EXOS™

White Paper

# IBM STORAGE SCALE INTEGRATION GUIDE

## Deploying IBM Storage Scale and Seagate Exos X and Exos CORVAULT Storage

SEAGATE

# CONTENTS

## Introduction

The purpose of this document is to provide a step-by-step guide to deploy and implement IBM Storage Scale GPFS on Seagate® Exos X® 5U84 and Exos CORVAULT™ systems. In our example we use the Exos X 5U84 array in conjunction with CentOS to validate the Storage Scale implementation and to provide a reference point for Seagate field teams and Seagate partners in their customer proof of concept (POC) Storage Scale engagement.


The entire procedure focuses on the following areas:

- Seagate storage configuration and performance optimization
- Deployment of a Storage Scale server
- Starting the Storage Scale cluster and mounting the Storage Scale file system

This document is not intended to replace any existing Storage Scale and/or Exos X series user reference guides or other documentation.

## Scope

IBM Storage Scale is a feature rich, parallel file system. Evolving from IBM GPFS, the software includes several software modules, each delivering a specific function. The following is a brief description of these software features:

- IBM Storage Scale Shared Data Access
- IBM Storage Scale File System Replication
- IBM Storage Scale CES (Cluster Export Service that includes Samba, NFS and Object support)
- IBM Storage Scale AFM (policy-drive data placement)
- IBM Storage Scale data protection
- IBM Storage Scale data encryption
- IBM Storage Scale HPO (High Performance Object) built on data access services

A complete full-scale feature and implementation assessment on Seagate storage is out of the scope of this document. This guide attempts to cover the procedures to follow to deploy a HA Storage Scale cluster over the Red Hat server platform to the point where the data is sharable at the Storage Scale mount point.

Storage Scale performance optimization is generally excluded in the validation procedures. However, we include a performance optimization discussion on Seagate storage referred to as “raw device.”

All procedures in this document focus on the functional aspects of the Seagate storage systems interfacing with Storage Scale.

# Lab Environment

## Hardware

In the test environment there are test elements that consist of Seagate storage, Red Hat host servers, and SAS direct connection between the Storage Scale host server and Seagate storage.

DAS topology is one of Storage Scale's common deployment options. For use cases where scale-out is not a consideration, we see DAS offers a much simpler approach for Storage Scale quick validation.

The hardware required for DAS deployment is listed as follows. However, we recommend Storage Scale users check the IBM Storage Scale FAQ for the hardware and software support matrix at https://www.ibm.com/docs/en/spectrum-scale. The hardware required is listed as follows:

| Storage | | |
|---|---|---|
| **Quantity** | **Description** | **Model** |
| 1 | Exos X 5U84 | 4865 |
| 84 | 14TB SATA | |

| Host Server | | | |
|---|---|---|---|
| **Quantity** | **Description** | **Model** | **CPU** |
| 4 | SuperMicro | SYS-5019P-WTR | Intel 5320 CPU @ 2.20GH |
| 4 | LSI SAS 2-ports 8GB/s HBA | 3180 | |
| 256 | GB of physical memory on each host server | | |
| 2 | HD Mini-SAS to HD Mini-SAS 12G cable/each serve | | |

Note: Among four host servers, two are dedicated to host Storage Scale NSD (Network Shared Disks) as primary and secondary nodes, and two host servers act as NFS clients for NFS access validation.

## Software

The software package and version information are provided below only for reference since any elements in the deployed test hardware may require dynamic software changes or updates to compatibly run with Storage Scale.

Note: CentOS is not a supported OS for the Storage Scale environment. It is used here only as an example.

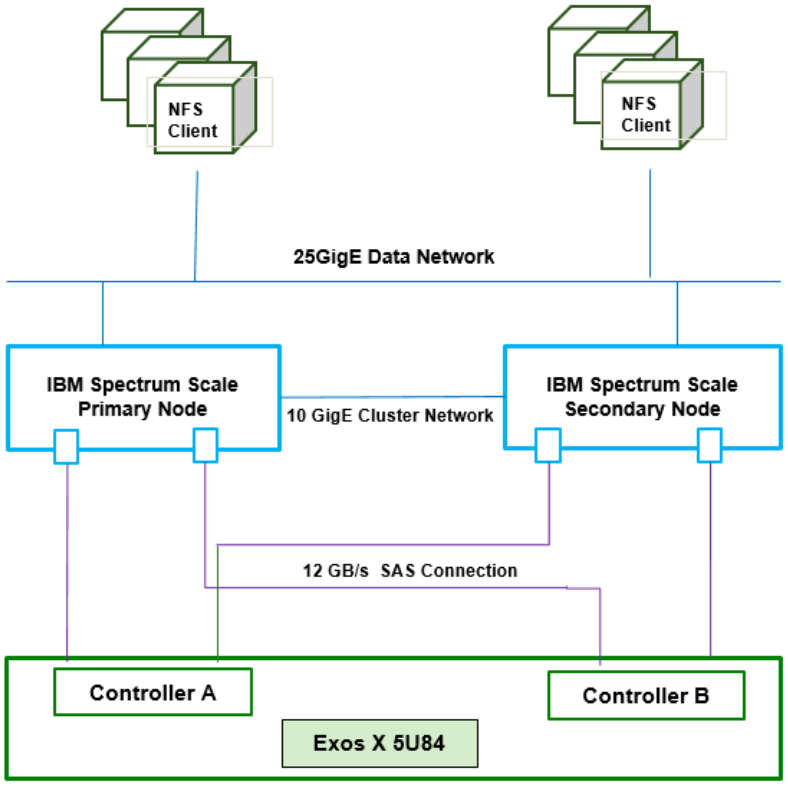
| Host Server | Software Version |
|---|---|
| NSD host OS | CentOS Linux release 8.4.2105 |
| Host OS Kernel | Linux 4.18.0-305.12.1.el8_4.x86_64 |
| SAS HBA driver | mpt3sas |
| SAS HBA firmware | 09.00.100.00 |
| IBM Spectrum Scale | Spectrum_Scale_Data_Management-5.1.3.0-x86_64-Linux |
| CES host server | Spectrum_Scale_Data_Management-5.1.3.0-x86_64-Linux |

| Storage | Firmware |
|---|---|
| 5U84 CNC | I200R001 |
| 18TB HDD | SEAGATE  E002 |

## Lab Connection Topology

The connection between Storage Scale hosts and the Seagate storage system is architected to include two types of connections for the base test in first phase and for scalability validation in the second phase. In phase one, the base test Storage Scale NSD host servers are connected to the storage via DAS with HD miniSAS cables. Each host has a single 8Gbps dual-port SAS HBA that is cross-connected to the Exos X storage system controllers. This test environment is not an ideal production connection primarily because the storage resources are not shared for redundancy, performance, and scalability. However, we feel this type of connection is sufficient for a POC.

The following diagram shows the connection topologies. The NFS client is not included in the diagram as the client can be anywhere on the network that Storage Scale has file level protocol services.



# Seagate Storage Configuration

Storage Scale storage is built on NSD and is used to host Storage Scale metadata and user data. You can create NSD on the base of single-path disk devices or multipath-capable disk devices.

You must configure the storage resources properly before you create Storage Scale NSDs. The storage configuration procedure can be done either through the web user interface (UI), or manually through the SSH CLI on Seagate storage.

For a better user experience, we recommend that you configure the storage using the web UI. However, if you intend to build interleaved LUNs on Seagate storage to accommodate Storage Scale NSDs, we recommend you use the CLI since, as of the release of this document, creating an interleaved LUN is still being tested.

