IBM Elastic Storage System 3000
Version 6.0.1.2

Quick Deployment Guide

SC28-3157-02

Note

Before using this information and the product it supports, read the information in "Notices" on page 61.

This edition applies to version 6 release 0 modification 1 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

Who should read this information

This information is intended for administrators of IBM Elastic Storage® System (ESS) that includes IBM Spectrum Scale RAID.

IBM Elastic Storage System information units

IBM Elastic Storage System (ESS) 3000 documentation consists of the following information units.

Information unit	Type of information	Intended users
Hardware Planning and Installation Guide	This unit provides ESS 3000 information including technical overview, planning, installing, troubleshooting, and cabling.	System administrators and IBM support team
Quick Deployment Guide	This unit provides ESS 3000 information including the software stack, deploying, upgrading, and best practices.	System administrators, analysts, installers, planners, and programmers of IBM Spectrum Scale clusters who are very experienced with the operating systems on which each IBM Spectrum Scale cluster is based
Service Guide	This unit provides ESS 3000 information including events, servicing, and parts listings.	System administrators and IBM support team
Problem Determination Guide	This unit provides ESS 3000 information including setting up call home, replacing servers, issues, maintenance procedures, and troubleshooting.	System administrators and IBM support team
Command Reference	This unit provides information about ESS commands and scripts.	System administrators and IBM support team
IBM Spectrum Scale RAID: Administration	This unit provides IBM Spectrum Scale RAID information including administering, monitoring, commands, and scripts.	• System administrators of IBM Spectrum Scale systems • Application programmers who are experienced with IBM Spectrum Scale systems and familiar with the terminology and concepts in the XDSM standard

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.

- Mellanox OFED (MLNX_OFED v4.9-0.1.7.0) Release Notes, go to https://docs.mellanox.com/display/OFEDv490170/Release%20Notes

Conventions used in this information

Table 1 on page viii 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*.

\	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 6.0.1.2 content stack

Component	Version
IBM Spectrum Scale	5.0.5.4
Red Hat® Enterprise Linux®	• The canister nodes in 5141-AF8 run Red Hat Enterprise Linux 8.1. • ESS management server (EMS) node: – POWER9™ (5105-22E) runs Red Hat Enterprise Linux 8.1. – POWER8® (5147-21L) runs Red Hat Enterprise Linux 7.7. • The container that resides on the EMS node runs Red Hat Enterprise Linux 7.7 Universal Base Image (UBI).
OFED	MLNX_OFED_LINUX-4.9-0.1.7.0 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 = 16.27.6008 • MT4120 = 16.27.6008 • MT4121 = 16.27.6008 • MT4122 = 16.27.6008
Kernel	4.18.0-193.29.1.el8_2
Systemd	239-18.el8
Network manager	1.20.0-3.el8
NVMe drive firmware	SN1MSN1M
Boot drive firmware (Smart)	1255
Boot drive firmware (Micron)	ML32
mpt3sas driver	32.100.00.00
Canister firmware (besides boot drive)	1111 (2.02.000_0B0G_1.73_FB300052_0C32.official)
BIOS level	52

Component	Version
Podman	• 1.4.4 (Red Hat Enterprise Linux 7.7) • 1.6.4 (Red Hat Enterprise Linux 8.1)
Ansible®	2.9.9
xCAT	2.15.1
Platform RPM	gpfs.ess.platform.ess3k-6.0.1-2.x86_64
Firmware RPM	gpfs.ess.firmware-6.0.0-6.x86_64
Support RPMs	• gpfs.gnr.support-ess3000-1.0.0-2.noarch • gpfs.gnr.support-essbase-1.0.0-2.noarch

ESS 3000 version 6.0.1.2 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_6.0.1.2_1202-03_dme.tgz`
- Data Access Edition

  `ess3000_6.0.1.2_1202-03_dae.tgz`

Fixes and improvements in ESS 3000 version 6.0.1.2

- Updated code stack
- General bug fixes and improvements

Support for signed RPMs

ESS or IBM Spectrum Scale RPMs are signed by IBM.

The PGP key is located in `/opt/ibm/ess/tools/conf`

```
-rw-r-xr-x 1 root root 907 Dec 1 07:45 SpectrumScale_public_key.pgp
```

You can check if an ESS or IBM Spectrum Scale RPM is signed by IBM as follows.

1. Import the PGP key.

   ```
   rpm --import  /opt/ibm/ess/tools/conf/SpectrumScale_public_key.pgp
   ```
2. Verify the RPM.

   ```
   rpm -K RPMFile
   ```

Security law changes

- New systems and switches shipped from manufacturing now have either an expired password (root password, switches) or one set to the serial number of the component (ASMI passwords).
- It is advised that all passwords are changed after the deployment is complete.

- The default root password for the OS is `ibmesscluster`. After the deployment is complete, it is advised that you change this password on each server. It is a best practice to use the same password on each node, but it is not mandatory.
- The default ASMI passwords (login, IPMI, HMC, etc.) are set to the serial number of the server. It is a best practice to set the IPMI password 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.

ESS 3000 and ESS 5000 server and networking requirements

In any scenario you must have an EMS node and a management switch. The management switch must be split into 2 VLANS.

- Management VLAN
- Service/FSP VLAN

You also need a high-speed switch (IB or Ethernet) for cluster communication.

ESS 3000

POWER8 or POWER9 EMS

POWER9 EMS is preferred if it is a new ESS 3000 system without legacy (POWER8) building-blocks.

- If you are adding ESS 3000 to a POWER8 EMS:
  - An additional connection for the container to the management VLAN must be added. A C10-T2 cable must be run to this VLAN.
  - A public/campus connection is recommended in C10-T3.
  - A management connection must be run from C10-T1 (This should be already in place if adding to an existing POWER8 EMS with legacy nodes).
- If you are using an ESS 3000 with a POWER9 EMS:
  - C11-T1 must be connected on the EMS to the management VLAN.
  - Port 1 on each ESS 3000 canister must be connected to the management VLAN.
  - C11-T2 must be connected on the EMS to the FSP VLAN.
  - HMC1 must be connected on the EMS to the FSP VLAN.

ESS 5000

POWER9 EMS support only

EMS must have the following connections:

- C11-T1 to the management VLAN
- C11-T2 to the FSP VLAN
- HMC1 to the FSP VLAN

ESS 5000 nodes must have the following connections:

- C11-T1 to the management VLAN
- HMC1 to the FSP VLAN

Chapter 2. ESS 3000 example setup flow

Assumptions:

New ESS customer

- Racked order
- POWER9 EMS
- One or more ESS 3000 systems
- Two or more POWER9 protocol nodes
- One or more ESS 5000 building-blocks

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

- TDA complete and system ordered.

  This includes network diagnostic and filling out of ESS 3000 worksheet.
- System arrives on-site.
- SSR arrives to physically set up the system (unpacking, power, etc.).
- SSR cables the high-speed switch connections to the ESS 3000 and ESS 5000 nodes, if they are available at the installation time.
- SSR begins code 20 checkout:
  - Starting with the Power® nodes, the SSR will use **essutils** to check and resolve any hardware issues and set IP addresses.
  - Ping test performed from the EMS node to the Power nodes (FSP and management VLANs).
  - ESS 3000 nodes are checked last and a ping test is performed from each canister to the EMS over the management VLAN.
  - After the visual inspection is completed, the system is handed over to the customer or LBS

  The system is handed over to the customer
- Customer downloads the latest ESS 3000 and ESS 5000 installation package (`.tgz` file) from IBM Fix Central.
- **[ESS 3000 only]** If high-speed ports need to be changed from Infiniband/Infiniband on a VPI card (MT4121), the customer must log in to each node and make the necessary changes before proceeding. For more information, see Appendix H, “ConnectX-5 VPI support on ESS 3000,” on page 55. By default, the ports are set to IB/IB.
- Customer sets up `/etc/hosts` using best practices.
- Customer sets up a public or campus connection on C11-T3.
- Customer extracts code in `/home/deploy` for the ESS 3000.
- Customer runs the installer binary file. The following steps are done as a result:
  - Checksum is verified
  - Installer package is extracted
  - **essmkyml** is run automatically and it prompts the user to confirm the following items:
    - Verify or set the EMS hostname and domain (FQDN).
    - Input the container host name.
    - Specify the management network bridge (POWER8 EMS only).
    - Start the container.
- By using the Ansible wrapper (**essrun**), the customer performs the following steps on the ESS 3000 nodes and EMS:

  - Customer configures environment and sets up passwordless SSH.
  - Customer upgrades the EMS node, if required.
  - Customer upgrades the canister nodes in parallel (offline), if required.
  - Customer creates network bonds, cluster, and file system (metadata and data combined).
  - Customer enables TRIM on the ESS 3000 file system. For more information, see *Managing TRIM support for storage space reclamation* in *IBM Spectrum Scale RAID: Administration*.
  - Customer leaves container (now on EMS) and sets up performance collection.
  - Customer performs final health checks.

