EXOS™

ISV Integration Guide

IBM STORAGE SCALE INTEGRATION GUIDE

Deploying IBM Storage Scale and Exos CORVAULT storage on Red Hat Enterprise Linux

SEAGATE

CONTENTS

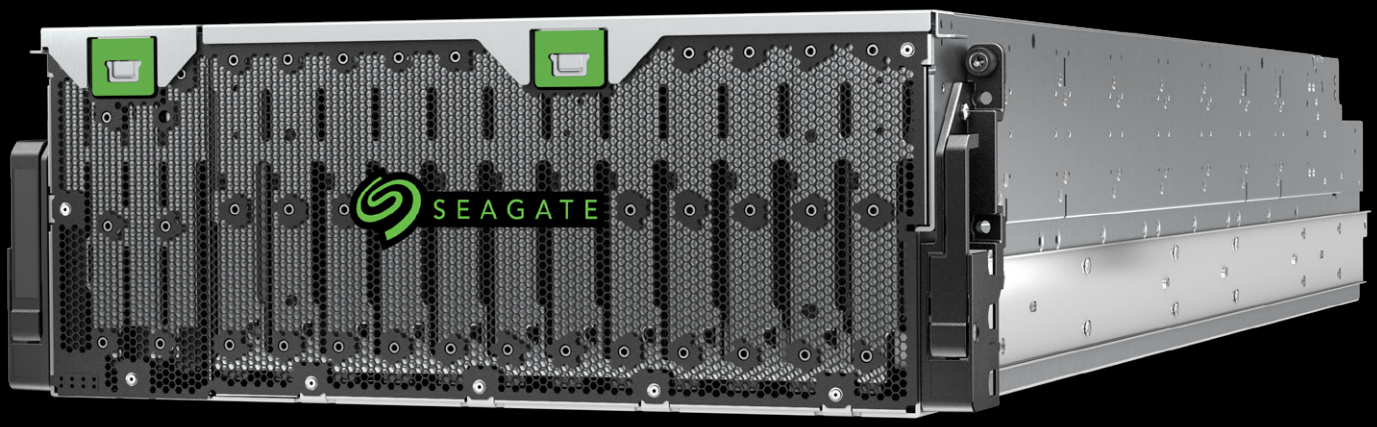
Introduction

The purpose of this document is to provide a step-by-step guide to deploy and implement IBM Storage Scale GPFS on Seagate® Exos® CORVAULT™ systems. In our example we use CORVAULT in conjunction with Red Hat host servers to validate Storage Scale deployment and implementation and to provide a reference point for Seagate field teams and Seagate partners in their customer engagement with Storage Scale deployment.

The entire procedure focuses on the following areas:

- Seagate storage configuration and performance optimization
- Deployment of a Storage Scale server with Red Hat Enterprise Linux
- Starting the Storage Scale cluster and mounting the Storage Scale file system
- Performance benchmark and tool configuration

This document may be used in conjunction with 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 Storage Scale, 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 the task of Storage Scale professional services. However, we included a section to discuss some of the performance optimizations seen with Seagate storage, Storage Scale, and the performance benchmark tools we used to obtain better performance.

We hope this information provides useful insights that will lead to further exploration of Storage Scale performance best practices in future deployments.

Lab Environment

Hardware

The Storage Scale test environment uses Seagate storage with 18TB HDDs, a Red Hat host server, and a management switch connected to the system management ports. The DAS (direct attached storage) topology is called for to connect Seagate storage and Storage Scale host servers.

DAS topology is one of the commonly used deployment options of Storage Scale and is a much simpler method to use for use cases where scale-out is not a consideration for Storage Scale deployment. This is why we adopted the DAS approach in our deployment.

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.

Storage	
Quantity	**Description**
1	Exos Corvault
106	18TB HDD

Host Server			
Quantity	**Description**	**Model**	**CPU**
4	Super Micro	SYS-220P-C9RT	Intel Xeon(R) Silver 4214@ 2.20GH
2	4-port LSI SAS HBA	9500-16	

Note: Out of four host server nodes, two nodes are important for Storage Scale deployment. These are dedicated primary and secondary Storage Scale cluster nodes and they are directly connected to the Seagate storage system using SAS. This defines the performance capability of Storage Scale. Users are advised to check the Storage Scale hardware support matrix to ensure the servers used for primary and secondary Storage Scale cluster nodes meet the requirements in processor, memory, and local storage.

In our testing, we also deployed two additional nodes to act as NFS client and FIO load generator. These two nodes do not have SAS connections to the Seagate storage system. Rather, the Storage Scale shared storage resource was exported through NFS and performance data was collected at the mount points of Storage Scale file level.

Software

The software package and version information are provided below only for reference since IBM Storage Scale may have different hardware requirements and therefore the software packages on the Storage Scale host may differ for each release.

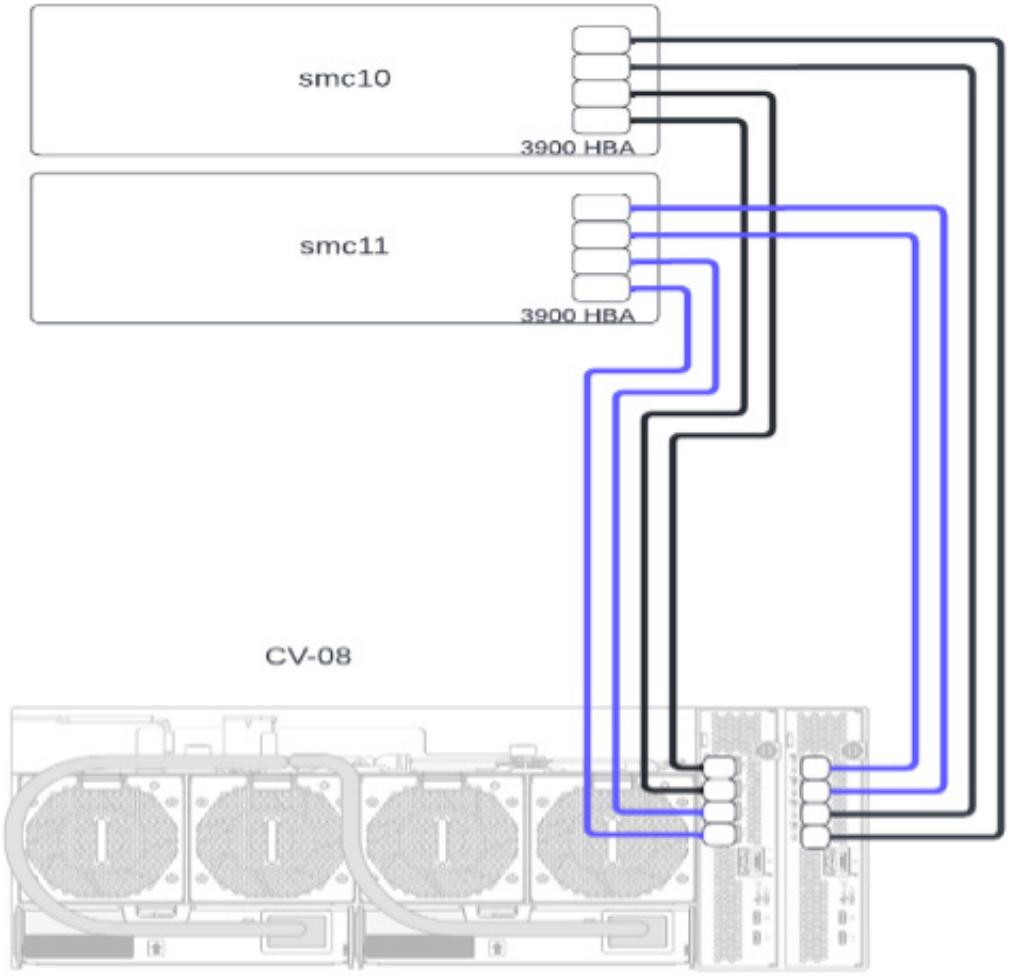
Host Server	Software Version
OS	Red Hat Enterprise Linux release 8.6
OS Kernel	4.18.0-305.25.1.el8_4.x86_64
SAS HBA driver	LSI MPT Fusion SAS 3.0 Device Driver
SAS HBA firmware	35.101.00.00
IBM Storage Scale	Spectrum_Scale_Data_Management-5.1.3.0-x86_64-Linux

Storage	Firmware
EXOS X 6575	S100C015e
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 test. In the test, Storage Scale NSD host servers (primary and second nodes) are connected to the storage via HD mini-SAS cables. Each of the hosts have a 4-port 12G SAS HBA, which is cross connected to both controllers of the CORVAULT system.

