IBM Elastic Storage System 3000
Version 6.0.0.2

Quick Deployment Guide

GC28-3106-03

Note

Before using this information and the product it supports, read the information in "Notices" on page 39.

This edition applies to version 6 release 0 modification 0 of the following product and to all subsequent releases and modifications until otherwise indicated in new editions:

- IBM Spectrum® Scale Data Management Edition for IBM® ESS (product number 5765-DME)
- IBM Spectrum Scale Data Access Edition for IBM ESS (product number 5765-DAE)

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

Contents

Tables

About this information

This information is intended as a guide for administering IBM Elastic Storage® System (ESS) 3000.

Who should read this information

This information is intended for administrators of IBM Elastic Storage System (ESS) 3000 systems that include IBM Spectrum Scale RAID.

Related information

Related information

For information about:

- IBM Spectrum Scale, see:

  http://www.ibm.com/support/knowledgecenter/STXKQY/ibmspectrumscale_welcome.html
- mmvdisk command, see mmvdisk documentation.

Conventions used in this information

Table 1 on page vii describes the typographic conventions used in this information. UNIX file name conventions are used throughout this information.

Table 1. Conventions

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

Table 1. Conventions (continued)

Convention	Usage
****	In command examples, a backslash indicates that the command or coding example continues on the next line. For example: `mkcondition -r IBM.FileSystem -e "PercentTotUsed > 90" \` `-E "PercentTotUsed < 85" -m p "FileSystem space used"`
{item}	Braces enclose a list from which you must choose an item in format and syntax descriptions.
[item]	Brackets enclose optional items in format and syntax descriptions.
<Ctrl-*x*>	The notation <Ctrl-*x*> indicates a control character sequence. For example, <Ctrl-c> means that you hold down the control key while pressing <c>.
item...	Ellipses indicate that you can repeat the preceding item one or more times.
|	In *synopsis* statements, vertical lines separate a list of choices. In other words, a vertical line means *Or*. In the left margin of the document, vertical lines indicate technical changes to the information.

How to submit your comments

To contact the IBM Spectrum Scale development organization, send your comments to the following email address:

`scale@us.ibm.com`

Chapter 1. ESS 3000 contents

ESS 3000 version 6.0.0.2 content stack

Component	Version
IBM Spectrum Scale	5.0.4.3 efix 2
Red Hat Enterprise Linux®	8.1 (5141-AF8) The container, and 5148-21L and 5148-22L nodes run Red Hat Enterprise Linux 7.7.
OFED	MLNX_OFED_LINUX-4.7-3.2.9.0 OFED firmware levels: • MT27500 = 10.16.1020 • MT4099 = 2.42.5000 • MT26448 = 2.9.1326 • MT4103 = 2.42.5000 • MT4113 = 10.16.1020 • MT4115 = 12.25.1020 • MT4117 = 14.25.1020 • MT4119 = 16.26.6000 • MT4120 = 16.26.6000 • MT4121 = 16.26.6000 • MT4122 = 16.26.6000
Kernel	4.18.0-147.5.1.el8_1
NVMe drive firmware	SN1ISN1I
Boot drive firmware	1236
Network adapter firmware	16.26.6000
Storage firmware	1111

ESS 3000 version 6.0.0.2 editions

Note: The package version mentioned in this document might be different than the version of the installation package available at IBM FixCentral.

The ESS 3000 software is available in two editions:

- Data Management Edition

  `ess3000_6.0.0.2_0417-01_dme.tgz`
- Data Access Edition

  `ess3000_6.0.0.2_0417-01_dae.tgz`

Changes from ESS 3000 version 6.0.0.1

- IBM Spectrum Scale version 5.0.4.3 efix 2
- Kernel version 4.18.0-147.5.1.el8_1

Fixes and improvements in ESS 3000 version 6.0.0.2

- Support for online Disk Capacity upgrades
- Enhancements to online and offline upgrade modes
- General bug fixes and improvements

Security law changes

- New systems and switches shipped from manufacturing now have either an expired password or one set to the serial number of the component.
- You must take input from the customer before deployment starts and change the desired passwords.
- The default root password for the OS is `ibmesscluster`. You are required to change it upon first login. This password must be set the same on each node.
- The default ASMI passwords (login, IPMI, HMC, etc.) are set to the serial number of the server. IPMI must be the same on each node.
- If the 1Gb Cumulus switch is shipped racked, the default password is the serial number (S11 number - label found on the back of the switch). If the switch is shipped unracked, you are required to set the password upon first login. The default password is `CumulusLinux!` but you will be prompted to change the password upon first login. If you have any issues logging in or you need help in setting up a VLAN with the switch, consult this documentation link.
- You must set all required passwords before the deployment begins.

Chapter 2. ESS 3000 example setup flow

Here is a high-level overview of the setup process for new ESS 3000 customers.

- TDA complete and system ordered
- Customer provides network worksheet
- System arrives on-site
- SSR uses the *ESS 3000 Hardware Planning and Installation Guide* to do the following tasks:
  - Rack system and apply power
  - Setup VLAN on the switch and change the password (if needed)
  - Run network cables (at minimum the management network)
  - Check the EMS for hardware issues and set management IP
  - Check the ESS 3000 Canisters for hardware issues, set the management IPs, perform ping tests

  The system is handed over to the customer.
- Customer changes root password on each node (must be the same)
- Customer sets up `/etc/hosts` file
- Customer extracts code in `/home/deploy`, accepts license, and installs the image
- Customer modifies the `yml` file
- Customer runs container (now inside)
- Customer configures environment and sets up passwordless SSH
- Customer runs installation check to determine if upgrade is needed
- Customer upgrades canisters in parallel (offline), if required
- Customer creates network bonds, cluster, and file system
- Customer leaves container (now on EMS) and sets up performance collection, GUI, and call home
- Customer performs final health checks
- Customer configures clients, protocol services etc and begins to use the system

Chapter 3. ESS 3000 best practices

- Upgrades must not be performed on an unhealthy system. Make sure that the following checks are clean before starting the upgrade:
  - **mmhealth node show -a --unhealthy**
  - **gnrhealthcheck**
  - **essinstallcheck -N localhost** (from each canister node)
  - IBM Spectrum Scale GUI is free of any issues.
- A clean network fabric is key. Consider the following (if possible):
  - Always keep the switch firmware updated.
  - Reboot the switch if it has been up for a long time.
  - Run fabric checks, such as **ibdiagnet** and **nsdperf**, periodically.
- If you have quorum set on an ESS 3000 node, both canisters must have the same attribute:
  - If Canister A is quorum node so must be Canister B.
  - If Canister A is non-quorum node, so must be Canister B.
- All ESS 3000 nodes must be at the same version. If you are adding additional nodes, upgrade the existing nodes to the latest version first before adding the new nodes.
- Management switch must be an isolated, flat network or VLAN(s).
- All canister root passwords must be changed and must be exactly the same before starting setup or upgrade.
- Tracing must be disabled on the GPFS cluster before performing an upgrade.

```
mmtracectl --stop
mmtracectl --off
```

- Ensure that you have enough space on the EMS node, especially in the partition where the installation package (`.tgz` file) is stored and extracted, prior to deployment. You need about 20GB of space to complete the container installation process.
- Turn on syslog redirection after installation or upgrade is complete. Doing this, centralizes all syslog files to the EMS node in the `/var/log/messages` directory. For more information, see *ESS 3000 deployment troubleshooting: Helpful podman, Ansible, and log information* in *ESS 3000: problem Determination Guide*.
- Enable chroynd (NTP) after installation or upgrade is complete. For more information, see Appendix C, “How to set up chronyd (NTP),” on page 35.

Chapter 4. ESS 3000 and ESS for Power considerations

The following lists describe the best practices and support statements when mixing ESS 3000 and ESS for Power®. The minimum configuration for ESS 3000 is as follows:

- 8247-21L Power 8 EMS running ESS 5.3.5 or later
- 1 or more ESS 3000 nodes
- 1/10Gb management switch with two isolated, flat VLANs (management and service)
- 1 high-speed switch (Ethernet or Infiniband)

The EMS node plays a few critical roles:

- Runs IBM Spectrum Scale and serves as a quorum node in smaller configurations
- Runs the IBM Spectrum Scale GUI
- Runs ESA for call home
- Serves as the location where the container runs for ESS 3000 (using podman)

An additional C10-T2 connection is added to the 1/10Gb switch (management VLAN) to support the container.

- The EMS node must have podman installed to run the ESS 3000 container.
- The EMS node is where the GUI, the ESA (call home), and the additional quorum function run from.
- If EMS node + ESS 3000 only:
  - The management switch does not have to be set up as a VLAN bust is a best practice to do so for service and management networks. Either way, the management switch must be on an isolated, flat network.
  - The EMS node must be at version 5.3.5 or later to support running the ESS 3000 container. You must upgrade the EMS node from the container so that IBM Spectrum Scale and other components are updated.
- If EMS node + ESS 3000 + ESS or protocol nodes:
  - The EMS node must be at ESS 5.3.5 or later and updated from container so that IBM Spectrum Scale and other components are updated.
  - The ESS or protocol nodes do not have to be upgraded to the latest ESS version but it is recommended to do so.
  - If you are upgrading the ESS or protocol nodes to the latest version, use the legacy flow to also upgrade the EMS (thus skip upgrading the EMS from container).
  - Update the ESS or protocol nodes, if wanted, before the ESS 3000.
  - The switch must have 2 isolated VLANs (management, service) to support this configuration.
  - Refer to the ESS 5.3.5x documentation on how to upgrade the EMS, ESS, or protocol nodes to the latest version.

