IBM Elastic Storage System 5000
Version 6.0.1.2

Quick Deployment Guide

SC28-3134-02

Note

Before using this information and the product it supports, read the information in "Notices" on page 41.

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 ix. 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) 5000 documentation consists of the following information units.

Information unit	Type of information	Intended users
Hardware Guide	This unit provides ESS 5000 information including system overview, installing, and troubleshooting.	System administrators and IBM support team
Quick Deployment Guide	This unit provides ESS 5000 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
Model 092 storage enclosures	This unit provides information including initial hardware installation and setup, and removal and installation of field-replaceable units (FRUs), customer-replaceable units (CRUs) for ESS 5000 Expansion – Model 092, 5147-092.	System administrators and IBM support team
Model 106 storage enclosures	This unit provides information including hardware installation and maintenance for ESS 5000 Expansion – Model 106.	System administrators and IBM support team
Problem Determination Guide	This unit provides ESS 5000 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.

Table 1. Conventions (continued)

Convention	Usage
|	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 5000 Software deployment preparation

Install the ESS software package and deploy the storage servers by using the following information. The goal is to create a cluster allowing client or protocol nodes to access the file systems.

ESS 5000 software stack

- Operating system: Red Hat® Enterprise Linux® 8.1 PPC64LE
- Container version: Red Hat Enterprise Linux 7.7 UBI
- IBM Spectrum Scale: 5.0.5 PTF4
- Kernel: 4.18.0-193.29.1.el8_2
- Systemd: 239-18.el8_1.5
- Network manager: 1.20.0-3.el8
- ndctl version: ndctl-65-1.el8
- OPAL PRD: opal-prd-3000.0-1.el8
- IPR level: 19512900
- Host adapter driver: 34.00.00.00
- Host adapter firmware: 16.00.11.00
- Enclosure firmware:
  - 5U92: E558
  - 4U106: 5266
- POWER9™ firmware: FW941.00 (VL941_035)
- Firmware RPM: gpfs.ess.firmware-6.0.0-6
- ESA: esagent.pLinux-4.5.5-0
- Ansible®: ansible-2.9.9-1.el7ae
- Podman: podman-1.6.4-1
- xCAT: 2.15.1
- OFED version: MLNX_OFED_LINUX-4.9-0.1.7.3

  OFED firmware versions:
  - MT27500 = 10.16.1200
  - MT4099 = 2.42.5000
  - MT26448 = 2.9.1326
  - MT4103 = 2.42.5000
  - MT4113 = 10.16.1200
  - MT4115 = 12.27.2008
  - MT4117 = 14.27.2008
  - MT4119 = 16.27.6008
  - MT4120 = 16.27.6008
  - MT4121 = 16.27.6008
  - MT4122 = 16.27.6008

Prerequisites

- This document (ESS 5000 Software Quick Deployment Guide)
- SSR completes physical hardware installation and code 20.

  SSR uses Worldwide Customized Installation Instructions (WCII) for racking, cabling, and disk placement information.

  SSR uses ESS 5000 Hardware Guide for hardware checkout and setting IP addresses.
- Worksheet notes from the SSR
- ESS 5000 tgz downloaded to the EMS node from Fix Central (If newer version is available).

  Data Access Edition or Data Management Edition: Must match the order; if the edition does not match your order, open a ticket with the IBM Service.
- High-speed switch and cables have been run and configured.
- Low-speed host names are ready to be defined based on the IP addresses the SSR have configured.
- High-speed host names (suffix of low speed) and IP addresses are ready to be defined.
- Container host name and IP address are ready to be defined into `/etc/hosts`.
- Host and domain name (FQDN) are defined in the `/etc/hosts` file.

