IBM Elastic Storage System 6.1.4.1

Deployment Guide

SC27-9873-02

Note

Before using this information and the product it supports, read the information in "Notices" on page 151.

This edition applies to Version 6 release 1 modification 4 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 xvi. 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

Figures

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 5147-102 documentation consists of the following information units.

Information unit	Type of information	Intended users
Hardware Planning and Installation Guide	This unit provides ESS 5147-102 information including technical overview, planning, installing, troubleshooting, and cabling.	System administrators and IBM support team
Quick Deployment Guide	This unit provides ESS information including the software stack, deploying, upgrading, setting up call home, and best practices.	System administrators, analysts, installers, planners, and programmers of IBM Spectrum Scale clusters who are very experienced with the operating systems on which each IBM Spectrum Scale cluster is based
Service Guide	This unit provides ESS 5147-102 information including servicing and parts listings.	System administrators and IBM support team
Problem Determination Guide	This unit provides ESS 5147-102 information including events, 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

IBM Elastic Storage System (ESS) 3500 documentation consists of the following information units.

Information unit	Type of information	Intended users
Hardware Planning and Installation Guide	This unit provides ESS 3500 information including technical overview, planning, installing, troubleshooting, and cabling.	System administrators and IBM support team

Information unit	Type of information	Intended users
Quick Deployment Guide	This unit provides ESS information including the software stack, deploying, upgrading, setting up call home, and best practices.	System administrators, analysts, installers, planners, and programmers of IBM Spectrum Scale clusters who are very experienced with the operating systems on which each IBM Spectrum Scale cluster is based
Service Guide	This unit provides ESS 3500 information including servicing and parts listings.	System administrators and IBM support team
Problem Determination Guide	This unit provides ESS 3500 information including events, 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

IBM Elastic Storage System (ESS) 3200 documentation consists of the following information units.

Information unit	Type of information	Intended users
Hardware Planning and Installation Guide	This unit provides ESS 3200 information including technical overview, planning, installing, troubleshooting, and cabling.	System administrators and IBM support team
Quick Deployment Guide	This unit provides ESS information including the software stack, deploying, upgrading, setting up call home, and best practices.	System administrators, analysts, installers, planners, and programmers of IBM Spectrum Scale clusters who are very experienced with the operating systems on which each IBM Spectrum Scale cluster is based
Service Guide	This unit provides ESS 3200 information including servicing and parts listings.	System administrators and IBM support team
Problem Determination Guide	This unit provides ESS 3200 information including events, 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

Information unit	Type of information	Intended users
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

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

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

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

Information unit	Type of information	Intended users
Quick Deployment Guide	This unit provides ESS 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

ESS Legacy documentation consists of the following information units.

Information unit	Type of information	Intended users
Quick Deployment Guide	This unit provides ESS 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
Problem Determination Guide	This unit provides 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

Information unit	Type of information	Intended users
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 IBM Documentation.
- mmvdisk command, see mmvdisk documentation.
- Mellanox OFED (MLNX_OFED_LINUX-4.9-5.1.0.2) Release Notes, go to https://docs.nvidia.com/networking/display/MLNXOFEDv494170/MLNX_OFED+Documentation+Rev+4.9-4.1.7.0+LTS.
- Mellanox OFED (MLNX_OFED_LINUX-5.4-3.0.3.0) Release Notes, go to https://docs.nvidia.com/networking/display/MLNXOFEDv562090/Release+Notes. (The Mellanox OFED 5.5.x is shipped with ESS 6.1.4.)
- IBM Elastic Storage System, see IBM Documentation.
- IBM Spectrum Scale call home, see Understanding call home.
- Installing IBM Spectrum Scale and CES protocols with the installation toolkit, see Installing IBM Spectrum Scale on Linux® nodes with the installation toolkit.
- Detailed information about the IBM Spectrum Scale installation toolkit, see Using the installation toolkit to perform installation tasks: Explanations and examples.
- CES HDFS, see Adding CES HDFS nodes into the centralized file system.
- Installation toolkit ESS support, see ESS awareness with the installation toolkit.
- IBM POWER8® servers, see https://www.ibm.com/docs/en/power-sys-solutions/0008-ESS?topic=P8ESS/p8hdx/5148_22l_landing.htm
- IBM POWER9™ servers, see https://www.ibm.com/docs/en/ess/6.1.0_ent?topic=guide-5105-22e-reference-information.

For the latest support information about IBM Spectrum Scale RAID, see the IBM Spectrum Scale RAID FAQ in IBM Documentation.

Conventions used in this information

Table 1 on page xv 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.

Table 1. Conventions (continued)

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

****	In command examples, a backslash indicates that the command or coding example continues on the next line. For example:
	`mkcondition -r IBM.FileSystem -e "PercentTotUsed > 90" \`
	`-E "PercentTotUsed < 85" -m p "FileSystem space used"`

{*item*}	Braces enclose a list from which you must choose an item in format and syntax descriptions.
[*item*]	Brackets enclose optional items in format and syntax descriptions.
<Ctrl-*x*>	The notation <Ctrl-*x*> indicates a control character sequence. For example, <Ctrl-c> means that you hold down the control key while pressing <c>.
***item*...**	Ellipses indicate that you can repeat the preceding item one or more times.
|	In *synopsis* statements, vertical lines separate a list of choices. In other words, a vertical line means *Or*. In the left margin of the document, vertical lines indicate technical changes to the information.

How to submit your comments

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

scale@us.ibm.com

Change history

Version	Changes	PDF form number
3	Defect fixes for ESS 6.1.4.1	SC27-9873-02
2	Updates for ESS 6.1.4.1	SC27-9873-01
1	Initial version for ESS 6.1.4.0	SC27-9873-00

Chapter 1. ESS deployment quick sheet

The quick sheet lists concise sets of steps for the ESS deployment procedures. For more information, see the respective procedures.

COMMON INSTRUCTIONS

1. Download the package from IBM Fix Central.
2. Populate the `/etc/hosts` file on the EMS node.
3. Extract the package.
   ```
   ess??00_6.1.x.x_????-??_dme_ppc64le.sh --start-container
   ```
4. Enter the following information:
   - Is the EMS hostname correct?
   - Type container hostname.
   - Type the container FSP IP address.
5. Run **essrun config load** with the password set by IBM SSR on your setup.
   ```
   essrun -N ems1,io1,io2,io3,io4 config load -p ibmesscluster
   ```
6. Run **essrun config check** with the password set by IBM SSR on your setup.
   ```
   essrun -N ems1,io1,io2,io3,io4 config check -p ibmesscluster
   ```

AFTER SPECIFIC BLOCK HAS BEEN RUN

1. Check health of the system, run the command.
   ```
   essrun -N io1,io2,io3,io4 healthcheck
   ```

NEW DEPLOYMENT

1. If the EMS node needs to be updated, run this command (You might be prompted to run the command again, if the kernel is changed).
   ```
   essrun -N ems1 update --offline
   ```
2. If canisters or I/O nodes need to be updated, run this command.
   ```
   essrun -N io1,io2,io3,io4 update --offline --no-fw-update
   ```
3. Create network bonds.
   ```
   essrun -N ems1,io1,io2,io3,io4 network --suffix=-hs
   ```
4. Set up the NTP server and clients.
   ```
   essrun -N io1,io2,io3,io4 timeset client --server-ip ntp.server.ip
   ```
5. Create the cluster.
   essrun -N io1,io2,io3,io4 cluster --suffix=-hs
6. Add the EMS node to the cluster.
   essrun -N io1 cluster --add-ems ems1 --suffix=-hs
7. Create the file system.
   essrun -N io1,io2,io3,io4 filesystem --suffix=-hs
8. If needed, update the firmware; on each canister/IO node, run these commands.
   mmchfirmware --type storage-enclosure
   mmchfirmware --type drive
9. Configure GUI hardware monitoring
   essrun -N ems1,io1,io2,io3,io4 gui enable --configure
10. Set up call home. For more information about the call home configuration, see Configuring call home on ESS systems.

OFFLINE UPDATE

1. Update the EMS node; from the EMS node, not from the container, run this command.
   ```
   mmshutdown -a
   systemctl stop gpfsgui
   ```
2. Update the EMS node; from the container, run this command [You may be prompted to run a second time if the kernel is changed].
   ```
   essrun -N ems1 update --offline
   ```
3. If the POWER firmware needs to be updated, refer to the online documentation.
4. After the EMS node is updated, from the container, run this command.
   ```
   essrun -N io1,io2,io3,io4 update --offline --no-fw-update
   ```
5. From each canister or IO node, run this command.
   ```
   mmchfirmware --type storage-enclosure
   mmchfirmware --type drive
   ```
6. Start IBM Spectrum Scale; from the EMS node, run this command.
   ```
   mmstartup -a
   ```
7. Enable **gpfsgui** start on reboot.
   ```
   systemctl enable gpfsgui
   ```
8. Start the GUI; from the EMS node, not from the container, run this command.
   ```
   systemctl start gpfsgui
   ```

ONLINE UPDATE

1. Update the EMS node; from the EMS node, not from the container, run this command.
   ```
   mmshutdown
   systemctl stop gpfsgui
   ```
2. Update the EMS node; from the container, run this command [You may be prompted to run a second time if the kernel is changed].
   ```
   essrun -N ems1 update --offline
   ```
3. Start IBM Spectrum Scale on the EMS node.
   ```
   mmstartup
   ```
4. If the POWER firmware needs to be updated, refer to the documentation.
5. After the EMS node is online, from the container, run this command.
   ```
   essrun -N io1,io2,io3,io4 update
   ```
6. Enable **gpfsgui** start on reboot.
   ```
   systemctl enable gpfsgui
   ```
7. Start the GUI; from the EMS node, not from the container, run this command.
   ```
   systemctl start gpfsgui
   ```
8. Rerun the **esscallhomeconf** command with the same arguments that you used when you configured the system. For more information see, Configuring call home on ESS systems.

Chapter 2. ESS software deployment preparation

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

	ESS 3500	ESS 3200	ESS 3000	ESS 5000	ESS Legacy
Runs on	POWER9 EMS	POWER9 EMS	POWER8 or POWER9 EMS	POWER9 EMS	POWER8 or POWER9 EMS
I/O node OS	Red Hat® Enterprise Linux 8.6 x86_64	Red Hat Enterprise Linux 8.6 x86_64	Red Hat Enterprise Linux 8.6 x86_64	Red Hat Enterprise Linux 8.6 x86_64	Red Hat Enterprise Linux 7.9 PPC64LE
Architecture	x86_64	x86_64	x86_64	PPC64LE	PPC64LE
IBM Spectrum Scale	5.1.4.1 efix17	5.1.4.1 efix17	5.1.4.1 efix17	5.1.4.1 efix17	5.1.4.1 efix17
Kernel	4.18.0-372.16.1.el8	4.18.0-372.16.1.el8	4.18.0-372.16.1.el8	4.18.0-372.16.1.el8	3.10.0-1160.71.1.el7.ppc64le
Systemd	239-58.el8.x86_64	239-58.el8.x86_64	239-58.el8.x86_64	239-58.el8.ppc64le	219-78.el7_9.5.ppc64leESS_DME_BASEIMAGE_3000
Network manager	1.36.0-7.el8_6.x86_64	1.36.0-7.el8_6.x86_64	1.36.0-7.el8_6.x86_64	1.36.0-0-7.el8_6.ppc64le	1.18.8-2.el7_9.ppc64le
GNU C Library	glibc-1.36.0-7.el8_6	glibc-1.36.0-7.el8_6	glibc-1.36.0-7.el8_6	glibc-1.36.0-7.el8_6	glibc-2.17-326.el7_9.ppc64le
OFED	MLNX_OFED_LINUX-5.6-2.0.9.0-rhel8.6-x86_64.iso	MLNX_OFED_LINUX-5.6-2.0.9.0 Separate binary for firmware (mlxfwmanager_sriov_dis_x86_64)	MLNX_OFED_LINUX-5.6-2.0.9.0-rhel8.6-x86_64.iso Separate binary for firmware	MLNX_OFED_LINUX-5.6-2.0.9.0-rhel8.6-ppc64le.iso Separate firmware binary	MLNX_OFED_LINUX-4.9-5.1.0.2-rhel7.9-ppc64le.iso Firmware binary included
Firmware RPM	6.0.0.51	6.0.0.51	6.0.0.51	6.0.0.51	6.0.0.51
SAS Adapter Firmware		N/A	N/A	16.00.11.00 - 4U106 and 5U92	16.00.11.00
Mpt3sas		N/A	N/A	38.00.00.00 - 5U92 (not in box) 41.00.00.00 - 4U106 (not in box)	34.00.00.00 (not in box)

	ESS 3500	ESS 3200	ESS 3000	ESS 5000	ESS Legacy
Platform RPM	gpfs.ess.platform.ess3500-5.1.4-1.17.x86_64.rpm	gpfs.ess.platform.ess3200-5.1.4-1.17x86_64.rpm	gpfs.ess.platform.ess3000-5.1.4-1.17.x86_64.rpm	N/A	N/A
Drive format		4 KiB + 0 B (non-FCM); 512 KiB+0 (FCM)	4 KiB + 0 B		
Support RPM	gpfs.gnr.support-ess3500-6.1.4-1.noarch.rpm	gpfs.gnr.support-ess3200-6.1.4-1.noarch.rpm	gpfs.gnr.support-ess3000-6.1.4-1.noarch.rpm	gpfs.gnr.support-ess5000-6.1.4-1.noarch.rpm	gpfs.gnr.support-essbase-6.1.4-1.noarch.rpm
Podman	1.6.4-11	1.6.4-11	1.6.4 RHEL6	1.6.4-11	1.6.4-11 (1.4.4 RHEL7)
Container version	Red Hat UBI 8.6	Red Hat UBI 8.6	Red Hat UBI 8.6	Red Hat UBI 8.6	Red Hat Enterprise Linux 7.9
Ansible®	2.9.27-1	2.9.27-1	2.9.27-1	2.9.27-1	2.9.27-1
xCAT	2.16.3 (For internal use only - not on IBM Fix Central.)	2.16.3 Not used in customer shipped image - only for SCT	2.16.3	2.16.3 (for SCT only)	2.16.3 (for SCT only)
PEMS			1111	N/A	N/A
ndctl		N/A	N/A	ndctl-65-1.el8	N/A
OPAL		opal-prd-ess.v4-1.el8.x86_64.rpm	N/A	opal-prd-3000.0-1.el8 opal-prd-ess.v4.1-1.el8.ppc64le.rpm	N/A

	ESS 3500	ESS 3200	ESS 3000	ESS 5000	ESS Legacy
System firmware	Canister firmware • BIOS: RWH3LJ-12.07.00 • BMC: 12.56 • Server0FPGA: 0110 • Server1FPGA: 0110 • Midplane1PrimaryFPGA: 0344 • Midplane1SecondaryFPGA: 0344 • Midplane2PrimaryFPGA: 0344 • Midplane2SecondaryFPGA: 0344 • Midplane3PrimaryFPGA: 0344 • Midplane3SecondaryFPGA: 0344 • DriveplanePrimaryFPGA: 0527 DriveplaneSecondaryFPGA: 0527	RWH1-12.16.00_12.52_0140_0140_0343_0343_0343_0326_954300P0_954300P0 • BMC: 12.52 • Server0FPGA: 0140 • Server1FPGA: 0140 • Midplane1PrimaryFPGA: 0343 • Midplane1SecondaryFPGA: 0343 • Midplane2PrimaryFPGA: 0343 • Midplane2SecondaryFPGA: 0343 • Midplane3PrimaryFPGA: 0343 • Midplane3SecondaryFPGA: 0343 • DriveplanePrimaryFPGA: 0326 • DriveplaneSecondaryFPGA: 0326	2.02.000_0B0G_1.73_FB30005	FW950.50 (FW950.105) NVDIMM ver: Bundled BPM ver: Bundled	FW860.B1 (SV860_243)
Boot drive	• Bootdrive1_Micron_7300_MTFDHBA960TDF: 954300P0 • Bootdrive2_Micron_7300_MTFDHBA960TDF: 954300P0	• Bootdrive1_Micron_7300_MTFDHBA960TDF: 954300P0 • Bootdrive2_Micron_7300_MTFDHBA960TDF: 954300P0	• SMART: Prod ID: SRM2S86Q800GQT51IM P/N: 01LL447IBM FRU: 01LL447 FW: 1361 • Micron: MTFDDAV960TDS P/N: 01LL446IBM FRU: 01LL587 FW: ML32	9F23	E700

<table>
<tr><th></th><th>ESS 3500</th><th>ESS 3200</th><th>ESS 3000</th><th>ESS 5000</th><th>ESS Legacy</th></tr>
<tr><td>Enclosure firmware</td><td>E11G</td><td>E114</td><td>N/A</td><td>5U92 - E558
4U106 - 5266</td><td>PPC64LE Slider
2U24 - 4230
5U84 - 4087
4U106 - 5284</td></tr>
<tr><td>NVMe firmware</td><td><table><tr><th>Prod ID</th><th>FRU</th><th>Firmware version</th></tr><tr><td>3.84 TB NVMe Tier-1 Flash</td><td>01LL727</td><td>SN5ASN5A</td></tr><tr><td>7.64 TB NVMe Tier-1 Flash</td><td>01LL728</td><td>SN5ASN5A</td></tr><tr><td>15.36 TB NVMe Tier-1 Flash</td><td>01LL729</td><td>SN5ASN5A</td></tr></table></td><td><table><tr><th>Prod ID</th><th>FRU</th><th>Firmware version</th></tr><tr><td>3.84 TB NVMe Tier-1 Flash</td><td>01LL727</td><td>SN5ASN5A</td></tr><tr><td>7.64 TB NVMe Tier-1 Flash</td><td>01LL728</td><td>SN5ASN5A</td></tr><tr><td>15.36 TB NVMe Tier-1 Flash</td><td>01LL729</td><td>SN5ASN5A</td></tr></table></td><td>SN1MSN1M</td><td>N/A</td><td>N/A</td></tr>
</table>

	ESS 3500	ESS 3200	ESS 3000	ESS 5000	ESS Legacy
Network adapter	• MT27500 = 10.16.1020 • MT4099 = 2.42.5000 • MT26448 = 2.9.1326 • MT4103 = 2.42.5000 • MT4113 = 10.16.1200 • MT4115 = 12.28.2006 • MT4117 = 14.32.1010 • MT4118 = 14.32.1010 • MT4119 = 16.33.1048 • MT4120 = 16.33.1048 • MT4121 = 16.33.1048 • MT4122 = 16.33.1048 • MT4123 = 20.33.1048 • MT4125 = 22.33.1048	• MT27500 = 10.16.1020 • MT4099 = 2.42.5000 • MT26448 = 2.9.1326 • MT4103 = 2.42.5000 • MT4113 = 10.16.1200 • MT4115 = 12.28.2006 • MT4117 = 14.32.1010 • MT4118 = 14.32.1010 • MT4119 = 16.33.1048 • MT4120 = 16.33.1048 • MT4121 = 16.33.1048 • MT4122 = 16.33.1048 • MT4123 = 20.33.1048 • MT4125 = 22.33.1048	CX5-VPI • MT27500 = 10.16.1020 • MT4099 = 2.42.5000 • MT26448 = 2.9.1326 • MT4103 = 2.42.5000 • MT4113 = 10.16.1200 • MT4115 = 12.28.2006 • MT4117 = 14.32.1010 • MT4118 = 14.32.1010 • MT4119 = 16.33.1048 • MT4120 = 16.33.1048 • MT4121 = 16.33.1048 • MT4122 = 16.33.1048 • MT4123 = 20.33.1048 • MT4125 = 22.33.1048	• MT27500 = 10.16.1020 • MT4099 = 2.42.5000 • MT26448 = 2.9.1326 • MT4103 = 2.42.5000 • MT4113 = 10.16.1200 • MT4115 = 12.28.2006 • MT4117 = 14.32.1010 • MT4118 = 14.32.1010 • MT4119 = 16.33.1048 • MT4120 = 16.33.1048 • MT4121 = 16.33.1048 • MT4122 = 16.33.1048 • MT4123 = 20.33.1048 • MT4125 = 22.33.1048	MT4120 CX-5 EN 01FT741 MT4121 CX-5 VPI 01LL584 MT4122 CX-5 SRIOV VF 01LL584 • MT27500 = 10.16.1020 • MT4099 = 2.42.5000 • MT26448 = 2.9.1326 • MT4103 = 2.42.5000 • MT4113 = 10.16.1200 • MT4115 = 12.28.2006 • MT4117 = 14.32.1010 • MT4118 = 14.32.1010 • MT4119 = 16.33.1048 • MT4120 = 16.33.1048 • MT4121 = 16.33.1048 • MT4122 = 16.33.1048 • MT4123 = 20.33.1048 • MT4125 = 22.33.1048
ESA	esagent.pLinux-4.5.7-0	esagent.pLinux-4.5.7-0	esagent.pLinux-4.5.7-0	esagent.pLinux-4.5.7-0	esagent.pLinux-4.5.7-0
BIOS	RWH3LJ-12.07.00	12.16.00	52	N/A	N/A
HAL	ibm.ess-hal-2.1.1.0-5.1.x86_64.rpm	ibm.ess-hal-2.1.1.0-5.1.x86_64.rpm	N/A	N/A	N/A