The following diagram depicts the connection topologies. In the diagram, host node smc10 and smc11 are configured as primary and second nodes in the Storage Scale cluster. CV-08 is a Seagate CORVAULT system with an 18TB disk drive and drive enclosure.



Seagate Storage Configuration

Network Shared Disk (NSD) is used to host user data and metadata in the Storage Scale operation. It can be created on the base of single-path disk devices or multipath-capable disk devices. Storage Scale runs off these NSDs.

Seagate storage resources need to be configured properly before they are consumed by Storage Scale for the NSDs. The storage configuration can be achieved through either the web user interface (UI) or through manual operation via the SSH CLI. For a better user experience we recommend that the storage configuration be done via the web UI.

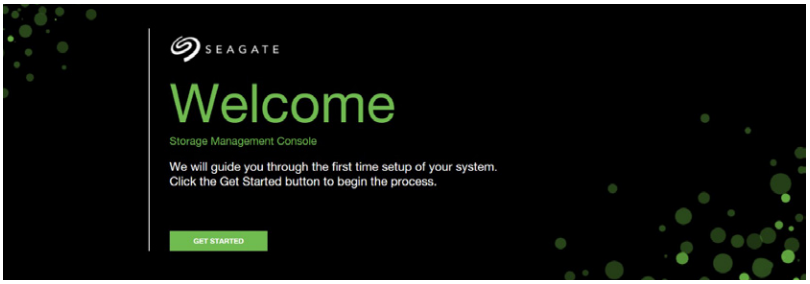
Note: As of this paper's release date, Seagate inter-leaved volumes creation is not available on CORVAULT from the web UI, However, this function will be available in a future firmware release.

User Onboarding

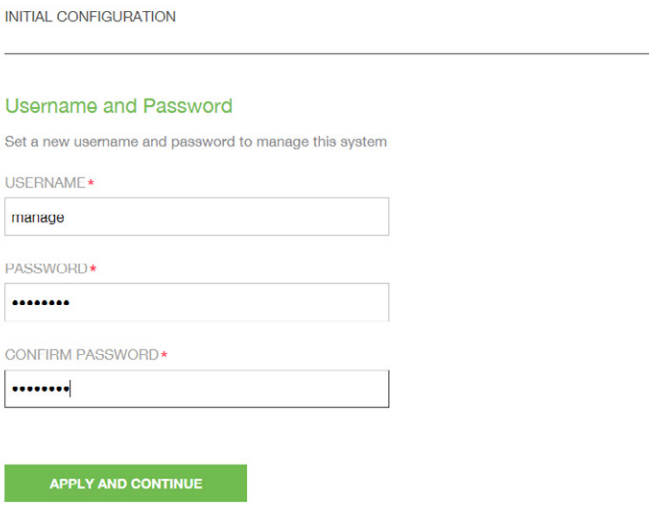
For simplicity, we skip the Seagate storage initialization and baseline configuration via serial console port and focus on the new user onboarding process.

Storage configuration is a three-step process that consists of system configuration, disk pool and disk group configuration, and storage resource provisioning and exporting. The following procedure walks you through the steps to bring the storage resource online.

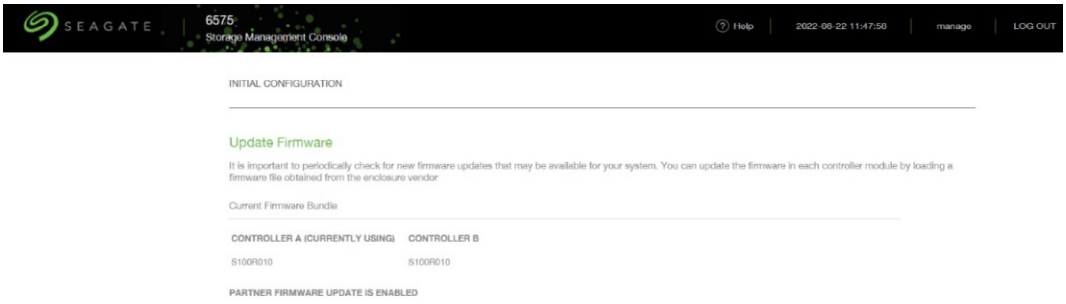
1. To begin user on-boarding, type `https://<IP_address_of_the_storage>.`



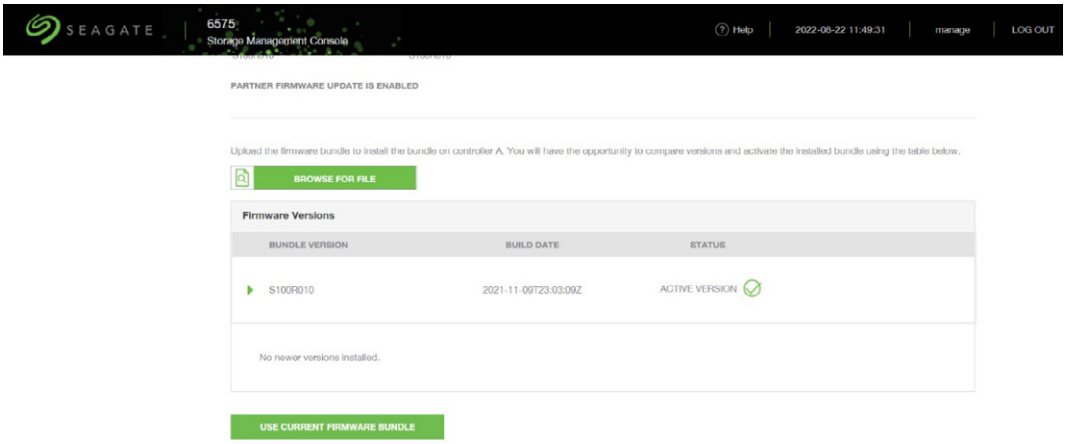
2. Create your admin user ID and password.