What is in the /home/deploy directory on the EMS node

- ESS 5000 tgz used in manufacturing (may not be the latest)
- ESS 3000 tgz used in manufacturing (may not be the latest)
- Red Hat Enterprise Linux 8.1 PPC64LE ISO (used for future EMS upgrades)
  - `rhel-8.1-server-ppc64le.iso`

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
   ```

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

Starting system state

SSR will assure that the following setups are done:

1. Each node's management IPs are set up (typically 192.168.x.x/24). By default, these are blank.
2. Each node's HMC1 IPs are set up (typically 10.0.0.x/24). By default, these are blank.
3. The EMS FSP interface (C11-T2) has an IP address set on the same HMC1 subnet, typically 10.0.0.x/24.
4. Each node and storage enclosure have clean hardware.
5. All nodes ping over the FSP and management interfaces.
   - On EMS, this means pinging from T2 to the other node’s HMC1 interfaces
   - EMS can ping the management interface of the IO nodes or POWER9 protocol nodes (T1 interfaces)
6. Consult the ESS 5000 Fix Central and determine if there is a newer version available. If so, download it to the EMS and place in `/home/deploy`. Remove any `.tgz` files that might have been there from manufacturing.
7. SSR has passed notes on to the installation worksheet. This might be helpful information that was encountered during code 20 of the hardware.

Other notes

- The following should be the case before starting a new installation (tasks done by manufacturing and the SSR):
  - SSR has ensured all hardware is clean, and IP addresses are set and pinging over the proper networks (through the code 20 operation).
  - `/etc/hosts` is blank

  - The ESS 5000 tgz file (for the correct edition) is in the `/home/deploy` directory. (If upgrade is needed, download from Fix Central and replace.)
  - Network bridges are cleared.
  - Images and containers are removed.
  - SSH keys are cleaned up and regenerated.
  - All firmware, operating system, RAID array (10) that are installed are at correct levels at the time of shipping for the latest version available of ESS 5000 (manufacturing task).
- Customer must make sure the high-speed connections are cabled and the switch is ready before starting.
- All node names and IP addresses in this document are examples.
- Changed root password should be same on each node, if possible. The default password is `ibmesscluster`. It is recommended to change the password after deployment is completed.
- Each server's IPMI and ASMI passwords are set to the server serial number. Consider changing these passwords when the deployment is complete.

Chapter 2. ESS 5000 Common installation instructions

The following common instructions need to be run for a new installation or an upgrade of an ESS 5000 system.

Note: If you have protocol nodes, add them to the commands provided in these instructions. The default `/etc/hosts` file has host names `prt1` and `prt2` for protocol nodes. You might have more than two protocol nodes.

1. Log in to the EMS node by using the management IP (set up by SSR by using the provided worksheet). The default password is `ibmesscluster`.
2. **Set up a campus or a public connection (interface enP1p8s0f2).** Connect an Ethernet cable to C11-T3 on the EMS node 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 (if changing is mandatory after SSR is finished). That includes renaming the interface, setting IP, or any other interaction with those interfaces.

   You can use the **`nmtui`** command to set the IP address of the campus interface. For more information, see Configuring IP networking with nmtui.
3. Complete the `/etc/hosts` file on the EMS node. 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 pr1-hs.localdomain.local prt1-hs
10.0.11.5 pr2-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, ensure 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.

You can set up the EMS FQDN manually or wait until prompted when the ESS deployment binary is started. At that time, the scripts confirms the FQDN and provides the user a chance to make changes.

- If you are planning to set up an ESS 3000 system with the ESS 5000 EMS node, add the ESS 3000 host names to `/etc/hosts` by using the same structure (low-speed (management) and high-speed (cluster) IP addresses, FQDNs, and short names).
- 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.

4. Clean up the old containers and images.

   Note: Typically, this is applicable only for upgrades.

   a. List the containers.

   ```
   podman ps -a
   ```

   b. Stop and remove the containers.

   ```
   podman stop ContainerName
   podman rm ContainerName -f
   ```

   c. List the images.

   ```
   podman images
   ```

   d. Remove the images.

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

   e. [Recommended] Remove container bridges as follows.

   1) List the currently configured bridges.

   ```
   nmcli c
   ```

   2) Clean up any existing bridges before the new container is set up. The bridge names must be `mgmt_bridge` and `fsp_bridge`.

   ```
   nmcli c del BridgeName
   ```

5. Extract the installation package.

   Note: Ensure that you check the version that is installed from manufacturing (SSR worksheet). If there is a newer version available on Fix Central, replace the existing image in `/home/deploy` with the new image and then remove the old tgz file before doing this step.

   ```
   cd /home/deploy