Changes in this release

- Support for IBM Spectrum Scale 5.1.4.1 efix17

- OFED 5.6.x (ESS 3000/ESS 5000/ESS 3200)
- Support for a new P8/P9 firmware (ESS Legacy/ESS 5000)
- Support for Red Hat Enterprise Linux 8.6 (ESS 3000/ESS 3200/ESS 3500)
- Support for a new glibc
- Support for a new kernel

POWER9 EMS stack

Item	Version
IBM Spectrum Scale	IBM Spectrum Scale 5.1.4.1 efix17
Operating system	Red Hat Enterprise Linux 8.6
ESS	ESS 6.1.4.1
Kernel	4.18.0-372.16.1.el8
Systemd	239-58.el8
Network Manager	1.36.0-7.el8_6.ppc64le
GNU C Library	1.36.0-7.el8_6
Mellanox OFED	MLNX_OFED_LINUX-5.6-2.0.9.0 Separate firmware binary (mlxfwmanager_sriov_dis_ppc64le)
ESA	4.5.7-0
Ansible	2.9.27-1
Podman	1.6.4
Container OS	Red Hat UBI 8.6
xCAT	2.16.3 (Not used in customer-shipped image; only for SCT)
Firmware RPM	gpfs.ess.firmware-6.1.4-02.ppc64le.rpm
System firmware	FW950.50 (FW950.105)
Boot drive adapter	IPR 19512c00
Boot drive firmware	• Firmware: 9F23 • Host adapter driver: 38.00.00.00 • Host adapter firmware: 16.00.11.00
1Gb NIC firmware	• Driver: tg3 • Version: 3.137 • Firmware version: 5719-v1.24i
Support RPM	• gpfs.gnr.support-ess3000-6.1.4-1.noarch.rpm • gpfs.gnr.support-ess3200-6.1.4-1.noarch.rpm • gpfs.gnr.support-essbase-6.1.4-1.noarch.rpm • gpfs.gnr.support-ess5000-6.1.4-1.noarch.rpm • gpfs.gnr.support-ess3500-6.1.4-1.noarch.rpm

Item	Version
Network adapter	• MT27500 = 10.16.1020 • MT4099 = 2.42.5000 • MT26448 = 2.9.1326 • MT4103 = 2.42.5000 • MT4113 = 10.16.1200 • MT4115 = 12.28.2006 • MT4117 = 14.32.1010 • MT4118 = 14.32.1010 • MT4119 = 16.32.2004 • MT4120 = 16.32.2004 • MT4121 = 16.32.2004 • MT4122 = 16.32.2004 • MT4123 = 20.32.2004 • MT4125 = 22.32.2004

Support matrix

Release	OS	Runs on	Can upgrade or deploy
ESS 3500 6.1.4	Red Hat Enterprise Linux 8.6 (x86_64)	POWER9 EMS	• ESS 3500 nodes • POWER9 EMS • POWER9 protocol nodes
ESS 3200 6.1.4	Red Hat Enterprise Linux 8.6 (x86_64)	• POWER9 EMS	• ESS 3200 nodes • POWER9 EMS • POWER9 protocol nodes
ESS 3000 6.1.4	Red Hat Enterprise Linux 8.6 (x86_64)	• POWER8 EMS • POWER9 EMS	• ESS 3000 nodes • POWER8 EMS • POWER9 EMS • POWER8 protocol nodes • POWER9 protocol nodes
ESS 5000 6.1.4	Red Hat Enterprise Linux 8.6 (PPC64LE)	• POWER9 EMS	• ESS 5000 nodes • POWER9 EMS • POWER9 protocol nodes

Release	OS	Runs on	Can upgrade or deploy
ESS Legacy 6.1.4	• Red Hat Enterprise Linux 8.6 (PPC64LE) • Red Hat Enterprise Linux 7.9 (PPC64LE)	• POWER8 EMS • POWER9 EMS	• ESS POWER8 I/O nodes (PPC64LE) • ESS POWER8 protocol nodes (PPC64LE) • ESS POWER9 protocol nodes (PPC64LE)* • POWER8 EMS • POWER9 EMS

Prerequisites

- This document (ESS 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 the respective ESS Hardware Guide (ESS 3000 or ESS 5000 or ESS 3200 or ESS 3500) for hardware checkout and setting IP addresses.
- Worksheet notes from the SSR
- Latest ESS xz downloaded to the EMS node from Fix Central (If a 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 that 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 in the `/etc/hosts` file.
- Host and domain name (FQDN) are defined in the `/etc/hosts` file.
- **ESS Legacy 6.1.x.x Only:** You must convert to `mmvdisk` before deploying the ESS Legacy 6.1.x.x container if you are coming from a non-container version such as ESS 5.3.x.x. If you have not done so already, convert to `mmvdisk` by using the following steps:

  1. Check whether there are any `mmvdisk` node classes.

     ```
     mmvdisk nodeclass list
     ```

     There should be one node class per ESS Legacy building-block. If the command output does not show `mmvdisk` for your ESS Legacy nodes, convert to `mmvdisk` before running the ESS Legacy 6.1.0.x container.

  2. Convert to `mmvdisk` by running the following command from one of the POWER8 I/O nodes or from the POWER8 EMS node.

     ```
     gssgenclusterrgs -G gss_ppc64 --suffix=-hs --convert
     ```

     You can also use `-N` with a comma-separated list of nodes.

     Note: Wait for 5 minutes for daemons to recycle. The file system remains up.

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

- ESS 3500 tgz used in manufacturing (may not be the latest)
- ESS 5000 tgz used in manufacturing (may not be the latest)

- ESS 3000 tgz used in manufacturing (may not be the latest)
- ESS Legacy tgz used in manufacturing (may not be the latest)
- ESS 3200 tgz used in manufacturing (may not be the latest)

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 whether 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, ESS 5000, 3500, and ESS Legacy networking requirements

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

- Management VLAN
- Service/FSP VLAN

Note: To future proof your environment for ESS 3200, modify any existing management switches to the new VLAN configuration. For more information, see Appendix O, “Switch VLAN configuration instructions,” on page 107.

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

ESS 3000

POWER8 or POWER9 EMS

It is recommended to buy a POWER9 EMS with ESS 3000. If you have a legacy environment (POWER8), it is recommended to migrate to IBM Spectrum Scale 5.1.x.x and use the POWER9 EMS as the single management server.

- 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 required 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).
  - Port 1 on each ESS 3000 canister must be connected to the management VLAN.
- 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.

Note: It is mandatory that you connect C11-T3 to a campus connection or run an additional management connection. If you do not do this step, you will lose the connection to the EMS node when the container starts.

ESS 5000 or ESS 3200

POWER9 EMS support only

EMS must have the following connections:

- C11-T1 to the management VLAN
- C11-T2 to the FSP VLAN
- C11-T3 to the campus network
- HMC1 to the FSP VLAN

ESS 5000 nodes must have the following connections:

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

ESS 3200 nodes must have the following connections:

- Single management connection per canister:
  - Each connection is split between 2 MAC addresses:
    1. BMC
    2. Operating system
  - The BMC connection requires a VLAN tag to be set for proper communication with the EMS node.

ESS 3200 requirements

- Management connections
  - Shared management port (visible to OS)
- BMC connection
  - Shared management port (visible to BMC)
- High-speed connections
  - InfiniBand or Ethernet

Management switch

- Typically, a 48-port switch
- Two VLANs required
  - Management VLAN (VLAN 102)
  - FSP/BMC VLAN (VLAN101)
- ESS 3200 dedicated trunk ports
  - Routes BMC traffic to VLAN 101

Note: The VLANs shown here are default for the IBM Cumulus switch. The VLAN value can be modified according to your environment.

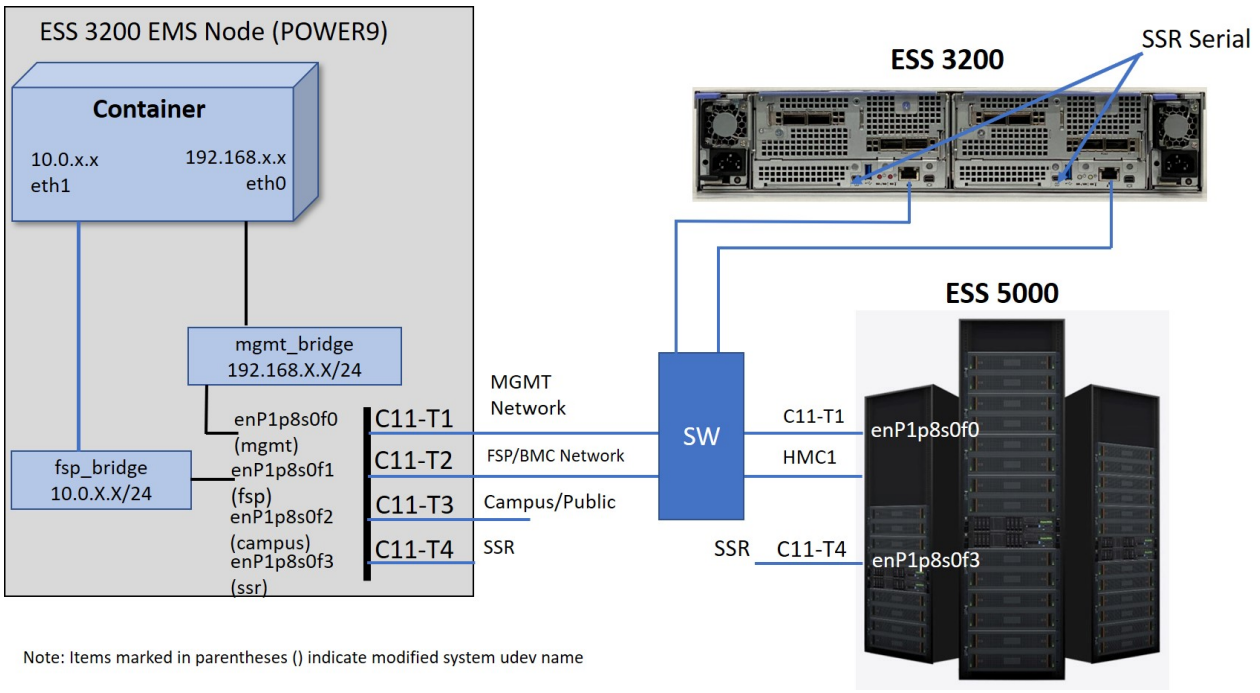


Figure 1. ESS 3200 container networking

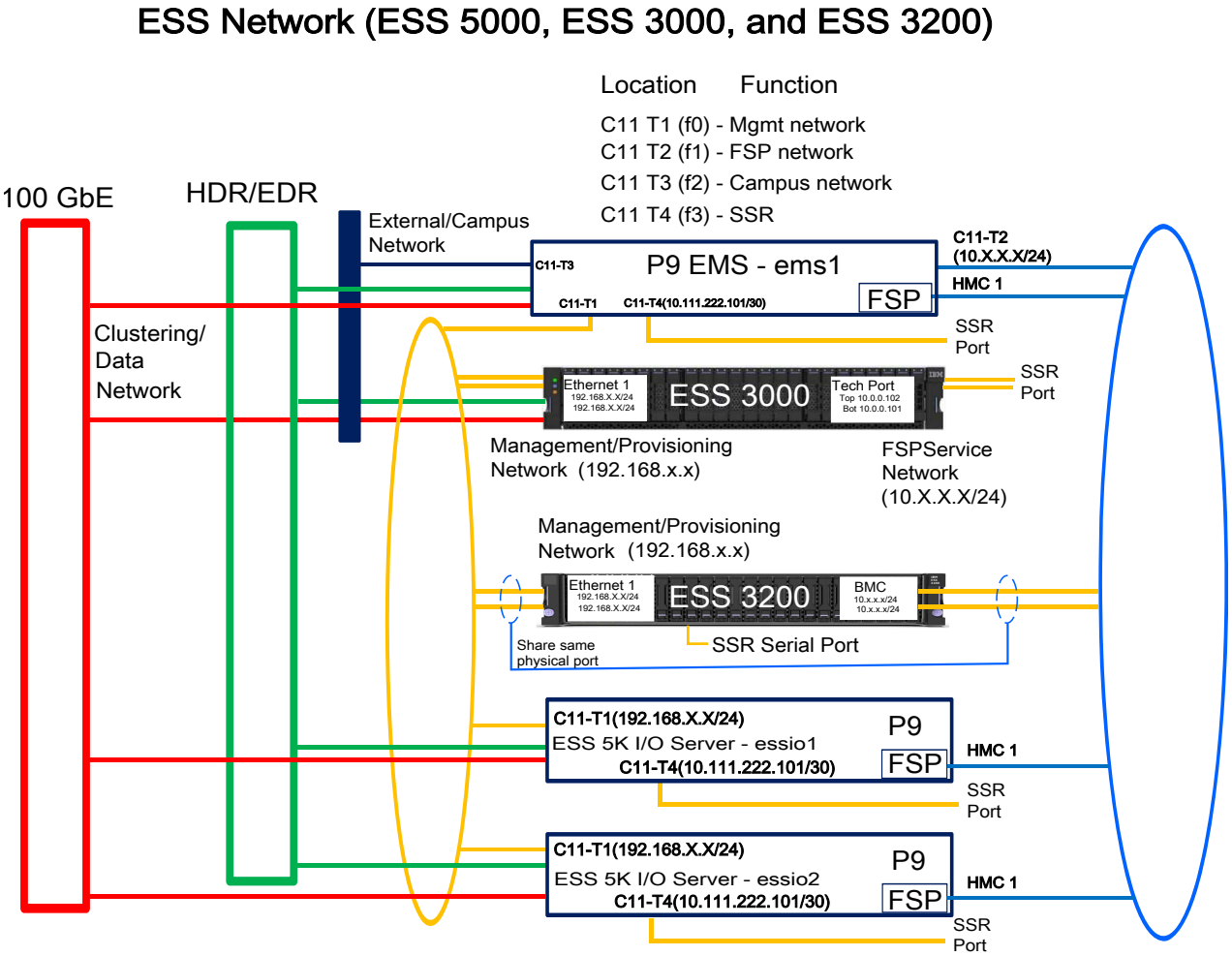


Figure 2. ESS 3200 network diagram

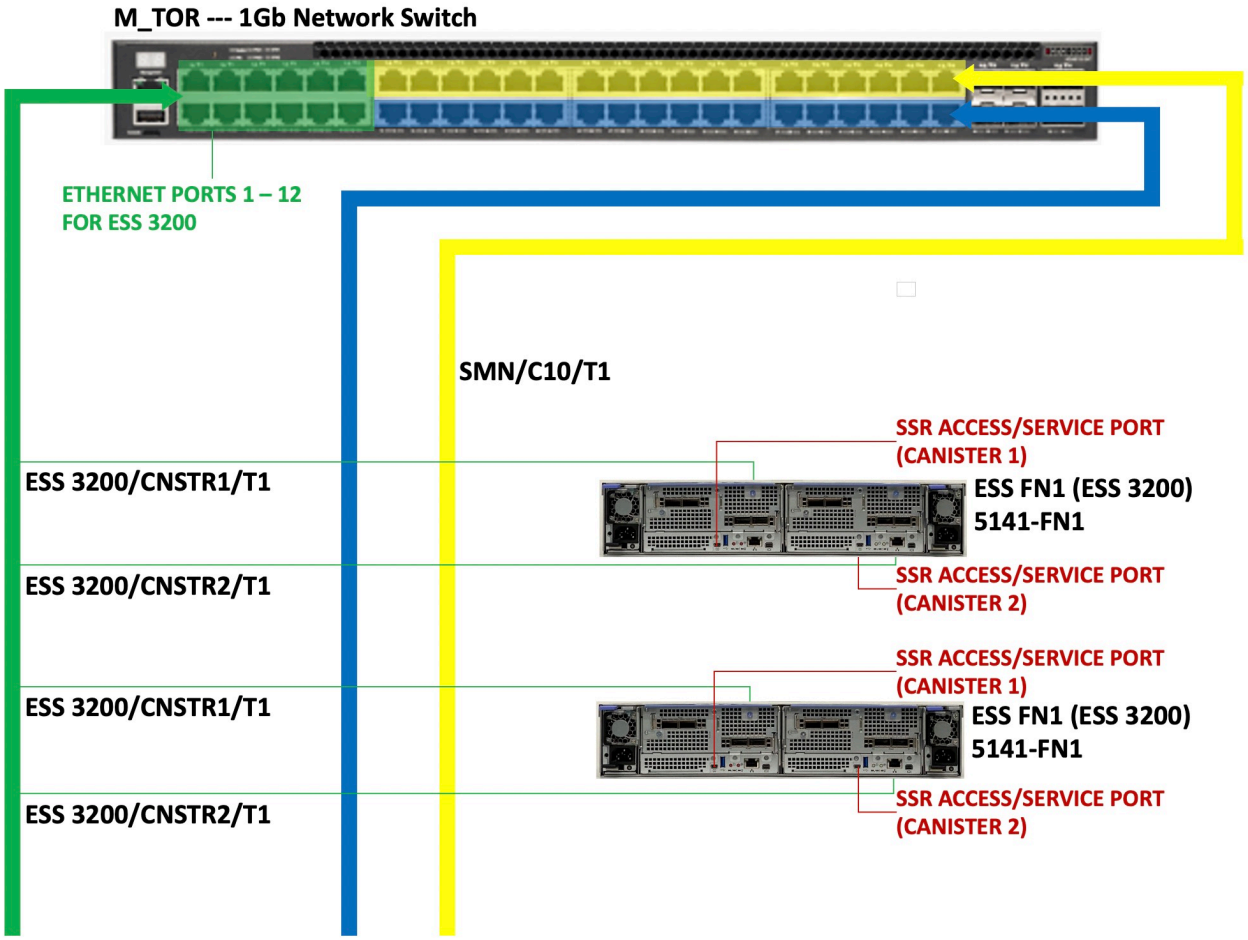


Figure 3. ESS 3200 Ethernet ports and switch

The ports highlighted in green are the ESS 3200 trunk ports. These are special ports that are for the ESS 3200 only. The reason for these ports is that each ESS 3200 canister has a single interface for both the BMC and the OS but unique MAC addresses. By using a VLAN tag, canister BMC MAC addresses are routed to the BMC/FSP/Service VLAN (Default is 101).
IBM racked orders have the switch preconfigured. Only the VLAN tag needs to be set. If you have an existing IBM Cumulus switch or customer supplied switch, it needs to be modified to accommodate the ESS 3200 trunk port requirement. For more information, see Appendix O, “Switch VLAN configuration instructions,” on page 107.

Note: It is mandatory that you connect C11-T3 to a campus connection or run an additional management connection. If you do not do this step, you will lose the connection to the EMS node when the container starts.

