for over 5 years and previously held roles within the IBM Platform Computing and IBM Smart analytics system (ISAS) testing teams. She has strong engineering professional skills in Software deployment testing, including automation using various scripting technologies, such as Python, shell scripting, Robot framework, Ansible, and Linux.



John Lewars is a Senior Technical Staff Member leading performance engineering work in the IBM Spectrum Scale development team. He has been with IBM for over 20 years, working first on some of IBM's largest high performance computing systems, and later on the IBM Spectrum Scale (formerly GPFS) development team. John's work on the IBM Spectrum Scale team includes working with large customer deployments and improving network resiliency, along with co-leading development of the team's first public cloud and container support deliverables.



Jonathan Terner is a Software Engineer and a new member of the IBM Spectrum Scale RAID team working in the United States. He has recently finished his bachelors in Computer Science and Mathematics from Binghamton University, where he focused on distributed systems and virtualization technology.



Luis Bolinches has been working with IBM Power Systems servers for over 15 years, and has been with IBM Spectrum Scale (formerly known as IBM General Parallel File System) for over 10 years. He works 20% for IBM Systems Lab Services in the Nordic region, and the other 80% as part of the IBM Spectrum Scale development team.



Olaf Weiser joined IBM as a seasoned professional over 9 years ago and has worked in the DACH TSS team delivering Power-based solutions to enterprise and HPC customers. He has developed deep skills in IBM Spectrum Scale (previously IBM GPFS) and has a worldwide reputation as the performance optimization specialist for IBM GPFS outside development and research. At the IBM European Storage Competence Center (ESCC), Olaf is working on Advanced Technical Support (ATS) and Lab Services and Skill Enablement tasks that are required to grow IBM Spectrum Scale business in EMEA.



Puneet Chaudhary is a Technical Solutions Architect working with the IBM Elastic Storage Server and IBM Spectrum Scale solutions. He has worked with IBM GPFS, now IBM Spectrum Scale, for many years.



Ravindra Sure works for IBM India as a Senior System Software Engineer. He has worked on developing workload schedulers for High Performance Computers, Parallel File Systems, Computing Cluster Network Management, and Parallel Programming. He has strong engineering professional skills in distributed systems, parallel computing, C, C++, Python, shell scripting, MPI, and Linux.



Robert Guthrie is a Senior Software Engineer in Austin, Texas. He works with the IBM Spectrum Scale development team on storage enclosures and NVMe. He joined IBM in 1996 working on CORBA-based enterprise management software. He is a software and systems solutions specialist with extensive expertise in networks, firmware, and middleware. He has had lead roles providing superior levels of design, development, test, and system integration functions for multiple projects serving large, international financial, insurance, and government enterprise clients. He has been working on storage products since 2008, including Information Archive, IBM Spectrum Protect, and IBM Elastic Storage Server.



Stefan Roth is a Software Engineer in IBM Research™ and Development in Kelsterbach, Germany. He works with the IBM Spectrum Scale development team on the graphical user interface. He joined IBM in 1996 and in the first years he developed software for IBM disk drives and semiconductor factories. Since 2008, he has worked on graphical user interfaces for various IBM storage products, such as Scale Out Network Attached Storage, IBM Storwize® V7000 Unified, IBM Spectrum Scale, and IBM Elastic Storage Server. He holds a technical college degree in Electrical Engineering from University of Applied Sciences, Darmstadt.



Todd M Tosseth works as an IBM Spectrum Scale Develop and Test Engineer at IBM. His job responsibilities include testing the scalability and customer-like environments for IBM Spectrum Scale and GPFS Data Protection Software development.



Vasfi Gucer is an IBM Technical Content Services Project Leader with the Digital Services Group. He has more than 20 years of experience in the areas of systems management, networking hardware, and software. He writes extensively and teaches IBM classes worldwide about IBM products. His focus has been primarily on cloud computing, including cloud storage technologies for the last 6 years. Vasfi is also an IBM Certified Senior IT Specialist, Project Management Professional (PMP), IT Infrastructure Library (ITIL) V2 Manager, and ITIL V3 Expert.



Wesley Jones serves as the test team lead for IBM Spectrum Scale Native RAID. He also serves as one of the principle deployment architects for IBM Elastic Storage Server. His focus areas are IBM Power servers, IBM Spectrum Scale (GPFS), cluster software (xCAT), Red Hat Linux, Networking (especially InfiniBand and Gigabit Ethernet), storage solutions, automation, Python, and more.

Thanks to the following people for their contributions to this project:

Ann Lund
IBM Redbooks®, Poughkeepsie Center

Steve Duersch, Mamdouh Khamis, Rezaul Islam, John Dorfner, Mary Jane Zajac, Stephen M Tee, Angela Pholphiboun, Jay Vaddi, Frank Mangion, Russell Kliegel, Kumaran Rajaram
IBM USA

Markus Rohwedder, Mathias Dietz
IBM Germany

Pramod Thekkepat Achutha
IBM India

Now you can become a published author, too

Here's an opportunity to spotlight your skills, grow your career, and become a published author—all at the same time. Join an IBM Redbooks residency project and help write a book in your area of expertise, while honing your experience using leading-edge technologies. Your efforts will help to increase product acceptance and customer satisfaction, as you expand your network of technical contacts and relationships. Residencies run from two to six weeks in length, and you can participate either in person or as a remote resident working from your home base.

Find out more about the residency program, browse the residency index, and apply online at:
ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us.

We want our books to be as helpful as possible. Send us your comments about this book or other IBM Redbooks publications in one of the following ways:

- ▶ Use the online **Contact us** review Redbooks form found at:

ibm.com/redbooks

- ▶ Send your comments in an email to:

redbooks@us.ibm.com

- ▶ Mail your comments to:

IBM Corporation, IBM Redbooks
Dept. HYTD Mail Station P099
2455 South Road
Poughkeepsie, NY 12601-5400

Stay connected to IBM Redbooks

- ▶ Find us on Facebook:
<http://www.facebook.com/IBMRedbooks>
- ▶ Follow us on Twitter:
<http://twitter.com/ibmredbooks>
- ▶ Look for us on LinkedIn:
<http://www.linkedin.com/groups?home=&gid=2130806>
- ▶ Explore new Redbooks publications, residencies, and workshops with the IBM Redbooks weekly newsletter:
<https://www.redbooks.ibm.com/Redbooks.nsf/subscribe?OpenForm>
- ▶ Stay current on recent Redbooks publications with RSS Feeds:
<http://www.redbooks.ibm.com/rss.html>



Introduction

This chapter introduces the IBM Elastic Storage System 3000 (ESS 3000) solution. It has the following sections:

- ▶ 1.1, “IBM Spectrum Scale RAID ” on page 2
- ▶ 1.2, “IBM Elastic Storage System (ESS)” on page 4
- ▶ 1.3, “IBM Elastic Storage System 3000” on page 5
- ▶ 1.4, “License considerations” on page 6

1.1 IBM Spectrum Scale RAID

The IBM Spectrum Scale RAID software in ESS 3000 uses local NVMe drives. Because RAID functions are handled by the software, ESS 3000 does not require an external RAID controller or acceleration hardware.

IBM Spectrum Scale RAID in ESS 3000 supports two and three fault-tolerant RAID codes. The two-fault tolerant codes include 8 data plus 2 parity, 4 data plus 2 parity, and 3-way replication. The three-fault tolerant codes include 8 data plus 3 parity, 4 data plus 3 parity, and 4-way replication. Figure 1-1 shows example RAID tracks consisting of data and parity strips.

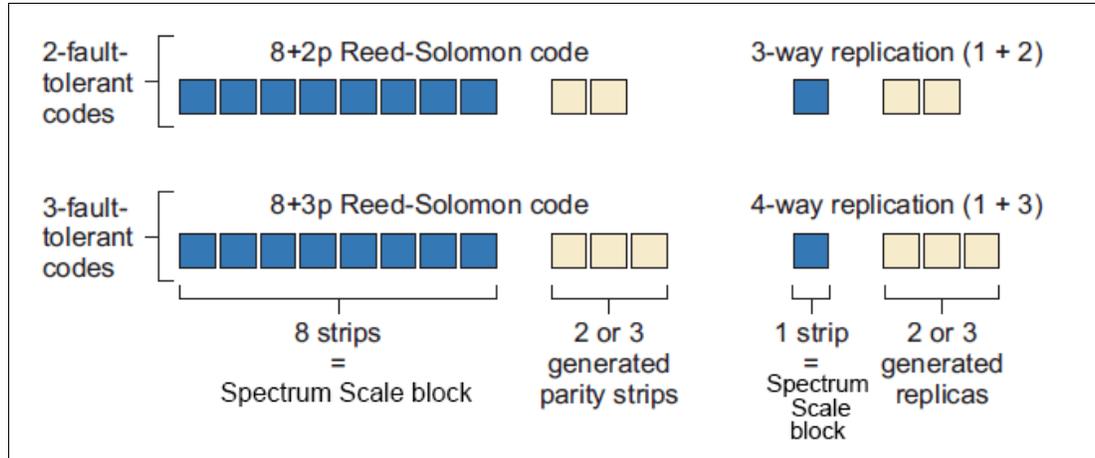


Figure 1-1 RAID tracks

1.1.1 Product history

In 2003, the Defense Advanced Research Project Agency (DARPA) started their High-Productivity Computing Systems (HPCS) program: Productive, Easy-to-use, Reliable Computing System (PERCS). IBM's proposal for DARPA's HPCS project was what today has become IBM Spectrum Scale RAID.

In 2007, IBM released the first market product based on IBM Spectrum Scale RAID, the P7IH. The system was based on the IBM POWER7® system and SAS disks, delivering tens of gigabytes per second of storage throughput already in 2007.

While P7IH was, and still is, a fantastic engineering machine, in 2012 IBM released the GSS platform that was running what is known today as IBM Spectrum Scale RAID but on commodity hardware.

In 2014, IBM superseded the GSS with the first ESS, based on the IBM POWER8® system but using commercially available servers and disk enclosures while still being based on the same IBM Spectrum Scale RAID that was designed in 2003.

IBM developed the technology on which ESS 3000 is based from its very beginning. We have a deep and unique understanding of this technology because we have been developing it starting 17 years ago up to today.

1.1.2 Distinguishing features

IBM Spectrum Scale RAID distributes data and parity information across node failure domains to tolerate unavailability or failure of all physical disks in a node. It also distributes spare capacity across nodes to maximize parallelism in rebuild operations.

IBM Spectrum Scale RAID implements end-to-end checksums and data versions to detect and correct the data integrity problems of traditional RAID. Data gets checked from the PDisk blocks on the ESS 3000 to the memory on the clients that connect over the network. It is the same checksum, not layers or serialized checksums that terminate in between the chain, so it really is an end-to-end checksum.

Figure 1-2 shows a simple example of declustered RAID. The left side shows a traditional RAID layout that consists of three 2-way mirrored RAID volumes and a dedicated spare disk that uses seven drives. The right side shows the equivalent declustered layout, which still uses seven drives. Here, the blocks of the three RAID volumes and the spare capacity are scattered over the seven disks.

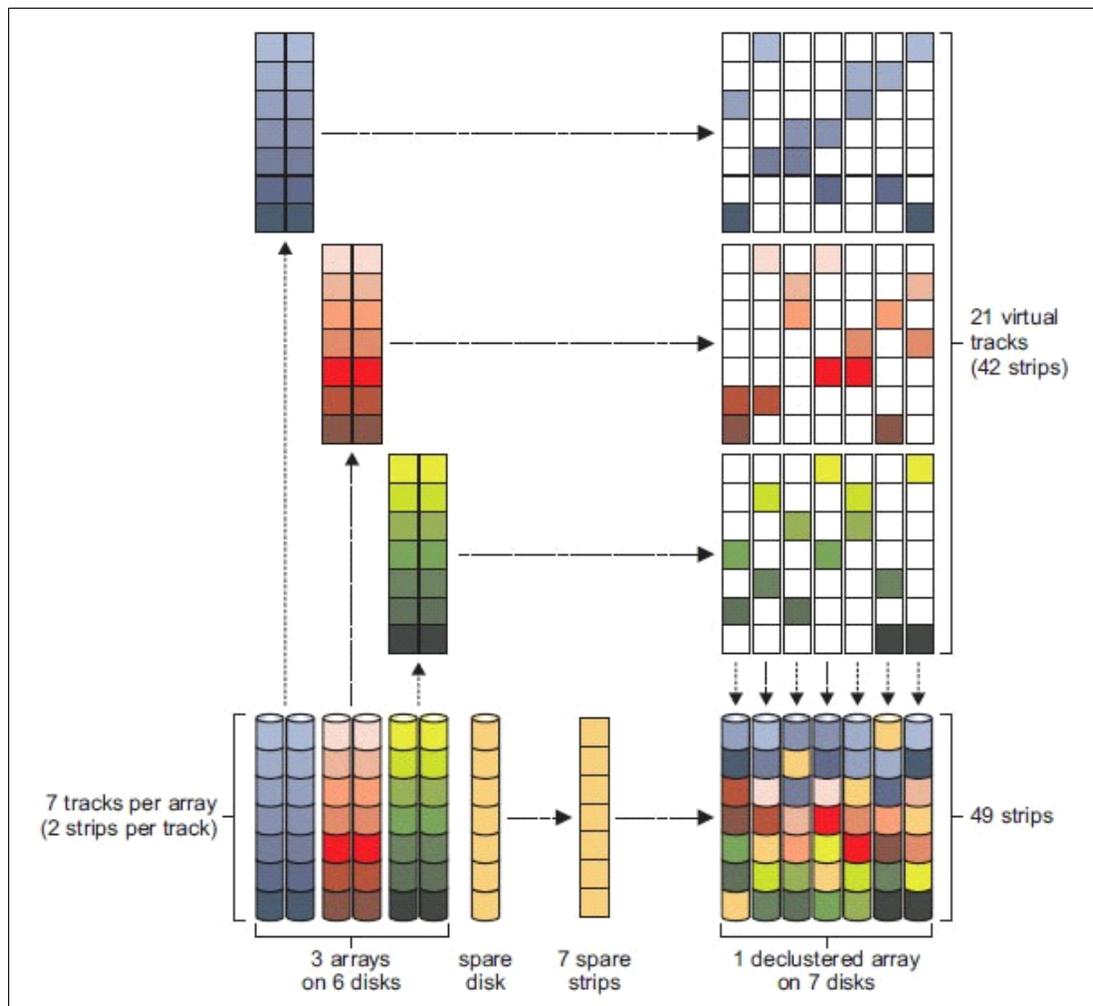


Figure 1-2 Declustered array versus 1+1 array

Figure 1-3 shows a significant advantage of declustered RAID layout over traditional RAID layout after a drive failure. With the traditional RAID layout on the left side of Figure 1-3, the system must copy the surviving replica of the failed drive to the spare drive, reading only from one drive and writing only to one drive.

However, with the declustered layout that is shown on the right of Figure 1-3, the affected replicas and the spares are distributed across all six surviving disks. This configuration rebuilds reads from all surviving disks and writes to all surviving disks, which greatly increases rebuild parallelism.

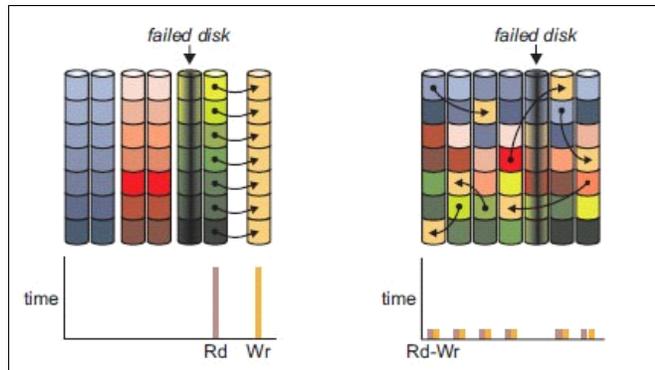


Figure 1-3 Array rebuild operation

Another advantage of the declustered RAID technology that is used by ESS 3000 (and other IBM systems) is that it minimizes the worst-case number of critical RAID tracks in the presence of multiple disk failures. ESS 3000 can then deal with restoring protection to critical RAID tracks as a high priority, while giving lower priority to RAID tracks that are not considered critical.

For example, consider an 8+3p RAID code on an array of 100 PDIs. In the traditional layout and declustered layout, the probability that a specific RAID track is critical is $11/100 \times 10/99 \times 9/98$ (0.1%). However, when a track is critical in the traditional RAID array, all tracks in the volume are critical, whereas with declustered RAID, only 0.1% of the tracks are critical. By prioritizing the rebuild of more critical tracks over less critical tracks, ESS 3000 quickly gets out of critical rebuild and then can tolerate another failure.

ESS 3000 adapts these priorities dynamically; if a *non-critical* RAID track is used and more drives fail, this RAID track's rebuild priority can be escalated to *critical*.

A third advantage of declustered RAID is that it makes it possible to support any number of drives in the array and to dynamically add and remove drives from the array. Adding a drive in a traditional RAID layout (except in the case of adding a spare) requires significant data reorganization and restriping. However, only targeted data movement is needed to rebalance the array to include the added drive in a declustered array.

1.2 IBM Elastic Storage System (ESS)

ESS is based on IBM Spectrum Scale Native RAID to provide the physical disk protection and tightly integrated with IBM Spectrum Scale to provide the file system access over the network to all of the IBM Spectrum Scale clients. There are other protocols that can be used to access the IBM Spectrum Scale file system.

Because it falls outside of the scope of this publication, for details about the ways to access an IBM Spectrum Scale file system, use the following IBM Knowledge Center link:

https://www.ibm.com/support/knowledgecenter/en/STXKQY_5.0.4/com.ibm.spectrum.scale.v5r04.doc/b11in_PlanningForIBMSpectrumScale.htm.

You can also see the following IBM Redpaper: *Introduction Guide to the IBM Elastic Storage Server*, REDP-5253.

1.3 IBM Elastic Storage System 3000

IBM Elastic Storage System 3000 is an all-Flash array platform. This storage platform uses NVMe-attached drives in ESS 3000 to provide significant performance improvements as compared to SAS-attached flash drives.

IBM Elastic Storage System 3000 can contain up to 24 NVMe-attached SSD drives, 12 drives (half populated) or 24 drives (fully populated).

For details on the ESS 3000, please visit the following IBM Knowledge Center link:

https://www.ibm.com/support/knowledgecenter/en/SSZL24_6.0.0/ess3000_600_welcome.html.

1.3.1 What is new in ESS 3000?

ESS 3000 being based on IBM Spectrum Scale and IBM Spectrum Scale RAID is not unique because there are other IBM products that are also based on those features (all the other ESS models and IBM Spectrum Scale Erasure Code Edition). However, there are some features that for today's ESS systems are unique on the ESS 3000, including the following prominent features:

- ▶ NVMe transport protocol for high performance of 2.5-inch (SFF) NVMe-attached flash drives.
- ▶ NVMe is designed specifically for flash technologies. It is a faster, less complicated storage drive transport protocol than SAS.
- ▶ NVMe offers better performance and lower latencies exclusively for solid-state drives through multiple I/O queues and other enhancements.
- ▶ Containerized Ansible playbooks that provide orchestration of complex tasks, such as cluster configuration, file system creation, and code update.
- ▶ Higher density and better performance per rack space than any other ESS available.

1.3.2 Value added

IBM Elastic Storage System 3000 is designed to meet and beat the challenge of managing data for analytics. Packaged in a compact 2U enclosure, ESS 3000 is a proven data management solution that speeds time to value for artificial intelligence / deep learning and high-performance computing workloads thanks to its blisteringly quick all-NVMe storage and simple, fast containerized software installation and upgrade.

Its no-compromise hardware and software design gives you the industry-leading performance required to keep data-hungry processors fully utilized. ESS 3000 is compatible with all IBM Elastic Storage Server models.

Fast time-to-value

ESS 3000 combines IBM Spectrum Scale file management software with NVMe flash storage for the ultimate in scale-out performance and simplicity, delivering 40 GBps of data throughput per 2U system.

Operational efficiency

Containerized software install and a powerful management GUI minimize demands on IT staff time and expertise. Dense storage within a 2U package means a small data center footprint.

Reliability

Software-defined erasure coding assures data recovery while using less space than data replication. Restores can take minutes rather than hours or days, and can be run without disrupting operations.

Deployment flexibility

Available in a wide range of capacities from tens to hundreds of terabytes per 2U. Deploy as a standalone system or scale out with additional ESS 3000 systems, or with IBM Elastic Storage Server.

1.4 License considerations

ESS 3000 follows the same license model as the other ESS products. The two currently available options are *Data Access* and *Data Management*.

ESS uses capacity based licensing. This means that a customer can connect as many clients as desired without extra license costs. For other types of configurations, please contact IBM or your IBM Business Partner for license details.

For more details about licensing on IBM Spectrum Scale, please visit the following IBM Knowledge Center link:

https://www.ibm.com/support/knowledgecenter/en/STXKQY_5.0.4/com.ibm.spectrum.scale.v5r04.doc/bl1ins_capacitylicense.htm



ESS 3000 architecture and overview

This chapter describes the architecture and provides an overview of IBM Elastic Storage System 3000 (ESS 3000). It covers the following topics:

- ▶ 2.1, “Platform” on page 8
- ▶ 2.2, “GUI enhancements” on page 11
- ▶ 2.3, “Software enhancements” on page 26
- ▶ 2.4, “RAS enhancements” on page 31

2.1 Platform

IBM Elastic Storage System 3000 (ESS 3000) is an all-Flash array platform. This storage platform uses NVMe-attached drives in ESS 3000 to provide significant performance improvements as compared to SAS-attached flash drives. This chapter provides an overview of the ESS 3000 platform.

2.1.1 Canisters and servers

We cover CPU, memory, and networking in this section.

CPU

ESS 3000 provides dual 14-core Intel Skylake 64-bit CPUs at 2.2 GHz per each of the two server canisters for a total of 4 CPUs per enclosure. Figure 2-1 shows two CPUs in a canister.