   tar zxvf ess5000_6.0.1.2_1204-02_dme.tgz

   ess5000_6.0.1.2_1204-02_dme.sh
   ess5000_6.0.1.2_1204-02_dme.sh.sha256
   ```

6. Accept the license and install the accepted image.

   ```
   ./ess5000_6.0.1.2_1204-02_dme.sh --text-only --start-container
   ```

   Note:

- The `--install-image` flag will be deprecated soon. Stop and remove any existing container.
- The `--text-only` flag is used to extract the contents of the tgz file after accepting the license agreement. Immediately afterward, the `--start-container` flag is used to do the following steps automatically.
  - Run **essmkyml** that prompts the user to:
    - Confirm EMS FQDN and change it.
    - Provide the container short name.
    - Provide a free IP address on the FSP subnet for the container FSP connection.

Press 1 to accept the license after reading the agreement.

Example of contents of the extracted installation package:

```
ess5000_6.0.1.2_1204-02_dme.dir/
ess5000_6.0.1.2_1204-02_dme.dir/ess5000_6.0.1.2_1204-02_dme.tar
ess5000_6.0.1.2_1204-02_dme.dir/ess5000_6.0.1.2_1204-02_dme_binaries.iso
ess5000_6.0.1.2_1204-02_dme.dir/rhel-8.1-server-ppc64le.iso
ess5000_6.0.1.2_1204-02_dme.dir/podman_rh8.tgz
ess5000_6.0.1.2_1204-02_dme.dir/essmgr
ess5000_6.0.1.2_1204-02_dme.dir/essmgr.yml
ess5000_6.0.1.2_1204-02_dme.dir/Release_note.ess5000_6.0.1.2_1204-02_dme.txt
ess5000_6.0.1.2_1204-02_dme.dir/classes/
ess5000_6.0.1.2_1204-02_dme.dir/classes/essmgr_yml.py
ess5000_6.0.1.2_1204-02_dme.dir/classes/__init__.py
ess5000_6.0.1.2_1204-02_dme.dir/essmkyml
```

For this step, you must provide these inputs:

- Container name (must be in `/etc/hosts` or be resolvable by using DNS)
- Container FSP IP address (must be on the same network block that is set on C11-T2)
- Confirmation of the EMS FQDN (must match what is set for the management IP in `/etc/hosts`). If this value needs to be changed or set, **essmkyml** helps with that task.
- 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 on the `/etc/hosts` file.

  ```
  Is the current EMS FQDN c145f05zems06.gpfs.net correct (y/n):
  ```

- Remember not to add the DNS domain `localdomain` to the input:

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

- Remember that the IP address must belong to the 10.0.0.x/24 network block (It is assumed that the recommended FSP network was used):

  ```
  Please type the FSP IP of the container [10.0.0.5]: 10.0.0.80
  ```

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.

At this point, if all checks are successful, the image is loaded and container is started. Example:

```
ESS 5000 CONTAINER root@cems0:/ #
```

7. Run the config load command.

   ```
   essrun -N essio1,essio2,ems1 config load -p ibmesscluster
   ```

   Note:

- Use the low-speed management host names. Specify the root password with `-p`.
- The password (-p) is the root password of the node. By default, it is `ibmesscluster`. Consider changing the root password after deployment is complete.
- This command attempts to connect to each node's FSP interface through IPMI by using the default password (serial number). If the password has changed, you are prompted to enter the new password.

  To determine the serial number, do the following:

  a. Log in to the node by using the management IP address.

  b. Issue this command: **`cat /proc/device-tree/system-id`**

After this command is run, you can use `-G` for future **essrun** steps (For example, `-G ess_ppc64le`).

Instructions if the latest ESS 5000 package version is the same as the one from manufacturing

If the ESS version on the system is already at the latest version shipped from manufacturing, proceed directly to the network bond creation step. If the version is different, use the instructions in the next section.

1. Create network bonds.

   ```
   essrun -G ess_ppc64le network --suffix=-hs
   essrun -N ems1 network --suffix=-hs
   ```

2. Run the network test.

   This test uses **nsdperf** to determine if the newly created network bonds are healthy.

   SSH from the container to an I/O node or the EMS node.

   ```
   ssh essio1
   ESSENV=TEST essnettest -N essio1,essio2 --suffix=-hs
   ```

   This command performs the test (with optional RDMA test after if Infiniband). Ensure that there are no errors in the output indicating dropped packets have exceeded thresholds. When completed, type `exit` to return back to the container.

3. Create the cluster.

   ```
   essrun -G ess_ppc64le cluster --suffix=-hs
   ```

4. Add the EMS node to the cluster.

   ```
   essrun -N essio1 cluster --add-ems ems1 --suffix=-hs
   ```

5. Create the file system.

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

   Note:

   - By default, this command attempts to use all the available space. If you need to create multiple file systems or a CES shared root file system for protocol nodes, consider using less space. For example:

     ```
     essrun -G ess_ppc64le filesystem --suffix=-hs --size 80%
     ```

   - This step creates combined metadata + data vdisk sets by using a default RAID code and block size. You can use additional flags to customize or use the **mmvdisk** command directly for advanced configurations.

Instructions if the latest ESS 5000 package version is not the same as the one from manufacturing

Note: Use this procedure if the ESS version from manufacturing is not the latest.

1. Update the EMS node.

   Important: [Online update only] Ensure that all ESS 5000 nodes are active by first running this command from one of the cluster nodes: **mmgetstate -N ess5k_ppc64le**. If any nodes are not active, quit the upgrade procedure and resolve this issue before proceeding with the upgrade.

   ```
   essrun -N ems1 update --offline