ESS 3500 network requirements

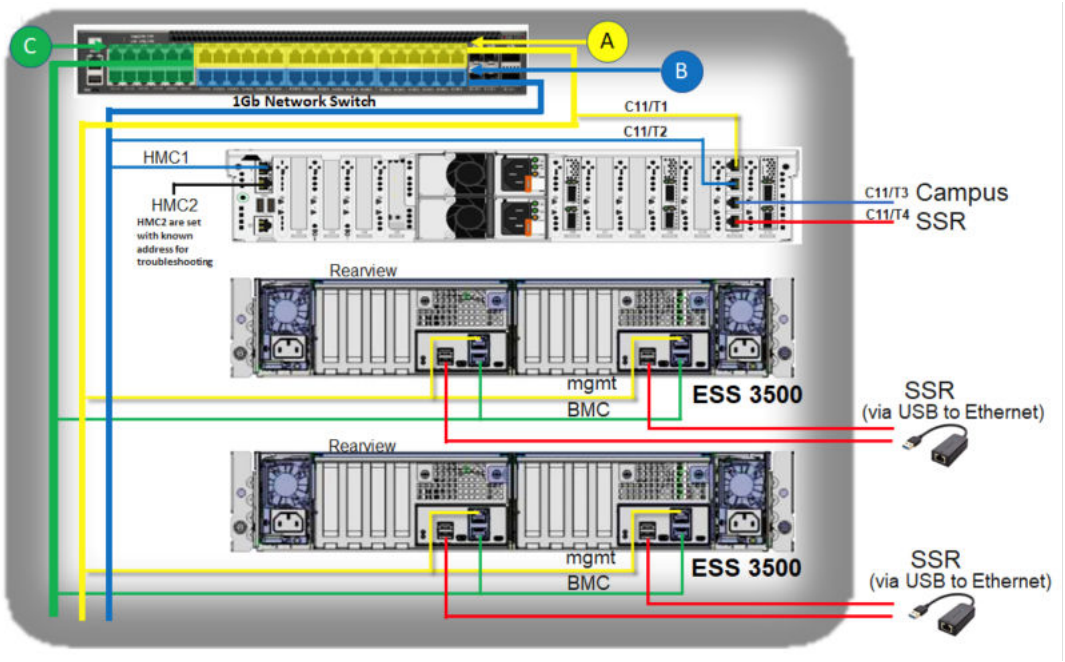


Figure 4. ESS 3500 network

ESS Legacy

POWER8 or POWER9 EMS supported

POWER8 EMS must have the following connections:

- C10-T1 to the management VLAN
- C10-T4 to the FSP/Service VLAN
- C10-T2 to the management VLAN
- C10-T3 optional campus connection
- HMC1 to the FSP/Service VLAN

POWER9 EMS must have the following connections:

- C11-T1 to the management VLAN
- C11-T2 to the FSP VLAN
- HMC1 to the FSP VLAN
- C11-T3 to the campus or management network/VLAN

POWER8 nodes:

- C12-T1 to the management VLAN
- HMC1 to the FSP VLAN

An output example after migrating the network from ESS 5.3.7.x before attempting to start the ESS 6.x.x.x container is as follows:

- Before creating the network bridges:

```
# ip a |grep "enP3\|bridge"
2: enP3p9s0f0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default
qlen 1000
    inet 192.168.45.20/24 brd 192.168.45.255 scope global noprefixroute enP3p9s0f0
3: enP3p9s0f1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default
qlen 1000
4: enP3p9s0f2: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default
qlen 1000
    inet 9.155.113.184/20 brd 9.155.127.255 scope global noprefixroute enP3p9s0f2
```

```
5: enP3p9s0f3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default
qlen 1000
```

```
# Sample essmgr.yml
CONTAINER:
  BKUP: /home/backup
  CONTAINER_DOMAIN_NAME: gpfs.net
  CONTAINER_HOSTNAME: cems0
  FSP_BRIDGE_IP: 10.0.0.2
  FSP_BRIDGE_NAME: fsp_bridge
  FSP_CONTAINER_IP: 10.0.0.5
  FSP_INTERFACE: enP3p9s0f3
  FSP_SUBNET: 10.0.0.0/24
  INSTALLER_HOSTNAME: ems1
  LOG: /home/log
  MGMT_BRIDGE_IP: 192.168.45.2
  MGMT_BRIDGE_NAME: mgmt_bridge
  MGMT_CONTAINER_IP: 192.168.45.80
  MGMT_INTERFACE: enP3p9s0f1
  MGMT_SUBNET: 192.168.45.0/24
```

- After creating the bridges (`./essmgr -n`):

```
# ip a |grep "enP3\|bridge"
2: enP3p9s0f0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default
qlen 1000
    inet 192.168.45.20/24 brd 192.168.45.255 scope global noprefixroute enP3p9s0f0
3: enP3p9s0f1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq master mgmt_bridge state
UP group default qlen 1000
4: enP3p9s0f2: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default
qlen 1000
    inet 9.155.113.184/20 brd 9.155.127.255 scope global noprefixroute enP3p9s0f2
5: enP3p9s0f3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq master fsp_bridge state UP
group default qlen 1000
65: mgmt_bridge: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group
default qlen 1000
    inet 192.168.45.2/24 brd 192.168.45.255 scope global noprefixroute mgmt_bridge
67: fsp_bridge: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group
default qlen 1000
    inet 10.0.0.2/24 brd 10.0.0.255 scope global noprefixroute fsp_bridge
```

Code version

ESS Legacy, ESS 3000, ESS 3200, ESS 5000, and ESS 3500 releases are included in ESS 6.1.4.x with two editions: Data Management Edition and Data Access Edition. An example of package names is as follows:

```
ess_6.1.4.1_0919-18_dme_ppc64le.tar.xz
ess_6.1.4.1_0919-18_dae_ppc64le.tar.xz
```

Note:

- The versions shown here might not be the GA version available on IBM FixCentral. It is recommended to go to IBM FixCentral and download the latest code.
- `ppc64le` in the package name implies that each container runs on a POWER®-based EMS. For details about functions supported by respective containers, see “Support matrix” on page 9.

You can download the latest 6.1.x.x code (6.1.4.1 is the latest) from IBM Fix Central by using the following link.

- IBM Fix Central download link

A unified container is offered with two versions each (Data management + Data access). Example package names for each container are as follows:

```
// Unified Container (Data Access and Data Management versions)
ESS_DAE_BASEIMAGE-6.1.4.1-ppc64LE-Linux.tgz
ESS_DME_BASEIMAGE-6.1.4.1-ppc64LE-Linux.tgz
```

Note: The container installs and runs on the EMS only. The EMS supported is Power-based only. Running container on a x86-based node is not supported as of now.

POWER8 considerations

If you are moving from an xCAT-based release (5.3.x) to a container-based release (6.1.x.x), the following considerations apply:

- You must add an additional management network connection to C10-T2.
- A public or additional management connection is mandatory in C10-T3.
- You must stop and uninstall xCAT and all xCAT dependencies before installing the container.

Remote management considerations

Data center access has become more restrictive nowadays. Here are some considerations to enable remote support:

- Always add a campus connection to the EMS (POWER8 and POWER9).
- Consider adding campus connections to the HMC2 ports on all POWER servers (ESS Legacy, ESS 5000, POWER8 or POWER9 EMS). Consider cabling this port to a public network and setting a campus IP. This will allow remote recovery or debug of the EMS in case of an outage.
- Consider adding campus connections to C11-T3 (POWER9 nodes) or C10-T3 (POWER8 nodes).
- Consult with service about adding USB to Ethernet dongle to enable campus connections on the ESS 3200 system.
- Add campus connection to a free port on each ESS 3000 canister. Also consider adding SMART PDUs on ESS 3000 frames to help remotely power cycle the system.

POWER8 + POWER9 considerations

- If both POWER8 and POWER9 EMS nodes are in an environment, it is recommended that you use only the POWER9 EMS for management functions (containers, GUI, ESA, collector).
- Only a single instance of all management services is recommended and solely on the POWER9 EMS.
- POWER8 only needs to exist as a management node if you are mixing a non-container-based release (5.3.x) with a container-based release (6.x.x.x).
- It is recommended that all nodes in the storage cluster contain the same ESS release and IBM Spectrum Scale version.
- It is recommended that you upgrade to the latest level before adding a building block.

Note: If you are mixing ESS Legacy 5.3.x and ESS 3000 on a POWER8 EMS, the following considerations apply:

- You cannot upgrade the EMS node from the ESS 3000 container.
- ESS 3000 detects if xCAT is installed on the host EMS node. If xCAT is installed, it stops the upgrade.
- You must upgrade the EMS node by using the legacy deployment procedure outlined in *ESS 5.3.x Quick Deployment Guide*.

Migrating from an ESS Legacy environment (xCAT-based 5.3.x) to an ESS Legacy container-based environment (6.1.x.x)

The following guidance is for customers migrating from an xCAT-based release to a container-based release for POWER8 offerings.

POWER9 EMS

You cannot run both POWER8 and POWER9 EMS nodes in the same environment for ESS Legacy. If you are moving a POWER9 EMS, migrate all services from the POWER8 EMS and uninstall xCAT. You can then re-use the POWER8 EMS for other purposes such as quorum node, client node, or spare EMS. The preference is to always use a POWER9 EMS if possible and you must not run multiple instances of GUI, performance monitoring collectors, etc. in the same cluster. For this requirement,

there are exceptions for certain stretch cluster environments and if you are mixing ESS Legacy and container-based deployments such as ESS 5.3.7 on POWER8 and ESS 6.0.2.x on POWER9.

POWER8 EMS

If you are migrating from ESS 5.3.x to ESS 6.1.0.x on a POWER8 EMS, do the following steps.

1. Stop and uninstall xCAT by doing the following steps on a POWER8 EMS, outside of the container.
   a. Stop xCAT.
   ```
   systemctl stop xcatd
   ```
   b. Uninstall xCAT.
   ```
   yum remove xCAT*
   ```
   c. Remove dependencies.
   ```
   yum remove dbus-devel dhcp bind java-1.8.0-openjdk
   ```
2. Add a container connection to C10-T2.
3. Add a campus connection to C10-T3, if it is not done already.
4. Update `/etc/hosts` with the desired container host name and IP address.

Other notes

- The following tasks must be complete 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 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 code levels are at the latest at time of manufacturing ship.
- Customer must make sure that 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 (POWER nodes only) are set to the server serial number. Consider changing these passwords when the deployment is complete.
- Check whether the SSSD service is running on EMS and other nodes. Shut down the SSSD service on those nodes manually, before you upgrade the nodes.
- RHEL server nodes might be communicating to root DNS directly and are not routed through internal DNS. If this is not permitted in the environment, you might override the default service configuration or disable it. For more information about background and resolution options, see https://access.redhat.com/solutions/3553031.

ESS best practices

- ESS 6.x.x.x uses a new embedded license. It is important to know that installation of any Red Hat packages outside of the deployment upgrade flow is not supported. The container image provides

everything required for a successful ESS deployment. If additional packages are needed, contact IBM for possible inclusion in future versions.

- For ESS 3000, consider enabling TRIM support. This is outlined in detail in *IBM Spectrum Scale RAID Administration*. By default, ESS 3000 only allocates 80% of available space. Consult with IBM development, if going beyond 80% makes sense for your environment, that is if you are not concerned about the performance implications due to this change.
- You must setup a campus or additional management connection before deploying the container.
- If running with a POWER8 and a POWER9 EMS in the same environment, it is best to move all containers to the POWER9 EMS. If there is a legacy PPC64LE system in the environment, it is best to migrate all nodes to ESS 6.1.x.x and decommission the POWER8 EMS altogether. This way you do not need to run multiple ESS GUI instances.
- If you have a POWER8 EMS, you must upgrade the EMS by using the legacy flow if there are xCAT based PPC64LE nodes in the environment (including protocol nodes). If there are just an ESS 3000 system and a POWER8 EMS, you can upgrade the EMS from the ESS 3000 container.
- If you are migrating the legacy nodes to ESS 6.1.x.x on the POWER8 EMS, you must first uninstall xCAT and all dependencies. It is best to migrate over to the POWER9 EMS if applicable.
- You must be at ESS 5.3.7 (Red Hat Enterprise Linux 7.7 / Python3) or later to run the ESS 3000 container on the POWER8 EMS.
- You must run the **essrun config load** command against all the storage nodes (including EMS and protocol nodes) in the cluster before enabling admin mode central or deploying the protocol nodes by using the installation toolkit.
- If you are running a stretch cluster, you must ensure that each node has a unique `hostid`. The `hostid` might be non-unique if the same IP addresses and host names are being used on both sides of the stretch cluster. Run **gnrhealthcheck** before creating recovery groups when adding nodes in a stretch cluster environment. You can manually check the `hostid` on all nodes as follows:

  ```
  mmdsh -N { NodeClass | CommaSeparatedListofNodes } hostid
  ```

  If `hostid` on any node is not unique, you must fix by running **genhostid**. These steps must be done when creating a recovery group in a stretch cluster.
- Consider placing your protocol nodes in file system maintenance mode before upgrades. This is not a requirement but you should strongly consider doing it. For more information, see File system maintenance mode.
- Do not try to update the EMS node while you are logged in over the high-speed network. Update the EMS node only through the management or the campus connection.
- After adding an I/O node to the cluster, run the **gnrhealthcheck** command to ensure that there are no issues before creating vdisk sets. For example, duplicate host IDs. Duplicate host IDs cause issues in the ESS environment.
- Run the container from a direct SSH connection. Do not SSH from an I/O node or any node that might be rebooted by the container.
- Do not log in and run the container over the high-speed network. You must log in through the campus connection.
- You must stop Spectrum Scale tracing **(mmtrace | mmtracectl)** before starting the container or deploying any node. The container attempts to block if tracing is detected, it is recommended to manually inspect each ESS node before attempting to deploy.
- Heavy IBM Spectrum Scale and I/O operations must be suspended before upgrading an ESS environment.

  Wait for any of the following commands that are performing file system maintenance tasks to complete:

  - **mmadddisk**
  - **mmapplypolicy**
  - **mmcheckquota**

- **mmdeldisk**
- **mmfsck**
- **mmlssnapshot**
- **mmrestorefs**
- **mmrestripefile**
- **mmrestripefs**
- **mmrpldisk**

Stop the creation and deletion of the snapshots by using the **mmcrsnapshot** and **mmdelsnapshot** commands during the upgrade.

Support notes and rules

- Multiple EMS nodes are not supported in the same cluster. If you are adding a POWER9 EMS to an existing cluster run by a POWER8 EMS, the POWER9 EMS must be the only one used for management functions such as GUI, performance monitoring collector, etc.
- Multiple GUI instances are not supported in the same cluster.
- One collector node must be run at a time in the cluster. This must be on the same node as the GUI.
- You cannot mix majoresagent.pLinux-4.5 IBM Spectrum Scale versions in the storage cluster. All nodes must be updated to the latest level.
- ESA must be running on the EMS.
- You can run call home on the EMS.
- If possible, run the client nodes in a separate cluster than the storage.
- The essrun (ESS deployment Ansible wrapper tool run within the container) tool does not use the GPFS admin network. It uses the management network only to communicate from the container to each of the nodes.
- If POWER8 EMS only, consolidate potential xCAT and non-xCAT offerings to container versions.

  Example: If you have ESS 5.3.7.x (Legacy POWER8 offering on Scale 5.0.5.x) and ESS 3000 (Containerized support for ESS 3000 on Scale 5.x.x.x and above), convert the Legacy 5.3.7.x to 6.1.x.x so that only containers are running on POWER8 EMS.

  Note: This only applies to situations where there was already Scale 5.1.x.x+ in the environment.

  Note: There is no container offering for BE so environments with BE would have to remain at 5.0.5 release level (but the POWER8 EMS could still move to all container version).
- If POWER8 EMS and POWER9 EMS are owned by the customer, it is recommended to consolidate to POWER9 EMS (all container versions).

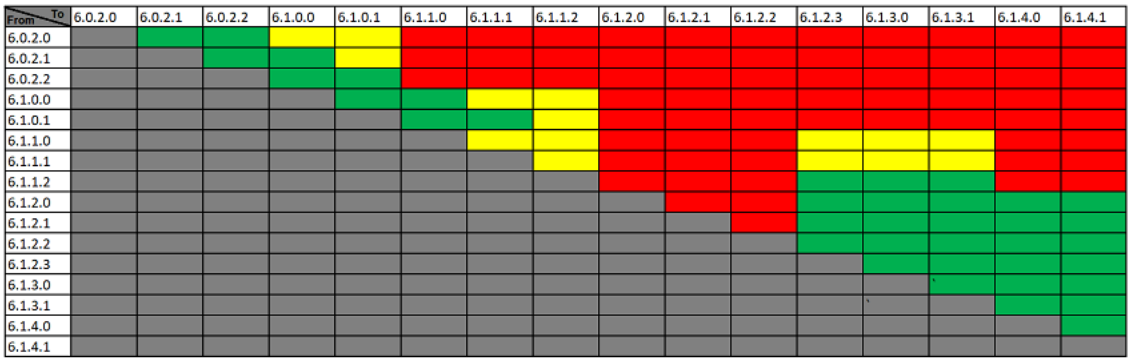
  Example: If POWER8 EMS was running 5.1.x.x (ESS 3000, ESS Legacy or both) and customer has a POWER9 EMS (running ESS 5000 or ESS 3200) then should migrate the containers from POWER8 EMS to POWER9 and discard the POWER8 EMS (single management node).
- If migrating from xCAT-based legacy offering to container based you must go from ESS 5.3.7.x.
- When you update ESS to 6.1.2.x for the first time, you must consider the implications of moving to MOFED 5.x. Review the following flash carefully for more information Mellanox OFED 5.x considerations in IBM ESS V6.1.2.x.
- IBM Spectrum Fusion, IBM Spectrum Scale Container Native, and IBM Spectrum Scale CSI utilize the GUI rest-api server for provisioning of storage to container applications. Persistent Volume (PV) provisioning will halt when the ESS GUI is shut down and remain halted for the duration of the ESS upgrade, until the GUI is restarted. Ensure that the OpenShift and Kubernetes administrators are aware of this impact before proceeding.
- For ESS 3500, you must keep 1.5 TB or more space free if future capacity MES is planned (performance to hybrid). Thus, it is recommended to not use all available space when you create a file system for

the performance model. The default allocation is 80% of available space when you use the **essrun filesystem** command (for x86 nodes).

Client nodes

Client nodes need to be at MOFED 4.9.x or higher and converted to **verbsRDMA** core libs after the ESS cluster is moved to 6.1.2.x or higher. Moving to **verbsRDMA** core libs is especially important if **verbsRDMA** is in use in the storage cluster.

Upgrade guidance



RED - NOT SUPPORTED
YELLOW - SUPPORTED NOT TESTED
GREEN - SUPPORTED

Note: Upgrades to ESS 6.1.2.x follow the N-2 rule. You can upgrade from ESS 6.1.2.x, 6.1.1.x (that is, 6.1.1.2) or 6.1.0.x.

Further legacy container migration guidance

You must migrate first to ESS 5.3.7.x before you upgrade to ESS 6.1.x.x (container version).

ESS 5.3.x.x upgrade guidance

- – You can upgrade to 5.3.7.x from 5.3.5.x (online) or 5.3.6.x (online).
  – For online upgrade you can jump one OS version and for offline upgrade you can jump two OS versions.

    Only exception is RHEL 7.7 to RHEL 7.9 upgrade. Because there is no RHEL 7.8.

    Online upgrade to RHEL 7.7 from RHEL 7.6 can be done.

    Upgrade to RHEL 7.7 from RHEL 7.5 must be done online.

ESS 6.1.x.x upgrade guidance

- It is recommended to convert from ESS 5.3.7.x to ESS 6.1.2.x and follow the normal N-X rules. To convert to ESS 6.1.2.x, use the following table (based on the RHEL 7.9 kernel):

Table 2. RHEL kernels

ESS	Kernel
6.1.2.4	3.10.0-1160.71.1.el7
6.1.2.3	3.10.0-1160.62.1.el7
6.1.2.2	3.10.0-1160.49.1.el7
5.3.7.6	3.10.0-1160.62.1.el7
5.3.7.5	3.10.0-1160.59.1.el

Table 2. RHEL kernels (continued)	
ESS	**Kernel**
5.3.7.4	3.10.0-1160.49.1.el7
5.3.7.3	3.10.0-1160.45.1.el7
5.3.7.2	3.10.0-1160.31.1
5.3.7.1	3.10.0-1160.24.1
5.3.7.0	3.10.0-1160.11.1.el7

An example of upgrade jump is as follows:

  - To upgrade to ESS 6.1.2.2, you can only upgrade from 5.3.7.4 or lower versions (that is, less than equal to 5.3.7.4).
  - To upgrade to ESS 6.1.2.3, you can only upgrade from 5.3.7.6 or lower versions.
- It is not recommended to upgrade from ESS 5.3.7.x to ESS 6.1.1.2 anymore. Upgrade directly to ESS 6.1.2.3 or ESS 6.1.2.4. If you are updating from ESS 6.1.1.2, upgrade to 6.1.2.3 or higher (do not upgrade to 6.1.2.2).
- For ESS 5.3.7.3, consider downgrading MOFED to MLNX_OFED_LINUX-4.9-3.1.5.3, and then convert to 6.1.2.3 or 6.1.2.4. This is to obtain full support for online upgrade when converting to RDMA core libs.
- When upgrading to 5.3.x.x, first upgrade to ESS 5.3.7.2 or ESS 5.3.7.3, and then upgrade to 6.1.2.3 or 6.1.2.4. This upgrade is to obtain full support for online upgrade when converting to RDMA core libs.
- You may need to modify the container unblock jumps from a specific 5.3.7.x level. Issue to the following command to upgrade the ESS level in the container:

```
vim /opt/ibm/ess/deploy/ansible/vars.yml
```

- Change (an example if you want to convert from ESS 5.3.7.1 or higher) LEGACY_SUPPORTED_VERSION: "5.3.7.3" to LEGACY_SUPPORTED_VERSION: "5.3.7.1".

For more information about the ESS 6.1.x.x upgrade, see IBM Spectrum Scale Alert: Mellanox OFED 5.x considerations in IBM ESS V6.1.2.x+.

Chapter 3. ESS common installation instructions

Note: You must convert to `mmvdisk` before using ESS Legacy.

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

Note: All version numbers, host names, IP addresses that are used in the following sections are examples.

These instructions are based on steps required for a POWER9 EMS. Important POWER8 notes are outlined where needed. The following build is used for example purposes.

```
ess_6.1.4.1_0919-18_dae_ppc64le.tar.xz
```

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) (The connection might be named 'campus').** 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 tool.
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 a supported ESS system with the p9 EMS node, add new ESS host names to `/etc/hosts` by using the same structure. For example, 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.