Chapter 5. ESS 3000 common setup instructions

The IBM Elastic Storage System 3000 (ESS 3000) installation package, which is a compressed file, contains a podman container with necessary key components such as RHEL 8.x, IBM Spectrum Scale RAID, MOFED, and firmware for various components of ESS 3000. The following sections describe the common tasks that need to be done for running an ESS 3000 software version. This includes upgrading from an existing container or running one for the first time.

Note:

- All version numbers, host names, IP addresses that are used in the following sections are examples.
- The root password is set to expire upon first login by using the default password **ibmesscluster**. Change the password on each server.
- Make sure that you use the correct edition (Data management (DME) or Data Access Edition (DAE) that you are entitled to. This is especially important for upgrades because you must use the same edition that was previously installed.

Note: The code level applied on the system in manufacturing is located in the `/home/deploy` on the EMS node. Compare the latest version of the ESS 3000 installation package available on the IBM FixCentral ESS 3000 version 6.0.0 page to the one in the `/home/deploy` directory on the EMS node. If the one on IBM FixCentral is newer, download and use that version.

How to identify the version that is currently installed

For new customers, the ESS 3000 installation package (`.tgz` file) is located in `/home/deploy` on the EMS node. This file name contains dae or dme depending on the edition. For example: `ess3000_6.0.0.2_0417-01_dme.tgz`. This file is used to deploy the system in manufacturing. To verify the version installed on each canister, use these steps:

1. SSH to the canister.
2. Run this command: **`essinstallcheck -N localhost`**

At the top the command output, the installed version is displayed. For example:

```
Installed version: ess3000_6.0.0.2_0417-01_dme
```

Go to IBM FixCentral and determine the latest ESS 3000 version. If it is newer than the installed version on your canisters, download and replace the `.tgz` file in `/home/deploy` of the EMS node. Use this file to install or upgrade your system.

Do the following steps if you are doing a new installation or an upgrade of an ESS 3000 system.

Configuring the EMS node for ESS 3000

In preparation for upgrading to a new version, it is advised to back up the current configuration first. This step is only recommended if you are upgrading from an existing ESS 3000 version. The backup allows you to restore the node configuration information and SSH keys when running the new container. When upgrading to a new version you can use this backup to restore your configuration.

1. **[Upgrade only]** Back up the current configuration as follows.
   a. Create a backup directory on the EMS node.

      ```
      mkdir -p /home/backup/6001/xcatdb
      ```

      Note: The directory listed here is an example. You can create any directory under `/home/backup` for this purpose.
   b. Enter the current container.
      1) Determine the current container image.

```
podman  ps -a
```

Example output:

```
[root@ems1]/home/backup/0417/xcatdb# podman ps -a
CONTAINER ID  IMAGE                                  COMMAND
CREATED       STATUS           PORTS  NAMES
198830dd1027  localhost/ess3000_6.0.0.2_dae:0417-01  /myStartupScript....  10 hours
ago  Up 10 hours ago         cems0
```

In this example, the container name is `cems0` and the status is Up. If the status is not Up, you can start the container by using the **podman start cems0** command.

2) Attach the container.

```
podman attach cems0
```

An alternative way to enter the container is to locate the container directory for the installed version and run **essmgr -r**. For example,

```
cd /home/deploy/ess3000_6.0.0.2_0417-01_dme.dir
./essmgr -r
```

By using either of these methods, you can enter the container.

If the container was stopped at the time, you must re-run one of the following configuration scripts depending on the version, once you are inside the container.

- Version 6.0.0: `/tmp/configure_cems.sh`
- Version 6.0.0.1 or later: `/tmp/restart_cems_services.sh`

c. Back up the xCAT configuration and keys.

```
dumpxCATdb -p /home/backup/6001/xcatdb

Backup complete
```

```
cp -a /etc/xcat/hostkeys/ /home/backup/6001/hostkeys
# ls -l /home/backup/6001/

total 4
drwxr-xr-x. 2 root root  166 Feb 25 03:26 hostkeys
drwxr-xr-x. 2 root root 4096 Feb 25 19:46 xcatdb
```

d. Type `exit` and then press `enter` to exit the container. This step stops the container.

You are now in the EMS node.

2. **[Upgrade only]** Clean up the existing environment. Make sure that you have enough space to extract the contents of the compressed file. You might need to clean up old large files in `/root` or in other directories in the EMS node. You need roughly 20 GB free space on the partition on which you plan to extract the compressed file.

a. Remove the current container as follows.

1) List the containers.

```
podman ps -a
```

2) Remove any containers that are listed.

```
podman rm ContainerName
```

Note: If a container is not currently in the `Exited` state, stop it by using the **podman stop** *ContainerName* command. Then, you can remove the container.

b. Remove the installed images as follows.

1) List the installed images.

```
podman images
```

2) Remove any images that are listed.

```
podman image rm ImageID -f
```

3. **[Upgrade or new installation]** Extract, verify, and run the new ESS 3000 container software.

   a. Expand the compressed ESS 3000 installation package that is located in /home/deploy on the EMS node. This is the file that either came from manufacturing or that was downloaded because a higher version was available. For more information, see "How to identify the version that is currently installed" on page 9.

   The name of the package is in this format: ess3000_6.0.0.2_0417-01_dme.tgz.

```
tar zxvf ess3000_6.0.0.2_0417-01_dme.tgz
```

   The contents of the installation package, before the license is accepted, are:

```
ess3000_6.0.0.2_0417-01_dme.sh
ess3000_6.0.0.2_0417-01_dme.sh.sha256
```

   The **tar** command extracts the contents of the compressed file into a directory under the directory where the extraction is done. In this example, the directory where the extraction is done is: /home/deploy.

   b. Verify the checksum of the ESS 3000 installation package.

```
sha256sum -c ess3000_6.0.0.2_0417-01_dme.sh.sha256

ess3000_6.0.0.2_0417-01_dme.sh: OK
```

   c. Accept the license and install the accepted image.

```
./ess3000_6.0.0.2_0417-01_dme.sh  --text-only --install-image
```

   You are presented the license acceptance prompt. Type 1 and press Enter to accept the license. A directory is created after the acceptance of license. The contents of the directory are:

```
ess3000_6.0.0.2_0417-01_dme.dir
├── ess3000_6.0.0.2_0417-01_dme.tar
├── essmgr
├── essmgr.yml
├── podman.tgz
└── Release_note.ess3000_6.0.0.2_0417-01_dme
```

   After you accept the license, the container image is installed on the node.

   Note: In ESS 3000 version 6.0.0.2, podman version 1.4.4 must be used and it is bundled in the installation package. After the installation package is extracted, podman.tgz is located in the /home/deploy/ess3000_6.0.0.2_0417-01_dme.dir directory on the EMS node in this example scenario. Perform the following steps to extract and install podman.

```
tar zxvf podman.tgz

cd podman
yum -y install *rpm
```

   If you cannot find podman.tgz and do not have podman installed (verify by using **rpm -qa | grep -i podman**), contact IBM service.

   When the installation is completed, a podman image list output similar to the following output is displayed.

```
# podman images
REPOSITORY                      TAG       IMAGE ID       CREATED      SIZE
localhost/ess3000_6.0.0.2_dme  0417-01   15cc579e359d   2 days ago   4.96 GB
```

d. Go to the directory where the installation package is extracted.

```
cd /home/deploy/ess3000_6.0.0.2_0417-01_dme.dir
```

4. Update the `/etc/hosts` file on the EMS node with the IP address, long name, and short name of the EMS and the canister nodes, and then customize the `essmgr.yml` file to match your environment.

   Note: Refer to *Chapter 4* and *Chapter 5* of ESS 3000 Hardware Planning and Installation Guide for guidance on filling out `/etc/hosts` and `essmgr.yml`. Much of this information is already decided through the TDA process because it is needed when filling out the *ESS 3000 Worksheet* (*Appendix A of ESS 3000 Hardware Planning and Installation Guide*) which is required by the SSR. The SSR uses that information to set the management interface IP addresses (low-speed). The customer needs to update the host names and domain names for those corresponding IP addresses. The high-speed names are user-defined but as a best practice a suffix should be added to the management host names. All of the high-speed IP addresses must be on the same subnet.

   Note: By default, the container bridge is already given an IP address of .1 on the /24 subnet. If you want to change this, refer to the guidance in *Host IP considerations* in *Chapter 4* of *ESS 3000 Hardware Planning and Installation Guide*.

   An example of the `/etc/hosts` is as follows. Note the management network to cluster network best practice of using suffix:

```
// High speed cluster names

192.0.2.11 ess3k1a-hs.test.net ess3k1a-hs
192.0.2.12 ess3k1b-hs.test.net ess3k1b-hs
192.0.2.21 ems1-hs.test.net ems1-hs

// 1Gb management hostnames

198.51.100.11 ess3k1a.test.net ess3k1a
198.51.100.12 ess3k1b.test.net ess3k1b
198.51.100.21 ems1.test.net ems1

// container IP

198.51.100.8 cems0.test.net cems0
```

   Note: The `essmgr.yml` file is located in the directory that is created after running the binary (`.sh` file) and accepting the license.

   If you are upgrading, save the existing `essmgr.yml` file for reference when updating the new file. The only changes you need to make to the original version are:

   - IMG_NAME: ess3000_6.0.0.2_dme
   - IMG_VERSION: 0417-01

   Note: The IMG_VERSION mentioned here is an example and it might change.

   These values must match the output of the new image that is installed when running the **podman images** command. If this is a new installation, you need to modify the key values in the `essmgr.yml` such as HOSTINF, HOSTSUBNET, etc.

   Note: The syntax in this file is sensitive. For example, there must always be a blank space between key: value. Refer to *Chapter 4* of *ESS 3000 Hardware Planning and Installation Guide* for which values are needed to be changed.