   Please enter 'accept' indicating that you want to update the following list of nodes: ems1
   >>> accept
   ```

   Note: If the kernel is changed, you are prompted to leave the container, reboot the EMS node, restart the container, and run this command again.

   For example:

   ```
   essrun -N ems1 --offline
   Exit
   systemctl reboot
   ```

   Navigate back to ESS 6.0.1.2 extracted directory and run the following commands:

   ```
   ./essmgr -r
   essrun -N ems1 --offline
   ```

2. Update the IO nodes.

   ```
   essrun -G ess_ppc64le update --offline
   ```

3. Create network bonds.

   ```
   essrun -G ess_ppc64le network --suffix=-hs
   essrun -N ems1 network --suffix=-hs
   ```

4. Run the network test.

   This test uses **nsdperf** to determine if the newly created network bonds are healthy.

   SSH from the container to an I/O node or the EMS node.

   ```
   ssh essio1
   ESSENV=TEST essnettest -N essio1,essio2 --suffix=-hs
   ```

   This command performs the test with an optional RDMA test afterward if there is Infiniband. Ensure that there are no errors in the output indicating dropped packets have exceeded thresholds. When completed, type `exit` to return back to the container.

5. Create the cluster.

   ```
   essrun -G ess_ppc64le cluster --suffix=-hs
   ```

6. Add the EMS node to the cluster.

   ```
   essrun -N essio1 cluster --add-ems ems1 --suffix=-hs
   ```

7. Create the file system.

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

   Note:

   - By default, this command attempts to use all the available space. If you need to create multiple file systems or a CES shared root file system for protocol nodes, consider using less space. For example:

```
essrun -G ess_ppc64le filesystem --suffix=-hs --size 80%
```

- This step creates combined metadata + data vdisk sets by using a default RAID code and block size. You can use additional flags to customize or use the **mmvdisk** command directly for advanced configurations.

Final setup instructions

1. From the EMS node (outside of the container), configure and start the performance monitoring collector.

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

2. From the EMS node (outside of the container), configure and start the performance monitoring sensors.

   ```
   mmchnode --perfmon -N ems1-hs,essio1-hs,essio2-hs
   ```

3. Capacity and fileset quota monitoring is not enabled in the GUI by default. You must correctly update the values and restrict collection to the EMS node only.

   a. To modify the GPFS Disk Capacity collection interval, run the following command.

      ```
      mmperfmon config update GPFSDiskCap.restrict=EMSNodeName
               GPFSDiskCap.period=PeriodInSeconds
      ```

      The recommended period is 86400 so that the collection is done once per day.

   b. To restrict GPFS Fileset Quota to run on the management server node only, run the following command.

      ```
      mmperfmon config update GPFSFilesetQuota.period=600 GPFSFilesetQuota.restrict=EMSNodeName
      ```

      Here the *EMSNodeName* must be the name shown in the **mmlscluster** output.

      Note: To enable quota, the filesystem quota checking must be enabled. Refer **mmchfs -Q** and **mmcheckquota** commands in *IBM Spectrum Scale: Command and Programming Reference*.

4. Verify that the values are set correctly in the performance monitoring configuration by running the **mmperfmon config show** command on the EMS node. Ensure that `GPFSDiskCap.period` is properly set, and `GPFSFilesetQuota` and `GPFSDiskCap` are both restricted to the EMS only.

   Note: If you are moving from manual configuration to auto configuration then all sensors are set to default. Make the necessary changes using the **mmperfmon** command to customize your environment accordingly. For information on how to configure various sensors using **mmperfmon**, see Manually installing IBM Spectrum Scale GUI.

5. Start the performance collector on the EMS node.

   ```
   systemctl start pmcollector
   ```

6. 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 public or campus IP address with `https` and walk through the System Setup wizard instructions.

7. Log in to each node and run the following command.

   ```
   essinstallcheck -N localhost
   ```

Doing this step verifies that all software and cluster versions are up-to-date.

8. From the EMS node, outside of the container, run the following final health check commands to verify your system health.

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

9. Set the time zone and set up Chrony.

   Before getting started, ensure that Chrony and time zone are set correctly on the EMS and I/O nodes. Refer to Appendix B, “How to set up chronyd (time server),” on page 27 to perform these tasks before proceeding.

10. 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

11. Refer to Appendix E, “Client node tuning recommendations,” on page 37.

Chapter 3. ESS 5000 upgrade instructions

Warning: You must have a clean and healthy system before starting any ESS upgrade (online or offline). At least, the following commands must run free of errors when run on any node outside of container:

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

You can also run the **essrun healthcheck** command instead.