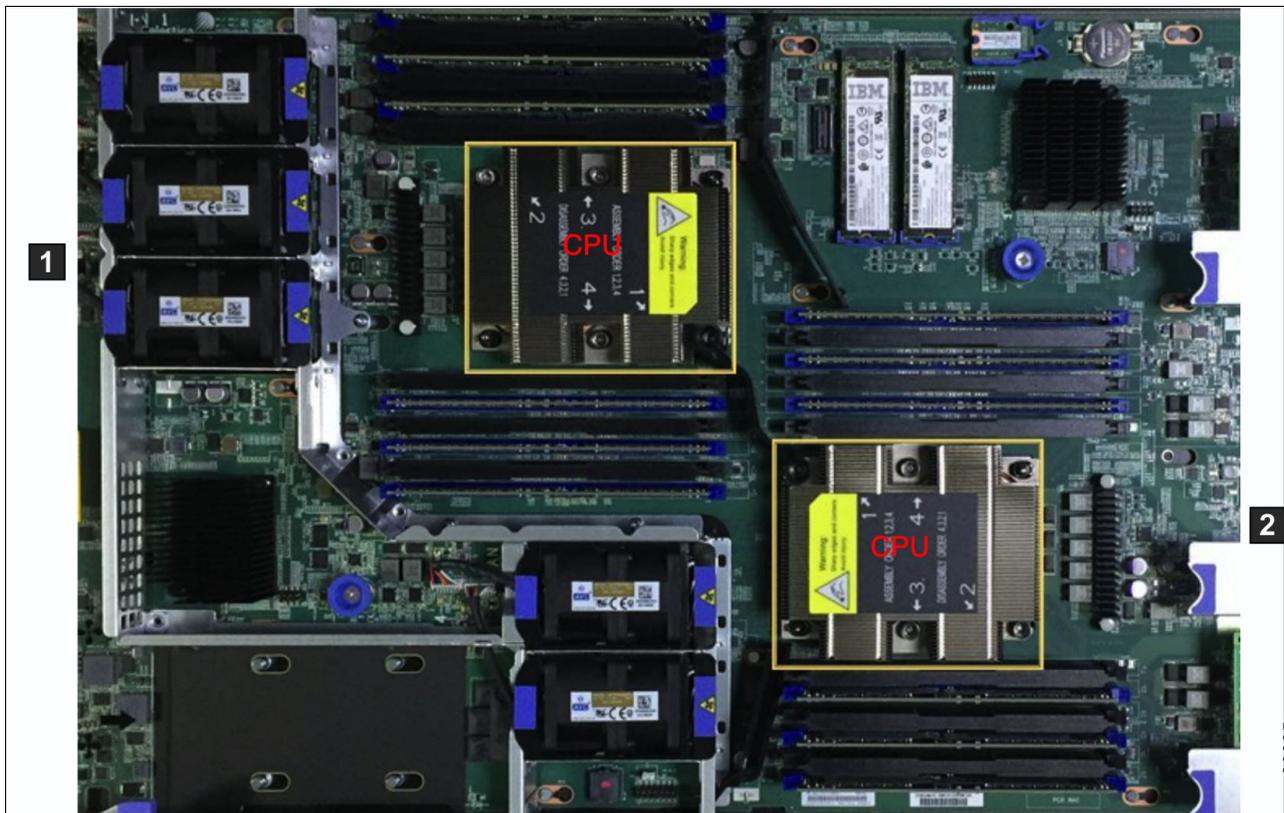


Figure 2-1 Two CPUs in a canister

Memory

Elastic Storage System 3000 uses the same memory features that control the initial amount of memory and subsequent memory upgrades. Each CPU has twelve DIMM slots, for a total of 24 DIMM slots per server canister, which means 48 DIMM slots per enclosure. The customer can install two distinct memory configurations in those 48 DIMM slots. Initially, every ESS 3000 ships with the ACG1 feature: 768 GB base cache memory (twenty-four 32 GB DIMMs, so 6 per CPU which translates to 384 GB per canister and 768 GB per enclosure).

The customer can order the following ACGB feature to upgrade to more memory at any time. ACGB can be ordered as either an MFI (factory install) or by post-sale MES upgrade (field install). This is a 768 GB memory upgrade (twenty four 32 GB DIMMs). Figure 2-2 shows the DIMMs in a canister.

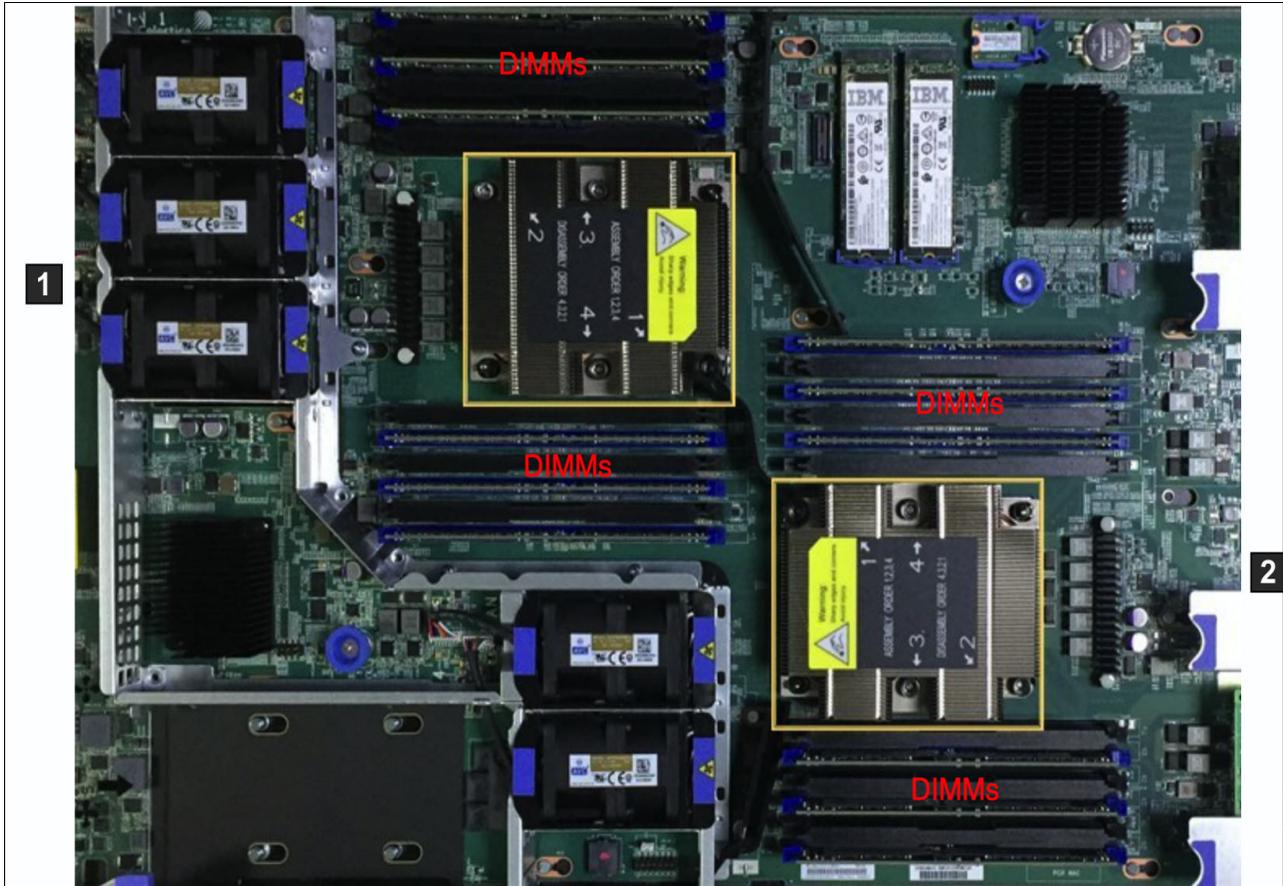


Figure 2-2 DIMMs in a canister

Networking

During production, the customer uses the high-speed network links (100 Gb Ethernet or 100 Gb InfiniBand) to serve the IBM Spectrum Scale cluster file system. Each ESS 3000 server canister also has three PCIe interface slots to support optional host interface adapters. Figure 2-3 shows 3 HBA ports per canister. In this example there are two HBAs per canister and one empty HBA slot.



Figure 2-3 Two HBAs per canister and one empty HBA slot

2.1.2 Peripheral Component Interconnect Express (PCIe)

The ESS 3000 has two node server canisters with a PCIe Gen3 fabric, communicating with an enclosure midplane for connectivity to 24 dual-port NVMe drive slots. Internally to the canisters are two processor modules (Intel Skylake), each with integrated PCIe Gen3 root complex with 48 lanes. Connected off of the CPUs, are 3 × 16 slots for HBA IO Adapters, one off of CPU socket 0 and the other two off of CPU socket 1. The interconnect between the CPUs and the NVMe drive slots is a Microsemi 96 lane PCIe Gen3 8546 PSX switch.

As shown in Figure 2-4, the 96 lanes are divided as follows: ×16 buses to each of the two CPUs, 48 lanes for 24 × 2 connections to the 24 NVMe drive slots, ×16 NTB between canisters for peer-to-peer canister communication over the midplane.

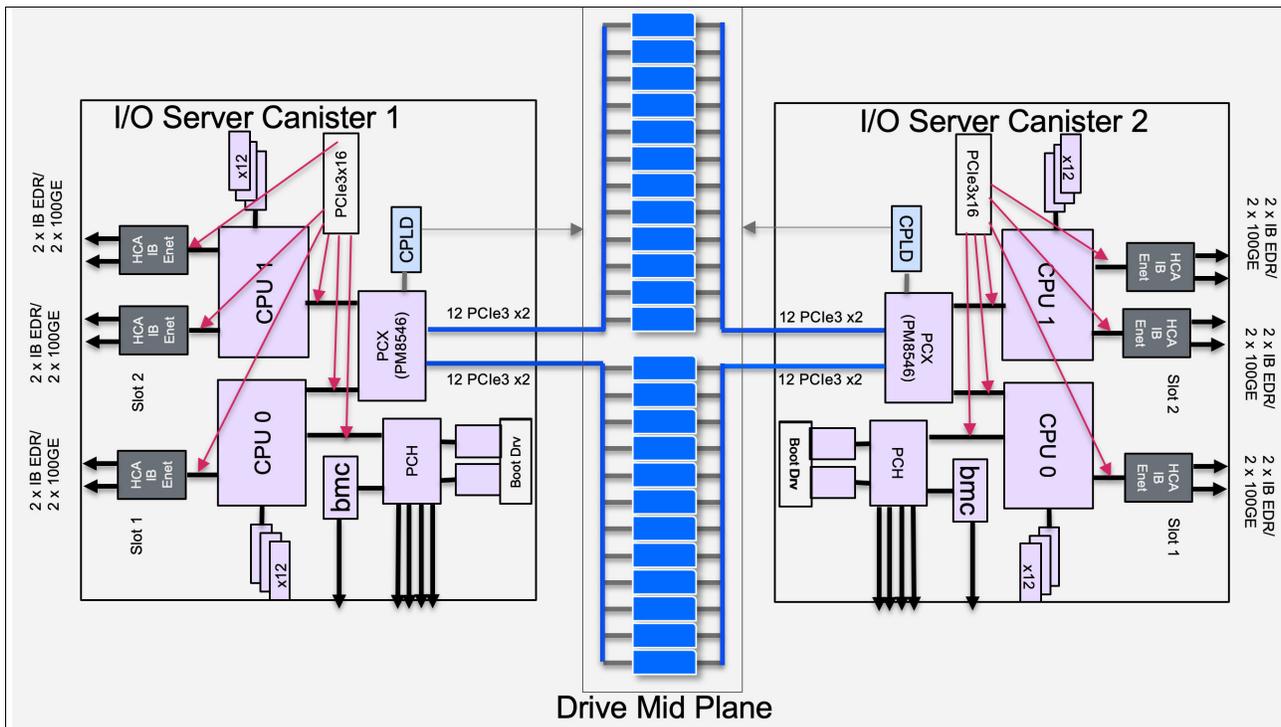


Figure 2-4 PCIe in ESS 3000

2.1.3 NVMe (non-volatile memory express)

Measurements in the IBM lab on a freshly installed and fully populated Elastic Storage system 3000 with a 4 MiB file system block size have achieved sequential read performance of over 43 GBps and sequential write performance of over 34 GBps when using an InfiniBand network with Remote Direct Memory Access (RDMA) enabled.

Note: The performance measurements referenced here were made using standard benchmarks in a controlled environment. The actual performance of any given Elastic Storage System 3000 will vary depending on a number of factors, such as the network, workload characteristics, and the configuration of the client nodes. Therefore, no assurance can be given that any given Elastic Storage System 3000 can achieve results similar to those stated here.

A freshly configured, fully populated Elastic Storage System 3000 can achieve a sequential read performance of up to 42 GBps and a sequential write performance of up to 32 GBps.

End-to-end checksum

The end-to-end checksum feature of IBM Spectrum Scale and IBM Spectrum Scale RAID software enables the system to prevent and correct silent disk errors or missing disk writes.

The IBM Spectrum Scale software on the client used to access data on the Elastic Storage Server 3000 knows that the IBM Spectrum Scale file system is based on IBM Spectrum Scale RAID Network Shared Disks, and during a write operation an 8-bytes checksum is calculated, appended to the data, and sent over the network to the IBM Spectrum Scale RAID server. The checksum is verified, and then IBM Spectrum Scale RAID writes the data along with its checksum on the disks and logs the version number on its metadata.

When a read operation is requested, IBM Spectrum Scale RAID verifies the checksum and version on its metadata. If it is OK, it sends the data to the client. If it is not OK, the data is rebuilt based on parity or replication and then sent to the client along with newly generated checksum.

2.2 GUI enhancements

A graphical user interface (GUI) service runs on the EMS server. It can be used to monitor the health of the ESS and to perform management tasks. This chapter provides a rough overview of the GUI and is by no means comprehensive.

Run the `systemctl` command on the EMS server to start or stop the GUI. Table 2-1 shows the `systemctl` command options.

Table 2-1 The `systemctl` command options

Command	Description
Start the GUI service	<code>systemctl start gpfsgui</code>
Check the status of the GUI service	<code>systemctl status gpfsgui</code>
Stop the GUI service	<code>systemctl stop gpfsgui</code>

To access the GUI, enter the IP address or host name of the EMS server in a web browser using the secure https mode (`https://<IP or hostname of EMS>`).

2.2.1 GUI users

GUI users must be created before the GUI can be used. To grant special rights, roles are assigned to the users.

When the GUI is used for the first time, an initial user must be created:

```
/usr/lpp/mmfs/gui/cli/mkuser <username> -g SecurityAdmin
```

After this, log into the GUI with the new user and create more users in the **Services** → **GUI** → **Users** page. By default, users are stored in an internal user repository. Alternatively, an external user repository can also be used. This can be configured in the **Services** → **GUI** → **External Authentication** page.

2.2.2 System setup wizard

Perform the following steps:

1. After logging into the GUI for the first time, the system setup wizard is launched. This looks up the systems information and performs several checks. See Figure 2-5.

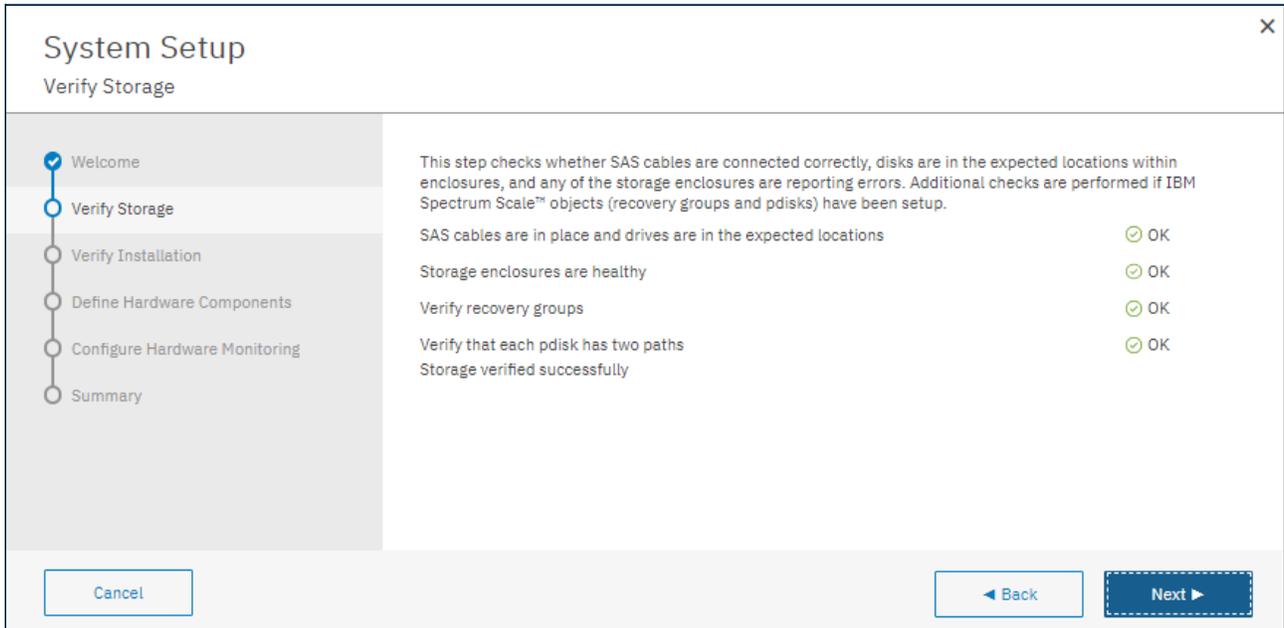


Figure 2-5 The System Setup wizard

- In the **Racks** step, the racks where the ESS 3000 systems are installed need to be defined. Either choose a predefined rack type or choose **Add new specification** in case none of the available rack types matches your rack. It is important that the selected rack type has the same number of height units. A meaningful name can be specified for the racks to create. See Figure 2-6.

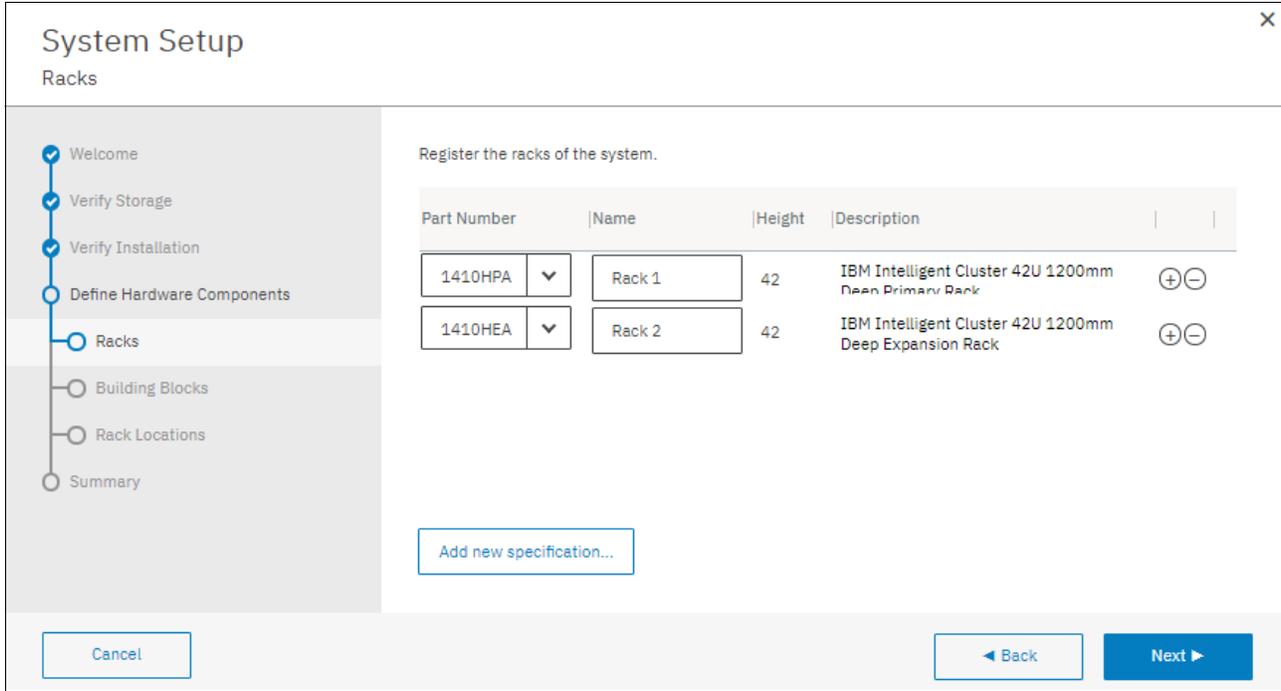


Figure 2-6 Specifying the racks

- The **Building Blocks** step displays one row for each ESS 3000 or other ESS models. Assign names to each building block or go with the default. See Figure 2-7.

