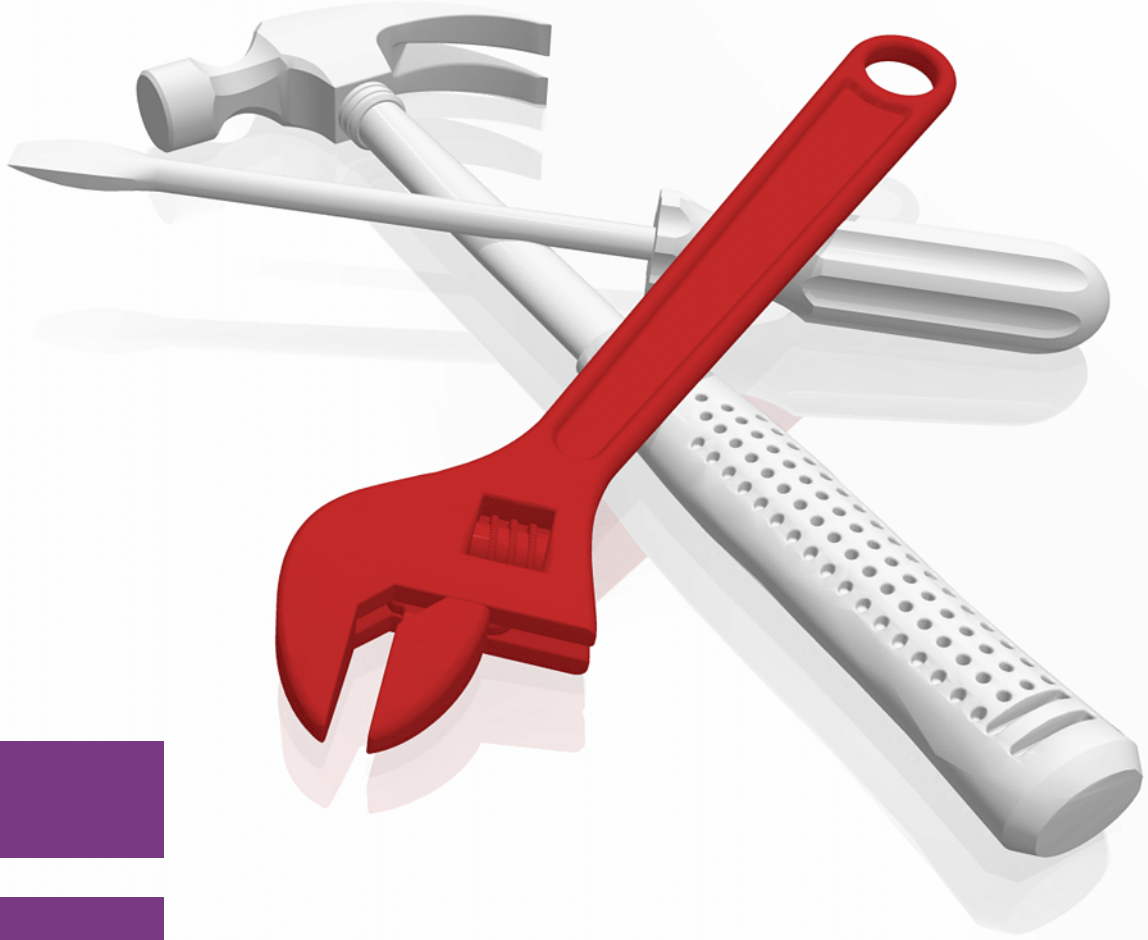
Redbooks
ibm.com/redbooks

IBM FlashSystem with IBM Storage Scale Solution:
Block Storage for High-Performance File Access

Rajsekhar Bharali
Abhishek Jain
Kedar Karmarkar
Chris Maestas
Rajan Mishra
Muthu Muthiah
Sarvesh Patel
Vijay Patidar
Mangesh Shirke



Storage

Systems Software

IBM

Redpaper

IBM Redbooks

IBM FlashSystem with IBM Storage Scale Solution: Block Storage for High-Performance File Access

December 2025

REDP-5764-00

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

First Edition (December 2025)

Contents

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 https://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.

IBM®	IBM FlashSystem®	Redbooks (logo)  ®
IBM Consulting®	IBM Research®	Storwize®
IBM FlashCore®	Redbooks®	

The following terms are trademarks of other companies:

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.

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

Red Hat, Ansible, OpenShift 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 describes a high performance file and block access solution using IBM FlashSystem® and IBM Storage Scale to provide file access interfaces suited for diverse workloads, which includes traditional, cloud-era and emerging, such as SAP, web applications,and genome analysis. It offers a step-by-step guide for implementation engineers to integrate IBM FlashSystem and legacy IBM Storwize® for block storage, paired with IBM Storage Scale for Network File System (NFS) file access and to use IBM Storage Insights for Ransomware Threat Detection.

This solution is tailored for small and medium businesses with small storage capacities and targeted for IBM FlashSystem customers who primarily need block storage with file access. We plan to promote this solution primarily through IBM business partners, providing customers with an integrated end-to-end solution. It is not intended for NAS, Citrix, or home directory use cases.

Authors

This publication was produced by a team of specialists from around the world working.

Rajsekhar Bharali Rajsekhar Bharali leads testing for IBM FlashSystem, focusing on Storage Partition Mobility and FlashSystem Grid, with 14+ years in the storage domain.

Abhishek Jain is a Master Inventor and Quality Architect for IBM Storage Scale, driving quality in container-native storage for OpenShift and Kubernetes. With 75+ US patents and multiple publications, he advocates hybrid cloud solutions.

Kedar Karmarkar is a Development Architect with IBM Storage Scale, bringing 25+ years of experience in storage, virtualization, and infrastructure software. He has led innovations across NAS, replication, and management solutions.

Chris Maestas is a CTO for Data and AI Storage Solutions at IBM. He is pioneer in the storage technology.

Rajan Mishra is a Software Engineer with IBM Storage Scale, leading deployment roadmap development and bringing 9.8+ years of experience. He specializes in Python, Golang, Java, Ansible, and Linux.

Muthu Muthiah is a Development Architect for IBM® Storage Scale, specializing in GUI-based management and deployment across on-prem, cloud, and system platforms.

Sarvesh Patel is an Architect for IBM FlashSystem, focusing on Quality, Automation, and CCU utility development. A Master Inventor with 200+ US patents, he leads multiple invention teams at IBM.

Vijay Patidar is an Advisory Software Engineer and Technical Lead for IBM Storage Insight, with 8+ years of experience in optimizing data storage solutions. He specializes in performance, reliability, and actionable insights.

Mangesh Shirke is part of IBM India Systems Development Lab and works on the IBM Storage Defender team with deep expertise in enterprise storage solutions.

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

- Phillip Gerrard, Project Leader, IBM Redbooks Storage Team
- Alistair Symon, Vice President of Development, IBM Storage
- Micah Robison, Vice President of Enterprise Storage Development
- Sandeep Patil, Distinguished Engineer, CTO - ISDL (Storage), IBM Master Inventor
- Brian Sherman, Distinguished Engineer, Storage ATG
- Chris Maestas, CTO, Data and AI Storage Solutions
- Vincent Hsu, Vice President, IBM Fellow, CTO for IBM Storage
- Mara Miranda, IBM Systems, Storage Scale
- Diana Laura Silva, System Test Engineer, IBM FlashSystem
- J Lakshmi Yoganand, Automation Test Engg. IBM Storage Insights
- Shiness George, Performance Test Engg. IBM FlashSystem
- Julian Hill, Performance Arch. IBM FlashSystem
- Luiz Gabriel Costa, Product Manager, IBM FlashSystem Ecosystem
- Gabriel Rodriguez, Product Manager. IBM FlashSystem Ecosystem
- Jim Comstock, Executive Director, Product Management
- Mamdouh Khamis, Product Manager, IBM Storage Scale System
- Tom O'Brien, Program Director, Product Management at IBM
- Olaf Weiser, HPC/AI Storage Specialist at IBM
- Kumar Gaurav, Ethernet dev lead, IBM FlashSystem
- Subhojit Roy, STSM, IBM FlashSystem

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 publications to be as helpful as possible. Send us your comments about this publication 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 LinkedIn:

  https://www.linkedin.com/groups/2130806

- Explore new Redbooks publications, residencies, and workshops with the IBM Redbooks weekly newsletter:

  https://www.redbooks.ibm.com/subscribe

- Stay current on recent Redbooks publications with RSS Feeds:

  https://www.redbooks.ibm.com/rss.html

IBM FlashSystem with IBM Storage Scale solution: Block storage for high-performance file access

Overview

In computing, a file system is used to control how data is stored and retrieved. Without a file system, information placed in a storage area would be one large body of data with no way to know where one piece of information stops and the next begins. IBM Storage Scale (formerly IBM GPFS) is the distributed file system that is being used by some of the supercomputers and many of the largest commercial companies in the world. IBM Storage Scale provides high-speed file access to applications running on multiple nodes of clusters, with concurrent access to a single and multiple file systems. Cluster nodes can be storage area network (SAN)-attached, network-attached, or a mix of SAN- and network-attached.

IBM SAN-attached storage virtualization products, which include IBM FlashSystem and earlier IBM Spectrum Virtualize and IBM Storwize products, offer dependable data availability up to 99.999% and can handle the massive amount of block-storage data packed with software features such as data tiering, real-time compression, thin provisioning, safeguarded copy, IBM FlashCore® Modules (FCMs), policy-based replication, policy-based high availability, FlashSystem grid, and so on.

This publication describes a high-performance file and block access solution that uses IBM FlashSystem along with IBM Storage Scale to deliver file interface for any type of workload such as traditional workload (SAP, ERP, database), Cloud Era workloads (web content, web apps, social media) or new workload (Genome analysis, IoT). This step-by-step guide helps an implementation engineer to easily install, configure, and integrate IBM Storage FlashSystem and older IBM Storwize products to provide block storage and IBM Storage Scale to create a file system to provide NFS and SMB file access interface. You can also configure IBM Storage Insights to monitor and provide diagnostic capability for ease of management.

Figure 1 on page 2 represents the high-level architecture of high-performance file and block access solution.

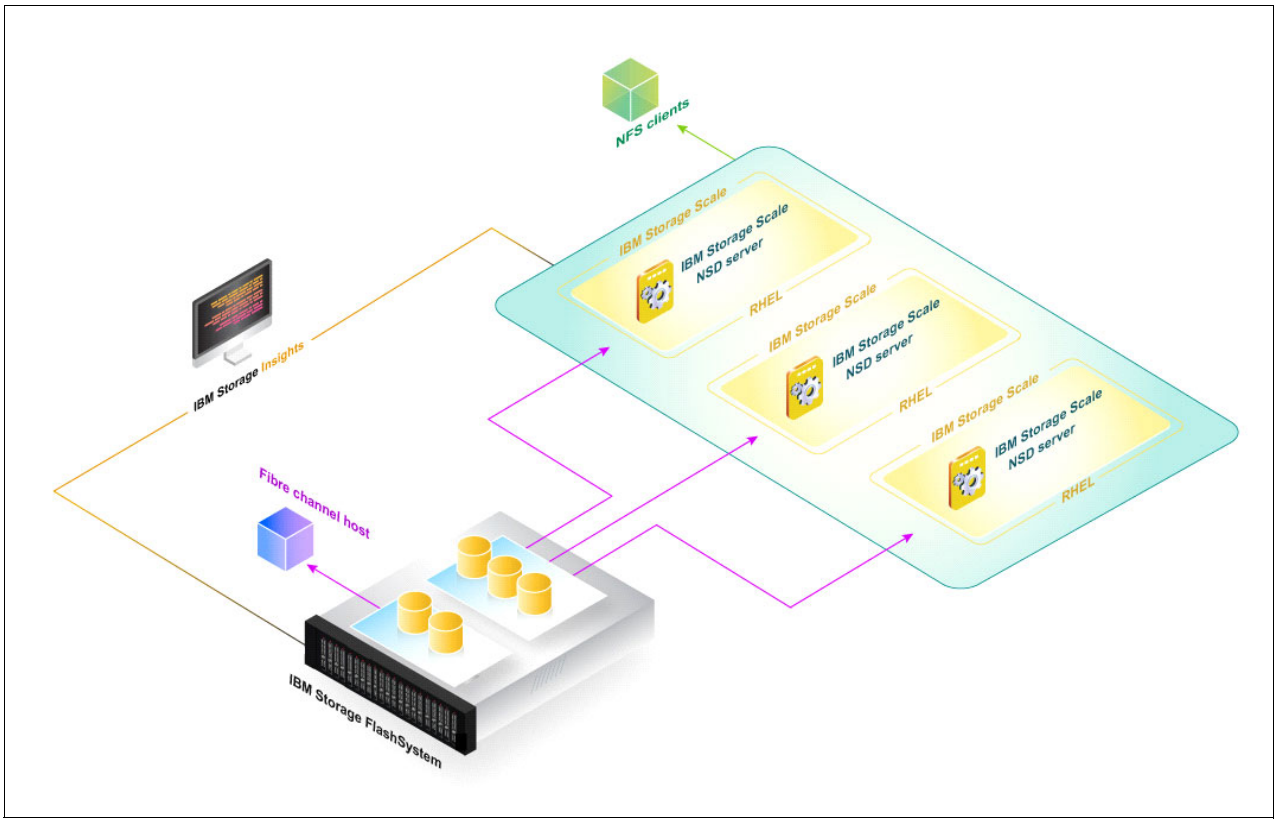


Figure 1 High-level, high-performance file and block access solution architecture

Figure 2 illustrates a more detailed view of the solution architecture.

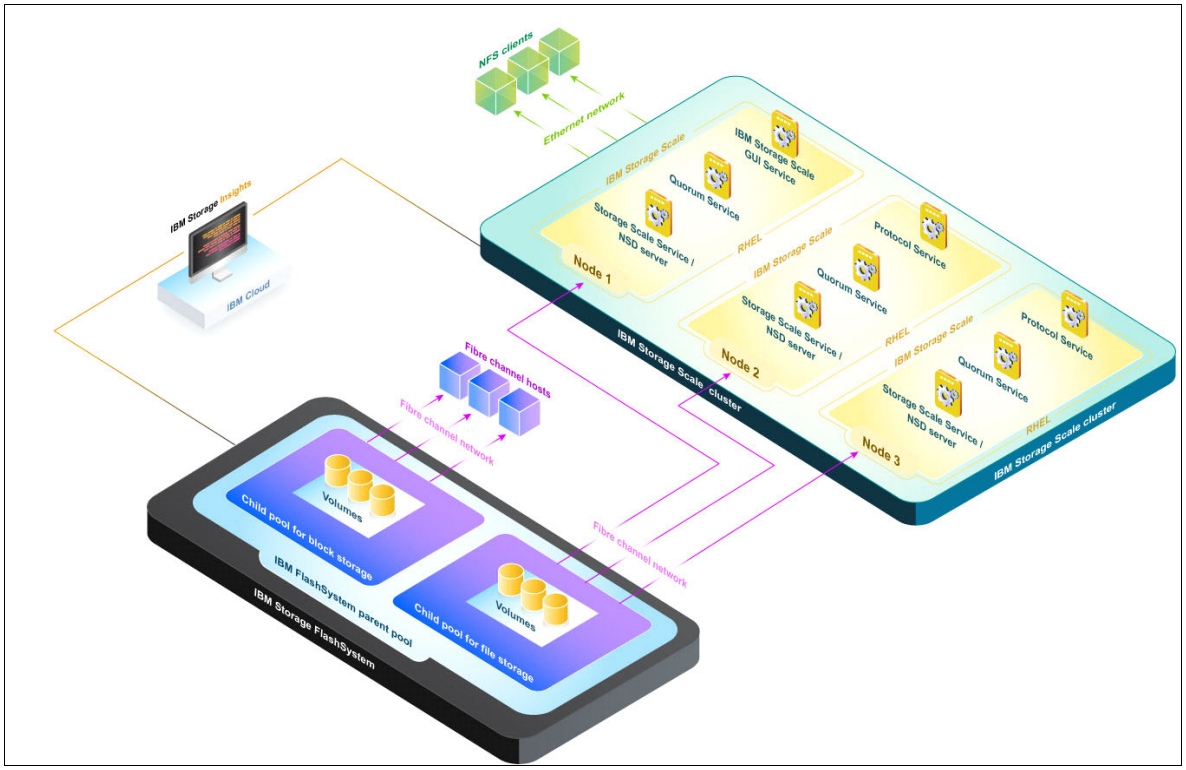


Figure 2 Detailed view of high-performance file and block access solution architecture

This publication includes the following aspects of block storage and high-performance file access with IBM Storage Scale:

- Use cases:
  - Understand where to use high-performance file and block access solution
  - Understand where not to use this solution
- Sizing and limitation:
  - System sizing (CPU, memory, and network)
  - Limitation on the number of concurrent NFS users
  - Limited user authentication and authorization capabilities
  - Local authentication and LDAP
- Total capacity and capacity split planning, which includes planning how to translate customer capacity requirements (block and file) to create volumes in IBM FlashSystem.
- Instructions for deployment:
  - IBM FlashSystem
  - IBM Storage Scale
  - Configuration of IBM FlashSystem and IBM Storage Scale
  - Best practice configuration based on previously defined sizing

Sizing and limitations

Because this solution uses IBM Storage Scale and IBM FlashSystem, it is important to size the required resources correctly. This section describes the minimum resource requirement for this solution.