   Here are some key items in the `essmgr.yml` file:

```
%YAML 1.1
---
container:
    CONTAINER_NAME: cems0                # hostname container will assume
    CONTAINER_IP: 198.51.100.8           # container IP address.
```

```
                                            # container IP address must be part of the common
                                            # subnet (HOSTSUBNET)  between container, host
and ESS FAB3

    # Container image name
    IMG_NAME: ess3000_6.0.0.2_dme
    IMG_VERSION: 0417-01

    # Host name associated with the ip address/interface for xCAT mgmt name given in
the /etc/hosts
    HOSTN: ems1
    DOMAIN_NAME: test.net
    HOSTSUBNET: 198.51.100.0/24

    # HOSTIP is assigned by default. It is the lowest address in the
    # defined subnet (it is 1 (one) added to the subnet). For example, if the subnet is
    # defined as 198.51.100.0/24, the host IP (HOSTIP) will be 198.51.100.1.
    # If you want some specific host IP, define a subnet where is it the lowest address.
    HOSTIP: 198.51.100.1

    # Must be provided. It is the interface the provisioning network will be attached to.
Because of the
    # bridging used you will not see any IP assigned to it in IP addr command.
    HOSTINF: enP4p1s0f1
    BRNAME: dmgtbr                          # Bridge Name

    # xCAT log and backup location. These are the location on the container hosting node.
    # So that xCAT logs and xCAT database is the container hosting node.
    XCAT_LOG: /home/log
    XCAT_BKUP: /home/backup
```

5. Enable the container bridge interface.

   Note: This step needs to be done only one time or upon the reboot of the EMS node. Run the **brctl show** command to see if a bridge is created against the container interface. If the bridge is not created, you might need to rerun the following command.

   ```
   ./essmgr -n
   ```

6. Start the container.

   ```
   ./essmgr -r
   ```

   This command sets the HOSTIP to the bridge interface. If the command fails, see *ESS 3000 deployment troubleshooting: Helpful podman, Ansible, and log information* in *ESS 3000: problem Determination Guide*.

   You are now inside the container (note the new host name on the terminal). If you want to exit the container without stopping it, enter Ctrl+p followed by Ctrl+q.

Chapter 6. ESS 3000 initial setup instructions

Before doing these steps, make sure that you have completed the steps in Chapter 5, “ESS 3000 common setup instructions,” on page 9.

Updating the EMS and canister nodes with new version of software, if applicable

These steps are needed to update the nodes before you create a cluster file system.

Note: If any command fails, observe the output and attempt to fix the problem, if applicable. In many cases, re-running the command is advised but you might want to contact IBM service if help is desired.

Note:

- The **ess3krun** command can be run only from within the container.
- The **ess3krun update** command only needs to be run if there is a newer version available. For more information, see “How to identify the version that is currently installed” on page 9.
- The following upgrade steps attempt to update all IBM Spectrum Scale RPMs, firmware, drivers, and settings to the latest levels for each ESS 3000 canister node. If applicable, the necessary RHEL packages are also upgraded.
- Do not run the update EMS node task if you have ESS building-blocks or protocol nodes. Updating EMS node is only needed if you have only an EMS node and ESS 3000 nodes. If you have ESS building-blocks or protocol nodes, use the legacy flow to update the EMS node to ESS 5.3.5 or later. For more information, see ESS 5.3.5x Quick Deployment Guide.
- The EMS node update only upgrades the IBM Spectrum Scale RPMs (GPFS, ESA, GUI, etc.). It does not update the OS, firmware, network drivers, etc. The update primarily keeps the EMS node at ESS 5.3.5.2 levels.

1. Configure the ESS 3000 setup.

   ```
   ess3krun  -N ess3k1a,ess3k1b config load -p  Password
   ```

   Where *Password* is the root password of each canister. It must be the same on each node.

   Important: You must specify only low-speed names with `-N` in the **ess3krun config load** command. Specifying high-speed names leads to an unstable update flow.

   This command discovers the nodes and places them into the configuration. It also attempts to fix, generate, and exchange the SSH keys. The SSH keys from the container must be shared with the container nodes and the EMS node.

   Note: After this step is performed, a group called `ess_x86_64` is created This allows you to use the `-G` option with **ess3krun** to run commands as a group instead of specifying nodes in a comma-separated list with `-N`.

2. Do the initial health check on the EMS node and the canister nodes.

   ```
   ess3krun -N ems1 healthcheck
   ess3krun -N ess3k1a,ess3k1b healthcheck
   ```

   Note: This command is designed to run all the health checks even if a failure occurs. Review the health check output and compare against the known issues to determine if you can proceed. It is desirable to have a clean health check before installation or upgrade of a system.

3. Update the EMS node.

   Note:

   - Do EMS update only if you do not have ESS or protocol nodes. If you have ESS or protocol nodes, refer to the *ESS 5.3.5x Quick Deployment Guide* and Chapter 4, “ESS 3000 and ESS for Power

considerations," on page 7. The following command updates IBM Spectrum Scale including firmware, GUI, and call home to ESS 5.3.5.2 code levels.

- If this is a new EMS node from manufacturing, it should already have ESS 5.3.5.2 installed. Use **gssinstallcheck -N localhost** to verify the ESS version. If the EMS is up-to-date, you can skip this step.

```
ess3krun -N ems1 update --offline
```

4. Update the canister nodes.

```
ess3krun -N ess3k1a,ess3k1b update --offline
```

Note:

- This step only needs to be done if you need to update the code on the canisters to a new version. If not, you can skip this step.
- Although there is no cluster created yet, using `--offline` allows the deployment procedure to update each canister in parallel to the latest version.

Setting up the cluster and the file system

Note: For the following example commands, a brand new environment is assumed. If you already have network bonds, cluster, and file system created, you might not need to run each command.

ESS 3000 cluster creation and setup is done by using Ansible playbooks that configure I/O node canisters and EMS node, and orchestrate various steps to accomplish complex tasks. **ess3krun** provides an interface to various Ansible playbooks for these tasks.

Note: The suffix is typically an extension of the management network name which is used when setting up the cluster. The host name choices, or alias', must be carefully considered when setting up the `/etc/hosts` file.

A best practice example of a suffix relationship:

```
# management network / low speed names
198.51.100.20 ess3k4a.test.net ess3k4a

# high speed names with suffix
192.0.2.20 ess3k4a-hs.test.net ess3k4a-hs
```

When passing the suffix in the following commands, refer to this relationship. You can also use an alias in `/etc/hosts`.

Note: If **ess3krun** *Nodes* **config load** is already run, you might use the `-G` option for the following commands instead of using `-N`. For example, **ess3krun -G** *ess_x86_64* **network --suffix=-hs**

This makes commands easier to run especially if there are many nodes. The following cluster and the file system setup steps are just examples in the smallest configuration (EMS and single ESS 3000). If you have multiple ESS 3000 systems, it is recommended to use `-G`.

Set up the cluster and the file system as follows.

Note: In the following two steps, the `/etc/hosts` file updates that were provided earlier are taken and the necessary high-speed network bond links for the cluster creation are created. These commands take all active InfiniBand or high-speed interface links and bond them into an interface called `bond0`. `-N` specifies comma-separated I/O node canisters and EMS node host names. The default suffix for the host name is `-hs`. You can change the default suffix by using the `--suffix` argument.

1. Create high-speed network bond for canister nodes.

```
ess3krun -N ess3k1a,ess3k1b network --suffix=-hs
```

Note: This command takes the best practice, default values for Infiniband and high-speed Ethernet. If you need to change options such as bonding mode, MTU, and so on, you might need to configure the

network interface bond after using **nmcli**. Depending on the options that are changed, you might also have to modify your switch configuration.

To see the various options available on the default bond interface, run the following command from one of the nodes (not the container):

```
nmcli con show bond-bond0
```

To modify an option, run the following command:

```
nmcli con mod bond-bond0 Options
```

For information about **nmcli**, see Configuring IP Networking with nmcli in Red Hat documentation.

2. Create high-speed network bonds for the EMS node.

```
ess3krun -N ems1 network --suffix=-hs
```

3. Create the IBM Spectrum Scale cluster.

```
ess3krun -N ess3k1a,ess3k1b cluster --suffix=-hs
```

Here *ess3k1a, ess3k1b* and *ess3k1a-hs and ess3k1b-hs* are the host names and high-speed host names of the I/O node canisters that are defined in `/etc/hosts`. The default cluster name is `test01`. The default cluster name and host name suffix can be changed by using optional arguments `--name` and `--suffix`. Use `--help` to find out more about the optional arguments.

4. Add the EMS node to the cluster.

The EMS node provides third quorum node function in a two-node cluster that is created on a single ESS 3000 system. The EMS node also hosts GUI server and the call home service agent. Create the high-speed network on the EMS node and add the EMS node to the cluster.

Note: Specify only a single canister node with the **-N** flag. For example, you can specify either ess3k1a or ess3k1b with the **-N** flag.

```
ess3krun -N ess3k1a cluster --add-ems ems1 --suffix=-hs
```

5. Set up the file system. For more information, see *Customizing file system parameters* in *ESS 3000: Problem Determination Guide*.

```
ess3krun -N ess3k1a,ess3k1b filesystem --suffix=-hs
```

The default file system name is `fs3k` and the host name suffix is `-hs`. The default file system and host name suffix can be changed by using optional arguments `--name` and `--suffix`.

Note: File system creation is done using **mmvdisk**. You can also use **mmvdisk** commands directly for these tasks. For more information, see mmvdisk documentation.