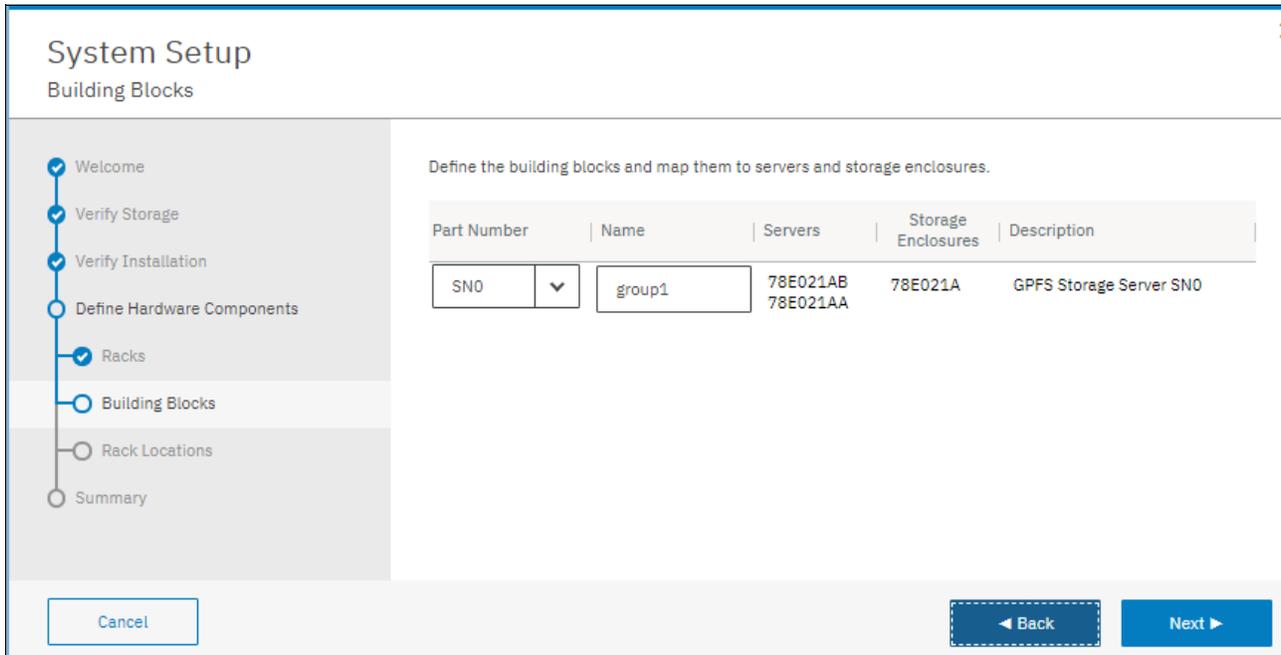


Figure 2-7 Define building blocks

- In the next step, the ESS 3000 systems are assigned to the rack locations in which they are mounted. See Figure 2-8.

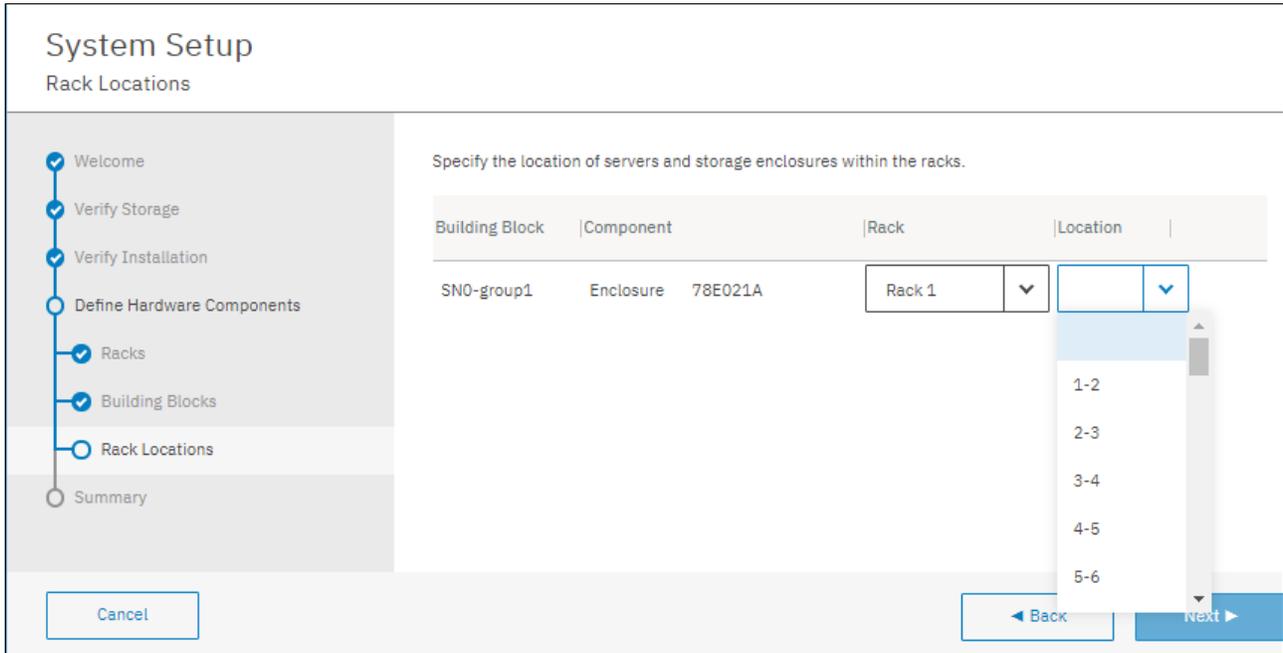


Figure 2-8 Assign rack locations

- The xCAT software is used to monitor the hardware of IBM POWER® based servers like the EMS server or protocol servers. Therefore, the **Configure Hardware Monitoring** page only appears if ESS 3000 systems are mixed with other ESS models in one cluster. In a pure ESS 3000 cluster, this step is not displayed because the ESS 3000 canister hardware is not monitored via xCAT. See Figure 2-9.

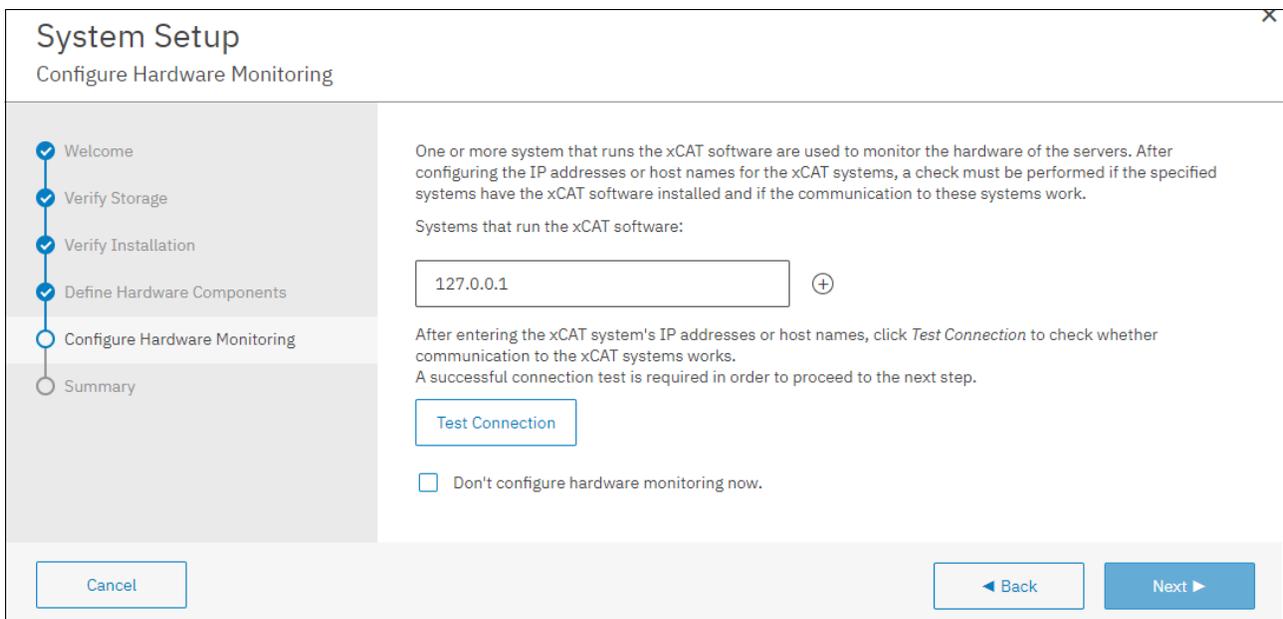


Figure 2-9 Configure xCAT

- The **Configure Hardware Monitoring** page allows to specify the IP address of the system where the xCAT software runs. The IP usually is 127.0.0.1 because xCAT runs on the EMS like the GUI. The connection to xCAT can also be configured at any later time by using the **Configure Hardware Monitoring** action in the **Monitoring** → **Hardware** page.

2.2.3 Using the GUI

After logging into the GUI, the **Overview** page is displayed. This page provides a good view on all objects in the system and their health state. Clicking the numbers or links displays a more detailed view. See Figure 2-10.

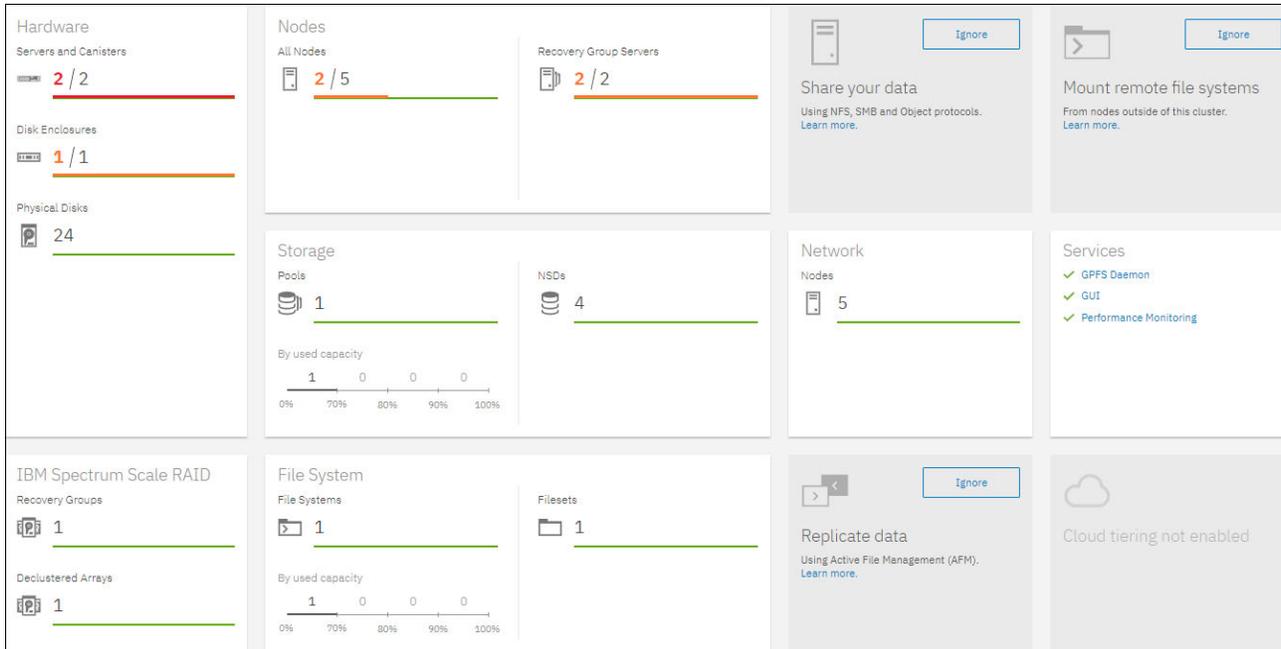


Figure 2-10 The Overview page

The header area of the GUI provides a quick view of the current health problems and tips for improvement, if available. Additionally, there are links to some help resources.

Use the navigation menu on the left side of the GUI page to navigate to other GUI pages (see Figure 2-11). Each GUI page has a unique URL that you can use to directly access the page, bookmark pages, and start the GUI in-context.

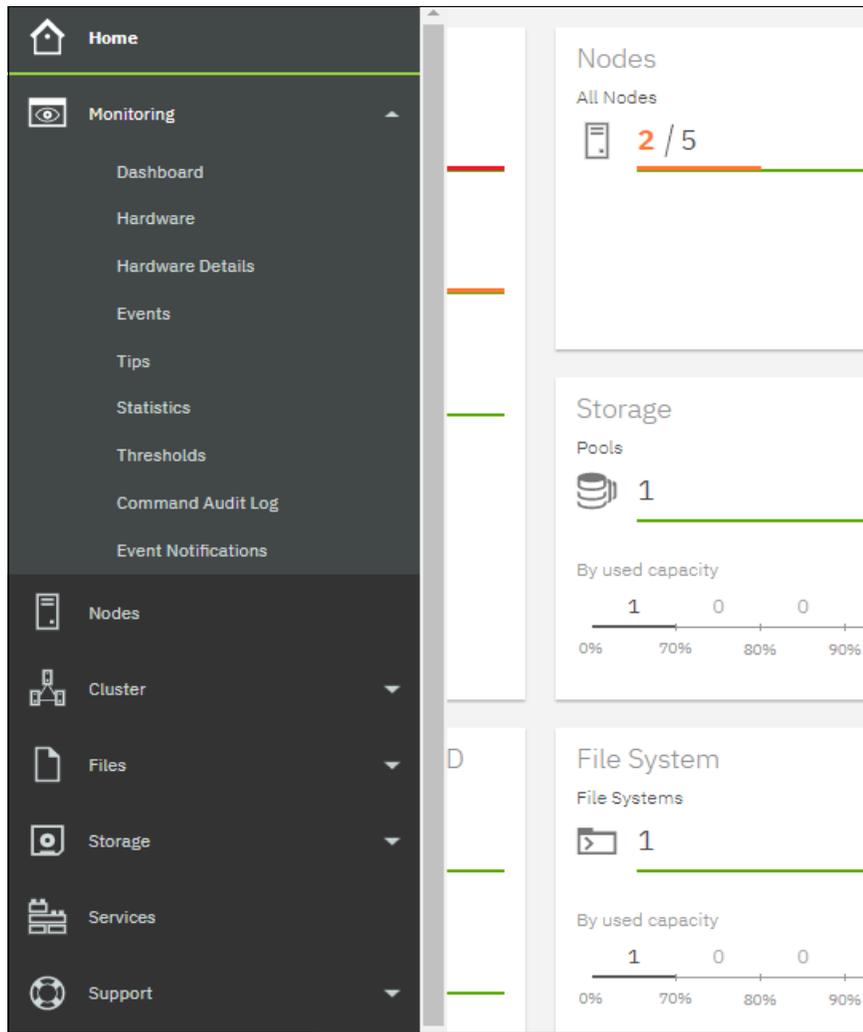


Figure 2-11 Navigation pane of the GUI

Some menus, such as **Protocols**, are only displayed when the related features, such as NFS, SMB, or AFM are enabled.

Most tables shown in the GUI have columns that are hidden by default. Right-click the table header and select the columns to display. See Figure 2-12.

The screenshot shows a table with columns: Name, State, CPU Usage, Product Version, Designated License, and Required License. A context menu is open on the right, listing various columns with checkboxes. The 'Name' column is checked, while others like 'Node Number', 'Protocol', 'Load', 'Memory Used', 'Bytes Read', 'Bytes Written', 'Read OPS', and 'Write OPS' are unchecked.

Name	State	CPU Usage	Product Version	Designated License	Required License
fsc- fab3-1-a.mainz.de.ibm.com	Degraded	0.25%	5.0.4.1	Server	Server/FPO
fsc- fab3-1-b.mainz.de.ibm.com	Degraded	0.42%	5.0.4.1	Server	Server/FPO
fsc- x36m3-30.mainz.de.ibm.com	Healthy	0.85%	5.0.4.1	Server	Server
fsc- x36m3-41.mainz.de.ibm.com	Healthy	0.14%	5.0.4.1	Server	Server
fsc- x36m3-31.mainz.de.ibm.com	Healthy	2.94%	5.0.4.1	Server	Server

Figure 2-12 Show and hide table columns

The table values can be sorted by clicking one of the column headers. A little arrow in the table header indicates the sorting.

Double-click a table row to open a more detailed view of the selected item.

2.2.4 Monitoring of ESS 3000 hardware

The **Monitoring** → **Hardware** page displays the ESS 3000 enclosures within the racks. A table lists all enclosures and the related canisters. See Figure 2-13.

Name	Serial Number	State	Rack	Location	Building Block	Type
5141-AF8-78E021A	78E021A	Healthy	Rack 1	10	group1	SN0 Enclosure
fsc- fab3-1-b.mainz.de.ibm.com	78E021AA	Healthy	Rack 1	10	group1	Canister/Server
fsc- fab3-1-a.mainz.de.ibm.com	78E021AB	Healthy	Rack 1	11	group1	Canister/Server
5141-AF8-78E021Y	78E021Y	Healthy	Rack 1	12	group2	SN0 Enclosure
fsc- fab3-2-b.mainz.de.ibm.com	78E021YA	Healthy	Rack 1	12	group2	Canister/Server
fsc- fab3-2-a.mainz.de.ibm.com	78E021YB	Healthy	Rack 1	13	group2	Canister/Server

Figure 2-13 Hardware page with two ESS 3000 systems

Use **Edit Rack Components** when ESS enclosures or servers have been added or removed, or if their rack location has changed.

The **Replace Broken Disks** action launches a guided procedure to replace broken disks if there are any.

Click **Configure Hardware Monitoring** to specify the IP address of an xCAT server if this is used to monitor server hardware. The xCAT software is not used to monitor the canisters of ESS 3000. Therefore, this action is only useful when mixing ESS 3000 with other ESS models in the same cluster, or to monitor the EMS server or POWER based protocol servers. Any other servers are not monitored through the GUI.

Click the ESS 3000 in the rack to see more information about the ESS 3000, including the disks and the two canisters (Figure 2-14). Move the mouse over the components, such as drives and power supplies, to see more information. Clicking components moves to a page with more detailed information. Broken disks are indicated with the color red, and a context menu (right-click) enables you to replace the selected broken disk.

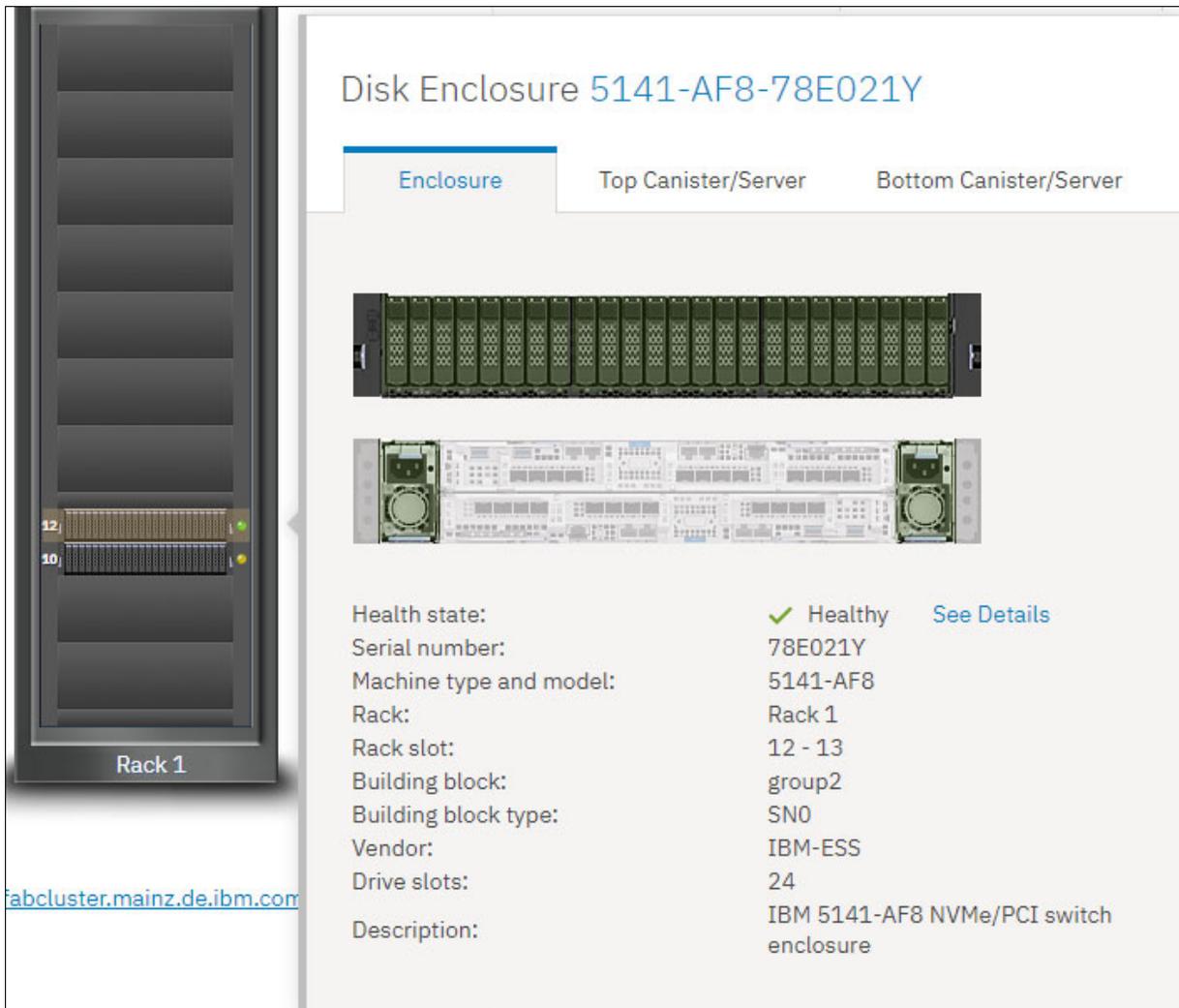


Figure 2-14 ESS 3000 details in the Monitoring ?Hardware page

If there is more than one rack, click the arrows displayed on the left and the right side of the rack to switch to another rack.

The **Monitoring** → **Hardware Details** page displays more detailed information and the health states of the ESS 3000 and its internal components. See Figure 2-15.