IBM Storage Scale node sizing

The sizing details that are mentioned in Table 1 are for running the operating system, the Storage Scale services, and the protocol services only. Consider adding resources for running other software.

Table 1 IBM Storage Scale node hardware requirements

Component	Sizing	Remarks or comments
CPU	x86 64-bit processor with 12 processor cores or more	NA
Memory	64 GB or more	NA
Network	Two 10 Gbps ports	Configure Ethernet ports in LACP bonding
FC ports	Two 16 Gb or higher Fibre Channel (FC) ports	NA

Note: The sizing details that are mentioned in Table 1 are for only *one* Storage Scale node. You need three such nodes to complete this configuration.

IBM FlashSystem storage sizing

See Table 2 for resource requirements in IBM FlashSystem storage.

Table 2 IBM FlashSystem storage resource requirements

Component	Sizing	Remarks or comments
Disk capacity	From the file system requirement	NA
FC ports	Two FC ports per storage nodes/controller, shared or dedicated (dedicated preferred)	NA

Management, administration, and observability

This solution involves IBM Storage Scale and IBM FlashSystem storage. Each has its own graphical user interface (GUI) and command line interface (CLI) console for management.

For better visibility and a consolidated observability view, you can also use IBM Storage Insights along with the solution.

Options for accessing Storage Scale and FlashSystem

See Example 1 to check the options available to log in and manage these products.

Example 1 Accessibility options of storage components

Product name	GUI	CLI	REST API
IBM Storage Scale	Yes	Yes	Yes
IBM FlashSystem	Yes	Yes	Yes
IBM Storage Insights	Yes	NA	Yes

You can access the GUI for all three products directly by using a web browser on the HTTPS connection. Additional software is not required to access the GUI.

The CLI for IBM Storage Scale and IBM FlashSystem can be accessed directly by using a Secure Shell (SSH) connection. Additional software is not required for accessing the CLI.

IBM Storage Insights

IBM Storage Insights is an IBM offering that is designed to provide intelligent storage management and monitoring capabilities for storage environments. It is a cloud-based storage management platform that helps organizations optimize their storage infrastructure and streamline storage management tasks. IBM Storage Insights provides an unparalleled level of visibility across your storage environment to help you manage complex storage infrastructures and make cost-saving decisions. At the block level, it provides the following features:

- Performance monitoring: Real-time performance monitoring and reporting for block storage volumes, allowing administrators to track I/O performance and identify performance bottlenecks.

- Capacity management: Capacity forecasting and reporting for block storage volumes to help organizations plan for future storage needs and optimize capacity usage.
- Alerting and notifications: Proactive alerts and notifications for critical events and potential issues in block storage systems.
- Ransomware Threat Detection (RTD) when using IBM FlashSystem and FCM4 drives

IBM Storage Insights includes a range of features for file storage monitoring:

- Capacity usage: The platform provides insights on capacity usage of your file storage systems, file shares, and file systems. You can monitor the usage trends and plan for future storage needs.
- Performance monitoring: Storage Insights tracks the performance metrics of your file storage, such as input/output operations per second (IOPS), throughput, and latency. It helps you identify performance bottlenecks and optimize file storage performance.
- Health and status monitoring: The platform offers a centralized dashboard that displays the health and status of your file storage systems. You can quickly check the overall health and any potential issues.
- File and file share analytics: Storage Insights collects data on file-level and file share-level analytics, sou you can nderstand file access patterns, file sizes, and file share usage.

In the high-performance file and block-access solution, block storage volume and file storage monitoring can be a critical requirement for your organization. Therefore, consider using IBM Storage Insights to determine the best solution for your specific needs. Application-specific insights and monitoring are typically provided by application monitoring and management tools that are designed to capture application-specific metrics and performance data. By using those tools, you can see a unified view of block and file storage into a single platform. Figure 3 shows a unified monitoring view for IBM FlashSystem and IBM Storage Scale.

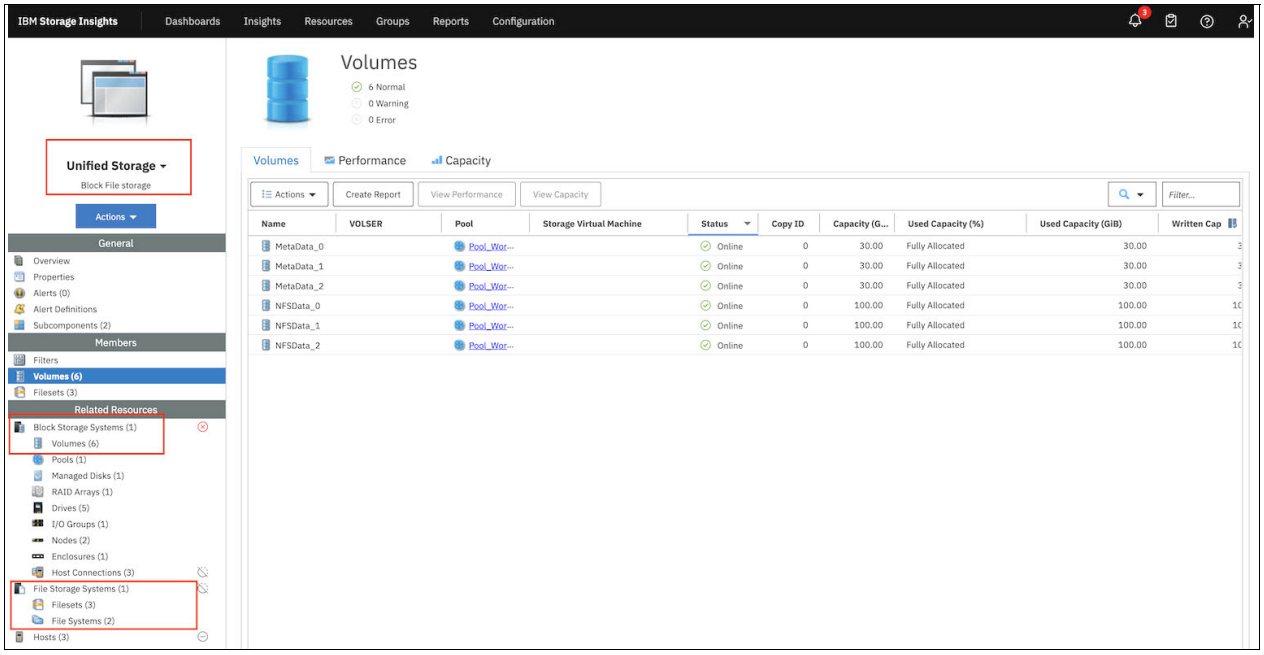


Figure 3 Unified monitoring view

Ransomware threat detection by using IBM Storage Insights Pro

You can use IBM Storage Insights Pro with IBM FlashSystem FCM4 drives on an IBM Storage Scale file system to help with ransomware threat detection.

Ransomware detection features

IBM FlashSystem incorporates advanced ransomware detection mechanisms that work in tandem with Storage Insights Pro to identify and alert on suspicious activities. These capabilities are designed to detect behavioral anomalies indicative of ransomware attacks, which include the following anomalies:

- Sudden spikes in file modifications or deletions.
- Unusual access patterns across volumes.
- Rapid encryption of large datasets.

When integrated with IBM Storage Insights Pro, IBM FlashSystem provides the following features:

- Generate Real-Time Alerts: When a ransomware-like pattern is detected, an alert is triggered in Storage Insights Pro.
- Provide Forensic Insights: Details about affected volumes, timestamps, and access patterns are logged for investigation.
- Enable Rapid Response: Administrators can isolate affected volumes, initiate snapshots, or trigger recovery workflows.

Ransomware threat detection test with WannaLaugh on IBM Storage Scale

In a recent test scenario, the ransomware detection capabilities were evaluated by using a simulated ransomware tool named WannaLaugh. WannaLaugh is a configurable ransomware emulator developed by the IBM Research® team. This tool is designed to safely mimic ransomware attacks without causing actual harm or spreading malware, making it a unique solution for studying ransomware behavior. For more information, see WannaLaugh: A Configurable Ransomware Emulator.

The environment consisted of IBM FlashSystem volumes that are mapped to an IBM Storage Scale file system with the following key observations:

- Detection time frame: The ransomware attack was successfully detected within 5–6 minutes of initiation. This rapid detection is critical for minimizing data loss and initiating timely remediation. On IBM FlashSystem, the threat detection is within 60 seconds, but because there are multiple layers of caching as well as RAIDs, detection might take more for this particular solution.
- Alert generation: Storage Insights Pro generated a security alert upon detecting the attack, confirming the effectiveness of the integrated threat detection mechanism.
- Volume-level detection behavior:
  - In environments where multiple volumes are created and mapped to IBM Storage Scale, data is distributed across these volumes.
  - The ransomware detection system might not alert on all volumes simultaneously. Instead, it identifies and alerts on some volumes where the attack pattern is most prominent.

– This behavior is because of the distributed nature of data and the detection algorithm's focus on anomalous activity within individual volumes.

Important Consideration: If any volume within the Storage Scale pool triggers a ransomware alert, treat the alert as a system-wide threat. The distributed architecture implies that the attack might be affecting multiple volumes, even if not all are flagged immediately. Therefore, a comprehensive response is recommended, which includes isolating the entire pool, initiating snapshots, and conducting forensic analysis.

Provisioning space to create a network file system

Users need to provision space from IBM FlashSystem storage to create a network file system. Use the following high-level steps:

1. Create a storage pool
2. Create a volume
3. Create a host
4. Map the volume to the host
5. Configure multipathing on the host
6. Scan the volume on the host

Note: Step 1 to step 4 can be performed by using the IBM FlashSystem GUI or the CLI. This publication provides steps when using both GUI and CLI.

Provision space to create a network file system using GUI

Perform the following steps to provision space from IBM FlashSystem storage.

Creating a storage pool

1. Create a parent storage pool from the IBM FlashSystem GUI by selecting **Pools** → **Create** → **Create pool**. (Figure 4)

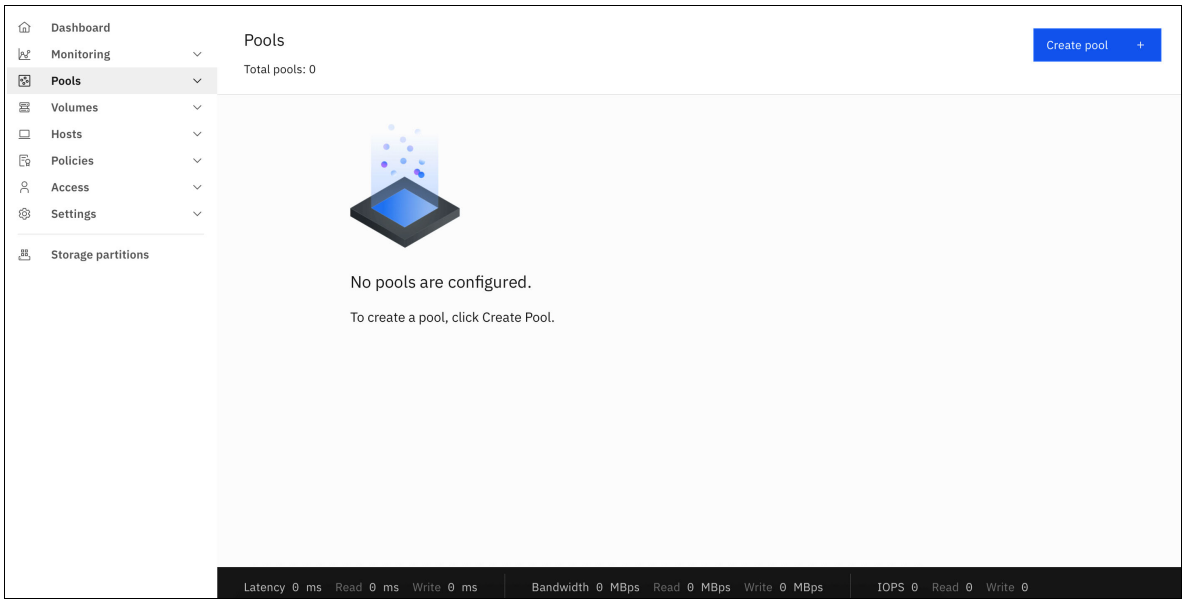


Figure 4 Create a storage pool

2. In the Create Pool dialog (Figure 5), enter the specified details in the following fields and click **Create**.
   a. **Pool Name:** Enter a user-defined name for the pool
   b. Data Reduction: Select this checkbox if you want to use deduplication or else clear it
   c. Provisioning Policy: Select the provisioning policy that the user wants to use with this pool