ESS 5000 and protocol node additions

- Customer shuts down the ESS 3000 container.
- Customer extracts the ESS 5000 tgz file.
- Customer accepts license and installs the container image.
  - **essmkyml** is automatically started and it prompts the user for the following items:
    - Confirmation of the EMS FQDN
    - Container host name.
    - Container FSP interface name

    If all checks pass, the ESS 5000 container is started.
- By using the Ansible wrapper (**essrun**), the customer performs the following steps on the ESS 3000 nodes and EMS:
  - Customer configures environment and sets up passwordless SSH (ESS 5000 nodes and protocol nodes).
  - Customer upgrades the ESS 5000 nodes and protocol nodes in parallel (offline), if required.
  - Customer creates network bonds (ESS 5000 nodes and protocol nodes).
  - Logging into one node, customer uses **essaddnode** to add the ESS 5000 nodes to the ESS 3000 environment.
  - Using **mmvdisk** directly, the customer creates data-only NSDs and adds them to the ESS 3000 file system (dataOnly pool).

    Note: Leave enough space for the CES shared root file system.
  - Customer sets up a policy file to move data between the two pools (system and data).
  - Back on the container, the customer uses **essrun** to create the CES shared root file system.

    This step can also be done on one of the ESS 5000 nodes by using **mmvdisk**.
  - Customer adds the ESS 5000 nodes into **mmperfmon** (sensor nodes).
  - Customer uses the IBM Spectrum Scale installation toolkit to install and configure the POWER9 protocol nodes
  - Time server is configured.
  - GUI and call home are now configured for all cluster nodes.
  - Final health checks are performed.
  - Client nodes are added and workload begins.

Chapter 3. ESS 3000 best practices

- Upgrades must not be performed on an unhealthy system. Ensure 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:
  - 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.

  There is some flexibility in this case, but if all nodes are not at the same level, including clients in same cluster, you will not be able to migrate the release level or the file system format.
- Management switch must be an isolated, flat network or VLAN(s).
- Tracing must be disabled on the GPFS cluster before performing an upgrade.

  ```
  mmtracectl --stop
  mmtracectl --off
  ```
- Enable chroynd (NTP) after installation or upgrade is complete. For more information, see Appendix E, “How to set up chronyd (time server),” on page 45.
- Consider enabling TRIM after installation or upgrade. For more information, see *Managing TRIM support for storage space reclamation* in *IBM Spectrum Scale RAID: Administration*.
- Do not use any special characters, underscores, or dashes in the host names other than the high speed suffix (example: -hs). Doing this might cause issues with the deployment procedure.

Chapter 4. ESS 3000 deployment scenarios

An ESS 3000 system might be deployed in one of the following scenarios.

Scenario 1 - New customer (no existing ESS infrastructure)

In this configuration, you receive the following items for a typical racked (or rackless) installation:

- 5105-22E POWER9 EMS node
- One or more ESS 3000 nodes
- 1/10Gb management switch with two isolated, flat VLANs (management and service)
- One high-speed switch (Ethernet or Infiniband)

Optional:

- One or more ESS 5000 building-blocks
- 2 or more POWER9 protocol nodes

This is the simplest configuration. ESS 6.0.1.x running on the POWER9 EMS node can run either ESS 3000 or ESS 5000 container and it can deploy or manage one or more POWER9 protocol nodes. If the EMS node needs to be upgraded, you can do so from either container.

Note: For the full EMS or protocol node upgrade (including repository files, IPR, etc.), you must use an ESS 5000 container.

Scenario 2 - Existing ESS 3000 customer adding ESS 5000 (I/O nodes or protocol nodes)

In this scenario, it is advised that you keep your ESS 3000 container running on the POWER8 EMS node. The ESS 5000 building-block(s) and protocol node(s) run on the POWER9 EMS node. Both EMS nodes are designated as collector nodes and they run an instance of the ESS GUI.

Scenario 3 - Existing ESS 3000 customer upgrading to ESS 6.0.1.x (with no legacy nodes)

In this scenario, the POWER8 EMS node is upgraded from the ESS 3000 (version 6.0.1.x) container. Only IBM Spectrum Scale, OFED, and tools RPMs are upgraded in this scenario. Upgrading the ESS 3000 nodes is the same as the normal procedure.

Scenario 4 - Existing ESS 3000 customer upgrading to ESS 6.0.1.x (with legacy nodes)

In this scenario, customers retain their POWER8 EMS node and run the new ESS 3000 6.0.1.x container. Because legacy nodes are in the picture, the EMS node, POWER8 building blocks, and POWER8 protocol nodes are all upgraded to ESS 5.3.6 using the legacy flow. When completed, the ESS 3000 canister nodes is updated from the ESS 6.0.1.x container.

For scenarios 3 and 4, customers are encouraged to continue running the ESS 3000 container from the POWER8 EMS node, even though they can buy a POWER9 EMS node and run the container from there. ESS 5000 nodes are only supported on POWER9 EMS node and legacy systems (including POWER8 protocol nodes) are only supported on POWER8 EMS node. For additional information on scenarios 3 and 4, see Appendix B, "ESS 3000: EMS upgrade scenarios," on page 31.

Scenario-specific guidance

- You do not need to upgrade to the latest ESS version on the EMS node to support ESS 3000. If upgrading from the ESS 3000 container, at least the IBM Spectrum Scale, OFED, and the tools RPMs are installed to match the versions of these items on the ESS 3000 canisters.
- If you chose to fully upgrade the POWER8 EMS node to the latest ESS version (ESS 5.3.6.x), you may do so by using the legacy upgrade instructions. For more information, see E.S.S 5.3.6x Quick Deployment Guide in the IBM Knowledge Center. Although, this is optional it is preferred by most customers to have the full stack at the latest versions.
- If you have any legacy nodes (Building-blocks or POWER8 protocol nodes), you must use the legacy instructions to deploy or upgrade.

- To fully upgrade the POWER9 EMS node, you must do so from the ESS 5000 container. If you are upgrading the POWER9 EMS node from an ESS 3000 container, only IBM Spectrum Scale, OFED, and tools RPMs are upgraded.

Important: Upgrade of the POWER8 EMS only works from the container if xCAT and the `gpfs.gss.tools` RPM are not installed. If you need to upgrade the EMS node in this scenario, you must remove both items before proceeding.

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.

Important: If high-speed ports need to be changed from Infiniband/Infiniband on a VPI card (MT4121), the customer must log on to each node and make the necessary changes before proceeding. For more information, see Appendix H, "ConnectX-5 VPI support on ESS 3000," on page 55. By default, the ports are set to IB/IB.

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`. SSR initially sets this to `ibmesscluster` during code 20. After the deployment is complete, customers are encouraged to change the root password on each ESS 3000 node. Ensure that the password is same on each node.
- 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 6.0.1 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_6.0.1.2_1202-03_dme.tgz`, which is the installation package for the Data Management Edition. 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 --get-version`**

A sample output is as follows:

```
Start of install check
nodelist:  localhost

Node: localhost                  Installed version:              ess3000_6.0.1.2_1202-03_dme
[PASS] essinstallcheck passed successfully
```

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.

Set up a campus or a public connection

Connect an Ethernet cable from C11-T3 (on POWER9 EMS) or C10-T3 (on POWER8 EMS) to your lab network. This connection serves as a way to access the GUI or the ESA agent (call home) from outside of the management network. The container creates a bridge to the management network, thus having a campus connection is highly advised.

Note:

- It is recommended but not mandatory to set up a campus or public connection. If you do not set up a campus or a public connection, you will temporarily lose your connection when the container bridge is created in a later step.

  This method is for configuring the campus network, not any other network in the EMS node. Do not modify T1, T2, or T4 connections in the system after they are set by SSR, and use the SSR method only to configure T1 and T2. That includes renaming the interface, setting IP, or any other interaction with those interfaces.
- For POWER9, the interface to set for the campus connection is enP1p8s0f2 (C11-T3).
- For POWER8, the interface to set for the campus connection is enP3p9s0f2 (C10-T3).

You can use the **`nmtui`** command to set the IP address of the campus interface. For more information, see Configuring IP networking with nmtui.

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

Complete the /etc/hosts file on the EMS node

Note: Setting the FQDN is no longer a hard requirement to proceed. It just needs to be defined in `/etc/hosts`. When the installer starts, you are prompted to confirm or set the FDQN.

This file must contain the low-speed (management) and high-speed (cluster) IP addresses, FQDNs, and short names. The high-speed names must contain a suffix to the low-speed names (For example, essio1-hs (high-speed name) to essio1 (low-speed name)). This file must also contain the container host name and the IP address.

```
127.0.0.1 localhost localhost.localdomain.local localhost4 localhost4.localdomain4

## Management IPs 192.168.45.0/24
192.168.45.20 ems1.localdomain.local ems1
192.168.45.21 essio1.localdomain.local essio1
192.168.45.22 essio2.localdomain.local essio2
192.168.45.23 prt1.localdomain.local prt1
192.168.45.24 prt2.localdomain.local prt2

## High-speed IPs 10.0.11.0/24
10.0.11.1 ems1-hs.localdomain.local ems1-hs
10.0.11.2 essio1-hs.localdomain.local essio1-hs
10.0.11.3 essio2-hs.localdomain.local essio2-hs
10.0.11.4 prt1-hs.localdomain.local prt1-hs
10.0.11.5 prt2-hs.localdomain.local prt2-hs

## Container info 192.168.45.0/24
192.168.45.80 cems0.localdomain.local cems0

## Protocol CES IPs
10.0.11.100 prt_ces1.localdomain.local prt_ces1
10.0.11.101 prt_ces1.localdomain.local prt_ces1
10.0.11.102 prt_ces2.localdomain.local prt_ces2
10.0.11.103 prt_ces2.localdomain.local prt_ces2
```

Note: `localdomain.local` is just an example and cannot be used for deployment. You must change it to a valid fully qualified domain name (FQDN) during the `/etc/hosts` setup. The domain must be the same for each network subnet that is defined. Also, make sure that you set the domain on the EMS node (**`hostnamectl set-hostname`** *NAME*).