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

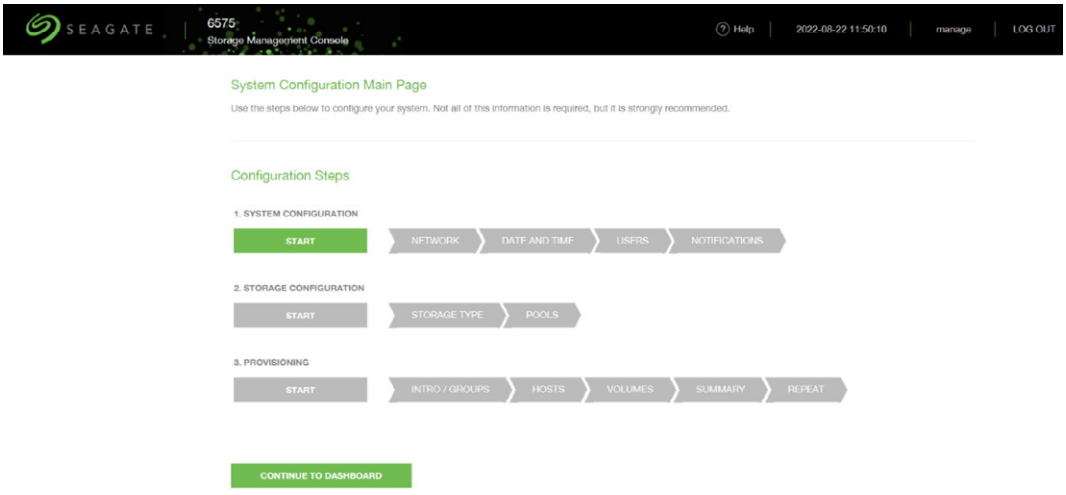


4. Follow the prompts to configure the storage system.

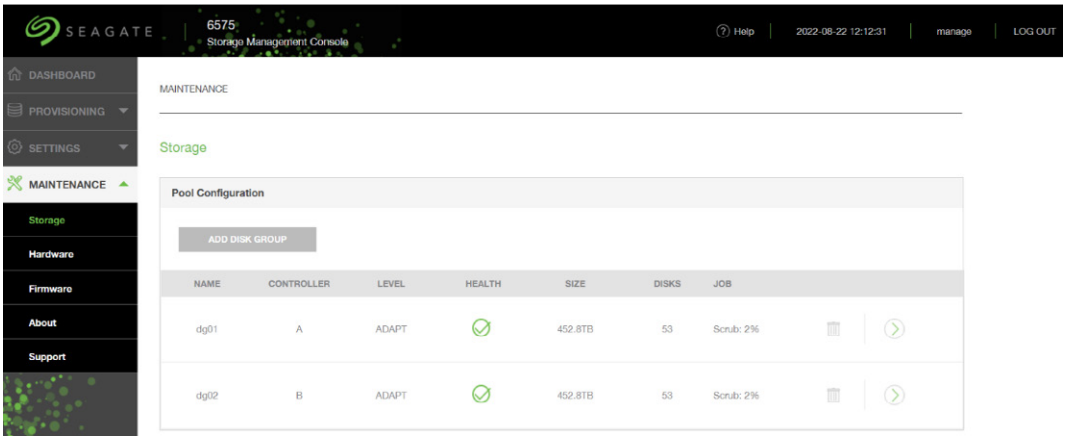


In this configuration, the user will need to go through the steps listed for each of three major configurations. Alternatively, the user can skip some of the steps to have a quick setup to go directly to the storage resource management configuration below.

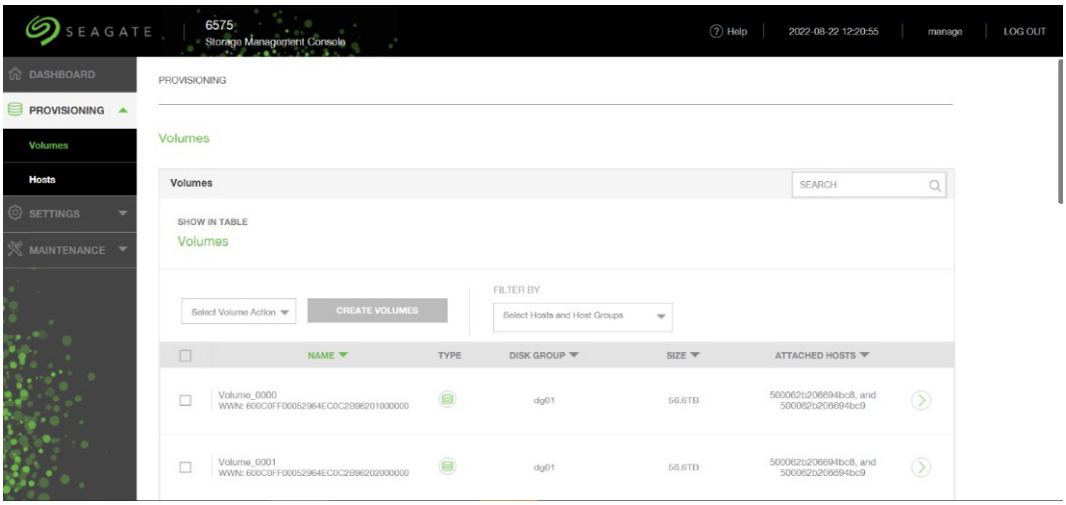
Note: We recommend that interleaved volumes be created when more than one volume is desired. At the time of testing, this functionality was only available via the CLI.



5. Create disk groups to include the disk drives. You must create a minimum of one disk group in the disk pool.



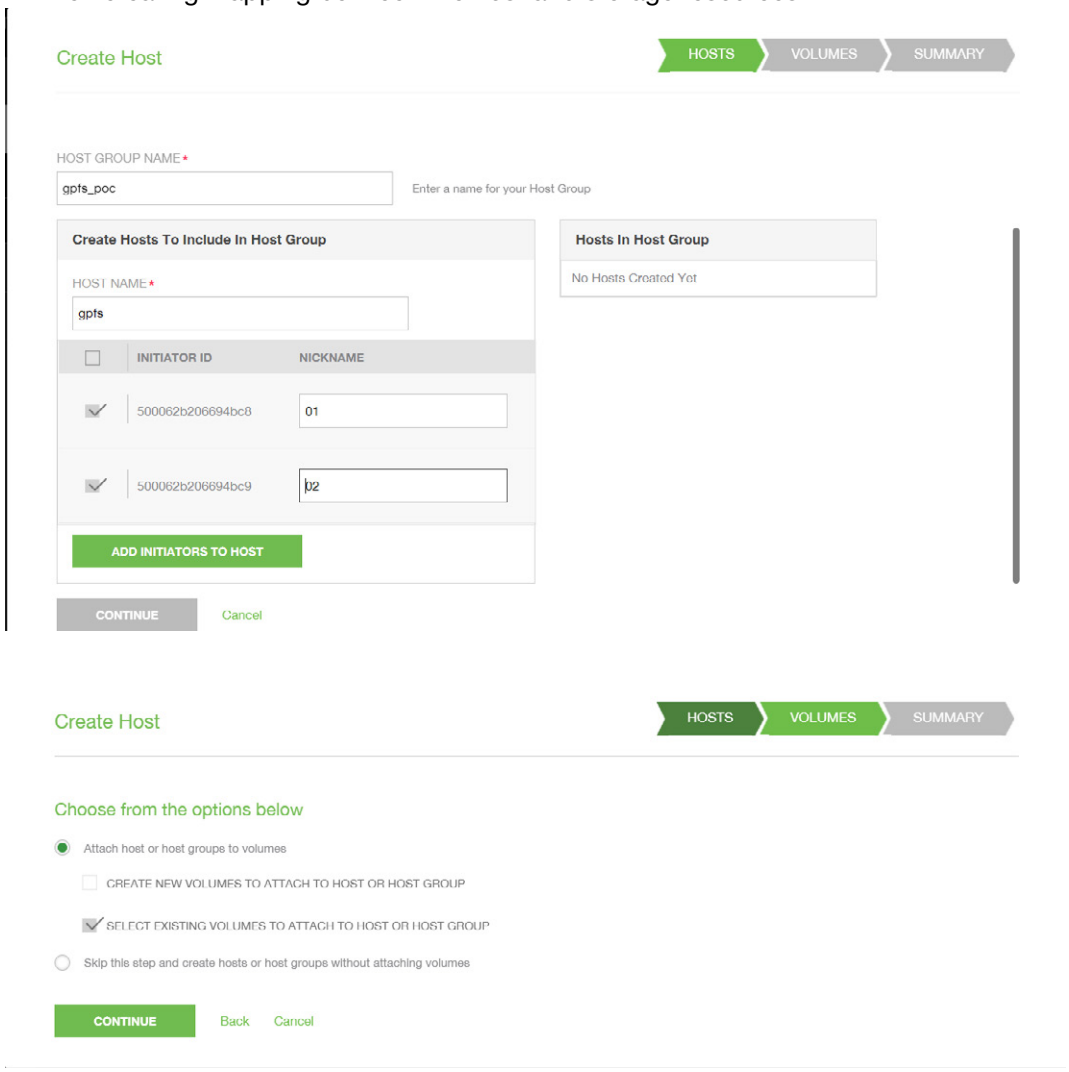
6. Create disk group volumes.



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.