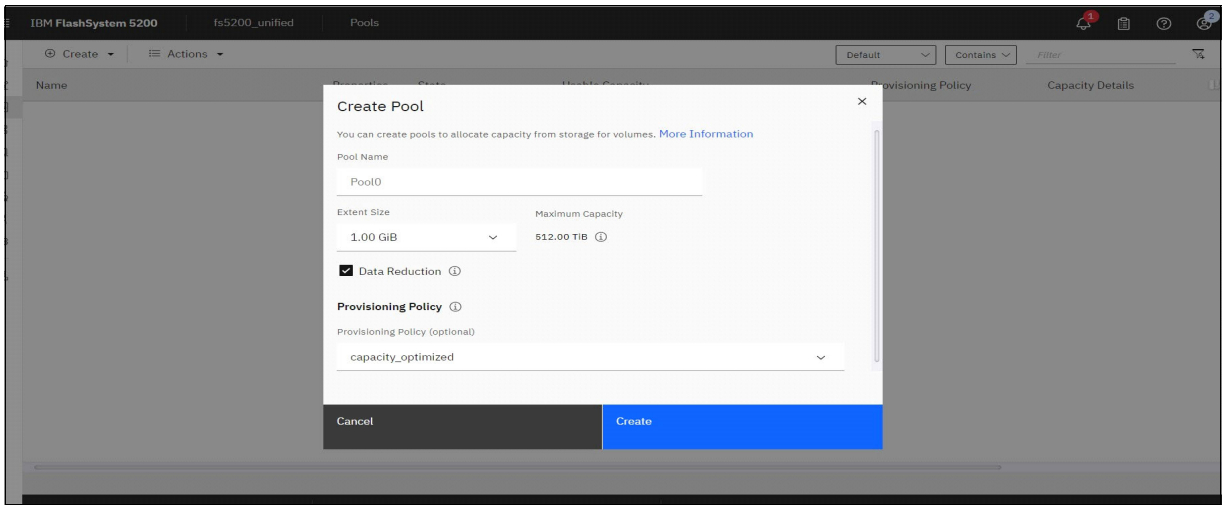


Figure 5 Create Pool dialog

3. Click **Pools** and select the storage pool.
4. Select **Actions** → **Add Storage**.(Figure 6)

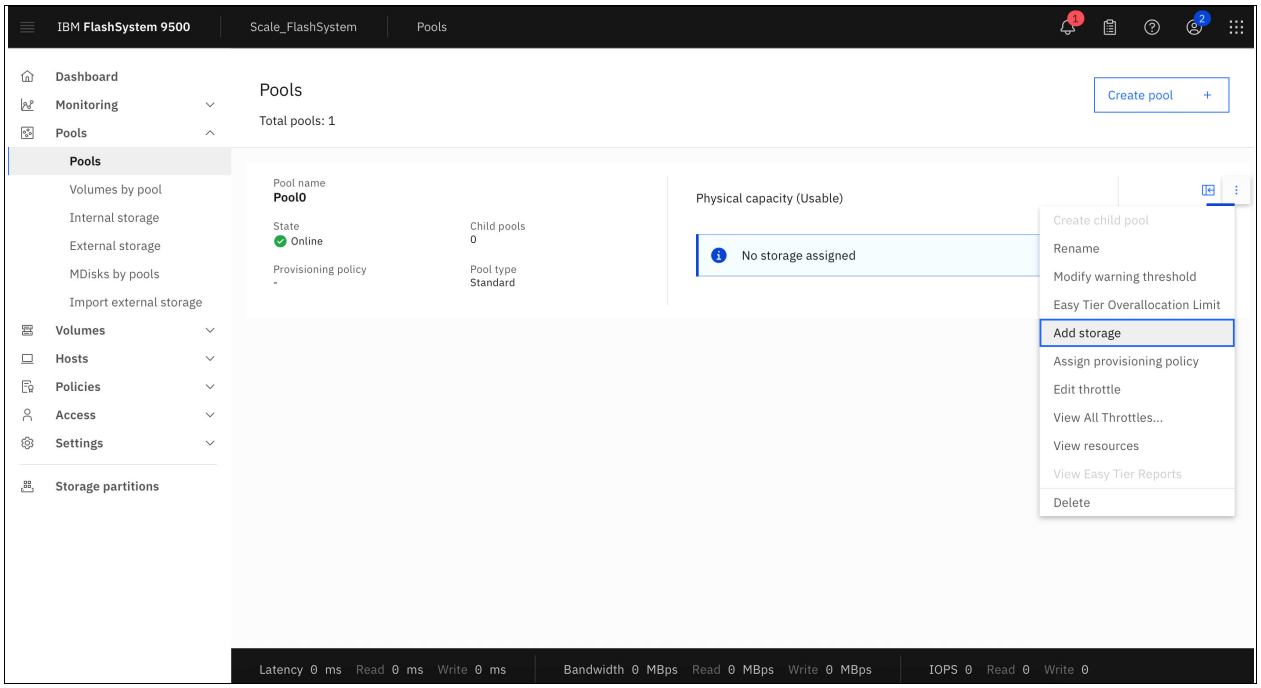


Figure 6 FlashSystem Actions menu

5. On the Add Storage page, select the required details such as raid level and number of drives, and click Add **Storage**. (Figure 7 on page 9)

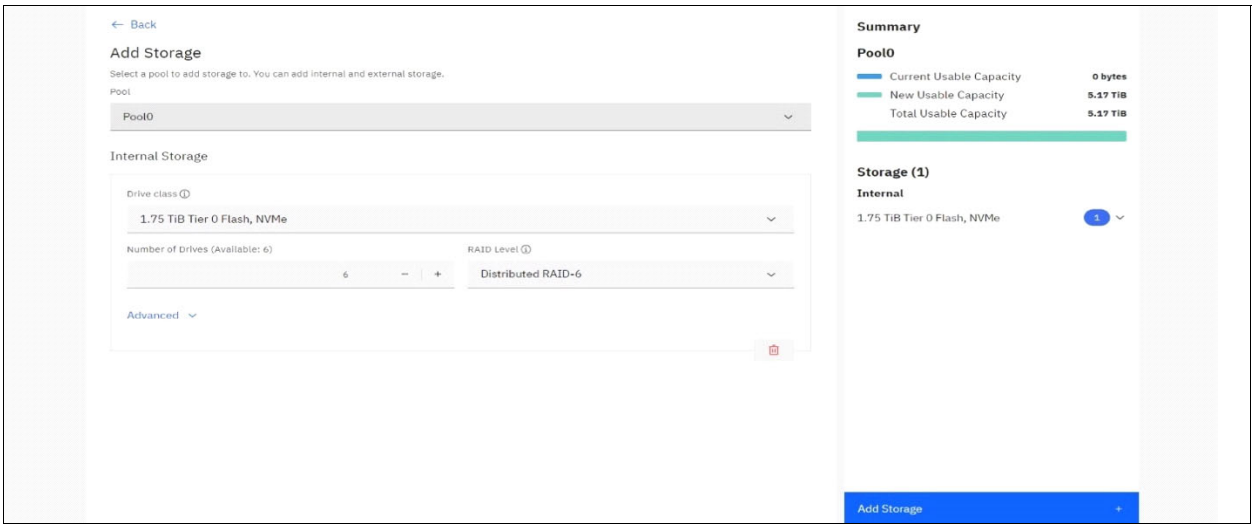


Figure 7 Add Storage page

6. Click **Pools** and select the pool created in the previous step.
7. Click **Actions** → **Create Child Pool**. (Figure 8)

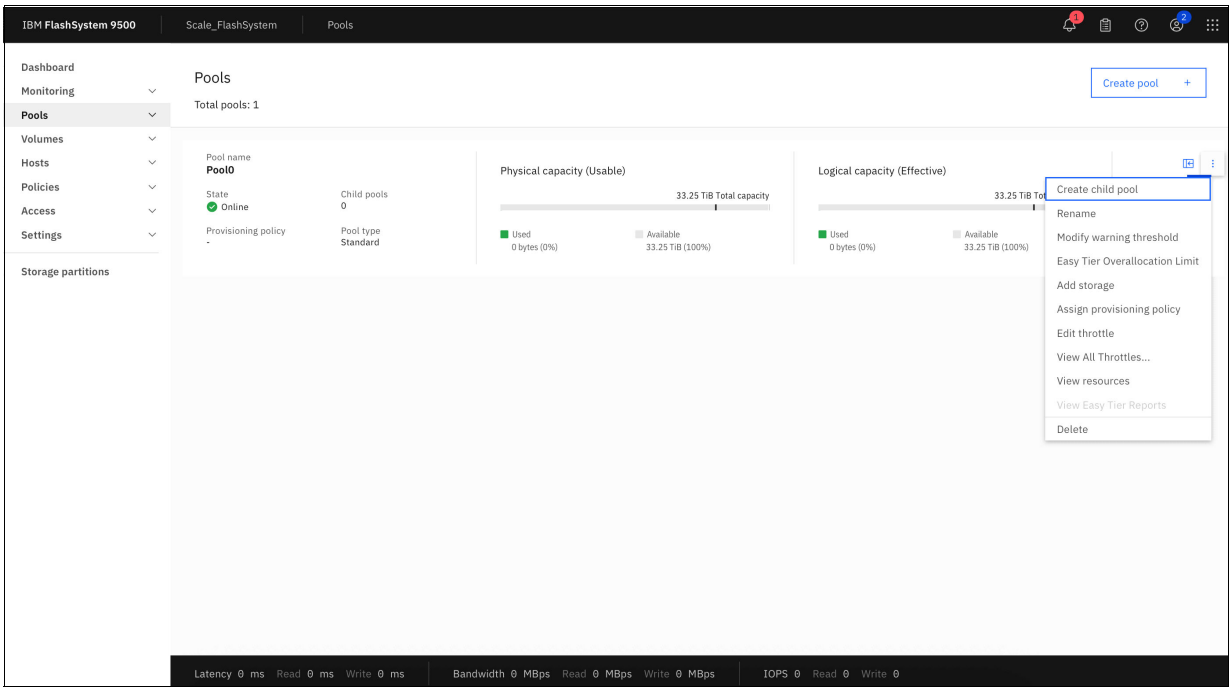


Figure 8 Create Child Pool

8. Enter a name for the child pool and click **Create**. (Figure 9)

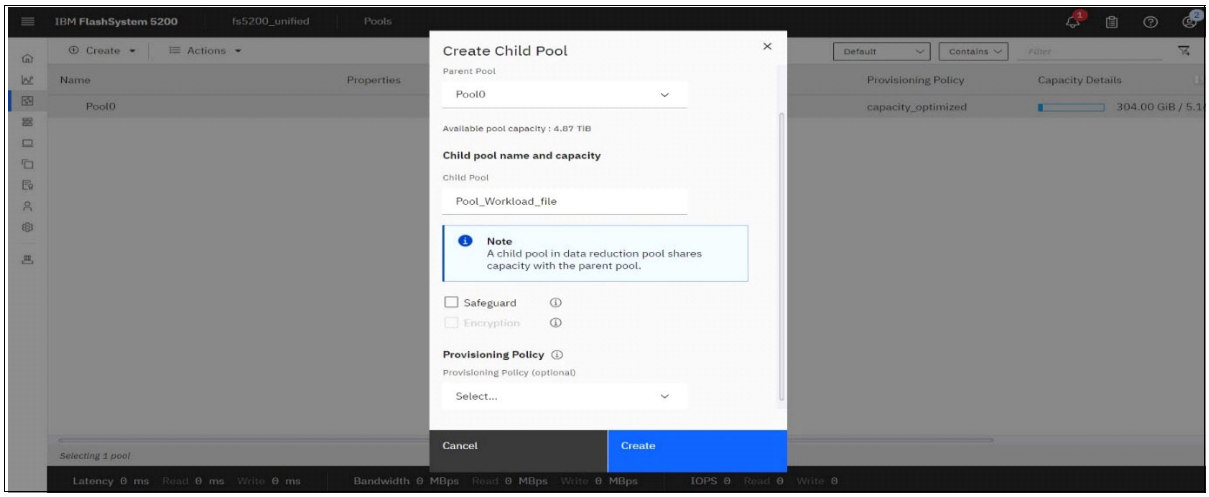


Figure 9 Enter Child Pool information

Note: Currently it is not recommended to use thin provisioned, compressed, and deduplicated volumes for this use case. Therefore, when you create a child pool, ensure that no provisioning policy is selected.

Creating a volume from the child pool

1. Open the FlashSystem GUI and click **Volumes** → **Create Volumes**. (Figure 10 and Figure 11 on page 11)

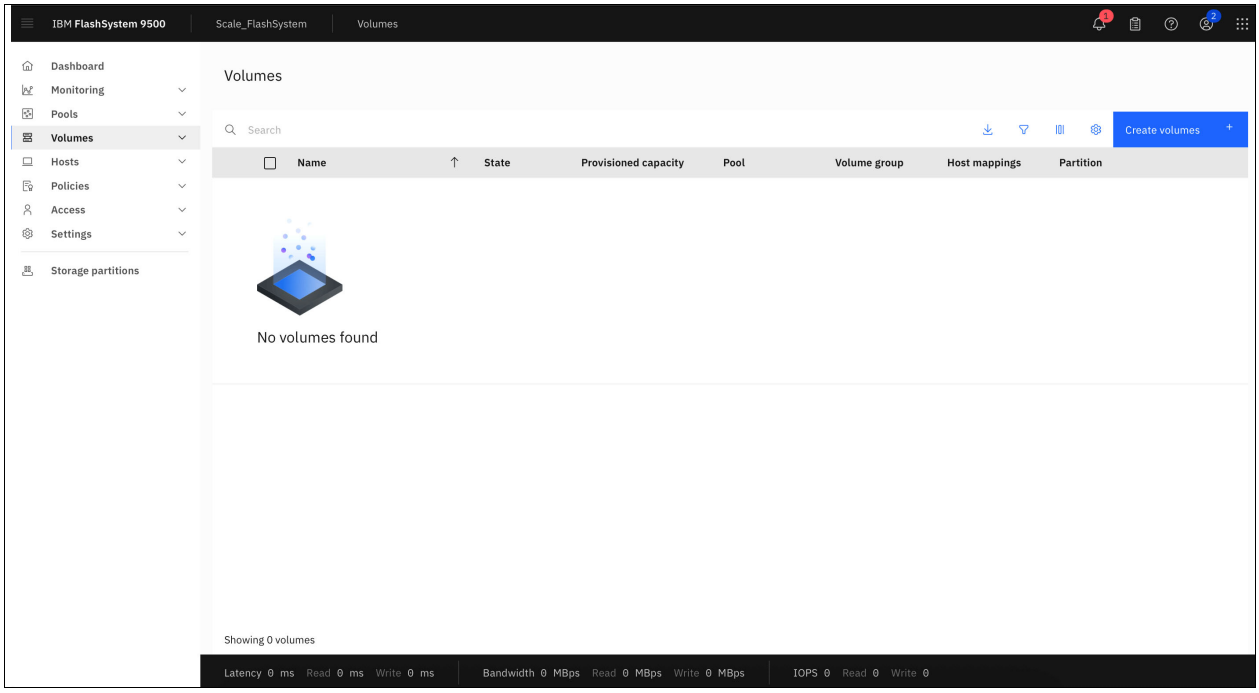


Figure 10 Volumes page

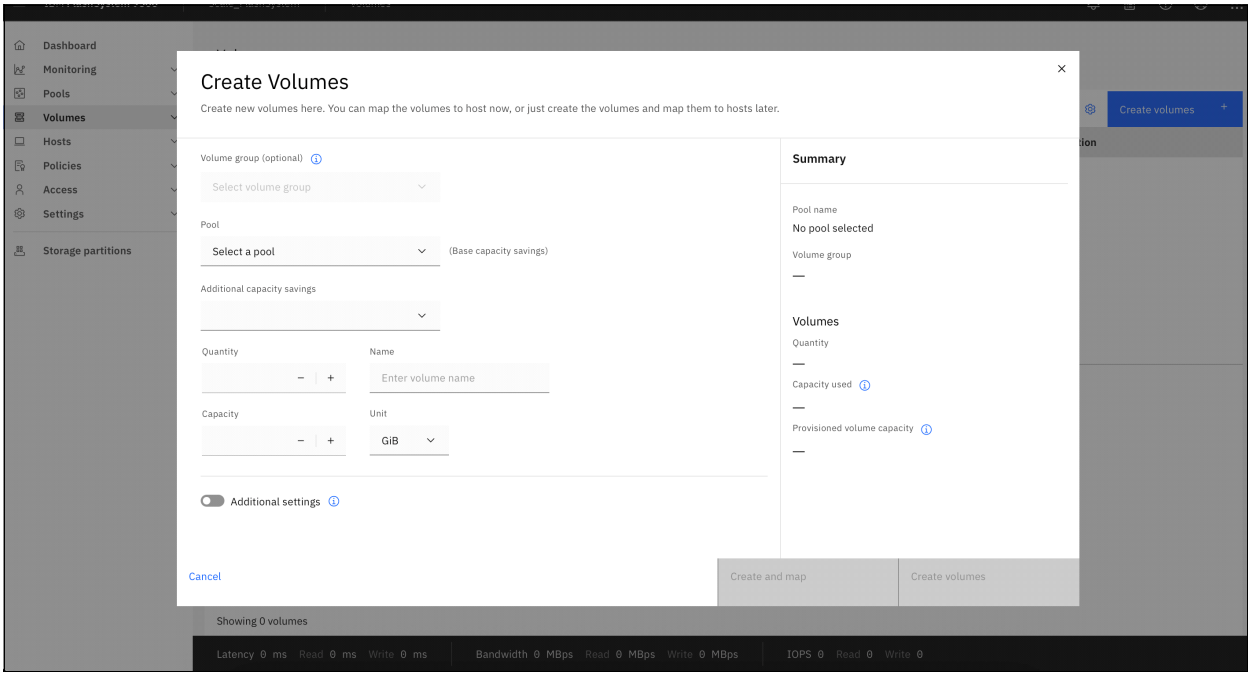


Figure 11 Creating a volume

2. In the Define Volume Properties dialog, enter the child pool name, define volume properties, and click **Create Volumes**: (Figure 12)
   - Define the capacity of the volume which is supplied by the user
   - Create a minimum of six volumes