## User On-Boarding

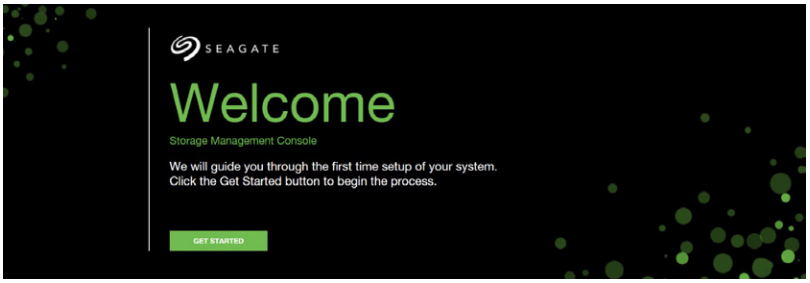
For simplicity, we skip the Seagate storage initial and baseline configuration via serial console port and focus on the storage's new user onboarding process until we get to the point where the storage resources are made available to the Storage Scale hosts.

Storage configuration includes four major steps consisting of storage system configuration, disk pool and disk group configuration, storage resource provisioning, and storage resource exporting.

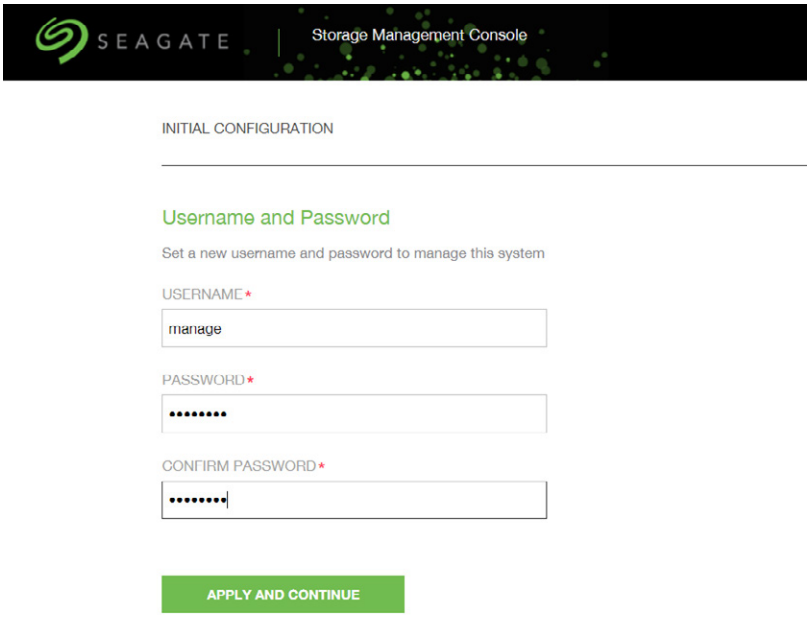
The following steps will walk you through the process to bring storage resources online.

Note: We use LUNs and volumes interchangeably to refer to Seagate storage resources.

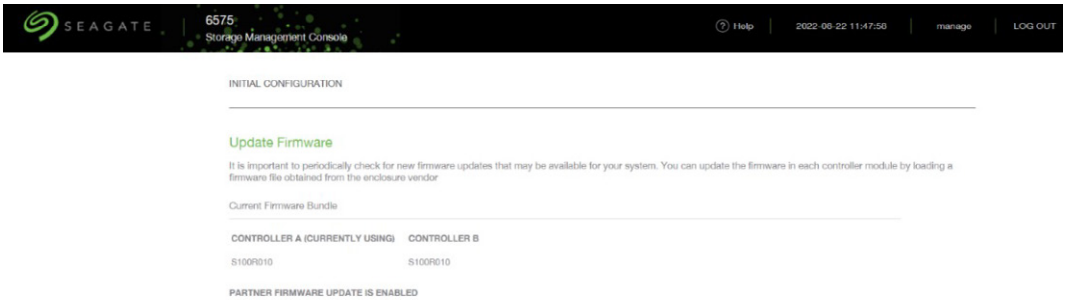
1. To begin user onboarding, type `https://<IP_address_of_the_storage>`.



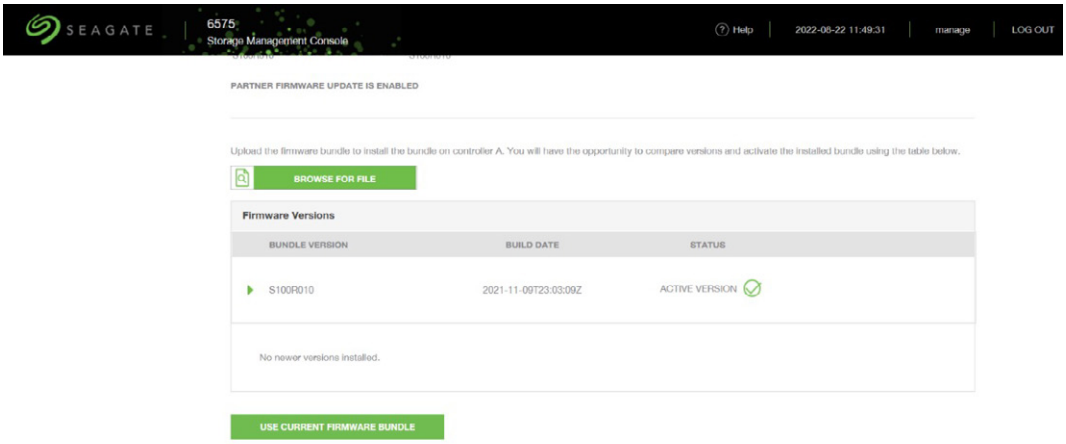

2. Create your admin user ID and password.



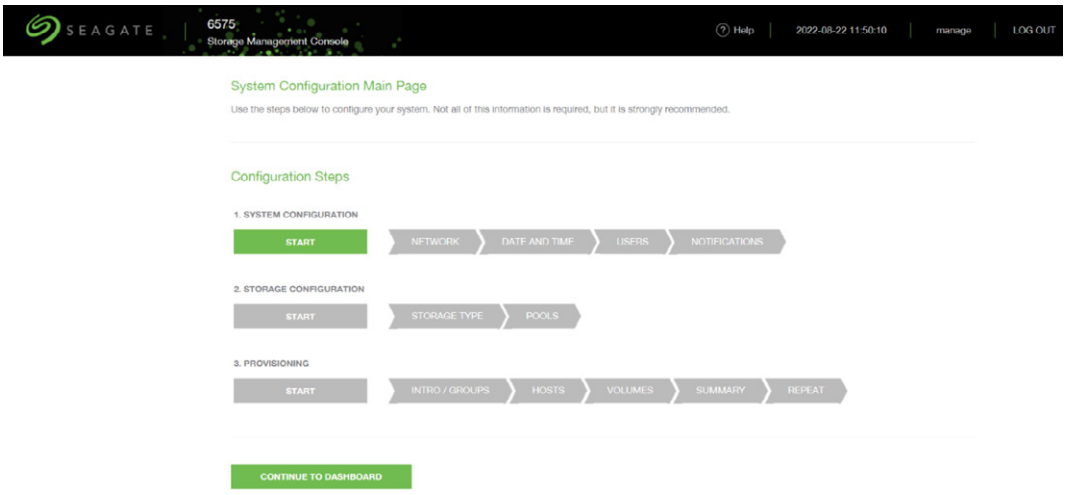
3. Accept the preloaded firmware unless you are advised to do otherwise.