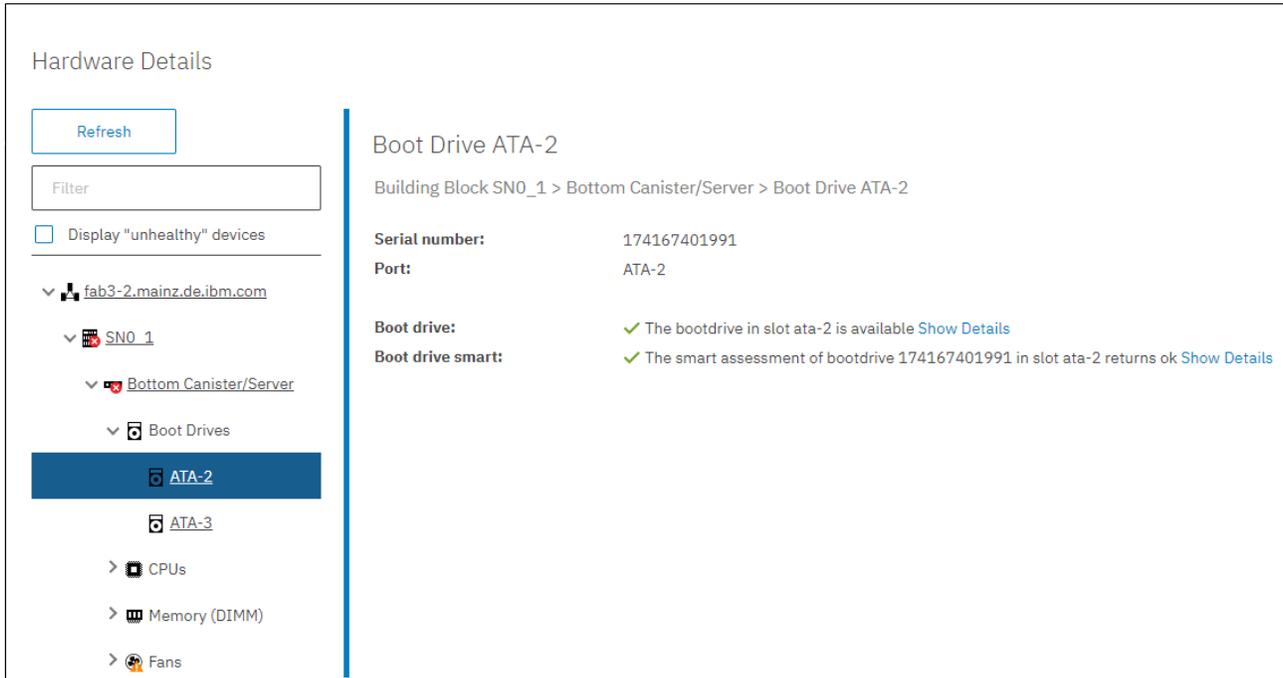


Figure 2-15 The Hardware Details page

This page allows the user to search for components by text, and filter the results to only display unhealthy hardware.

Click the > icon on the tree nodes to display subsequent children, for example to display all CPUs of the canister in Figure 2-15.

2.2.5 Storage

The **Storage** menu provides various views into the storage, such as the physical disks, declustered arrays, recovery groups, virtual disks, NSDs, and storage pools (Figure 2-16).

View Details Actions Export									
Clustered Array	Status	Health State	Capacity	Hardware Type	FRU	Location	Firmware	SSD Endurance	
1	✓ Normal	✓ Healthy	3.49 TiB	NVMe	KCM5DRUG3T84	Rack r1 U20-21, Enclosure 5141-AF8-78E021A Drive 3	C5SC	0 %	
1	✓ Normal	✓ Healthy	3.49 TiB	NVMe	KCM5DRUG3T84	Rack r1 U20-21, Enclosure 5141-AF8-78E021A Drive 7	C5SC	0 %	
1	✓ Normal	✓ Healthy	3.49 TiB	NVMe	KCM5DRUG3T84	Rack r1 U20-21, Enclosure 5141-AF8-78E021A Drive 12	C5SC	0 %	
1	✓ Normal	✓ Healthy	3.49 TiB	NVMe	KCM5DRUG3T84	Rack r1 U20-21, Enclosure 5141-AF8-78E021A Drive 22	C5SC	0 %	
1	✓ Normal	✓ Healthy	3.49 TiB	NVMe	KCM5DRUG3T84	Rack r1 U20-21, Enclosure 5141-AF8-78E021A Drive 21	C5SC	0 %	
1	✓ Normal	✓ Healthy	3.49 TiB	NVMe	KCM5DRUG3T84	Rack r1 U20-21, Enclosure 5141-AF8-78E021A Drive 20	C5SC	0 %	
1	✓ Normal	✓ Healthy	3.49 TiB	NVMe	KCM5DRUG3T84	Rack r1 U20-21, Enclosure 5141-AF8-78E021A Drive 8	C5SC	0 %	
1	✓ Normal	✓ Healthy	3.49 TiB	NVMe	KCM5DRUG3T84	Rack r1 U20-21, Enclosure 5141-AF8-78E021A Drive 10	C5SC	0 %	
1	✓ Normal	✓ Healthy	3.49 TiB	NVMe	KCM5DRUG3T84	Rack r1 U20-21, Enclosure 5141-AF8-78E021A Drive 19	C5SC	0 %	
1	✓ Normal	✓ Healthy	3.49 TiB	NVMe	KCM5DRUG3T84	Rack r1 U20-21, Enclosure 5141-AF8-78E021A Drive 18	C5SC	0 %	
1	✓ Normal	✓ Healthy	3.49 TiB	NVMe	KCM5DRUG3T84	Rack r1 U20-21, Enclosure 5141-AF8-78E021A Drive 6	C5SC	0 %	
1	✓ Normal	✓ Healthy	3.49 TiB	NVMe	KCM5DRUG3T84	Rack r1 U20-21, Enclosure 5141-AF8-78E021A Drive 5	C5SC	0 %	

Figure 2-16 Storage menu

2.2.6 Replace broken disks

The GUI provides a guided procedure that can be used to replace broken disks. Make sure that the replacement disks have the same FRUs as the disks that you want to replace.

The procedure can be launched from different places. A good place to look for broken disks is the **Storage** → **Physical Disks** page shown in Figure 2-17.

Physical Disks									
Replace Broken Disks View Details Actions Export									
Name	Recovery Group	Declassified Array	Status	Health State	Capacity	Hardware Type	FRU	Location	Recovery Group Server Node
e1s22	FAB3_1RG	DA1	✗ Replaceable	✗ Failed	3.49 TiB	NVMe	KCM5DRUG3T84	Rack r1 U20-21, Enclosure 5141-AF8-78E021A Drive 22	
e1s09	FAB3_1RG	DA1	✗ Replaceable	✗ Failed	3.49 TiB	NVMe	KCM5DRUG3T84	Rack r1 U20-21, Enclosure 5141-AF8-78E021A Drive 9	
e1s03	FAB3_1RG	DA1	✓ Normal	✓ Healthy	3.49 TiB	NVMe	KCM5DRUG3T84	Rack r1 U20-21, Enclosure 5141-AF8-78E021A Drive 3	fscc-fab3-1-b.mainz.de.ibm.com
e1s12	FAB3_1RG	DA1	✓ Normal	✓ Healthy	3.49 TiB	NVMe	KCM5DRUG3T84	Rack r1 U20-21, Enclosure 5141-AF8-78E021A Drive 12	fscc-fab3-1-b.mainz.de.ibm.com
e1s21	FAB3_1RG	DA1	✓ Normal	✓ Healthy	3.49 TiB	NVMe	KCM5DRUG3T84	Rack r1 U20-21, Enclosure 5141-AF8-78E021A Drive 21	fscc-fab3-1-b.mainz.de.ibm.com
e1s20	FAB3_1RG	DA1	✓ Normal	✓ Healthy	3.49 TiB	NVMe	KCM5DRUG3T84	Rack r1 U20-21, Enclosure 5141-AF8-78E021A Drive 20	fscc-fab3-1-b.mainz.de.ibm.com
e1s08	FAB3_1RG	DA1	✓ Normal	✓ Healthy	3.49 TiB	NVMe	KCM5DRUG3T84	Rack r1 U20-21, Enclosure 5141-AF8-78E021A Drive 8	fscc-fab3-1-b.mainz.de.ibm.com
e1s07	FAB3_1RG	DA1	✓ Normal	✓ Healthy	3.49 TiB	NVMe	KCM5DRUG3T84	Rack r1 U20-21, Enclosure 5141-AF8-78E021A Drive 7	fscc-fab3-1-b.mainz.de.ibm.com
e1s10	FAB3_1RG	DA1	✓ Normal	✓ Healthy	3.49 TiB	NVMe	KCM5DRUG3T84	Rack r1 U20-21, Enclosure 5141-AF8-78E021A Drive 10	fscc-fab3-1-b.mainz.de.ibm.com

Figure 2-17 Physical Disks page

Choose the **Replace Broken Disks** action to get a list of all broken disks and choose some to replace. Optionally, select an individual disk from the table and choose the **Replace Disk** action to replace the selected disk. In both cases, a fix procedure guides you while you replace the disks. See Figure 2-18.

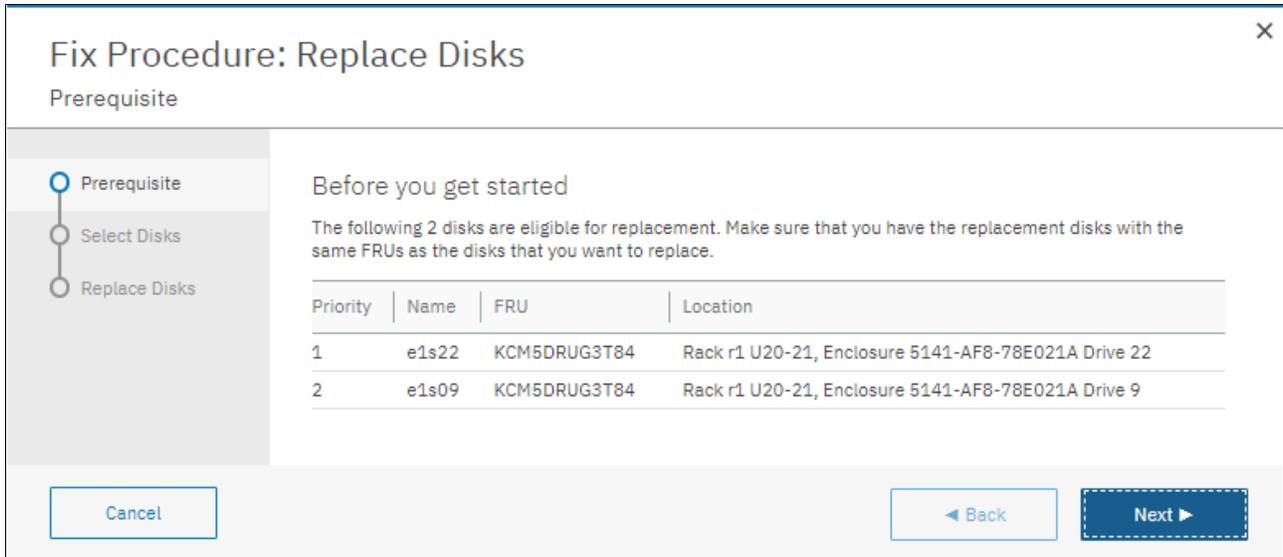


Figure 2-18 Fix procedure for replacing disks

2.2.7 Health events

Use the **Monitoring** → **Events** page to review the entire set of events that are reported in the ESS system. Under the **Event Groups** tab, all individual events with the same name are grouped into single rows, which is especially useful when dealing with a large volume of events. The **Individual Events** tab lists all the events, irrespective of the multiple occurrences. Events are assigned to a component, such as canister, enclosure, file system, SMB. Click the a component in the bar chart above the grid to filter for events of the selected component. See Figure 2-19.

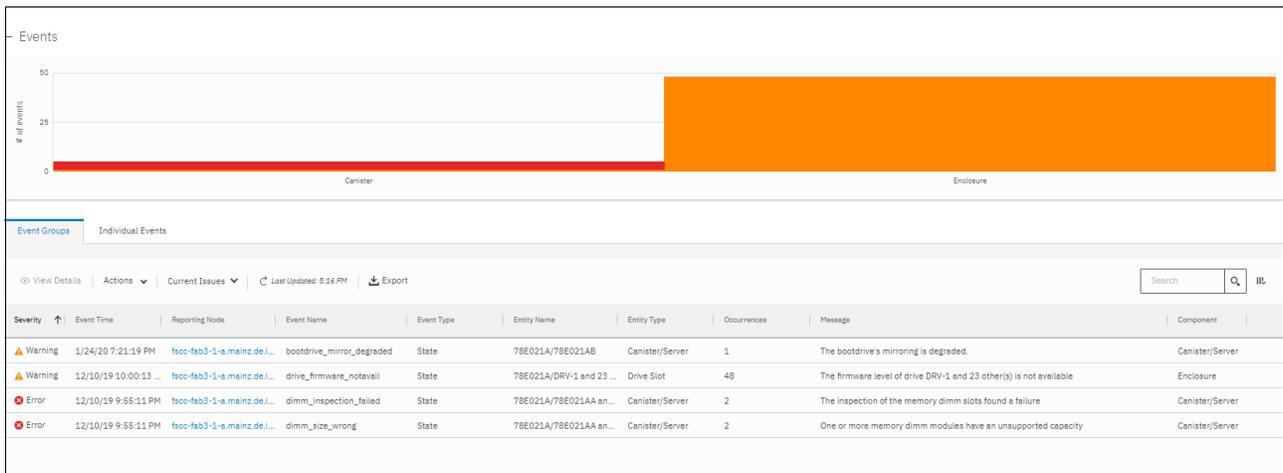


Figure 2-19 The Events page

The following filter options by event type are available as a drop-down list in the **Events** page shown in Figure 2-19 on page 21:

- ▶ **Current Issues** displays all unfixed errors and warnings.
- ▶ **Notices** displays all transient messages of type “notice” that were not marked as read. While active state events disappear when the related problem is solved, notices will stay forever until they are marked as read.
- ▶ **Current State** displays all events that define the current state of the entities, and excludes notices and historic events.
- ▶ **All Events** displays all messages, even historic messages and messages that are marked as read. This filter is not available in the Event Groups view because of performance implications.

You can mark events of type **Notices** as read to change the status of the event in the Events view. The status icons become gray if an error or warning is fixed, or if it is marked as read.

Some issues can be resolved by running the **Run Fix Procedure** action.

2.2.8 Event notification

The system can send emails and Simple Network Management Protocol (SNMP) notifications when new health events appear. Any combination of these notification methods can be used simultaneously. Use the **Monitoring** → **Event Notifications** page in the GUI to configure event notifications.

Sending emails

Use the **Monitoring** → **Event Notifications** → **Email Server** page to configure the email server where the emails should be sent. In addition to the email server, an email subject and the senders name can also be configured. The **Test Email** action enables you to send a test-email to an email address. See Figure 2-20.

The screenshot shows the 'Event Notifications' configuration page with the 'Email Server' tab selected. The page contains the following elements:

- Event Notifications** (Page Title)
- Email Server** (Active Tab), **Email Recipients**, **SNMP Manager** (Other Tabs)
- Email notifications enabled**
- IP address or host name:**
- Port:**
- Sender's email address:**
- Password:**
- Use different login:**
- Sender's name:**
- Subject:**
- Header:**
- Footer:**
- Test email address:**
-

Figure 2-20 Configure the email server

The emails can be sent to multiple email recipients, which are defined in the **Monitoring** → **Event Notifications** → **Email Recipients** page. For each recipient, you can select the components for which to receive emails, and the **For minimum severity level** (Tip, Info, Warning, or Error). Instead of receiving a separate email per event, optionally a daily summary email can be sent. Another option is to receive a **Daily Quota report**. See Figure 2-21.

Figure 2-21 Create email recipient

Sending SNMP notifications

Use the **Monitoring** → **Event Notifications** → **SNMP Manager** page to define one or more SNMP managers that will receive an SNMP notification for each new event. As opposed to Email notification, for SNMP notification no filters can be applied, and an SNMP notification is sent for any health event that occurs in the system.

The SNMP objects that are included in the event notifications are listed in Table 2-2.

Table 2-2 SNMP objects included in the event notifications

OID	Description	Example
.1.3.6.1.4.1.2.6.212.10.1.1	Cluster ID	317908494245422510
.1.3.6.1.4.1.2.6.212.10.1.2	Entity type	Drive Slot
.1.3.6.1.4.1.2.6.212.10.1.3	Entity name	SV44727220/DRV-1-6
.1.3.6.1.4.1.2.6.212.10.1.4	Component	Enclosure
.1.3.6.1.4.1.2.6.212.10.1.5	Severity	WARNING
.1.3.6.1.4.1.2.6.212.10.1.6	Date and time	17.10.2019 13:27:42.518
.1.3.6.1.4.1.2.6.212.10.1.7	Event name	drive_firmware_wrong

OID	Description	Example
.1.3.6.1.4.1.2.6.212.10.1.8	Message	The firmware level of drive DRV-1-6 is wrong.
.1.3.6.1.4.1.2.6.212.10.1.9	Reporting node	gssoi2.spectrum

Example 2-1 shows an SNMP event notification that is sent when a performance monitoring sensor is shut down.

Example 2-1 Event notification

```
SNMPv2-MIB::snmpTrapOID.0 = OID: SNMPv2-SMI::enterprises.2.6.212.10.0.1
SNMPv2-SMI::enterprises.2.6.212.10.1.1 = STRING: "317908494245422510"
SNMPv2-SMI::enterprises.2.6.212.10.1.2 = STRING: "NODE"
SNMPv2-SMI::enterprises.2.6.212.10.1.3 = STRING: "gss-11"
SNMPv2-SMI::enterprises.2.6.212.10.1.4 = STRING: "PERFMON"
SNMPv2-SMI::enterprises.2.6.212.10.1.5 = STRING: "ERROR"
SNMPv2-SMI::enterprises.2.6.212.10.1.6 = STRING: "18.02.2016 12:46:44.839"
SNMPv2-SMI::enterprises.2.6.212.10.1.7 = STRING: "pmsensors_down"
SNMPv2-SMI::enterprises.2.6.212.10.1.8 = STRING: "pmsensors service should be started and is
stopped"
SNMPv2-SMI::enterprises.2.6.212.10.1.9 = STRING: "gss-11"
```

The OID range .1.3.6.1.4.1.2.6.212.10.0.1 denotes ESS GUI event notification (trap), and .1.3.6.1.4.1.2.6.212.10.1.x denotes ESS GUI event notification parameters (objects).

The SNMP Management Information Base (MIB) file is available at the following location of each GUI node:

```
/usr/lpp/mmfs/gui/IBM-SPECTRUM-SCALE-GUI-MIB.txt
```

2.2.9 Dashboards

The **Monitoring** → **Dashboard** page provides an easy-to-read, single-page, real-time user interface that provides a quick overview of the system performance.

There are some default dashboards that are included with the product. Users can further modify or delete the default dashboards to suit their requirements, and can create additional new dashboards. The same dashboards are available to all GUI users, so modifications will be visible to all users.

A dashboard consists of several dashboard widgets that can be displayed within a chosen layout. There are widgets available to display performance metrics, system health events, file system capacity by file set, file sets with the largest growth rate in the last week, and timelines that correlate performance charts with health events. See Figure 2-22.

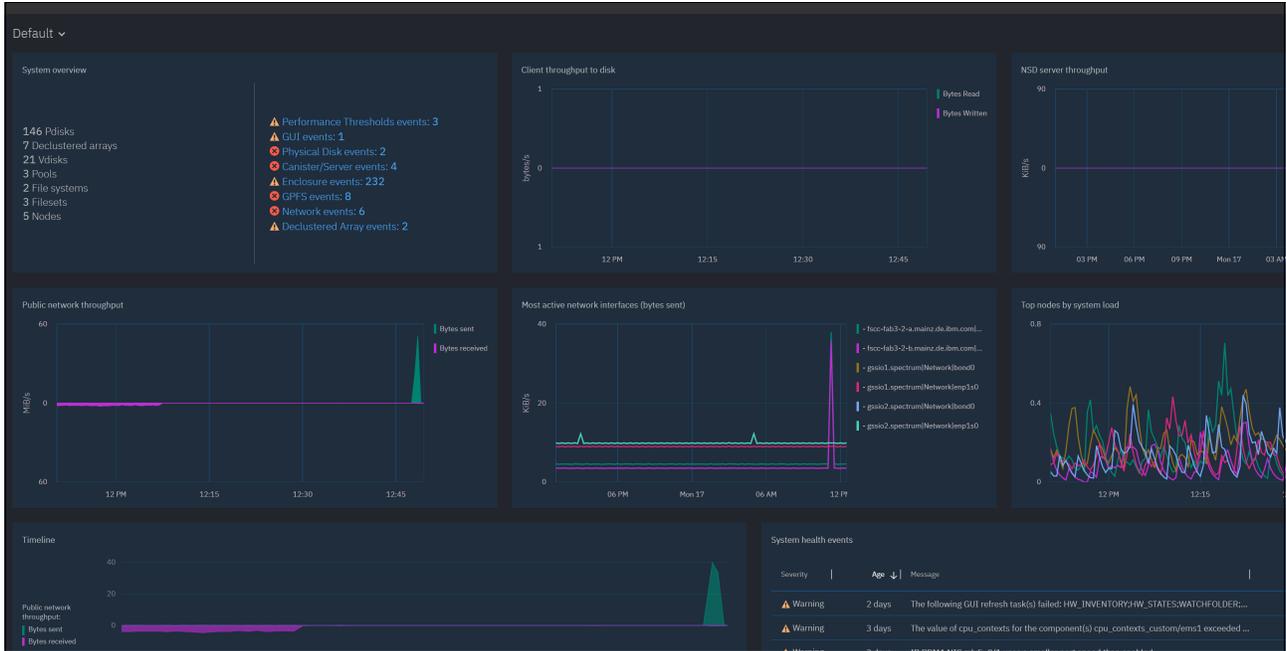


Figure 2-22 The dashboard

2.2.10 More information

The previous sections provided a rough overview of the GUI. For more detailed information on the GUI, read the *Monitoring and Managing the IBM Elastic Storage Server Using the GUI*, REDP-5471 IBM Redpaper publication and use the online help pages that are included within the GUI.

2.3 Software enhancements

In this section, we describe the software enhancements in ESS 3000.

2.3.1 Containerized deployment

The IBM Lab Services team can install an Elastic Storage Server 3000 as an included service part of acquisition. Alternatively, the customer's IT team can do the installation. The following documents provide information that you need for proper deployment, installation, and upgrade procedures for an IBM ESS 3000:

- ▶ *ESS 3000 Quick Deployment Guide:* ibm.biz/Bdqx3e
- ▶ *IBM ESS 3000: Planning for the system, service maintenance packages, and service procedures:* ibm.biz/Bdqx3a

2.3.2 Ansible

Ansible is used for complex orchestration tasks: automate in a language that approaches plain English, using SSH, with no agents to install on remote systems.

For IBM ESS 3000, ansible is used for performing updates, cluster creation, file system creation, and security options management. Because ansible provides support for running *tasks* over multiple hosts, update is easy to perform in this version.