6. From the container, run manual installation check of the I/O node canisters.

```
essinstallcheck -N ess3k1a,ess3k1b
```

If there are any issues reported, you might have to rerun the update.

a. On each canister node, disable swap after doing the installation check.

```
swapoff -a
sed -i '/swap/d' /etc/fstab
```

7. Run the final health check on EMS node and canister nodes.

```
ess3krun -N ems1 healthcheck
ess3krun -N ess3k1a,ess3k1b healthcheck
```

8. Exit the container by typing `exit` and pressing `enter`.

Doing the final setup

The final setup consists of the following steps:

- Setting up of performance sensors and collectors, and starting the GUI.
- Adding protocol nodes to the cluster. Refer to IBM Spectrum Scale documentation in IBM Knowledge Center for detailed instructions for adding protocol nodes into a cluster.

1. From the EMS node (outside of the container), set up the performance monitoring collector.

   ```
   mmperfmon config generate --collectors ems1-hs
   ```

2. Define the performance monitoring sensors.

   ```
   mmchnode --perfmon -N ems1-hs,ess_x86_64
   ```

   ess_x86_64 is the node class containing the ESS 3000 canister names. If you have multiple ESS 3000 systems, specifying each node name can make this command very long, which can be avoided by specifying node classes.

3. Start the GUI.

   ```
   systemctl start gpfsgui
   ```

   a. Create the GUI admin user.

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

   b. In a web browser, enter the EMS node IP address with `https` and walk through the wizard setup instructions.

4. Set up disk call home. For more information, see Drive call home.

5. Set the time zone and set up NTP.

   Before getting started, make sure that NTP and time zone are set correctly on the EMS and canister nodes. Refer to Appendix C, “How to set up chronyd (NTP),” on page 35 to perform these tasks before proceeding.

Chapter 7. ESS 3000 upgrade instructions

ESS 3000 upgrade can be done by using one of the following methods.

- Online upgrade
- Offline upgrade

Online upgrade

Assumptions:

- Cluster is created with EMS, one or more ESS 3000 nodes, and optionally one or more ESS building blocks or protocol nodes.
- Cluster is created and file system is built.
- GPFS is active on all ESS 3000 nodes and quorum is achieved.
- Quorum or no-quorum is set for each pair of canisters. Both canister nodes must have the same quorum attribute.
- New container is installed that will update the code on the EMS and canister nodes.
- GUI and collector services are stopped on the EMS before starting the upgrade.

Before starting the online upgrade, make sure that all ESS 3000 nodes are active by running the following command from one of the cluster nodes:

```
mmgetstate -N ess_x86_64
```

Use the following online upgrade steps.

1. Complete the steps in Chapter 5, “ESS 3000 common setup instructions,” on page 9. These steps include obtaining the new ESS 3000 code, backing up the original container, and installing and running the new one. After doing these steps, you should be in the new container (ESS 3000 version 6.0.0.2).
2. Run the configuration load.

   ```
   ess3krun -N ess3k1a,ess3k1b config load
   ```

   You can add -p *Password* if you want to fix the SSH keys. Most users will use this option. If SSH keys fail, you are prompted to re-run the command with this flag.
3. If applicable, update the EMS node.

   ```
   ess3krun -N ems1 update --offline
   ```

   Note:

   - Run the EMS update only if there are no ESS or protocol nodes in the environment. If there are ESS or protocol nodes in your setup, refer to *ESS 5.3.5x Quick Deployment Guide*.
   - EMS updates must be performed offline with GPFS down. Make sure that you start GPFS before moving to the next step so it is an active participant in the cluster, potentially for quorum.
4. Adjust the page pool to 60% for each node class as follows. For more information, see *Configuring the GPGS page pool size to the 60% target* in *ESS 3000: Problem Determination Guide*.
5. Update the canister nodes by using one the following commands. The following example steps are for a rolling upgrade.

   Example: With 3 ESS 3000 nodes and 1 EMS node, there are 7 total nodes in the cluster. All nodes must be active to start the upgrade with quorum achieved. If all nodes are quorum nodes (user defined), quorum in this situation means 4 out of 7 nodes must be active at a given time for the cluster to stay up. For a given ESS 3000 system to maintain file system access, at least one of the two canister nodes must have active ownership of the recovery group. When doing the following example steps, all

of canister A's (3 nodes) are updated and brought back as active. Then, all of the Canister B nodes (3 nodes) are updated. This is achieved while the cluster and file system remain available to users.

- Update by using the group of all configured ESS 3000 nodes.

```
ess3krun -G ess_x86_64 update
```

- Update by using the individual nodes.

```
ess3krun -N ess3k1a,ess3k1b,ess3k2a,ess3k2b,ess3k3a,ess3k3b update
```

6. Log in to one of the canister nodes and update the drive firmware.

```
ssh ess3k1a
mmchfirmware --type drive -N ess3k1a-hs,ess3k1b-hs
```

7. Run installation check on each canister node and adjust swap.

```
essinstallcheck -N localhost
swapoff -a
sed -i '/swap/d' /etc/fstab
```

Note: SSH to each node individually to run these commands. When you are done, exit back to the container.

8. Run health checks on the EMS node and canister nodes.

```
ess3krun -N ems1 healthcheck
ess3krun -N ess3k1a,ess3k1b healthcheck
```

Note: Adjust the command for the number of ESS 3000 nodes you have or use the group of all configured ESS 3000 nodes.

9. Exit the container and then restart GUI and collector services on the EMS node.

```
systemctl start pmcollector
systemctl start gpfsgui
```

Offline upgrade

Assumptions:

- Canister updates are done in parallel for all specified nodes.
- If GFPS is up on a given node, you are asked if it is OK to shut down GPFS.
- You assume the risks of potential quorum loss.
- The GPFS GUI and collector must be down.

Use the following offline upgrade steps.

1. Complete the steps in Chapter 5, “ESS 3000 common setup instructions,” on page 9. These steps include obtaining the new ESS 3000 code, backing up the original container, and installing and running the new one. After doing these steps, you should be in the new container (ESS 3000 version 6.0.0.2).
2. Run the configuration load.

```
ess3krun -N ess3k1a,ess3k1b config load
```

You can add -p *Password* if you want to fix the SSH keys. Most users will use this option. If SSH keys fail, you are prompted to re-run the command with this flag.

3. If applicable, update the EMS node.

```
ess3krun -N ems1 update --offline
```

Note:

- Run the EMS update only if there are no ESS or protocol nodes in the environment. If there are ESS or protocol nodes in your setup, refer to *ESS 5.3.5x Quick Deployment Guide*.
- EMS updates must be performed offline with GPFS down.

4. Update the canister nodes by using one the following commands. The following example steps are for an offline upgrade.

   Example: With 3 ESS 3000 nodes and 1 EMS node, there are 7 total nodes in the cluster. GPFS must be shut down on the nodes that you want to update to begin or you are prompted to shut down GPFS on these nodes.

   - Update by using the group of all configured ESS 3000 nodes.

     ```
     ess3krun -G ess_x86_64 update --offline
     ```

   - Update by using the individual nodes.

     ```
     ess3krun -N ess3k1a,ess3k1b,ess3k2a,ess3k2b,ess3k3a,ess3k3b update --offline
     ```

     Note: If you need to update an individual node, use `-N`.

5. Log in to one of the canister nodes and update the drive firmware.

   ```
   ssh ess3k1a
   mmchfirmware --type drive -N ess3k1a-hs,ess3k1b-hs
   ```

6. Run installation check on each canister node and adjust swap.

   ```
   essinstallcheck -N localhost
   swapoff -a
   sed -i '/swap/d' /etc/fstab
   ```

   Note: SSH to each node individually to run these commands. When you are done, exit back to the container.

After the update is complete, you can start the cluster, or individual nodes, back up. This includes any services such as GUI or call home.

Appendix A. ESS 3000 known issues

Known issues in ESS 3000 version 6.0.0.2

The following table describes the known issues in IBM Elastic Storage System 3000 version 6.0.0.2 and how to resolve these issues.

<table>
<tr><th>Issue</th><th>Resolution or action</th></tr>
<tr><td>JAVA_HOME might be pointing to the wrong version which might cause ESA start to fail:

In the following example, note how java is pointing to the wrong location. This causes the ESA start to fail:

<pre># ls -alt
total 20
drwxr-xr-x.   2 root
root 4096 Nov 22 15:02 .
lrwxrwxrwx   1 root
root   62 Nov 22 15:02
java -> /usr/lib/jvm/
java-11-
openjdk-11.0.ea.28-7.
el7.ppc64le/bin/java
lrwxrwxrwx   1 root
root   70 Nov 22 15:02
java.1.gz -> /usr/
share/man/ man1/java-
java-11-openjdk-11.0.ea.
28-7.el7.ppc64le.1.gz
lrwxrwxrwx   1 root
root   61 Nov 22 15:02
jjs -> /usr/lib/jvm /
java-11-openjdk-11.0.ea.
28-7.el7.ppc64le/bin/jjs</pre></td><td>To fix the problem, remove the current <code>java</code> symbolic link, update the <code>java</code> pointer, and retry the ESA activation.

1. Remove the current java symbolic link.

<pre># cd /etc/alternatives/
rm java
rm: remove symbolic link ‘java’? y</pre>
2. Update the <code>java</code> pointer.