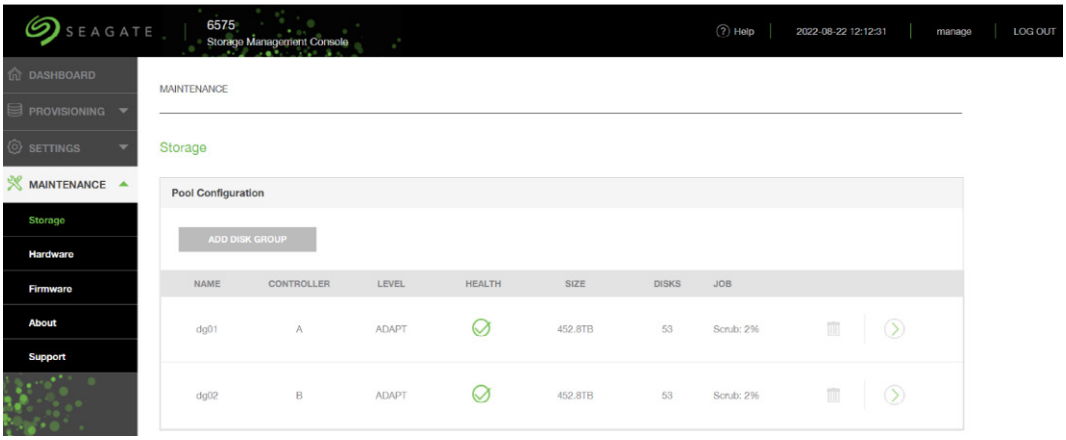

4. Follow the prompts to configure the storage system.



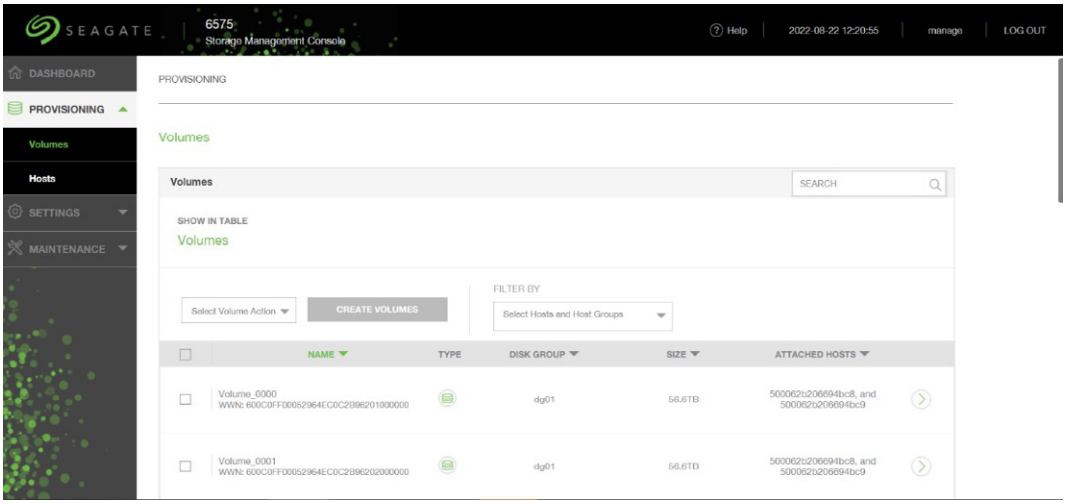

In this configuration, the user will need to go through the steps listed for each of three major configurations. Alternatively, the user can skip some of the steps to have a quick setup in order to go directly to the storage resource management configuration below.