```
essrun -G NodeGroup healthcheck
```

ESS 5000 I/O nodes upgrade can be done by using one of the following methods. In both methods, drive, host-adapter, and enclosure firmware are upgraded if the cluster is created and needed.

- "I/O nodes online upgrade" on page 13
- "I/O nodes offline upgrade" on page 14

For ESS 5000 EMS and protocol nodes, upgrade can be done by using "EMS and protocol nodes offline upgrade" on page 15. In this method, kernel, MOFED, and IBM Spectrum Scale (EMS only) are upgraded.

I/O nodes online upgrade

Assumptions:

- The cluster is created with EMS, one or more ESS 5000 nodes, and optionally one or more ESS building blocks or protocol nodes.
- The file system is built and recovery groups are active and healthy.
- GPFS is active on all ESS 5000 nodes and quorum is achieved.
- New container is installed that will update the code on the EMS and I/O nodes.
- GUI and collector services are stopped on the EMS before starting the upgrade.

Before starting the online upgrade, make sure that all ESS 5000 nodes are active by running the following command from one of the cluster nodes:

```
mmgetstate -N ess5k_ppc64le
```

Use the following online upgrade steps.

1. Complete the steps in Chapter 2, "ESS 5000 Common installation instructions," on page 5. These steps include obtaining the new ESS 5000 code, and installing and running the new container. After doing these steps, you should be in the new container (ESS 5000 version 6.0.1.2).
2. Run the configuration load.

   ```
   essrun -N essio1,essio2,ems1 config load -p ibmesscluster
   ```

3. Update the POWER9 EMS node by using "EMS and protocol nodes offline upgrade" on page 15.
4. Update the ESS 5000 nodes by using one of the following commands.

   Important: For doing an online upgrade, recovery groups must be correctly created in both I/O nodes from the ESS 5000 cluster. Quorum is checked early in the process. If no quorum is achieved, the upgrade stops.

   - Update by using the group of all configured ESS 5000 nodes.

     ```
     essrun -G ess_ppc64le update
     ```

- Update by using the individual nodes.

```
essrun -N essio1,essio2 update
```

5. Run installation check on each node.

```
essinstallcheck
```

6. Change the **autoload** parameter to enable GPFS to automatically start on all nodes.

```
mmchconfig autoload=yes
```

7. Start the performance monitoring sensors on each node.

```
systemctl start pmsensors
```

8. Run health checks by using one of the following methods.
   - Run health checks by using the container.

```
essrun -N essio1,essio2 healthcheck
```

   - Run manual health checks on each node.

```
gnrhealthcheck
mmhealth node show
```

I/O nodes offline upgrade

Assumptions:

- 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 2, “ESS 5000 Common installation instructions,” on page 5. These steps include obtaining the new ESS 5000 code, and installing and running the new container. After doing these steps, you should be in the new container (ESS 5000 version 6.0.1.2).
2. Run the configuration load.