<pre># ln -s /usr/lpp/mmfs/java java
ls -alt | grep -i java
lrwxrwxrwx   1 root root   18 Nov 22 16:03 java -> /usr/lpp/mmfs/
java</pre>
<pre>cd /opt/ibm/
ln -s /etc/alternatives/java java-ppc64le-80
ls -alt
total 0
drwxr-xr-x. 5 root   root   62 Nov 22 16:04 .
lrwxrwxrwx   1 root   root   22 Nov 22 16:04 java-ppc64le-80
-> /etc/alternatives/java
dr-xr-x--- 12 root   root   151 Nov 22 15:48 esa
drwxr-xr-x. 10 root   root   119 Nov 7 16:09 ..
drwx------   8 scalemgmt scalemgmt 121 Nov 7 16:00 wlp
drwxr-xr-x. 7 root   root   68 Nov 7 14:36 gss

vi /opt/ibm/esa/runtime/conf/javaHome.sh

cat /opt/ibm/esa/runtime/conf/javaHome.sh
JAVA_HOME=/opt/ibm/java-ppc64le-80/jre</pre>
3. Retry the ESA activation.

<pre># /opt/ibm/esa/bin/activator -C -p 5024 -w -Y</pre></td></tr>
<tr><td>The hardware CPU validation GPFS callback is only active for one node in the cluster. This callback prevents GPFS from starting if a CPU socket is missing.</td><td>No action is required.</td></tr>
<tr><td>During rolling upgrade, mmhealth might show the error <code>local_exported_fs_unavail</code> even though the file system is still mounted.</td><td>During a rolling upgrade (Updating of one ESS 3000 canister node at a time but maintaining quorum), mmhealth might display an error indicating that the local exported file system is unavailable. This message is erroneous.

<pre>Component   Status   Status Change Reasons

GPFS   HEALTHY   6 min. ago   -
NETWORK   HEALTHY   20 min. ago   -
FILESYSTEM   DEGRADED 18 min. ago   local_exported_fs_unavail(gpfs1)</pre></td></tr>
</table>

Issue	Resolution or action
	`` DISK HEALTHY 6 min. ago - NATIVE_RAID HEALTHY 6 min. ago - PERFMON HEALTHY 19 min. ago - THRESHOLD HEALTHY 20 min. ago - ``` The workaround is to restart **mmsysmon** on each node called out by **mmhealth**.
During upgrade, if the container had an unintended loss of connection with the target canister(s), there might be a timeout of up to 2 hours in the Ansible update task.	Wait for the timeout and retry the **ess3krun** update task.
During storage MES upgrade, you are required to update the drive firmware to complete the task. Some of the drives might not update on the first pass of running the command.	Re-run the **mmchfirmware –type drive** command which should resolve the issue and update the remaining drives.
When running **ess3krun** commands, you might see messages such as these: ``` Thursday 16 April 2020 20:52:44 +0000 (0:00:00.572) 0:13:19.792 ******** Thursday 16 April 2020 20:52:45 +0000 (0:00:00.575) 0:13:20.367 ******** Thursday 16 April 2020 20:52:46 +0000 (0:00:00.577) 0:13:20.944 ******** Thursday 16 April 2020 20:52:46 +0000 (0:00:00.576) 0:13:21.521 ******** Thursday 16 April 2020 20:52:47 +0000 (0:00:00.570) 0:13:22.091 ******** Thursday 16 April 2020 20:52:47 +0000 (0:00:00.571) 0:13:22.663 ******** ```	This is a restriction in the Ansible timestamp module. It shows timestamps even for the “skipped” tasks. If you want to remove timestamps from the output, change the `ansible.cfg` file inside the container as follows: 1. vim `/etc/ansible/ansible.cfg` 2. Rmove `,profile_tasks` on line 7. 3. Save and quit: `esc + :wq`
When running the **ess3krun config load** command, you might see a failure such as this: ``` stderr: |- rc=2 code=186 Failed to obtain the enclosure device name with rc=2 rc=2 code=669 ```	This failure means that the pems module is not running the canister. For fixing this, do the following: 1. Log in to the failed canister and run the following commands: ``` cd /install/ess/otherpkgs/rhels8/x86_64/gpfs yum reinstall gpfs.ess.platform.ess3k* ``` 2. When the installation finishes, wait until the **lsmod | grep pems** command returns output similar to this: ``` pemsmod 188416 0 scsi_transport_sas 45056 1 pemsmod ``` 3. Retry the **ess3krun config load** command from the container.

Issue	Resolution or action
Running **ess3krun -N node1,node2,… config load** command with high-speed names causes issues with the upgrade task using the `-G` flag.	The **ess3krun config load** command is an Ansible wrapper that attempts to discover the ESS 3000 canister node positions, place them into groups, and fix the SSH keys between the servers. This command must always be run using the low-speed or management names. You must not use the high-speed names with this command. For example: **ess3krun -N ess3k1a,ess3k1b config load** If you run this command using the high-speed or cluster names, this might result in issues when performing the update task. Example of what not to do: **ess3krun -N ess3k1a-hs,ess3k1b-hs config load** To confirm that the config run is set up correctly, use the **lsdef** command. This command returns only the low-speed or management names defined in `/etc/hosts`.

Appendix B. Security-related settings in ESS 3000

The following topics describe how to enable security related settings in ESS 3000.

Enabling firewall in ESS

Enabling firewall in an ESS environment is a one-step process and it can be enabled for EMS, I/O server nodes, and protocol nodes by using the **firewall** sub-command of the **ess3krun** command.

By default, any node in an ESS cluster has firewall disabled. You can run the **firewall** sub-command of the **ess3krun** command. This command can be run after the deployment of EMS node or I/O server nodes is complete.

- Enable firewall on the EMS node by running the **firewall** sub-command with the `enable` option.

  ```
  # ess3krun -N ems1 firewall enable
  ```

  You can check the status of the firewall as follows.

  ```
  # firewall-cmd --state
  running
  ```

  You can verify the open firewall ports by running **firewall** sub-command with the `verify` option. When the command completes, the required ports in firewall are verified.

  ```
  # ess3krun -N ems1 firewall verify
  ```

- Enable firewall on I/O server nodes by running the **firewall** sub-command with the `enable` option.

  ```
  # ess3krun -N ess_x86_64 firewall enable
  ```

  You can check the status of the firewall as follows.

  ```
  # firewall-cmd --state
  running
  ```

  You can verify the open firewall ports by running the **firewall** sub-command with the `verify` option. When the command completes, the required ports in firewall are verified.

  ```
  # ess3krun -N ess_x86_64 firewall verify
  ```

- Disable firewall on the EMS node by running the **firewall** sub-command with the `disable` option.

  ```
  # ess3krun -N ems1 firewall disable
  ```

- Disable firewall on I/O server nodes by running the **firewall** sub-command with the `disable` option.

  ```
  # ess3krun -N ess_x86_64 firewall disable
  ```

  Protocol node consideration: Protocol node deployment is not supported with ESS 3000 6.0.0.2 container.

Enabling SELinux in ESS

Enabling SELinux in an ESS environment is a two-step process and it can be enabled for EMS and I/O server nodes using the **selinux** sub-command of the **ess3krun** command.

By default, any node in an ESS cluster has SELinux disabled. You can run the **selinux** sub-command of the **ess3krun** command to enable or disable SELinux on nodes. This command can be run after the deployment of EMS node or I/O server nodes is complete.

- Enable SELinux on the EMS node as follows.
  a) Run the **selinux** sub-command on the EMS node.

```
# ess3krun -N ems1 selinux permissive
```

  Note: Make sure that you reboot the node when the **selinux** sub-command completes.

  b) Reboot the node.

```
# systemctl reboot
```

  The node is rebooted and it comes up with SELinux in Permissive mode.

```
# sestatus
SELinux status:                 enabled
SELinuxfs mount:                /sys/fs/selinux
SELinux root directory:         /etc/selinux
Loaded policy name:             targeted
Current mode:                   permissive
Mode from config file:          permissive
Policy MLS status:              enabled
Policy deny_unknown status:     allowed
Max kernel policy version:      31
```

  c) Rerun the **selinux** sub-command with the `enable` option to enforce SELinux.

```
# ess3krun -N ems1 selinux enable
```

  No reboot is required in this case.

```
# sestatus
SELinux status:                 enabled
SELinuxfs mount:                /sys/fs/selinux
SELinux root directory:         /etc/selinux
Loaded policy name:             targeted
Current mode:                   enforcing
Mode from config file:          enforcing
Policy MLS status:              enabled
Policy deny_unknown status:     allowed
Max kernel policy version:      31
```

  After SELinux is enabled, kernel logs any activity in the `/var/log/audit/audit.log` file.

- Enable SELinux on I/O server nodes as follows.
  a) Run the **selinux** sub-command on the I/O server nodes.

```
# ess3krun -G ess_x86_64 selinux permissive
```

  Note: Make sure that you reboot the node when the **selinux** sub-command completes.

  b) Reboot the I/O server nodes.

```
# systemctl reboot
```

  The node is rebooted and it comes up with SELinux in Permissive mode.

  c) Rerun the **selinux** sub-command with the `enable` option to enforce SELinux.

```
# ess3krun -G ess_x86_64 selinux enable
```

No reboot is required in this case.

After SELinux is enabled, kernel logs any activity in the `/var/log/audit/audit.log` file.

- Disable SELinux on ESS nodes as follows.
  - To disable SELinux on the EMS node, use the following command.

    ```
    # ess3krun -N ems1 selinux disable
    ```

    Reboot the node after the command completes. When the node comes up after reboots, SELinux is disabled.

    You can check the status as follows.

    ```
    # sestatus
    SELinux status:                 disabled
    ```

  - To disable SELinux on the I/O server nodes, use the following command.

    ```
    # ess3krun -G ess_x86_64 selinux disable
    ```

    Reboot the node after the command completes. When the node comes up after reboots, SELinux is disabled. Any I/O server node name can also be used instead of the group name.

  Protocol node consideration: Protocol node deployment is not supported with ESS 3000 version 6.0.0.2 container.