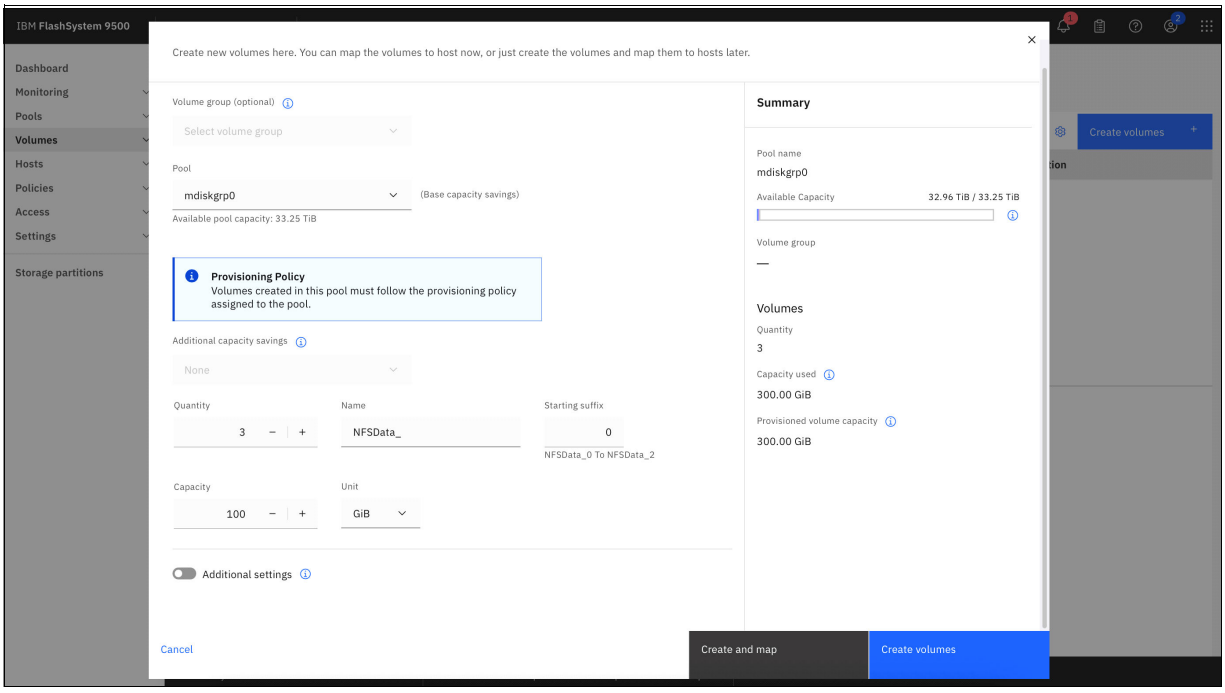


Figure 12 Define Volume Properties

Creating host and host cluster objects

1. Log in to the FlashSystem GUI and click **Hosts** → **Add Host**. (Figure 13)
2. In the Add Host dialog, enter the following details and click **Save**:
   - Name: *<Desired name for the host>*
   - Host Connection: Fibre Channel (SCSI)
   - Host Port: *<WWPN of the server HBA port>*
   - Host type: Generic

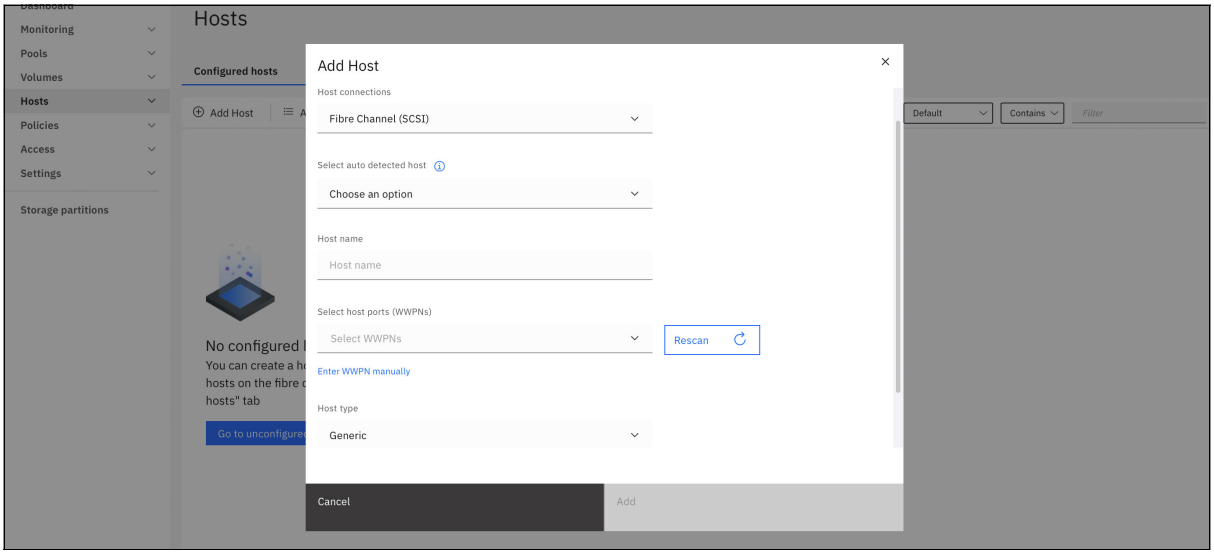


Figure 13 Add Host

3. Click **Hosts** → **Host Clusters** and click **Create Host Cluster**. (Figure 14)
4. In the Create Host Cluster dialog, enter the following details:
   - Name: *<Enter a name for the host cluster>*
   - Select all the host objects that you want to add to the host cluster and click **Create**.

   In this example, all NSD (Network Shared Disk) servers are selected.

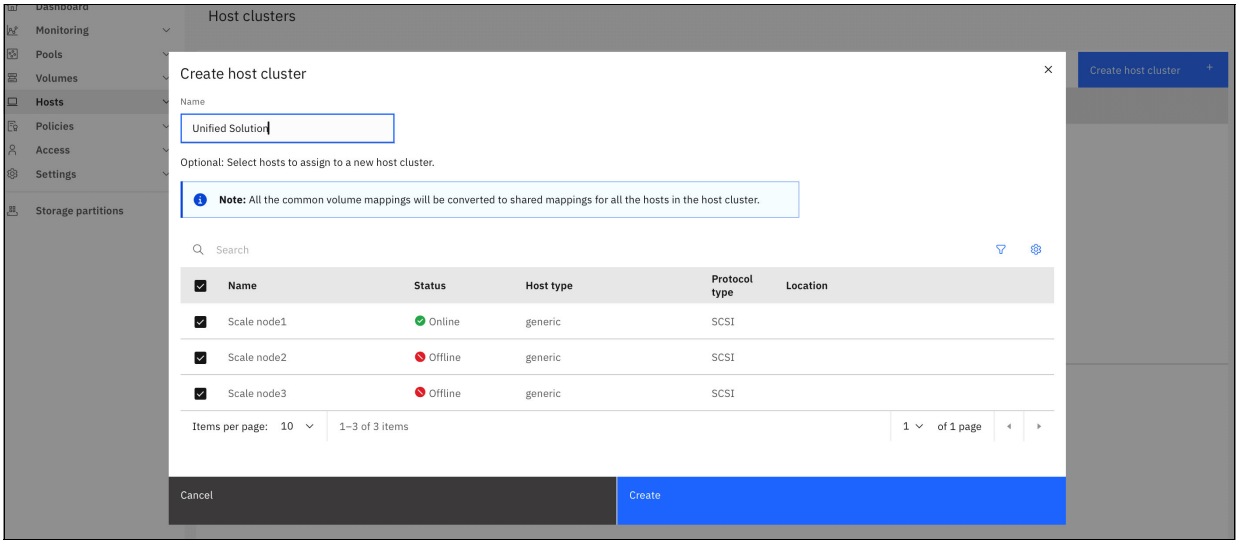


Figure 14 Create Host Cluster

5. Verify the host cluster members. (Figure 15)

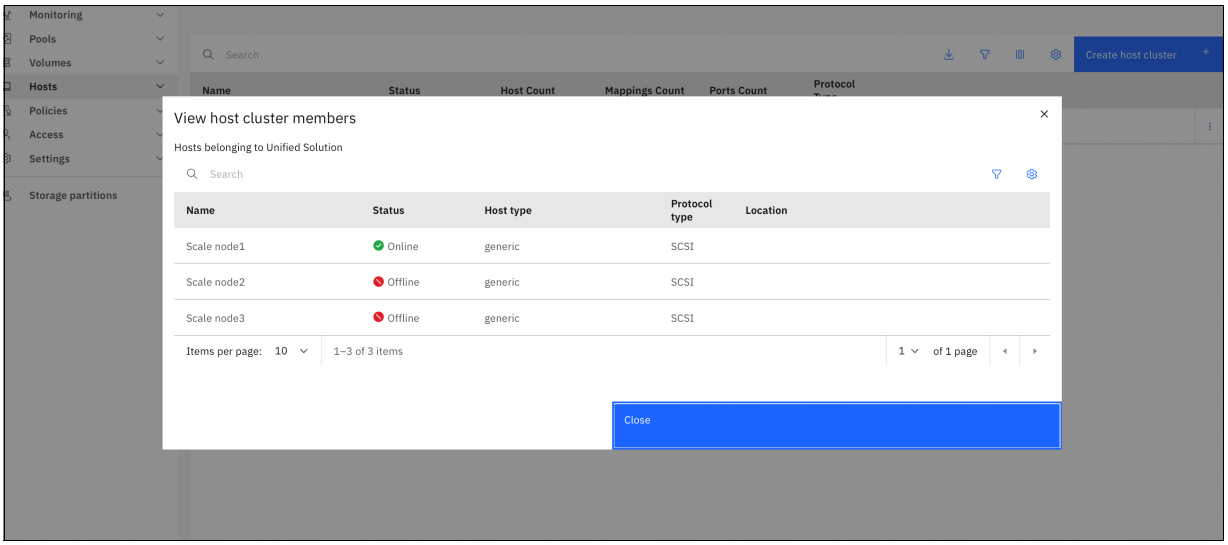


Figure 15 Create Host Cluster: Summary

Mapping the volume to the host cluster

1. Log in to the FlashSystem GUI and click **Volumes**.
2. Select the volumes to be mapped to the host cluster and click **Actions** → **Create Host Mappings**. (Figure 16)