   Bridges are cleaned up automatically. However, if you want to clean up bridges manually, complete the following steps. An option is available to prevent cleanup if desired.

   Note: Typically, this is applicable only for upgrades.

   If podman is not installed, install it. For more information about the podman installation, see this step.

   ```
   ./ess_6.1.4.1_0919-18_dae_ppc64le --start-container
   ```

   A sample output is as follows:

   ```
   which: no podman in (/root/.local/bin:/root/bin:/opt/ibm/ess/
   tools/bin:/usr/lpp/mmfs/bin:/sbin:/bin:/usr/sbin:/usr/bin:/usr/X11R6/bin)warning:
   conmon-2.0.6-1.module+el8.1.1+5259+bcdd613a.ppc64le.rpm: Header V3 RSA/SHA256 Signature,
   key ID fd431d51: NOKEY
   [/usr/lib/tmpfiles.d/pmcollector.conf:1] Line references path below legacy
   directory /var/run/, updating /var/run/perfmon → /run/perfmon; please update the
   tmpfiles.d/ drop-in file accordingly.
   [/usr/lib/tmpfiles.d/pmsensors.conf:1] Line references path below legacy
   directory /var/run/, updating /var/run/perfmon → /run/perfmon; please update the
   tmpfiles.d/ drop-in file accordingly.
   [/usr/lib/tmpfiles.d/postgresql.conf:1] Line references path below legacy
   directory /var/run/, updating /var/run/postgresql → /run/postgresql; please update the
   tmpfiles.d/ drop-in file accordingly.
   -- [INFO] Podman and it's dependencies installed--
   -- [INFO] Code extraction completed successfully--
   -- [INFO] Python3 link found. Continuing--
   ```

   Note: If a node was reinstalled, podman will not be available. Thus, you can skip the cleanup.

   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.

i) List the currently configured bridges.

```
nmcli c
```

ii) 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
```

f. Bring up the interfaces.

```
nmcli c up fsp
nmcli c up mgmt
```

5. Do additional clean up (POWER8 legacy to container migration only).

- Check whether a POWER8 or POWER9 node runs the following check:

  POWER8

```
# lscpu | grep -i "Model name"
Model name:            POWER8E (raw), altivec supported
```

  POWER9

```
lscpu | grep -i "Model name"
Model name:           POWER9, altivec supported
```

- If you are using a POWER8 EMS and converting from the xCAT-based deployment to container, you must first stop and uninstall xCAT as follows.

```
systemctl stop xcatd
yum -y remove xCAT
```

- Make sure that the DHCP server is not longer running.

```
ps -ef | grep -i dhcp
```

6. Stop the GUI temporarily until upgrade or conversion from xCAT deployment to container is complete.

```
systemctl stop gpfsgui
```

When you are updating the EMS, shut down the GUI. Do not start the GUI until you finish upgrading the EMS. Because the GUI is shut down, any containers that access the ESS storage through the REST API cannot access the storage temporarily.

7. 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

xz --decompress ess_6.1.4.1_0919-18_dae_ppc64le.tar.xz
tar xvf ess_6.1.4.1_0919-18_dae_ppc64le.tar

ess_6.1.4.1_0919-18_dae_ppc64le
ess_6.1.4.1_0919-18_dae_ppc64le.sha256
```

8. Accept the license and install the accepted image.

```
./ess_6.1.4.1_0919-18_dme_ppc64le --start-container
```

During this step, you are first prompted to accept the license agreement. Press 1 to accept. You are then prompted to input answers to 3 questions before the installation starts (2 questions for ESS 3000).

- Confirm or set EMS FQDN.
- Provide the container short name.
- Provide a free IP address on the FSP subnet for the container FSP connection.

Example of contents of the extracted installation package:

```
├── 70-persistent-net-ems.rules
├── classes
│   ├── essmgr.py
│   ├── essmgr_yml.py
│   ├── _init_.py
│   └── _pycache_
│   ├── essmgr.cpython-36.pyc
│   ├── essmgr_yml.cpython-36.pyc
│   └── _init_.cpython-36.pyc
├── ess_6.1.4.1_0919-18_dme_xcat_ppc64le_binaries.iso
├── ess_6.1.4.1_0919-18_dme_xcat_ppc64le.tar
├── essmgr
├── essmgr_p8.yml
├── essmgr_p9.yml
├── essmgr.yml
├── essmkyml
├── logs
│   ├── essmgr.yml_2022-05-16_18-16-28
│   └── essmkyml_2022-05-16_18-16-28_log
├── podman_rh7_ppc64le.tgz
├── podman_rh8_ppc64le.tgz
├── python3_rh7_ppc64le.tgz
├── python3-site-packages_rh7_ppc64le.tgz
├── Release_note.ess_6.1.4.1_0919-18_dme_xcat_ppc64le.txt
├── rhels-7.9-server-extra.iso
├── rhels-7.9-server-ppc64le.iso
├── rhels-8.6-server-extra.iso
├── rhels-8.6-server-ppc64le.iso
└── rhels-8.6-server-x86_64.iso
```

In this step, you are prompted to 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. **essmkyml** is located in the extracted directory (example: `/home/deploy/ess5000_/home/deploy/ess_6.1.4.1_0919-18_dae_ppc64le.dir`).
- 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 preceding questions apply to ESS 3200, 3500, ESS 5000, and ESS Legacy on POWER9 EMS. If you are on the POWER8 EMS (ESS 3000 or ESS Legacy only), you are asked for the EMS hostname, container name, and FSP bridge IP.

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

```
ESS UNIFIED v6.1.4.1 CONTAINER root@cems0:/ #
```

9. Check and fix passwordless ssh in EMS and all nodes by using essutils.
10. Run the **essrun config load** command. This command determines the node information based on VPD and also exchange the SSH keys.

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

Note:

- Always include the EMS in this command along with all nodes of the same type in the building-blocks.
- 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.
- The **config load** command needs to be run with `-N` on all the nodes of the cluster to update the `hosts.yml` file for BMC. If config load is run on the selected nodes only (For example, only on essio nodes, then ems node), the `hosts.yml` file will be overwritten with latest nodes of config load.

After this command is run, you can use `-N` *NodeGroup* for future **essrun** steps (For example, `-N ess_ppc64le`). There are different node group names for ESS 3000 and ESS Legacy.

11. Run the **essrun config check** command. This command does a check of the various nodes looking for potential issues prior to upgrade. Review the output carefully and make changes as needed before proceeding.

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

Chapter 4. ESS new deployment instructions

Use these instructions if you are deploying a new cluster or a new file system.

Note: The POWER8 or POWER9 firmware is not automatically upgraded by the **essrun** automation. For information about manually upgrading the server firmware, see Appendix G, "Upgrading the POWER9 firmware," on page 79. You may use the **essinstallcheck** command to determine if a firmware upgrade is required after upgrading to ESS 6.1.x.x.

Before you start with these steps, you must complete the steps in Chapter 3, "ESS common installation instructions," on page 23.

The following steps are covered in this topic:

- Upgrading the EMS and I/O nodes, if required.
- Creating network bonds.
- Creating the cluster.
- Adding the EMS node to the cluster.
- Creating the file system.
- Configuring performance monitoring and starting the GUI.
- Setting up call home.
- Setting up time server.
- Final health checks.

Note: You can update by using the management node names (management) or after the **config load** is run, you can update by using a group of nodes. The groups are as follows:

- PPC64LE - ESS 5000 and ESS Legacy: `ess_ppc64le`
- x86_64 - ESS 3000: `ess_x86_64`

When the group is referenced in these instructions, `ess_ppc64le` is used as an example. If you are in an ESS 3000 environment, use `ess_x86_64`.

For the EMS node, you can use the group `ems`.

At this point, the user has already determined whether an upgrade is required. If the version initially found in `/home/deploy` on the EMS node is earlier than the latest available on IBM Fix Central, the latest version should be already downloaded and deployed according to Chapter 3, "ESS common installation instructions," on page 23.

1. If an upgrade is required, upgrade the EMS node.

```
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 update --offline
exit
systemctl reboot
```

Navigate back to ESS 6.1.4.x extracted directory and run the following commands:

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

- You cannot upgrade a POWER8 EMS currently running ESS Legacy code (5.3.x with xCAT control) from an ESS 3000 container. If xCAT is installed on the host, you must first uninstall it and clean up any dependencies before attempting an EMS upgrade from the container. If ESS Legacy deployment is needed, do not remove xCAT and deploy the ESS Legacy. Otherwise, remove xCAT and use the container to upgrade EMS and I/O nodes.

  Note: To check which nodes belong to certain nodeclasses, issue the following command:

  ```
  lsdef -t group for ansible-inventory -i /vpd/Inventory --graph
  ```

- When you are updating the EMS, shut down the GUI. Do not start the GUI until you finish upgrading the EMS. Because the GUI is shut down, any containers that access the ESS storage through the REST API cannot access the storage temporarily.

2. If required, update the I/O nodes.

   ```
   essrun -N <nodeclasses> update --offline
   ```

   Where node classes are as follows:

   - legacy: gss_ppc64le
   - 5000: ess_ppc64le
   - 3000: ess_x86_64
   - 3200: ess3200_x86_64
   - 3500: ess3500_x86_64
   - Protocol Power9: ces_ppc64le

   Update all node classes for all building blocks except EMS.

3. If required, update I/O and protocol nodes at the same time.

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

4. Create network bonds.

   ```
   essrun -N essio1,essio2 network --suffix=-hs
   essrun -N ems1 network --suffix=-hs
   ```

   You can create bonds on I/O and EMS nodes at the same time:

   ```
   essrun -N essio1,essio2,em1 network --suffix=-hs
   ```

5. Run the network test.

   This test uses **nsdperf** to determine whether 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
   ```

   Note: After you SSH to an I/O node, you exit from the container

   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 to create a cluster.

6. Create the cluster.

   ```
   essrun -N <nodeclasses> cluster --suffix=-hs
   ```

7. Add the EMS node to the cluster.

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

8. Create the file system.

```
essrun -N ess3500_x86_64 filesystem --name fs1 --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 by using less space. For example:

```
essrun -N ESSIONodeInCluster(management hostname) filesystem --suffix=-hs --size 80%
```

  For more options such as blocksize, filesystem size, or RAID code, see the **essrun** command in the ESS Command Reference.
- 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.
- If you are updating ESS 3000, ESS 3200, and ESS 3500 the default set-size is 80% and it must not be increased. If you are updating ESS 5000 and ESS Legacy, the default set-size is 100%. For additional options, see *essrun command*. The default block size for PPC64LE is 16M whereas for ESS 3000 it is 4M.
- If you are deploying protocol nodes, make sure that you leave space for CES shared root file system. Adjust the set-size slightly lower when you are creating this required file system for protocol nodes.
- For ESS 3500, you must keep 1.5 TB or more space free if future capacity MES is planned (performance to hybrid). Thus, it is recommended to not use all available space when you create a file system for the performance model. The default allocation is 80% of available space when you use the **essrun filesystem** command (for x86 nodes).

Final setup instructions

1. Log in to each node and run following command.

```
essinstallcheck -N localhost
```

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

   Note: For ESS 5000, a check is added that flags whether the WCE bit is enabled on any drive. If the WCE bit is enabled, refer the published flash for the recommended action.

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

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

3. 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 H, “How to set up chronyd (time server) in non-ESS nodes,” on page 81 to perform these tasks before proceeding.

4. Set up call home. For more information, see Appendix D, “Configuring call home in ESS 5000, ESS 3000, ESS 3200, ESS 3500, and ESS Legacy,” on page 49.

   The supported call home configurations are:

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

- Drive call home

5. Set up the GUI, configure performance monitoring, add a GUI user, set a GUI user password, and start the gpfsgui.
   a. To configure GUI hardware monitoring from the container and configure performance monitoring, issue the following command:

      ```
      essrun -N ems1,essio1,essio2 gui --configure
      ```

6. Refer to Appendix M, “Client node tuning recommendations,” on page 99.

Chapter 5. ESS upgrade instructions

Note: The POWER8 or POWER9 firmware is not automatically upgraded by the **essrun** automation. For information about manually upgrading the server firmware, see Appendix G, "Upgrading the POWER9 firmware," on page 79. You may use the **essinstallcheck** command to determine if a firmware upgrade is required after upgrading to ESS 6.1.x.x.

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
mmnetverify -N all
```

You can also run the **essrun healthcheck** command instead, from inside the container.

```
essrun -N NodeList healthcheck
```

Upgrade can be done by using the following methods

- Offline upgrade: This method requires a given node or nodes to have GPFS shut down before beginning. This method is faster than online update, in which nodes are upgraded in parallel including firmware, but the system is typically taken down for a period of time.
- Online upgrade: This method allows the cluster to stay fully available and the code is typically updated one node per building-block in parallel.

Note:

- The EMS node and protocol node upgrades are available only in the offline mode.
- Where NodeClass is an ESS 3000, ESS 5000, or ESS Legacy node class. For more information, see the *mmlsnodeclass command* in *IBM Spectrum Scale: Command and Programming Reference*.
- On protocol nodes, IBM Spectrum Scale is not automatically upgraded by the **essrun** automation. It must be upgraded by using the installation toolkit or by using the **rpm** command. For more information, see Upgrading IBM Spectrum Scale protocol nodes and Upgrading IBM Spectrum Scale components with installation kit in IBM Spectrum Scale documentation.

Online upgrade assumptions (I/O nodes only):

- The cluster is created with EMS, one or more ESS 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 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 nodes are active by running the following command from one of the cluster nodes:

```
mmgetstate -N NodeClass
```

Where *NodeClass* is your ESS 3000, ESS 5000, or ESS Legacy node class. For more information, see *mmlsnodeclass command*.

Offline upgrade assumptions (EMS or protocol nodes only):

- You assume the risks of potential quorum loss.
- The GPFS GUI and collector must be down.

Note:

- Before upgrading the protocol nodes, you might need to stop services on a specific protocol node before the upgrade starts. Because during a protocol node upgrade, the installation toolkit does not unmount a file system.
- IBM Spectrum Fusion, IBM Spectrum Scale Container Native, and IBM Spectrum Scale CSI utilize the GUI rest-api server for provisioning of storage to container applications. Persistent Volume (PV) provisioning will halt when the ESS GUI is shut down and remain halted for the duration of the ESS upgrade, until the GUI is restarted. Ensure that the OpenShift and Kubernetes administrators are aware of this impact before proceeding.

1. Complete the steps in Chapter 3, "ESS common installation instructions," on page 23. Make sure that you add the protocol nodes to the configuration load if you are planning to upgrade protocol nodes.
2. Update the EMS node first.

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

   If kernel version changed during the update, you are prompted to exit the container, reboot, rerun the container, and rerun the update command.

   ```
   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
   ```

   Note:

   - You cannot upgrade a POWER8 EMS currently running ESS Legacy code (5.3.x with xCAT control) from an ESS 3000 container. If xCAT is installed on the host, you must first uninstall it and cleanup any dependencies before attempting an EMS upgrade from the container. Do not remove xCAT if legacy deployment is not needed, typically only if you are moving to ESS Legacy 6.1.0.x container. If you are still using an ESS Legacy deployment (5.3.x), update the EMS by using the upgrade instructions outlined in *ESS 5.3.x Quick Deployment Guide*.
   - When you are updating the EMS, shut down the GUI. Do not start the GUI until you finish upgrading the EMS. Because the GUI is shut down, any containers that access the ESS storage through the REST API cannot access the storage temporarily.
3. Update the protocol nodes.

   ```
   essrun -N prt01,prt02 update --offline
   ```
4. Run installation check on each node type by logging in to EMS node and protocol nodes.

   ```
   essinstallcheck
   ```
5. Do ESS I/O nodes offline update as follows.

   When you upgrade ESS 5000 nodes, you cannot upgrade the firmware because of a restriction. Add the `--no-fw-update` option to prevent firmware upgrades. For example,

   ```
   essrun -N ess5kio1,ess5kio2 update --no-fw-update
   ```

   Add the `--offline` option, when you attempt an offline only upgrade.

   Important: For doing an offline update, GPFS must be down in the ESS 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.

   If you want to do an online update of I/O nodes, refer to "Update ESS I/O nodes online" on page 35.

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

```
essrun -N 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
```

These command examples show ESS 5000 node and node classes, but you can use these commands with any of the supported ESS node types, such as ESS 3200 and ESS 3500.

After offline update is done, proceed to starting GPFS on the nodes.

6. Run installation check on each node from outside the container.

```
essinstallcheck
```

Note: For ESS 5000, a check is added that flags whether the WCE bit is enabled on any drive. If the WCE bit is enabled, refer the published flash for the recommended action.

7. Start GPFS on all nodes.

```
mmstartup -N NodeList | NodeGroup
```

Wait for a few minutes and then check the state.

```
mmgetstate -N NodeList | NodeGroup
mmgetstate -s
```

Note: If any protocol nodes are updated, ensure that you restart CES services on those nodes.

Update ESS I/O nodes online

ESS 3500 does not update the firmware automatically during the deployment. You must issue the **mmchfirmware** command to update the firmware. When you upgrade ESS 5000 nodes, you cannot upgrade the firmware because of a restriction.

1. Update the I/O nodes online 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 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 I/O nodes.

```
essrun -N ess_ppc64le update
```

- Update by using the individual nodes.

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

Note: Consider using the `--serial` option for online upgrade. This will allow you to perform an online update one node at a time (or to the required level). For example:

```
essrun -N essio1,essio2,essio3,essio4 update --serial 1
```

The `--serial 1` option is the default option, if the `--serial 1` option is not specified.

The command performs an online update of two building blocks but one node at a time. See "Serial option for online upgrade" on page 36 for details.

2. Run installation check on each updated node.

```
essinstallcheck
```

Note: For ESS 5000, a check is added that flags whether the WCE bit is enabled on any drive. If the WCE bit is enabled, refer the published flash for the recommended action.

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

```
mmchconfig autoload=yes
```

Final steps for online and offline upgrade

Do the following final steps after the online or the offline update is complete.

1. Start the performance monitoring collector on the EMS node.

```
systemctl start pmcollector
```

2. Start the performance monitoring sensors on each node.

```
mmdsh -N NodeList | NodeGroup "systemctl restart pmsensors"
```

3. Enable **gpfsgui** start on reboot.

```
systemctl enable gpfsgui
```

4. Start the GUI on the EMS node.

```
systemctl start gpfsgui
```

5. Run manual health checks.