NAME must be the FQDN of the management interface (T1) of the EMS node. If you need to set other names for campus, or other interfaces, those names must be the alias but not the main host name as returned by the **`hostnamectl`** command.

Configuring the EMS node for ESS 3000

1. **[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
```

c. Clean up network bridges.

- For versions earlier than ESS 3000 6.0.1.0, the container bridge is under the **brctl** control. Clean up by using the following commands:

```
brctl show
ip link set BridgeName down ; brctl delbr BridgeName
```

- For versions ESS 3000 6.0.1.0 or later, the container bridge is under the **nmcli** control. Clean up by using the following commands:

```
nmcli c
nmcli c del BridgeName
```

2. **[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 11.

The name of the package is in this format: `ess3000_6.0.1.2_1202-03_dme.tgz`.

```
tar zxvf ess3000_6.0.1.2_1202-03_dme.tgz
```

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

```
ess3000_6.0.1.2_1202-03_dme.sh
ess3000_6.0.1.2_1202-03_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. Accept the license and install the accepted image.

```
./ess3000_6.0.1.2_1202-03_dme.sh  --text-only --start-container
```

Note: The `--install-image` flag will be deprecated soon. Stop and remove any existing container.

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_6.0.1.2_1202-03_dme.dir
├─ ess3000_6.0.1.2_1202-03_dme_binaries.iso
├─ ess3000_6.0.1.2_1202-03_dme.tar
├─ rhels-8.1-server-x86_64.iso
├─ podman_rh7.tgz
├─ Release_note.ess3000_6.0.1.2_1202-03_dme.txt
├─ podman_rh8.tgz
├─ essmkyml
├─ essmgr_p9.yml
├─ essmgr_p8.yml
├─ essmgr
├─ classes
├─ 3k_rh7_deps.tgz
├─ data
```

After you accept the license, the binary runs the following sequence automatically.

1) Extracts the contents of the tgz file.
2) Runs **essmkyml** and prompts users to:
   a) confirm the EMS FQDN
   b) confirm the container hostname
   c) set the management bridge IP (if POWER8 EMS)

3. Container installation questions:

   For this step, you must provide these inputs:

   - container name (must be in /etc/hosts or be resolvable by using DNS)
   - container management IP address (must be on the same network block that is set on C11-T1; C10-T1, if POWER8 EMS)
   - The EMS host name must be defined in /etc/hosts and associated with the f0 interface.

   ```
   Is the current EMS FQDN p8ems2.test.net correct (y/n):
   ```

   - If the FQDN is already set, enter y.

     If the FQDN is incorrect or it needs to be changed, enter n and then enter the new FQDN.
   - The EMS host name must be on the management network (also called xCAT). Other networks can be aliases (A) or canonical names (CNAME) on DNS or in the /etc/hosts file.
   - Remember not to add the DNS domain localdomain to the input:

   ```
   Please type the desired and resolvable short hostname [ess5k-cems0]:
   ```

   - **Note: [POWER8 only]** Ensure that a unique management bridge is set on the subnet (used for host to container communication). Remember that reg IP address must belong to the 192.168.20.0/24 network block.

   ```
   Please type the Management bridge IP [192.168.20.2]:
   ```

   Note: The values in parentheses ([ ]) are just examples or the last entered values.

   If all of the checks pass, the essmgr.yml file is written and you can proceed to bridge creation, if applicable, and running the container.

   Note: The original essmgr.yml file and detailed logs of checks that are performed are stored in the ./logs directory.

   If all provided inputs in this step pass the verification, the following tasks are done.

   - The podman image is installed.
   - Management bridge is set.
   - Management IP address is moved to the bridge or, on POWER8, unique management bridge IP address is set.

- Container is started.

After these tasks are done, the user is not inside the container and they can continue with the deployment.

For example:

```
2020-12-01 23:45:10,417 INFO:    Going to start the container. On further runs use 'essmgr'
command to manage this container
-- [INFO] Going to run container image - ess3000_6.0.1.2_dme:1202-03
-- [INFO] Bridge does not exist. Adding... --
-- [INFO] Adding interface enP3p9s0f1 to bridge mgmt_bridge --
-- [INFO] Automatic initialization of the container begin shortly --
-- [INFO] Please be patient as the startup can take a few minutes --
-- [INFO] until you get the shell inside the container after init --
-- [INFO] ISO file rhels-8.1-server-x86_64.iso found. Mounting it now --
-- [INFO] ISO file ess3000_6.0.1.2_1202-03_dme_binaries.iso found. Mounting it now --
-- Starting container first time. Initializing container --
-- Configuring container for first time use --
-- Container initialization finished --
-- Refer /tmp/configure_cems.log file for detailed logs --
ESS 3000 CONTAINER root@cems0:/ #
```

Chapter 6. ESS 3000 initial setup instructions

Before doing these steps, ensure that you have completed the steps in Chapter 5, “ESS 3000 common setup instructions,” on page 11.

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 **essrun** command can be run only from within the container.
- The **essrun 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 11.
- 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.
- [POWER8 EMS only] 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 5.3.6 or later. For more information, see ESS 5.3.6x Quick Deployment Guide.
- The POWER8 EMS node update only upgrades the IBM Spectrum Scale, OFED, and gpfs.ess.tools RPM. It does not update the OS, firmware, or any other items in the stack. The update primarily keeps the EMS node at ESS 5.3.6.x levels. If you want to fully update to ESS 5.3.6.2 or 6.0.1.2 on the EMS, it must be a POWER9 EMS and the update must be done by using the ESS 5000 container.

1. Configure the ESS 3000 setup.

   ```
   essrun  -N ess3k1a,ess3k1b,ems1 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 **essrun 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 **essrun** to run commands as a group instead of specifying nodes in a comma-separated list with `-N`.

2. Update the EMS node.

   Note:

   - Do EMS update only if you do not have POWER8 ESS or POWER8 protocol nodes. If you have ESS or protocol nodes, refer to the *ESS 5.3.6x Quick Deployment Guide* and Chapter 4, “ESS 3000 deployment scenarios,” on page 9. The following command updates IBM Spectrum Scale including firmware, GUI, and call home to ESS 5.3.6.x code levels.
   - If this is a new EMS node from manufacturing (POWER9 EMS), it should already have ESS 6.0.1.x installed. Use **essinstallcheck -N localhost** to verify the ESS version. If the EMS is up-to-date, you can skip this step.

If this is a POWER8 EMS node without any legacy nodes (I/O server or protocol nodes), use **gssinstallcheck -N localhost** to verify the ESS version.

If this is a new POWER8 EMS node from manufacturing, it should already have ESS 5.3.6.x installed. If the EMS is up-to-date, you can skip this step. For more information, see Appendix B, "ESS 3000: EMS upgrade scenarios," on page 31.

```
essrun -N ems1 update --offline
```

3. Update the canister nodes.

```
essrun -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. **essrun** 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 **essrun** *Nodes* **config load** is already run, you might use the `-G` option for the following commands instead of using `-N`. For example, **essrun -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. Review VPI considerations, if applicable.

   If your system has ConnectX 5 VPI cards and if not already completed, ensure that the VPI cards' ports are already in the desired mode. By default, each port is configured as IB/IB. If you require any other configuration, see Appendix H, "ConnectX-5 VPI support on ESS 3000," on page 55. If you want a mix of IB and Ethernet per card, you cannot run the Ansible network playbook. Use **essgennetworks** from one of the canisters to create multiple bonds, unless you are using IB just for RDMA.

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

   ```
   essrun -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.

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

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

4. Create the IBM Spectrum Scale cluster.

   ```
   essrun -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.

5. 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.

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

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

   ```
   essrun -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.

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

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

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. From the EMS node, run the final health checks.

   ```
   mmhealth node show -a
   gnrhealthcheck
   ```

   On each canister node, run the following command:

   ```
   essinstallcheck -N localhost
   ```

5. Set up call home. For more information, see Drive call home.

   The supported call home configurations are:

   - Software call home
   - Node call home (including for protocol nodes)
   - Drive call home

6. Set the time zone and set up Chrony.

   Before getting started, ensure that NTP and time zone are set correctly on the EMS and canister nodes. Refer to Appendix E, "How to set up chronyd (time server)," on page 45 to perform these tasks before proceeding.

7. Refer to Appendix I, "Client node tuning recommendations," on page 57.

Chapter 7. ESS 3000 upgrade instructions

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

- Online upgrade
- Offline upgrade

Important: You can only fully update the POWER9 EMS node to ESS 6.0.1.2 from the ESS 5000 6.0.1.2 container. If you have an ESS 3000 container only, it only updates the following items:

- IBM Spectrum Scale
- OFED
- gpfs.ess.tools RPM

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 11. 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 6.0.1.x).
2. Run the configuration load.