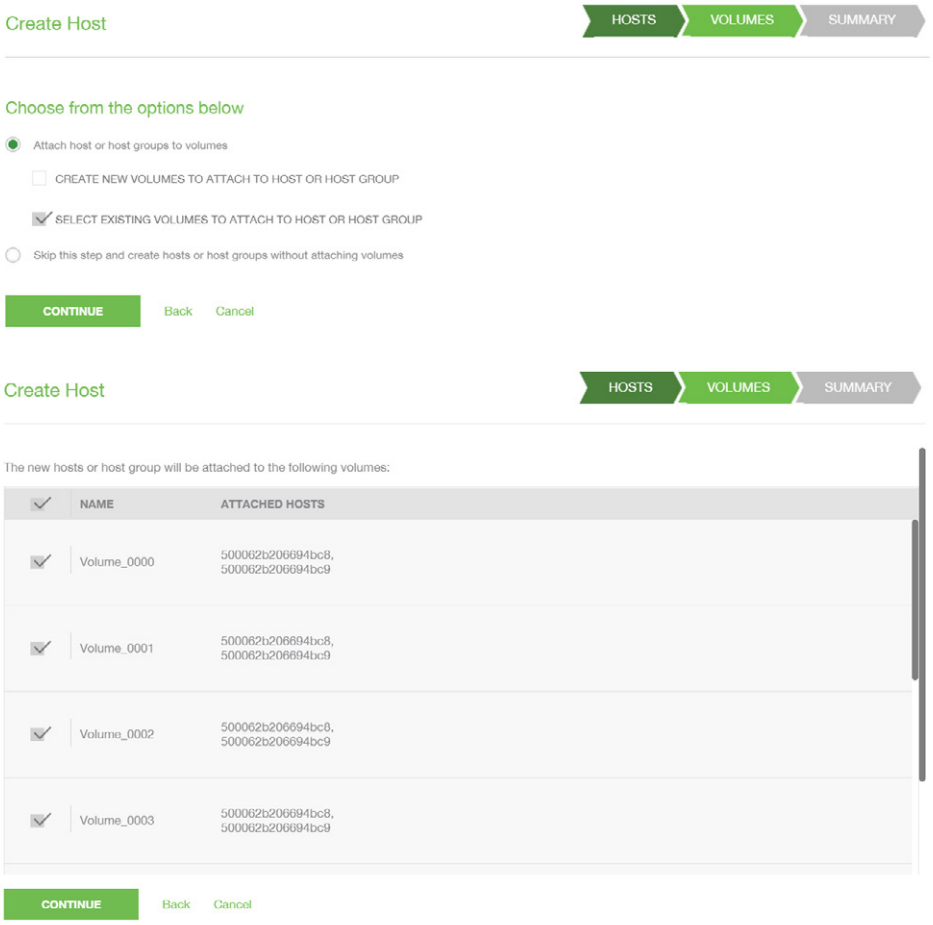
7. Create a host group to include the Storage Scale host as initiators as shown in the following example.

   Note: Creating a host group is optional—the user can use the host initiator when creating mapping between the host and storage resources.

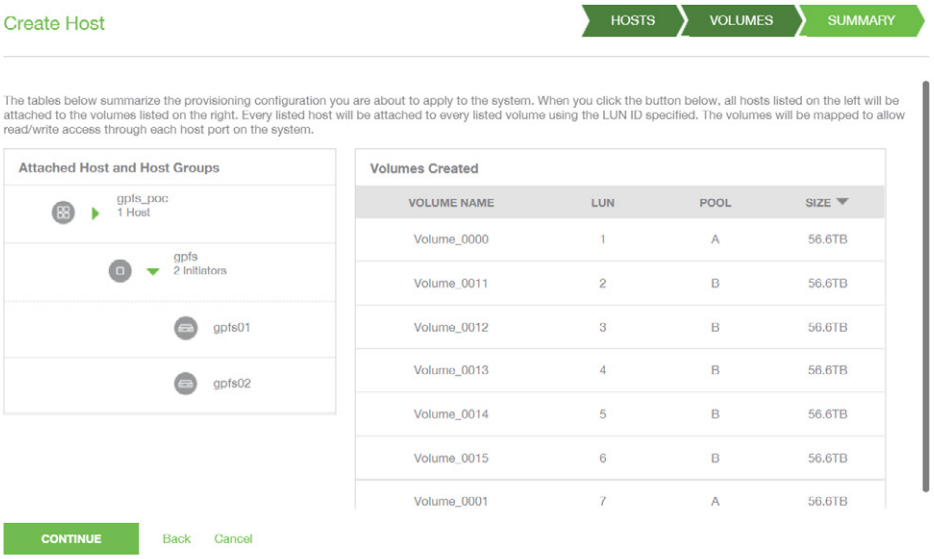


8. Create maps to connect the Storage Scale hosts to the storage volumes.

   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 and verify if these storage resources are made available to the Storage Scale hosts for further processes.



Storage Connection Verification

On the primary and secondary host nodes, the following CLI commands will help you identify that the storage targets are available to create the Storage Scale NSD.

1. On the Storage Scale host nodes, identify that the HBAs are on the host by verifying that they are reported to the host OS correctly. The driver module of the LSI HBA in the snapshot below (mpt3sas) indicates the HBA is detected and correctly connected to the host node.

```
[root@smc10 fioTemplete]# lsscsi --host
[0]    megaraid_sas
[1]    mpt3sas
[2]    ahci
[3]    ahci
[4]    ahci
[5]    ahci
[6]    ahci
```

2. Use the following CLI commands to verify that the disk drive and storage enclosures are correctly connected.

```
[root@smc10 ~]# lsscsi |grep -i Seagate
[1:0:9:0]    enclosu SEAGATE  6575             S100  -
[1:0:9:4]    disk    SEAGATE  6575             S100  /dev/sdc
[1:0:9:5]    disk    SEAGATE  6575             S100  /dev/sdd
[1:0:9:6]    disk    SEAGATE  6575             S100  /dev/sde
[1:0:9:7]    disk    SEAGATE  6575             S100  /dev/sdf
[1:0:9:8]    disk    SEAGATE  6575             S100  /dev/sdg
[1:0:9:9]    disk    SEAGATE  6575             S100  /dev/sdh
[1:0:9:10]   disk    SEAGATE  6575             S100  /dev/sdi
[1:0:9:11]   disk    SEAGATE  6575             S100  /dev/sdj
[1:0:10:0]   enclosu SEAGATE  6575             S100  -
[1:0:10:4]   disk    SEAGATE  6575             S100  /dev/sds
[1:0:10:5]   disk    SEAGATE  6575             S100  /dev/sdt
[1:0:10:6]   disk    SEAGATE  6575             S100  /dev/sdu
[1:0:10:7]   disk    SEAGATE  6575             S100  /dev/sdv
[1:0:10:8]   disk    SEAGATE  6575             S100  /dev/sdw
[1:0:10:9]   disk    SEAGATE  6575             S100  /dev/sdx
[1:0:10:10]  disk    SEAGATE  6575             S100  /dev/sdy
[1:0:10:11]  disk    SEAGATE  6575             S100  /dev/sdz
[1:0:11:0]   enclosu SEAGATE  6575             S100  -
[1:0:11:4]   disk    SEAGATE  6575             S100  /dev/sdk
[1:0:11:5]   disk    SEAGATE  6575             S100  /dev/sdl
[1:0:11:6]   disk    SEAGATE  6575             S100  /dev/sdm
[1:0:11:7]   disk    SEAGATE  6575             S100  /dev/sdn
[1:0:11:8]   disk    SEAGATE  6575             S100  /dev/sdo
[1:0:11:9]   disk    SEAGATE  6575             S100  /dev/sdp
[1:0:11:10]  disk    SEAGATE  6575             S100  /dev/sdq
[1:0:11:11]  disk    SEAGATE  6575             S100  /dev/sdr
[1:0:12:0]   enclosu SEAGATE  6575             S100  -
[1:0:12:4]   disk    SEAGATE  6575             S100  /dev/sdaa
[1:0:12:5]   disk    SEAGATE  6575             S100  /dev/sdab
[1:0:12:6]   disk    SEAGATE  6575             S100  /dev/sdac
[1:0:12:7]   disk    SEAGATE  6575             S100  /dev/sdad
[1:0:12:8]   disk    SEAGATE  6575             S100  /dev/sdae
[1:0:12:9]   disk    SEAGATE  6575             S100  /dev/sdaf
[1:0:12:10]  disk    SEAGATE  6575             S100  /dev/sdag
[1:0:12:11]  disk    SEAGATE  6575             S100  /dev/sdah
```