```
gnrhealthcheck
mmhealth node show -a
mmnetverify -N all
```

6. Start the ESA GUI (call home), if applicable.

For POWER9 nodes, refer to Appendix G, “Upgrading the POWER9 firmware,” on page 79 to update the system firmware.

Serial option for online upgrade

A new `--serial` option is added to the **essrun** command for updates.

With the **essrun** command `--serial` option, you can specify a number or the word `all`. The specified number is the number of nodes that need to be updated at the same time. If it is an online update, the **essrun** command iterates on the specified number of nodes until it finishes with the first set of nodes. Then, it moves to the next set and iterates over the specified number of nodes, and so on, until the specified nodes are updated.

For example, in a *NodeList* (**essrun -N** *NodeList*) where the values are ['1a', '1b', '2a', '2b', '3a', '3b', '4a', '4b', '5a'] and the number specified with `--serial` is 2, the update is done in the following order:

First round of update
1a, 1b

Second round of update
2a, 2b

Third round of update
3a, 3b

Fourth round of update
4a, 4b

Fifth round of update
5a

If the command is **essrun -N essio1,essio2,essio3,essio4 update --serial 1** then the result is as follows:

1. Update essio1 (canister A BB1)
2. Update essio3 (canister A BB2)
3. Update essio2 (canister B BB1)
4. Update essio4 (canister B BB2)

If the command is **essrun -N essio1,essio2,essio3,essio4 update --serial 2** then the result is as follows:

1. Update essio1 and essio3 (canister A BB1 and canister A BB2)
2. Update essio2 and essio4 (canister B BB1 and canister B BB2)

Appendix A. ESS known issues

Known issues in ESS

For information about ESS 5.3.7.x known issues, see *Known issues* in ESS 5.3.7.x Quick Deployment Guide.

The following table describes the known issues in IBM Elastic Storage System (ESS) and how to resolve these issues.

Issue	Resolution or action

Running **ess_ssr_setup** hangs if recovery group descriptors exist. This will simply hang and does not indicate that descriptors were found.
An SSR might see the following output:
`Do you want to continue and perform changes and tests in this node? (y/n): y`
` 2022-08-19 09:38:44,234 INFO: Going to set the root user password of this node to the password typed before`
` 2022-08-19 09:38:44,433 INFO: Run 'Root_password_set' completed successfully`
` 2022-08-19 09:38:44,602 INFO: Run 'Passwordless root SSH localhost' completed successfully`
` 2022-08-19 09:38:44,602 INFO: Going to perform storage tests on this node`
` 2022-08-19 09:38:44,602 INFO: Going to run 'Quick storage configuration check'`
` 2022-08-19 09:38:45,484 INFO: Run 'Quick storage configuration check' completed successfully`
` 2022-08-19 09:38:45,484 INFO: Going to run 'Check enclosure cabling and paths to disks'`
` 2022-08-19 09:39:38,432 INFO: Run 'Check enclosure cabling and paths to disks' completed successfully`
` 2022-08-19 09:39:38,432 INFO: Going to run 'Check disks for IO operations'`
Product
• ESS 5000
• ESS 3500 (4u102)
	The SSR should contact service for help.

When the **essinstallcheck** command is run, an error might occur. Customers might face this issue when they run the **essrun healthcheck**command (that runs the **essinstallcheck** command) with multiple nodes or group. The **mmvdisk** locks other nodes when querying is the cause of this error. **Product** • ESS 3000 • ESS 3200 • ESS 5000 • ESS 3500 • ESS 3500 (4u102)	• Check Ansible logs before you run the **mmvdisk** command (in **essinstallcheck**) and retry when ready. • In the field, run the **healthcheck** command on individual nodes if the error is seen.

Issue	Resolution or action
The **essrun** update might hang with 'waiting for free locks'. The **mmapply policy** causes locks. `Friday 12 August 2022  21:02:23 +0000 (0:00:00.536)  0:34:26.503 ********* FAILED - RETRYING: Waiting for free Locks (100 retries left). FAILED - RETRYING: Waiting for free Locks (99 retries left).` **Product** • ESS 3000 • ESS 3200 • ESS 5000 • ESS 3500 • ESS 3500 (4u102)	1. Check if policy is being applied and wait until it finishes to run update. 2. Run the **mmcommon showlocks** command to check what is causing the lock. For more information about locks, see IBM Spectrum Scale Administration Guide.
GUI: wizard setup is not allowed to move past Rack Locations. Moving backward through GUI and then trying to move forward is blocked. **Product** • ESS 3000 • ESS 3200 • ESS 5000 • ESS 3500 • ESS 3500 (4u102)	1. Back up to a Location information and re-enter the data. 2. Clean up the GUI DB and start the wizard setup again.
When creating additional file systems in a tiered storage environment you might encounter a MIGRATION callback error. `mmaddcallback: Callback identifier "MIGRATION" already exists or was specified multiple times.` If a callback exists, file system creation will fail. **Product** • ESS 3000 • ESS 3200 • ESS 5000 • ESS 3500 • ESS 3500 (4u102)	Delete the callback and create the file system again.
Many call home events for temperature sensor causes canister1_inlet_id1 failure. `TS008179389 2022-08-05 02:16:50 New Case Opened 78E4007:canister:78E4007A/ 1:canister1_inlet_id1:Temperature sensor canister1_inlet_id1 is failed TS008179399 2022-08-05 03:14:19 New Case Opened 78E4007:canister:78E4007A/ 1:canister1_inlet_id1:Temperature sensor canister1_inlet_id1 is failed TS008179432 2022-08-05 07:15:33`	• If this issue occurs in field, turn up the AC to cool ambient temperature. • If this issue occurs, customers/CE must lower ambient lab temperature.

Issue	Resolution or action
`New Case Opened  78E4007:canister:78E4007B/ 0:canister2_inlet_id0:Temperature sensor canister2_inlet_id0 is failed` **Product** • ESS 3500	
Both ESS 3500 power supplies blink red-orange LED once every second during the I/O load. **Product** • ESS 3500	If this problem occurs, reseat the power supply.
The **essrun** ONLINE update failed on the **mmchfirmware -N localhost --type drive**. **Product** • ESS 5000	SSR and customers must manually run the **mmchfirmware** command after the deployment completes.

Appendix B. Adding additional nodes or building block(s)

The following example procedure describes how to add IBM Elastic Storage System nodes to an existing IBM Elastic Storage System environment thus extending the existing file system.

The example is based on the following assumptions:

- POWER9 EMS and one or more IBM Elastic Storage System nodes in the cluster (three node quorum or more)
- Adding an IBM Elastic Storage System server (two canisters) without code upgrade in the existing EMS
- 4M block size, 8+2p raid code, 80% set-size (defaults)

Refer to the following steps to add new IBM Elastic Storage System nodes to an existing cluster. If needed, refer to the preceding sections for the details of the following steps.

1. Add a new building block or nodes and I/O node names to existing EMS `/etc/hosts` including management and high-speed connection.
2. Reinstall the deployed code image if it is not in EMS and run the container clean to bring the new `/etc/hosts` in the container.

   ```
   # essrun -N ems,essio1,essio2,essio3,essio4 config load -p ibmesscluster
   ```

3. Run config load against the existing nodes and the new nodes.

   ```
   # essrun -N ems,essio1,essio2,essio3,essio4 config check -p ibmesscluster
   ```

4. Create network bond connections for the new nodes.

   ```
   # essrun -N essio3,essio4 network --suffix=-hs
   ```

5. Outside the container, run the **essnettest** command against the new nodes.

   Run this command from one of the nodes in the cluster and use this command to test the health of the high-speed network connections. For more information, see the **essnettest** command help.

6. Add new nodes to the existing cluster by submitting the following command from the container:

   ```
   # essrun -N ems cluster --add-nodes essio3,essio4 --suffix=-hs
   ```

   Note: Add only the same nodetype building blocks at a time. For example:

   ```
   # essrun -N ems1 cluster --suffix=-hs --add-nodes ess3k1a,ess3k1b
   ```

   Do not add node type building blocks as follows:

   ```
   # essrun -N ems1 cluster --suffix=-hs --add-nodes ess3k1a,ess3k1b,ess5k1a,ess5k1b
   ```

7. Create node class and recovery groups by submitting the following command from the container:

   ```
   # essrun -N essio3,essio4 filesystem --rg-only --suffix=-hs
   ```

8. Get the list of recovery groups by submitting the following command outside the container:

   ```
   mmvdisk rg list
   ```

9. Define the vdisk set.

   Note: For ESS 3500, you must keep 1.5 TB or more space free if future capacity MES is planned (performance to hybrid). Thus, it is recommended to not use all available space when you create a file

system for the performance model. The default allocation is 80% of available space when you use the **essrun filesystem** command (for x86 nodes).

a. Use the **ssh** command to connect to one of the new nodes.

b. Define the vdisk set by submitting the following command:

```
mmvdisk vs define --vs vs_fs1_essio3_hs_essio4_hs --rg ess3500_essio3_hs_essio4_hs
--code 8+2p --bs 4M --ss 80% --nsd-usage dataAndMetadata --sp system
```

Note: Assume that the existing file system has the attributes that are mentioned in the command. For vdisk set name, use a unique name similar to the existing IBM Elastic Storage System vdisk set.

For more information, see **mmvdisk** online command reference.

10. Add the vdisk set to an existing file system by submitting the following command:

```
mmvdisk fs add --file-system fs1 --vdisk-set vs_fs1_essio3_hs_essio4_hs
```

Note: Assume that the original file system name is *fs1*.

11. Update the component database on the EMS by submitting the following command from outside of the container:

```
mmaddcompspec default --replace
```

12. Restripe the file system by submitting the following command:

```
mmrestripefs fs1 -b
```

13. Add new nodes to GUI monitoring.

```
mmchnode --perfmon -N essio5-hs,essio6-hs
```

Adding nodes to GUI

1. From the container, issue the following command to configure and start the performance monitoring sensors.

```
cd /home/deploy/ess_6.1.4.1_0919-18_dme_ppc64le.dir
./essmgr  -r
essrun -N essio3,essio4 gui --add-to-hosts
```

Note: Ensure that the passwordless ssh is configured from new nodes to EMS.

2. Check whether GUI web page shows new ESS nodes.

Proceed to add the new nodes as sensors, re-configure call home, and rerun the GUI wizard to identify the new nodes.

Real-world ESS building block addition notes

These notes are the raw instructions that came from a recent IBM Elastic Storage System building-block add operation. It is important to add this information for users who need additional help/context.

Assumptions/examples:

- *essio3* and *essio4* are the new nodes being added (*essio3-hs*, *essio4-hs*).
- *cems0* is the container name.
- *essio1* is a node already in the cluster.
- *fs1* is the existing filesystem name.

To add new canisters to an existing cluster, refer to the following steps.

1. Add new IBM Elastic Storage System nodes, including management and high-speed connections to `/etc/hosts` on the EMS node.
2. Reinstall the image and run the container clean to bring the new `/etc/hosts` inside the container. Only need to reinstall/install a new image if the version on the new nodes is down-level.
3. Run config load against the existing nodes and the new nodes. Then do an offline update of both new canisters by using the following command:

   ```
   run essrun -N essio3,essio4 update --offline
   ```

4. To create network bond connections for the new nodes, issue the following command:

   ```
   essrun -N essio3,essio4 network --suffix=-hs
   ```

5. Run **essnettest** against the new nodes, **ssh** to *essio3* and run the following command:

   ```
   ESSENV=TEST essnettest -N essio3,essio4 --suffix=-hs
   ```

   Run this command from one of the nodes in the cluster and use this command to test the health of the high-speed network connections. For more information, see the **essnettest** command help.
6. Add new nodes to the existing cluster by submitting the following command from the container:

   ```
   root@cems0:/ # essrun -N essio1 cluster --add-nodes essio3,essio4 --suffix=-hs
   ```

   Note: The *essio1* node is an existing node in the cluster.
7. Create node class and recovery groups by submitting the following command from the container:

   ```
   root@cems0:/ # essrun -N essio3,essio4 filesystem --rg-only --suffix=-hs
   ```

8. Get the list of recovery groups by submitting the following command (from one of the new nodes (i.e. *essio3*):

   ```
   mmvdisk rg list
   ```

9. Define the vdisk set.
   a) Use the **ssh** command to connect to one of the new nodes.

      ```
      cd /home/deploy/ess_6.1.4.1_0919-18_dae_ppc64le.dir
      ./essmgr  -r
      essrun -N essio3,essio4 gui --add-to-hosts
      ```

      Note: Ensure that the passwordless ssh is configured from new nodes to EMS.

      Note: Assume that the existing file system has the attributes that are mentioned in the command.

      For vdisk set name, use a unique name similar to the existing IBM Elastic Storage System vdisk set. Verify existing disk set names by listing vdisk sets.

      ```
      mmvdisk vs list --vs all
      ```

   b) Define the vdisk set by submitting the following command (it is an example and may vary in different environment):

      For more information, see **mmvdisk** online command reference.
10. Create the vdisk set by submitting the following command:

    ```
    mmvdisk vs create --vs vs_fs2
    ```

11. Add the vdisk set to an existing file system by submitting the following command:

    ```
    mmvdisk fs add --file-system fs1 --vdisk-set vs_fs2
    ```

    Note: Assume that the original file system name is *fs1*.

12. Update the component database on the EMS by submitting the following command from outside of the container:

```
mmaddcompspec default --replace
```

13. Restripe the file system by submitting the following command:

```
cd /home/deploy/ess_6.1.4.1_0919-18_dae_ppc64le.dir
./essmgr  -r
essrun -N essio3,essio4 gui --add-to-hosts
```

Note: Ensure that the passwordless ssh is configured from new nodes to EMS.

After you complete the procedure, do the following steps:

1. Reconfigure call home, for more information, see Configuring call home.
2. Rerun the GUI wizard to identify the new nodes.

   From a web browser, login to the EMS over the management network and select **Edit Rack Components** to rerun the wizard discovery.

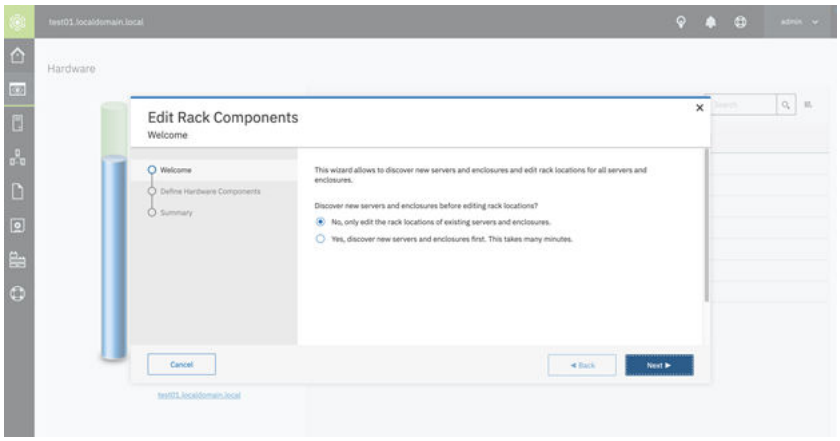


Figure 5. Showing ESS GUI wizard

Appendix C. Cleaning up the container environment

You cannot run multiple containers at the same time. To start a new container, you need to stop any containers that are currently running.

Refer to the following manual and automated cleanup steps that you can run to start in a known good state (POWER9 EMS).

- To stop any existing containers, submit the commands:

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

- To remove any existing containers, submit the following commands:

```
podman ps -a
podman rm ContainerName -f
```

- To remove any existing images, submit the following commands:

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

- To clean up network bridges, submit the following commands:

```
nmcli c

nmcli c del bridge-slave-mgmt
nmcli c del bridge-slave-fsp

nmcli c del mgmt_bridge
nmcli c del fsp_bridge
```

- To bring up management and FSP interfaces, submit the following commands:

```
ifup mgmt_bridge
ifup fsp_bridge
```

Note: An IP address must be set on the `mgmt` (management) and `fsp` (FSP) interfaces.

Appendix D. Configuring call home in ESS 5000, ESS 3000, ESS 3200, ESS 3500, and ESS Legacy

In ESS 5000, ESS 3000, ESS 3200, and ESS Legacy systems, ESS version 6.1.1.x can generate call home events when a drive in an attached enclosure needs to be replaced. ESS 5000, ESS 3000, ESS 3200, and ESS Legacy can also generate call home events for other hardware-related events in the I/O server nodes, protocol nodes, and client nodes that need service.

ESS 5000 and ESS Legacy hardware events rely on POWER system OPAL logs. ESS 3000 and ESS 3200 hardware events rely on **mmcallhome** and **mmhealth** commands.

ESS version 6.1.x automatically opens an IBM Service Request with service data, such as the location and field replaceable unit (FRU) number to carry out the service task.

Disk call home for ESS 5000, ESS 3000, ESS 3200, ESS 3500, and ESS Legacy

The IBM Spectrum Scale RAID pdisk is an abstraction of a physical disk. A pdisk corresponds to exactly one physical disk, and belongs to exactly one de-clustered array within exactly one recovery group.

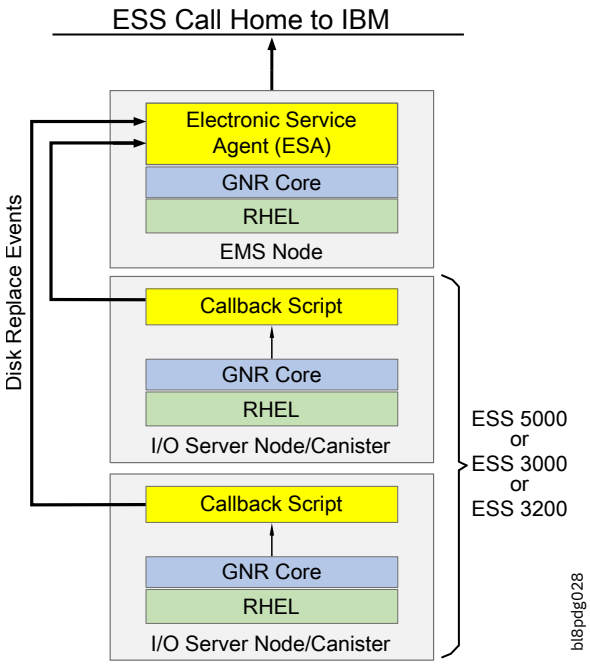


Figure 6. ESS Call Home Block Diagram

From ESS 3500, the unified call home is used to create service requests. Although ESA is deprecated and will be removed, it is supported as a backup for the next releases.

If a cluster has ESS 3500 nodes, EMS nodes, and at least one ESS 3500 node, the unified call home is used to create service tickets. For older nodes without any ESS 3500 nodes, ESA is used to create service tickets.

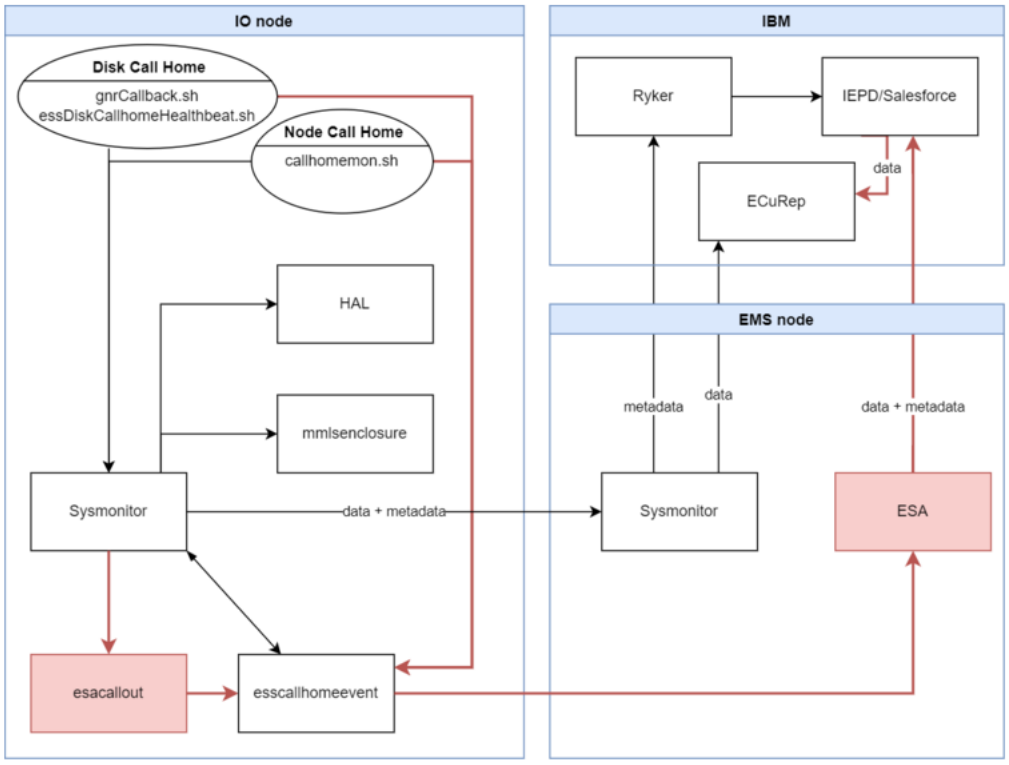


Figure 7. ESS unified call home

The attributes of a pdisk includes the following:

- The state of the pdisk
- The disk's unique worldwide name (WWN)
- The disk's field replaceable unit (FRU) code
- The disk's physical location code

When the pdisk state is ok, the `pdisk` is healthy and functioning normally. When the pdisk is in a `diagnosing` state, the IBM Spectrum Scale RAID disk hospital is performing a diagnosis task after an error has occurred.

The disk hospital is a key feature of the IBM Spectrum Scale RAID that asynchronously diagnoses errors and faults in the storage subsystem. When the pdisk is in a `missing` state, it indicates that the IBM Spectrum Scale RAID is unable to communicate with a disk. If a missing disk becomes reconnected and functions properly, its state changes back to ok. For a complete list of pdisk states and further information on pdisk configuration and administration, see IBM Spectrum Scale RAID Administration.

Any pdisk that is in the dead, `missing`, `failing` or `slow` state is known as a non-functioning pdisk.

When the disk hospital concludes that a disk is no longer operating effectively and the number of non-functioning pdisks reaches or exceeds the replacement threshold of their de-clustered array, the disk hospital adds the replace flag to the pdisk state. The replace flag indicates the physical disk corresponding to the pdisk that must be replaced as soon as possible. When the pdisk state becomes `replace`, the drive replacement callback script is run.

The callback script communicates with the ESA or the unified call home. The ESA is installed in the ESS as part of the Management Server (EMS). The unified call home is installed on all the nodes.

The EMS node initiates a call home task. The ESA or the unified call home is responsible for automatically opening a Service Request (PMR) with IBM support, and managing the end-to-end life cycle of the problem.

Installing the IBM Electronic Service Agent

IBM Electronic Service Agent (ESA) for PowerLinux version 4.5.5.1 or later can monitor the ESS systems. ESA is pre-installed on the EMS node when the EMS node is shipped.

The `esagent` rpm is also provided in the ESS 5000, ESS 3000, ESS 3200, ESS Legacy `binaries.iso` file in the container package. The ISO is mounted when **essmgr** is run to start the container. When mounted, the rpm file can be found at the following location:

- ESS 5000, ESS 3000, and ESS 3200: /install/ess/otherpkgs/rhels8/ppc64le/ess/
- ESS Legacy: /install/ess/otherpkgs/rhels7/ppc64le/ess/

1. Verify if `esagent` RPM is already installed on EMS by issuing the following command:

   ```
   # rpm -qa | grep esagent
   ```

   A sample output is as follows:

   ```
   yum install esagent.pLinux-4.5.7-0.noarch.rpm
   ```

2. If it is not installed, issue the following command:

   ```
   # cd /install/ess/otherpkgs/rhels8/ppc64le/ess/
   # yum install esagent.pLinux-4.5.7-0.noarch.rpm
   ```

Configuring call home on ESS systems

To configure call home on ESS systems, the first step is to activate and configure Electronic Service Agent (ESA).

After ESA is configured with the customer details, ESS systems can be configured for call home event generation by using the Electronic Service Agent (ESA). This is done by using the **esscallhomeconf** command.

Note: ESA is not activated by default. When you run **esscallhomeconf** for the first time, before the activation of ESA, you will get a message such as the following message:

```
[root@ems1 tmp]# esscallhomeconf -E ems1 –show
[E] IBM Electronic Service Agent (ESA) is not activated.
[I] Activate ESA by running: /opt/ibm/esa/bin/activator -C.
[I] Alternatively use --esa-config switch and provide all customer details to do ESA activation
from here only.
[I] See --esa-config switch for further CLI activation of ESA
Exiting...
```

After ESA is successfully configured, you need to configure ESS systems to generate call home events.

Entities or systems that can generate events are called endpoints. The EMS, I/O server nodes, and attached enclosures can be endpoints in ESS. Servers and enclosure endpoints can generate events. Server can generate hardware events which could be CPU, DIMM, OS Disk, etc. Typically, these events are also logged in the OPAL log.

In ESS, ESA is only installed on the EMS node, and it automatically discovers the EMS as `PrimarySystem`. The EMS node and I/O server nodes must be registered to ESA as endpoints.

The **esscallhomeconf** command is used to perform the registration task. The command also registers enclosures attached to the I/O servers by default.

Software call home can also be registered based on the customer information given while configuring the ESA agent. A software call home group `auto` is configured by default and the EMS node acts as the software call home server. The weekly and daily software call home data collection configuration is also activated by default. The software call home uses the ESA network connection settings to upload the data to IBM. The ESA agent network setup must be complete and working for the software call home to work.

Activate and configure ESA and then configure call home as follows.

Configuration of ESA or unified call home

You can configure ESA or unified call home by using the following **esscallhomeconf** command. For more information, see ESS CLI **esscallhomeconf**.

Configuration of ESA using esscallhomeconf

The **esscallhomeconf** command has a switch called **--esa-config**. With the introduction of **--esa-config**, ESA configuration can be done by using the CLI. The **esscallhomeconf** command used with the **--esa-config** switch not only activates and configures ESA with the required customer information, but it also configures ESS systems to generate call home events. This command requires customer information such as customer name, email ID, server location, etc.

The usage information of **esscallhomeconf** is as follows.