   ```
   essrun -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.

   ```
   essrun -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 5.3.6x 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. 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.

  ```
  essrun -G ess_x86_64 update
  ```

- Update by using the individual nodes.

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

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

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

6. Start the performance monitoring sensors on each canister node.

   ```
   systemctl start pmsensors
   ```

7. 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 11. 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 6.0.1.x).
2. Run the configuration load.

   ```
   essrun -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.

   ```
   essrun -N ems1 update --offline
   ```

   Note:

   - Run the EMS update only if there are no POWER8 ESS or POWER8 protocol nodes in the environment. If there are ESS or protocol nodes in your setup, refer to *ESS 5.3.6x Quick Deployment Guide*.

     If you have POWER9 EMS, you cannot upgrade from the ESS 3000 container. You must update from the ESS 5000 container.

- 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.

     ```
     essrun -G ess_x86_64 update --offline
     ```

   - Update by using the individual nodes.

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

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

5. 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. This includes any services such as GUI, sensors, or call home. After the cluster is online, run the following health checks from the EMS node and ensure that the GUI is free of any issues:

- **`gnrhealthcheck`**
- **`mmhealth node show -a`**

Appendix A. IBM Elastic Storage System (ESS) known issues

Known issues in ESS version 6.0.1.2

The following table describes the known issues in ESS version 6.0.1.2 and how to resolve these issues.

Issue	Resolution or action	Product

`JAVA_HOME` might be pointing to the wrong version which might cause ESA startup to fail:
In the following example, note how Java™ is pointing to the wrong location. This causes the ESA startup to fail:
`# 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`
	To fix the problem, remove the current `java` symbolic link, update the `java` pointer, and retry the ESA activation.
	1. Remove the current `java` symbolic link.
	`# cd /etc/alternatives/`
	`# rm java`
	`rm: remove symbolic link ‘java’? y`
	2. Update the `java` pointer.
	`# 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`
	`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`
	3. Retry the ESA activation.
	`# /opt/ibm/esa/bin/activator -C -p 5024 -w -Y`
		ESS 3000

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.	No action is required.	ESS 3000
During rolling upgrade, **mmhealth** might show the error `local_exported_ fs_unavail` even though the file system is still mounted.	During a rolling upgrade (Updating of one ESS I/O 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.	- ESS 3000 - ESS 5000

<table>
<tr><th>Issue</th><th>Resolution or action</th><th>Product</th></tr>
<tr><td></td><td>

```
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)
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**.

</td><td></td></tr>
<tr><td>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.</td><td>Wait for the timeout and retry the <b>essrun</b> update task.</td><td>ESS 3000</td></tr>
<tr><td>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.</td><td>Re-run the <b>mmchfirmware –type drive</b> command which should resolve the issue and update the remaining drives.</td><td>ESS 3000</td></tr>
<tr><td>

When running **essrun** 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
********
```

</td><td>

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. Remove `,profile_tasks` on line 7.
3. Save and quit: `esc + :wq`

</td><td>

- ESS 3000
- ESS 5000

</td></tr>
<tr><td>

When running the **essrun config load** command, you might see a failure such as this:

```
stderr: |-
rc=2 code=186
Failed to obtain the enclosure
device
```

</td><td>

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*
   ```

</td><td>ESS 3000</td></tr>
</table>

Issue	Resolution or action	Product

`name with rc=2`
`rc=2 code=669`
	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 **essrun config load** command from the container.	

Running **essrun -N node1,node2,… config load** command with high-speed names causes issues with the upgrade task using the `-G` flag.	The **essrun 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: **essrun -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: **essrun -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`.	• ESS 3000 • ESS 5000

After reboot of an ESS 5000 node, systemd could be loaded incorrectly.
Users might see the following error when trying to start GPFS:
`Failed to activate service`
`'org.freedesktop.systemd1':`
`timed out`
	Power off the system and then power it on again.
	1. Run the following command from the container:
	rpower *NodeName* **off**
	2. Wait for at least 30 seconds and run the following command to verify that the system is off:
	rpower *NodeName* **status**
	3. Restart the system with the following command.
	rpower *NodeName* **on**
		ESS 5000

In ESS 5000 SLx series, after pulling a hard drive out for a long time wherein the drive has finished draining, when you re-insert the drive, the drive could not be recovered.	Run the following command from EMS or IO node to revive the drive: **mmvdisk pdisk change --rg** *RGName* **--pdisk** *PdiskName* **--revive** Where *RGName* is the recovery group that the drive belongs to and *PdiskName* is the drive's pdisk name.	ESS 5000
After the deployment is complete, if firmware on the enclosure, drive, or HBA adapter does not match the expected level, and if you run **essinstallcheck**, the following	The error about **mmvdisk** settings can be ignored. The resolution is to update the mismatched firmware levels on enclosure, adapter, or HBA adapters to the correct levels. You can run the **mmvdisk configuration check** command to confirm. List the **mmvdisk** node classes: **mmvdisk nc list**	• ESS 3000 • ESS 5000

Issue	Resolution or action	Product
mmvdisk settings related error message is displayed: `[ERROR] mmvdisk settings do NOT match best practices. Run mmvdisk server configure --verify --node-class ess5k_ppc64le_mmvdisk to debug.`	**Note: essinstallcheck** detects inconsistencies from **mmvdisk** best practices for all node classes in the cluster and stops immediately if an issue is found.	
When running **essinstallcheck** you might see an error message similar to: `System Firmware could not be obtained which will lead to a false-positive` **`PASS`** `message when the script completes.`	Rerun **essinstallcheck** which should properly query the firmware level.	ESS 5000
When running the **essrun - N** *Node* **healthcheck** command, the **essinstallcheck** script might fail due to incorrect error verification which might lead to an impression that there is a problem where there is none.	This health check command (**essrun - N** *Node* **healthcheck**) is removed from the ESS documentation and it is advised to use the manual commands to verify system health after deployment. Run the following commands for health check: • **gnrhealthcheck** • **mmhealth node show -a** • **essinstallcheck -N localhost**: This command needs to be run on each node.	• ESS 3000 • ESS 5000
During command-less disk replacement, there is a limit on how many disks can be replaced at one time.	For command-less disk replacement using commands, only replace up to 2 disks at a time. If command-less disk replacement is enabled, and more than 2 disks are replaceable, replace the 1st 2 disks, and then use the commands to replace the 3rd and subsequent disks.	• ESS 3000 • ESS 5000
Issue reported with command-less disk replacement warning LEDs.	The replaceable disk will have the amber led on, but not blinking. Disk replacement should still succeed.	ESS 5000

After upgrading an ESS 3000 node to version 6.0.1.2, the `pmsensors` service needs to be manually started.	After the ESS 3000 upgrade is complete, the `pmsensors` service does not automatically start. You must manually start the service for performance monitoring to be restored. On each ESS 3000 canister, run the following command:
	`systemctl start pmsensors`
	For checking the status of the service, run the following command:
	`systemctl status --no-pager pmsensors`
		ESS 3000

ESS commands such as **essstoragequickcheck**, **essinstallcheck** must be run using **-N localhost**. If using the hostname such as **-N ess3k1a**, an error occurs.	There is currently an issue with running the ESS deployment commands by using the hostname of a node. The workaround is to run checks locally on each node by using localhost. For example: **`essstoragequickcheck -N localhost`**	• ESS 3000 • ESS 5000

Issue	Resolution or action	Product

Hyperthreading might be enabled on an ESS 3000 system due to an incorrect kernel grub flag being set.	Hyperthreading needs to be disabled on ESS 3000 systems. This is ensured in following ways:
	• Disabled in BIOS
	• Disabled using the tuned profile
	• Disabled using the grub command line
	When disabled with the grub command line, the issue occurs because the grub configuration had an incorrect flag set in earlier versions. To resolve this issue, do the following:
	1. Edit the `/etc/grub2.cfg` file to change nohup with `nosmt`.
	Before change:
	`set`
	`default_kernelopts="root=UUID=9a4a93b8-2e6b-4ba6-`
	`bda4-a7f8c3cb908f`
	`ro nvme.sgl_threshold=0 sshd=1 pcie_ports=native`
	`nohup`
	`resume=UUID=c939121b-526a-4d44-8d33-693f2fb7f018`
	`rd.md.uuid=f6dbf6f2:8ac82ed6:875ca663:0094ac11`
	`rd.md.uuid=06c2d5b0:c6603a1e:5df4b4d3:98fd5adc`
	`rhgb quiet crashkernel=4096M"`
	After change:
	`set`
	`default_kernelopts="root=UUID=9a4a93b8-2e6b-4ba6-`
	`bda4-a7f8c3cb908f`
	`ro nvme.sgl_threshold=0 sshd=1 pcie_ports=native`
	`nosmt`
	`resume=UUID=c939121b-526a-4d44-8d33-693f2fb7f018`
	`rd.md.uuid=f6dbf6f2:8ac82ed6:875ca663:0094ac11`
	`rd.md.uuid=06c2d5b0:c6603a1e:5df4b4d3:98fd5adc`
	`rhgb quiet crashkernel=4096M"`
	2. Reboot the node for the changes to take effect.
		ESS 3000