Ansible streamlines file system creation by providing intelligent default settings and the flexibility to create new file systems or integrate with existing file systems in the cluster. You can create a VDisk and then later you can attach it to an existing file system if you have an existing cluster.

Various security features, such as Firewall, Admincentral, sudo, and SELinux, can be deployed on one or all the nodes from IBM ESS managerial controls using ansible in a containerized environment.

Users can use the ansible playbook to customise the security features required on a node, post its deployment, and log the status details of each executed task in relevant files.

You can see the *ESS Deployment Guide* at ibm.biz/Bdqx3e for details about security features usage.

2.3.3 The mmvdisk command

Although on previous versions of ESS it was possible to use other commands to manage IBM Spectrum Scale RAID, on ESS 3000 the only way to manage IBM Spectrum Scale RAID is with the `mmvdisk` command.

The `mmvdisk` command is an integrated command suite for IBM Spectrum Scale RAID. It greatly simplifies IBM Spectrum Scale RAID administration, and encourages and enforces consistent best practices with regard to server, recovery group, VDisk NSD, and file system configuration.

The `mmvdisk` command can be used to manage new IBM Spectrum Scale RAID installations. If you are integrating ESS 3000 with a setup that already has other ESS systems that are non-`mmvdisk` recovery groups, those need to be online converted into `mmvdisk` recovery groups before adding the ESS 3000 into the same cluster.

For more information about the `mmvdisk` command, see the following IBM Knowledge Center: https://www.ibm.com/support/knowledgecenter/en/SSYSP8_5.3.5/com.ibm.spectrum.scale.raid.v5r04.adm.doc/b18adm_mmvdisk.htm.

2.3.4 The mmhealth command

The `mmhealth` command is aimed to be the “one stop” for all things health-related on an IBM Spectrum Scale cluster. Although a cluster might be made up of many different types of components, `mmhealth` gives a holistic status of the health of the important cluster health issues.

For any type of cluster that includes the status of the GPFS daemons, the NODE software status, tracking of EVENTS that happened to the cluster, and the FILESYSTEM health status. The depth of the details for one NODE depends on a few factors:

- ▶ If the node is a software-only node, where IBM Spectrum Scale only formats external block devices to the cluster
- ▶ If the system is one that uses IBM Spectrum Scale RAID, such as the ESS 3000

In the ESS 3000 case, **mmhealth** monitors and reports the following non-exhaustive list:

- ▶ Hardware specific, same as other ESS hardware solutions:
 - Temperature of different sensors of the enclosure
 - Power supply hardware status
 - Fan speeds and status
 - Voltage sensors data
 - Firmware levels reporting and monitoring
 - Boot drive status and monitoring
- ▶ IBM Spectrum Scale RAID specific, the same as other IBM Spectrum Scale RAID solutions:
 - Recovery Groups status and monitoring
 - Declustered Array status and monitoring
 - Physical drives status and monitoring
 - VDisks status and monitoring
- ▶ IBM Spectrum Scale Software related, same as other IBM Spectrum Scale software related:
 - NSD status and monitoring
 - Network communication status and monitoring
 - GUI status and monitoring (of the GUI nodes)
 - CES status and monitoring (off the CES nodes)
 - File system status and monitoring
 - Pool status and monitoring
 - NSD protocol and statistics, and other protocols' statistics when applicable

The **mmhealth** command provides IBM Spectrum Scale software-related checks across all node and device types present in the cluster. Software RAID checks are present across all GNR offerings (such as ESS 5000, ESS 3000, and ECE). For devices (such as ESS 3000) that are integrated with IBM Spectrum Scale hardware, you also get the hardware checks and monitoring.

For details about how to operate with the **mmhealth** command, see the IBM Knowledge Center at the following link:

https://www.ibm.com/support/knowledgecenter/STXKQY_5.0.4/com.ibm.spectrum.scale.v5r04.doc/b11adm_mmhealth.htm

The mmhealth command changes to support ESS 3000

The **mmhealth** command, as described in “The mmhealth command” on page 27, has a specific component monitoring when an IBM Spectrum Scale Native RAID (GNR) environment is in-use. As of the December 2019 release of ESS 3000 (version 600x), **mmhealth** now supports GNR health monitoring on the 5148-AF8 solution (x86 based NVME platform).

The `mmhealth` command has been extended to support the additional hardware components of the ESS3000, and to address the needs of users looking to monitor the environment. This section initially references much of the current `mmhealth` information available through IBM Redbooks, command references, administration documents, and other publicly available resources. The second section describes the specific additional changes made in `mmhealth` to support ESS3000.

Support in the `mmhealth` command for GNR

This section provides pointers to much of the currently available documentation regarding `mmhealth` and GNR specific support.

The following link shows all of the current RAS events supported by `mmhealth`. These include all events supported by IBM Spectrum Scale, with a subset specific to GNR:

https://www.ibm.com/support/knowledgecenter/SSYSP8_5.3.5/com.ibm.ess.v5r35.pdg.doc/b11pdg_rasevents.htm.

Canister events are new to ESS 3000 and are described in “Canister events” on page 29. The rest of the events are applicable to ESS (legacy) and ESS 3000. See *IBM Spectrum Scale Erasure Code Edition: Planning and Implementation Guide*, REDP-5557.

This book describes the `mmhealth` command in the following sections:

- ▶ Section 7.7 shows general command usage
- ▶ Section 7.8 shows example use case scenarios

<https://www.redbooks.ibm.com/abstracts/redp5557.html?Open>

The following link is the main page for `mmhealth`. It shows the complete command usage, features, and examples:

https://www.ibm.com/support/knowledgecenter/STXKQY_5.0.4/com.ibm.spectrum.scale.v5r04.doc/b11adm_mmhealth.htm.

The `mmhealth` command changes to support ESS 3000

There are several major changes between legacy ESS and ESS 3000. The `mmhealth` command had to be updated to support monitoring these new features. The following list includes some of these differences:

- ▶ Architecture change to x86_64 from Power
- ▶ Support for NVME drives
- ▶ No external storage enclosures
- ▶ Dual canister design within single building block

The `mmhealth` command had to make several changes to support ESS 3000. The Canister events category was included to support many of the differences between legacy ESS and ESS 3000. The Server category also had to be adjusted. Both of these adjustments and other changes to `mmhealth` are included in the following sections.

Canister events

These events are new and specifically added to support the new Canister-based building-block configuration of the ESS 3000. Events related to the boot drive, temperature, CPU, memory, and more are included here:

https://www.ibm.com/support/knowledgecenter/SSYSP8_5.3.5/com.ibm.ess.v5r35.pdg.doc/b11pdg_rasevents_cannister.htm

A new command (**ess3kplt**) was created by GNR to provide CPU and memory health information to **mmhealth**:

/opt/ibm/gss/tools/bin/ess3kplt

Command usage:

ess3kplt -h

usage: ess3kplt [-h] [-t SELECTION] [-Y] [-v] [--local]

Optional arguments:

- h, --help show this help message and exit
- t SELECTION Provide selection keyword: [memory|cpu|all]
- Y Select report listing
- v Enable additional output
- local Select localhost option

This program can be used to inspect memory or CPU resources. Example 2-2 shows a sample output.

Example 2-2 Sample output

2020-03-07T23:29:56.783578 Retrieving resource info...

ESS3K Mem Inspection:

InspectionPassed: True
Total Available Slots: 24 (expected 24)
Total Installed Slots: 12 (expected 12 or 24)
DIMM Capacity Errors: 0 (Number of DIMMs with a size different from 32 GB)
DIMM Speed Errors: 0 (Number of DIMMs with a speed of neither 2400 nor 2666 MT/s)
Inspection DateTime: 2020-03-07 23:29:56.960696

ESS3K Cpu Inspection:

InspectionPassed: True
Total CPU Sockets: 2 (expected 2)
Total Populated Sockets: 2 (expected 2)
Total Enabled CPU Sockets: 2 (expected 2)
Total Cores: 28 (expected 28)
Total Enabled Cores: 28 (expected 28)
Total HTT: 2
Total Threads: 56 (expected 28 or 56)
CPU Speed Errors : 0 (Number of CPUs with a speed different from 2200 MHz)
Inspection DateTime: 2020-03-07 23:29:57.139119

Example 2-3 shows a sample verbose output.

Example 2-3 Sample verbose output:

ess3kplt -Y

ess3kplt:memory:HEADER:version:reserved:reserved:location:size:speedMTs:
ess3kplt:memorySummary:HEADER:version:reserved:reserved:availableSlots:installedSlots:capacityError:speedError:inspectionPassed:
ess3kplt:memory:0:1::CPU1_DIMM_A0:32 GB:2400:
ess3kplt:memory:0:1::CPU1_DIMM_A1:::
ess3kplt:memory:0:1::CPU1_DIMM_B0:32 GB:2400:
ess3kplt:memory:0:1::CPU1_DIMM_B1:::
ess3kplt:memory:0:1::CPU1_DIMM_C0:32 GB:2400:
ess3kplt:memory:0:1::CPU1_DIMM_C1:::
ess3kplt:memory:0:1::CPU1_DIMM_D0:32 GB:2400:

```

ess3kplt:memory:0:1:::CPU1_DIMM_D1:::
ess3kplt:memory:0:1:::CPU1_DIMM_E0:32 GB:2400:
ess3kplt:memory:0:1:::CPU1_DIMM_E1:::
ess3kplt:memory:0:1:::CPU1_DIMM_F0:32 GB:2400:
ess3kplt:memory:0:1:::CPU1_DIMM_F1:::
ess3kplt:memory:0:1:::CPU2_DIMM_A0:32 GB:2400:
ess3kplt:memory:0:1:::CPU2_DIMM_A1:::
ess3kplt:memory:0:1:::CPU2_DIMM_B0:32 GB:2400:
ess3kplt:memory:0:1:::CPU2_DIMM_B1:::
ess3kplt:memory:0:1:::CPU2_DIMM_C0:32 GB:2400:
ess3kplt:memory:0:1:::CPU2_DIMM_C1:::
ess3kplt:memory:0:1:::CPU2_DIMM_D0:32 GB:2400:
ess3kplt:memory:0:1:::CPU2_DIMM_D1:::
ess3kplt:memory:0:1:::CPU2_DIMM_E0:32 GB:2400:
ess3kplt:memory:0:1:::CPU2_DIMM_E1:::
ess3kplt:memory:0:1:::CPU2_DIMM_F0:32 GB:2400:
ess3kplt:memory:0:1:::CPU2_DIMM_F1:::
ess3kplt:memorySummary:0:1:::24:12:0:0:true:
ess3kplt:cpu:HEADER:version:reserved:reserved:location:speedMHz:status:status2:numCores:numCores
Enabled:numThreads:
ess3kplt:cpuSummary:HEADER:version:reserved:reserved:totalSockets:populatedSockets:enabledSocket
s:totalCores:enabledCores:totalThreads:speedErrors:inspectionPassed:
ess3kplt:cpu:0:1:::CPU0:2200:populated:enabled:14:14:28:
ess3kplt:cpu:0:1:::CPU1:2200:populated:enabled:14:14:28:
ess3kplt:cpuSummary:0:1:::2:2:2:28:28:56:0:true:

```

CPU and DIMM related events that `mmhealth` reports rely on the `ess3kplt` command in the ESS3000 environment.

2.4 RAS enhancements

ESS 3000 is an extension of the Elastic Storage Server (ESS) product family, and it is built on a common hardware platform shared with other IBM storage systems. This common hardware platform was originally designed to support various offerings by the FlashSystem and Storwize product families. That is why ESS 3000 includes the AF8 in its MTM (machine type model) to indicate the specific configuration of the common hardware platform.

ESS 3000 is targeted at delivering the following key traits in *Appliance customer experience*:

- ▶ Easy to order
- ▶ Easy to install
- ▶ Easy to upgrade
- ▶ Easy to use
- ▶ Easy to service

The following list includes the key components:

- ▶ Common IBM storage enclosure with commercial NVMe drives
- ▶ Red Hat Enterprise Linux (RHEL) 8.x with NVMe support
- ▶ IBM Spectrum Scale 5.0.4 software features and functions
- ▶ IBM Spectrum Scale Software RAID, also known as *GPFS Native RAID* (GNR)

ESS 3000 is a customer setup (CSU) product with a combination of customer-replaceable units (CRUs) and field-replaceable units (FRUs).

2.4.1 Enclosure overview

ESS 3000 offered a Samsung-only NVMe drive with the following capacity options at its initial GA in 4Q 2019 with either 12-drive or 24-drive installation options:

- ▶ 1.9 TB
- ▶ 3.8 TB
- ▶ 7.6 TB
- ▶ 15.3 TB

ESS 3000 uses a mirrored set of 800 GB M.2 SSD drives as the boot disks. The M.2 SSD includes the *Power Loss Protection (PLP)* feature. It offers only two per-canister memory configurations, as shown in Table 2-3.

Table 2-3 Memory configurations

Memory	Configuration details
384 GB	Half populated with 12 × 32 GB (6 slots × 2 CPUs) DIMMs
768 GB	Fully populated with 24 × 32 GB (12 slots × 2 CPUs) DIMMs

Given the IBM Spectrum Scale GNR design and the M.2 PLP feature to ensure the data persistency for GNR log files maintained in the boot disks, ESS 3000 does not require a Battery Backup Unit (BBU).

ESS 3000 offers two I/O adapter options:

- ▶ EC64: PCIe gen4 dual-port 100 Gb EDR InfiniBand adapter
- ▶ EC67: PCIe gen4 dual-port 100 Gb RoCE Ethernet adapter

Figure 2-23 shows the front view of the ESS 3000 enclosure.

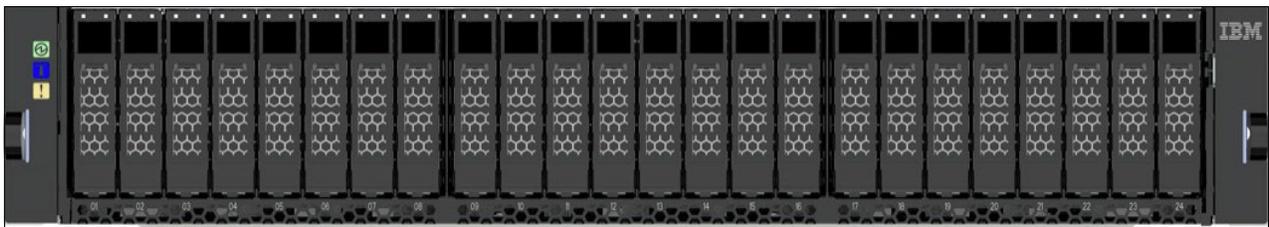


Figure 2-23 Front view of the ESS 3000 enclosure

Figure 2-24 shows the rear view of the ESS 3000 enclosure.



Figure 2-24 Rear view of the ESS 3000 enclosure

Figure 2-25 shows the rear view of a canister (bottom position). It highlights the key connectors for I/O and system management.

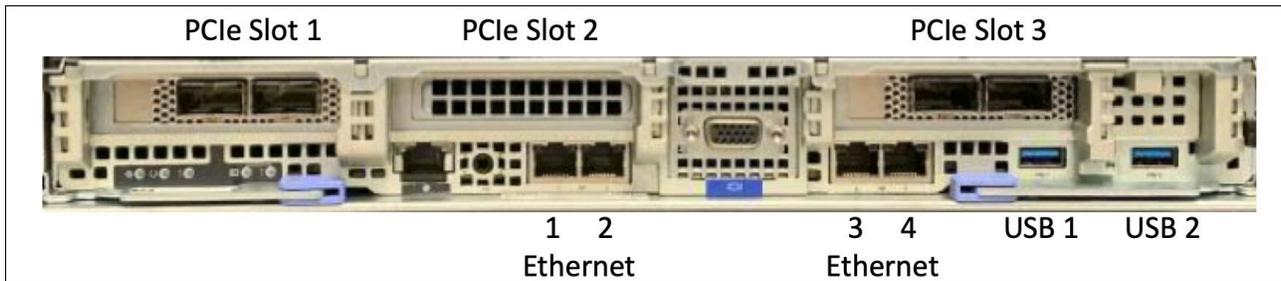


Figure 2-25 Rear view of a canister

In a typical configuration that includes 2 high-speed, dual-port adapters. They are installed in the Slot 1 and Slot 3 positions and referenced as *PCIe Adapter 1* and *PCIe Adapter 2* that also show the port numbers, as shown in Figure 2-26. The Slot 2 position is only used when the third adapter is installed in the Adapter miscellaneous equipment specification (MES) use case. There are two options to choose from for an Adapter MES: EC64 (InfiniBand) and EC67 (ethernet).

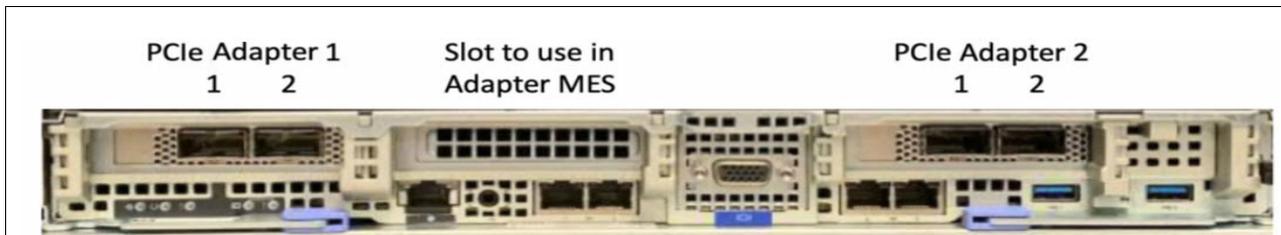


Figure 2-26 PCIe adapters and Adapter MES

2.4.2 Machine type model and warranty

ESS 3000 has a single MTM value: 5141-AF8. It includes a 3-year warranty (flex: 1-year warranty + 2 years maintenance).

ESS 3000 also offers same-day service upgrade options, as well as optional priced services including lab-based services (LBS) installation.

2.4.3 Components: FRU versus CRU

Compared to previous ESS models, ESS 3000 includes fewer replaceable parts based on its dual-canister architecture.

In general, ESS 3000 service strategy takes the following approach:

- ▶ For any HW component that is located inside a canister and requires opening up the cover of a canister in order to access it, it is considered to be FRU and requires IBM service personnel to perform such actions.
- ▶ For other HW components that can be accessed for maintenance without removing a canister, it is recommended as CRU that a user can perform a repair by following using GUI or instructions provided in IBM Knowledge Center for ESS.

The following are key HW components that fall into the FRU category:

- ▶ Canister
- ▶ Memory DIMM
- ▶ Adapter
- ▶ M.2 boot drive

The following are key HW components that fall into the CRU category:

- ▶ NVMe drive
- ▶ Drive blank
- ▶ Power supply unit

2.4.4 RAS features

ESS 3000 RAS features consists of the following features:

- ▶ Monitoring:
 - Hardware components
 - Firmware levels
 - GNR and IBM Spectrum Scale components
- ▶ Event notification
- ▶ Call home:
 - Software call home
 - Hardware call home
- ▶ IBM Spectrum Scale Healthchecker
- ▶ First time data capture (FTDC)

2.4.5 Maintenance and service procedures

Maintenance and service procedures are maintained in IBM Knowledge Center for ESS 3000 (https://www.ibm.com/support/knowledgecenter/SSYSP8/sts_welcome.html). Because the knowledge center is being actively maintained, it is recommended to always search for IBM ESS 3000 documentation in a web browser. After you enter the home page, you can find a Servicing option listed to bring you to the appropriate procedures. If preferred, you can also look up the PDF to download in the Table of Contents tab.

2.4.6 Software related RAS enhancements

The following section provides an overview of the software-related RAS enhancements in ESS 3000.

PDisk redundancy

If you compare the number of total PDisk paths in previous ESS configurations to the ESS 3000 you will notice a discrepancy. In previous ESS configurations, there are 2 paths from the active server and 2 paths from the backup server (that are not enabled unless the recovery group fails over to that server). In ESS 3000, each server only has 1 path to the data PDisks, but the paths from both servers are active.

The important takeaway is that each data PDisk still has 2 active paths.

Shared recovery group

In order to achieve optimal performance on the Elastic Storage System, we implemented a shared recovery group layout that allows both servers in an Elastic Storage System 3000 building block to concurrently access all of the available drives and their bandwidth. Instead of two paired recovery groups as in regular ESS, both servers access a single shared recovery group. This configuration allows the servers to drive the full bandwidth that the disks are capable of in both of the supported configurations: fully populated (24 disk) and half populated (12 disk). In this shared recovery group model, two user log groups are created per server node, allowing I/O to be evenly distributed across both nodes.

Serviceability enhancements

There are several enhancements for enclosure status.

Field Replaceable Unit (FRU) and Location

The enclosure status that the `mm1senclosure` command (and consequently the GUI) displays has been updated to provide the Field Replaceable Unit (FRU) number and the location of that FRU within the enclosure. Example 2-4 shows a failed temperature sensor.

Example 2-4 Failed temperature sensor

```
-----  
>mm1senclosure all -L --not-ok  
      needs nodes  
serial number  service  
-----  
78E00HL      yes      c202f06fs01a.gpfs.net  
  
component type  serial number  component id  failed value  unit  properties  fru  location  
-----  
tempSensor      78E00HL      41           yes  0      C           0111518  canister2_inlet
```

New enclosure components

There are several new enclosure components being reported by the `mm1senclosure` command.

- ▶ **canister:** The ESS 3000 is more appliance-like, failure of the top or bottom canister (for example server nodes) are called out.
- ▶ **cpu:** The failed CPUs that are associated with a canister are called out.
- ▶ **dimmm:** The memory modules that are associated with a canister are called out.

2.4.7 Integrated call home

The ESS 3000 will call home disk-related events. If one (or more) of the NVMe drives is reporting a problem from GNR and needs replacement, an event is caught by the monitoring service on the EMS and sent to the Electronic Service Agent (ESA) agent for processing. ESA screens the events (prevents duplicates, for example) and creates a PHM if action is needed.

The Elastic Storage Server Version 5.3.5 Problem Determination Guide (https://www.ibm.com/support/knowledgecenter/en/SSYSP8_5.3.5/ess_pdg.pdf) describes in detail how call-home works.