5. Create disk groups to include the disk drives. You must create at least one disk group in the disk pool.

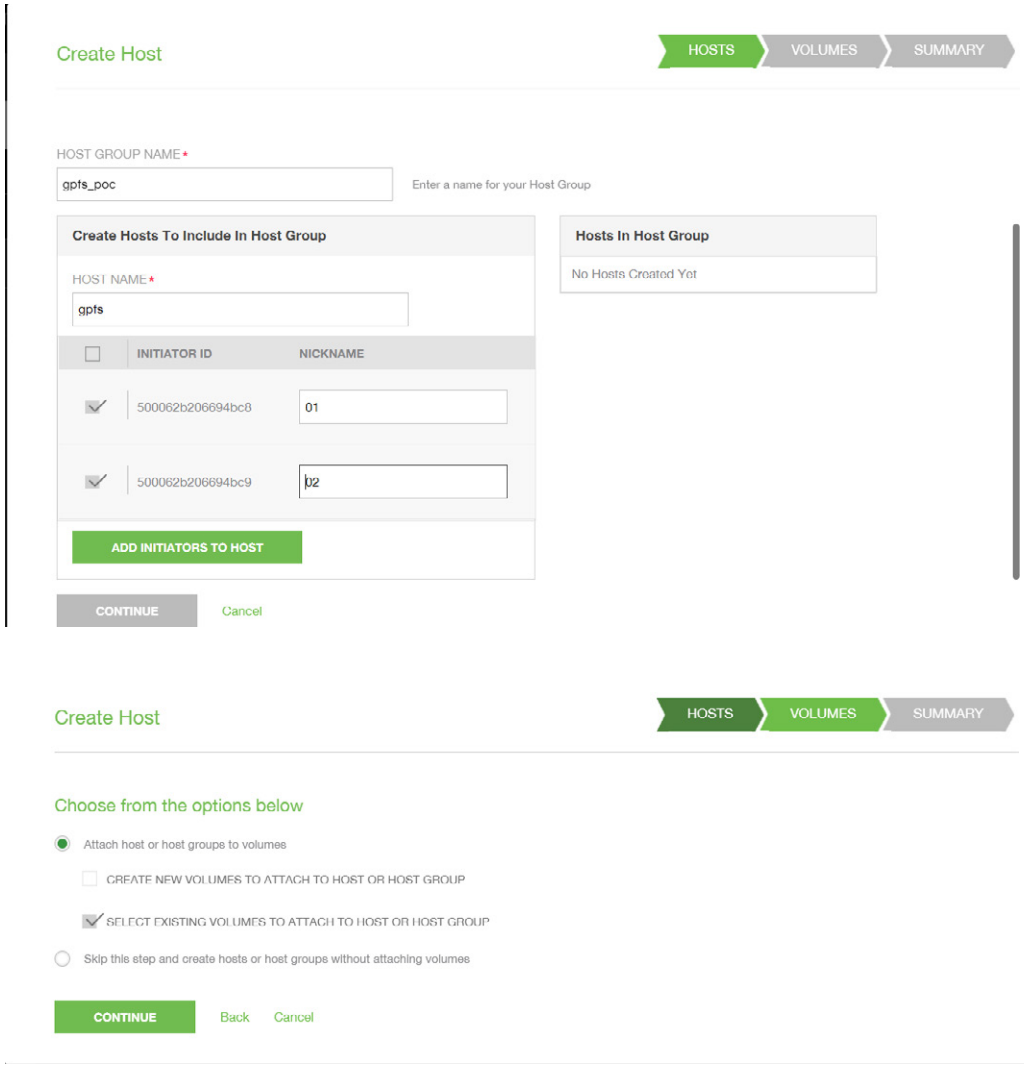



6. Create volumes on top of disk groups.



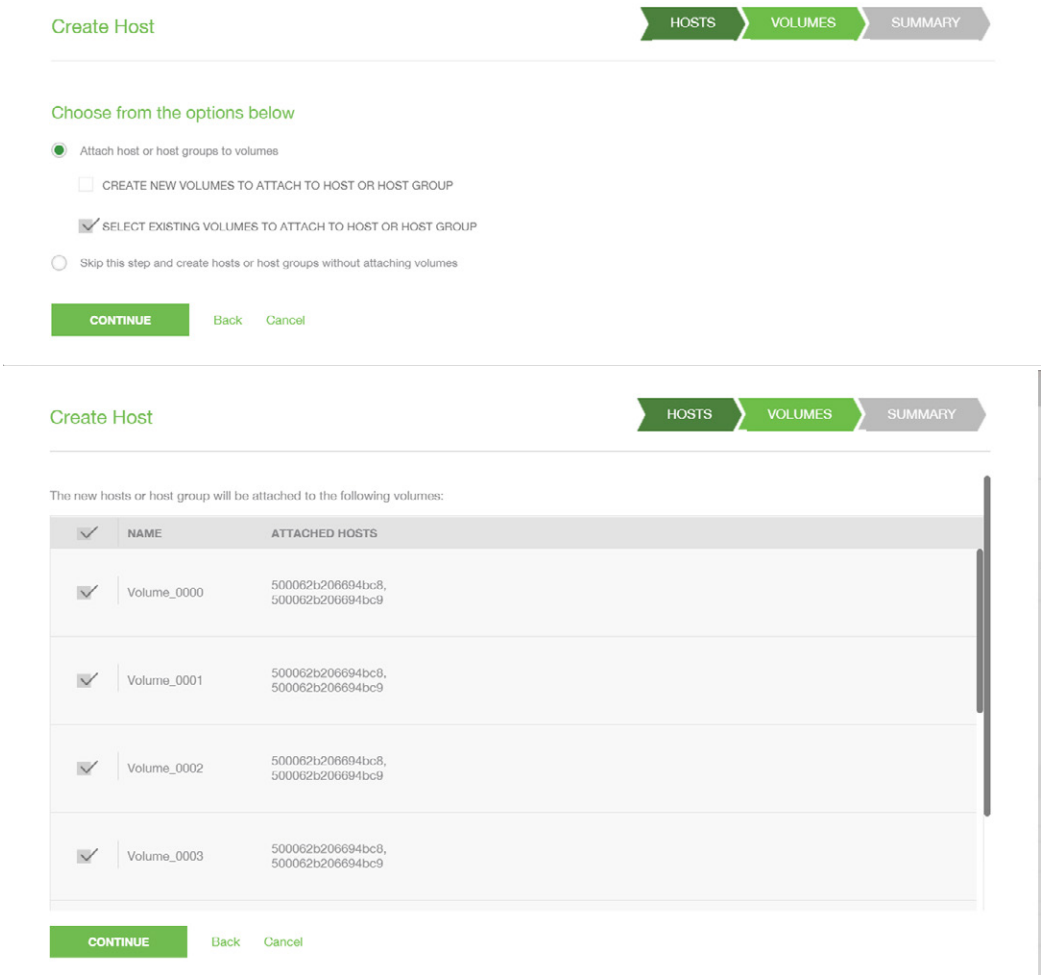

Note: There are best practices to follow when creating volumes, as they can be created with parameters specific to distinct user applications to ensure optimal performance and capacity.

7. Create a host group to include the Storage Scale host as initiators as shown in the following example.

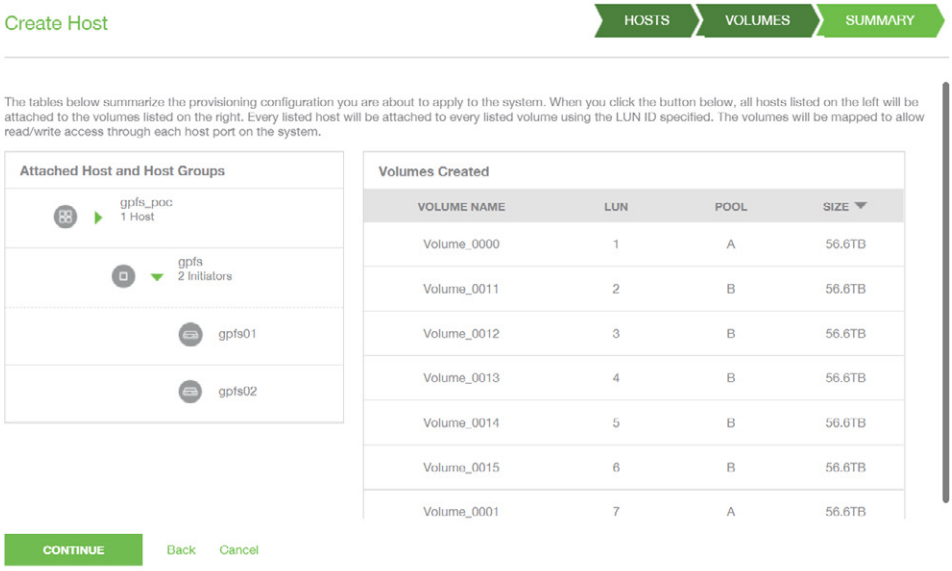


8. Create maps to connect the Storage Scale hosts to the storage volumes.

Note: The volumes or LUNs should be cross-mapped to obtain host and controller level storage HA or redundancy.




9. Once complete, the storage is ready and a summary of storage creation and configuration displays. The user can log into the host to verify that it can see the Seagate storage target.



## Storage Connection Verification

On a direct attached host, the following CLI commands will help you identify the storage targets that are visible to the host(s).

**A. Identify the HBA Installation.**

From the Storage Scale host, verify that the HBAs and HBA drivers are installed correctly. The mpt3sas command in the example indicates the host can see the HBA.

```
[root@sm247 ~]# lsscsi --host
[0]    mpt3sas
[1]    ahci
[2]    ahci
[3]    ahci
[4]    ahci
[5]    ahci
[6]    ahci
[7]    ahci
[8]    ahci
[9]    ahci
[10]    ahci
```

**B. Verify disk/enclosure connections.**

Use the following CLI commands to verify that the disk drive, storage controllers, and disk enclosures are correctly reported to the Storage Scale hosts.

```
[root@sm247 ~]# lsscsi |grep -i Seagate
[0:0:0:0]    disk    SEAGATE  4006             I200  /dev/sdc
[0:0:0:1]    disk    SEAGATE  4006             I200  /dev/sdd
[0:0:0:2]    disk    SEAGATE  4006             I200  /dev/sde
[0:0:1:0]    disk    SEAGATE  4006             I200  /dev/sdf
[0:0:1:1]    disk    SEAGATE  4006             I200  /dev/sdg
[0:0:1:2]    disk    SEAGATE  4006             I200  /dev/sdh
```

```
[root@sm247 ~]# lsscsi -d
[0:0:0:0]    disk    SEAGATE  4006             I200  /dev/sdc [8:32]
[0:0:0:1]    disk    SEAGATE  4006             I200  /dev/sdd [8:48]
[0:0:0:2]    disk    SEAGATE  4006             I200  /dev/sde [8:64]
[0:0:1:0]    disk    SEAGATE  4006             I200  /dev/sdf [8:80]
[0:0:1:1]    disk    SEAGATE  4006             I200  /dev/sdg [8:96]
[0:0:1:2]    disk    SEAGATE  4006             I200  /dev/sdh [8:112]
[3:0:0:0]    disk    ATA      ST1000NM0055-1V4 TN05  /dev/sda [8:0]
[4:0:0:0]    disk    ATA      ST1000NM0055-1V4 TN05  /dev/sdb [8:16]
```

The disk drive capacity can also be listed through the host-side SCSI device details, as shown.

```
[root@sm247 ~]# lsscsi -s
[0:0:0:0]    disk    SEAGATE  4006             I200  /dev/sdc   3.19TB
[0:0:0:1]    disk    SEAGATE  4006             I200  /dev/sdd    265TB
[0:0:0:2]    disk    SEAGATE  4006             I200  /dev/sde    266TB
[0:0:1:0]    disk    SEAGATE  4006             I200  /dev/sdf   3.19TB
[0:0:1:1]    disk    SEAGATE  4006             I200  /dev/sdg    265TB
[0:0:1:2]    disk    SEAGATE  4006             I200  /dev/sdh    266TB
```

At this point, the storage onboarding process is complete.
More Seagate documentation can be found at
https://www.seagate.com/support/raid-storage-systems/corvault.

# Storage Scale Host Software Installation

This section describes the processes to prepare the host OS to install Storage Scale.

## Multipath Consideration

If the storage device is multipath-capable and you want to use multipath capabilities for the storage device IO redundancy, properly install and configure the host multipath package for Storage Scale consumption.

**Multipath Packages**

The multipath package version varies based on the OS version. At the time of Storage Scale testing, the following version is used for the multipath package on CentOS.

```
[root@sm247 scsi_host]# rpm -qa |grep multi*
device-mapper-multipath-0.8.4-10.el8.x86_64
device-mapper-multipath-libs-0.8.4-10.el8.x86_64
```

### Multipath Configuration

Linux typically stores their `multipath.conf` file at **`/etc/multipath.conf`**. If there is no such file at the location, you need to create it. We used `multipath.conf`.