Race condition in `opal-elog` that can hit a kernel panic in function `elog_work_fn`. This is experienced when the GUI is running HW_INVENTORY commands to POWER servers.	This issue was found with RHEL 7 kernel (Bugzilla 1873189) while `opal-elog` is handling an excessive amount of OPAL error log events. The GUI runs **`ipmi fru print`** commands as part of its HW_INVENTORY checks. The bug might be hit during these intervals due to the excessive amount of OPAL events are being generated. A fix is being worked on by Red Hat to provide a new kernel to address this race condition. There is a known issue with OPAL on Power nodes wherein too many OPAL requests might cause a system hang. This issue does not affect ESS 3000 nodes. In response, consider disabling the **HW_INVENTORY GUI** task to reduce requests to the FSP. `/usr/lpp/mmfs/gui/cli/chtask HW_INVENTORY --inactive`	• ESS 3000 (EMS node only) • ESS 5000
Redeploying the EMS node fails due to wrong firmware version in `otherpkglist`.	The osimage used to deploy the EMS is: `rhels8.1-ppc64le-install-ems`	• ESS 3000

Issue	Resolution or action	Product

	Currently, the `otherpkglist` has a bug. It is pointing to the old firmware version (6004):
	`/opt/ibm/ess/xcat/install/rh/`
	`ems.rhels8.ppc64le.otherpkgs.pkglist`
	`gpfs/gpfs.ess.firmware-6.0.0-4*`
	To fix, replace `gpfs/gpfs.ess.firmware-6.0.0-4*` with:
	`gpfs/gpfs.ess.firmware*`
		- ESS 5000

`No suitable node found` error when running deployment commands on EMS.	Currently, the ESS code mistakenly looks for the NVMe driver on the EMS node to determine the node type. This issue causes commands to not work when run on the host EMS.
	To fix, remove the NVMe driver and update the modules configuration file on the EMS node as follows:
	`Modprobe -r nvme`
	`echo sg >  /etc/modules-load.d/ess.conf`
		- ESS 3000
		- ESS 5000

`essinstallcheck` on the EMS might flag the ipr RPM as unsigned:
`[ERROR] File /install/ess/`
`otherpkgs/rhels8/ppc64le/`
`firmware/`
`pci.1014034A.51-19512900-1.Linu`
`x.noarch.rpm is not signed.`
	The pci RPM should not be in the list of RPMs checked for signing status. Ignore this error.	- ESS 3000
		- ESS 5000

Python related issues on POWER EMS node:
`ERROR:Detected default version`
`Python 2.7.5 as python 2.`
`Python 2 is not supported`
`ERROR:Either we are running on`
`python 2 or the default OS`
`python version is python 2.`
`This is not supported`
	Fix these errors as follows:
	`update-alternatives --install /usr/bin/python`
	`python /usr/bin/python3 1`
	`update-alternatives --config python`
	`python --version`
		ESS 3000

Appendix B. ESS 3000: EMS upgrade scenarios

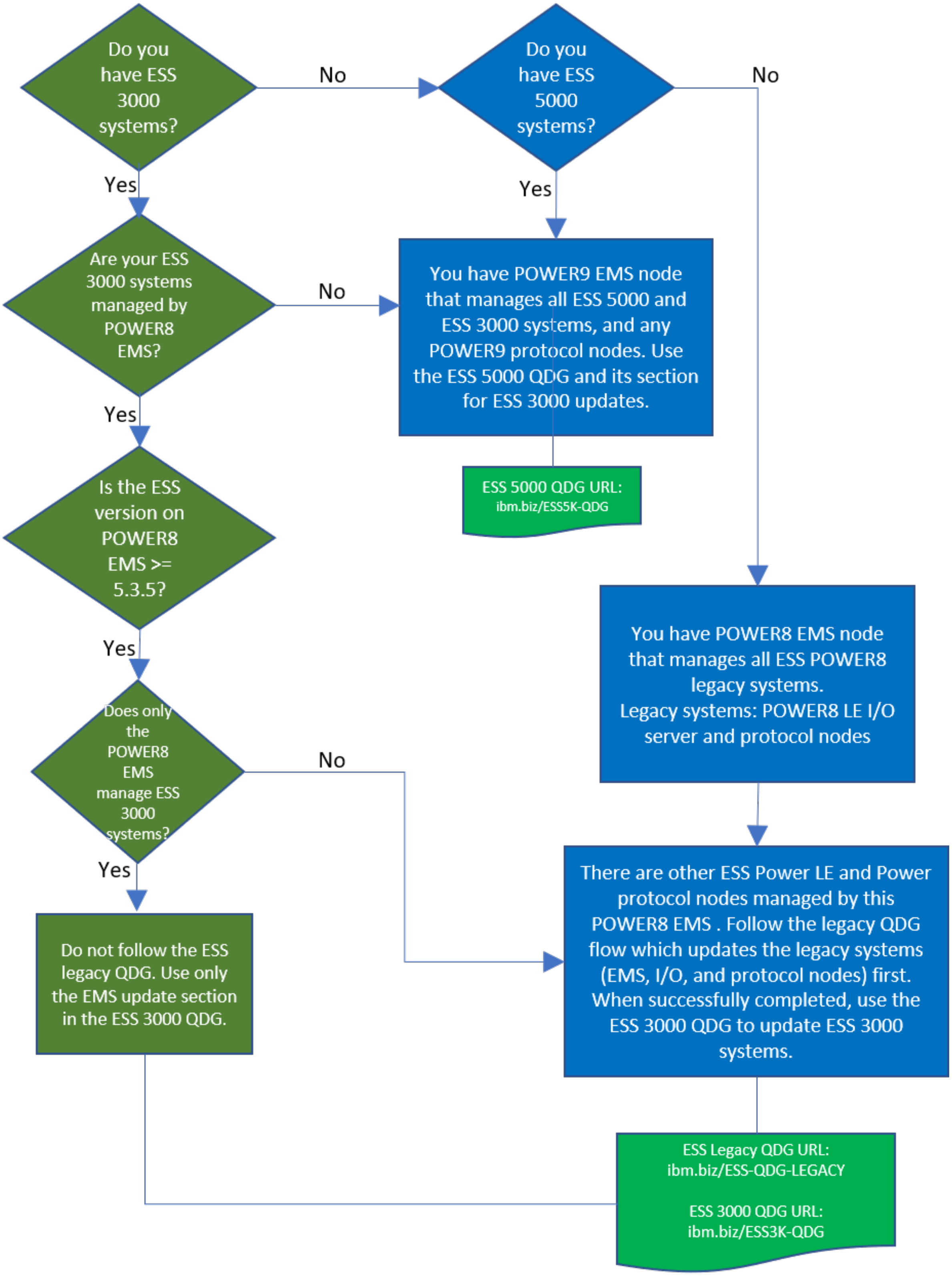
Learn how and when to upgrade the POWER8 EMS node in an ESS 3000 environment.

- If POWER8 EMS, the upgrade procedure updates IBM Spectrum Scale, OFED, and the gpfs.ess.tools RPM. For the upgrade procedure to work, xCAT and gpfs.gss.tools RPM must not be installed on the host EMS.
- If POWER9 EMS, you cannot upgrade from the ESS 3000 container. Full upgrade is supported from the ESS 5000 container only.

There are two scenarios in which this upgrade might be needed:

- Legacy ESS setup with ESS 3000 added
- Standalone ESS 3000 setup

In both scenarios, there is a single POWER8 EMS node that manages ESS 3000 and legacy ESS and protocol nodes.


Do you have ESS 3000 systems?
No
Do you have ESS 5000 systems?
No
Yes
Yes
Are your ESS 3000 systems managed by POWER8 EMS?
No
You have POWER9 EMS node that manages all ESS 5000 and ESS 3000 systems, and any POWER9 protocol nodes. Use the ESS 5000 QDG and its section for ESS 3000 updates.
Yes
ESS 5000 QDG URL: ibm.biz/ESS5K-QDG
Is the ESS version on POWER8 EMS >= 5.3.5?
Yes
You have POWER8 EMS node that manages all ESS POWER8 legacy systems. Legacy systems: POWER8 LE I/O server and protocol nodes
Does only the POWER8 EMS manage ESS 3000 systems?
No
Yes
There are other ESS Power LE and Power protocol nodes managed by this POWER8 EMS . Follow the legacy QDG flow which updates the legacy systems (EMS, I/O, and protocol nodes) first. When successfully completed, use the ESS 3000 QDG to update ESS 3000 systems.
Do not follow the ESS legacy QDG. Use only the EMS update section in the ESS 3000 QDG.
ESS Legacy QDG URL: ibm.biz/ESS-QDG-LEGACY
ESS 3000 QDG URL: ibm.biz/ESS3K-QDG

The minimum requirements for both scenarios are:

- POWER8:
  - EMS is at ESS 5.3.5 or later.

  - POWER8 protocol node deployment is not supported from ESS 3000 container (Must use the legacy method).
  - If legacy nodes, update EMS by using the legacy method.
  - If legacy nodes are not involved, you can update EMS by using the container (EMS host cannot have xCAT or the tools RPM installed).
    - If updating from ESS 3000 container, then IBM Spectrum Scale, tools RPM, and OFED are updated.
    - POWER8 EMS update is not supported from the ESS 5000 container.
- POWER9
  - EMS is at ESS 6.0.1.0 or later.
  - Full EMS update is supported from ESS 5000 container only.
  - Do not update I/O or protocol nodes from the ESS 3000 container. Protocol node update is only supported from ESS 5000 container.