For a complete background and overview of ESA please go to the following link:

<https://www.ibm.com/support/knowledgecenter/linuxonibm/liaao/liaaokickoff.htm>

Example 2-5 is an example of how to generate a problem reported by ESA.

Example 2-5 How to generate a problem reported by ESA

Enable call-home monitoring on the EMS:

```
[root@ems9 samples]# ./callhomemon.sh
/usr/lib/python2.7/site-packages/urllib3/connectionpool.py:769: InsecureRequestWarning:
Unverified HTTPS request is being made. Adding certificate verification is strongly advised.
See: https://urllib3.readthedocs.org/en/latest/security.html
  InsecureRequestWarning)
Event successfully sent to ESA node for the end point 78E00T4, system.id
7d4eb5480ee6d08d187a8d2fcd91b13c, location 78E00T4-23, fru 3.84TB NVMe G3 .
/usr/lib/python2.7/site-packages/urllib3/connectionpool.py:769: InsecureRequestWarning:
Unverified HTTPS request is being made. Adding certificate verification is strongly advised.
See: https://urllib3.readthedocs.org/en/latest/security.html
  InsecureRequestWarning)
Event successfully sent to ESA node for the end point 78E00T4, system.id
7d4eb5480ee6d08d187a8d2fcd91b13c, location 78E00T4-24, fru 3.84TB NVMe G3 .
[I] Callhome Disk monitor successful.
bash: /sbin/opal-elog-parse: No such file or directory
[W] No serviceable event found on node fab3a-ib.
bash: /sbin/opal-elog-parse: No such file or directory
[W] No serviceable event found on node fab3b-ib.
[W] No serviceable event found on node ems9-ib.
```

Simulate two disks dead:

```
[root@ems9 ~]# mmvdisk pdisk change --rg ess3k --pdisk els24 --simulate-dead
[root@ems9 ~]# mmvdisk pdisk change --rg ess3k --pdisk els23 --simulate-dead
```

Ensure ESA caught the problem:

```
[root@ems9 bin]# ./esacli problem
Problem list:

Problem 3704184685dc4fd88e6090ed9d2f6665d:
    Status:          Open
    Service request: 31227754000

Problem 1f12e9b0405344e18d75528bdad6db04:
    Status:          Open
    Service request: 31228754000
```

Example 2-6 shows a PMR generated (PMR text portion).

Example 2-6 PMR generated (PMR text portion)

```
+SYSTEM GENERATED TEXT--D/T5141AF8--POCID=RAL -----20/02/21-14:25--UE   page 2
+SYSTEM GENERATED TEXT--D/T5141AF8--POCID=RAL -----20/02/21-14:25--UU
+SYSTEM GENERATED TEXT--D/T5141AF8--POCID=RAL -----20/02/21-14:25--UT
==> Machine Location Phone: 5121234567
RuleConfigID: 6,705   Service Request Type: H
Group ID: IPHMC      Group Name: 5141   Group Qualifier: AF8
ISO Country Code: US Customer ID:      Area: 02
```

vSessionId: PR12020022119255233945 OS:
Correlation Id:

Warranty: Y Extended Warranty: Y Status: 2
Entitlement System: CCPF Entitled: Y

Subject Type: IEPD/hardware
HTTPS/SES Dire/CNN - Electronic Customer Care Session.
Submitter:

page 3

REPORTING DEVICE : 5141 REPORTING MODEL : AF8
REPORTING SERIAL# : 78E00T4
REPORTING CEC TYPE : 5141 REPORTING CEC MODEL : AF8
REPORTING CEC SERIAL# : 78E00T4 REPORTING CEC LPAR :

OPERATING SYSTEM : TYPE: RELEASE LEVEL :
TAPES / CUM : CUM HIPER : LIC HIPER :
PROBLEM ID : NA SEVERITY : 2
PROBLEM DATE : 2020/02/21 PROBLEM TIME : 19:23:58
SYMPTOM : DSK00001

NUMBER OF OTHER SOFTWARE PRODUCTS: 1

COMPONENT : 5765DRPAS RELEASE : NA
PRODUCT NAME : LEVEL :

page 4

***** FAILING UNIT INFORMATION *****

Service Agent Date, Time: 2020-02-21 19:23:58 UTC
UTC: 2020-02-21 19:23:58 GMT
Machine Type/Feat: 5141
Model: AF8
Serial: 78E00T4
Unit Name: 78E00T4
Sys Feature/Fnc 20: 0
Bundled Problem Report: 0 of 0
Indicator Mode (LP/GL): GL
Sys Attn/Info Act (Y/N): Y

***** SUBSYSTEM AND OS INFORMATION *****

Server FW Level:
HMC FW Level:
Power System FW Level:
OS TYPE: NA OS Partition ID: 1
OS Level: NA
Device Level:

page 5

***** CALLOUT INFORMATION *****

FRU Count: 1
FRU 1 PN / Procedure: 3.84TB NVMe G3 SN: S43RNX0M500808 Priority: 5.46
Fault Indicator Activated (Y/N): Y
Location: e1s23:Rack ems9 U01-02, Enclosure 5141-AF8-78E00T4 Drive
23
Previous Local Problem ID:
Previous External Problem ID:
Previous Replace Date:

CONTACT :
ROLE :
Phone 1: Ext: Notes:
COMPANY : IBM
Address Type: PRIMARY
Address :

page 6

US

Language Preference :
Contact Preference :
Email Address :
InstantMessage :

FRU Data:

PART FRU	MODULE FRU	MESSAGE FRU				
MCH TYPE	PART NUM	PROB	MOD ID	MOD PROB	MSG ID	MSG PROB
Disk	3.84TB	100				

+SYSTEM GENERATED TEXT--D/T5141AF8--POCID=RAL -----20/02/21-14:25--UT
***** START OF NORMAL NOTE LOG *****

page 7

system.partition.id : 1
eed : [{data.location=NA, data.description=mm1spdisk output}]
event-details : e1s23:DA1:ess3k simulatedDead/replace/01008.160
event.serviceability : callhome
event.id : 3704184685dc4fd88e6090ed9d2f665d
event.original.timestamp : 2020-02-21 19:23:58
srcWords : string
resource.name : e1s23:DA1:ess3k
esa.problem.id : 3704184685dc4fd88e6090ed9d2f665d
esa.system.id : 7d4eb5480ee6d08d187a8d2fcd91b13c
event-description : ESS500-ReplaceDisk-78E00T4-23
error.code : DSK00001
failingunit-additional-data : {indicator-mode=GL,
bundle-problem-report-max=0, system-attention-info=true,
sbundle-problem-report-max=0, system-feature=0,
bundle-problem-report-number=0}
esa.event.type : Problems
frus extra details : [{fru.serial: S43RX0M500808, fru.priority: 5.46,
location.code: e1s23:Rack ems9 U01-02, Enclosure 5141-AF8-78E00T4 Drive
23, fru.type: P, fru.probability: 100, original.part.number: 3.84TB NVMe
G3 Tier-1 Flash, fault.indicator.activated: true, controlling.machine:
ems9, enclosure.info: 78E00T4-23, replacement.part.number: 3.84TB NVMe
G3 Tier-1 Flash, fru.description: 3.84TB NVMe G3, additional.data: Disk
5.3.5.1-20200219T203042Z_ppc64le_datamanagement
kernel: 3.10.0-957.35.2.el7.ppc64le
MOFED: MLNX_OFED_LINUX-4.7-3.2.9.1:
mm1spdisk output =>

page 8

mm1spdisk:pdisk:HEADER:version:reserved:reserved:replacementPriority:pdiskName:paths:recoveryGroup:declusteredArray:state:capacity:freeSpace:fru:location:WWN:server:reads:writes:bytesReadInGiB:bytesWrittenInGiB:IOErrors:IOTimeouts:mediaErrors:checksumErrors:pathErrors:relativePerformance:dataBadness:rgIndex:userLocation:userCondition:vendor:product:revision:

page 9

```
serial:hardwareType:nPathsActive:nPathsTotal:expectedPathsActive:expecte
dPathsTotal:nPathsActiveWrong:nPathsTotalWrong:SMARTwriteErrorsCorrected
WithoutDelay:SMARTwriteErrorsCorrectedWithDelay:SMARTwriteErrorsNotCorre
cted:SMARTreadErrorsCorrectedWithoutDelay:SMARTreadErrorsCorrectedWithDe
lay:SMARTreadErrorsNotCorrected:SMARTnonMediumErrors:SMARTpowerOnMinutes
:SMARTlastUpdate:nsdFormatVersion:paxosAreaOffset:paxosAreaSize:logicalB
lockSize:ssdEndurancePercentage:
mm1spdisk:pdisk:0:2:::5.46:e1s23::ess3k:DA1:simulatedDead/replace/01008.
160:3839700762624:3835405795328:3.84TB NVMe G3
:78E00T4-23:eui.343352304D5008080025385800000004:
:153159690:582661708:14080.194:48678.507:0:0:0:0:0:0:0.947:0.000:16:Rac
k ems9 U01-02, Enclosure 5141-AF8-78E00T4 Drive
23:replaceable:144D:3.84TB NVMe G3 Tier-1
Flash:SN1IISN1I:S43RX0M500808:NVMe:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:Thu Feb
20 18.43.48 2020 (1582242228):Unknown:Unknown:Unknown:4096:1:}]
***** END OF NORMAL NOTE LOG *****
+SYSTEM GENERATED TEXT--D/T5141AF8--POCID=RAL -----20/02/21-14:25--UT
***** Start of Self Defining Form *****
```

page 10

ProblemDetails

```
ReportingApplication: Service Agent - csESS
ReportingProduct: ESS-600
ProductMTMS: IBM 5141-AF8 78E00T4
```

```
***** End of Self Defining Form *****
+SYSTEM GENERATED TEXT--D/T5141AF8--POCID=RAL -----20/02/21-14:25--UT
IEPDESKTOP: Additional information such as hardware or software
inventory for this machine may be available on the eDesktop located at
https://w3spp.cnp.eventsqslb.ibm.com/wps/myportal/eservice.
```

page 11

```
IEPDCCPF01: CCPF method [UpdateEntitlement] resulted in entitled.
+SYSTEM GENERATED TEXT--D/T5141AF8--L095/ESSAD -----20/02/21-14:25--UE
S7> CALL SYS DOWN=
+SYSTEM GENERATED TEXT--D/T5141AF8--POCID=RAL -----20/02/21-14:25--UT
IEPDRTE: Routing for Assign Type [Primary] Routing ID = [ESSCH] Routing
Level = [5] Queue = [null] Center = [null]
+SERMON AMERICA -D/T5141AF8--L095/ESSAD -P2S2-20/02/21-14:31--CC
S6> SERVICE GIVEN= 99
Close call: Exempted Customer
```

2.4.8 Software call home

Software call home is supported in ESS through the **gsscallhomeconf** command. Software call home will leverage **mmcallhome** to send back data to IBM on a regular basis for problem analysis if issues occur.

For more information about **mmcallhome**, see the main page here:

https://www.ibm.com/support/knowledgecenter/STXKQY_5.0.4/com.ibm.spectrum.scale.v5r04.doc/b11adm_mmcallhome.htm

For more information about the **gsscallhomeconf** command, see this main page:

https://www.ibm.com/support/knowledgecenter/SSYSP8_5.3.5/com.ibm.ess.v5r35.cmd.doc/b18dep_gsscallhomeconf.htm

2.4.9 Performance

Measurements in the IBM lab on a freshly installed and fully populated Elastic Storage System 3000 (ESS 3000) with a 4 MiB file system block size have achieved sequential read performance of over 43 GBps and sequential write performance of over 34 GBps when using an InfiniBand network with Remote Direct Memory Access (RDMA) enabled.

Note: The performance measurements referenced here were made using standard benchmarks in a controlled environment. The actual performance of any given Elastic Storage System 3000 will vary depending on a number of factors, such as the network, workload characteristics, and the configuration of the client nodes. Therefore, no assurance can be given that any given Elastic Storage System 3000 can achieve results similar to those stated here.

Some factors related to Elastic Storage System 3000 performance are listed in the following sections.

Network

From a network perspective, the first choice that must be made is to decide between configuring IBM Spectrum Scale to use Ethernet or InfiniBand. When integrating an ESS 3000 into an existing environment, the choice of network is often predetermined by the existing environment, but some trade-offs between these two options should be considered.

Ethernet networks are more common in most data centers, because InfiniBand configurations have often been associated with more niche high-performance computing (HPC) solutions. When choosing an Ethernet configuration, it's worth noting that choosing TCP/IP for the data transport protocol will reduce the total performance that the ESS 3000 is capable of. This is because TCP/IP latencies are higher than RDMA latencies, and the total read and write bandwidth that can be delivered is reduced when using TCP/IP only.

Using RDMA over ethernet (RoCE) is not, as of the writing of this document, generally supported on the ESS 3000 without an RPQ. If you're interested in a RoCE solution, consult with your sales representative, and follow-up will be done to determine the viability of RoCE in your environment.

To realize the available bandwidth of higher throughput Ethernet networks (for example 100 Gbps) using TCP/IP only, a single ESS 3000 building block might require multiple TCP/IP connections (so multiple client nodes might be required to achieve optimal bandwidth).

InfiniBand networks have traditionally been capable of higher bandwidths and lower latencies but the gap between these two technologies is narrowing. Deployments using InfiniBand have traditionally enabled RDMA, and easy-to-implement RDMA support has been one of the differentiators between InfiniBand and Ethernet in regard to achieving optimal bandwidth and latency with minimal CPU resources.

The following sections describe some of the relevant IBM Spectrum Scale network configuration options that define how the network is used by IBM Spectrum Scale.

Configuration via mmchnode

See the **mmchnode** manual page at

https://www.ibm.com/support/knowledgecenter/STXKQY_5.0.4/com.ibm.spectrum.scale.v5r04.doc/b11adm_mmchnode.htm for more details.

Daemon node name defines the host name or IP address over which the GPFS daemons communicate via TCP/IP. (Some IBM Spectrum Scale daemon communication always occurs over TCP/IP.) It is assigned when running `mmcluster` or `mmaddnode`, and can later be changed via the `mmchnode -daemon-interface` option.

Configuration via `mmchconfig`

See the `mmchconfig` page at

https://www.ibm.com/support/knowledgecenter/STXKQY_5.0.4/com.ibm.spectrum.scale.v5r04.doc/blladm_mmchnode.htm for more details:

- ▶ The `verbsRdma` option controls whether RDMA (instead of TCP) is used for data transfers. Valid values are `enable` and `disable`.
- ▶ The `verbsRdmaSend` option controls whether RDMA (instead of TCP) is used for most non-data transfer IBM Spectrum Scale daemon-to-daemon communication. Valid values are `enable` and `disable`.
- ▶ The `verbsPorts` option specifies the device names and port numbers that are used for RDMA transfers between IBM Spectrum Scale client and server nodes. You must enable `verbsRdma` to enable `verbsPorts`.

NVMe drives

Non-volatile memory express (NVMe) is an interface by which non-volatile storage media can be accessed via a PCIe bus. As a result of the efficiency of the protocol, NVMe generally provides better performance over alternatives, such as Serial Advanced Technology Attachment (SATA), when comparing devices that share the same underlying technology.

Note: A NAND (short for NOT AND) flash NVMe drive has the potential for improved performance over a NAND flash SATA drive because of its more efficient bus connection and protocol improvements. For example, NVMe allows for longer command queues.

Note that the time it takes for the NVMe drives to complete an I/O request accounts for only a portion of the overall time that it takes for IBM Spectrum Scale to complete the I/O request. To get a sense of where the time is being spent in handling I/Os, you can run the following command on the ESS 3000 and client nodes that are making the I/O requests:

```
/usr/lpp/mmfs/bin/mmdiag --iohist
```

At the lowest layer, we have the physical disk (pd) I/O times, obtained by running this command on an ESS 3000 server and looking at the NVMe drive I/O latencies. For example, in Example 2-7, 144 (512 byte) sectors were read in 187 microseconds.

Example 2-7 144 (512 byte) sectors were read in 187 microseconds

I/O start time	RW	Buf type	disk:sectorNum	nSec	time ms	tag1	tag2	Disk UID typ	NSD node	context
[..]										[..]
[..]										[..]
23:38:56.285967	R	data	11:3794335312	144	0.187	1259008	0	COA8CA73:5D34DEA6 pd		Pdisk [..]

To look at the I/O latencies of requests at the NSD layer on the ESS 3000 server, look for `srv` layer I/O times. These times show I/O latencies that account for more than just the disk I/O times, including network shared disk processing on the server.

On the Elastic Storage System 3000, disk I/Os are generally faster than on previous Elastic Storage System models, which also means that the ratio of time spent in remote procedure call (RPC) overhead will tend to be higher, relative to the actual disk I/O times. For this reason, systems that support RDMA should enable the `verbsRdmaSend` option discussed previously, so that RPCs can be handled via low latency RDMA operations.

Example 2-8 shows a 352 microsecond NSD server (srv) layer I/O on an ESS 3000, which corresponds to the previously shown 144 microsecond PDisk I/O (the I/O shows 144 sectors previously, instead of the 128 sectors shown in the following example, because the PDisk layer I/O accounts for additional sectors read for checksum validation).

Example 2-8 352 microsecond NSD server (srv) layer I/O on an Elastic Storage System 3000

I/O start time	RW	Buf type	disk:sectorNum	nSec	time ms	tag1	tag2	Disk UID typ	NSD node	context
23:38:56.285897	R	data	4:17514496	128	0.352	1259008	0	C0A8CA73:5D34DF0B srv	13.2.202.30	NSDWorker

Note that, if an I/O is satisfied from the GNR cache, the NSD I/O request will be displayed as per the previous example, but there will be no corresponding pdisk level I/O (since the data was read from the GNR cache on the server side, no disk I/O was required). Note that, as drive accesses on the Elastic Storage System 3000 are more efficient than comparable drive accesses on non-NVMe devices, the relative benefit of data residing in the GNR disk cache will be lower, but, since elements can be more efficiently swapped in and out of the cache, there is still potential for good caching related performance improvements.

To see the latency of I/O requests from the client's perspective, on the client look for 'cli' I/O times in the output of `mmdiag --iohist` (These times include network processing time and the time that requests wait for initial processing on the server). For example, looking at the previously shown 128 sector I/O, we see it took about 864 microseconds from the client's perspective (see Example 2-9).

Example 2-9 mmdiag --iohist output

I/O start time	RW	Buf type	disk:sectorNum	nSec	time ms	tag1	tag2	Disk UID typ	NSD node	context
23:38:56.285573	R	data	4:17514752	128	0.864	1259008	0	C0A8CA71:5D34DF24 cli	13.1.202.90	MBHandler

Shared Recovery Group

In order to achieve optimal performance on the Elastic Storage System, we implemented a shared recovery group layout that allows both servers in an ESS 3000 building block to concurrently access all of the available drives and their bandwidth. Instead of two paired recovery groups as in regular ESS, both servers access a single shared recovery group, which allows the servers to drive the full bandwidth that the disks are capable of in both of the supported configurations: fully populated (24 disk) and half populated (12 disk). In this shared recovery group model, two user log groups are created per server node, allowing I/O to be evenly distributed across both nodes.

In the shared recovery group model, optimal performance is achieved when both servers access all the drives in parallel, because a single ESS 3000 server doesn't have sufficient PCIe bandwidth to drive all 24 disks at full bandwidth. When diagnosing performance issues, verify that both servers in an ESS 3000 building block are delivering roughly equivalent bandwidth (if they're not, look for problems on the slower server).

For reference, see the following website:

https://www.ibm.com/support/knowledgecenter/SSZL24_6.0.0/com.ibm.ess3000.v6r00.pdf.doc/b18pdg_rgissues_3000.htm

Also see the `mmvdisk` and `mmvdisk recoverygroup` pages:

https://www.ibm.com/support/knowledgecenter/SSYSP8_5.3.5/raid_admin.pdf



Planning considerations

This chapter provides planning considerations for installing ESS 3000. It includes the following sections:

- ▶ 3.1, “Planning” on page 44
- ▶ 3.2, “Standalone environment” on page 45
- ▶ 3.3, “Mixed environment” on page 46

3.1 Planning

When ordering an ESS 3000 there are certain functional and non-functional requirements that need to be fulfilled before the actual order can be run. Those include the following requirements.

3.1.1 Technical and Delivery Assessment (TDA)

A TDA is an internal IBM process that includes a technical inspection of a completed solution design. Technical Subject Matter Experts (SMEs) who were not involved in the solution design participate to determine:

- Will it work?
- Is the implementation sound?
- Will it meet customer requirements and expectations?

There are two TDA processes. First is the pre-sales TDA. This can be done using the FOS Design Engine tool that can be found on the following link: <http://www.ibm.biz/FOSDesignEngine>.

Second is the pre-install TDA. SMEs also evaluate the customer's readiness to install, implement, and support the proposed solution. This can be done with the IMPACT tool that can be found on the following link: <https://www.ibm.com/tools/impact/>.

The previous TDA processes have assessment questions, but also baseline benchmarks, that need to be performed before the order can be fulfilled. Those tools are driven by IBM sales or resellers, so they can help and direct you regarding this process.

3.1.2 Hardware remarks

These include the hardware parts that are mandatory to have but are not included inside of the ESS 3000 building block (2U). There are two types of those requirements:

- ▶ The ones that must be IBM-provided, such as the management switch
- ▶ The ones that can be either customer-provided or IBM-provided, such as the high-speed network or rack

Rack solution

The ESS 3000 comes with at least one rack from IBM; it can come with more if multiple building blocks (BB) are ordered. The rack also holds the management server and the management switch. If the high-speed switches are ordered from IBM, those are also included in the rack.

Although the preferred option is the rack version of the ESS 3000 solution, it is possible to order ESS 3000 without the rack. If you choose to follow this path, you need to verify that the rack can hold the weight of the solution, and that the PDUs on the rack are the right ones for the ESS 3000 solution. In addition to that, you need to contact IBM to configure the management switch that comes with the solution.

Management switch

The ESS 3000 first building block on a site comes with a management switch from IBM (8831-S52). This switch is part of the ESS 3000, and it is not an independent part that can be replaced with an equivalent hardware by the customer.