```
usage: esscallhomeconf [ -h | --help ] -E ESA-AGENT [ --prefix  PREFIX] [ --suffix SUFFIX ]  {
                [--verbose ] [ --esa-hostname-fqdn ESA_HOSTNAME_FQDN ]
                [--stop-auto-event-report] [ -N NODE-LIST ] [ --show ]
                [--register {node,all } ] [ --icn ICN]
                [--serial SOLN-SERIAL ] [ --model SOLN-MODEL ]
                [--proxy-ip PROXY-HOSTNAME ] [ --proxy-port PROXY-PORT ]
                [--proxy-userid PROXY-USERNAME ] [ --proxy-password PROXY-PASSWORD ]
                [--esa-config] [-m CUSTOMER_INFO_M]
                [-u CUSTOMER_INFO_U] [-n CUSTOMER_INFO_N]
                [-e CUSTOMER_INFO_E] [-t CUSTOMER_INFO_T]
                [-f CUSTOMER_INFO_F] [-j CUSTOMER_INFO_J]
                [-k CUSTOMER_INFO_K] [-g CUSTOMER_INFO_G]
                [-a CUSTOMER_INFO_A] [-Z CUSTOMER_INFO_Z]
                [-y CUSTOMER_INFO_Y] [-r CUSTOMER_INFO_R]
                [-b CUSTOMER_INFO_B] [-s CUSTOMER_INFO_S]
                [-i CUSTOMER_INFO_I] [-p CUSTOMER_INFO_P]
                [-w] [-Y]

optional arguments:
  -h, --help            show this help message and exit
  -E ESA-AGENT          Provide nodename for esa agent node
  --prefix PREFIX       Provide hostname prefix. Use = between --prefix and
                        value if the value starts with -.
  --suffix SUFFIX       Provide hostname suffix. Use = between --suffix and
                        value if the value starts with -.
  --verbose             Provide verbose output
  --esa-hostname-fqdn ESA_HOSTNAME_FQDN
                        Fully qualified domain name of ESA server for
                        certificate validation.
  --stop-auto-event-report
                        Stop report of automatic event to ESA in case of any
                        hardware call home event reported to system.
  -N NODE-LIST          Provide a list of nodes to configure.
  --show                Show call home configuration details.
  --register {node,all}
                        Register endpoints(nodes, enclosure or all) with ESA.
                        hardware callhome
  --icn ICN             Provide IBM Customer Number for Software callhome.
  --serial SOLN-SERIAL  Provide ESS solution serial number.
  --model SOLN-MODEL    Provide ESS model. Applicable only for BE (ppc64)
                        models.
  --proxy-ip PROXY-HOSTNAME
                        Provides the IP address or the hostname for the proxy configuration.
  --proxy-port PROXY-PORT
                        Provides the port number for the proxy configuration.
  --proxy-userid PROXY-USERNAME
                        Provides the user ID for the proxy configuration.
  --proxy-password PROXY-PASSWORD
                        Provides the password for the proxy configuration.

  --esa-config          Provide info for configuration of ESA via CLI.
  -m ESA_CONFIG_M       name of organization that owns or is responsible for
                        this system
  -u ESA_CONFIG_U       country or region where the system is located
  -n ESA_CONFIG_N       name of the primary person in your organization who
is responsible for this system
  -e ESA_CONFIG_E       email address for the primary contact person (e.g.
                        myuserid@mycompany.com)
  -t ESA_CONFIG_T       telephone number where the primary contact person can be reached
  -f ESA_CONFIG_F    secondary person in your organization who is
                     responsible for this system
  -j ESA_CONFIG_J    secondary person email address (e.g.
```

```
                      myuserid@mycompany.com)
  -k ESA_CONFIG_K     secondary person telephone number where the person can
                      be reached
  -g ESA_CONFIG_G     country or region of the contact person
  -a ESA_CONFIG_A     state or province where the system is located
  -z ESA_CONFIG_Z     postal code where the system is located
  -y ESA_CONFIG_Y     city where the system is located
  -r ESA_CONFIG_R     address where the system is located
  -b ESA_CONFIG_B     building where the system is located
  -s ESA_CONFIG_S     telephone number where the system is located
  -i ESA_CONFIG_I     IBM ID
  -p ESA_CONFIG_P     port number on which the subsystem listens for
                      incoming client requests. Default: 5024
  -w                  Add firewall rules to access ESA UI from remote
                      systems. Default: True
  -Y                  accept license agreement without displaying it.
                      Default: False
```

There are several switches which start with ESA_CONFIG that can be used with the **--esa-config** switch of the **esscallhomeconf** command to activate ESA by using the CLI.

Attention: You can configure software call home without running the **esscallhomeconf** command on the ESS system by using the **mmcallhome** command. However, it is recommended to not enable software call home with **mmcallhome**. Instead, use the **esscallhomeconf** command for this purpose on ESS systems including ESS 3000, ESS 3200, ESS 5000, and ESS Legacy 5.x systems.

ESS 3500 installation, the GUI setup runs the **esscallhomeconf** command, and the ESA and the unified call home are configured. To ensure the ESA and the unified call home are configured and support earlier ESS versions, reconfigure the call home by using the **esscallhomeconf** command (and not the ESA Web User Interface).

An example command output is as follows.

```
# /opt/ibm/ess/tools/bin/esscallhomeconf -E essems1 -N essems1,essio1,essio2 \
--esa-config --register all --icn 123456789 \
-m IBM -n "Jane Doe" -e janedoe@example.com \
-t 5121234567 -g "United States" -s 5121234567
-u "United States" -r "11400 Burnet Rd" \
-y Austin -a TX -z 78758 -b 045 -f "Pablo Marquez"\
-j pablom1@us.ibm.com -k 5121234567\
-Y -i lopezro@us.ibm.com --crvpd \
--serial 212867A --model 8247-21L\

[I] ESA is activated but the configuration was not done.
[I] Activating ESA via CLI using information provided by --esa-config switch
[I] Successfully activated the ESA with customer detail...
2021-02-18T09:41:18.190176 Generating node list...
2021-02-18T09:41:35.966228 nodelist:  essems1   essio1  essio2
Existing vpd file found. --crvpd option is ignored.
End point essems1 registered successfully with systemid 1dab83cc3b9409d5bbf6e657c7e312c8
End point essio1 registered successfully with systemid f7e01a43e9a7464da6cfbe757ca9a669
End point essio2 registered successfully with systemid 1438fddb414738cf60dcade90570059
Skipping node essems1 as it's not an IO node. Only IO nodes are attached to enclosures. Thus
only IO nodes are eligible to be registered their enclosures here.
End point enclosure G51704M registered successfully with systemid
e2b14722f6940b1c410c6ec4452ded9d
End point enclosure G517022 registered successfully with systemid
506794d52b9fd5f7c580ca8e48a051cc
ESA configuration for ESS Call home is complete.
Started configuring software callhome
Checking for ESA is activated or not before continuing.
Fetching customer detail from ESA.
Customer detail has been successfully fetched from ESA.
Setting software callhome customer detail.
Successfully set the customer detail for software callhome.
Enabled daily schedule for software callhome.
Enabled weekly schedule for software callhome.
Direct connection will be used for software calhome.
Successfully set the direct connection settings for software callhome.
Enabled software callhome capability.
Creating callhome automatic group
Created auto group for software call home and enabled it.
Software callhome configuration completed.
```

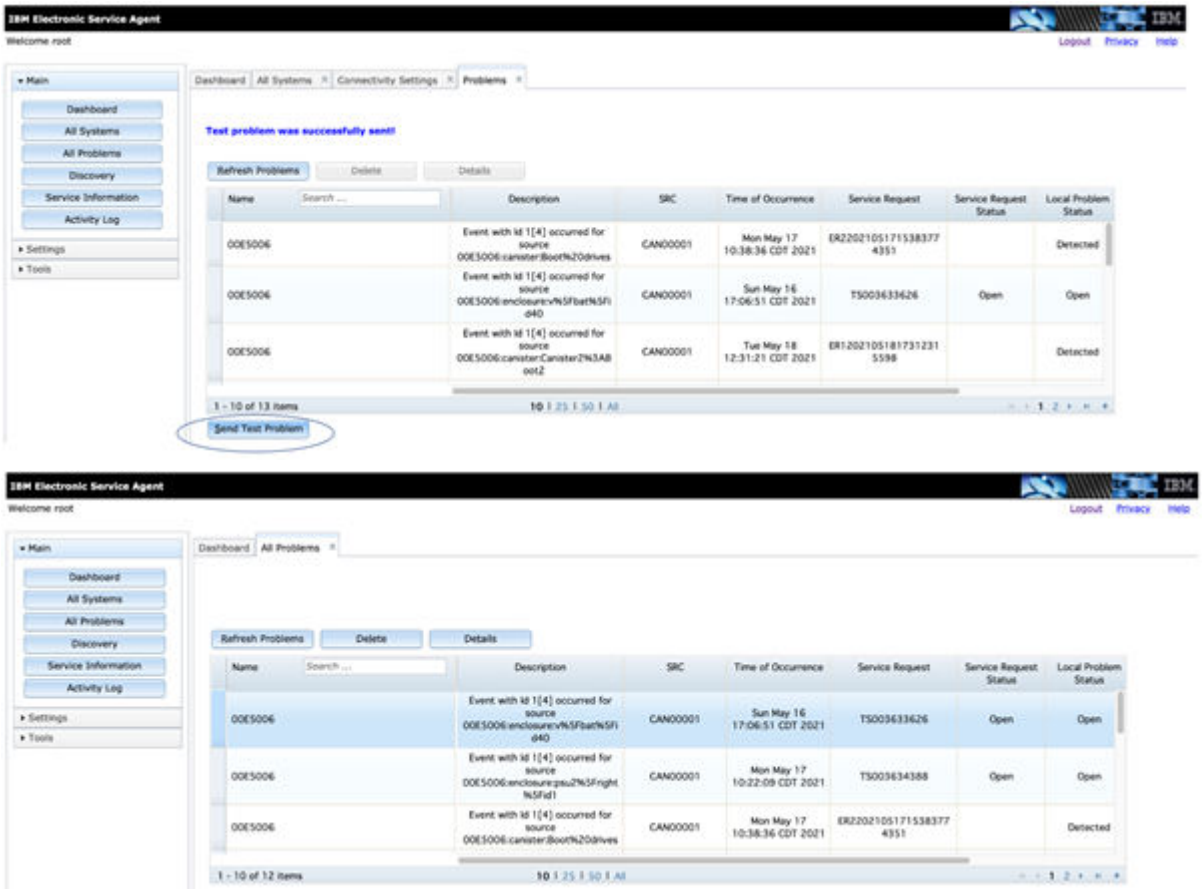
This example shows that ESA is activated and configured, and nodes including enclosures as part of single command line argument are registered. Software call home is set up using the same command line.

```
# esscallhomeconf -E essems1 --show

System id and system name from ESA agent
{
    "506794d52b9fd5f7c580ca8e48a051cc": "G517022",
    "e2b14722f6940b1c410c6ec4452ded9d": "G51704M",
    "1dab83cc3b9409d5bbf6e657c7e312c8": "essems1",
    "f7e01a43e9a7464da6cfbe757ca9a669": "essio1",
    "1438fddb414738cf60dcade90570059": "essio2"
}
```

If for some reason configuration of ESA is successful but configuration of call home fails, it can be done separately by using the **esscallhomeconf** command.

Examples of successful output of test call home connectivity (essinstallcheck) from the ESA web user interface for ESS 5000, ESS 3000, ESS 3200, and ESS Legacy (not ESS 3500):



Configuring proxy for call home

You can configure proxy for call home events by providing the proxy related parameters during configuration of call home. You can also set it after configuring ESA or unified call home.

The following switches are required for setting up proxy for call home.

```
--proxy-ip PROXY-IP-HOSTNAME
                        Provide hostname or IP for proxy
                        configuration.
--proxy-port PROXY-PORT
                        Provide port number for proxy
                        configuration.
--proxy-userid PROXY-USERNAME
                        Provide userid for proxy configuration.
```

```
--proxy-password PROXY-PASSWORD
                        Provide password for proxy
```

Proxy configuration requires ESA or unified call home to be configured otherwise it will fail.

This is an example of configuring proxy along with the configuration inf ESA for call home events.

```
# /opt/ibm/ess/tools/bin/esscallhomeconf -N essems1,essio2  -E essems1
--register all --stop-auto-event-report --verify-esa-cert no --proxy host.example.com
-port 5028 -userid johndoe -password secret -m IBM -n "Jane Doe" -e janedoe@example.com
-t 5121234567 -g "United States" -s 5121234567 -u "United States" -r "11400 Burnet Rd"
-y Austin -a TX -z 78758 -b 045 -f "Pablo Marquez" -j pablom1@us.ibm.com -k 5121234567
-p 5024 -w -Y -i lopezro@us.ibm.com --esa-config --icn 123456789

[I] ESA is activated but the configuration was not done.    >>>> ESA was activated.
[I] Activating ESA via CLI using information provided by --esa-config switch
[I] Successfully activated the ESA with customer detail...
[I] Successfully setup Proxy                >>> proxy set successfully.
[E] Unable to change the setting to stop automatic reporting of hardware event to ESA.
```

```
# /opt/ibm/esa/bin/esacli connectionSettings --display
Connectivity
        Connection Number             1
        Type                          Direct connect
        Proxy IP address or host name
        Proxy port                    0
        Destination user name

       Connection Number             2
        Type                          Proxy
        Proxy IP address or host name host.example.com
        Proxy port                    5028
        Destination user name         johndoe
```

This is an example of configuring proxy after the successful configuration of ESA or unified call home for call home events.

```
# /opt/ibm/ess/tools/bin/esscallhomeconf -N essems1,essio2  -E essems1 --register all
--stop-auto-event-report --verify-esa-cert no --proxy host.example.com -port 5028 -userid
johndoe -password secret
[I] Successfully setup Proxy
[root@essems1 bin]# /opt/ibm/esa/bin/esacli connectionSettings --display
Connectivity
        Connection Number             1
        Type                          Direct connect
        Proxy IP address or host name
        Proxy port                    0
        Destination user name

        Connection Number             2
        Type                          Proxy
        Proxy IP address or host name host.example.com
        Proxy port                    5028
        Destination user name         johndoe
```

ESS call home logs and location

The **esscallhomeconf** command logs the progress and error messages in the `/var/log/messages` file, the `/var/log/ess/` folder, and `/var/adm/ras/mmsysmonitor.log` file.

There is a **--verbose** option that provides more details of the progress and error messages. The following example displays the type of information sent to the `/var/log/messages` file on the EMS node by the **esscallhomeconf** command.

```
# grep essems4 /var/log/messages | grep esscallhomeconf
Feb 23 10:07:14 essems4 /esscallhomeconf: End point ems1-ib registered successfully with
systemid ed28297131f0d2b469edffc505a9708c
Feb 23 10:07:20 essems4 /esscallhomeconf: [I] End point essio11-ib registered successfully with
systemid db25a7e21ff4298243078806f964c495
Feb 23 10:07:20 essems4 /esscallhomeconf: [I] End point essio12-ib registered successfully with
systemid f9887f2bba6dee858146206dda96eb48
Feb 23 10:07:51 essems4 /esscallhomeconf: [I] End point proto11-ib registered successfully with
systemid 12ca060f30f9276cd52828fc117b0675
Feb 23 10:26:59 essems1 /esscallhomeconf: [I] End point enclosure EB15089 registered
```

```
successfully with systemid 72fadb281627047372f9ada47ed2fcb4
Feb 23 10:27:05 essems1 /esscallhomeconf: [I] End point enclosure EB15094 registered
successfully with systemid b266c524642846255f38a493e99bf10a
Feb 23 10:27:05 essems1 /esscallhomeconf: [I] End point enclosure EB15090 registered
successfully with systemid 8f9a45df3eb6137f6890ab18cf4c2957
End point enclosure EB15090
Feb 23 10:27:28 essems1 /esscallhomeconf: [I] ESA configuration for ESS Call home is complete.
Feb 23 10:28:04 essems1 /esscallhomeconf: [I] Software callhome configuration completed.
```

Attention: The **esscallhomeconf** command also configures the IBM Spectrum Scale call home setup. The IBM Spectrum Scale call home feature collects files, logs, traces, and details of certain system health events from the I/O and EMS nodes and services running on those nodes. These details are shared with the IBM support center for monitoring and problem determination. For more information on IBM Spectrum Scale call home, see IBM Spectrum Scale documentation in IBM Documentation.

Note: The ESS 3000, ESS 3200, and ESS 3500 hardware call home is backed by software call home. In other words, software call home must be configured by using the **esscallhomeconf** command, without the **--no-swcallhome** switch in the ESS 3000 or the ESS 3200 environment. Otherwise, the ESS 3000 or the ESS 3200 hardware failure events are not reported to ESA and a PMR does not get opened.

The endpoints are visible in the ESA portal after registration, as shown in the following figure:

Figure 8. ESA portal after node registration

Name

Shows the name of the endpoints that are discovered or registered.

SystemHealth

Shows the health of the discovered endpoints. A green icon (√) indicates that the discovered system is working fine. The red (X) icon indicates that the discovered endpoint has some problem.

ESAStatus

Shows that the endpoint is reachable. It is updated whenever there is a communication between the ESA and the endpoint.

SystemType

Shows the type of system being used. Following are the various ESS device types that the ESA supports.

ESS Device type	Icon
ESS Application	
Disk	
Disk Enclosure	
Management Server	
Node	
Physical Server	
Virtual Server	
Other	

Figure 9. List of icons showing various ESS device types

Detailed information about the node can be obtained by selecting **System Information**. Here is an example of the system information:

System Information

Property	Value
Name	essio12.isst.gpfs.ibm.net
Machine Type	8247
Machine Model	22L
Serial Number	2145B3A
Manufacturer	IBM
Operating System	Linux
OS Type	Linux
OS Version	3.10.0-327.36.3.el7.ppc64
OS Additional Version	
IP Address	192.168.1.103 192.168.2.103
Firmware	
PM Enabled	No
ESA Status	Offline
System ID	898fb33e04f5ea12f2f5c7ec0f8516d4

Figure 10. System information details

When an endpoint is successfully registered, the ESA assigns a unique system identification (system id) to the endpoint. The system ID can be viewed using the **--show** option.
For example:

```
# esscallhomeconf -E ems4 --show
System id and system name from ESA agent
{
    "32eb1da04b60c8dbc1aaaa9b0bd74976": "78ZA006",
    "6304ce01ebe6dfb956627e90ae2cb912": "ems4-ce",
    "a575bdce45efcfdd49aa0b9702b22ab9": "essio41-ce",
```

```
    "5ad0ba8d31795a4fb5b327fd92ad860c": "essio42-ce"
}
```

When an event is generated by an endpoint, the node associated with the endpoint must provide the system id of the endpoint as part of the event. The ESA then assigns a unique event id for the event. The system id of the endpoints are stored in a file called `esaepinfo01.json` in the `/vpd` directory of the EMS and I/O servers that are registered. The following example displays a typical `esaepinfo01.json` file:

```
# cat /vpd/esaepinfo01.json
{
    "encl": {
        "78ZA006": "32eb1da04b60c8dbc1aaaa9b0bd74976"
    },
    "esaagent": "ems4",
    "node": {
        "ems4-ce": "6304ce01ebe6dfb956627e90ae2cb912",
        "essio41-ce": "a575bdce45efcfdd49aa0b9702b22ab9",
        "essio42-ce": "5ad0ba8d31795a4fb5b327fd92ad860c"
    }
}
```

The endpoints are visible in the ESA portal after registration. For more information, see IBM Spectrum Scale call home documentation.

Overview of a problem report

After ESA or unified call home is activated, and the endpoints for the nodes and enclosures are registered, they can send an event request to initiate a call home.

For example, when `replace` is added to a pdisk state, indicating that the corresponding physical drive needs to be replaced, an event request is sent to ESA with the associated system id of the enclosure where the physical drive resides. After ESA receives the request it generates a call home event. Each server in the ESS is configured to enable callback for IBM Spectrum Scale RAID related events. These callbacks are configured during the cluster creation, and updated during the code upgrade. ESA can filter out duplicate events when event requests are generated from different nodes for the same physical drive. ESA returns an event identification value when the event is successfully processed. The ESA portal updates the status of the endpoints. The following figure shows the status of the enclosures when the enclosure contains one or more physical drives identified for replacement.

ESS 5000, ESS 3000, ESS 3200, ESS 3500, and ESS Legacy Disk enclosure failure call home event

- `pdReplacePdisk` - When a pDisk needs replacement inside an enclosure, ESA will get an event and a corresponding PMR will get opened. This event is not meant to report the server operating disk failure.

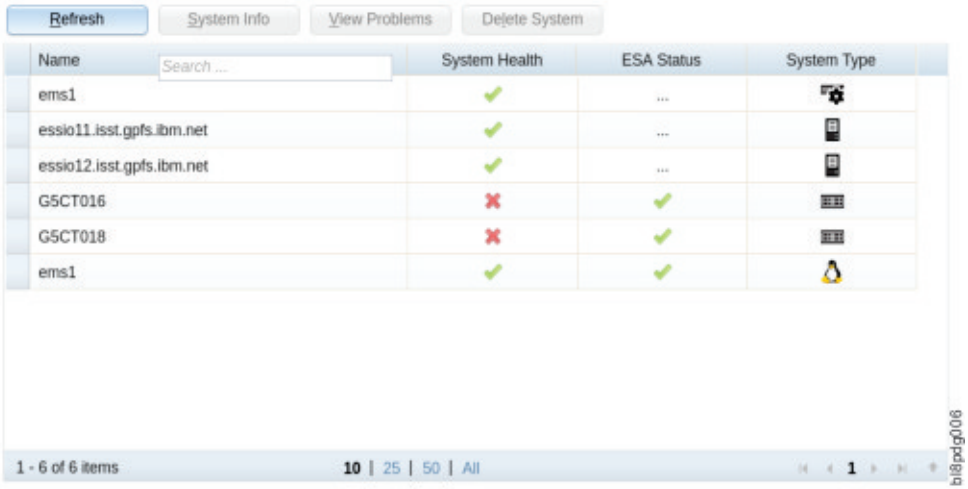


Figure 11. ESA portal showing enclosures with drive replacement events

Another example is POWER9 or POWER8 hardware failure which could be an ESS 5000 or ESS Legacy IO node or EMS node or protocol node hardware failure. Hardware failure event could be DIMM failure, power supply failure, or fan failure in POWER9 or POWER8 nodes. Any of the failure events reported by the POWER nodes are recorded in OPAL log and ESS hardware call home function reads the OPAL events which need service and it is a call home event.

ESS 5000, ESS 3000, ESS 3200, ESS 3500, and ESS Legacy Hardware failure event raised in OPAL log reported by event type:

- `nodeEvent` - Any POWER9 node hardware failure which is reported in the OPAL log is a call home event that is reported with this event.

Similarly, ESS 3000, ESS 3200, and ESS 3500 x86 nodes can also report any of the hardware failure events with **mmhealth**. The ESS 3000 call home depends on software call home and **mmhealth**. In other words, software call home must be configured and **mmhealth** must detect the issue with hardware to get it reported to ESA and a PMR getting opened.

While running **esscallhomeconf** command, make sure to not use the **--no-swcallhome** switch in the ESS 3000, ESS 3200, and ESS 3500 environment. Otherwise, the ESS 3000, ESS 3200, or ESS 3500 hardware failure events are not reported to ESA and a PMR does not get opened.

ESS 3000, ESS 3200, and ESS 3500 supported hardware events:

- `bootDrvFail` - When boot drive failed at canister.
- `canFailed` - When canister failed.
- `bootDrvMissing` - When boot drive missing.
- `bootDrvSmtFailed` - When boot drive smart failed.
- `canFanFailed` - When canister FAN stopped working.
- `fanFailed` - When FAN failed.
- `psFailed` - Power supply failure.
- `psFanFailed` - Power supply FAN failure.

Problem details section of ESA

The problem descriptions of the events can be seen by selecting the endpoint. You can select an endpoint by clicking the red X.

You can use the **mmcallhome ticket list/delete** command to display unified call home information on ESS 3500 only.

```
mmcallhome ticket list
```

A sample output is as follows:

```
Tickets Opened
Ticket      Creation             Status          Description
Number      Date
-------------------------------------------------------------------------------------------------
TS008145928 2022-04-25 10:50:35 New Case Opened 00EG01E:enclosure:00EG01E/0:psu1_fan1_id0:Fan
                                         psu1_fan1_id0 is failed.
TS008146144 2022-04-25 15:30:30 New Case Opened 00EG01E:enclosure:00EG01E/7:enclosure_fan6_id7:Fan
                                         enclosure_fan6_id7 is failed.
[root@emsdev2 ~]#
```

The rest information in this section applies to ESA on ESS 5000, ESS 3000, and ESS 3200.

The following figure shows an example of the problem description.

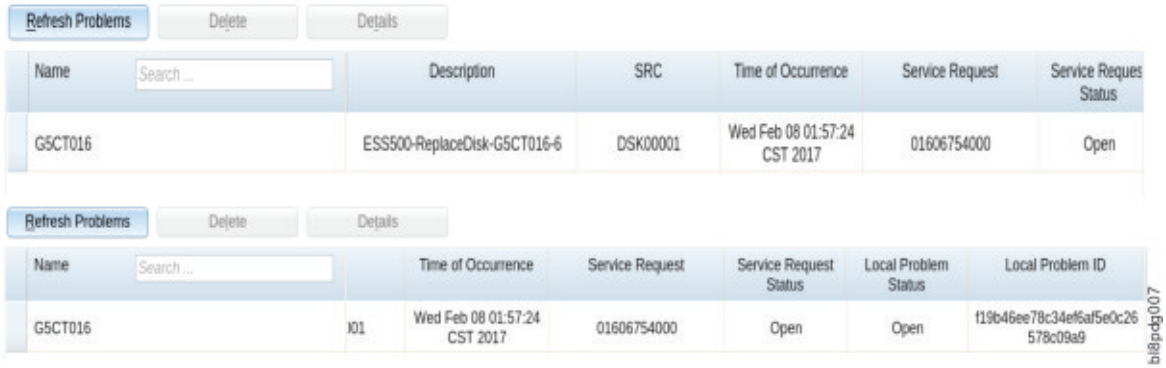


Figure 12. Problem Description

Name

It is the serial number of the enclosure containing the drive to be replaced.

Description

It is a short description of the problem. It shows ESS version or generation, service task name and location code. This field is used in the synopsis of the problem (PMR) report.

SRC

It is the Service Reference Code (SRC). An SRC identifies the system component area. For example, DSK XXXXX, that detected the error and additional codes describing the error condition. It is used by the support team to perform further problem analysis, and determine service tasks associated with the error code and event.

Time of Occurrence

It is the time when the event is reported to the ESA. The time is reported by the endpoints in the UTC time format, which ESA displays in local format.

Service request

It identifies the problem number (PMR number).

Service Request Status

It indicates reporting status of the problem. The status can be one of the following:

Open

No action is taken on the problem.

Pending

The system is in the process of reporting to the IBM support.

Failed

All attempts to report the problem information to the IBM support has failed. The ESA automatically retries several times to report the problem. The number of retries can be configured. Once failed, no further attempts are made.

Reported

The problem is successfully reported to the IBM support.

Closed

The problem is processed and closed.

Local Problem ID

It is the unique identification or event id that identifies a problem.

Problem Details

Further details of a problem can be obtained by clicking the **Details** button. The following figure shows an example of a problem detail.

Problem Summary

Property	Value
Description	ESS500-ReplaceDisk-G5CT018-5
Error Code	DSK00001
Local Problem Status	Open
Problem ID	53c76032dbb54069a28db04a7c229bc3
Is Test Problem?	false
Problem Occurence Date/Time	2/8/17 1:57 AM

Transmission Summary

Property	Value
Service Information Sent to IBM support	Yes
Last Attempt to Send	2/8/17 1:57 AM
Number of Attempts	1

Service request information

Property	Value
Problem Severity	
Service Request Number	01605754000
Service Request Status	Open
Last Changed	2/8/17 1:57 AM

Figure 13. Example of a problem summary

If an event is successfully reported to the ESA, and an event ID is received from the ESA, the node reporting the event uploads additional support data to the ESA that are attached to the problem (PMR) for further analysis by the IBM support team.

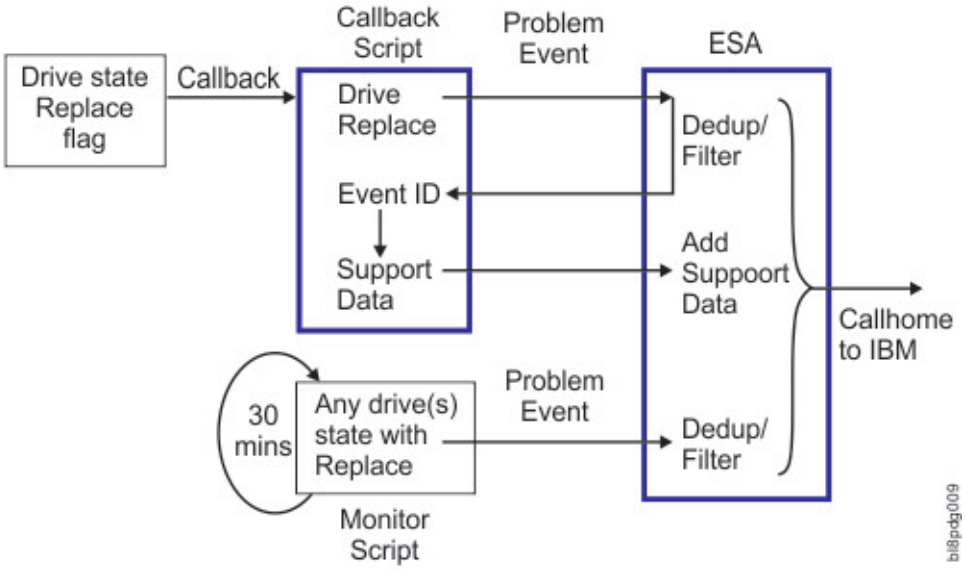


Figure 14. Call home event flow

The callback script logs information in the `/var/log/messages` file during the problem reporting episode. The following examples display the messages logged in the `/var/log/message` file generated by the `essio11` node:

- Callback script is invoked when the drive state changes to `replace`. The callback script sends an event to the ESA:

```
Feb 8 01:57:24 essio11 esscallhomeevent: [I] Event successfully sent
for end point G5CT016, system.id 524e48d68ad875ffbeeec5f3c07e1acf,
 location G5CT016-6, fru 00LY195.
```

- The ESA responds by returning a unique event ID for the system ID in the json format.

```
Feb 8 01:57:24 essio11 esscallhomeevent:
{#012 "status-details": "Received and ESA is processing",
#012 "event.id": "f19b46ee78c34ef6af5e0c26578c09a9",
#012 "system.id": "524e48d68ad875ffbeeec5f3c07e1acf",
#012 "last-activity": "Received and ESA is processing"
#012}
```

Note: Here #012 represents the new line feed \n.

- The callback script runs the **ionodedatacol.sh** script to collect the support data. It collects the last 10000 lines of `mmfs.log.latest` file and the last 24 hours of the kernel messages in the journal into a .tgz file.

```
Feb 8 01:58:15 essio11 esscallhomeevent: [I] Callhome data collector
/opt/ibm/gss/tools/samples/ionodechdatacol.sh finished

Feb 8 01:58:15 essio11 esscallhomeevent: [I] Data upload successful
for end point 524e48d68ad875ffbeeec5f3c07e1acf
and event.id f19b46ee78c34ef6af5e0c26578c09a9
```

Call home monitoring of ESS 5000, ESS 3000, ESS 3200, and ESS Legacy systems and their disk enclosures

A callback is a one-time event. Therefore, it is triggered when the disk state changes to `replace`. If ESA misses the event, for example if the EMS node is down for maintenance, the call home event is not generated by ESA.

Important: Information in this section is not applicable for ESS 3500, because ESS 3500 uses the unified call home to monitor ESS systems and disk enclosures.

To mitigate this situation, the `callhomemon.sh` script is provided in the `/opt/ibm/gss/tools/samples` directory of the EMS node. This script checks for pdisks that are in the `replace` state, and sends an event to ESA to generate a call home event if there is no open PMR for the corresponding physical drive. This script can be run on a periodic interval. For example, every 30 minutes.

In the EMS node, create a cronjob as follows:

1. Open crontab editor by using the following command:

```
# crontab -e
```

2. Setup a periodic cronjob by adding the following line:

```
*/30 * * * * /opt/ibm/ess/tools/samples/callhomemon.sh
```

3. View the cronjob by using the following command:

```
# crontab -l
*/30 * * * * /opt/ibm/ess/tools/samples/callhomemon.sh
```

The call home monitoring protects against missing a call home due to ESA missing a callback event. If a problem report is not already created, the call home monitoring ensures that a problem report is created.

Note: When the call home problem report is generated by the monitoring script, as opposed to being triggered by the callback, the problem support data is not automatically uploaded. In this scenario, the IBM support can request support data from the customer.

Note: A PMR is created because of the periodic checking of the replaced drive state. For example, when the callback event is missed, additional support data is not provided for the ESA agent.

Attention:

- In case of ESS 5000 or ESS Legacy systems, if the hardware event reported by OPAL or any disk enclosures disk error event attached to POWER IO nodes, which was missed because ESA was down or due to some other issue in the EMS node, then events can be triggered manually by invoking the **/opt/ibm/gss/tools/samples/callhomemon.sh** script. The **callhomemon.sh** script reports any missed or new hardware event reported by any POWER9 or POWER8 node, which might be a part of ESS 5000 or ESS Legacy cluster (such as POWER EMS Node, POWER protocol nodes, etc.) and ESS 3000 or ESS 3200 disk enclosures disk failure event only, if it is a part of ESS cluster.
- In case of ESS 3000 or ESS 3200 systems, if the hardware event reported by **mmhealth** is missed because ESA is down or due to some other issue in the EMS node then event can be re-triggered by using **mmhealth node eventlog --clear** followed by **mmsysmoncontrol restart**. In case the disk enclosures disk event is missed on the ESS 3000 or the ESS 3200 system because the ESA is down or due to some other issue at EMS node then event can be triggered manually by invoking the **/opt/ibm/gss/tools/samples/callhomemon.sh** script. The **callhomemon.sh** script in case of ESS 3000 or ESS 3200 only re-sends the missed or old disk enclosures disk error event and any missed or new hardware event reported by any POWER9 which might be a part of an ESS 3000 or an ESS 3200 cluster (such as POWER EMS node and POWER protocol nodes).
- In case of ESS 3500 systems, if a hardware event reported by the **mmhealth** command is missed because EMS is down or some issue, the event is re-triggered automatically when the EMS comes back online.

Upload data

The following support data is uploaded when the ESS system disks in enclosures display a drive replace notification on an ESS 5000, ESS 3000, ESS 3200, ESS 3500, or ESS Legacy system.

- The output of **mmlspdisk** command for the pdisk that is in replace state.
- Additional support data is provided only when the event is initiated as a response to a callback. The following information is supplied in a .tgz file as additional support data:
  - Last 10000 lines of `mmfs.log.latest` from the node which generates the event.
  - Last 24 hours of the kernel messages (from journal) from the node which generates the event.

The following support data is uploaded when the system displays any hardware issue in an ESS 5000 or an ESS Legacy system.

- The output of the **opal_elog_parse** command for the serviceable event that caused failure.
- Additional support data is provided only when the event is initiated as a response to a callback. The following information is supplied in a .tgz file as additional support data:
  - Last 10000 lines of `mmfs.log.latest` from the node which generates the event.
  - Last 24 hours of the kernel messages (from journal) from the node which generates the event.

The following support data is uploaded when the system displays any hardware issue in an ESS 3000 or an ESS 3200 system.

- The output of the **mmhealth** command and the actual component that caused failure.
- Additional support data is provided only when the event is initiated as a response to a callback. The following information is supplied in a .tgz file as additional support data:
  - Last 10000 lines of `mmfs.log.latest` from the node which generates the event.
  - Last 24 hours of the kernel messages (from journal) from the node which generates the event.

Uninstalling, reinstalling, and troubleshooting the IBM Electronic Service Agent

The ESA agent is preinstalled in the EMS node from the factory.

Issue the following command to remove the rpm if needed:

```
# yum remove esagent.pLinux-4.5.7-1.noarch
```

Issue the following command to reinstall the rpm files for the ESA agent:

```
# yum localinstall  path/esagent.pLinux-4.5.7-1.noarch.rpm
```

Where the path is `/install/ess/otherpkgs/rhels8/ppc64le/ess`. The path can also be `/opt/ibm/ess/mnt/installer/otherpkgs/rhels7/ppc64le/ess` or `/opt/ibm/ess/mnt/installer/otherpkgs/rhels8/x86_64/ess` if **essmgr** is run to start the container.

Attention:

- The ESA agent requires the Open JDK 8 files to be installed on the provided node by the using the standard RHEL DVD. The `gpfs.java` package must be installed on the node but make sure that `JAVA_HOME` is not pointing to the `gpfs.java` installed directory because `gpfs.java` contains JDK 11 which is not compatible with ESA.
- Make sure that there is no other JDK such as JDK 11 or any other version of the JDK other than JDK 8 (provided by standard RHEL DVD) installed on the EMS node. ESA uses the default Java™ version which is in default the PATH and it must be Open JDK v8.x. If a JDK version other than JDK 8 is installed and the default PATH to Java uses a different version of JDK, the ESA activation will fail and call home function will not work.
- Administrator can clean any other or older version of JDK by using the **yum remove <install_java>** command and then mount the DVD ISO provided by the ESS binary and get the Open JDK 8 installed on EMS node.
- Do not change the `gpfs.java` package either by re-installing or uninstalling or else GPFS GUI functionality will not work.

Test call home

The configuration and setup for call home must be tested to ensure that the disk replace event can trigger a call home.

The test is composed of three steps:

- ESA connectivity to IBM - Check connectivity from ESA to IBM network. This might not be required if done during the activation.

  ```
  /opt/ibm/esa/bin/verifyConnectivity -t
  ```

- ESA test call home - Test call home from the ESA portal. Go to **All systems** > **System Health** for the endpoint from which you would like to generate a test call home. Click **Send Test Problem**' from the newly opened **Problems** tab.
- ESS call home script setup to ensure that the callback script is set up correctly.

  Verify that the periodic monitoring is set up.