Scenario: Legacy ESS setup with ESS 3000 added

This scenario comprises a POWER8 EMS, one or more ESS PPC64LE building blocks, and optionally two or more POWER8 PPC64LE protocol nodes. This is in addition to one or more ESS 3000 nodes.

Upgrade the legacy ESS nodes

In this scenario, obtain the latest ESS code stack and the *ESS: Quick Deployment Guide* and fully upgrade the EMS, ESS building-blocks, and (if applicable) protocol nodes first.

- ESS code on IBM Fix Central
- ESS 5.3.6 Quick Deployment Guide
- Box folder containing deployment items

Make sure that you download and use the edition that matches your current environment (DAE or DME). Read the *ESS: Quick Deployment Guide* carefully and follow the upgrade flow, which will result in the EMS, I/O server nodes, and (if applicable) protocol nodes getting upgraded to ESS 5.3.6.x version.

Note: You must stop the ESS 3000 container before upgrading.

```
podman ps -a
podman stop ContainerName
```

Upgrade the ESS 3000 nodes

Obtain the latest ESS 3000 code and the *ESS 3000 Quick Deployment Guide*, and upgrade to the latest level.

Note: Do not run the EMS upgrade or any EMS related health checks from the ESS 3000 container. For example, do not run the following commands:

```
essrun ems1 update --offline
essrun ems1 healthcheck
```

- ESS 3000 code on IBM Fix Central
- ESS 3000 6.0.1 Quick Deployment Guide
- ESS 5.3.6 Protocols Quick Deployment Guide

Scenario: Standalone ESS 3000 setup

In this scenario, there is only the POWER8 EMS exclusively managing one or more ESS 3000 systems. If you have POWER8 protocol nodes, use the upgrade instructions in the earlier scenario.

Upgrade the EMS node

Upgrade the EMS node from the container, which only upgrades IBM Spectrum Scale.

Note: Do not run any EMS health checks from the container. Even though it might be specified to do so in the *ESS 3000 Quick Deployment Guide*.

Upgrade the EMS node as follows.

```
essrun -N ems1 update --offline
```

IBM Spectrum Scale is upgraded on the EMS upgrade to match the ESS 3000 version. The firmware, OFED, IPR levels, etc., remain at the same ESS version, which is acceptable.

At the end of the ESS 3000 upgrade:

- EMS remains at same ESS version as before except for IBM Spectrum Scale and associated ESS 3000 support RPMs
- ESS 3000 nodes are fully upgraded to the latest version

Appendix C. Converting an older EMS node for ESS 3000 compatibility

When an older EMS is used for ESS 3000, certain steps need to be done to address some of the field issues.

Default ESS 3000 EMS node (POWER8) prerequisites:

- 5147-21L EMS (PPC64LE)
- Red Hat Enterprise Linux 7.7 Server (PPC64LE)
- ESS 5.3.5 or later
- Python 3 installed
- Additional management network connection to `enP3p9s0f1`

Note: It is assumed that the EMS node is not being used to manage legacy nodes. If the EMS node is being used to manage legacy nodes, use the ESS 5.3.6.x documentation to upgrade the EMS node.

1. If GPFS is in use, shut down GPFS and the GUI.
2. If xCAT is installed but you do not have legacy ESS nodes, uninstall xCAT.

   ```
   yum -y remove xCAT
   ```

Upgrade the EMS node to Red Hat Enterprise Linux 7.7 Server (PPC64LE) as follows.

3. Clean up any existing repositories.

   ```
   cd /etc/yum.repos.d ; rm -f *repo ; yum clean all
   ```

4. Uninstall OFED.

   ```
   /sbin/ofed_uninstall.sh --force
   ```

5. Create the repository file.

   a) Obtain the Red Hat Enterprise Linux 7.7 full server ISO (PPC64LE) and place it on the EMS node.

   b) Mount the ISO file.

      ```
      mount -o loop ISOFile /mnt
      ```

   c) Create the `/etc/yum.repos.d/rh7.repo` file with the following contents.

      ```
      [RH7]
      name=DVD for Red Hat Enterprise Linux 7.7 Server
      mediaid=1359576196.686790
      metadata_expire=-1
      gpgcheck=1
      cost=500
      enabled=1
      baseurl=file:///mnt/disc/
      gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-release
      ```

   d) Run the following commands.

      ```
      yum clean all
      yum repolist enabled
      ```

6. Upgrade the OS.

   ```
   yum -y update
   ```

7. Reboot the node.

   ```
   systemctl reboot
   ```

8. Install Python3.

   ```
   mount -o loop ISOFile /mnt
   yum -y install python3
   ```

9. Set up Python3 alternatives.

   ```
   update-alternatives --install /usr/bin/python
   python /usr/bin/python3 1
   update-alternatives --config python
   python --version
   ```

Appendix D. Security-related settings in ESS 3000

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

- “Enabling security in ESS” on page 37
- “Enabling firewall in ESS” on page 37
- “Working with sudo user in an ESS Environment” on page 38
- “Using the central administration mode in an ESS environment” on page 41

Enabling security in ESS

Enabling security 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 **essrun** script.

By default, any node in an ESS environment has security disabled. This script can be run after the deployment of the EMS node or the I/O server nodes is complete.

By default, any node in an ESS environment has security disabled. When you enable security on the node, the following changes occur:

- OS hardening is enabled by disabling TCP timestamps and ICMP protocol in network packets on the node.
- The HTTPd server is disabled from running on the node.

  Note: All services that are using the HTTPd server might be affected when HTTPd is disabled.
- Strong ciphers, Macs, and KexAlgorithms are enabled on the node.
- SSH timeout is set to 300 seconds (5 minutes).
- Enable security on the EMS node by running the **security** sub-command with the enable option.

  ```
  # essrun -N ems1 security enable
  ```
- Enable security on the I/O server nodes by running the **security** sub-command with the enable option.

  ```
  # essrun -N ess_ppc64le security enable
  ```
- Disable security on the EMS node by running the **security** sub-command with the disable option.

  ```
  # essrun -N ems1 security disable
  ```
- Check the status of the security settings on a node as follows.

  ```
  # essrun -N ems1 security verify
  ```

Protocol node consideration: You can also use these steps to enable security on protocol nodes.

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 **essrun** command.

By default, any node in an ESS cluster has firewall disabled. You can run the **firewall** sub-command of the **essrun** 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.

  ```
  # essrun -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.

```
# essrun -N ems1 firewall verify
```

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

```
# essrun -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.

```
# essrun -N ess_x86_64 firewall verify
```

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

```
# essrun -N ems1 firewall disable
```

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

```
# essrun -N ess_x86_64 firewall disable
```

Protocol node consideration: Protocol node deployment is not supported with ESS 3000 6.0.1.x container.

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 **essrun** command.

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

- "Enabling sudo on Linux nodes" on page 38
- "Disabling sudo on Linux nodes" on page 39
- "Enabling sudo with GPFS cluster" on page 39
- "Disabling sudo with GPFS cluster" on page 40
- "I/O server nodes" on page 41
- "Protocol nodes" on page 41
- "Help text sudo sub-command" on page 41

Enabling sudo on Linux nodes

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

```
# essrun -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.

```
# essrun -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.

```
# essrun -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.

```
# essrun -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.

```
# essrun -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 6.0.1.x container.

Help text sudo sub-command

```
# /opt/ibm/ess/tools/bin/essrun sudo --help
usage: essrun 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 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, 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 **essrun** configures adminMode=central in an ESS 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 41
- “Disabling the central administration mode” on page 42
- “Help text admincentral sub-command” on page 43

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 nodes as physical EMS and container EMS share the same SSH public and private keys.

```
# essrun -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.

```
# essrun -N cems0 admincentral disable
```

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

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

positional arguments:
  {enable, disable}

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

Appendix E. How to set up chronyd (time server)

Note: The following time server setup documentation is for general reference. You can configure the time server as suitable for your environment. In the simplest example, the EMS host is used as the time server and the I/O nodes (or protocol nodes) are used as clients. Customers might want to have all nodes point to an external time server. Use online references for more detailed instructions for setting up Chrony.

Chrony is the preferred method of setting up a time server. NTP is considered deprecated. Chrony uses the NTP protocol.

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
```

Appendix F. ESS 5000 protocol node deployment by using the IBM Spectrum Scale installation toolkit

The following guidance is for adding a protocol node after storage deployment in an ESS 5000 environment.

Note: The following instructions for protocol node deployment by using the installation toolkit is just an example scenario. For detailed and latest information, see the following topics.

- Installing IBM Spectrum Scale on Linux nodes with the installation toolkit
- Configuring the CES and protocol configuration

Prerequisites

- During file system creation, adequate space is available for CES shared root file system. For more information, see "During file system creation, adequate space is available for CES shared root file system" on page 47
- ESS 5000 container has the protocol node management IP addresses defined. For more information, see "ESS 5000 container has the protocol node management IP addresses defined" on page 47.
- ESS 5000 container has the CES IP addresses defined. For more information, see "ESS 5000 container has the CES IP addresses defined" on page 48.

During file system creation, adequate space is available for CES shared root file system

In a default ESS setup, you can use the Ansible based file system task to create the recovery groups, vdisk sets, and file system. By default, during this task, 100% of the available space is attempted to be consumed. If you plan to include protocol nodes in your setup, you must leave enough free space for the required CES shared root file system. Use the **`--size`** flag to adjust the space consumed accordingly.