Alternatively, you can issue lsscsi –d to extract the same information about Seagate storage.

```
smc10:~ # lsscsi -d
[0:1:124:0]  enclosu BROADCOM VirtualSES       03    -
[0:3:110:0]  disk    BROADCOM MR9560-8i        5.18  /dev/sda [8:0]
[0:3:111:0]  disk    BROADCOM MR9560-8i        5.18  /dev/sdb [8:16]
[1:0:0:0]    enclosu SEAGATE  6575             S100  -
[1:0:0:1]    disk    SEAGATE  6575             S100  /dev/sdc [8:32]
[1:0:0:2]    disk    SEAGATE  6575             S100  /dev/sdd [8:48]
[1:0:0:3]    disk    SEAGATE  6575             S100  /dev/sde [8:64]
[1:0:0:4]    disk    SEAGATE  6575             S100  /dev/sdf [8:80]
[1:0:1:0]    enclosu SEAGATE  6575             S100  -
[1:0:1:1]    disk    SEAGATE  6575             S100  /dev/sdg [8:96]
[1:0:1:2]    disk    SEAGATE  6575             S100  /dev/sdl [8:176]
[1:0:1:3]    disk    SEAGATE  6575             S100  /dev/sdm [8:192]
[1:0:1:4]    disk    SEAGATE  6575             S100  /dev/sdj [8:144]
[1:0:2:0]    enclosu SEAGATE  6575             S100  -
[1:0:2:1]    disk    SEAGATE  6575             S100  /dev/sdk [8:160]
[1:0:2:2]    disk    SEAGATE  6575             S100  /dev/sdi [8:128]
[1:0:2:3]    disk    SEAGATE  6575             S100  /dev/sdh [8:112]
[1:0:2:4]    disk    SEAGATE  6575             S100  /dev/sdn [8:208]
[1:0:3:0]    enclosu SEAGATE  6575             S100  -
[1:0:3:1]    disk    SEAGATE  6575             S100  /dev/sdo [8:224]
[1:0:3:2]    disk    SEAGATE  6575             S100  /dev/sdp [8:240]
[1:0:3:3]    disk    SEAGATE  6575             S100  /dev/sdq [65:0]
[1:0:3:4]    disk    SEAGATE  6575             S100  /dev/sdr [65:16]
[1:0:4:0]    enclosu BROADCOM VirtualSES       03    -
[16:0:0:0]   disk    STT      USB_RMP          1100  /dev/sds [65:32]
```

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

```
smc10:~ # lsscsi -s
[0:1:124:0]  enclosu BROADCOM VirtualSES       03    -              -
[0:3:110:0]  disk    BROADCOM MR9560-8i        5.18  /dev/sda   1.91TB
[0:3:111:0]  disk    BROADCOM MR9560-8i        5.18  /dev/sdb   1.91TB
[1:0:0:0]    enclosu SEAGATE  6575             S100  -              -
[1:0:0:1]    disk    SEAGATE  6575             S100  /dev/sdc    172TB
[1:0:0:2]    disk    SEAGATE  6575             S100  /dev/sdd    172TB
[1:0:0:3]    disk    SEAGATE  6575             S100  /dev/sde    172TB
[1:0:0:4]    disk    SEAGATE  6575             S100  /dev/sdf    186TB
[1:0:1:0]    enclosu SEAGATE  6575             S100  -              -
[1:0:1:1]    disk    SEAGATE  6575             S100  /dev/sdg    172TB
[1:0:1:2]    disk    SEAGATE  6575             S100  /dev/sdl    172TB
[1:0:1:3]    disk    SEAGATE  6575             S100  /dev/sdm    172TB
[1:0:1:4]    disk    SEAGATE  6575             S100  /dev/sdj    186TB
[1:0:2:0]    enclosu SEAGATE  6575             S100  -              -
[1:0:2:1]    disk    SEAGATE  6575             S100  /dev/sdk    172TB
[1:0:2:2]    disk    SEAGATE  6575             S100  /dev/sdi    172TB
[1:0:2:3]    disk    SEAGATE  6575             S100  /dev/sdh    172TB
[1:0:2:4]    disk    SEAGATE  6575             S100  /dev/sdn    186TB
[1:0:3:0]    enclosu SEAGATE  6575             S100  -              -
[1:0:3:1]    disk    SEAGATE  6575             S100  /dev/sdo    172TB
[1:0:3:2]    disk    SEAGATE  6575             S100  /dev/sdp    172TB
[1:0:3:3]    disk    SEAGATE  6575             S100  /dev/sdq    172TB
[1:0:3:4]    disk    SEAGATE  6575             S100  /dev/sdr    186TB
[1:0:4:0]    enclosu BROADCOM VirtualSES       03    -              -
[16:0:0:0]   disk    STT      USB_RMP          1100  /dev/sds   31.4GB
```

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

Storage Scale Host Software Installation

Multipath Consideration

This section describes the processes to prepare the host OS for installing Storage Scale. If the storage device is multipath-capable and device IO level redundancy is desired, we suggest that host multipath be configured before installing Storage Scale. Multipath software package versions vary based on the host OS. We used the following version at the time of Storage Scale testing.

```
[root@smc10 ~]# rpm -qa |grep mult*
device-mapper-multipath-libs-0.8.4-22.el8.x86_64
device-mapper-multipath-0.8.4-22.el8.x86_64
```

Multipath Configuration

Linux typically stores their `multipath.conf` file at `/etc/multipath.conf`. If there is no `multipath.conf` at the location, you need to create it. We used the following a `multipath.conf` to validate Storage Scale.

```
defaults {
#       user_friendly_names yes
#       bindings_file "/etc/multipath/bindings"
        find_multipaths yes
#       enable_foreign "^$"
}

blacklist_exceptions {
#        property "(SCSI_IDENT_|ID_WWN)"
}

devices {
    device {
        vendor "SEAGATE"
        product "6575"
#       polling_interval 0
#       path_grouping_policy multibus
        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
#       path_checker directio
#       hardware_handler "1 alua"
        failback immediate
#       rr_weight priorities
#       rr_weight uniform
#       rr_min_io_rq 1
        no_path_retry 5
#       alias_prefix          "mpath"
    }
}
blacklist {
#wwid "360030480255f82f02a51d1e8e83bfd30"
#wwid "360030480255f82f02a51d1ac8fa3095e"
wwid "360030480255f3cf02a51ccf211470a4b"
wwid "360030480255f3cf02a51cccad4ea94f9"
}
```

For a detailed explanation of `multipath.conf`, refer to the information at the 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.

Verify Multipath

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

```
# multipath -ll
```

```
smc10:~ # multipath -ll
mpathc (3600c0ff000535a948eedc26201000000) dm-2 SEAGATE,6575
size=170T features='0' hwhandler='1 alua' wp=rw
`-+- policy='round-robin 0' prio=50 status=active
  |- 1:0:0:4 sdb 8:16 active ready running
  `- 1:0:2:4 sdq 65:0 active ready running
mpathd (3600c0ff000535a948bedc26201000000) dm-4 SEAGATE,6575
size=157T features='0' hwhandler='1 alua' wp=rw
`-+- policy='round-robin 0' prio=50 status=active
  |- 1:0:0:1 sde 8:64  active ready running
  `- 1:0:2:1 sdn 8:208 active ready running
mpathf (3600c0ff000535a9c89edc26201000000) dm-6 SEAGATE,6575
size=157T features='0' hwhandler='1 alua' wp=rw
`-+- policy='round-robin 0' prio=50 status=active
  |- 1:0:1:3 sdh 8:112 active ready running
  `- 1:0:3:3 sdp 8:240 active ready running
mpathg (3600c0ff000535a9c87edc26201000000) dm-1 SEAGATE,6575
size=157T features='0' hwhandler='1 alua' wp=rw
`-+- policy='round-robin 0' prio=50 status=active
  |- 1:0:1:1 sdg 8:96  active ready running
  `- 1:0:3:1 sdo 8:224 active ready running
mpathh (3600c0ff000535a948cedc26201000000) dm-5 SEAGATE,6575
size=157T features='0' hwhandler='1 alua' wp=rw
`-+- policy='round-robin 0' prio=50 status=active
  |- 1:0:2:2 sdk 8:160 active ready running
  `- 1:0:0:2 sdc 8:32  active ready running
mpathi (3600c0ff000535a9c88edc26201000000) dm-3 SEAGATE,6575
size=157T features='0' hwhandler='1 alua' wp=rw
`-+- policy='round-robin 0' prio=50 status=active
  |- 1:0:3:2 sdl 8:176 active ready running
  `- 1:0:1:2 sdi 8:128 active ready running
mpathj (3600c0ff000535a9c8aedc26201000000) dm-7 SEAGATE,6575
size=170T features='0' hwhandler='1 alua' wp=rw
`-+- policy='round-robin 0' prio=50 status=active
  |- 1:0:1:4 sdj 8:144 active ready running
  `- 1:0:3:4 sdr 65:16 active ready running
```

Prepare the Storage Scale Host

Host OS & Kernel Update

The Linux kernel and OS release version must meet the minimum requirement specified in the Storage Scale installation guide. The user is encouraged to read the IBM Storage Scale FAQ for the specific release version at https://www.ibm.com/docs/en/spectrum-scale.