```
essrun -N essio1,essio2,ems1 config load -p ibmesscluster
```

3. Update the ESS 5000 nodes by using one the following commands.

   Important: For doing an offline update, GPFS must be down in the ESS 5000 cluster. The GPFS status is checked. If it is up on a given node, you are asked if it is OK to shut it down.

   - Update by using the group of all configured ESS 5000 nodes.

```
essrun -G ess5k_ppc64le update --offline
```

   - Update by using the individual nodes.

```
essrun -N essio1,essio2 update --offline
```

   - Update one node at a time.

```
essrun -N essio1 update --offline
```

4. Run installation check on each node.

```
essinstallcheck
```

5. Start GPFS and set the **autoload** parameter to enable GPFS to automatically start on all nodes.

```
mmchconfig autoload=yes
mmstartup
```

6. Start the performance monitoring sensors on each node.

```
systemctl start pmsensors
```

7. Run health checks by using one of the following methods.
   - Run health checks by using the container.

     ```
     essrun -N essio1,essio2 healthcheck
     ```

   - Run manual health checks on each node.

     ```
     gnrhealthcheck
     mmhealth node show
     ```

EMS and protocol nodes offline upgrade

Assumptions:

- 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.

Note: This step updates the EMS node to ESS 6.0.1.2. This includes items such as:

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

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

Use the following offline upgrade steps.

1. Complete the steps in Chapter 2, “ESS 5000 Common installation instructions,” on page 5. These steps include obtaining the new ESS 5000 code, and installing and running the new container. After doing these steps, you should be in the new container (ESS 5000 version 6.0.1.2).
2. Run the configuration load.

   ```
   essrun -N essio1,essio2,ems1,prt01,prt02 config load -p ibmesscluster
   ```

3. Update the ESS 5000 nodes by using the following commands.

   Important: For doing an offline update, GPFS must be down in the ESS 5000 cluster. The GPFS status is checked. If it is up on a given node, you are asked if it is OK to shut it down.

   a. Update the EMS node.

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

   b. Update the protocol nodes.

      ```
      essrun -N prt01,prt02 update --offline
      ```

4. If kernel is updated on the ESS 5000 EMS node, you are promoted to exit the container and reboot.

```
Seems that kernel has changed. This will require a reboot
Please exit container and reboot ems1
Restart container (./essmgr -r) once ems1 is back and run update again.
```

   After the reboot and restarting the container, run the EMS node update again.

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

5. Run installation check on each node.

```
essinstallcheck
```

6. Start GPFS and set the **autoload** parameter to enable GPFS to automatically start on all nodes.

```
mmchconfig autoload=yes
mmstartup
```

7. Start the performance monitoring sensors on each node.

```
systemctl start pmsensors
```

8. **[EMS node only]** Exit the container and then restart GUI and collector services on the EMS node.

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

9. Run health checks by using one of the following methods.
   - Run health checks by using the container.

```
essrun -N ems1,prt01,prt02  healthcheck
```

   - Run manual health checks on each node.

```
mmhealth node show
```

Chapter 4. ESS 5000 Re-installation and cleanup instructions

If you need to redeploy the ESS 5000 nodes cleanly from the container, perform the following steps. For cleanup instructions, see “Cleanup procedure” on page 18.

1. Launch the container.

   Note: If you are doing a first time container installation, see Chapter 2, “ESS 5000 Common installation instructions,” on page 5.

   ```
   podman attach  ContainerName

   OR

   ./essmgr -r
   ```

   Confirm that the domain is set correctly.

   ```
   tabdump site | grep -I domain
   ```

   If not, you might need to exit and remove container and then trey again after fixing `/etc/hosts` and `essmgr.yml`.

2. Define the nodes.

   ```
   mkdef -t node essio2 groups=all,ess_ppc64le installnic=mac mac=b0:26:28:e6:63:e4
   ip=10.0.0.3 netboot=petitboot arch=ppc64le bmc=172.16.0.117 bmcpassword=abc123
   mgt=ipmi cons=ipmi addkcmdline="modprobe.blacklist=mpt3sas R::log_buf_len=4M"
   chain="runcmd=fspipsetup.sh"