  Additional information: Any mentioned security item is an optional feature and you can enable it on demand for an ESS cluster. Security commands can be run using the **ess3krun** command after deployment of the node is done and before creating the GPFS cluster. In upgrade cases, any such security commands must be run after stopping the GPFS cluster. Do not attempt to run any security command while GPFS cluster is up and running.

  Container consideration: Make sure that none of the security command is run against the container node. The container has a very light footprint of Red Hat Enterprise Linux 7.x OS on which any security parameters are not supported.

Working with sudo user in an ESS Environment

Enabling sudo requires a sudo-capable user (`gpfsadmin`) to be added to all nodes which are a part of or which are going to be a part of an ESS cluster. Sudo must be enabled for EMS and I/O server nodes by using the **sudo** sub-command of the **ess3krun** command.

Note: Sudo user across all GPFS nodes must have the same Linux group ID and user ID.

Enabling sudo on Linux nodes

You can enable sudo configuration on a Linux node by using the `enable` option of the **sudo** sub-command.

```
# ess3krun -N ems1 sudo enable
```

This command creates the gpfsadmin Linux user and gpfs Linux group on the node and performs all necessary sudoers set up. For detailed information, see the /etc/sudoers.d/ess_sudoers file.

User can now log in to the node server using the gpfsadmin user and they can perform GPFS administration tasks.

Make sure that the **sudo** sub-command is run on all GPFS nodes (EMS node, I/O server nodes, and any client nodes) as part of the cluster to be completely compliant with the sudo requirement. Change the node name in the **sudo** sub-command accordingly. Enabling sudo also allows the gpfsadmin user to administer xCAT and the GPFS GUI on the EMS node.

Disabling sudo on Linux nodes

You can disable sudo configuration on a Linux node by using enable option of the **sudo** sub-command.

```
# ess3krun -N ems1 sudo disable
```

Disabling sudo reverts the xCAT policy table to its previous state, deletes /etc/sudoers.d/ess_sudoers file, and deletes the gpfsadmin user from the Linux node. Make sure that you have disabled sudo user configuration on all GPFS nodes (EMS node, I/O server nodes, and any client nodes) as part of the cluster to be completely compliant with the sudo requirement. Change the node name in the **sudo** sub-command accordingly.

Important: You must not disable sudo user until the GPFS cluster is set to configure not to use sudo wrapper and sudo user. Failing to do so might result in cluster corruption.

Enabling sudo with GPFS cluster

Once the sudo feature is enabled, make sure that you use --use-sudo-wrapper and --sudo-user options while creating a new GPFS cluster by using **essgencluster**. For more information, see **essgencluster** command. If there is an existing cluster available, it must be converted to use sudo wrapper and sudo user by using the **sudo** sub-command. For more information, see sudo sub-command help text.

For example, consider a cluster which is created earlier and it is not using sudo wrapper and sudo user.

```
# mmlscluster

GPFS cluster information
========================
  GPFS cluster name:         scalecluster.gpfs.net
  GPFS cluster id:           15270568330550226974
  GPFS UID domain:           scalecluster.gpfs.net
  Remote shell command:      /usr/bin/ssh (No SUDO Wrapper used here)
  Remote file copy command:  /usr/bin/scp (No SUDO Wrapper used here)
  Repository type:           CCR

 Node  Daemon node name   IP address     Admin node name      Designation
-----------------------------------------------------------------------------
   1   io3-10g.gpfs.net   198.51.100.14  io3-10g.gpfs.net     quorum-manager
   2   io4-10g.gpfs.net   198.51.100.15  io4-10g.gpfs.net     quorum-manager
   3   ems1-10g.gpfs.net  198.51.100.13  ems2-10g.gpfs.net    quorum
```

You can configure the cluster to use sudo by issuing the following command.

```
# ess3krun -N ems1 sudo use_sudo_wrapper
```

```
# mmlscluster

GPFS cluster information
========================
  GPFS cluster name:         scalecluster.gpfs.net
  GPFS cluster id:           15270568330550226974
  GPFS UID domain:           scalecluster.gpfs.net
  Remote shell command:      sudo wrapper in use (SUDO wrapper now in use)
  Remote file copy command:  sudo wrapper in use (SUDO wrapper now in use)
  Repository type:           CCR

 Node  Daemon node name   IP address     Admin node name    Designation
```

```
--------------------------------------------------------------------------
   1   io3-10g.gpfs.net   198.51.100.14  io3-10g.gpfs.net   quorum-manager
   2   io4-10g.gpfs.net   198.51.100.15  io4-10g.gpfs.net   quorum-manager
   3   ems2-10g.gpfs.net  198.51.100.13  ems2-10g.gpfs.net  quorum
```

In the preceding **mmlscluster** command output, remote shell and remote copy commands are changed to use sudo wrapper (**sshwrap** and **scpwrap**).

The sudoUser **mmlsconfig** parameter is now set to gpfsadmin.

```
# mmlsconfig sudoUser
sudoUser gpfsadmin
```

Important:

- The **sudo** sub-command must not be used for nodes other than the EMS node.
- The IBM Spectrum Scale GUI services must be restarted by using **systemctl restart gpfsgui** after enabling or disabling sudo in a GPFS cluster.
- The sudo user password must be set to a new password before using it.

Disabling sudo with GPFS cluster

You can unconfigure a sudo-enabled GPFS cluster to not use sudo wrapper by using the sudo no_sudo_wrapper switch of the **sudo** sub-command.

For example, consider a cluster which is created earlier and it is using sudo wrapper and sudo user.

```
# mmlscluster

GPFS cluster information
========================
  GPFS cluster name:         scalecluster.gpfs.net
  GPFS cluster id:           15270568330550226974
  GPFS UID domain:           scalecluster.gpfs.net
  Remote shell command:      sudo wrapper in use (SUDO wrapper now in use)
  Remote file copy command:  sudo wrapper in use (SUDO wrapper now in use)
  Repository type:           CCR

 Node  Daemon node name    IP address     Admin node name   Designation
--------------------------------------------------------------------------
   1   io3-10g.gpfs.net   198.51.100.14  io3-10g.gpfs.net   quorum-manager
   2   io4-10g.gpfs.net   198.51.100.15  io4-10g.gpfs.net   quorum-manager
   3   ems2-10g.gpfs.net  198.51.100.13  ems2-10g.gpfs.net  quorum
```

You can configure the cluster to not to use sudo by issuing the following command.

```
# ess3krun -N ems1 sudo no_sudo_wrapper
```

```
# mmlscluster

GPFS cluster information
========================
  GPFS cluster name:         scalecluster.gpfs.net
  GPFS cluster id:           15270568330550226974
  GPFS UID domain:           scalecluster.gpfs.net
  Remote shell command:      /usr/bin/ssh (No SUDO Wrapper used here)
  Remote file copy command:  /usr/bin/scp (No SUDO Wrapper used here)
  Repository type:           CCR

 Node  Daemon node name      IP address      Admin node name   Designation
-----------------------------------------------------------------------------
   1   io3-10g.gpfs.net     198.51.100.14   io3-10g.gpfs.net   quorum-manager
   2   io4-10g.gpfs.net     198.51.100.15   io4-10g.gpfs.net   quorum-manager
   3   ems1-10g.gpfs.net    198.51.100.13   ems2-10g.gpfs.net  quorum
```

In the preceding **mmlscluster** command output, remote shell and remote copy commands are changed to use ssh and scp instead of sudo wrapper (**sshwrap** and **scpwrap**).

The sudoUser **mmlsconfig** parameter is now set to undefined.

```
# mmlsconfig sudoUser
sudoUser (undefined)
```

Important:

- The **sudo** sub-command must not be used for nodes other than the EMS node.
- The IBM Spectrum Scale GUI services must be restarted by using **systemctl restart gpfsgui** after enabling or disabling sudo in a GPFS cluster.

I/O server nodes

I/O server nodes must also have sudo user gpfsadmin configured if the ESS cluster is going to be managed with a sudo user.

```
# ess3krun -G ess_x86_64 sudo enable
```

Important: The **sudo** sub-command must not be used for nodes other than the EMS node.

Protocol nodes

Protocol node deployment is not supported with ESS 3000 6.0.0.2 container.

Help text sudo sub-command

```
# /opt/ibm/ess/tools/bin/ess3krun sudo --help
usage: ess3krun sudo [-h] [--user SUDO_USER] [--group SUDO_GROUP]
                     {enable,disable,use_sudo_wrapper,no_sudo_wrapper}

positional arguments:
  {enable,disable,use_sudo_wrapper,no_sudo_wrapper}

optional arguments:
  -h, --help            show this help message and exit
  --user SUDO_USER      Provide sudo user name
  --group SUDO_GROUP    Provide group name
```

Using the central administration mode in an ESS 3000 environment

Enabling the central administration mode, by setting adminMode attribute to central, prevents unwanted passwordless SSH access from any non-admin GPFS nodes to any another GPFS node. In case of ESS 3000, it is assumed that the EMS node is the only node which acts as an admin mode.
For more information, see *adminMode configuration attribute* in *IBM Spectrum Scale: Administration Guide*.

Running the **admincentral** sub-command along with **ess3krun** configures adminMode=central in an ESS 3000 cluster. By default, passwordless SSH setup between all nodes is enabled.

Only the EMS node is allowed to do passwordless SSH to all other GPFS nodes. However, other nodes such as the I/O server nodes, protocol nodes, and client nodes cannot do SSH back to the EMS or other GPFS nodes once adminMode is set to central and the node security context is updated.

- “Enabling the central administration mode” on page 32
- “Disabling the central administration mode” on page 33
- “Help text admincentral sub-command” on page 34

Enabling the central administration mode

Enabling the central administration mode is a two-step procedure.