For example: **`essrun -G ess_ppc64le filesystem --suffix=-hs --size 80%`**

Running this command leaves approximately 20% space available for the CES shared root file system or additional vdisks. If you are in a mixed ESS 3000 and ESS 5000 environment, you might not use the **`essrun filesystem`** task due to more complex storage pool requirements. In that case, when using **`mmvdisk`**, make sure that you leave adequate space for the CES shared root file system. The CES shared root file system requires around 20 GB of space for operation.

ESS 5000 container has the protocol node management IP addresses defined

Before running the ESS 5000 container make sure to add the protocol node management IP addresses to `/etc/hosts`. These IP addresses are given to the SSR through the TDA process and they are already set. The customer needs to define host names and add the IP addresses to the EMS node `/etc/hosts` file before running the container.

You also need to define the high-speed IP address and host names. The IP addresses get set when running the Ansible network bonding task but the host names and IP addresses must be defined in `/etc/hosts` before the container starts. The high-speed host names must add a suffix of the management names. The IP addresses are user definable. Consult the network administrator for guidance.

For example:

```
# Protocol management IPs
192.168.45.23 prt1.localdomain prt1
192.168.45.24 prt2.localdomain prt2
# Protocol high-speed IPs
```

```
11.0.0.4 pr1-hs.localdomain prt1-hs
11.0.0.5 pr2-hs.localdomain prt2-hs
```

Note: `localdomain` is an example domain. The domain must be changed and also match that of the other nodes.

ESS 5000 container has the CES IP addresses defined

The final item that must be defined before starting the ESS 5000 container are the CES IP addresses. The following example shows the usage of two IP addresses per node over the high-speed network. Consult the IBM Spectrum Scale documentation for best practices.

```
11.0.0.100 prt_ces1.localdomain prt_ces1
11.0.0.101 prt_ces2.localdomain prt_ces2
11.0.0.102 prt_ces3.localdomain prt_ces3
11.0.0.103 prt_ces4.localdomain prt_ces4
```

Starting state in the example scenario

- ESS storage is deployed and configured.
- Adequate space (approximately 20 GB) is available for CES shared root file system.
- Protocol node required host names and IP addresses is defined on the EMS prior to starting the container.
- You are logged in from the ESS 5000 container.

Instructions for deploying protocol nodes in an ESS 5000 environment

Do the following steps from the ESS 5000 container.

1. Ping the management IP addresses of the protocol nodes.

   ```
   ping IPAdress1,...IPAddressN
   ```

   Each protocol node must respond to the ping test indicating they have an IP address set and it is on the same subnet as the container.
2. Run the config load task.

   ```
   essrun -N prt1,prt2 config load -p RootPassword
   ```

   If you have more than one node, you can specify them in a comma-separated list.
3. Create network bonds.

   Note: Make sure that the nodes are connected to the high-speed switch before doing this step.

   ```
   essrun -N prt1,prt2 network --suffix=-hs
   ```

4. Install the CES shared root file system.

   ```
   essrun -G ess_ppc64le filesystem --suffix=-hs --ces
   ```

5. Log out of the container and run the SSH setup on the EMS node.

   a. Press **Ctrl + p** then **Ctrl + q** to exit the container.

   b. Run the following commands for SSH setup on the EMS node.

   ```
   mkdir -p /root/pem_key
   cp /root/.ssh/id_rsa /root/pem_key/id_rsa
   ssh-keygen -p -N "" -m pem -f /root/pem_key/id_rsa (type yes after running this command)
   ./Spectrum_Scale_Data_Management-5.0.5.4-ppc64LE-Linux-install --silent
   cd /usr/lpp/mmfs/5.0.5.4/installer/
   ./spectrumscale setup -s EMSNodeHighSpeedIP -i /root/pem_key/id_rsa -st ess
   ```

6. On the EMS node, locate the installation package and run the installer. **./Spectrum_Scale_Data_Management-5.0.5.4-ppc64LE-Linux-install --silent**

   Note: Start `localrepo_AppStream` and `localrepo_BaseOs` in protocol nodes before starting the installation. For configuring the repositories, use the **essrun -G ces_ppc64le update --offline** command.

7. On the EMS node, do the following steps.

   a. Change the directory to the installer directory.

   ```
   cd /usr/lpp/mmfs/5.0.5.4/installer/
   ```

   b. List the current configuration.

   ```
   ./spectrumscale node list
   ```

   c. Populate the current cluster configuration in the cluster definition file.

   ```
   ./spectrumscale config populate -N EMSNodeHighSpeedName
   ```

   d. Designate the admin node.

   ```
   ./spectrumscale node add EMSNodeHighSpeedIP -a
   ```

   e. Add the protocol node.

   ```
   ./spectrumscale node add ProtocolNodeHighSpeedIP -p
   ```

   f. Run the installation precheck.

   ```
   ./spectrumscale install -pr
   ```

   g. Regenerate the SSH keys.

   ```
   ./spectrumscale setup -s EMSNodeHighSpeedIP -i /root/pem_key/id_rsa -st ess
   ```

   h. Set the port range.

   ```
   ./spectrumscale config gpfs --ephemeral_port_range 60000-61000
   ```

   i. Run the installation procedure on the node.

   ```
   ./spectrumscale install
   ```

   j. Configure the export IP pool.

   ```
   ./spectrumscale config protocols -e CESIP1,CESIP2,...
   ```

   k. Set the CES shared root file system.

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

   l. Enable protocols.

   ```
   ./spectrumscale enable smb nfs hdfs
   ```

   m. Confirm the settings.

   ```
   ./spectrumscale node list
   ```

   n. Run the deployment precheck.

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

   o. Run the deployment procedure on the node.

```
./spectrumscale deploy
```

Appendix G. Sample scenario: ESS 3000 and ESS 5000 mixed cluster and file system

Use these instructions for setting up ESS 3000 and ESS 5000 mixed cluster and file system.

The following high-level tasks need to be done for setting up ESS 3000 and ESS 5000 mixed cluster:

- Deploy an ESS 3000 system (including cluster, file system, GUI).
- Deploy an ESS 5000 system (adding to cluster, create recovery groups, etc.).
- Create the ESS 5000 vdisks and add to the existing file system.
- Create a policy file.
- Adjust sensors.
- Add ESS 5000 nodes to the GUI.

Note: These instructions contain summarized steps and references to documents that cover the items in more detail. The goal is to give an example scenario to help clients understand aspects of this procedure. At the end of this procedure, if you have POWER9 protocol nodes, for guidance in implementing them into your environment, see Appendix F, "ESS 5000 protocol node deployment by using the IBM Spectrum Scale installation toolkit," on page 47.

Prerequisites

- SSR has completed code 20 on both the ESS 3000 and ESS 5000 nodes (including EMS)

  SSR works on Power nodes and the EMS node first, then the ESS 3000 system.
- Public connection setup on C11-T3 (f3 connection on EMS)
- ESS 3000 and ESS 5000 nodes have been added to `/etc/hosts`
  - Low-speed names FQDNs , short names, and IP addresses
  - High-speed names FQDNs, short names, and IP addresses (add suffix of low-speed names)
- Host name and domain set on EMS
- Latest code for ESS 3000 and ESS 5000 stored in /home/deploy on EMS
- For information on how to deploy the ESS 3000 system, see ESS 3000 Quick Deployment Guide.
- For information on using the **mmvdisk** command, see mmvdisk in ESS documentation.

Summarized version of steps for deploying ESS 3000 building blocks

1. Extract the ESS 3000 installation package: **tar zxvf** *ESS3000InstallationPackage*
2. Accept the license and deploy the container: **sh ess3000_6.0.1.2_1202-03_dae.sh --text-only --start-container**

After logging in to the container, do the following steps:

1. Run the config load command.

   ```
   essrun -N ESS3000Node1,ESS3000Node2,EMSNode config load -p RootPassword
   ```

   Note: Use the low-speed names.
2. If required, update the EMS node.

   ```
   essrun -N EMSNode update --offline
   ```
3. Update the ESS 3000 nodes.

```
essrun -N ESS3000Node1,ESS3000Node2 update --offline
```

4. Create network bonds.

   ```
   essrun -N ESS3000Node1,ESS3000Node2,EMSNode network --suffix=Suffix
   ```

5. Create the cluster.

   ```
   essrun -G ESS3000NodeGroup cluster --suffix=Suffix
   ```

   Note: To obtain the group name, use **lsdef -t group**.

6. Add the EMS node to the cluster.

   ```
   essrun -N ESS3000Node1 cluster --suffix=Suffix --add-ems EMSNode
   ```

7. Create the file system.

   ```
   essrun -G ESS3000NodeGroup filesystem --suffix=Suffix
   ```

   Note: This command creates a combined data and metadata vdisk in the system pool. The file system name must be `fs3k`.

Type `exit` and press `Enter` to exit the container. Proceed with the instructions on how to setup the collector, sensors, and run the GUI wizard.

The current ESS 3000 container should be in the `stopped` state. To confirm, use the **podman ps -a** command.

If the ESS 3000 container is not in the `stopped` state, use the **podman stop** *ContainerName* command.