Remember that this is not a best performance configuration; it is a reference point for a quick startup. The raw device name such as "`sda`," "`sdb`," etc., are host-specific. For optimal performance, you need to explore further tuning of each parameter for specific Storage Scale deployment.

As a best practice, you may want to exclude the local disk drives or any non-Seagate volumes/LUNS from the `multipath.conf` file.

Note: "`sda`" and "`sdb`" are Storage Scale host local disks so they are excluded from multipath configuration.

```
[root@sm247 ~]# cat /etc/multipath.conf
# device-mapper-multipath configuration file

# For a complete list of the default configuration values, run either:
# # multipath -t
# or
# # multipathd show config

# For a list of configuration options with descriptions, see the
# multipath.conf man page.

defaults {
        user_friendly_names yes
        find_multipaths yes
        enable_foreign "^$"
}

blacklist_exceptions {
        property "(SCSI_IDENT_|ID_WWN)"

}
devices {
    device {
        vendor "SEAGATE"
        product "4565"
        path_grouping_policy group_by_prio
        uid_attribute "ID_SERIAL"
        prio alua
#       path_selector "round-robin 0"
        path_selector "queue-length 0"
        path_checker tur
        failback immediate
        no_path_retry 5
    }
}
blacklist {
devnode "sda"
devnode "sdb"
}
```

For a detailed explanation of **`multipath.conf`**, refer to this link: https://www.thegeekdiary.com/understanding-the-dm-multipath-configuration-file-etc-multipath-conf.

Reboot the host after creating this configuration file in order for multipath to take effect.

### Multipath Verification

At the prompt, issue `multipath -ll` to ensure there are two active paths to each device according to the defined configuration file.

```
# multipath -ll
```

```
[root@sm247 ~]# multipath -ll
mpathbm (3600c0ff0006463417c61766301000000) dm-5 SEAGATE,4006
size=242T features='0' hwhandler='1 alua' wp=rw
|-+- policy='service-time 0' prio=50 status=active
| `- 0:0:1:2 sdh 8:112 active ready running
`-+- policy='service-time 0' prio=10 status=enabled
  `- 0:0:0:2 sde 8:64  active ready running
mpathbl (3600c0ff0006463417b61766301000000) dm-4 SEAGATE,4006
size=242T features='0' hwhandler='1 alua' wp=rw
|-+- policy='service-time 0' prio=50 status=active
| `- 0:0:1:1 sdg 8:96  active ready running
`-+- policy='service-time 0' prio=10 status=enabled
  `- 0:0:0:1 sdd 8:48  active ready running
mpathbk (3600c0ff000646341796l766301000000) dm-3 SEAGATE,4006
size=2.9T features='0' hwhandler='1 alua' wp=rw
|-+- policy='service-time 0' prio=50 status=active
| `- 0:0:1:0 sdf 8:80  active ready running
`-+- policy='service-time 0' prio=10 status=enabled
  `- 0:0:0:0 sdc 8:32  active ready running
```

# PREPARE THE STORAGE SCALE HOST

## Host OS and Kernel Update

The Linux kernel and OS release version must meet the minimum requirement specified in the Storage Scale installation guide.

In the POC, Storage Scale was installed over CentOS (CentOS Linux release 8.4.2105). For complete installation instructions of IBM Spectrum_Scale_DM_513_x86_64_LNX.tar, refer to IBM documentation at https://www.ibm.com/docs/en/spectrum-scale/5.1.3?topic=quick-reference.
In this POC, the following kernel version and tools must exist to install Storage Scale v. 5.1.3.

```
[root@sm247 gpfs_repo]# uname --kernel-release
4.18.0-305.25.1.el8_4.x86_64
```

When encountering errors such as "`error: Cannot find a valid kernel header file, the file is not at expected location`," we recommend a Linux kernel update. The following header files and tool utilities are needed for a successful installation.

If the host is connected to the network, use yum to update the kernel and install the tool utilities.

```
# yum -y install kernel-devel cpp gcc gcc-c++ kernel-headers
```

```
# yum install ksh perl m4 net-tools -y
```

Note: When installing or upgrading yum packages, "yum install" may not work properly if the local host CentOS Linux repo is not configured correctly or some files in `/etc/yum.repos.d` are missing or not updated. Check and update the following files under the `yum.repos.d` directory in order to run yum update successfully.
Storage Scale requires that `bsh` is running for a successful Storage Scale installation. Do the following to ensure that the bsh shell is under the correct user environment. If bsh is not running correctly, perform an

```
[root@sm247 yum.repos.d]# pwd
/etc/yum.repos.d
[root@sm247 yum.repos.d]# ll
total 48
-rw-r--r--. 1 root root  898 Jun 20 19:07 CentOS-Linux-AppStream.repo
-rw-r--r--. 1 root root  781 Jun 20 19:25 CentOS-Linux-BaseOS.repo
-rw-r--r--. 1 root root 1134 Jun  7 11:44 CentOS-Linux-ContinuousRelease.repo
-rw-r--r--. 1 root root  318 Sep 14  2021 CentOS-Linux-Debuginfo.repo
-rw-r--r--. 1 root root  736 Jun  7 11:44 CentOS-Linux-Devel.repo
-rw-r--r--. 1 root root  768 Jun 20 21:57 CentOS-Linux-Extras.repo
-rw-r--r--. 1 root root  723 Jun  7 11:44 CentOS-Linux-FastTrack.repo
-rw-r--r--. 1 root root  744 Jun  7 11:44 CentOS-Linux-HighAvailability.repo
-rw-r--r--. 1 root root  693 Sep 14  2021 CentOS-Linux-Media.repo
-rw-r--r--. 1 root root  710 Jun  7 11:44 CentOS-Linux-Plus.repo
-rw-r--r--. 1 root root  728 Jun  7 11:44 CentOS-Linux-PowerTools.repo
-rw-r--r--. 1 root root 1124 Sep 14  2021 CentOS-Linux-Sources.repo
```

```
[root@sm247 yum.repos.d]# cat ~/.bash_profile
# .bash_profile

# Get the aliases and functions
if [ -f ~/.bashrc ]; then
        . ~/.bashrc
fi

# User specific environment and startup programs

PATH=$PATH:$HOME/bin
export PATH=$PATH:$HOME/bin:/usr/lpp/mmfs/bin
export WCOLL=/nodes
```

update to export the path correctly as follows.

```
#cat  ~/.bash_profile
# vi ~/.bash_profile - example
if [ -f ~/.bashrc ]; then
        . ~/.bashrc
Fi
PATH=$PATH:$HOME/bin
Export PATH=$PATH:$HOME/bin:/usr/lpp/mmfs/bin
Export WCOLL=/nodes
```

### Host FQDN

Storage Scale requires that each NSD node in the cluster has a FQDN so all NSD nodes can communicate with each other and the storage resources can later be exported through its global name. In this example,  Scale host is updated for our name-to-IP resolution.

i) Assign a name to the host.

```
dhcp-192-168-53-197:~ # hostnamectl set-hostname sm247
```

```
[root@sm247 gpfs_repo]# hostnamectl
   Static hostname: sm247
         Icon name: computer-server
           Chassis: server
        Machine ID: da89522d7e114d4da2e6b2541f7608ee
           Boot ID: 10917f1a42e74f8c8284b9dbdbf42417
  Operating System: CentOS Linux 8
       CPE OS Name: cpe:/o:centos:centos:8
            Kernel: Linux 4.18.0-305.25.1.el8_4.x86_64
```