In our test, Storage Scale 5.1.3 was installed over Red Hat. For complete IBM Spectrum_Scale_DM_513_x86_64_LNX.tar installation instructions, please refer to IBM documentation at https://www.ibm.com/docs/en/spectrum-scale/5.1.3?topic=quick-reference.

For this installation the following kernel version and tools must exist.
If an error such as "`error: Cannot find a valid kernel header file, the file is not at expected location`" occurs during the Storage Scale installation, we recommend a Linux kernel update. `yum` can be used to update the kernel and install the tool utilities as shown in the following example.

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

```
# yum -y install kernel-devel cpp gcc gcc-c++ kernel-headers
# yum install ksh perl m4 net-tools -y
```

Note that "yum install" may not work properly if the local host Red Hat repo is not configured correctly or some files in `/etc/yum.repos.d` are missing or not updated. The following files must be updated under the yum.repos.d directory before running `yum update` successfully.

```
[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
```

Host Shell Environment

Storage Scale requires that **bsh** is running for a successful Storage Scale installation. Do the following to ensure that **bsh** is under the correct user environment. If **bsh** is not running correctly, perform an update to export the path correctly. An example is shown for reference.

```
[root@smc10 fioTemplete]# 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
export PATH=$PATH:$HOME/bin:/usr/lpp/mmfs/bin
export WCOLL=/nodes

#export PATH
```

Host FQDN

Storage Scale requires that each NSD node in the cluster has a FQDN (full qualified domain name) so all NSD nodes can communicate with each other and the storage resources can later be exported through its global name.

Note: Due to the lack of AD (Active Directory) or LDAP DNS services in the TME lab environment, we bypassed this requirement by using the local host file to assign an ASIIC name to each host in the cluster.

1. Assign a name to the host in the CLI by typing:
`dhcp-192-168-53-197:~ # hostnamectl set-hostname smc10`

```
[root@smc10 ~]# hostnamectl
   Static hostname: smc10
         Icon name: computer-server
           Chassis: server
        Machine ID: 6fcba1bb8211429abaf6ab1544add71d
           Boot ID: 99dcfcf7c9db41d9b8788ea4af542b11
  Operating System: Red Hat Enterprise Linux 8.6 (Ootpa)
       CPE OS Name: cpe:/o:redhat:enterprise_linux:8::baseos
            Kernel: Linux 4.18.0-305.25.1.el8_4.x86_64
      Architecture: x86-64
```

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

```
[root@smc10 ~]# cat /etc/hosts
127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6
127.0.0.1 localhost
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
192.168.53.225  smc11
192.168.53.223  smc12
192.168.53.195  smc13
192.168.53.252  smc14
192.168.53.220  smc15
192.168.68.27   smc101
192.168.68.253  cesip
[root@smc10 ~]#
```

Host Passwordless SSH Access

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

3. Check if the host `sec_id rsa` file exists.

```
[root@smc10 ~]# ll ~/.ssh/id_*
-rw-------. 1 root root 3369 Sep  9 17:57 /root/.ssh/id_rsa
-rw-r--r--. 1 root root  735 Sep  9 17:57 /root/.ssh/id_rsa.pub
```

4. 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:~ #
```

5. 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
```

At the same time edit the entries in the `/etc/ssh/sshd_config` to read as follows:

PasswordAuthentication **no**

ChallengeResponseAuthentication **no**

UsePAM **no**

Restart the **ssh** process

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

6. Verify that SHH using the host node name works.

```
[root@smc10 ~]# ssh smc11
Activate the web console with: systemctl enable --now cockpit.socket

Register this system with Red Hat Insights: insights-client --register
Create an account or view all your systems at https://red.ht/insights-dashboard
Last login: Tue Nov 22 12:36:55 2022 from 10.127.35.241
[root@smc11 ~]#
```

Building the Storage Scale Server

1. Install Storage Scale rpm on the primary and secondary host nodes.

```
# 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 gpfs installation files are located in /usr/lpp/mmfs. Find the gpfs_rpms directory and install the required rpms. For minimum installation you will need to install base, ext, gskit, gpl, msg, and docs.

```
# cd /usr/lpp/mmfs/5.1.3.0/gpfs_rpms
```

```
# rpm -ivh gpfs.{base,ext,gskit,gpl,msg,docs}*.rpm
```

Note: For NFS protocol export service, the following rpms are required:

```
# rpm -ivh gpfs.{nfs-ganesha-*, gpfs.nfs-ganesha-debuginfo-*,nfs-ganesha-gpfs-*, nfs-ganesha-utils-*}.rpm
```

a. Build the Storage Scale portable layer.

This is an executable, portable package that installs Storage Scale on a host node automatically. Create this portable layer as shown.

```
[root@smc10 ~]# /usr/lpp/mmfs/bin/mmbuildgpl
-------------------------------------------------------
mmbuildgpl: Building GPL (5.1.3.0) module begins at Fri Dec  2 15:53:30 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 Fri Dec  2 15:53:46 PST 2022.
-------------------------------------------------------
[root@smc10 ~]#
```

2. Add Storage Scale in the user path environment to ensure that its related CLI command works.

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

b. Export `PATH=$PATH:$HOME/bin:/usr/lpp/mmfs/bin`.

c. Validate that the cluster is installed correctly using the # mmlscluster command. If the user receives the following outputs, the Storage Scale installation is completed.

```
[root@smc10 ~]# mmlscluster

GPFS cluster information
========================
  GPFS cluster name:         cluster01.gpfs
  GPFS cluster id:           15064223649192720658
  GPFS UID domain:           cluster01.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   smc10            192.168.53.198  smc10            quorum-manager
   2   smc11            192.168.53.225  smc11            quorum-manager
   3   smc12            192.168.53.223  smc12
```

Configuration of the Storage Scale Cluster Node and NSD

To make the node and NSD configuration easy, we recommend that the user creates a Storage Scale node list and NSD stanza. Once created, the user can use these stanzas to expedite the Storage Scale deployment. A node list stanza example is shown below.

```
[root@smc10 gpfs]# cat nodelist
smc10:quorum-manager
smc11:quorum-manager
smc12:
```

After the node list is created, you can create a Storage Scale cluster named `cluster01.gpfs` as shown to pass in the node list onto the Storage Scale cluster.

```
# mmcrcluster -C cluster01.gpfs  -N nodeList -p smc10 -s smc11
```

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

Accept the license agreement and add the cluster license. The user only needs to do this on either the primary node or secondary node, and does not need to add the license key on each node in the cluster.

```
/usr/lpp/mmfs/bin/mmchlicense server --accept -N smc10, smc11, smc12
```

Start the Storage Scale Cluster

Since our Storage Scale test is conducted in a non-production environment, we disabled the firewall on the cluster nodes to prevent potential network security issues. Follow the steps below to start the cluster.

Enter the command `# mmstartup -a`.

Enter the command `# mmlscluster` to verify that the Storage Scale cluster can be started and is in a running state. Once the cluster is brought up the first time, the Storage Scale cluster may go through a period of "**arbitrating**" for a minute or two.

```
[root@smc10 gpfs]# mmlscluster

GPFS cluster information
========================
  GPFS cluster name:         cluster01.gpfs
  GPFS cluster id:           15064223649192720658
  GPFS UID domain:           cluster01.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   smc10            192.168.53.198  smc10            quorum-manager
   2   smc11            192.168.53.225  smc11            quorum-manager
   3   smc12            192.168.53.223  smc12
```

Enter the command `# mmgetstate -L -a`.