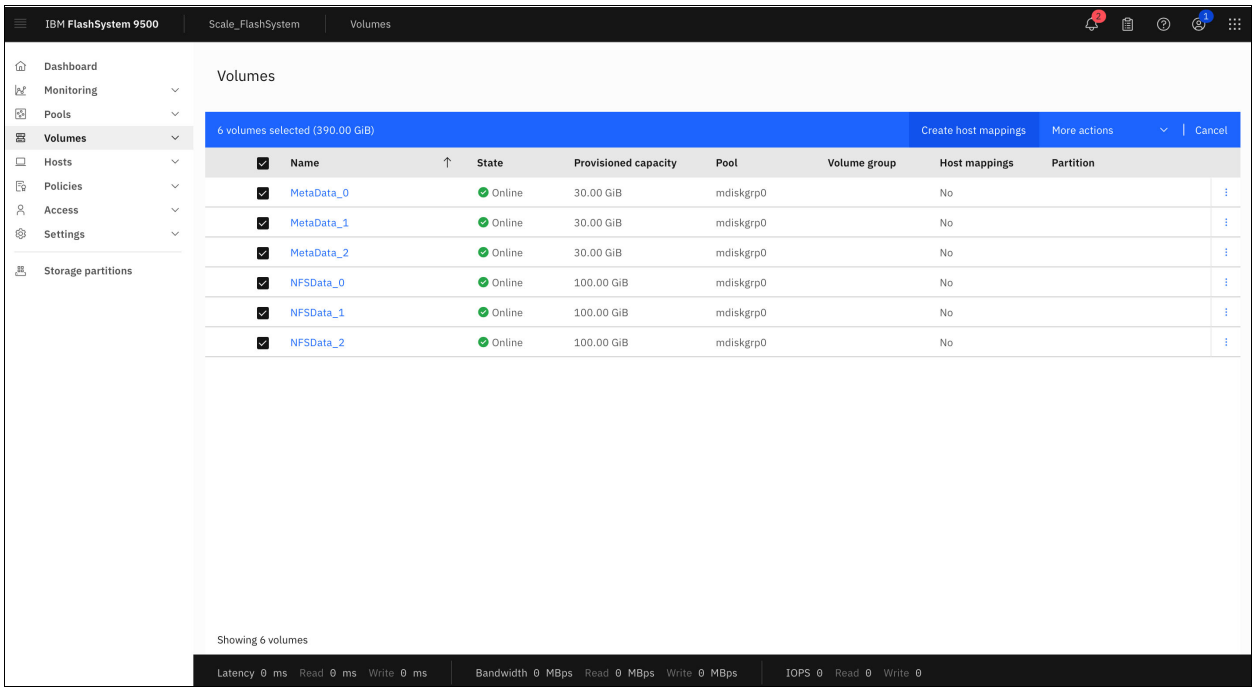


Figure 16 Select volumes

3. In the Create Mappings to section, select **Host Clusters** and click **Next**. (Figure 17 on page 14)

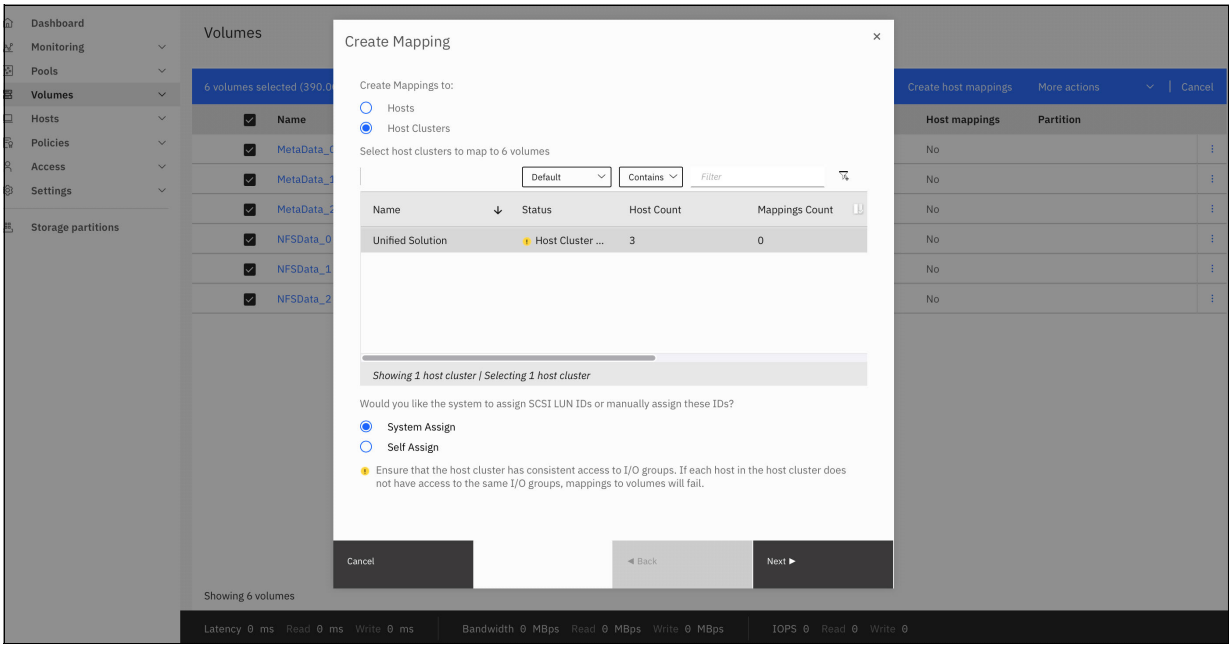


Figure 17 Create Mapping

4. Click **Map Volumes**.(Figure 18)

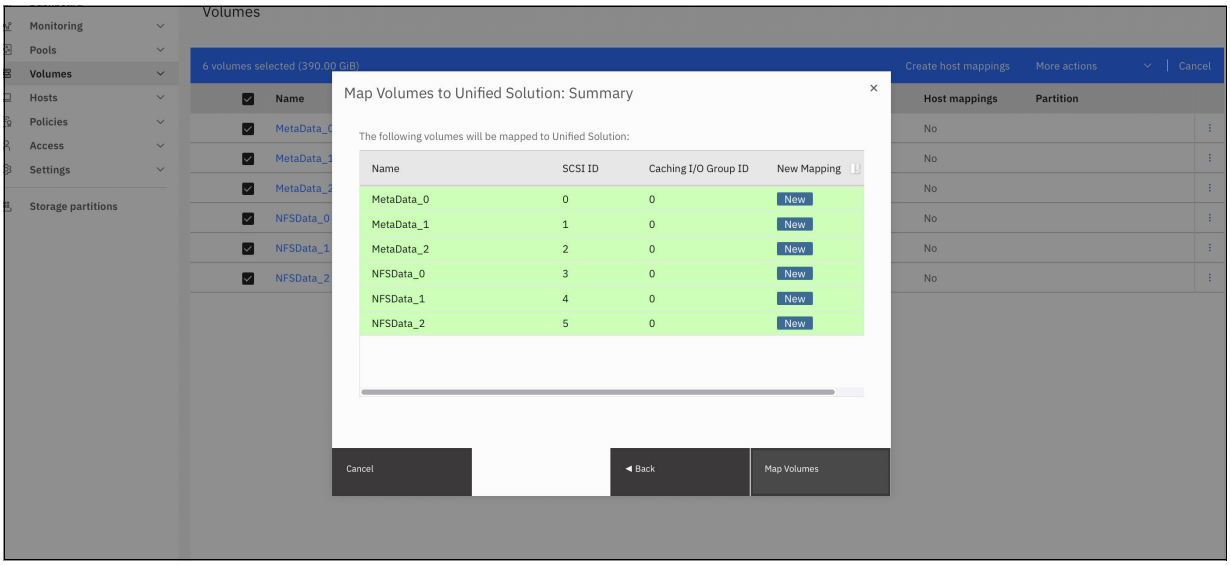


Figure 18 Map volumes

Provisioning space to create a network file system by using the CLI

To create a network filesystem, you must allocate capacity from the backend storage. This solution uses IBM FlashSystem as backend storage. This section describes the process to allocate space from IBM FlashSytem storage to IBM Storage Scale by using the CLI.

Perform the following steps to provision space from IBM FlashSystem storage to create network filesystem:

Creating a storage pool and a volume in the child pool

1. Create a parent pool: **`mkmdiskgrp -name pool1 -ext 2048`**

   For more information, see mkmdiskgrp.
2. Add capacity to the pool (enter the command on one line):

   ```
   mkdistributedarray -level raid5 -driveclass 0 -drivecount 56 -stripewidth 8
   pool1
   ```

   For more information, see mkdistributedarray.
3. Create a child pool:

   ```
   mkmdiskgrp -name child_pool1-size 100 -unit tb -parentmdiskgrp pool1
   ```

   For more information, see mkmdiskgrp.
4. Create a volume in the child pool:

   ```
   mkvolume -gui -name NFSData_0 -pool 1 -size 107374182400 -unit b
   ```

   For more information, see mkvolume.

Creating a host and host cluster objects and mapping the volume

1. Create a host:

   ```
   mkhost -fcwwpn -iogrp 0:1:2:3 -name <hostname> -protocol fcscsi -type generic 4
   ```

   For more information, see mkhost.
2. Create a host cluster

   ```
   mkhostcluster -name <name> -seedfromhost <host objects>
   ```

   For more information, see mkhostcluster.
3. Map the volume to the host cluster:

   ```
   mkvolumehostclustermap -force -hostcluster 0 -scsi 0 15
   ```

   For more information, see mkvolumehostclustermap.

Configure multipathing on host

Multipathing settings can change based on operating systems. This publication provides multipath settings for Red Hat Enterprise Linux (RHEL). For more information for multipath settings for other operating systems, see SCSI Fibre Channel host attachments.

Note: Multipathing configuration should be completed on all Storage Scale nodes.

Perform the following steps to configure multipathing in RHEL:

1. Check whether the `/etc/multipath.conf` file is available. If the file is not available, copy it from `/usr/share/doc/device-mapper-multipath-0.4.7/multipath.conf.defaults` by using the following command:

   ```
   cp /usr/share/doc/device-mapper-multipath/multipath.conf /etc/multipath.conf
   ```

2. Edit the `/etc/multipath.conf` file and add the script from Example 2 into the devices section at the end of the file (no need to uncomment any of the commented lines).

   Example 2 Devices script

   ```
   devices {
           device {
                  vendor "IBM"
                   product "2145"
                   path_grouping_policy "group_by_prio"
                   path_selector "service-time 0"
                   prio "alua"
                   path_checker "tur"
                   failback "immediate"
                   no_path_retry 5
                   rr_weight uniform
                   rr_min_io_rq "1"
                   dev_loss_tmo 120
           }
       }
   ```

3. Ensure that the multipath module is loaded into the kernel with the following command:

   ```
   modprobe --verbose dm-multipath
   ```

   Sample output:

   ```
   [root@Scalenode3 ~]# modprobe --verbose dm-multipath insmod
   /lib/modules/4.18.0-372.9.1.el8.x86_64/kernel/drivers/md/dm-multipath.ko.xz
   ```

4. Restart the multipath service:

   ```
   systemctl restart multipathd
   ```

5. Ensure that the multipath service is running.

   ```
   systemctl status multipathd
   ```

   Example 3 shows a sample output.

   Example 3 Sample output of multipathd status

   ```
   [root@Scalenode2 ~]# systemctl status multipathd
   ? multipathd.service - Device-Mapper Multipath Device Controller
   Loaded: loaded (/usr/lib/systemd/system/multipathd.service; enabled; vendor
   preset: enabled)
   Active: active (running) since Thu 2023-07-20 06:16:41 IST; 9s ago
   ```

```
Process: 305143 ExecStartPre=/sbin/multipath -A (code=exited, status=0/SUCCESS)
Process: 305141 ExecStartPre=/sbin/modprobe -a scsi_dh_alua scsi_dh_emc
scsi_dh_rdac dm-multipath (code=exited, status=0/SUCCESS)
   Main PID: 305145 (multipathd)
Status: "up"
Tasks: 7
Memory: 15.7M
CGroup: /system.slice/multipathd.service
  ··305145 /sbin/multipathd -d -s
Jul 20 06:16:40 Scalenode2 multipathd[305145]: --------start up--------
Jul 20 06:16:40 Scalenode2 multipathd[305145]: read /etc/multipath.conf
Jul 20 06:16:40 Scalenode2 multipathd[305145]: path checkers start up
Jul 20 06:16:40 Scalenode2 multipathd[305145]: mpatha: load table [0 209715200
multipath 1 queue_if_no_path 1 alua 2 1 service-time 0 >
Jul 20 06:16:41 Scalenode2 multipathd[305145]: mpathb: load table [0 209715200
multipath 1 queue_if_no_path 1 alua 2 1 service-time 0 >
Jul 20 06:16:41 Scalenode2 systemd[1]: Started Device-Mapper Multipath Device
Controller
```

6. Run the **`multipath -ll`** command to verify whether the multipath configuration is working as expected.

For more information, see Configuring the Linux operating system.

For additional information about Linux host multipathing, see Multipath support for hosts that are running the Linux operating system.

Scanning the volumes on the host

After the volumes are mapped from the storage and multipathing is configured on the host, the volumes must be scanned on the host. Perform the following steps to scan the volumes on the host:

1. Scan new devices:

   ```
   sh rescan-scsi-bus.sh -r
   ```

2. Flush all unused multipath device maps:

   ```
   multipath -F
   ```

3. List multipath devices:

   ```
   multipath
   ```

4. Verify that the logical unit numbers (LUNs) are discovered successfully without a reboot:

   ```
   multipath -ll
   ```

5. If the Linux host does not have the `rescan-scsi-bus.sh` file, run the following SCSI rescan command to scan the new LUNs:

   ```
   echo "- - -" > /sys/class/scsi_host/hostX/scan
   ```

Prerequisites IBM Storage Scale

This section describes the steps that are required to complete the prerequisites before installing IBM Storage Scale.

Updating the /etc/hosts file for local DNS name resolution

IBM Storage Scale uses DNS name for communication between nodes. Edit the `/etc/hosts` file for local name resolution and create and entry for each IBM Storage Scale node by using the following format:

<IP Address> <FQDN> <Short Name/Alias>

Example 4 shows an example of the entries in an /etc/hosts file:

Example 4 Example /etc/hosts file

```
127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6
192.168.69.115 scalenode1.local scalenode1
192.168.69.116 scalenode2.local scalenode2
192.168.69.117 scalenode3.local scalenode3
```

After updating the `/etc/hosts` file, ping the nodes by using the fully qualified domain name (FQDN) and alias. Use the ping command to verify the connection by using the DNS name and the alias. See Example 5.

Example 5 Verify connection with the ping command

```
ping scalenode1.local
ping scalenode1
ping scalenode2.local
ping scalenode2
ping scalenode3.local
ping scalenode3
```

Configuring passwordless SSH between IBM Storage Scale nodes

IBM Storage Scale requires passwordless SSH for the root user between nodes to communicate with each other. Perform the following steps to configure passwordless SSH for root user:

1. Log in to IBM Storage Scale node#1 as root and generate an SSH key pair by using the following command (The passphrase must be empty.):

   ```
   [root@Scalenode1 ~]# ssh-keygen -t rs
   Generating public/private rsa key pair.
   Enter file in which to save the key (/root/.ssh/id_rsa):
   Created directory '/root/.ssh'.
   Enter passphrase (empty for no passphrase):
   Enter same passphrase again:
   Your identification has been saved in /root/.ssh/id_rsa.
   Your public key has been saved in /root/.ssh/id_rsa.pub.
   The key fingerprint is:
   SHA256:8rR4slUGysX+jWeRKI/cI3i+rjP6x6rVWI6mvgNNdGQ root@Scalenode1
   ```

2. Repeat step#1 for IBM Storage Scale node#2 and IBM Storage Scale node#3.
3. Set up passwordless SSH for the root user on IBM Storage Scale node#1 as shown in using the following command:

   Example 6 Setting up passwordless SSH

   ```
   [root@Scalenode1 ansible-toolkit]# ssh-copy-id scalenode1
   ```

```
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed:
"/root/.ssh/id_rsa.pub"
The authenticity of host 'scalenode1 (192.168.63.22)' can't be established.
ECDSA key fingerprint is SHA256:piV7wP6SqX5DxthQs9h4GduqGT6/ZLQ2YDDKMyh/58A.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter
out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are
prompted now it is to install the new keys
root@scalenode1's password:
Number of key(s) added: 1
```

4. Verify that the passwordless SSH is working on IBM Storage Scale node#1 by using the following command:

   ```
   ssh 'scalenode1
   ```

   Note: The verification is successful if the system connects without requesting a username and password.

5. Repeat steps 3–4 for IBM Storage Scale node#2 and IBM Storage Scale node#3
6. Copy the SSH key from IBM Storage Scale node#1 to IBM Storage Scale node#2 and IBM Storage Scale node#3 by using the following command:

   ```
   ssh-copy-id <IBM Storage Scale node name>
   ```

   Note: Perform this step on IBM Storage Scale node#1.

   Example 7 provides an example of the expected output.

   Example 7 Output example copying SSH key

   ```
   [root@Scalenode1 ~]# ssh-copy-id scalenode2
   /usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed:
   "/root/.ssh/id_rsa.pub"
   The authenticity of host 'scalenode2 (192.168.63.21)' can't be established.
   ECDSA key fingerprint is SHA256:9x7dfCF3Q6nxavhEwTD6Y9UPOMPj9KetfpLhG+7esk0.
   Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
   /usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter
   out any that are already installed
   /usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are
   prompted now it is to install the new keys
   root@scalenode2's password:
   Number of key(s) added: 1
   ```

7. Repeat step#6 to copy the SSH key from IBM Storage Scale node#2 to IBM Storage Scale node#1 and IBM Storage Scale node#3.
8. Repeat step#6 to copy the SSH key from IBM Storage Scale node#3 to IBM Storage Scale node#1 and IBM Storage Scale node#2.

9. Verify if the promptless SSH is working between all nodes and themselves by using IP, FQDN, and hostname. Verify SSH access from each node as shown in Example 8, Example 9, and Example 10.

Example 8 Verify promptless SSH connections from scalenode1

```
[root@Scalenode1 ~]# ssh root@scalenode1
[root@Scalenode1 ~]# ssh root@scalenode2
[root@Scalenode1 ~]# ssh root@scalenode3
[root@Scalenode1 ~]# ssh root@scalenode1.local
[root@Scalenode1 ~]# ssh root@scalenode2.local
[root@Scalenode1 ~]# ssh root@scalenode3.local
```

Example 9 Verify promptless SSH connections from scalenode2

```
[root@Scalenode2 ~]# ssh root@scalenode2
[root@Scalenode2 ~]# ssh root@scalenode1
[root@Scalenode2 ~]# ssh root@scalenode3
[root@Scalenode2 ~]# ssh root@scalenode1.local
[root@Scalenode2 ~]# ssh root@scalenode2.local
[root@Scalenode2 ~]# ssh root@scalenode3.local
```

Example 10 Verify promptless SSH connections from scalenode3

```
[root@Scalenode3 ~]# ssh root@scalenode3
[root@Scalenode3 ~]# ssh root@scalenode1
[root@Scalenode3 ~]# ssh root@scalenode2
[root@Scalenode3 ~]# ssh root@scalenode1.local
[root@Scalenode3 ~]# ssh root@scalenode2.local
[root@Scalenode3 ~]# ssh root@scalenode3.local
```

At this point in the configuration, the nodes can communicate with each other without prompting for user ID and password.