Summarized version of steps to add ESS 5000

1. Extract the ESS 5000 installation package: **tar zxvf** *ESS5000InstallationPackage*
2. Verify the integrity of the installation package: **sha256sum -c** *Extractedsha256sumFile*
3. Accept the license and deploy the container: **sh ess5000_6.0.1.2_1202-03_dae.sh --text-only --start-container**

After you have logged into the container, do the following steps:

1. Run the config load command.

   ```
   essrun -N ESS5000Node1,ESS5000Node2,ESS3000Node1,ESS3000Node2,EMSNode config load -p
   ibmesscluster
   ```

   Note: If you plan to add protocol nodes in the cluster, include them in the list of nodes that you are specifying in this command.

2. Update the nodes.

   ```
   essrun -N ESS5000Node1,ESS5000Node2 update --offline
   ```

3. Create network bonds.

   ```
   essrun -N ESS5000Node1,ESS5000Node2 network --suffix=Suffix
   ```

4. Add the ESS 5000 nodes to the existing cluster.

   a. SSH to one of the ESS 5000 I/O server nodes. For example:

      ```
      ssh ESS5000Node1
      ```

   b. Run this command.

      ```
      essaddnode -N ESS5000Node1-hs,ESS5000Node2-hs --cluster-node ESS3000Node-hs --nodetype
      ess5k --accept-license
      ```

Note:

- Use the high-speed names.
- If there is an error, you might need to log in to each ESS 5000 node and start GPFS.

```
mmbuildgpl
mmstartup
```

Type `exit` and press `Enter` to exit the container. Running these commands, takes you to the ESS 5000 node.

5. Create **`mmvdisk`** artifacts.

   a. Create the node class.

   ```
   mmvdisk nc create --node-class ess5k_ppc64le_mmvdisk -N
   ListOfESS5000Nodes_highspeedsuffix
   ```

   b. Configure the node class.

   ```
   mmvdisk server configure --nc ess5k_ppc64le_mmvdisk --recycle one
   ```

   c. Create recovery groups.

   ```
   mmvdisk rg create --rg ess5k_rg1,ess5k_rg2 --nc ess5k_ppc64le_mmvdisk
   ```

   d. Define vdiskset.

   ```
   mmvdisk vs define --vs vs_fs5k_1 --rg ess5k_rg1,ess5k_rg2 code 8+2p --bs 16M --ss 80% --
   nsd-usage dataOnly --sp data
   ```

   e. Create vdiskset.

   ```
   mmvdisk vs create --vs vs_fs5k_1
   ```

   f. Add vdiskset to the file system.

   ```
   mmvdisk fs add --file-system fs3k --vdisk-set vs_fs5k_1
   ```

   g. Add the policy file.

   Define your policy file. This can be used to move data from the system pool to the data pool when thresholds hit. For more information, see Overview of policies.

   You can also use the GUI to define policies. For more information, see Creating and applying ILM policy by using GUI.

   The following example rule ingests the writes on the ESS 3000 and moves the data to ESS 5000 when it reaches 75% capacity on the ESS 3000:

   - Add callback for automatic movement of data between pools:

   ```
   mmaddcallback MIGRATION --command /usr/lpp/mmfs/bin/mmstartpolicy --event
   lowDiskSpace,noDiskSpace --parms "%eventName %fsName"
   ```

   - Write the policy into a file with the following content:

   ```
   RULE 'clean_system' MIGRATE FROM POOL 'system' THRESHOLD(75,25) WEIGHT(KB_ALLOCATED) TO
   POOL 'data'
   ```

   Note: You need to understand the implications of this rule before applying it in your system. When capacity on ESS 3000 reaches 75%, it migrates files (larger ones first) out of the system pool to the data pool until the capacity reaches 25%.

   h. On the EMS node, run the following command.

   ```
   mmaddcompspec default --replace
   ```

At this point, add the ESS 5000 nodes to the `pmsensors` list and use the **Edit rack components** option in the GUI to slot the new nodes into the frame.

If you want to add protocol nodes, see Appendix F, “ESS 5000 protocol node deployment by using the IBM Spectrum Scale installation toolkit,” on page 47.

Appendix H. ConnectX-5 VPI support on ESS 3000

Check and enable adapter port configurations on the VPI adapter as follows.

The MT4121 adapter (AJP1) allows users to configure the VPI card ports as suitable for your environment. You can choose one of the following options for each adapter (2 ports):

- Have both ports IB/IB
- Have one port IB one port Ethernet
- Have both ports Ethernet

If any port is changed, the node must be rebooted for the changes to take effect.

The following options are added to **`essgennetworks`** to support VPI.

--query

Queries the port type of the Mellanox interface.

--devices *Devices*

Name of the Mellanox device name. Specifying `all` queries all devices attached to node. Provide comma-separated device names to query mode rather than one device at a given time.

--change {`InfiniBand,Ethernet`}

Changes the Mellanox port type to InfiniBand or Ethernet and vice versa.

--port {`P1,P2`}

Specifies the port number of the Mellanox VPI card.

The following example shows the usage of the **`essgennetworks`** command to check and enable adapter port configurations on the VPI adapter.

1. Query the port type of all attached devices.

```
# essgennetworks -N localhost --query --devices all
2020-09-23T04:24:03.397420 [INFO] Starting network generation
2020-09-23T04:24:03.579361 [INFO] nodelist:  localhost
[ERROR] Mellanox Software Tools services are not running.
        Make sure Mellanox Software Tools running configuring VPI adapters.
        Make sure you must start Mellanox Software Tools using "/bin/mst start"
        command before starting the configuration of the VPI adapters.
```

2. Start Mellanox Software Tools (MST).

```
# /bin/mst start
Starting MST (Mellanox Software Tools) driver set
Loading MST PCI module - Success
[warn] mst_pciconf is already loaded, skipping
Create devices
Unloading MST PCI module (unused) - Success
```

3. Query the port type of all attached devices again.

```
# essgennetworks -N localhost --query --devices all
2020-09-23T04:24:18.083995 [INFO] Starting network generation
2020-09-23T04:24:18.268935 [INFO] nodelist:  localhost
[INFO} Device          /dev/mst/mt4121_pciconf1 link type currently configured at system.
[INFO] Port 1 is set to InfiniBand
[INFO] Port 2 is set to InfiniBand
[INFO} Device          /dev/mst/mt4121_pciconf0 link type currently configured at system.
[INFO] Port 1 is set to InfiniBand
[INFO] Port 2 is set to InfiniBand
```

4. Convert the P1 port of the device listed in the preceding command to Ethernet from InfiniBand.

```
# essgennetworks -N localhost --change Ethernet --devices /dev/mst/mt4121_pciconf1 --port P1
2020-09-23T03:45:52.322096 [INFO] Starting network generation
2020-09-23T03:45:52.510535 [INFO] nodelist:  localhost
```

```
[INFO} Changing /dev/mst/mt4121_pciconf1 Port P1 link type to Ethernet
[INFO] Successfully changes the Port type to Ethernet for Port P1
```

5. Reboot they node and query the port type of all attached devices again.

```
 # essgennetworks -N localhost --query --devices all
2020-09-23T04:05:55.774019 [INFO] Starting network generation
2020-09-23T04:05:55.960088 [INFO] nodelist:  localhost
[INFO} Device          /dev/mst/mt4121_pciconf1 link type currently configured at system.
[INFO] Port 1 is set to Ethernet
[INFO] Port 2 is set to InfiniBand
[INFO} Device          /dev/mst/mt4121_pciconf0 link type currently configured at system.
[INFO] Port 1 is set to InfiniBand
[INFO] Port 2 is set to InfiniBand
```

6. Verify that the port type of the P1 port is changed to Ethernet.

```
# mlxconfig -d  /dev/mst/mt4121_pciconf1 query | grep -i link_type
         LINK_TYPE_P1                        ETH(2)
         LINK_TYPE_P2                        IB(1)
```

Appendix I. Client node tuning recommendations

IBM Spectrum Scale node configuration is optimized for running IBM Spectrum Scale RAID functions.

ESS cluster node configuration is optimized for running IBM Spectrum Scale RAID functions. Protocols, other gateways, or any other non-ESS services must not be run on ESS management server nodes or I/O server nodes. In a cluster with high IO load, avoid using ESS nodes as cluster manager or filesystem manager. For optimal performance the NSD client nodes accessing ESS nodes should be properly configured. ESS ships with `gssClientConfig.sh script` located in `/usr/lpp/mmfs/samples/gss/` directory. This script can be used to configure the client as follows:

```
/usr/lpp/mmfs/samples/gss/gssClientConfig.sh <Comma Separated list of
client nodes or nodeclass>
```

You can run the following to see configuration parameter settings without setting them:

```
/usr/lpp/mmfs/samples/gss/gssClientConfig.sh -D
```

After running this script, restart GPFS on the affected nodes for the optimized configuration settings to take effect.

Important: Do not run **`gssClientConfig.sh`** unless you fully understand the impact of each setting on the customer environment. Make use of the `-D` option to decide if all or some of the settings might be applied. Then, individually update each client node settings as required.

Accessibility features for the system

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

ESS 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 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
See *Elastic Storage System (ESS)*.

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 provides 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 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 node that hosts the ESS 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 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 ESS 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 ESS 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 ESS 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 ESS, 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

SC28-3157-02