```
[root@smc10 gpfs]# mmgetstate -L -a

 Node number  Node name  Quorum  Nodes up  Total nodes  GPFS state    Remarks
-------------------------------------------------------------------------------
          1  smc10        2        2          3      active        quorum node
          2  smc11        2        2          3      active        quorum node
          3  smc12        2        2          3      active
```

All cluster nodes should be listed as "active" if they are working correctly. If they're stuck arbitrating for longer than a minute or two, check is the passwordless ssh, as that could cause the arbitrating. Also, and this is counterintuitive, every cluster node must be able to SSH into itself without a password. Therefore, make sure that passwordless access is working. It is a best practice to shut down the cluster before conducting any cluster level changes and reconfiguration. To do so enter the shutdown command: **`# mmshutdown -a`**

Configure NSD

The NSDs are storage building blocks that Storage Scale uses to store user data and metadata. Once you have the storage devices ready, it's time to create NSD stanza.

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 following example, NSDs can be created upon system raw devices such as `sdc`, `sdd` and `sde`, etc.

```
[root@smc10 gpfs]# lsblk
NAME                                    MAJ:MIN RM  SIZE RO TYPE  MOUNTPOINT
sda                                       8:0    0  1.8T  0 disk
├─sda1                                    8:1    0  600M  0 part  /boot/efi
├─sda2                                    8:2    0    1G  0 part  /boot
└─sda3                                    8:3    0  1.8T  0 part
  ├─rhel-root                           253:1    0   70G  0 lvm   /
  ├─rhel-swap                           253:3    0    4G  0 lvm   [SWAP]
  └─rhel-home                           253:10   0  1.7T  0 lvm   /home
sdb                                       8:16   0  1.8T  0 disk
sdc                                       8:32   0 83.4T  0 disk
└─3600c0ff000f48b7f4da33c6303000000     253:4    0 83.4T  0 mpath
  └─3600c0ff000f48b7f4da33c6303000000p1 253:13   0 83.4T  0 part
sdd                                       8:48   0 83.4T  0 disk
└─3600c0ff000f48b7f4da33c6304000000     253:5    0 83.4T  0 mpath
  └─3600c0ff000f48b7f4da33c6304000000p1 253:14   0 83.4T  0 part
sde                                       8:64   0 83.4T  0 disk
└─3600c0ff000f48b7f4da33c6305000000     253:6    0 83.4T  0 mpath
  └─3600c0ff000f48b7f4da33c6305000000p1 253:15   0 83.4T  0 part
sdf                                       8:80   0 83.4T  0 disk
└─3600c0ff000f48b7f4da33c6306000000     253:7    0 83.4T  0 mpath
  └─3600c0ff000f48b7f4da33c6306000000p1 253:16   0 83.4T  0 part
sdg                                       8:96   0 83.4T  0 disk
└─3600c0ff000f48b2944a33c6305000000     253:8    0 83.4T  0 mpath
  └─3600c0ff000f48b2944a33c6305000000p1 253:17   0 83.4T  0 part
```

Alternatively they can be built on top of a multipath device such as `mpathbm` as shown in the example below.

```
[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
```

The NSD stanza file is created to more easily manage and create the NSD. Alternatively you can manually create individual NSDs as shown.

```
#cat NSDstanza
```

```
[root@smc10 gpfs]# cat nsdstanza
#nsd for smc10
%nsd: nsd=nsd01 device=/dev/dm-0 servers=smc10 failureGroup=1
%nsd: nsd=nsd02 device=/dev/dm-2 servers=smc10 failureGroup=1
%nsd: nsd=nsd03 device=/dev/dm-4 servers=smc10 failureGroup=1
%nsd: nsd=nsd04 device=/dev/dm-5 servers=smc10 failureGroup=1
%nsd: nsd=nsd05 device=/dev/dm-6 servers=smc10 failureGroup=1
%nsd: nsd=nsd06 device=/dev/dm-7 servers=smc10 failureGroup=1
%nsd: nsd=nsd07 device=/dev/dm-8 servers=smc10 failureGroup=1
%nsd: nsd=nsd08 device=/dev/dm-9 servers=smc10 failureGroup=1

#nsd for smc11
%nsd: nsd=nsd11 device=/dev/dm-3 servers=smc11 failureGroup=1
%nsd: nsd=nsd12 device=/dev/dm-4 servers=smc11 failureGroup=1
%nsd: nsd=nsd13 device=/dev/dm-5 servers=smc11 failureGroup=1
%nsd: nsd=nsd14 device=/dev/dm-6 servers=smc11 failureGroup=1
%nsd: nsd=nsd15 device=/dev/dm-7 servers=smc11 failureGroup=1
%nsd: nsd=nsd16 device=/dev/dm-8 servers=smc11 failureGroup=1
%nsd: nsd=nsd17 device=/dev/dm-9 servers=smc11 failureGroup=1
%nsd: nsd=nsd18 device=/dev/dm-10 servers=smc11 failureGroup=1
```

Notice that we used `/dev/dm-xx` device (raw devices) here when we created Storage Scale NSDs. The underlying reason for creating NSDs on a raw device is that, content on the dm-multipath man page on Linux suggests that the user use the multipath device alias such as `/dev/mapper/mpathX`.

Note: `pathX` is a multipath-capable device here and is managed by Linux Device Mapper Multipath (DMM). The multipath devices are the only ones guaranteed to remain boot-consistent on Linux because `/dev/dm-x` devices re-enumerated themselves each time the system reboots and the device names could be changed when formatting is performed on them.

Storage Scale recognizes the `mpath` device names are symbolic links to the `/dev/dm-x` devices as shown below.

```
[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
```

The official documentation from the IBM Storage Scale site says to use the `/dev/dm-x` devices instead and to leave the device IO multipathing for Storage Scale Native RAID to handle its own disk multi-pathing algorithm. See the IBM link here regarding this matter: https://www.ibm.com/docs/en/spectrum-scale/4.2.3?topic=issues-gpfs-is-not-using-underlying-multipath-device

For simplicity, we used `/dev/dm-x` instead in this POC test. The following outputs show that Storage Scale thinks the devices are type DMM, which is Linux DMMs, rather than the multipath device Storage Scale recognizes.

```
[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
```

Further research and more testing are required to work out a better solution to this issue.

Create NSDs

1. Enter the CLI command **`# mmcrnsd -F NSDstanza`**. If the NSDs were part of a previous pool, you can add the new NSDs with the **`-v`** option to overwrite the previous ones.
2. List the NSDs by entering the CLI command **`# mmcrnsd -F NSDstanza -v no`**. Since the file system hasn't been created yet, they should all be listed as "free disk."
3. Enter the command `# mmlsnsd`

```
[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
```

4. Configure one of the disks as a "tie breaker" disk to avoid a split-brain condition by entering the CLI command **`# mmchconfig tiebreakerDisks="nsd247"`**
5. Un-configure a disk as a tie breaker disk by entering the CLI command

   `# mmchconfig tiebreakerDisks=""`
6. Delete the configured NSDs by entering the CLI command

   `# mmdelnsd -F NSDstanza`
7. Delete an individual NSD by entering the CLI command .

   `# mmdelnsd nsd01`

Create and Format the Storage Scale File System

This process creates the Storage Scale on the NSD devices and formats the file system using the user-defined configuration `# 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` - Formats with a 1M block size.

`-m 2` - Sets the default number of metadata replicas to two.

`-M 2` - Sets the max number of metadata replicas to two.

`-r 2` - Sets the default number of data replicas to two.

`-R 2` - Sets the max number of data replicas to two.

`-n 32` - Sets the estimated number of clients to 32. Formats the file system with the correct degree of parallelism.

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

1. Verify that the file system was created properly using this command to list the file system created with all file system parameter details.

   # mmlsfs cv01

```
[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
```

2. Mount the file system using

   # mmmount all -a

3. Verify that the file system has been mounted correctly. Check disk space on every node in the Storage Scale cluster using the CLI command **# df -kh**

```
[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 on all participating Storage Scale nodes in the cluster. The example shows that **cv01** is mounted at **/gpfs/cv01** on this particular 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
```

4. Check the replication settings for your file system.