Use **vi** or some other text editor to edit the local host file to reflect such changes on the host name.

```
[root@sm247 gpfs_repo]# cat /etc/hosts
#
# hosts         This file describes a number of hostname-to-address
#               mappings for the TCP/IP subsystem.  It is mostly
#               used at boot time, when no name servers are running.
#               On small systems, this file can be used instead of a
#               "named" name server.
# Syntax:
#
# IP-Address  Full-Qualified-Hostname  Short-Hostname
#

127.0.0.1       localhost

# special IPv6 addresses
::1             localhost ipv6-localhost ipv6-loopback

fe00::0         ipv6-localnet

ff00::0         ipv6-mcastprefix
ff02::1         ipv6-allnodes
ff02::2         ipv6-allrouters
ff02::3         ipv6-allhosts

192.168.53.218  sm47
192.168.53.219  sm53
192.168.53.247  sm247
192.168.53.250  sm250
192.168.53.198  smc10
192.168.53.197  smc11
192.168.53.196  smc12
192.168.53.195  smc13
192.168.53.203  smc14
192.168.53.220  smc15
```

### Host passwordless ssh access

Storage Scale requires that **SSH** access to each of the hosts in the cluster be passwordless for successful installation and cluster operation. The following steps describe how to make the Storage Scale host server have passwordless **SSH** access.

i) Check if the host sec_id rsa file exists.

```
[root@sm247 gpfs_repo]# pwd
/root/gpfs_repo
[root@sm247 gpfs_repo]# ls ~/.ssh/id_*
/root/.ssh/id_rsa  /root/.ssh/id_rsa.pub
[root@sm247 gpfs_repo]#
```

ii) If the rsa.pub file does not exist, generate one by issuing the following CLI command.

```
[root@sm247 gpfs_repo]# ssh-keygen -t rsa -b 4096
```

The following shows a successful creation of rsa.pub file.

```
dhcp-192-168-53-195:~ # ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
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:sWzRxcOJg5khRsklGdOx6ZXyJ2kZIl6RWFmeejiYlNI root@dhcp-192-168-53-195
The key's randomart image is:
+---[RSA 3072]----+
|     oBO*X.+..   |
|     o==B*O+=    |
|    . E O B. .   |
|     + B @ +     |
|      + S B .    |
|       . + o     |
|                 |
|                 |
|                 |
+----[SHA256]-----+
dhcp-192-168-53-195:~ #
```

iii) Use this CLI command to copy the local host rsa.pub to each of the hosts in the cluster.

```
[root@sm247 gpfs repo]# ssh-copy-id root@IP or Host name remote node
```

a) Edit the entries in the /etc/ssh/sshd_config on each Storage Scale host to read as follows.

```
PasswordAuthentication no
ChallengeResponseAuthentication no
UsePAM no
```

b) Restart the SSH process by entering the following commands.

```
#systemctl restart ssh
#systemctl restart sshd
```

**Install the Storage Scale rpm on each host server.**

- `# tar xvf Spectrum_Scale_DM_513_x86_64_LNX.tar, then run the installation package and accept the license.`
- `# ./Spectrum_Scale_Protocols_Standard-5.2.1.0-x86_64-Linux-install`

The Storage Scale installation files are in `/usr/lpp/mmfs`. Find the Storage Scale_rpms directory and install the required rpms: base, ext, gskit, gpl, msg, and docs.

- `# cd /usr/lpp/mmfs/5.2.1.0/Storage Scale_rpms`
- `# rpm –ivh Storage Scale. {base,ext,gskit,gpl,msg,docs}*.rpm`

**Build the Storage Scale portable layer.**

There is an executable file in Storage Scale binary that automatically creates this portable Storage Scale package. This example is using CentOS (which is technically unsupported; only RHEL is supported), the install script will fail. The workaround is to append “Red Hat Enterprise Linux” to the end of the first line in `/etc/redhat-release` and then the build script so it will run properly.

**Note:** This workaround may not be necessary depending on the Linux version you are running.

```
[root@sm247 ~]# /usr/lpp/mmfs/bin/mmbuildgpl
------------------------------------------------------
mmbuildgpl: Building GPL (5.1.3.0) module begins at Mon Nov 28 12:35:37 PST 2022.
------------------------------------------------------
Verifying Kernel Header...
  kernel version = 41800305 (418000305025001, 4.18.0-305.25.1.el8_4.x86_64, 4.18.0-305.25.1)
  module include dir = /lib/modules/4.18.0-305.25.1.el8_4.x86_64/build/include
  module build dir   = /lib/modules/4.18.0-305.25.1.el8_4.x86_64/build
  kernel source dir  = /usr/src/linux-4.18.0-305.25.1.el8_4.x86_64/include
  Found valid kernel header file under /usr/src/kernels/4.18.0-305.25.1.el8_4.x86_64/include
Getting Kernel Cipher mode...
   Will use skcipher routines
Verifying Compiler...
  make is present at /bin/make
  cpp is present at /bin/cpp
  gcc is present at /bin/gcc
  g++ is present at /bin/g++
  ld is present at /bin/ld
Verifying libelf devel package...
  Verifying  elfutils-libelf-devel is installed ...
    Command: /bin/rpm -q  elfutils-libelf-devel
    The required package  elfutils-libelf-devel is installed
Verifying Additional System Headers...
  Verifying kernel-headers is installed ...
    Command: /bin/rpm -q kernel-headers
    The required package kernel-headers is installed
make World ...
make InstallImages ...
------------------------------------------------------
mmbuildgpl: Building GPL module completed successfully at Mon Nov 28 12:35:55 PST 2022.
------------------------------------------------------
```

The example above shows that a Storage Scale portable layer was successfully created.

**Configure the user path to ensure Storage Scale related CLI commands work.**

i) Edit your `.bashrc` and add `/usr/lpp/mmfs/bin` to your path.

ii) Export `PATH=$PATH:$HOME/bin:/usr/lpp/mmfs/bin`

iii) Validate the cluster is installed correctly.

```
[root@sm247 gpfs_repo]#  mmlscluster

GPFS cluster information
========================
  GPFS cluster name:         Seagate.gpfs
  GPFS cluster id:           8250500687422949
  GPFS UID domain:           Seagate.gpfs
  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   sm247             192.168.53.247  sm247            quorum
   2   sm250             192.168.53.250  sm250            quorum-manager
   3   sm47              192.168.53.218  sm47             quorum-manager
   4   sm53              192.168.53.219  sm53
```

**Configure NDS and the node list for Storage Scale.**

i) Now that Storage Scale is installed it must be configured. Create a Storage Scale Node Configuration File to designate the host node that you want to include in the Storage Scale cluster. This node list is a text file that contains the host names and roles you'd like to assign for the host nodes in the Storage Scale cluster and will simplify the workflow.

```
[root@sm247 gpfs_repo]# cat nodelist
sm247:quorum
sm250:quorum-manager
sm47:quorum-manager
sm53:client
```

ii) After the node list is created, create a cluster named Seagate Storage Scale, pass in the node list, and specify the host names of the primary and secondary nodes. The utility will **SSH** into all the nodes and add them to the cluster.

```
# mmcrcluster -C Seagate.gpfs  -N nodeList -p sm247 -s sm250
```

iii) Run mmlscluster to verify that all the nodes have been added and the cluster is running.

```
# mmlscluster
```

```
[root@sm247 gpfs_repo]#  mmlscluster

GPFS cluster information
========================
  GPFS cluster name:         Seagate.gpfs
  GPFS cluster id:           8250500687422949
  GPFS UID domain:           Seagate.gpfs
  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   sm247            192.168.53.247  sm247            quorum
   2   sm250            192.168.53.250  sm250            quorum-manager
   3   sm47             192.168.53.218  sm47             quorum-manager
   4   sm53             192.168.53.219  sm53
```

iv) Accept the license agreement and add the cluster license. Here `sm247`, `sm250`, and `sm47` are Storage Scale cluster nodes.

```
/usr/lpp/mmfs/bin/mmchlicense server --accept -N sm247, sm250, sm47
```

**Start the Storage Scale Cluster.**

Since your test would be in a non-production environment, you may want to disable the firewall on the cluster node host to save time on troublshooting any firewall related issues.