Ensuring that firewall ports are open

The Linux operating system that is running on IBM Storage Scale nodes might have a firewall enabled, which can block communication between nodes. Ensure that all necessary ports are open for internal (for GPFS and the installer) and external (for the protocols) communication. For more information about the ports that need to be open, see Securing the IBM Storage Scale system using firewall.

Note: The installation toolkit verifies whether the firewall daemon is running and displays a warning if it is running. If the required ports are open, you can ignore the warning.

Configuring OS repositories

Configure a base repository on every node. It is recommended to configure the BaseOS and the AppStream repositories.

For more information about configuring a local RHEL9 repository, see Need to set up yum repository for locally-mounted DVD on Red Hat Enterprise Linux 9.

Install Python 3.8 or later

The IBM Storage Scale installation toolkit uses Ansible and Python on the back end to perform the deployment process. Therefore, ensure that the correct version of Python is installed on the IBM Storage Scale Nodes.

RHEL9, by default, has the required Python version installed. If Python is installed, skip this step. However, if you uninstalled Python, from Red Hat Enterprise Linux 9, Perform the following steps to install Python 3.8 or later versions:

1. Verify whether Python is installed by using the following commands:
   a. Determine the path for Python:

      ```
      [root@Scalenode1 ansible-toolkit]# which python3
      /usr/bin/python3
      ```

   b. Determine the version of Python:

      ```
      [root@Scalenode1 ansible-toolkit]# python3 --version
      Python 3.6.8
      ```

2. If the Python version is earlier than 3.8, update Python version. (See using the following command:

Example 11 Updating Python

```
[root@mm-u-poc1 ansible-toolkit]# dnf install python3
Updating Subscription Management repositories.
Last metadata expiration check: 2:36:27 ago on Thu 20 Jul 2023 12:34:05 PM PDT.
Dependencies resolved.
================================================================================
Package             Architecture    Version
Repository                                  Size
================================================================================
Installing:
python36           x86_64        3.6.8-38.module+el8.5.0+12207+5c5719bc
rhel-8-for-x86_64-appstream-rpms           19 k
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>Output is
Trimmed<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
Installed:
python3-pip-9.0.3-22.el8.noarch
python36-3.6.8-38.module+el8.5.0+12207+5c5719bc.x86_64
Complete!
```

Installing OS-dependent packages

IBM Storage Scale requires OS-level packages to be installed. Run the command in Example 12 on page 22 to install the required packages and dependencies.

Note: Change the versions in `kernel-devel-5.14.0-503.11.1.el9_5.x86_64` and `kernel-headers-5.14.0-503.11.1.el9_5.x86_64` to match the installed version. You can determine the current kernel version by running the **`uname -a`** command.

Example 12 Installing OS-level packages

```
yum install kernel-devel-5.14.0-503.11.1.el9_5.x86_64
kernel-headers-5.14.0-503.11.1.el9_5.x86_64 cpp gcc gcc-c++ elfutils
elfutils-devel make rpcbind
```

Install these dependencies on all Storage Scale nodes.

Copying and extracting IBM Storage Scale packages

This section outlines steps to copy and extract the IBM Storage Scale packages required for the deployment.

Copying IBM Storage Scale package

Perform the following steps to copy the IBM Storage Scale package:

1. Create a temporary directory in IBM Storage Scale node#1 using the following command:

   ```
   mkdir <directory name>
   ```

2. Copy the IBM Storage Scale package binary file to the temporary directory using secure copy protocol (scp) or WinSCP. The name of the IBM Storage Scale binary file is `Storage_Scale_Data_Management-5.2.3.4-x86_64-Linux-install`.

   The name can vary depending on the license, version, and platform.

Extracting IBM Storage Scale package

Perform the following steps to extract IBM Storage Scale package:

1. Go to the temporary directory created in IBM Storage Scale Node#1 where the IBM Storage Scale binary file is copied.
2. Use the following command (on one line) to add execute permission to the IBM Storage Scale binary file:

   ```
   [root@Scalenode1 GA]# chmod +x
   Storage_Scale_Data_Management-5.2.3.4-x86_64-Linux-install
   ```

3. Install IBM Storage Scale software by using the command in Example 13. Enter **1** to accept the license terms.

   Example 13 Running Storage Scale

   ```
   [root@Scalenode1 GA]#
   ./Storage_Scale_Data_Management-5.2.3.4-x86_64-Linux-install
   Extracting License Acceptance Process Tool to /usr/lpp/mmfs/5.2.3.4 ...
   tail -n +680 ./Storage_Scale_Data_Management-5.2.3.4-x86_64-Linux-install | tar
   -C /usr/lpp/mmfs/5.2.3.4 -xvz --exclude=installer --exclude=*_rpms
   --exclude=*_debs --exclude=*rpm  --exclude=*gz --exclude=*deb --exclude=*tools*
   1> /dev/null
   >>>>>>>>>>>>>>>>>>>>>>>>>>>Output is trimmed<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
   Press Enter to continue viewing the license agreement, or
   enter "1" to accept the agreement, "2" to decline it, "3"
   to print it, "4" to read non-IBM terms, or "99" to go back
   to the previous screen.
   License Agreement Terms accepted.
   Extracting Product RPMs to /usr/lpp/mmfs/5.2.3.4 ...
   >>>>>>>>>>>>>>>>>>>>>>>>>>>Output is trimmed<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
   ```

Installing IBM Storage Scale and creating a cluster

Use the following high-level steps to create an IBM Storage Scale cluster:

1. Create IBM Storage Scale cluster configuration.
2. Run IBM Storage Scale install pre-check.
3. Run IBM Storage Scale deployment.
4. Configure protocols in IBM Storage Scale Cluster.
5. Configure authentication for NFS exports/shares.
6. Create NFS exports/shares.
7. Install and configure IBM Storage Scale GUI.

Creating the IBM Storage Scale cluster configuration

Use the following procedure to create IBM Storage Scale cluster configuration. This process is documented in the PDF file IBM Storage Scale 5.2.2 Protocols Quick Overview. Examine the column under **Before starting** to ensure the prerequisites are met and the correct files are extracted. For more information on configuring the installation toolkit and creating the IBM Storage Scale cluster configuration, see the linked PDF file.

Note: Ensure that all three IBM Storage Scale nodes can communicate with each other over the Ethernet network before proceeding.

Setting up IBM Storage Scale nodes:

Perform the following steps to set up IBM Storage Scale nodes:

1. Navigate to the IBM Storage Scale installer path on IBM Storage Scale node#1 by using the following command:

   ```
   cd /usr/lpp/mmfs/5.2.3.4/ansible-toolkit/
   ```

2. Configure IBM Storage Scale node#1 by using the following command:

   ```
   ./spectrumscale setup -s <IBM Storage Scale Node#1 IP address>
   ```

   Example 14 lists an example of the output:

   Example 14 Output of the spectrumscale setup command

```
[root@Scalenode1 ansible-toolkit]# ./spectrumscale setup -s 192.168.69.115
[ INFO  ] Installing prerequisites for install node
[ INFO  ] Installing Ansible..
[ INFO  ] Installing ansible-core....
[ INFO  ] Install Toolkit setup type is set to IBM Storage Scale (default). If
an ESS is in the cluster, run this command to set ESS mode: ./spectrumscale
setup -s server_ip -st ess
[ INFO  ] Your ansible controller node has been configured to use the IP
192.168.69.115 to communicate with other nodes.
[ INFO  ] Port 10080 will be used for package distribution.
[ INFO  ] SUCCESS
[ INFO  ] Tip : Designate protocol, nsd and admin nodes in your environment to
use during install:./spectrumscale -v node add <node> -p  -a -n
```

3. Add IBM Storage Scale node#1 in installation toolkit configuration by using the following command:

   ```
   ./spectrumscale node add <node name> -q -m -a -n -p -g
   ```

   Example 15 lists an example of the output.

   Example 15 Output of the spectrumscale node add command

   ```
   [root@Scalenode1 ansible-toolkit]# ./spectrumscale node add scalenode1.local -q
   -m -a -n -p -g
   [ INFO  ] Adding node scalenode1.local as a GPFS node.
   [ INFO  ] Setting scalenode1.local as a protocol node.
   [ INFO  ] Configuration updated.
   [ INFO  ] Tip : If all node designations are complete, configure the protocol
   environment as needed: ./spectrumscale config protocols -f cesSharedRoot -m
   /ibm/cesSharedRoot
   [ INFO  ] Adding node scalenode1.local as a quorum node.
   [ INFO  ] Adding node scalenode1.local as a manager node.
   [ INFO  ] Adding node scalenode1.local as an NSD server.
   [ INFO  ] Configuration updated.
   [ INFO  ] Tip :If all node designations are complete, add NSDs to your cluster
   definition and define required filessytems:./spectrumscale nsd add <device> -p
   <primary node> -s <secondary node> -fs <file system>
   [ INFO  ] Setting scalenode1.local as an admin node.
   [ INFO  ] Configuration updated.
   [ INFO  ] Tip : Designate protocol or nsd nodes in your environment to use
   during install:./spectrumscale node add <node> -p -n
   [ INFO  ] Setting scalenode1.local as a GUI server.
   ```

4. Add IBM Storage Scale node#2 and IBM Storage Scale node#3 in the install toolkit configuration by using the following command:

   ```
   ./spectrumscale node add <node name> -q -m -a -n -p
   ```

   Example 16 lists the command and output to add IBM Storage Scale node#2.

   Example 16 Adding scalenode2

   ```
   [root@Scalenode1 ansible-toolkit]# ./spectrumscale node add scalenode2.local -q
   -m -a -n -p
   [ INFO  ] Adding node scalenode2.local as a GPFS node.
   [ INFO  ] Setting scalenode2.local as a protocol node.
   [ INFO  ] Configuration updated.
   [ INFO  ] Tip : If all node designations are complete, configure the protocol
   environment as needed: ./spectrumscale config protocols -f cesSharedRoot -m
   /ibm/cesSharedRoot
   [ INFO  ] Adding node scalenode2.local as a quorum node.
   [ INFO  ] Adding node scalenode2.local as a manager node.
   [ INFO  ] Adding node scalenode2.local as an NSD server.
   [ INFO  ] Configuration updated.
   [ INFO  ] Tip :If all node designations are complete, add NSDs to your cluster
   definition and define required filessytems:./spectrumscale nsd add <device> -p
   <primary node> -s <secondary node> -fs <file system>
   [ INFO  ] Setting scalenode2.local as an admin node.
   [ INFO  ] Configuration updated.
   [ INFO  ] Tip : Designate protocol or nsd nodes in your environment to use
   during install:./spectrumscale node add <node> -p -n
   ```

Example 17 lists the command and output to add IBM Storage Scale node#3:

Example 17 Adding scalenode3

```
[root@Scalenode1 ansible-toolkit]# ./spectrumscale node add scalenode3.local -q
-m -a -n -p
[ INFO  ] Adding node scalenode3.local as a GPFS node.
[ INFO  ] Setting scalenode3.local as a protocol node.
[ INFO  ] Configuration updated.
[ INFO  ] Tip : If all node designations are complete, configure the protocol
environment as needed: ./spectrumscale config protocols -f cesSharedRoot -m
/ibm/cesSharedRoot
[ INFO  ] Adding node scalenode3.local as a quorum node.
[ INFO  ] Adding node scalenode3.local as a manager node.
[ INFO  ] Adding node scalenode3.local as an NSD server.
[ INFO  ] Configuration updated.
[ INFO  ] Tip :If all node designations are complete, add NSDs to your cluster
definition and define required filessytems:./spectrumscale nsd add <device> -p
<primary node> -s <secondary node> -fs <file system>
[ INFO  ] Setting scalenode3.local as an admin node.
[ INFO  ] Configuration updated.
[ INFO  ] Tip : Designate protocol or nsd nodes in your environment to use
during install:./spectrumscale node add <node> -p -n
```

5. Verify that all the nodes are added successfully by using the following command:

   ```
   ./spectrumscale node list
   ```

   Example 18 provides an example of the expected output.

Example 18 Expected output of node list command

```
[ INFO  ] List of nodes in current configuration:
[ INFO  ] [Installer Node]
[ INFO  ] 192.168.69.115
[ INFO  ]
[ INFO  ] [Cluster Details]
[ INFO  ] No cluster name configured
[ INFO  ] Setup Type: IBM Storage Scale
[ INFO  ]
[ INFO  ] [Protocols]
[ INFO  ] S3     : Disabled
[ INFO  ] SMB    : Disabled
[ INFO  ] NFS    : Disabled
[ INFO  ] HDFS   : Disabled
[ INFO  ]
[ INFO  ] [Extended Features]
[ INFO  ] File Audit logging     : Disabled
[ INFO  ] Management GUI         : Enabled
[ INFO  ] Performance Monitoring : Enabled
[ INFO  ] Callhome               : Enabled
[ INFO  ]
[ INFO  ] GPFS             Admin  Quorum  Manager   NSD   Protocol   GUI
Callhome   OS   Arch
[ INFO  ] Node              Node   Node     Node   Server   Node    Server
Server
[ INFO  ] scalenode1.local   X      X       X      X       X       X
rhel9  x86_64
```

```
[ INFO  ] scalenode2.local   X       X       X       X       X
rhel9  x86_64
[ INFO  ] scalenode3.local   X       X       X       X       X
rhel9  x86_64
[ INFO  ]
[ INFO  ] [Export IP address]
[ INFO  ] No export IP addresses configured
```

Setting up Network Shared Disks to create a file system:

Network Shared Disks (NSDs) are logical mappings of storage disks, LUNs, and volumes in an IBM Storage Scale configuration. A single disk, LUN, and volume is configured as one NSD in an IBM Storage Scale configuration. NSDs are used to create file systems. One file system can have one or more NSDs.

Setting up NSDs to create data file system:

IBM Storage Scale uses file system for two purposes: storing user data and storing cluster information. This section describes the steps to configure file system for storing user data.

Perform the following steps to create data file system on IBM Storage Scale:

1. List the available disks on IBM Storage Scale node#1 using the following command and save the disk IDs to be used for file system creation:

   ```
   [root@Scalenode1 ansible-toolkit]# multipath -l
   ```

   Example 19 provides an example output.

   Example 19 Output of multipath command

   ```
   mpatha (360050764009882d2080000000000000c) dm-3 IBM,2145
   size=100G features='1 queue_if_no_path' hwhandler='1 alua' wp=rw
   |-+- policy='service-time 0' prio=0 status=enabled
   | |- 1:0:1:0  sdi 8:128  active undef running
   | `- 1:0:2:0  sdo 8:224  active undef running
   `-+- policy='service-time 0' prio=0 status=enabled
     |- 1:0:0:0  sdc 8:32   active undef running
     `- 1:0:3:0  sdu 65:64  active undef running
         >>>>>>>>>>>>>>>>>>>>>>>>>>Output is
   trimmed<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
   mpathf (360050764009882d20800000000000011) dm-12 IBM,2145
   size=30G features='1 queue_if_no_path' hwhandler='1 alua' wp=rw
   |-+- policy='service-time 0' prio=0 status=enabled
   | |- 1:0:0:5  sdh 8:112  active undef running
   | `- 1:0:3:5  sdz 65:144 active undef running
   `-+- policy='service-time 0' prio=0 status=enabled
     |- 1:0:1:5  sdn 8:208  active undef running
     `- 1:0:2:5  sdt 65:48  active undef running
   ```

2. Add each disk to the NSD configuration using the following command (on one line):

   ```
   ./spectrumscale nsd add -p scalenode1.local -s
   scalenode2.local,scalenode3.local -po system -u dataAndMetadata -fs
   <filesystem_name> -name <nsd_name> <disk_device_id>
   ```

   Example 20 on page 27 provides an example output.