   mkdef -t node essio1 groups=all,ess_ppc64le installnic=mac mac=b0:26:28:e5:81:88
   ip=10.0.0.2 netboot=petitboot arch=ppc64le bmc=172.16.0.118 bmcpassword=abc123
   mgt=ipmi cons=ipmi addkcmdline="modprobe.blacklist=mpt3sas R::log_buf_len=4M"
   chain="runcmd=fspipsetup.sh"
   ```

   Do a ping test to the BMC IP addresses. If you cannot ping each BMC IP address, you might have issues with the container bridges. Fix the problem and then try again.

3. Run nodeset on the `ess_ppc64le` group.

   ```
   nodeset ess_ppc64le osimage=rhels8.1-ppc64le-install

   essio1: install rhels8.1-ppc64le-ess
   essio2: install rhels8.1-ppc64le-ess
   ```

4. Run the **makedhcp** command.

   ```
   makedhcp -n
   ```

5. Confirm that DHCP is working.

   ```
   makedhcp -q ess_ppc64le

   essio1: ip-address = 10.10.0.2, hardware-address = b0:26:28:e5:81:88
   essio2: ip-address = 10.10.0.3, hardware-address = b0:26:28:e6:63:e4
   ```

6. Power off the nodes.

   ```
   rpower ess_ppc64le off

   essio1: off
   essio2: off
   ```

7. Set network boot.

```
rsetboot ess_ppc64le net

essio2: Network
essio1: Network
```

8. Power on the nodes.

```
rpower ess_ppc64le on

essio2: on
essio1: on
```

9. Install by using **ipmitool**.

```
Example (pointing to BMC IP):

ipmitool -I lanplus -H 172.16.0.117 -P abc123 sol activate
ipmitool -I lanplus -H 172.16.0.118 -P abc123 sol activate
```

10. After the nodes are deployed, set the root password.

```
You are required to change your password immediately (administrator enforced)
Last login: Thu Apr 23 22:15:27 2020 from 10.10.0.70
WARNING: Your password has expired.
You must change your password now and login again!
Changing password for user root.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
```

11. On each node, view the last parts of the `xcat.log` to confirm when the post scripts have completed.

```
tail -f /var/log/xcat/xcat.log
```

12. Once deployment is completed on all nodes, run **essinstallcheck** to verify that the installation is successful. If everything is clean, proceed with the additional deployment steps.

Cleanup procedure

Run these steps if you do not already have a clean EMS node. Typically, this is done prior to upgrading an existing environment.

1. List the containers.

```
podman ps -a
```

2. Stop and remove the containers.

```
podman stop ContainerName
podman rm ContainerName -f
```

3. List the images.

```
podman images
```

4. Remove the images.

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

5. List the bridges.

```
nmcli c
```

6. Remove the FSP and the management bridges.

```
nmcli c del fsp_bridge
nmcli c del mgmt_bridge
nmcli c del bridge-slave-enP1p8s0f0
nmcli c del bridge-slave-enP1p8s0f1
```

7. Rename the f0 and f1 connections.

```
ifup enP1p8s0f0
ifup enP1p8s0f1
```

8. [Optional] Clean up OFED.

```
cd /etc/sysconfig/network-scripts
rm -f *bond*
rm -f ifcfg-ib*
nmcli c reload
/sbin/ofed_uninstall.sh --force
```

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

Issue	Resolution or action	Product
	`` 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**.	
During upgrade, if the container had an unintended loss of connection with the target canister(s), there might be a timeout of up to 2 hours in the Ansible update task.	Wait for the timeout and retry the **essrun** update task.	ESS 3000
During storage MES upgrade, you are required to update the drive firmware to complete the task. Some of the drives might not update on the first pass of running the command.	Re-run the **mmchfirmware -type drive** command which should resolve the issue and update the remaining drives.	ESS 3000
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 ******** ```	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`	• ESS 3000 • ESS 5000
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 ```	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* ```	ESS 3000

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.Linux.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. 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 C. 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 29
- 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 29.
- ESS 5000 container has the CES IP addresses defined. For more information, see "ESS 5000 container has the CES IP addresses defined" on page 30.

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 D. 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 C, "ESS 5000 protocol node deployment by using the IBM Spectrum Scale installation toolkit," on page 29.

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 C, “ESS 5000 protocol node deployment by using the IBM Spectrum Scale installation toolkit,” on page 29.

Appendix E. 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

SC28-3134-02