```
# 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 clients)

1. Add a Client.

   If you need to add another node, go through the normal installation procedure on the new Node and then run mmadnode to add the node to the cluster.

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

2. Delete a Client using the CLI command

   ```
   # mmdelnode -n client1
   ```

3. Change a server's role using either of these CLI commands:

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

TROUBLESHOOTING

This section explores the procedures to find Storage Scale system logs. The first thing when running into an issue is to consult the logs from both the Storage Scale layer and the host OS layer to determine the next step in troubleshooting.

Most Storage Scale log files are stored in `/var/adm/ras/mmfs.log.latest`. There is one on everyparticipating Storage Scale node and this is where the troubleshooting begins.

```
[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, 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 our testing, we included links to the closest available release (ver. 5.0.0 above).

PERFORMANCE

Performance Considerations

Good Storage Scale performance is subject to, and defined by, the host IO's subsystem performance optimization and the optimization at the Storage Scale level. The lack of either one will cause Storage Scale performance to suffer. In this section we discuss some of the best practices in optimization of raw storage devices and leave most of the Storage Scale best performance practices to IBM subject matter experts.

Some of the performance-related tuning parameters are listed below, and we recommend that you consult IBM professional services for Storage Scale performance optimization.

Performance Tuning on Storage Scale

Storage Scale now groups some of the performance tuning under system quality of service (QoS). Run the following commands to tune Storage Scale for Seagate CORVAULT systems. You can run them from either the primary node or the secondary node in the Storage Scale cluster.

"Nodelist" in the example is the NSD node list to use when creating Storage Scale cluster nodes.

```
# mmchconfig maxMBps=10000 -N nodelist
```

maxMBpS affects the depth of prefetching for sequential file access. It's similar to queue depth but at the Storage Scale GNR (Storage Scale Native RAID) level. This number should be set at least as large as the maximum expected hardware bandwidth.

```
# mmchconfig worker1Threads=1024 -N nodelist
```

maxFilesToCache should be set fairly large to assist with local workload. It can be set very large in small client clusters, but should remain small on clients in large clusters to avoid excessive memory use on the token servers. The stat cache is not effective on Linux, so it should always be small.

```
# mmchconfig maxReceiverThreads=128 -N nodelist
```

This command determines the RDMA (Remote Direct Memory Access) port buffer size.

```
# mmchconfig nsdMaxWorkerThreads=2048 -N nodelist
```

The maximum number of NSD threads on an NSD server that concurrently transfers data with NSD clients.

```
# mmchconfig nsdMinWorkerThreads=128 -N nodelist
```

The minimum number of NSD threads on an NSD server that concurrently transfers data with NSD clients.

```
# mmchconfig nsdMultiQueue=512 -N nodelist
```

Sets the queue depth on NSD devices on the cluster nodes.

```
# mmchconfig nsdSmallThreadRatio=1 -N nodelist
```

The ratio of the number of small threads to the number of large threads. The recommendation is to change this to two for most workloads.

```
# mmchconfig prefetchAggressiveness=1 -N nodelist
```

PrefetchAggressiveness defines how aggressive to prefetch data. It has four levels defined as:

- 0 means never prefetch
- 1 means prefetch on second access if sequential
- 2 means prefetch on first access at offset 0 or second sequential access anywhere else
- 3 means prefetch on first access anywhere

Storage Scale has to be re-started after the tuning. To restart Storage Scale, enter the following CLI commands:

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

Wait until all Storage Scale nodes are active, then mount the file system using

```
# mmmount all -a
```

Performance Tuning on Multipath Device

Now we further explore performance optimization on the CORVAULT and host OS IO subsystem. We recommended that Storage Scale host storage device IO parameters be checked to ensure consistency with IO device queue depth and scheduler characteristics match the type of storage media used in the test.

The user can use these CLI commands to verify and update the parameters to fit performance needs in the above-mentioned areas. We believe that without a satisfactory performance on the multipath devices, it would be difficult to achieve acceptable Storage Scale performance.

Determine the device names:

```
[root@smc10 queue]# multipath -ll
3600c0ff000f48b2944a33c6308000000 dm-2 SEAGATE,6575
size=83T features='1 queue_if_no_path' hwhandler='1 alua' wp=rw
|-+- policy='queue-length 0' prio=50 status=active
| |- 1:0:9:11  sdj  8:144  active ready running
| `- 1:0:10:11 sdz  65:144 active ready running
`-+- policy='queue-length 0' prio=10 status=enabled
  |- 1:0:11:11 sdr  65:16  active ready running
  `- 1:0:12:11 sdah 66:16  active ready running
3600c0ff000f48b7f4da33c6306000000 dm-7 SEAGATE,6575
```

Device scheduler:

```
[root@smc10 queue]# cat /sys/block/dm-2/queue/scheduler
[mq-deadline] kyber bfq none
```

CFQ (cfq): The default scheduler for many Linux distributions; it places synchronous requests, submitted by processes, into a number of per-process queues and then allocates time slices for each of the queues to access the disk.

Noop scheduler (noop): The simplest I/O scheduler for the Linux kernel based on the First In First Out (FIFO) queue concept. This scheduler is best suited for SSDs.

Mq-Deadline scheduler (deadline): Attempts to guarantee a start service time for a request.

Device queue depth:

```
[root@smc10 queue]# cat /sys/block/dm-7/queue/nr_requests
256
```

In the Storage Scale test we used mq-Deadline scheduler and a default queue depth of 256 for each multipath device.

Performance Benchmark tool

In benchmarking the raw storage device, we selected the FIO benchmark tool for Linux. The parameters for 4-pillar performance evaluation on a raw device are sequential read and write, random read and write, and mixed workload (30/70 write vs read). The FIO parameters are kept consistent on all the cluster nodes (primary and secondary node) that host Storage Scale NSDs.

We listed the FIO configuration parameters for the five types of workloads we tested. However, these parameters may produce different results depending on the hardware and software used. We suggest that the user contact a Seagate Systems professional if they encounter any performance issues in their exploration of Seagate storage.

Sequential write

```
[root@smc10 fioTemplete]# cat sequential-write.fio
[global]
bs=65536
direct=1
ioengine=libaio
randrepeat=0
time_based=1
runtime=300
filesize=40G
numjobs=1
rw=write
#rwmixread=70
name=smC10_seqwrite
group_reporting=1


[smC10-job1]
iodepth=64
filename=/dev/dm-0
```

Sequential read

```
[root@smc10 fioTemplete]# cat sequnetial-read.fio
[global]
bs=65536
direct=1
ioengine=libaio
randrepeat=0
time_based=1
runtime=300
filesize=40G
numjobs=1
rw=read
#rwmixread=70
name=smC10_seqread
group_reporting=1


[smC10-job1]
iodepth=64
filename=/dev/dm-0
```

Random write

```
[root@smc10 fioTemplete]# cat random-write.fio
[global]
bs=65536
direct=1
ioengine=libaio
randrepeat=0
time_based=1
runtime=300
filesize=40G
numjobs=1
rw=randrw
rwmixread=0
name=smC10_randrw
group_reporting=1


[smC10-job1]
iodepth=64
filename=/dev/dm-0
```

Random read

```
[root@smc10 fioTemplete]# cat random-read.fio
[global]
bs=65536
direct=1
ioengine=libaio
randrepeat=0
time_based=1
runtime=300
filesize=40G
numjobs=1
rw=randread
#rwmixread=70
name=smC10_randread
group_reporting=1


[smC10-job1]
iodepth=64
filename=/dev/dm-0
```

Mixed workload

```
[root@smc10 fioTemplete]# cat mixedload.fio
[global]
bs=65536
direct=1
ioengine=libaio
randrepeat=0
time_based=1
runtime=300
filesize=40G
numjobs=1
#rw=randrw
rwmixread=70
name=smC10_randrw
group_reporting=1


[smC10-job1]
iodepth=64
filename=/dev/dm-0
```

Ready to Learn More?

Visit us at **www.seagate.com**

seagate.com

SEAGATE