High-speed network

As with any other IBM Spectrum Scale system, the ESS 3000 requires a high-speed network to be used as storage network. On some documentation, this network is referred to as a *Clustering network* in the product documentation. The hardware for the high-speed network can be provided by IBM or the customer. If it is being provided by the customer, it must be compatible with the network interfaces that the ESS 3000 supports. See section 2.1.1, “Canisters and servers” on page 8 to see the available network options on ESS 3000.

Enterprise Management Server (EMS)

The ESS 3000 would come with a management server (5148-21L) on standalone installs or on IBM Spectrum Scale setups where an EMS is not available with previously installed ESS. This is also a mandatory bundle of the solution when another EMS is not available. For more details about the EMS server, see the following IBM Knowledge Center link:

https://www.ibm.com/support/knowledgecenter/en/POWER8/p8hdx/5148_211_landing.htm.

3.2 Standalone environment

This section is about best practices for deploying and making the most out of a standalone Elastic Storage System 3000 (ESS 3000).

A standalone Elastic Storage System 3000 unit, known as a *building block*, must minimally consist of the following components:

- ▶ One EMS node in a 2U form factor
- ▶ One ESS 3000 node in 2U form factor
- ▶ 1 GbE or 10 GbE Network switch for management network (1U)
- ▶ 100 Gb high speed IB or Ethernet network for internode communication (1U)

The EMS node acts as the administrative end point for your ESS 3000 environment. It performs the following functions:

- ▶ Hosts the Spectrum Scale GUI
- ▶ Hosts Call Home services
- ▶ Hosts system health and monitoring tools.
- ▶ Manages cluster configuration, file system creation, and software updates
- ▶ Acts as a cluster quorum node

The Elastic Storage System 3000 features a brand-new container-based deployment model that focuses on ease of use. The container runs on the EMS node. All of the configurations tasks that were performed by the `gssutils` utility in legacy ESS are now implemented as ansible playbooks that are run inside of the container. These playbooks are accessed using the `ess3krun` command.

The `ess3krun` tool handles almost the entire deployment process, and is used to install software, apply updates, and deploy the cluster and file system. Only minimum initial user input is required, and most of that is covered by the TDA process prior to setting up the system. The `ess3krun` tool automatically configures system tuneables to get the most out of a single ESS 3000 system. File system parameters and IBM Spectrum Scale RAID Erasure code selection can be customized from their defaults before file system creation.

For more information about deployment customization, see the *ESS 3000 Quick Deployment Guide*:

https://www.ibm.com/support/knowledgecenter/SSZL24_6.0.0/pdf/ess3000_sdg.pdf?view=kc.

The following are some of the recommended practices:

- ▶ Refrain from running admin commands directly on the ESS 3000 I/O canisters. Use the EMS node instead.
- ▶ Do not mount the file system on the ESS 3000 I/O canisters because this consumes additional resources. The file system must be mounted on the EMS node for the GUI to function properly.
- ▶ To access the file system managed by the ESS 3000 building block, you must use external GPFS client nodes or protocol nodes.
- ▶ On a single building block deployment, the I/O canister nodes are specified as GPFS cluster/file system manager nodes while the EMS node isn't. Although the EMS node is considered the building block's primary management server, avoid specifying the EMS node as a manager node. The GPFS management role is an internal designation used to manager the cluster and the file system, and does not directly concern the function of the EMS node.

3.3 Mixed environment

This section is about interacting with already existing ESS environments and considerations to take into account when integrating an ESS 3000 in a mixed-vendor environment for migration purposes, or as another high-end storage tier.

3.3.1 Adding ESS 3000 to an existing ESS cluster

ESS 3000 comes with its own containerized deployment automatism. However, this automated deployment is intended basically to work for initial deployments, and for some subsets of various configuration scenarios. IBM Spectrum Scale can be configured to fit into much wider complex cluster scenarios. In this chapter, we describe a practical approach to take advantage of the deployment tools and the more flexible native commands to customize IBM Spectrum Scale by using the `mmvdisk` command.

Consider an example that has a more complex but pretty common setup, consisting of a storage cluster and 2 independent client clusters. As highlighted in Figure 3-1 on page 47, we now add the ESS 3000 as one more physical back end to the environment.

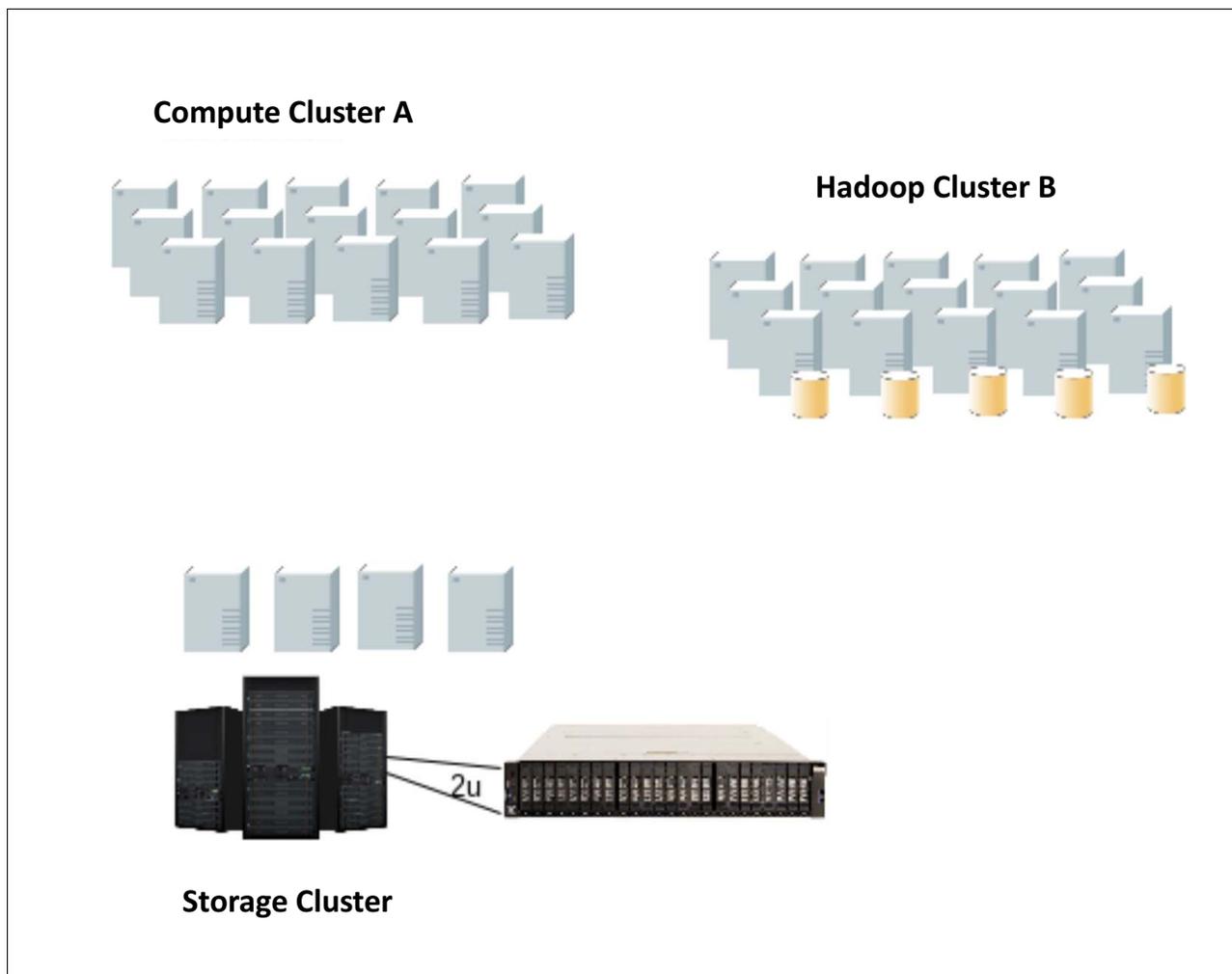


Figure 3-1 A mixed environment

Referring to initial setup, we start with deploying the hardware and firmware directly from the default deployment procedure:

1. First, you need to verify the prerequisites. The existing ESS storage cluster should be upgraded primarily to the latest level. *The minimum required level on the EMS is ESS 535 to work with the new ESS 3000 deployment.*
2. Follow all of the steps from the beginning of the Quick Deployment Guide until you reach this step “Update the canister nodes.”
3. Make sure that your previous steps have run successfully, so that you can access the canister nodes. Finally, as described in the Quick Deployment Guide, run the following command: **ess3krun -N ess3k1a,ess3k1b --offline update.**

After this step, the ESS 3000 is ready to be used for adding it into the existing cluster. Adding the nodes into the cluster and configuring them can be done with the **ess3krun** command for certain scenarios.

4. Next, you need to add the nodes into the cluster. Make sure that the mandatory environmental requirements are met, such as networking, time synchronization, DNS, user ID management, SSH access, and so on.

- Then add the ESS 3000 IO nodes to the cluster by logging in to a node which is already part of the cluster. See Example 3-1, which shows the existing cluster.

Example 3-1 Starting point

```
root@ems1 .ssh]# mmlscluster
GPFS cluster information
```

```
=====
```

```
GPFS cluster name:      main.spectrum
GPFS cluster id:       11272164317294188905
GPFS UID domain:      main.spectrum
Remote shell command:  /usr/bin/ssh
Remote file copy command: /usr/bin/scp
Repository type:      CCR
```

Node	Daemon node name	IP address	Admin node name	Designation
1	ems1rd.test	10.10.10.20	ems1.spectrum	quorum-manager
2	gssiolrd.test	10.10.10.21	gssiol.spectrum	
3	gssio2rd.test	10.10.10.22	gssio2.spectrum	

- Now we add the 2 canister nodes to the cluster and accept the license agreement. Run the following command:

```
[root@ems1 .ssh]# mmaddnode -N fsc- fab3-2-a, fsc- fab3-2-b
```

See Example 3-2.

Example 3-2 Add the 2 canister nodes

```
Thu Jan 23 11:37:39 CET 2020: mmaddnode: Processing node fsc- fab3-2-a.test
Thu Jan 23 11:37:40 CET 2020: mmaddnode: Processing node fsc- fab3-2-b.test
mmaddnode: Command successfully completed
mmaddnode: Warning: Not all nodes have proper GPFS license designations.
Use the mmchlicense command to designate licenses as needed.
mmaddnode: Propagating the cluster configuration data to all
affected nodes. This is an asynchronous process.
```

```
[root@ems1 .ssh]# mmchlicense server --accept -N all
```

The following nodes will be designated as possessing server licenses:

```
ems1.spectrum
gssiol.spectrum
gssio2.spectrum
fsc- fab3-2-a.test
fsc- fab3-2-b.test
```

```
mmchlicense: Command successfully completed
mmchlicense: Propagating the cluster configuration data to all
affected nodes. This is an asynchronous process.
[root@ems1 .ssh]#
```

- From now on, the `mmvdisk` command will help to completely configure the ESS RGs and disks. The ESS 3000 hardware support information is not available on the POWER8 EMS node. Therefore, you need to run the `mmvdisk` command from an ESS 3000 canister node. Proceed from one of the ESS 3000 canister nodes. Verify that the ESS 3000 is set up correctly by using the following command: `mmhealth cluster show`.

8. The next step is the creation of the RG group. The IBM Spectrum Scale Native RAID can be configured with several topologies. The classical known ESS models were deployed in the twin-tailed building block approach. Now with the new model of an ESS 3000 building block, the RG layout that was chosen for the topology, is the so called *shared RG model*.

So, only one RG per ESS 3000 gets created. Before proceeding, make sure to check that the hardware is ready and recognized by `mmvdisk`. The `mmvdisk` command scans and checks the environment and maps the hardware to an ESS 3000 known topology. See Example 3-3.

Example 3-3 The mmvdisk command

```
mmvdisk server list -N all --disk-topology
```

node number	server	needs attention	matching metric	disk topology
1	c202f06fs04a-ib0.gpfs.net	no	100/100	ESS3K SNO 24 NVMe
2	c202f06fs04b-ib0.gpfs.net	no	100/100	ESS3K SNO 24 NVMe

Note: If you receive a different output and the matching metric is not 100/100, report the issue to IBM.

9. Next, you need to configure the recovery group (RG). Don't use special characters when naming your RG. The underscore and dash and dot are allowed in `mmvdisk` RG names. See Example 3-4.

Example 3-4 Configure the recovery group

```
[root@fscs-fab3-2-a nvme_test]# mmvdisk rg create --recovery-group RGess3k
--node-class ess3k
mmvdisk: Checking node class configuration.
mmvdisk: Checking daemon status on node 'fscs-fab3-2-a.test'.
mmvdisk: Checking daemon status on node 'fscs-fab3-2-b.test'.
mmvdisk: Node 'fscs-fab3-2-a.test' has a shared recovery group disk topology.
mmvdisk: Node 'fscs-fab3-2-b.test' has a shared recovery group disk topology.
mmvdisk: Creating recovery group 'RGess3k'.
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: Created recovery group 'RGess3k'.
[root@fscs-fab3-2-a nvme_test]#
```

10. After creating the RG, the ESS 3000 storage can now be used for creating new file systems or creating NSDs (based on VDisks) to extend or enhance existing file systems. In our example, the new ESS 3000 was integrated in the old Storage cluster and by having the RG set up now, we continue with Scenario-1.

3.3.2 Scenario-1: Using ESS 3000 for metadata NSDs for the existing file system

In this example, we describe how to use some of the ESS 3000 storage to create NSDs to place the metadata from the existing file system. The following steps are performed on an ESS 3000 system:

1. First we need to create a VDisk set. A VDisk set is needed to collect all required information for the subsequent steps to create disks. In a VDisk set, we define the blocksize, the erasure coding level, the size, and some more specific settings for the disks.

We will show how to create a VDisk set in the next steps. Within the VDisk set definition, there is also the placement of the disk specified by giving the RG names and DA. In ESS 3000 there is only one DA, but keep in mind, that same `mmvdisk` command can be used to manage other building blocks with multiple DAs, for example the Hybrid models. A VDisk set can span multiple building blocks to allocate space for the disks.

Because we want to add disks into the existing file system, we need to check the specifications, such as the blocksize, maybe failure groups, and so on. Example 3-5 shows how to check the blocksize.

Example 3-5 Checking the blocksize

```
# check existing fs blocksize

[root@fscc-fab3-2-a nvme_test]# mmvdisk fs list

file system  vdisk sets
-----  -----
essGL2_4m    vsESSgl2

[root@fscc-fab3-2-a nvme_test]# mm1sfs essGL2_4m -B
flag          value          description
-----  -----  -----
-B           4194304       Block size
[root@fscc-fab3-2-a nvme_test]#
[root@fscc-fab3-2-a nvme_test]#
```

2. You can check further settings, for example, the smallest or largest disk size per pool. For demonstration purposes, we just list the block size and continue.
3. So you see that the file system block size is 4 MB. In our example, it is enough to add a little space (5% of the ESS 3000's total capacity) for holding metadata. The `mmvdisk` command then calculates the allocation of the requested space, and creates a balanced amount of VDIs within all of the recovery groups, given by the administrator.
4. As previously, we continue, using the `mmvdisk` command.

Tip: The set-size can also be specified in absolute capacity numbers by, for example, “G” for GB, “T” for TB, or in percentage of all available space in a DA from the RG.

If no DA is specified when using `mmvdisk`, then the default DA1 is chosen for allocating the space. Depending on the ESS 3000 model (capacity of drives) and erasure coding, 5% will give approximately 72 TB of usable space.

Note: You can verify the available space in the RG by using the `mm1srecoverygroup` command.

Checking the existing available space should look like the following (Example 3-6).

Example 3-6 Checking the existing available space

```
[root@fscs-fab3-2-a ~]# mmf recoverygroup RGess3k -L
```

recovery group	declustered		pdisks	current	allowable
	arrays	vdisks		format version	format version
RGess3k	1	9	24	5.0.0.0	5.0.0.0

declustered array	needs service	vdisks	pdisks	spares	replace threshold	free space	scrub duration	background activity	
								task	progress priority
DA1	no	9	24	2,14	2	76 TiB	14 days	scrub	26% low

vdisk	RAID code	declustered array	vdisk size	block size	checksum granularity	state	remarks
RG003ROOTLOGHOME	4WayReplication	DA1	4096 MiB	2 MiB	4096	ok	log
RG003LG001LOGHOME	4WayReplication	DA1	4096 MiB	2 MiB	4096	ok	log
RG003LG002LOGHOME	4WayReplication	DA1	4096 MiB	2 MiB	4096	ok	

- When you create your VDisk set, specify a name that is self-explanatory behind the **-vdisk-set** parameter. Furthermore, we need to specify the usage for the VDisks / NSDs by the parameter **--nsd-usage metadataOnly**.

The creation of a VDisk set should look like this (Example 3-7).

Example 3-7 Creation of a VDisk set

```
[root@fscs-fab3-2-a nvme_test]# mmvdisk vs define --vdisk-set set4m --recovery-group RGess3k
--code 8+2p --block-size 4M --set-size 5% --nsd-usage metadataOnly
mmvdisk: Vdisk set 'set4m' has been defined.
mmvdisk: Recovery group 'RGess3k' has been defined in vdisk set 'set4m'.
```

vdisk set	member count	vdisks size	raw size	created	file system and attributes
set4m	4	755 GiB	960 GiB	no	-, DA1, 8+2p, 4 MiB, metadataOnly, system

recovery group	declustered array	type	total capacity	raw free	free%	all vdisk sets defined in the declustered array
RGess3k	DA1	NVMe	76 TiB	72 TiB	95%	set4m

node class	available	required	vdisk set map memory per server
			required per vdisk set
ess3k	5113 MiB	5062 MiB	set4m (133 MiB)

Tip: This step should finish quickly, because only the definitions are set up.

- In the next step, as shown in Example 3-8, the VDisk gets created and subsequently the NSDs. Depending on the total size, this step might take a little while.

Example 3-8 The VDisk gets created

```
[root@fsc-2-a nvme_test]# mmvdisk vs create --vdisk-set set4m
mmvdisk: 4 vdisks and 4 NSDs will be created in vdisk set 'set4m'.
mmvdisk: (mmcrvdisk) [I] Processing vdisk RG003LG001VS002
mmvdisk: (mmcrvdisk) [I] Processing vdisk RG003LG002VS002
mmvdisk: (mmcrvdisk) [I] Processing vdisk RG003LG003VS002
mmvdisk: (mmcrvdisk) [I] Processing vdisk RG003LG004VS002
mmvdisk: Created all vdisks in vdisk set 'set4m'.
mmvdisk: (mmcrnsd) Processing disk RG003LG001VS002
mmvdisk: (mmcrnsd) Processing disk RG003LG002VS002
mmvdisk: (mmcrnsd) Processing disk RG003LG003VS002
mmvdisk: (mmcrnsd) Processing disk RG003LG004VS002
mmvdisk: Created all NSDs in vdisk set 'set4m'.
[root@fsc-2-a nvme_test]#
```

- Now we need to create the NSDs. To demonstrate the next step, check the status of existing disks in the file system, and then add the newly created NSDs. Pay attention to the columns about failure group, metadata, and pool definitions. See Example 3-9.

Example 3-9 Check the status of existing disks

```
root@fsc-2-a nvme_test]# mmlsdisk essGL2_4m -L
disk          driver  sector  failure holds  holds
storage
name          type    size    group metadata data  status      availability disk id pool
remarks
-----
-----
RG001VS001   nsd     512     1 yes     yes  ready      up          1
system      desc
RG002VS001   nsd     512     2 yes     yes  ready      up          2
system      desc
Number of quorum disks: 2
Read quorum value:     2
Write quorum value:    2
[root@fsc-2-a nvme_test]#
```

- Now, add the NSDs by using the `mmvdisk` command, as shown in Example 3-10.

Example 3-10 Adding the NSDs

```
[root@fsc-2-a nvme_test]# mmvdisk fs add --file-system essGL2_4m --vdisk-set set4m
mmvdisk: The following disks of essGL2_4m will be formatted on node ems1:
mmvdisk:   RG003LG001VS002: size 773368 MB
mmvdisk:   RG003LG002VS002: size 773368 MB
mmvdisk:   RG003LG003VS002: size 773368 MB
mmvdisk:   RG003LG004VS002: size 773368 MB
mmvdisk: Extending Allocation Map
mmvdisk: Checking Allocation Map for storage pool system
mmvdisk: 22 % complete on Sun Jan 26 10:04:55 2020
mmvdisk: 43 % complete on Sun Jan 26 10:05:00 2020
mmvdisk: 63 % complete on Sun Jan 26 10:05:05 2020
mmvdisk: 84 % complete on Sun Jan 26 10:05:10 2020
```

```
mmvdisk: 100 % complete on Sun Jan 26 10:05:14 2020
mmvdisk: Completed adding disks to file system essGL2_4m.
[root@fscc-fab3-2-a nvme_test]#
```

9. After successful completion, check the status of the file system again and see the new disks. See Example 3-11.

Example 3-11 Checking the status of the file system

```
[root@fscc-fab3-2-a nvme_test]# mmlsdisk essGL2_4m -L
disk      driver  sector  failure holds  holds  storage
name      type    size    group metadata data  status  availability disk id pool  remarks
-----
RG001VS001 nsd      512      1 yes    yes  ready  up      1 system  desc
RG002VS001 nsd      512      2 yes    yes  ready  up      2 system  desc
RG003LG001VS002 nsd      512      1 yes    no   ready  up      3 system  desc
RG003LG002VS002 nsd      512      2 yes    no   ready  up      4 system
RG003LG003VS002 nsd      512      1 yes    no   ready  up      5 system
RG003LG004VS002 nsd      512      2 yes    no   ready  up      6 system
Number of quorum disks: 3
Read quorum value: 2
Write quorum value: 2
```

3.3.3 Scenario-2: Using ESS 3000 to create a new file system

Similar to 3.3.2, “Scenario-1: Using ESS 3000 for metadata NSDs for the existing file system” on page 50, the available capacity from the new RG can also be used to create a new file system.