Example 20 Output of nsd add command

```
[root@Scalenode1 ansible-toolkit]# ./spectrumscale nsd add -p scalenode1.local
-s scalenode2.local,scalenode3.local -po system -u dataAndMetadata -fs FS1
-name data_nsd_1 /dev/dm-3
[ INFO  ] Connecting to scalenode1.local to check devices and expand wildcards.
[ INFO  ] Looking up details of /dev/dm-3.
[ INFO  ] The installer will create the new file system FS1 if it does not
exist.
[ INFO  ] Adding NSD data_nsd_1 on scalenode1.local using device /dev/dm-3.
[ INFO  ] Configuration updated
[ INFO  ] Tip : If all node designations and any required protocol
configurations are complete, proceed to check the installation configuration:
./spectrumscale install –precheck
      >>>>>>>>>>>>>>>>>>>>>>>>>>Output is
trimmed<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
[root@Scalenode1 ansible-toolkit]# ./spectrumscale nsd add -p scalenode1.local
-s scalenode2.local,scalenode3.local -po system -u dataAndMetadata -fs FS1
-name data_nsd_3 /dev/dm-6
[ INFO  ] Connecting to scalenode1.local to check devices and expand wildcards.
[ INFO  ] Looking up details of /dev/dm-6.
[ INFO  ] Adding NSD data_nsd_3 on scalenode1.local using device /dev/dm-6.
[ INFO  ] Configuration updated
[ INFO  ] Tip : If all node designations and any required protocol
configurations are complete, proceed to check the installation configuration:
./spectrumscale install –precheck
```

3. Verify if all the NSDs are defined correctly by using the following command:

   ```
   ./spectrumscale nsd list
   ```

   Example 21 provides an example output.

Example 21 Output of nsd list command

```
[root@Scalenode1 ansible-toolkit]# ./spectrumscale nsd list
[ INFO  ] Name       FS  Size(GB) Usage           FG Pool   Device    Servers
[ INFO  ] data_nsd_1 FS1 100.0    dataAndMetadata 1  system /dev/dm-3
scalenode1.local,scalenode2.local,scalenode3.local
[ INFO  ] data_nsd_2 FS1 100.0    dataAndMetadata 1  system /dev/dm-4
scalenode1.local,scalenode2.local,scalenode3.local
[ INFO  ] data_nsd_3 FS1 100.0    dataAndMetadata 1  system /dev/dm-6
scalenode1.local,scalenode2.local,scalenode3.local
```

Setting up NSDs to create quorum file system for protocol nodes:

This section defines steps to configure file system for storing cluster information. This file system is also used as a quorum in protocol node configuration. Perform the following steps to create quorum file system on IBM Storage Scale:

1. Add each disk to the NSD configuration by using the following command (on one line):

   ```
   ./spectrumscale nsd add -p scalenode1.local -s
   scalenode2.local,scalenode3.local -po system -u dataAndMetadata -fs
   <filesystem_name> -name <nsd_name> <disk_device_id>
   ```

   Example 22 on page 28 lists an example output.

Example 22 Output of nsd add command

```
[root@Scalenode1 ansible-toolkit]# ./spectrumscale nsd add -p scalenode1.local
-s scalenode2.local,scalenode3.local -po system -u dataAndMetadata -fs
cesSharedRoot  -name ces_nsd_1 /dev/dm-8
[ INFO  ] Connecting to scalenode1.local to check devices and expand wildcards.
[ INFO  ] Looking up details of /dev/dm-8.
[ INFO  ] The installer will create the new file system cesSharedRoot if it
does not exist.
[ INFO  ] Adding NSD ces_nsd_1 on scalenode1.local using device /dev/dm-8.
[ INFO  ] Configuration updated
[ INFO  ] Tip : If all node designations and any required protocol
configurations are complete, proceed to check the installation configuration:
./spectrumscale install --precheck
[root@Scalenode1 ansible-toolkit]# ./spectrumscale nsd add -p scalenode1.local
-s scalenode2.local,scalenode3.local -po system -u dataAndMetadata -fs
cesSharedRoot  -name ces_nsd_3 /dev/dm-12
[ INFO  ] Connecting to scalenode1.local to check devices and expand wildcards.
[ INFO  ] Looking up details of /dev/dm-12.
[ INFO  ] Adding NSD ces_nsd_3 on scalenode1.local using device /dev/dm-12.
[ INFO  ] Configuration updated
[ INFO  ] Tip : If all node designations and any required protocol
configurations are complete, proceed to check the installation configuration:
./spectrumscale install –precheck
```

2. Verify whether all the NSDs are defined correctly by using the following command:

   ```
   [root@Scalenode1 ansible-toolkit]# ./spectrumscale nsd list
   ```

   Example 23 lists an example output.

Example 23 Output of nsd list command

```
[ INFO  ] Name       FS            Size(GB) Usage           FG Pool   Device
Servers
[ INFO  ] data_nsd_1 FS1           100.0    dataAndMetadata 1  system /dev/dm-3
scalenode1.local,scalenode2.local,scalenode3.local
[ INFO  ] data_nsd_2 FS1           100.0    dataAndMetadata 1  system /dev/dm-4
scalenode1.local,scalenode2.local,scalenode3.local
[ INFO  ] data_nsd_3 FS1           100.0    dataAndMetadata 1  system /dev/dm-6
scalenode1.local,scalenode2.local,scalenode3.local
[ INFO  ] ces_nsd_1  cesSharedRoot 30.0     dataAndMetadata 1  system /dev/dm-8
scalenode1.local,scalenode2.local,scalenode3.local
[ INFO  ] ces_nsd_2  cesSharedRoot 30.0     dataAndMetadata 1  system
/dev/dm-10 scalenode1.local,scalenode2.local,scalenode3.local
[ INFO  ] ces_nsd_3  cesSharedRoot 30.0     dataAndMetadata 1  system
/dev/dm-12 scalenode1.local,scalenode2.local,scalenode3.local
```

Disabling call home

During the installation of IBM Storage Scale cluster using install toolkit, disable the call home function.

Call home is an automatic email alerting and call logging feature available in IBM Storage Scale. It requires access to SMTP server. Users can enable and configure call home by using the IBM Storage Scale GUI.

Disable call home feature by using the following command:

```
./spectrumscale callhome disable
```

The following lines show an example of the expected output:

```
[root@Scalenode1 ansible-toolkit]# ./spectrumscale callhome disable
[ INFO  ] Disabling the callhome.
[ INFO  ] Configuration updated.
```

Running IBM Storage Scale installation precheck

After defining the IBM Storage Scale configuration, run the install precheck. This step verifies any issues with the configuration before proceeding with the installation. Perform the following steps to complete install precheck:

1. Navigate to the IBM Storage Scale installer path on IBM Storage Scale node#1 by using the following command:

   ```
   cd /usr/lpp/mmfs/5.2.3.4/ansible-toolkit/
   ```

2. Run the installation precheck by using the following command:

   ```
   ./spectrumscale install --precheck
   ```

Example 24 provides an example of the expected output.

Example 24 Output of precheck command

```
[root@Scalenode1 ansible-toolkit]# ./spectrumscale install --precheck
[ INFO  ] Checking cluster defnition json file inventory is
ansible/vars/scale_clusterdefinition.json
[ INFO  ] Logging to file:
/usr/lpp/mmfs/5.2.3.4/ansible-toolkit/logs/INSTALL-PRECHECK-18-10-2025_02:14:59
.log
[ INFO  ] Validating configuration
[ WARN  ] Ensure that base OS repositories are configured and enabled so that
package dependencies can be satisfied during installation.
>>>>>>>>>>>>>>>>>>>>>>>>>>>>Output is trimmed<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
[ INFO  ] ok: [scalenode3.local]
[ INFO  ] ok: [scalenode2.local]
[ INFO  ] ok: [scalenode1.local]
[ INFO  ] PLAY RECAP
*********************************************************************
[ INFO  ] localhost                  : ok=4    changed=1    unreachable=0
failed=0    skipped=2    rescued=0    ignored=0
[ INFO  ] scalenode1.local           : ok=10   changed=0    unreachable=0
failed=0    skipped=35   rescued=0    ignored=0
[ INFO  ] scalenode2.local           : ok=10   changed=0    unreachable=0
failed=0    skipped=27   rescued=0    ignored=0
[ INFO  ] scalenode3.local           : ok=10   changed=0    unreachable=0
failed=0    skipped=27   rescued=0    ignored=0
[ INFO  ] Pre-check successful for install.
[ INFO  ] Tip : ./spectrumscale install
```

Deploying IBM Storage Scale

After the Install precheck is successful, deploy IBM Storage Scale. Ensure that the required operating system packages are installed before starting the deployment.

Installing IBM Storage Scale binary files and creating a cluster

This step initiates the deployment of the IBM Storage Scale cluster. Perform the following steps to deploy the IBM Storage Scale cluster:

1. Navigate to IBM Storage Scale Installer on IBM Storage Scale node#1 by using the following command:

   ```
   cd /usr/lpp/mmfs/5.2.3.4/ansible-toolkit/
   ```

2. Install the IBM Storage Scale binaries and create the cluster by using the following command:

   ```
   ./spectrumscale install
   ```

   Example 25 provides an example of the expected output.

Example 25 Output of install command

```
[root@Scalenode1 ansible-toolkit]# ./spectrumscale install
[ INFO  ] Checking cluster defnition json file inventory is
ansible/vars/scale_clusterdefinition.json
[ INFO  ] Logging to file:
/usr/lpp/mmfs/5.2.3.4/ansible-toolkit/logs/INSTALL-18-10-2025_02:32:24.log
[ INFO  ] Validating configuration
[ WARN  ] Ensure that base OS repositories are configured and enabled so that
package dependencies can be satisfied during installation.
>>>>>>>>>>>>>>>>>>>>>>>>>Output is trimmed<<<<<<<<<<<<<<<<<<<<<<<<<<<<
[ INFO  ] PLAY RECAP
*********************************************************************
[ INFO  ] localhost                  : ok=4    changed=1    unreachable=0
failed=0    skipped=2    rescued=0    ignored=0
[ INFO  ] scalenode1.local           : ok=163  changed=38   unreachable=0
failed=0    skipped=403  rescued=0    ignored=0
[ INFO  ] scalenode2.local           : ok=84   changed=14   unreachable=0
failed=0    skipped=189  rescued=0    ignored=0
[ INFO  ] scalenode3.local           : ok=80   changed=14   unreachable=0
failed=0    skipped=193  rescued=0    ignored=0
[ INFO  ] Destroying repository...
[ INFO  ] Checking for a successful install
[ INFO  ] Checking state of GPFS
[ INFO  ] Checking state of GPFS on all nodes
[ INFO  ] GPFS active on all nodes
[ INFO  ] GPFS ACTIVE
[ INFO  ] Checking state of NSDs
[ INFO  ] NSDs ACTIVE
[ INFO  ] Checking state of Performance Monitoring
[ INFO  ] Running Performance Monitoring post-install checks
[ INFO  ] pmcollector running on all nodes
[ INFO  ] pmsensors running on all nodes
[ INFO  ] Performance Monitoring ACTIVE
[ INFO  ] Checking state of GUI
[ INFO  ] Running Graphical User Interface  post-install checks
[ INFO  ] Graphical User Interface running on all GUI servers
```

```
[ INFO  ] Enter one of the following addresses into a web browser to access the
Graphical User Interface: scalenode1.local
[ INFO  ] GUI ACTIVE
[ INFO  ] SUCCESS
[ INFO  ] All services running
[ INFO  ] StanzaFile and NodeDesc file for NSD, filesystem, and cluster setup
have been saved to /usr/lpp/mmfs folder on node: scalenode1.local
[ INFO  ] Installation successful. 3 GPFS nodes active in cluster
scalenode1.local. Completed in 11 minutes 25 seconds.
[ INFO  ] Tip :If all node designations and any required protocol
configurations are complete, proceed to check the deploy
configuration:./spectrumscale deploy --precheck
```

Verifying cluster deployment status

Perform the following steps to verify the cluster deployment status:

1. Edit the `.bashrc` file on IBM Storage Scale node#1 and add the path to the IBM Storage Scale binary file:

   ```
   export PATH=$PATH:/usr/lpp/mmfs/bin
   ```

2. Logout and login again on IBM Storage Scale node#1.
3. Verify the status of the IBM Storage Scale cluster by using the following command:

   ```
   mmgetstate -a
   ```

   Example 26 provides an example of the expected output.

   Example 26 Output of mmgetstate command

   ```
   [root@Scalenode1 ansible-toolkit]# mmgetstate -a
    Node number  Node name   GPFS state
   -------------------------------------
              1  scalenode1  active
              2  scalenode2  active
              3  scalenode3  active
   ```

4. Verify the file system status by using the following command:

   ```
   mmlsfs all
   ```

   Example 27 provides an example of the expected output.

   Example 27 Output of mmlsfs all command

   ```
   [root@Scalenode1 ansible-toolkit]# mmlsfs all
   File system attributes for /dev/FS1:
   ====================================
   flag                value                    description
   ------------------ ------------------------ -----------------------------------
   >>>>>>>>>>>>>>>>>>>>>>>>>>>Output is trimmed <<<<<<<<<<<<<<<<<<<<<<<<<<<
   -T                 /ibm/FS1                 Default mount point
   --mount-priority   0                        Mount priority
   File system attributes for /dev/cesSharedRoot:
   ==============================================
   flag                value                    description
   ------------------ ------------------------ -----------------------------------
   >>>>>>>>>>>>>>>>>>>>>>>>>>>Output is trimmed <<<<<<<<<<<<<<<<<<<<<<<<<<<
   T                 /ibm/cesSharedRoot       Default mount point
   ```

```
--mount-priority   0                          Mount priority
```

Configuring protocols in IBM Storage Scale cluster

Protocol services in IBM Storage Scale cluster enables NFS protocols. Protocol services are also called as Cluster Export Services (CES).

The following high-level steps outline the process for configuring protocols in an IBM Storage Scale Cluster:

1. Create the configuration file for configuring protocols in the IBM Storage Scale cluster.
2. Run protocols configuration prechecks.
3. Deploy protocols in the IBM Storage Scale cluster.

Creating a configuration file for configuring protocols

Perform the following steps to create a configuration file for configuring protocols in the IBM Storage Scale cluster.

1. Navigate to the IBM Storage Scale Installer path on IBM Storage Scale node#1 by using the following command:

   ```
   cd /usr/lpp/mmfs/5.2.3.4/ansible-toolkit/
   ```

2. Configure the protocols by using the following command:

   ```
   ./spectrumscale config protocols -f cesSharedRoot -m /ibm/cesSharedRoot
   ```

   Example 28 provides an example of the expected output.

   Example 28 Output of config protocols -f command

   ```
   [root@Scalenode1 ansible-toolkit]# ./spectrumscale config protocols -f
   cesSharedRoot -m /ibm/cesSharedRoot
   [ INFO  ] Setting filesystem to cesSharedRoot
   [ INFO  ] Setting mountpoint to /ibm/cesSharedRoot
   [ INFO  ] Tip :Enable NFS, S3, SMB or HDFS protocols as
   appropriate:./spectrumscale enable nfs|s3|smb|hdfs
   ```

3. Define protocol IP addresses by using the following command:

   ```
   ./spectrumscale config protocols -e <IP1>,<IP2>.<IP3>
   ```

   Example 29 provides an example of the expected output.