1. Run the **admincentral** sub-command with the enable option against the container node.

Important: You must enable `adminMode=central` by using container node as xCAT services run inside the container node not on the physical EMS node. However, once `adminMode=central` is enabled, the physical EMS node can act as an admin node for ESS 3000 nodes as physical EMS and container EMS share the same SSH public and private keys.

```
# ess3krun -N cems0 admincentral enable
```

Note: After running this command any future deployment of new nodes only have the `adminMode` attribute set to `central`, by default. For existing nodes in the cluster, you must update the xCAT security context by running the following command.

2. Update the xCAT security context using the **`updatenode Node -k`** script.

```
# updatenode gss_ppc64,ces_ppc64 -V -k
...
Password: <Type EMS node root Password here>
...
...
```

Note:

- If you do not run the **`updatenode Node -k`** command, the central administration mode gets enabled for any new nodes deployed using the current EMS node. However, existing nodes can still do passwordless SSH between each other.
- In case of an upgrade, if you want to enable the central administration mode then run the same commands.
- Make sure that you do not run **`updatenode admin_node -V -k`** on the EMS node which is the admin node.
- Running the **`admincentral`** sub-command against non-container nodes is not allowed. In other words, with the `-N` option the container node name must be specified as an argument.

The **`admincentral`** sub-command can be run after the deployment of the EMS node, I/O server nodes, or protocol nodes is completed.

Disabling the central administration mode

Disabling the central administration mode is a two-step procedure.

1. Run the **`admincentral`** sub-command with the `disable` option.

```
# ess3krun -N cems0 admincentral enable
```

Note: After running this command any future deployment of new nodes only have the central administration mode disabled. For existing nodes in the cluster, you must update the xCAT security context by running the following command.

2. Update the xCAT security context using the **`updatenode Node -k`** script.

```
# updatenode gss_ppc64,ces_ppc64 -V -k
...
Password: <Type EMS node root Password here>
...
...
```

Note:

- If you do not run the **`updatenode Node -k`** command, the central administration mode gets disabled for any new nodes deployed using the current EMS node. However, existing nodes cannot do passwordless SSH between each other.
- In case of an upgrade, if you want to disable the central administration mode then run the same commands.
- Make sure that you do not run **`updatenode admin_node -V -k`** on the EMS node which is the admin node.

- Running **admincentral** sub-command against non-container nodes is not allowed. In other words, with the -N option the container node name must be specified as an argument.

Help text admincentral sub-command

```
# ess3krun -N cems0 admincentral -h
usage: ess3krun admincentral [-h] {enable,disable}

positional arguments:
  {enable, disable}

optional arguments:
  -h, --help        show this help message and exit
```

Appendix C. How to set up chronyd (NTP)

For the following example steps, it is assumed that the EMS node is the `chronyd` server and there is no public internet synchronization.

- Do the following steps on the EMS node, outside of the container.
  a) Set the time zone and the date locally.
  b) Edit the contents of the `/etc/chrony.conf` file as follows.

```
# Use public servers from the pool.ntp.org project.
# Please consider joining the pool (http://www.pool.ntp.org/join.html).
#server 0.rhel.pool.ntp.org iburst
#server 1.rhel.pool.ntp.org iburst
#server 2.rhel.pool.ntp.org iburst
#server 3.rhel.pool.ntp.org iburst
server 192.168.7.1 prefer iburst

# Record the rate at which the system clock gains/losses time.
driftfile /var/lib/chrony/drift
local stratum 8
manual

# Allow the system clock to be stepped in the first three updates
# if its offset is larger than 1 second.
makestep 1.0 3

# Enable kernel synchronization of the real-time clock (RTC).
rtcsync

# Enable hardware timestamping on all interfaces that support it.
#hwtimestamp *

# Increase the minimum number of selectable sources required to adjust
# the system clock.
#minsources 2

# Allow NTP client access from local network.
#allow 192.168.0.0/16
allow 192.168.7.0/24

# Serve time even if not synchronized to a time source.
#local stratum 10

# Specify file containing keys for NTP authentication.
#keyfile /etc/chrony.keys

# Specify directory for log files.
logdir /var/log/chrony

# Select which information is logged.
#log measurements statistics tracking
```

  c) Save the changes in `/etc/chrony.conf` file.
  d) Restart `chronyd`.

```
systemctl restart chronyd

chronyc makestep
chronyc ntpdata
timedatectl
```

- Do the following steps on the client nodes (canister nodes or ESS nodes).
  a) Edit the contents of the `/etc/chrony.conf` file as follows.

```
# Use public servers from the pool.ntp.org project.
# Please consider joining the pool (http://www.pool.ntp.org/join.html).
#server 0.rhel.pool.ntp.org iburst
#server 1.rhel.pool.ntp.org iburst
#server 2.rhel.pool.ntp.org iburst
#server 3.rhel.pool.ntp.org iburst
```

```
server 192.168.7.1 prefer iburst

# Record the rate at which the system clock gains/losses time.
server master iburst
driftfile /var/lib/chrony/drift
logdir /var/log/chrony
log measurements statistics tracking

# Allow the system clock to be stepped in the first three updates
# if its offset is larger than 1 second.
makestep 1.0 3

# Enable kernel synchronization of the real-time clock (RTC).
rtcsync

# Enable hardware timestamping on all interfaces that support it.
#hwtimestamp *

# Increase the minimum number of selectable sources required to adjust
# the system clock.
#minsources 2

# Allow NTP client access from local network.
#allow 192.168.0.0/16
#allow 192.168.7.0/24

# Serve time even if not synchronized to a time source.
#local stratum 10

# Specify file containing keys for NTP authentication.
#keyfile /etc/chrony.keys

# Specify directory for log files.
logdir /var/log/chrony

# Select which information is logged.
#log measurements statistics tracking
```

b) Save the changes in the `/etc/chrony.conf` file.

c) Restart `chronyd`.

```
systemctl restart chronyd

chronyc makestep
chronyc ntpdata
timedatectl
```

Accessibility features for IBM Spectrum Scale RAID

Accessibility features help users who have a disability, such as restricted mobility or limited vision, to use information technology products successfully.

Accessibility features

The following list includes the major accessibility features in IBM Spectrum Scale RAID:

- Keyboard-only operation
- Interfaces that are commonly used by screen readers
- Keys that are discernible by touch but do not activate just by touching them
- Industry-standard devices for ports and connectors
- The attachment of alternative input and output devices

IBM Knowledge Center, and its related publications, are accessibility-enabled. The accessibility features are described in IBM Knowledge Center (www.ibm.com/support/knowledgecenter).

Keyboard navigation

This product uses standard Microsoft Windows navigation keys.

IBM and accessibility

See the IBM Human Ability and Accessibility Center (www.ibm.com/able) for more information about the commitment that IBM has to accessibility.

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing IBM Corporation North Castle Drive Armonk, NY 10504-1785 U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing Legal and Intellectual Property Law IBM Japan Ltd. 19-21,

Nihonbashi-Hakozakicho, Chuo-ku Tokyo 103-8510, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
Dept. 30ZA/Building 707
Mail Station P300
2455 South Road,
Poughkeepsie, NY 12601-5400
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment or a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml.

Intel is a trademark of Intel Corporation or its subsidiaries in the United States and other countries.

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

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, and Windows NT are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Glossary

This glossary provides terms and definitions for the ESS 3000 solution.

The following cross-references are used in this glossary:

- *See* refers you from a non-preferred term to the preferred term or from an abbreviation to the spelled-out form.
- *See also* refers you to a related or contrasting term.

For other terms and definitions, see the IBM Terminology website (opens in new window):

http://www.ibm.com/software/globalization/terminology

B

building block
A pair of servers with shared disk enclosures attached.

BOOTP
See *Bootstrap Protocol (BOOTP)*.

Bootstrap Protocol (BOOTP)
A computer networking protocol that is used in IP networks to automatically assign an IP address to network devices from a configuration server.

C

CEC
See *central processor complex (CPC)*.

central electronic complex (CEC)
See *central processor complex (CPC)*.

central processor complex (CPC)
A physical collection of hardware that consists of channels, timers, main storage, and one or more central processors.

cluster
A loosely-coupled collection of independent systems, or *nodes*, organized into a network for the purpose of sharing resources and communicating with each other. See also *GPFS cluster*.

cluster manager
The node that monitors node status using disk leases, detects failures, drives recovery, and selects file system managers. The cluster manager is the node with the lowest node number among the quorum nodes that are operating at a particular time.

compute node
A node with a mounted GPFS file system that is used specifically to run a customer job. ESS 3000 disks are not directly visible from and are not managed by this type of node.

CPC
See *central processor complex (CPC)*.

D

DA
See *declustered array (DA)*.

datagram
A basic transfer unit associated with a packet-switched network.

DCM
See *drawer control module (DCM)*.

declustered array (DA)
A disjoint subset of the pdisks in a recovery group.

dependent fileset
A fileset that shares the inode space of an existing independent fileset.

DFM
See *direct FSP management (DFM)*.

DHCP
See *Dynamic Host Configuration Protocol (DHCP)*.

direct FSP management (DFM)
The ability of the xCAT software to communicate directly with the Power Systems server's service processor without the use of the HMC for management.

drawer control module (DCM)
Essentially, a SAS expander on a storage enclosure drawer.

Dynamic Host Configuration Protocol (DHCP)
A standardized network protocol that is used on IP networks to dynamically distribute such network configuration parameters as IP addresses for interfaces and services.

E

Elastic Storage System (ESS 3000)
A high-performance, GPFS NSD solution made up of one or more building blocks that runs on IBM Power Systems servers. The ESS 3000 software runs on ESS 3000 nodes - management server nodes and I/O server nodes.