We will create a file system by using the `mmvdisk` command. The creation of a new file system by using the `mmvdisk` command is very convenient. In the following example, we document the commands for creating a file system with a 2 MB block size. More details about `mmvdisk` can be found in 3.3.2, “Scenario-1: Using ESS 3000 for metadata NSDs for the existing file system” on page 50. Here we just list the commands.

1. First we will define the recovery options, as shown in Example 3-12.

Example 3-12 Defining the recovery groups

```
[root@fscc-fab3-2-a ~]# mmvdisk vs define --vdisk-set set2M --recovery-group RGess3k --code 8+2p
--block-size 2M --set-size 2T
mmvdisk: Vdisk set 'set2M' has been defined.
mmvdisk: Recovery group 'RGess3k' has been defined in vdisk set 'set2M'.
```

```

          member vdisks
vdisk set  count  size  raw size  created  file system and attributes
-----
set2M      4  519 GiB  660 GiB  no      -, DA1, 8+2p, 2 MiB, dataAndMetadata, system

          declustered          capacity          all vdisk sets defined
recovery group  array  type  total raw  free raw  free%  in the declustered array
-----
RGess3k        DA1          NVMe    76 TiB   73 TiB   96%  gpfs0_system_0, set2M

          vdisk set map memory per server
node class  available  required  required per vdisk set
-----
ess3k       5113 MiB  5032 MiB  gpfs0_system_0 (2048 KiB), set2M (101 MiB)
```

```
[root@fscc-fab3-2-a ~]# mmvdisk vdiskset create --vdisk-set set2M
mmvdisk: 4 vdisks and 4 NSDs will be created in vdisk set 'set2M'.
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: Created all vdisks in vdisk set 'set2M'.
mmvdisk: (mmcrnsd) Processing disk RG003LG001VS003
mmvdisk: (mmcrnsd) Processing disk RG003LG002VS003
mmvdisk: (mmcrnsd) Processing disk RG003LG003VS003
mmvdisk: (mmcrnsd) Processing disk RG003LG004VS003
mmvdisk: Created all NSDs in vdisk set 'set2M'.
[root@fscc-fab3-2-a ~]#
```

2. In the next step, we create the file system. The `mmvdisk` command enables you to specify all known `mmcrfs` options by the `--mmcrfs` option. The command should look like this (Example 3-13).

Example 3-13 Creating the file system

```
[root@fscc-fab3-2-a ~]# mmvdisk filesystem create --file-system ess3k2M --vdisk-set set2M
--mmcrfs -S relatime -L 256M -T /gpfs/ess3k2M
mmvdisk: Creating file system 'ess3k2M'.
mmvdisk: The following disks of ess3k2M will be formatted on node ems1:
mmvdisk:   RG003LG001VS003: size 531756 MB
mmvdisk:   RG003LG002VS003: size 531756 MB
mmvdisk:   RG003LG003VS003: size 531756 MB
mmvdisk:   RG003LG004VS003: size 531756 MB
mmvdisk: Formatting file system ...
mmvdisk: Disks up to size 4.14 TB can be added to storage pool system.
mmvdisk: Creating Inode File
mmvdisk: Creating Allocation Maps
mmvdisk: Creating Log Files
mmvdisk:   53 % complete on Tue Feb 11 09:07:00 2020
mmvdisk:  100 % complete on Tue Feb 11 09:07:02 2020
mmvdisk: Clearing Inode Allocation Map
mmvdisk: Clearing Block Allocation Map
mmvdisk: Formatting Allocation Map for storage pool system
mmvdisk: Completed creation of file system /dev/ess3k2M.
[root@fscc-fab3-2-a ~]#
```

3. We now configure the remote cluster relation. The file system can now be mounted and is ready to be used in the local cluster. Following our example, we need to add the new file system to the remote cluster configuration. For detailed documentation, see https://www.ibm.com/support/knowledgecenter/STXKQY_5.0.4/com.ibm.spectrum.scale.v5r04.doc/b11adv_admmcch.htm.

Therefore, if you have a valid GPFS multi-cluster configuration established, you simply add the new file system to the configuration by granting access with the `mmauth` command, as shown in Example 3-14.

Example 3-14 The mmauth command

```
[root@ems1 ssl]# mmauth grant all -f ess3k2M
mmauth: Granting cluster eagle1.spectrum access to file system ess3k2M:
        access type rw; root credentials will not be remapped.
```

```
mmauth: Granting cluster fabcluster.mainz.de.ibm.com access to file system  
ess3k2M:
```

```
    access type rw; root credentials will not be remapped.
```

```
mmauth: Propagating the cluster configuration data to all affected nodes.
```

```
mmauth: Command successfully completed
```

```
[root@ems1 ssl]#
```



Use cases

This chapter discusses ESS 3000 use cases. It includes the following topics:

- ▶ 4.1, “Metadata and High Speed Data Tiering” on page 58
- ▶ 4.2, “Database use cases” on page 58
- ▶ 4.3, “Artificial intelligence (AI) and machine learning (ML)” on page 62
- ▶ 4.4, “Other uses cases” on page 62

4.1 Metadata and High Speed Data Tiering

Metadata generally refers to *data about data*, and in the context of IBM Spectrum Scale *metadata* refers to various on-disk data structures that are necessary to manage user data. Note that directory entries and inodes are defined as metadata, but at times the distinction between data and metadata might not be obvious.

For example, in the case of a 4 KB inode, although the inode itself might contain user data, the inode is still classified as IBM Spectrum Scale metadata and are placed in a metadata pool if data and metadata are separated. Another example is the case of directory blocks, which are classified as metadata but also contain user file and directory names.

In some use cases, performance improvements might be obtained by moving IBM Spectrum Scale metadata to a faster tier, which can be accomplished by placing faster Elastic Storage System 3000 storage in its own storage pool. See 3.3.2, “Scenario-1: Using ESS 3000 for metadata NSDs for the existing file system” on page 50 for details about how to implement this technique.

This approach to metadata tiering can be adopted when trying to optimize the performance of metadata operations, such as listing directories and making `stat()` calls on files. See IBM Knowledge Center for IBM Spectrum Scale documentation on User Storage Pools (https://www.ibm.com/support/knowledgecenter/en/STXKQY_5.0.4/com.ibm.spectrum.scale.v5r04.doc/b1ladv_userpool.htm) for more details.

Another alternative tiering approach involves, instead of tiering data on the basis of a data/metadata classification, using the IBM Spectrum Scale File Heat function (https://www.ibm.com/support/knowledgecenter/en/STXKQY_5.0.4/com.ibm.spectrum.scale.v5r04.doc/b1ladv_fileheat.htm) to migrate data between storage pools based on how frequently data is accessed. For more details on this approach see the following sections in the IBM Knowledge Center documentation for IBM Spectrum Scale:

- ▶ File Heat: Tracking File Access Temperature:
https://www.ibm.com/support/knowledgecenter/en/STXKQY_5.0.4/com.ibm.spectrum.scale.v5r04.doc/b1ladm_enablingtheobjectheatmappolicy.htm
- ▶ Object Heatmap Tiering:
https://www.ibm.com/support/knowledgecenter/en/STXKQY_5.0.4/com.ibm.spectrum.scale.v5r04.doc/b1lins_objectheatmapdatatiering.htm
- ▶ Enabling the Object Heatmap Policy:
https://www.ibm.com/support/knowledgecenter/en/STXKQY_5.0.4/com.ibm.spectrum.scale.v5r04.doc/b1ladm_enablingtheobjectheatmappolicy.htm

4.2 Database use cases

For a very long time, IBM Spectrum Scale, formerly IBM General Parallel File System (IBM GPFS), is successfully used as a storage solution for scale-out databases. Oracle RAC and IBM DB2® pureScale® are some well-known examples. The powerful capabilities and flexibility of SpectroScale opened up further successful use cases in the database business in the last few years. SAP scale-out solutions are running on GPFS as successfully as large HANA DB scale-up environments.

With the IBM Elastic Storage Server based on IBM Spectrum Scale, RAID capabilities are added to the file system. By using the intelligent internal logic of the IBM Spectrum Scale RAID code, reasonable performance and significant disk failure recovery improvements are achieved.

This section provides an update about new code enhancements and simplification in deployment and configuring for Spectroscopy with ESS and SAP HANA. As opposed to other storage solutions in IBM, ESS is the most flexible and powerful solution to run SAP HANA applications on IBM hardware.

4.2.1 IBM Spectrum Scale for SAP HANA

The updates and improvements in Spectroscopy for running SAP HANA business were engineered and developed in a close partnership with SAP and Bosch.

Empowering I/O capabilities in a SAP HANA environment by IBM ESS has become very easy. The first prototype solution with GPFS and SAP HANA needed a lot of additional manual configuration steps to get optimized performance. With all newer ESS releases (the current ESS models and ESS Release 5.3.5 or higher and SAP HANA 2.0), the installation works perfectly fine out of the default deployment.

The main advantages of using IBM Spectrum Scale/ESS and SAP HANA are the ease of administration and the solution's very high performance. When the file systems are provided, you can easily deploy and add HANA servers to the server farm. They only need to be connected to the network.

The basic idea to provide a shared file system for multiple HANA instances adds a lot of flexibility and make it easier to maintain free space and utilize Storage resources. IBM Spectrum Scale with ESS/GNR adds data protection and high performance all at once. You can read about further benefits in the Redbooks publication *SAP HANA and ESS: A Winning Combination*, REDP-5436.

See how simple it is and follow the next steps to deploy an ESS 3000 and provide a file system to run SAP HANA workloads.

4.2.2 Deploy ESS

Simply follow the instructions in Chapter 3, "Planning considerations" on page 43 to deploy your ESS environment. When the ESS is set up correctly, you can proceed with the next step.

4.2.3 Creating a file system

We have to create at least three file systems as shown, in Example 4-1 on page 60. One is for storing the log data, a second is for the real data, and a third is for storing so-called *shared data*. Later, you can run multiple SAP HANA instances in those sets of file systems.

Tip: As a rule of thumb, no more than 16 production HANA databases should be located on one ESS building block. For test and QA, there are use cases with more than 50 HANA instances per building block.

Example 4-1 Creating file systems

```
mmvdisk vs define --vdisk-set PRODdata --recovery-group ESS3k --code 8+2p --block-size 1M
--set-size 150t
mmvdisk vs define --vdisk-set PRODlog --recovery-group ESS3k --code 8+2p --block-size 1M
--set-size 150t
mmvdisk vs define --vdisk-set PRODshared --recovery-group ESS3k --code 8+2p --block-size 1M
--set-size 150t

mmvdisk vs create --vdisk-set PRODlog,PRODdata, PRODshared

mmvdisk fs create --file-system PRODlog --vdisk-set PRODlog --mmcrfs -T /gpfs/PROD/log -S
relatime -E no
mmvdisk fs create --file-system PRODdata --vdisk-set PRODdata --mmcrfs -T /gpfs/PROD/data -S
relatime -E no
mmvdisk fs create --file-system PRODshared --vdisk-set PRODshared --mmcrfs -T /gpfs/PROD/shared
-S relatime -E no
```

According to your own preferences you might want to auto mount the file systems (default). In case you don't want to auto mount them, specify **-A no** in your **mmvdisk fs create** command line.

4.2.4 Preparing your clients (HANA nodes)

Perform the following steps to prepare your clients:

1. Install IBM Spectrum Scale on your HANA nodes and add them to the cluster. An alternative way to connect to the ESS storage is configuring a remote mount relation. See https://www.ibm.com/support/knowledgecenter/STXKQY_5.0.4/com.ibm.spectrum.scale.v5r04.doc/b11adv_admmcch.htm.

Although the ESS needs no further settings and manual tuning, the client side needs to be tuned with several standard parameters, depending on the available memory or network type for running database workloads.

2. You can create a node class and apply the following settings to the node class, as described in the next step. Note that this step is optional. You can also assign the HANA settings on a per-node basis later:

```
mmcrnodeclass hananode -N saphana1,saphana2
```

3. In the next step, apply the needed changes to the node class or your clients:

```
mmchconfig
dioSmallSeqWriteBatching=yes,dioSmallSeqWriteThreshold=4194304,ignorePrefetchLUNCount=yes,pagepool=24G,prefetchPct=60,workerThreads=1024 -N hananode
```

4. IBM Spectrum Scale can run in an Ethernet network. We suggest that you use at least 40 GbE network speeds. In terms of latency and bandwidth, it is highly recommended to use 100 GbE networks. A further additional optimization to scale beyond Ethernet is the use of an InfiniBand network architecture. Running on InfiniBand has a lower latency and scales linearly over the amount of ports per node. In case you have an InfiniBand network, you need to enable it in the Spectroscope configuration:

```
mmchconfig verbsPorts="mlx5_0/1 mlx5_1/1",verbsRdma=yes,verbsRdmaSend=yes -N hananodes
```

5. The IBM Spectrum Scale file systems are automatically available on all members in the cluster. Adding a HANA node means adding it to the cluster.

See the `mmaddnode` and `mm1scluster` commands at https://www.ibm.com/support/knowledgecenter/STXKQY_5.0.4/com.ibm.spectrum.scale.v5r04.doc/b11adm_command.htm.

- In case you use nodeclasses, do not forget to adjust the nodeclass accordingly using the following command: `mmchnodeclass ClassName add -N newNextHanaNode`.

4.2.5 Using the file systems

SAP expects its file systems on a dedicated specific location. A best practice is to use a data structure, as shown in Example 4-2.

Example 4-2 File systems on a dedicated specific location

```
/hana/log/$SID
/hana/data/$SID
/hana/shared/$SID
```

Adding more HANA instances is simply creating a subdirectory in each of the file systems with the appropriate SID directory name. So, to provide a shared file system for multiple HANA instances, you need to mount them as bind mounts to make sure that each HANA node sees only its own name space to run the application.

For the SAP System ID PT9, the mount should be done as shown in Example 4-3.

Example 4-3 Mounting file systems

```
cat /etc/fstab
[...]
/gpfs/PROD/data/PT9      /hana/data/PT9          none   bind,noauto   0 0
/gpfs/PROD/log/PT9      /hana/log/PT9           none   bind,noauto   0 0
/gpfs/PROD/shared/PT9   /hana/shared/PT9       none   bind,noauto   0 0
/gpfs/solagent/rb3h0723/ /agent                  none   bind,noauto   0 0
BACKUP                  /gpfs/BACKUP           gpfs
rw,mtime,relatime,quota=userquota;groupquota;filesetquota,dev=BACKUP,noauto 0 0
PRODData                /gpfs/PROD/data       gpfs
rw,mtime,relatime,quota=userquota;groupquota;filesetquota;perfileset,dev=PRODData,noauto 0 0
PRODlog                 /gpfs/PROD/log        gpfs
rw,mtime,relatime,quota=userquota;groupquota;filesetquota;perfileset,dev=PRODlog,noauto 0 0
PRODshared              /gpfs/PROD/shared     gpfs   rw,mtime,relatime,quota=userquota;groupquota;filesetquota,
```

As you can see, if there is enough free space left, new SAP instances can be added simply with the `mkdir` command. The `/etc/fstab` from the next HANA server (PQ9) would then look like that shown in Example 4-4.

Example 4-4 The /etc/fstab

```
/gpfs/PROD/data/PT9      /hana/data/PQ9          none   bind,noauto   0 0
/gpfs/PROD/log/PT9      /hana/log/PQ9           none   bind,noauto   0 0
/gpfs/PROD/shared/PT9   /hana/shared/PQ9       none   bind,noauto   0 0
/gpfs/solagent/rb3h0723/ /agent                  none   bind,noauto   0 0
[...]
PRODData                /gpfs/PROD/data       gpfs
rw,mtime,relatime,quota=userquota;groupquota;filesetquota;perfileset,dev=PRODData,noauto 0 0
PRODlog                 /gpfs/PROD/log        gpfs
rw,mtime,relatime,quota=userquota;groupquota;filesetquota;perfileset,dev=PRODlog,noauto 0 0
PRODshared              /gpfs/PROD/shared     gpfs
rw,mtime,relatime,quota=userquota;groupquota;filesetquota,dev=PRODshared,noauto 0 0
```

From the perspective of providing the storage to the HANA nodes, the node is now ready for the application.

4.3 Artificial intelligence (AI) and machine learning (ML)

NVIDIA and IBM have created a reference architecture for NVIDIA DGX and ESS 3000 working together on AI and ML workloads. The reference architecture can be found on the following link: <https://www.ibm.com/downloads/cas/MNEQQQVP>. Together NVIDIA and IBM provide an integrated, individually scalable compute and storage solution with end-to-end parallel throughput from flash to GPU for accelerated DL training and inference.

This section is intended for enterprise leaders, solution architects, and other readers interested in learning how the IBM Spectrum Storage™ for AI with NVIDIA DGX systems simplifies and accelerates AI. The scalable infrastructure solution integrates the NVIDIA DGX-1 systems and NVIDIA DGX-2 systems with IBM Spectrum Scale file storage software, which powers the IBM Elastic Storage Server (ESS) family of storage systems that includes the new IBM Elastic Storage System (ESS 3000).

The reference architecture covers the linear growth of the AI or ML system from both the GPU workloads on the NVIDIA DGX systems. It also demonstrates the linear growth capabilities of 40 GBps per ESS 3000 unit for read random workloads.

In the market for AI and ML workloads, systems other than NVIDIA DGX are available, and all of those can benefit from the outstanding performance capabilities of the ESS 3000 system and the IBM Power systems with GPUs, such as the AC922 server.

4.4 Other uses cases

ESS 3000 runs IBM Spectrum Scale as its file system, so some use cases and planning that apply to other members of the IBM Spectrum Scale family also apply for ESS 3000.

4.4.1 IBM Spectrum Scale with big data and analytics solutions

IBM Spectrum Scale is flexible and scalable software-defined file storage for analytics workloads. Enterprises around the globe deploy IBM Spectrum Scale to form large data lakes and content repositories to perform high-performance computing (HPC) and analytics workloads. It is known to scale performance and capacity without bottlenecks.

Hortonworks Data Platform (HDP) is a leader in Hadoop and Spark distributions. HDP addresses the needs of data-at-rest, powers real-time customer applications, and delivers robust analytics that accelerate decision making and innovation. IBM Spectrum Scale solves the challenge of explosive growth of unstructured data against a flat IT budget. IBM Spectrum Scale provides unified file and object software-defined storage for high-performance, large-scale workloads, and it can be deployed on-premises or in the cloud.

IBM Spectrum Scale is POSIX compatible, so it supports various applications and workloads. By using IBM Spectrum Scale HDFS Transparency Hadoop connector, you can analyze file and object data in place, with no data transfer or data movement. Traditional systems and analytics systems use and share data that is hosted on IBM Spectrum Scale file systems.

Hadoop and Spark services can use a storage system to save IT costs because no special-purpose storage is required to perform the analytics. IBM Spectrum Scale features a rich set of enterprise-level data management and protection features. These features include snapshots, information lifecycle management (ILM), compression, and encryption, all of which provide more value than traditional analytic systems do. For more information, see the following document: <http://www.redbooks.ibm.com/abstracts/redp5397.html?Open>.

4.4.2 Genomics Medicine workloads in IBM Spectrum Scale

IT administrators, physicians, data scientists, researchers, bioinformaticians, and other professionals who are involved in the genomics workflow need the right foundation to achieve their research objectives efficiently. At the same time, they want to improve patient care and outcomes. Thus, it is important to understand the different stages of the genomics workload and the key characteristics of it.

Advanced genomics medicine customers are outgrowing network-attached storage (NAS). The move from a traditional NAS system or a modern scale-out NAS system to a parallel file system like IBM Spectrum Scale requires a new set of skills. Thus, the IBM Spectrum Scale Blueprint for Genomics Medicine Workloads must provide basic background information. It must also offer optional professional services to help customers successfully transition to the new infrastructure.

For more information, see the following document:
<http://www.redbooks.ibm.com/abstracts/redp5479.html>

4.4.3 IBM Cloud Object Store

You can combine IBM Cloud™ Object Storage with IBM Spectrum Scale through the Transparent Cloud Tier (TCT). When you use TCT, Cloud Object Store can be used as a colder tier, it can be an on-site Cloud Object Store, or a remote Cloud Object Store.

For more information, see the following web page:

<https://www.ibm.com/common/ssi/cgi-bin/ssialias?htmlfid=WUS12361USEN>

Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this book.

IBM Redbooks

The following IBM Redbooks publications provide additional information about the topic in this document. Note that some publications referenced in this list might be available in softcopy only.

- ▶ *Monitoring and Managing the IBM Elastic Storage Server Using the GUI*, REDP5471
- ▶ *SAP HANA and ESS: A Winning Combination*, REDP-5436
- ▶ *Introduction Guide to the IBM Elastic Storage Server*, REDP5253

You can search for, view, download or order these documents and other Redbooks, Redpapers, Web Docs, draft and additional materials, at the following website:

ibm.com/redbooks

Online resources

These websites are also relevant as further information sources:

- ▶ IBM ESS 3000 IBM Knowledge Center:
https://www.ibm.com/support/knowledgecenter/en/SSZL24_6.0.0/ess3000_600_welcome.html
- ▶ IBM Spectrum Scale V 5.0.4 Planning Considerations:
https://www.ibm.com/support/knowledgecenter/en/STXKQY_5.0.4/com.ibm.spectrum.scale.v5r04.doc/b11in_PlanningForIBMSpectrumScale.htm
- ▶ Licensing on IBM Spectrum Scale
https://www.ibm.com/support/knowledgecenter/en/STXKQY_5.0.4/com.ibm.spectrum.scale.v5r04.doc/b11ins_capacitylicense.htm
- ▶ Using IBM Cloud Object Storage with IBM Spectrum Scale:
<https://www-01.ibm.com/common/ssi/cgi-bin/ssialias?htmlfid=WUS12361USEN>
- ▶ mmvdisk command reference:
https://www.ibm.com/support/knowledgecenter/en/SSYSP8_5.3.5/com.ibm.spectrum.scale.raid.v5r04.adm.doc/b18adm_mmvdisk.htm

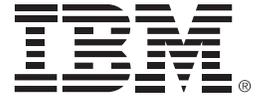
Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services



SG24-8443-00

ISBN 0738458635

Printed in U.S.A.

Get connected