   Example 29 Output of the config protocols -e command

   ```
   [root@Scalenode1 ansible-toolkit]# ./spectrumscale config protocols -e
   192.168.69.114,192.168.69.122,192.168.69.123
   [ INFO  ] Setting export_ip_pool to ['192.168.69.114', '192.168.69.122',
   '192.168.69.123']
   [ INFO  ] Tip :Enable NFS, S3, SMB or HDFS protocols as
   appropriate:./spectrumscale enable nfs|s3|smb|hdfs
   ```

4. Enable NFS service on protocol nodes by using the following command:

   ```
   ./spectrumscale enable nfs
   ```

   Example 30 on page 33 provides an example of the expected output.

Example 30 Output of enable nfs command

```
[root@Scalenode1 ansible-toolkit]# ./spectrumscale enable nfs
[ INFO  ] Enabling NFS on all protocol nodes.
[ INFO  ] Tip :If all node designations and any required protocol
configurations are complete, proceed to check the installation
configuration:./spectrumscale deploy –precheck
```

5. Update `/etc/hosts` file with CES IP address in all three IBM Storage Scale Nodes and add the following lines:

```
192.168.69.114  scale-ces1.local       scale-ces1
192.168.69.122  scale-ces2.local       scale-ces2
192.168.69.123  scale-ces3.local       scale-ces3
```

Note: If a DNS service is running in the environment, a DNS record can be created instead of adding the editing the file.

All three IP addresses can be bound to a single record in DNS using round-robin load balancing.

Protocol configuration prechecks

Perform the following steps to run prechecks for protocol configuration:

1. Navigate to the IBM Storage Scale Installer path on IBM Storage Scale node#1 by using the following command:

```
cd /usr/lpp/mmfs/5.2.3.4/ansible-toolkit/
```

2. Run the protocol configuration precheck by using the following command:

```
./spectrumscale deploy --precheck
```

Example 31 provides an example of the expected output.

Example 31 Output of deploy precheck command

```
./spectrumscale deploy --precheck
[ INFO  ] Logging to file:
/usr/lpp/mmfs/5.2.3.4/ansible-toolkit/logs/DEPLOY-PRECHECK-18-10-2025_02:51:31.
log
[ INFO  ] Validating configuration
[ WARN  ] Ensure that base OS repositories are configured and enabled so that
package dependencies can be satisfied during installation.
>>>>>>>>>>>>>>>>>>>>>>>>>>Output is trimmed <<<<<<<<<<<<<<<<<<<<<<<<<<
[ INFO  ] PLAY RECAP
*********************************************************************
[ INFO  ] localhost                  : ok=4    changed=1    unreachable=0
failed=0    skipped=2    rescued=0    ignored=0
[ INFO  ] scalenode1.local           : ok=10   changed=0    unreachable=0
failed=0    skipped=35   rescued=0    ignored=0
[ INFO  ] scalenode2.local           : ok=10   changed=0    unreachable=0
failed=0    skipped=27   rescued=0    ignored=0
[ INFO  ] scalenode3.local           : ok=10   changed=0    unreachable=0
failed=0    skipped=27   rescued=0    ignored=0
[ INFO  ] Pre-check successful for deploy.
[ INFO  ] Tip : ./spectrumscale deploy
```

Deploying protocols in IBM Storage Scale cluster

Perform the following steps to deploy protocols in an IBM Storage Scale cluster.

1. Navigate to IBM Storage Scale Installer path on IBM Storage Scale node#1 using the following command:

   ```
   cd /usr/lpp/mmfs/5.2.3.4/ansible-toolkit/
   ```

2. Deploy the protocols by using the following command:

   ```
   ./spectrumscale deploy
   ```

   Example 32 provides an example of the expected output.

Example 32 Output of deploy command

```
./spectrumscale deploy
[ INFO  ] Logging to file:
/usr/lpp/mmfs/5.2.3.4/ansible-toolkit/logs/DEPLOY-18-10-2025_02:53:12.log
[ INFO  ] Validating configuration
[ WARN  ] Ensure that base OS repositories are configured and enabled so that
package dependencies can be satisfied during installation.
>>>>>>>>>>>>>>>>>>>>>>>>>>>Output is trimmed<<<<<<<<<<<<<<<<<<<<<<<<<<<
[ INFO  ] PLAY RECAP
*********************************************************************
[ INFO  ] localhost                  : ok=4    changed=1    unreachable=0
failed=0    skipped=2    rescued=0    ignored=0
[ INFO  ] scalenode1.local           : ok=135  changed=31   unreachable=0
failed=0    skipped=300  rescued=0    ignored=0
[ INFO  ] scalenode2.local           : ok=82   changed=15   unreachable=0
failed=0    skipped=173  rescued=0    ignored=0
[ INFO  ] scalenode3.local           : ok=80   changed=15   unreachable=0
failed=0    skipped=175  rescued=0    ignored=0
>>>>>>>>>>>>>>>>>>>>>>>>>>>Output is trimmed<<<<<<<<<<<<<<<<<<<<<<<<<<<
[ INFO  ] Enter one of the following addresses into a web browser to access the
Graphical User Interface: scalenode1.local
[ INFO  ] GUI ACTIVE
[ INFO  ] SUCCESS
[ INFO  ] All services running
[ INFO  ] StanzaFile and NodeDesc file for NSD, filesystem, and cluster setup
have been saved to /usr/lpp/mmfs folder on node: scalenode1.local
[ INFO  ] Successfully installed and configured protocols. 3 protocol nodes
were enabled. Components installed: file system, Cluster Export Services, NFS,
Performance Monitoring, GUI, FILE AUDIT LOGGING. It took 5 minutes 19 seconds.
```

Verifying system health after protocol deployment

Perform the following steps to verify the system health after protocol deployment:

1. Verify IBM Storage Scale cluster status by using the following command:

   ```
   mmgetstate -a
   ```

   Example 33 on page 35 provides an example of the expected output.

Example 33 Output of mmgetstate command

```
[root@Scalenode1 ansible-toolkit]# mmgetstate -a
 Node number  Node name    GPFS state
-------------------------------------
           1  scalenode1  active
           2  scalenode2  active
           3  scalenode3  active
```

2. Verify CES IP assignment and distribution by using the following command:

   `mmces address list`

   Example 34 provides an example of the expected output.

Example 34 Output of mmces command

```
[root@Scalenode1 ansible-toolkit]# mmces address list
Address       Node               Ces Group   Attributes
------------ ------------------ ----------- ------------
192.168.69.114   scalenode1.local   none        none
192.168.69.122   scalenode2.local   none        none
192.168.69.123   scalenode3.local   none        none
```

3. Verify individual node health by using the following command:

   `mmhealth node show -N scalenode1,scalenode2,scalenode3`

   Example 35 provides an example of the expected output.

Example 35 Output of mmhealth command

```
[root@Scalenode1 ansible-toolkit]# mmhealth node show -N
scalenode1,scalenode2,scalenode3
Node name:      scalenode1.local
Node status:    HEALTHY
Status Change:  56 min. ago
Component      Status        Status Change     Reasons & Notices
--------------------------------------------------------------------------------
GPFS           HEALTHY       56 min. ago       -
NETWORK        HEALTHY       58 min. ago       -
FILESYSTEM     HEALTHY       57 min. ago       -
DISK           HEALTHY       47 min. ago       -
PERFMON        HEALTHY       45 min. ago       -
THRESHOLD      HEALTHY       44 min. ago       -
Node name:      scalenode2.local
Node status:    TIPS
Status Change:  56 min. ago
Component      Status        Status Change     Reasons & Notices
--------------------------------------------------------------------------------
GPFS           TIPS          56 min. ago       callhome_not_enabled
NETWORK        HEALTHY       58 min. ago       -
FILESYSTEM     HEALTHY       57 min. ago       -
DISK           HEALTHY       47 min. ago       -
CES            TIPS          5 min. ago
nfs_sensors_not_configured(NFSIO)
CESIP          HEALTHY       6 min. ago        -
FILESYSMGR     HEALTHY       46 min. ago       -
PERFMON        HEALTHY       45 min. ago       -
```

```
THRESHOLD      HEALTHY       46 min. ago       -
Node name:      scalenode3.local
Node status:    HEALTHY
Status Change:  56 min. ago
Component      Status        Status Change     Reasons & Notices
--------------------------------------------------------------------------------
GPFS           HEALTHY       56 min. ago       -
NETWORK        HEALTHY       58 min. ago       -
FILESYSTEM     HEALTHY       57 min. ago       -
DISK           HEALTHY       47 min. ago       -
CES            HEALTHY       5 min. ago        -
FILESYSMGR     HEALTHY       46 min. ago       -
PERFMON        HEALTHY       45 min. ago       -
THRESHOLD      HEALTHY       46 min. ago       -
[root@Scalenode1 ansible-toolkit]# mmces state show NFS -N
scalenode2,scalenode3
NODE                NFS
----------------- -----
scalenode2.local   TIPS
scalenode3.local   HEALTHY
[root@Scalenode1 ansible-toolkit]#
```

Configuring authentication for NFS exports or shares

NFS exports operate over the network, making it essential to configure an authentication mechanism for client access. IBM Storage Scale provides an option to configure authentication.

To configure authentication by using the **mmuserauth** command, see Configuring authentication and ID mapping for file access.

Use the following high-level steps to create NFS exports:

1. Create fileset
2. Link fileset
3. Create and export NFS export and share

Creating a fileset

A fileset is a subtree of a file system namespace that acts similar to an independent file system. File sets provide a means of partitioning the file system to allow administrative operations at a finer granularity than the entire file system. Perform the following steps to create a fileset:

1. Create a fileset by using the following command:

   ```
   mmcrfileset FS1 testfset1 --inode-space=new
   ```

   The following example shows the expected output:

   ```
   [root@Scalenode1 ~]# mmcrfileset FS1 testfset1 --inode-space=new
   Fileset testfset1 created with id 1 root inode 524291.
   ```

2. Verify the status of the fileset by using the following command:

   ```
   mmlsfileset FS1
   ```

   Example 36 on page 37 provides an example of the expected output.

Example 36 Output of mmlsfileset command

```
[root@Scalenode1 ~]# mmlsfileset FS1
Filesets in file system 'FS1':
Name                     Status    Path
root                     Linked    /ibm/FS1
testfset1                Unlinked  --
```

Linking a fileset

After the fileset is created, link the fileset to a directory. Perform the following steps to link the file set:

Link the file set by using the following command:

```
mmlinkfileset FS1 testfset1 -J /ibm/FS1/testfset1
```

Example 37 provides an example of the expected output.

Example 37 Output of mmlinkfileset command

```
[root@Scalenode1 ~]#  mmlinkfileset FS1 testfset1 -J /ibm/FS1/testfset1
Fileset testfset1 linked at /ibm/FS1/testfset1
[root@Scalenode1 ~]# mmlsfileset FS1
Filesets in file system 'FS1':
Name                     Status    Path
root                     Linked    /ibm/FS1
testfset1                Linked    /ibm/FS1/testfset1
```

Creating and exporting an NFS export or share

An NFS export must be created on top of a file set. After the file set is created and linked, you can create the NFS export.

Create the NFS export Using the following command:

```
mmnfs export add /ibm/FS1/testfset1 --client
```

The following example shows an expected output:

```
[root@Scalenode1 ~]# mmnfs export add /ibm/FS1/testfset1 --client
"192.168.111.99/32(Access_Type=RW,Protocols=3:4)"
mmnfs: The NFS export was created successfully
mmnfs: Restarting NFS services.
[root@Scalenode1 ~]#
```

You can mount NFS export on the NFS client.

Creating an IBM Storage Scale GUI user

Define an IBM Storage Scale GUI user and password by using the following command:

```
/usr/lpp/mmfs/gui/cli/mkuser admin -g SecurityAdmin
```

The following example shows an expected output:

```
[root@Scalenode1 ansible-toolkit]# /usr/lpp/mmfs/gui/cli/mkuser admin -g
SecurityAdmin
EFSSG1007A Enter password for User :
```

```
EFSSG0225I Repeat the password:
EFSSG0019I The user admin has been successfully created.
EFSSG1000I The command completed successfully.
```

After creating the user ID, access the IBM Storage Scale GUI by using the following steps:

1. Open web browser and enter below address

   `https://<IBM Storage Scale Server#1 IP address>`

2. Enter the previously created user ID and password.

Using IBM Storage Scale in OpenShift Container Platform

IBM Storage Scale Container Native enables the deployment of IBM Storage Scale within a Red Hat OpenShift Container Platform and supported Kubernetes environment. This deployment can leverage SAN-attached storage, such as IBM FlashSystem, to provide a persistent data store accessible as a local file system.

Containerized applications can use this data store through the IBM Storage Scale CSI driver by provisioning Persistent Volumes (PVs). This solution delivers seamless integration with IBM FlashSystem, offering container-native POSIX file storage access that ensures fast, scalable, and reliable persistent storage for containerized workloads. The added benefits of using IBM FlashSystem for Ransomware Detection as listed in “Ransomware threat detection by using IBM Storage Insights Pro” on page 6 and, for example, FlashSystem grid architecture, policy-based replication, and AIOps capability helps further differentiate this deployment for modernized workloads.

This is a well-suited deployment for containerized workloads that benefit from persistent volumes provisioned over a filesystem where there is need for POSIX semantics, higher scalability for number of volumes, and so on.

For more information, see IBM Storage Scale Container Interface Drive.

Performance of high-speed file and block access

Performance is a critical consideration when selecting a storage solution for modern IT infrastructure. To validate the capabilities of the proposed high-performance block and file storage solution, comprehensive performance testing was conducted to determine the maximum achievable throughput under controlled conditions.

System Performance Overview

The IBM FlashSystem, when fully populated, is capable of delivering up to 100 GB/s of read throughput. See “Sizing and limitations” on page 3 to determine performance requirements. For detailed configuration options and official performance benchmarks for the IBM FlashSystem 9500, see IBM Storage FlashSystem 9500.

When using IBM Storage Scale level, this solution can also achieve throughput of up to 100 GB/s, depending on configuration and workload characteristics. It is designed to support both file and block workloads concurrently, making appropriate workload sizing and balancing essential for achieving optimal performance.

Key Performance Considerations

When planning or evaluating performance expectations for this solution, consider the following factors:

- Hardware and environment dependency: Actual performance results can vary based on the specific hardware configuration and overall system environment.
- Network performance: Because IBM Storage Scale uses Ethernet for data communication, then network bandwidth, latency, and overall Ethernet performance are crucial for sustained throughput.
- Workload segregation: You can configure separate pools (child or parent) within the FlashSystem to isolate workloads and improve predictability.
- Quality of Service (QoS): To maintain consistent performance and prevent race conditions between file and block workloads, you can configure QoS policies at the FlashSystem pool level for better workload management and control.

Summary

This Redbooks publication provides user step by step instruction to deploy a file-storage and block-storage access solution by using IBM Storage Scale and IBM FlashSystem storage. The purpose of this publication is to ease the deployment for the implementation teams. IBM business partners and clients with limited skills of the mentioned technologies can also deploy the solution by using this publication.

Online resources

For more information, see the following IBM Documentation:

- IBM Storage FlashSystem 9500, 9200 and 9100
- IBM Storage Scale
- IBM Storage Scale Container Native - Direct attach use case

Help from IBM

IBM Support and downloads

ibm.com/support

Services from IBM Consulting®

ibm.com/services

IBM Training

ibm.com/training

REDP-5764-00

ISBN 0738462357

Printed in U.S.A.