ESS 3000 Management Server (EMS)
An xCAT server is required to discover the I/O server nodes (working with the HMC), provision the operating system (OS) on the I/O server nodes, and deploy the ESS software on the management node and I/O server nodes. One management server is required for each ESS 3000 system composed of one or more building blocks.

encryption key
A mathematical value that allows components to verify that they are in communication with the expected server. Encryption keys are based on a public or private key pair that is created during the installation process. See also *file encryption key (FEK)*, *master encryption key (MEK)*.

ESS 3000
See *Elastic Storage System (ESS 3000)*.

environmental service module (ESM)
Essentially, a SAS expander that attaches to the storage enclosure drives. In the case of multiple drawers in a storage enclosure, the ESM attaches to drawer control modules.

ESM
See *environmental service module (ESM)*.

Extreme Cluster/Cloud Administration Toolkit (xCAT)
Scalable, open-source cluster management software. The management infrastructure of ESS is deployed by xCAT.

F

failback
Cluster recovery from failover following repair. See also *failover*.

failover
(1) The assumption of file system duties by another node when a node fails. (2) The process of transferring all control of the ESS to a single cluster in the ESS when the other clusters in the ESS fails. See also *cluster*. (3) The routing of all transactions to a second controller when the first controller fails. See also *cluster*.

failure group
A collection of disks that share common access paths or adapter connection, and could all become unavailable through a single hardware failure.

FEK
See *file encryption key (FEK)*.

file encryption key (FEK)
A key used to encrypt sectors of an individual file. See also *encryption key*.

file system
The methods and data structures used to control how data is stored and retrieved.

file system descriptor
A data structure containing key information about a file system. This information includes the disks assigned to the file system (*stripe group*), the current state of the file system, and pointers to key files such as quota files and log files.

file system descriptor quorum
The number of disks needed in order to write the file system descriptor correctly.

file system manager
The provider of services for all the nodes using a single file system. A file system manager processes changes to the state or description of the file system, controls the regions of disks that are allocated to each node, and controls token management and quota management.

fileset
A hierarchical grouping of files managed as a unit for balancing workload across a cluster. See also *dependent fileset*, *independent fileset*.

fileset snapshot
A snapshot of an independent fileset plus all dependent filesets.

flexible service processor (FSP)
Firmware that provices diagnosis, initialization, configuration, runtime error detection, and correction. Connects to the HMC.

FQDN
See *fully-qualified domain name (FQDN)*.

FSP
See *flexible service processor (FSP)*.

fully-qualified domain name (FQDN)
The complete domain name for a specific computer, or host, on the Internet. The FQDN consists of two parts: the hostname and the domain name.

G

GPFS cluster
A cluster of nodes defined as being available for use by GPFS file systems.

GPFS portability layer
The interface module that each installation must build for its specific hardware platform and Linux distribution.

GPFS Storage Server (GSS)
A high-performance, GPFS NSD solution made up of one or more building blocks that runs on System x servers.

GSS
See *GPFS Storage Server (GSS)*.

H

Hardware Management Console (HMC)
Standard interface for configuring and operating partitioned (LPAR) and SMP systems.

HMC
See *Hardware Management Console (HMC)*.

I

IBM Security Key Lifecycle Manager (ISKLM)
For GPFS encryption, the ISKLM is used as an RKM server to store MEKs.

independent fileset
A fileset that has its own inode space.

indirect block
A block that contains pointers to other blocks.

inode
The internal structure that describes the individual files in the file system. There is one inode for each file.

inode space
A collection of inode number ranges reserved for an independent fileset, which enables more efficient per-fileset functions.

Internet Protocol (IP)
The primary communication protocol for relaying datagrams across network boundaries. Its routing function enables internetworking and essentially establishes the Internet.

I/O server node
An ESS node that is attached to the ESS 3000 storage enclosures. It is the NSD server for the GPFS cluster.

IP
See *Internet Protocol (IP)*.

IP over InfiniBand (IPoIB)
Provides an IP network emulation layer on top of InfiniBand RDMA networks, which allows existing applications to run over InfiniBand networks unmodified.

IPoIB
See *IP over InfiniBand (IPoIB)*.

ISKLM
See *IBM Security Key Lifecycle Manager (ISKLM)*.

J

JBOD array
The total collection of disks and enclosures over which a recovery group pair is defined.

K

kernel
The part of an operating system that contains programs for such tasks as input/output, management and control of hardware, and the scheduling of user tasks.

L

LACP
See *Link Aggregation Control Protocol (LACP)*.

Link Aggregation Control Protocol (LACP)
Provides a way to control the bundling of several physical ports together to form a single logical channel.

logical partition (LPAR)
A subset of a server's hardware resources virtualized as a separate computer, each with its own operating system. See also *node*.

LPAR
See *logical partition (LPAR)*.

M

management network
A network that is primarily responsible for booting and installing the designated server and compute nodes from the management server.

management server (MS)
An ESS 3000 node that hosts the ESS 3000 GUI and xCAT and is not connected to storage. It must be part of a GPFS cluster. From a system management perspective, it is the central coordinator of the cluster. It also serves as a client node in an ESS 3000 building block.

master encryption key (MEK)
A key that is used to encrypt other keys. See also *encryption key*.

maximum transmission unit (MTU)
The largest packet or frame, specified in octets (eight-bit bytes), that can be sent in a packet- or frame-based network, such as the Internet. The TCP uses the MTU to determine the maximum size of each packet in any transmission.

MEK
See *master encryption key (MEK)*.

metadata
A data structure that contains access information about file data. Such structures include inodes, indirect blocks, and directories. These data structures are not accessible to user applications.

MS
See *management server (MS)*.

MTU
See *maximum transmission unit (MTU)*.

N

Network File System (NFS)
A protocol (developed by Sun Microsystems, Incorporated) that allows any host in a network to gain access to another host or netgroup and their file directories.

Network Shared Disk (NSD)
A component for cluster-wide disk naming and access.

NSD volume ID
A unique 16-digit hexadecimal number that is used to identify and access all NSDs.

node
An individual operating-system image within a cluster. Depending on the way in which the computer system is partitioned, it can contain one or more nodes. In a Power Systems environment, synonymous with *logical partition*.

node descriptor
A definition that indicates how IBM Spectrum Scale uses a node. Possible functions include: manager node, client node, quorum node, and non-quorum node.

node number
A number that is generated and maintained by IBM Spectrum Scale as the cluster is created, and as nodes are added to or deleted from the cluster.

node quorum
The minimum number of nodes that must be running in order for the daemon to start.

node quorum with tiebreaker disks
A form of quorum that allows IBM Spectrum Scale to run with as little as one quorum node available, as long as there is access to a majority of the quorum disks.

non-quorum node
A node in a cluster that is not counted for the purposes of quorum determination.

O

OFED
See *OpenFabrics Enterprise Distribution (OFED)*.

OpenFabrics Enterprise Distribution (OFED)
An open-source software stack includes software drivers, core kernel code, middleware, and user-level interfaces.

P

pdisk
A physical disk.

PortFast
A Cisco network function that can be configured to resolve any problems that could be caused by the amount of time STP takes to transition ports to the Forwarding state.

R

RAID
See *redundant array of independent disks (RAID)*.

RDMA
See *remote direct memory access (RDMA)*.

redundant array of independent disks (RAID)
A collection of two or more disk physical drives that present to the host an image of one or more logical disk drives. In the event of a single physical device failure, the data can be read or regenerated from the other disk drives in the array due to data redundancy.

recovery
The process of restoring access to file system data when a failure has occurred. Recovery can involve reconstructing data or providing alternative routing through a different server.

recovery group (RG)
A collection of disks that is set up by IBM Spectrum Scale RAID, in which each disk is connected physically to two servers: a primary server and a backup server.

remote direct memory access (RDMA)
A direct memory access from the memory of one computer into that of another without involving either one's operating system. This permits high-throughput, low-latency networking, which is especially useful in massively-parallel computer clusters.

RGD
See *recovery group data (RGD)*.

remote key management server (RKM server)
A server that is used to store master encryption keys.

RG
See *recovery group (RG)*.

recovery group data (RGD)
Data that is associated with a recovery group.

RKM server
See *remote key management server (RKM server)*.

S

SAS
See *Serial Attached SCSI (SAS)*.

secure shell (SSH)
A cryptographic (encrypted) network protocol for initiating text-based shell sessions securely on remote computers.

Serial Attached SCSI (SAS)
A point-to-point serial protocol that moves data to and from such computer storage devices as hard drives and tape drives.

service network
A private network that is dedicated to managing POWER8® servers. Provides Ethernet-based connectivity among the FSP, CPC, HMC, and management server.

SMP
See *symmetric multiprocessing (SMP)*.

Spanning Tree Protocol (STP)
A network protocol that ensures a loop-free topology for any bridged Ethernet local-area network. The basic function of STP is to prevent bridge loops and the broadcast radiation that results from them.

SSH
See *secure shell (SSH)*.

STP
See *Spanning Tree Protocol (STP)*.

symmetric multiprocessing (SMP)
A computer architecture that provides fast performance by making multiple processors available to complete individual processes simultaneously.

T

TCP
See *Transmission Control Protocol (TCP)*.

Transmission Control Protocol (TCP)
A core protocol of the Internet Protocol Suite that provides reliable, ordered, and error-checked delivery of a stream of octets between applications running on hosts communicating over an IP network.

V

VCD
See *vdisk configuration data (VCD)*.

vdisk
A virtual disk.

vdisk configuration data (VCD)
Configuration data that is associated with a virtual disk.

X

xCAT
See *Extreme Cluster/Cloud Administration Toolkit*.

Index

Product Number: 5765-DME
5765-DAE

GC28-3106-03