a) Enter the following command to start the cluster.

```
# mmstartup -a
```

Verify that the command is running. The first time you run this, your nodes will be “arbitrating” for a minute or two.

```
[root@sm247 gpfs_repo]# mmstartup -a
Thu Jul 28 22:45:02 PDT 2022: mmstartup: Starting GPFS ...
```

b) Enter **# mmlscluster**.

```
oot@sm247 ~]# mmlscluster

FS cluster information
======================
GPFS cluster name:         Seagate.gpfs
GPFS cluster id:           8250500687422949
GPFS UID domain:           Seagate.gpfs
Remote shell command:      /usr/bin/ssh
Remote file copy command:  /usr/bin/scp
Repository type:           CCR

ode  Daemon node name  IP address      Admin node name  Designation
-----------------------------------------------------------------------
 1   sm247            192.168.53.247  sm247            quorum
 2   sm250            192.168.53.250  sm250            quorum-manager
 3   sm47             192.168.53.218  sm47             quorum-manager
 4   sm53             192.168.53.219  sm53
```

c) Enter **# mmgetstate -L -a.**

```
[root@sm247 ~]# mmgetstate -L -a

 Node number  Node name  Quorum  Nodes up  Total nodes  GPFS state    Remarks
-------------------------------------------------------------------------------
           1  sm247         2         2            4  active        quorum node
           2  sm250         0         0            4  unknown       quorum node
           3  sm47          2         2            4  active        quorum node
           4  sm53          2         2            4  active
```

All cluster nodes should be listed as active if they are working correctly. If they’re stuck—arbitrating for longer than a couple of minutes—you probably don’t have passwordless SSH set up correctly. Also, and this is counterintuitive, every server must be able to SSH into itself without a password so be sure that it’s working. In the following screen shot, the node sm250 lost connection so it’s shown as unknown because the SSH was not working when SSHing into itself. If you need to shut it down to reconfigure something, the shutdown command is:

```
# mmshutdown -a
```

### Configure Network Shared Disks (Nsd)

The NSDs are the essential building blocks that Storage Scale uses to store data and metadata. To list the available block devices on the local node host, enter **`# lsblk.`**

```
[root@sm247 gpfs_repo]# lsblk
NAME          MAJ:MIN RM   SIZE RO TYPE  MOUNTPOINT
sda             8:0    0 931.5G  0 disk
├─sda1          8:1    0   600M  0 part  /boot/efi
├─sda2          8:2    0     1G  0 part  /boot
└─sda3          8:3    0 929.9G  0 part
  ├─cl-root   253:0    0    70G  0 lvm   /
  ├─cl-swap   253:1    0     4G  0 lvm   [SWAP]
  └─cl-home   253:2    0   1.8T  0 lvm   /home
sdb             8:16   0 931.5G  0 disk
└─sdb1          8:17   0 931.5G  0 part
  └─cl-home   253:2    0   1.8T  0 lvm   /home
sdc             8:32   0 127.3T  0 disk
├─sdc1          8:33   0 127.3T  0 part
└─mpathbc     253:3    0 127.3T  0 mpath
  └─mpathbc1  253:5    0 127.3T  0 part
sdd             8:48   0 127.3T  0 disk
├─sdd1          8:49   0 127.3T  0 part
└─mpathbg     253:4    0 127.3T  0 mpath
  └─mpathbg1  253:6    0 127.3T  0 part
sde             8:64   0 127.3T  0 disk
├─sde1          8:65   0 127.3T  0 part
└─mpathbc     253:3    0 127.3T  0 mpath
  └─mpathbc1  253:5    0 127.3T  0 part
sdf             8:80   0 127.3T  0 disk
├─sdf1          8:81   0 127.3T  0 part
└─mpathbg     253:4    0 127.3T  0 mpath
  └─mpathbg1  253:6    0 127.3T  0 part
```

In theory, Storage Scale NSD can be created on top of either raw devices (such as the ones listed under lsblk), or on the devices shown in the multipath outputs in the section above. Once you have the storage devices ready, it's time to create an NSD stanza file.

Creating a NSD stanza would save you time and make the NSD modification easier in our POC, raw devices are used instead of multipath devices. You can also manually create individual NSDs instead of using the NSD stanza approach.

a) Enter the #cat NSDstanza CLI command to create an NSD stanza.

```
[root@sm247 gpfs_repo]# cat NSDstanza
%nsd: nsd=nsd247 device=/dev/dm-3 servers=sm247 failureGroup=1
%nsd: nsd=nsd248 device=/dev/dm-4 servers=sm247 failureGroup=1

%nsd: nsd=nsd250 device=/dev/dm-3 servers=sm250 failureGroup=3
%nsd: nsd=nsd251 device=/dev/dm-4 servers=sm250 failureGroup=3
```

b) We used `/dev/dm-xx` device (raw devices) for Storage Scale NSDs instead of using multipath devices because, according to the `dm-multipath` man page, we are not supposed to use these devices. We are supposed to use the `/dev/mapper/mpathx` device names, namely, to use the alias under Linux Device Mapper Multipath (DMM), as they are the only ones guaranteed to remain boot consistent (`/dev/dm-x` devices reenumerated themselves and the device names could be different when they are formatted). However, Storage Scale doesn't recognize the `mpath` devices as valid block devices since they are symbolic links to the `/dev/dm-x` device sas shown, and Storage Scale Native RAID performs its own disk multi-pathing. The official IBM documentation says to use the `/dev/dm-x` devices. For the sake of saving time, we used `/dev/dm-x` instead in this POC test.

(https://www.ibm.com/docs/en/spectrum-scale/4.2.3?topic=issues-Storage Scale-is-not-using-underlying-multipath-device)

```
[root@sm247 mapper]# ll |grep mpath
lrwxrwxrwx. 1 root root       7 Jul 26 16:55 mpathbc -> ../dm-3
lrwxrwxrwx. 1 root root       7 Jul 26 16:55 mpathbc1 -> ../dm-5
lrwxrwxrwx. 1 root root       7 Jul 26 16:55 mpathbg -> ../dm-4
lrwxrwxrwx. 1 root root       7 Jul 26 16:55 mpathbg1 -> ../dm-6
```

For Reference: The following outputs show that Storage Scale thinks the devices are Linux DMMs rather than the multipath device DMMs that Storage Scale recognizes.

**# mmcrnsd -F NSDstanza**

```
[root@sm247 etc]# /usr/lpp/mmfs/bin/mmdevdiscover | grep dmm
dm-0 dmm
dm-1 dmm
dm-2 dmm
dm-3 dmm
dm-4 dmm
dm-5 dmm
dm-6 dmm
```

If your NSDs were part of a previous pool, you can add them with the **-v** option to over-write.

**# mmcrnsd -F NSDstanza -v no**

List your NSDs. Since the file system hasn't been created yet, they should all be listed as "free disk."

**# mmlsnsd**

Configure one of the disks as a "tie breaker" disk to avoid a split-brain condition.

```
[root@sm247 gpfs_repo]#  mmlsnsd

 File system   Disk name       NSD servers
---------------------------------------------
 cv01          nsd49           sm47
 (free disk)   nsd247          sm247
 (free disk)   nsd248          sm247
 (free disk)   nsd250          sm250
 (free disk)   nsd251          sm250
 (free disk)   nsd50           sm47
```

**# mmchconfig tiebreakerDisks="nsd247"**

Un-configure a disk as a tie breaker disk.

**# mmchconfig tiebreakerDisks=""**

Delete configured NSDs.

**# mmdelnsd -F NSDstanza**

Delete an individual NSD.

**# mmdelnsd nsd01**

**Create and format the Storage Scale file system**

a) Enter the following command.

```
# mmcrfs fs1 -F NSDstanza -B 1M -m 2 -M 2 -r 2 -R 2 -n 32 -T /gpfs/cv01, where
```

**Cv01** - the name of the Storage Scale file system.

**-F NSDstanza** - Pass in the stanza file.

**-B 1M** - Format with a 1M block size.

**-m 2** - Set the default number of metadata replicas to 2.

**-M 2** - Set the max number of metadata replicas to 2.

**-r 2** - Set the default number of data replicas to 2.

**-R 2** - Set the max number of data replicas to 2.

**-n 32** - Set the estimated number of clients to 32. Format the file system with the correct degree of parallelism.

**-T /gpfs/fs1**- Set the mount point to /gpfs/cv01.

b) Verify that the file system was created properly by entering the `# mmlsfs cv01` command to list the file system parameters.

```
[root@sm247 gpfs_repo]# mmlsfs cv01
flag                value                    description
------------------- ------------------------ -----------------------------------
 -f                 8192                     Minimum fragment (subblock) size in bytes
 -i                 4096                     Inode size in bytes
 -I                 32768                    Indirect block size in bytes
 -m                 2                        Default number of metadata replicas
 -M                 2                        Maximum number of metadata replicas
 -r                 2                        Default number of data replicas
 -R                 2                        Maximum number of data replicas
 -j                 cluster                  Block allocation type
 -D                 nfs4                     File locking semantics in effect
 -k                 all                      ACL semantics in effect
 -n                 64                       Estimated number of nodes that will mount file system
 -B                 4194304                  Block size
 -Q                 none                     Quotas accounting enabled
                    none                     Quotas enforced
                    none                     Default quotas enabled
 --perfileset-quota no                       Per-fileset quota enforcement
 --filesetdf        no                       Fileset df enabled?
 -V                 27.00 (5.1.3.0)          File system version
 --create-time      Tue Jun 21 20:20:22 2022 File system creation time
 -z                 no                       Is DMAPI enabled?
 -L                 33554432                 Logfile size
 -E                 yes                      Exact mtime mount option
 -S                 relatime                 Suppress atime mount option
 -K                 whenpossible             Strict replica allocation option
 --fastea           yes                      Fast external attributes enabled?
 --encryption       no                       Encryption enabled?
 --inode-limit      134217728                Maximum number of inodes
 --log-replicas     0                        Number of log replicas
 --is4KAligned      yes                      is4KAligned?
 --rapid-repair     yes                      rapidRepair enabled?
 --write-cache-threshold 0                   HAWC Threshold (max 65536)
 --subblocks-per-full-block 512              Number of subblocks per full block
 -P                 system                   Disk storage pools in file system
 --file-audit-log   no                       File Audit Logging enabled?
 --maintenance-mode no                       Maintenance Mode enabled?
 --flush-on-close   no                       flush cache on file close enabled?
 -d                 nsd49                    Disks in file system
 -A                 yes                      Automatic mount option
 -o                 none                     Additional mount options
 -T                 /gpfs/cv01               Default mount point
 --mount-priority   0                        Mount priority
```

c) Mount the file system by entering the CLI command

```
# mmmount all -a
```

d) Verify that it's been mounted correctly by entering the # def –kh command. Check disk space on every node in the Storage Scale cluster.

```
[root@sm247 gpfs_repo]# df -kh
Filesystem           Size  Used Avail Use% Mounted on
devtmpfs              94G   40G   54G  43% /dev
tmpfs                 94G  4.0K   94G   1% /dev/shm
tmpfs                 94G   50M   94G   1% /run
tmpfs                 94G     0   94G   0% /sys/fs/cgroup
/dev/mapper/cl-root   70G   20G   51G  28% /
/dev/mapper/cl-home  1.8T   13G  1.8T   1% /home
/dev/sda2           1014M  402M  613M  40% /boot
/dev/sda1            599M  7.3M  592M   2% /boot/efi
tmpfs                 19G   16K   19G   1% /run/user/42
tmpfs                 19G     0   19G   0% /run/user/0
cv01                 128T   69G  128T   1% /gpfs/cv01
```

Alternatively, you can verify the Storage Scale file system is mounted correctly by checking the Local File System table. The example below shows that cv01 is mounted at `/gpfs/cv01` on this node.

```
[root@sm247 gpfs_repo]# cat /etc/fstab

#
# /etc/fstab
# Created by anaconda on Fri Aug 27 04:44:29 2021
#
# Accessible filesystems, by reference, are maintained under '/dev/disk/'.
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info.
#
# After editing this file, run 'systemctl daemon-reload' to update systemd
# units generated from this file.
#
/dev/mapper/cl-root     /                       xfs     defaults        0 0
UUID=14d524eb-4d85-47c5-b95a-71e5a9bd3680 /boot                   xfs     defaults        0 0
UUID=DA99-8F65          /boot/efi               vfat    umask=0077,shortname=winnt 0 2
/dev/mapper/cl-home     /home                   xfs     defaults        0 0
/dev/mapper/cl-swap     none                    swap    defaults        0 0
cv01                /gpfs/cv01          gpfs       rw,mtime,relatime,dev=cv01,noauto 0 0
```

Check the replication settings for your file system by entering the CLI command

**`# mmlsfs fs1 -mrMR`**

```
[root@sm247 gpfs_repo]# mmlsfs cv01 -mrMR
flag             value                    description
---------------- ------------------------ -----------------------------------
 -m              2                        Default number of metadata replicas
 -r              2                        Default number of data replicas
 -M              2                        Maximum number of metadata replicas
 -R              2                        Maximum number of data replicas
```

### Client Operation (Add/Remove)

a) Add a client

If you need to add another node, follow the normal installation procedure on the new node and then run **`mmadnode`** to add the node to the cluster.

**`# mmaddnode -N client1:client`**

b) Delete a client using the CLI command

**`# mmdelnode -n client1`**

c) Change a server's Role using either of these CLI commands:

**`# mmchnode --quorum --manager -N servername`**

**`# mmchnode --client -N servername`**

## Performance Tuning and Troubleshooting

This section includes a few short procedures to help you find Storage Scale system logs and some basic performance parameter configurations. These performance tuning parameters are not optimal to deliver the best performance. We recommend that you contact IBM professional services to optimize performance.

### Performance Tuning

Storage Scale now groups some of the performance tuning under system quality of service (QoS). Run the following commands to tune Storage Scale for the Seagate 4006 storage system. These can be run from any Node in the cluster.

```
# mmchconfig maxMBps=10000 -N nodelist
# mmchconfig worker1Threads=1024 -N nodelist
# mmchconfig maxReceiverThreads=128 -N nodelist
# mmchconfig nsdMaxWorkerThreads=2048 -N nodelist
# mmchconfig nsdMinWorkerThreads=128 -N nodelist
# mmchconfig nsdMultiQueue=512 -N nodelist
# mmchconfig nsdSmallThreadRatio=1 -N nodelist
# mmchconfig nsdThreadsPerQueue=4 -N nodelist
# mmchconfig prefetchAggressiveness=1 -N nodelist
```

The Storage Scale process has to be re-started after the tuning. Enter the following commands to restart Storage Scale.

```
# mmumount all -a
# mmshutdown -a
# mmstartup -a
```

Wait until all Storage Scale nodes are active, then mount the file system by entering the CLI command

```
# mmmount all -a
```

### Troubleshooting

When you first run into an issue, check the logs from both Storage Scale and the Host OS. Storage Scale log files are stored in `/var/adm/ras/mmfs.log.latest`. There is one on every physical Storage Scale node.

```
[root@sm247 gpfs_repo]# ll /var/adm/ras/mmfs.log.latest
lrwxrwxrwx. 1 root root 34 Jul 28 22:45 /var/adm/ras/mmfs.log.latest -> mmfs.log.2022.07.28.22.45.03.sm247
[root@sm247 gpfs_repo]#
```

More troubleshooting related information can be found at IBM's online document depot, found at https://www.ibm.com/docs/en/spectrum-scale/5.0.0?topic=troubleshooting. Since Storage Scale 5.1.3 is a non-released version at the time of our testing, we included links to the closest available release (ver. 5.0.0 above).