  ```
  crontab -l
  [root@ems1 deploy]# crontab -l

  */30 * * * * /opt/ibm/ess/tools/samples/callhomemon.sh
  ```

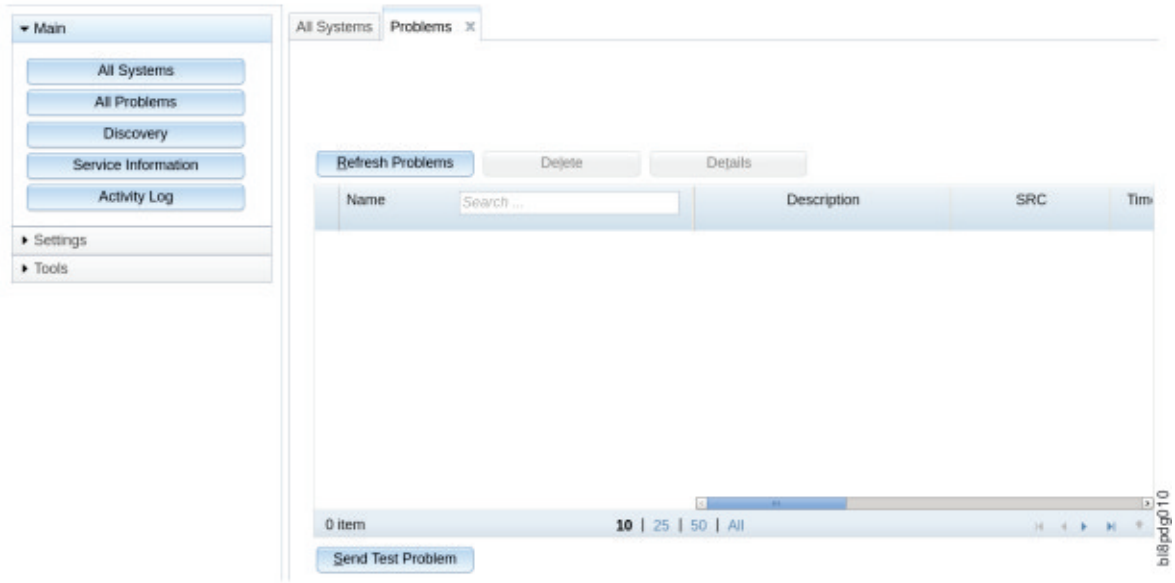


Figure 15. Sending a test problem with ESA web user interface

Callback script test

Verify that the system is healthy by issuing the **gnrhealthcheck** command. You must also verify that the active recovery group (RG) server is the primary recovery group server for all recovery groups. For more recovery group details, see the *IBM Spectrum Scale RAID: Administration* guide.

Example:

To test the callback script, select a pdisk from each enclosure alternating recovery groups. The purpose of the test call home events is to ensure that all the attached enclosures can generate call home events by using both the I/O servers in the building block.

For an ESS 5000 building block, select a pdisk from each enclosure alternating recovery groups. The purpose of the test call home events is to ensure that all the attached enclosures can generate call home events by using both the I/O server nodes in the building block.

In an ESS 5000 system, select a pdisk from each enclosure alternating the paired recovery groups. The purpose of the test call home is to ensure that all the attached enclosures can generate call home events by using both the I/O Server nodes in the building block.

For example, in a SC2 system with 5147-106 enclosures, one can select pdisks e1s001 (left RG, rgL) and e2s106 (right RG, rgR). Similarly, for a SL2 system with 5147-092 enclosures, one can select pdisks e1s02 (left RG, rgL) and e2s92 (right RG, rgR). You must find the corresponding recovery group and active server for these pdisks. Send a disk event to the ESA from the active recovery group server as shown in the following steps:

1. ssh to essio11.

   Here the paired recovery groups are rg1L and rg1R, and the corresponding active I/O server nodes are essio11-ib and essio12-ib.

   Note: This applies to both ESA and unified call home and all flavors of ESS.

   Select the event to see the details:

   ```
   esscallhomeevent --event test
   ```

2. ssh to essio12 and run the following command:

   ```
   esscallhomeevent --event test
   ```

Post setup activities

Perform the following post setup activity.

- Delete any test problems.

essinstallcheck enhancement of software and hardware call home

essinstallcheck is now capable of checking and verifying that the systems are configured with software and hardware call home. If the cluster is not yet created, the command skips checking for software call home.

```
# essinstallcheck -N localhost
Start of install check
nodelist:  localhost
Getting package information.
[WARN] Package check cannot be performed other than on EMS node. Checking nodes.
================== Summary of node: localhost ============================
[INFO] Getting system firmware level. May take a long time...
[INFO] Getting system profile setting.
[INFO] Spectrum Scale RAID is not active, cannot get gpfs Installed
version:
[OK] Linux kernel installed:                    4.18.0-372.16.1.el8_6.x86_64
[ERROR] Systemd not at min recommended level: 239-58.el8_6.x86_64
[ERROR] Networkmgr not at min recommended level: 1.36.0-7.el8_6.x86_64
[OK] Mellanox OFED level:                               MLNX_OFED_LINUX-5.6-2.0.9.0
[OK] IPR SAS FW:                                19512B00
[OK] ipraid RAID level:                         10
[ERROR] ipraid RAID Status: found Degraded expected Optimized
[OK] IPR SAS queue depth:                       64
[ERROR] System Firmware :                       found FW860.81 (SV860_215) expected min
FW860.90 (SV860_226)
[OK] System profile setting:                    scale
[OK] System profile verification PASSED.
[INFO] Cluster not yet created skipping rsyslog check
[OK] Host adapter driver:                       34.00.00.00
Performing Spectrum Scale RAID configuration check.
[OK] New disk prep script: /usr/lpp/mmfs/bin/tspreparenewpdiskforuse
[OK] Network adapter MT4099 firmware: 16.27.2008, net adapter count: 3
[OK] Network adapter firmware
[INFO] Storage firmware check is not required as GPFS cluster does not exist.
[OK] Node is not reserving KVM memory.
[OK] IBM Electronic Service Agent (ESA) is activated for Callhome service.
[OK] Software callhome check skipped as cluster not configured.
End of install check
[PASS] essinstallcheck passed successfully
```

You can view two more lines in the **essinstallcheck** output (in bold face) which mention that ESA is activated (ESA activation indicates that the hardware call home is also configured for this ESS) and software call home has been configured for this node. This is a very important check which enables customers to configure hardware and software call home after the cluster creation and the file system creation is done.

Remember: Enable the hardware and the software call home at the end of the ESS system deployment when the file system is active, nodes are ready to serve the file system, and none of the configuration is pending.

Call home pre-installation worksheets

Fill in the following call home worksheets before the installation.

General Information	
Primary Contact for Providing Site Access: (Required)	

General Information	
Contact Information for Primary Contact (Phone & Email): (Required) **Note:** If the phone number is provided, indicate if it is a mobile number or office number.	
System Location Settings **Note:** The sequence of the following information aligns with the configuration panels in the Electronic Service Agent.	
Country or Region: (Required) **Note:** You need to provide the Alpha-2 Code for this entry. You can obtain this information by searching for the country code on ISO Online Browsing Platform.	
State or Province: (Required)	
Postal Code: (Required)	
City: (Required)	
Street Address: (Required)	
Building, Floor, Office: (Required)	
Telephone Number: (Required)	

Service Contact Settings - Company Information	
Company Name: (Required)	
Street Address: (Optional)	
City: (Optional)	
State or Province: (Optional)	

Service Contact Settings - Company Information	
Country or Region: (Required) **Note:** You need to provide the Alpha-2 Code for this entry. You can obtain this information by searching for the country code on ISO Online Browsing Platform.	
Postal Code: (Optional)	
Fax Number: (Optional)	
Alternate Fax Number: (Optional)	
Help Desk Number: (Optional)	
Service Contact Settings - Primary Contact	
Contact Name: (Required)	
Telephone Number: (Required)	
Alternate Telephone Number: (Optional)	
E-mail Address: (Required)	
Alternate E-mail Address: (Optional)	
Pager Number: (Optional)	
Service Contact Settings - Secondary Contact	
Contact Name: (Required)	
Telephone Number: (Required)	
Alternate Telephone Number: (Optional)	

Service Contact Settings - Company Information	
E-mail Address: (Required)	
Notifications Settings - Email Notification	
Enable E-mail Notification (Y or N): (Required)	
SMTP Server Name: (Required)	
Port: (Required)	
User ID: (Optional)	
Password: (Optional)	
Email Address(s) to Receive Notifications: (Required)	
Notifications Settings - SNMP Traps	
Enable SNMP Trap (Y or N): (Required)	
Target Network Host: (Required)	
Community: (Required)	
Port: (Required)	
IBM ID Settings IBM provides customized web tools and functions that use information collected by IBM Electronic Service Agent. The access to these functions is managed by an association between your IBM ID(s) and the Electronic Service Agent information from your systems. The association is made using this page. To obtain an IBM ID, which is used by many IBM web sites, go to http://www.ibm.com/registration	
IBM ID(s) to Add: (Required)	

Appendix E. Security-related settings in ESS

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

Working with firewall in ESS

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

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

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

  ```
  # essrun -N ems1 firewall enable
  ```

  You can check the status of the firewall as follows.

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

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

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

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

  ```
  # essrun -N nodeList firewall enable
  ```

  You can check the status of the firewall as follows.

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

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

  ```
  # essrun -N nodeList firewall verify
  ```

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

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

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

  ```
  # essrun -N nodeList firewall verify
  ```

Working with SELinux in ESS

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

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

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

```
# essrun -N ems1 selinux permissive
```

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

  b) Reboot the node.

```
# systemctl reboot
```

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

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

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

```
# essrun -N ems1 selinux enable
```

  No reboot is required in this case.

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

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

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

```
# essrun -N ess_x86_64 selinux permissive
```

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

  b) Reboot the I/O server nodes.

```
# systemctl reboot
```

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

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

```
# essrun -N ess_x86_64 selinux enable
```

No reboot is required in this case.

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

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

    ```
    # essrun -N ems1 selinux disable
    ```

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

    You can check the status as follows.

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

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

    ```
    # essrun -N ess_x86_64 selinux disable
    ```

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

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

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

Working with sudo in ESS

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

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

- "Enabling sudo on ESS" on page 73
- "Disabling sudo on ESS" on page 74

Enabling sudo on ESS

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

```
# essrun -N nodeList sudo enable
```

Note: To configure a sudo user, see the **essrun sudo** command in the Elastic Storage System: Command Reference.

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

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

Make sure that the **sudo** sub-command is run on all GPFS nodes (EMS node, I/O server nodes, and any client nodes) as part of the cluster to be completely compliant with the sudo requirement. Change the

node name in the **sudo** sub-command accordingly. Enabling sudo also allows the `gpfsadmin` user to administer xCAT and the GPFS GUI on the EMS node.

Disabling sudo on ESS

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

```
# essrun -N nodeList sudo disable
```

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

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

Enabling root access on ESS

To enable root access, issue the following command:

```
# essrun -N nodeList sudo enable_root --sudo-password YourSudoPassword
```

Disabling root access on ESS

To enable root access, issue the following command:

```
# essrun -N nodeList sudo disable
```

Note: You can disable root in all ESS nodes except ESS 3000.

Working with Central Administration mode in ESS

Note: To enable the central administration mode, you must have unique public keys on each node. By default, ESS sets up a shared public key. You must manually configure unique keys on each node before enabling this feature.

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

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

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

- "Enabling the central administration mode" on page 74
- "Disabling the central administration mode" on page 75
- "Help text admincentral sub-command" on page 76

Enabling the central administration mode

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

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

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

```
# essrun -N cems0 admincentral enable
```

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

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

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

Note:

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

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

Disabling the central administration mode

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

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

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

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

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

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

Note:

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

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

Help text admincentral sub-command

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

positional arguments:
  {enable, disable}

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

Appendix F. Enabling Chrony timeset in ESS

Enabling Chrony timeset in an ESS environment is done as follows. It can be enabled for EMS, I/O server nodes, and protocol nodes by using the **timeset** sub-command of the **essrun** command.

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

Note: The Chrony timeset works in a client-server model. You need to edit the client's `chrony.conf` file with the management IP address of the Chrony server. You also need to edit the server's `chrony.conf` with the comma-separated IP addresses of clients. The file location is `/etc/chrony.conf`.

Enable Chrony on client nodes:

```
# essrun -N nodeList timeset client --server-ip your.ntp.server.ip
```

Note: If you need to set up a node as a server issue the following command:

```
# essrun -N ems1 timeset server --server-network your.ntp.server.network/mask
```

This command requires EMS to have firewall enabled. Because NTP requests come from certain ports that are blocked for EMS nodes by default.

Appendix G. Upgrading the POWER9 firmware

The POWER9 firmware must be upgraded manually. If the firmware is not at the latest level, do the following steps.

1. Copy the firmware `img` file from the container to the POWER9 EMS or the POWER9 protocol node.

   ```
   cd /install/ess/otherpkgs/rhels8/ppc64le/firmware/
   sftp EMSNode
   mput 01VL950_105_045.img
   ```

2. Shut down the container.

   ```
   podman stop ContainerName
   ```

3. Upgrade the firmware.

   a) Run the following command.

   ```
   update_flash -v -f 01VL950_105_045.img
   ```

   b) If there are no issues, execute the update.

   ```
   update_flash -f 01VL950_105_045.img
   ```

   The system restarts and the firmware is upgraded. This process might take up to 1.5 hours per node.

Note: If you plan to upgrade the POWER8 EMS firmware, you can retrieve the code (01SV860_243_165.img file) from the following location inside the container:

```
/install/ess/otherpkgs/rhels7/ppc64le/firmware/
```

Follow the same steps to upgrade the POWER8 EMS firmware. The level after the upgrade will be SV860_243_165 (FW860.B1). The level after upgrade of the POWER9 firmware will be FW950.50 (FW950.105).

Manually updating a boot drive for ESS 5000

Complete the following steps to update a boot drive:

1. Get the `HDD10KV9-4K-9F23-LINUX.rpm` file that is available in the `/install/ess/otherpkgs/rhels8/ppc64le/firmware` container and transfer it to the I/O nodes.

   ```
   scp <complete file with path> ionode:/root
   ```

2. Install the rpm on an I/O node.

   ```
   # rpm -ivh /root/HDD10KV9-4K-9F23-LINUX.rpm
   ```

3. Check the installed package.

   ```
   # rpm -ql ibm.sas.HDD10KV9-4K-9F23-1.noarch
   /lib/firmware/IBM-ST1200M.A1800017.39463233
   /lib/firmware/IBM-ST1800M.A1800017.39463233
   /lib/firmware/IBM-ST600MM.A1800017.39463233
   ```

4. Check whether any boot drives need to be updated.

   ```
   # lsscsi -g | grep -i storage
   ```

   A sample output is as follows:

```
[1:0:0:0]    storage IBM      ST1800MM0139     9F14  -          /dev/sg1
[1:0:1:0]    storage IBM      ST1800MM0139     9F14  -          /dev/sg2
```

Here,

ST1800MM0139
This is the product ID.

/dev/sg1 and /dev/sg2
These are the storage devices.

9F14
This is the device version.

5. Check whether the firmware file matches to the product ID.
6. If the firmware file and the product ID are matching, upgrade the firmware.

```
/lib/firmware/IBM-ST1800M.A1800017.39463233
```

7. Upgrade the code on each device.

```
# iprconfig -c update-ucode /dev/sg1 /lib/firmware/IBM-ST1800M.A1800017.39463233
```

A sample output is as follows:

```
# iprconfig[147710]: 0:0:0:0: Updating device microcode using /lib/firmware/IBM-
ST1800M.A1800017.39463233 from 39463233 (9F23) to 39463233 (9F23)
```

8. Upgrade the same code on the /dev/sg2 device.

```
# iprconfig -c update-ucode /dev/sg2 /lib/firmware/IBM-ST1800M.A1800017.39463233
```

A sample output is as follows:

```
# iprconfig[147710]: 0:0:0:0: Updating device microcode using /lib/firmware/IBM-
ST1800M.A1800017.39463233 from 39463233 (9F23) to 39463233 (9F23)
```

9. After both devices are successfully updated, reboot the I/O node.
10. Check whether the firmware version is changed.

```
# lsscsi -g | grep -i storage
```

A sample output is as follows:

```
[0:0:0:0]    storage IBM      ST1800MM0139     9F23  -          /dev/sg1
[0:0:1:0]    storage IBM      ST1800MM0139     9F23  -          /dev/sg2
```

The firmware version is changed from 9F14 to 9F23.

Appendix H. How to set up chronyd (time server) in non-ESS nodes

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.

  Note: Replace the server and the allow range with the network settings specific to your setup.

```
# 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.

  Note: Replace the server and the allow range with the network settings specific to your setup.

```
# 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 I. ESS 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 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 83
- ESS container has the protocol node management IP addresses defined. For more information, see "ESS container has the protocol node management IP addresses defined" on page 83.
- ESS container has the CES IP addresses defined. For more information, see "ESS container has the CES IP addresses defined" on page 84.

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 -N 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 storage 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 container has the protocol node management IP addresses defined

Before running the ESS 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 container has the CES IP addresses defined

The final item that must be defined before starting the ESS 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 container.

Instructions for deploying protocol nodes in an ESS environment

Do the following steps from the ESS 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 ems1,essio1,essio2,prt1,prt2 config load -p RootPassword
   ```

   If you have more than one node, you can specify them in a comma-separated list. Make sure that you add all ESS nodes in this **config load** command before continuing.

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 -N ess_ppc64le filesystem --suffix=-hs --ces
   ```

5. Install IBM Spectrum Scale by using the installation toolkit and set up CES.

   Use the IBM Spectrum Scale documentation for installing IBM Spectrum Scale by using the installation toolkit and for enabling the required services for the customer environment. For more information, see:

   - Using the installation toolkit to perform installation tasks: Explanations and examples.
   - Adding CES HDFS nodes into the centralized file system.
   - ESS awareness with the installation toolkit.

Appendix J. Protocol virtual machine deployment on ESS 3500 I/O nodes

Until ESS 6.1.2, the deployment of protocol nodes was supported through POWER 64 LE nodes. However, from ESS 6.1.3.1, the protocol nodes deployment is moved on ESS 3500 nodes to reduce the initial deployment cost and support the protocol deployment in the ESS 3500 environment.

From ESS 6.1.3.1, you can run a virtual machine (VM) on an ESS 3500 I/O node canister to support protocol services such as NSF and SMB, which are enabled for an ESS cluster.

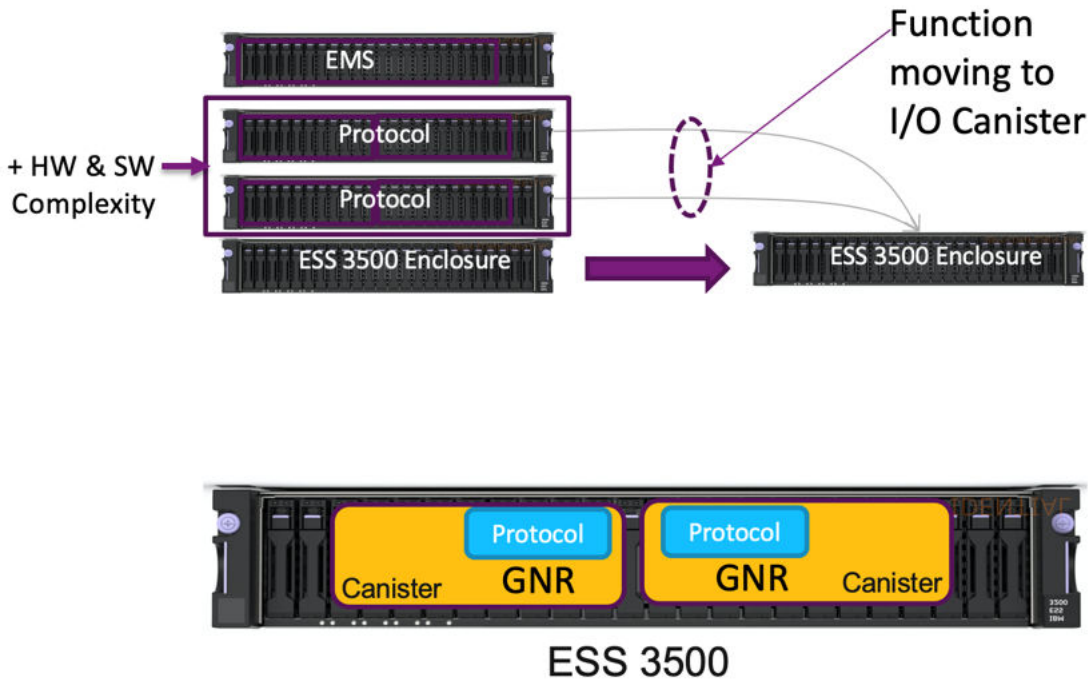


Figure 16. Protocol VM deployment on ESS 3500 I/O nodes

Hardware requirements

The hardware requirements to run protocol nodes as a VM on an I/O node are as follows:

- 8 CPU cores
- 64 GB of RAM
- 64 GB of */essvm* file system

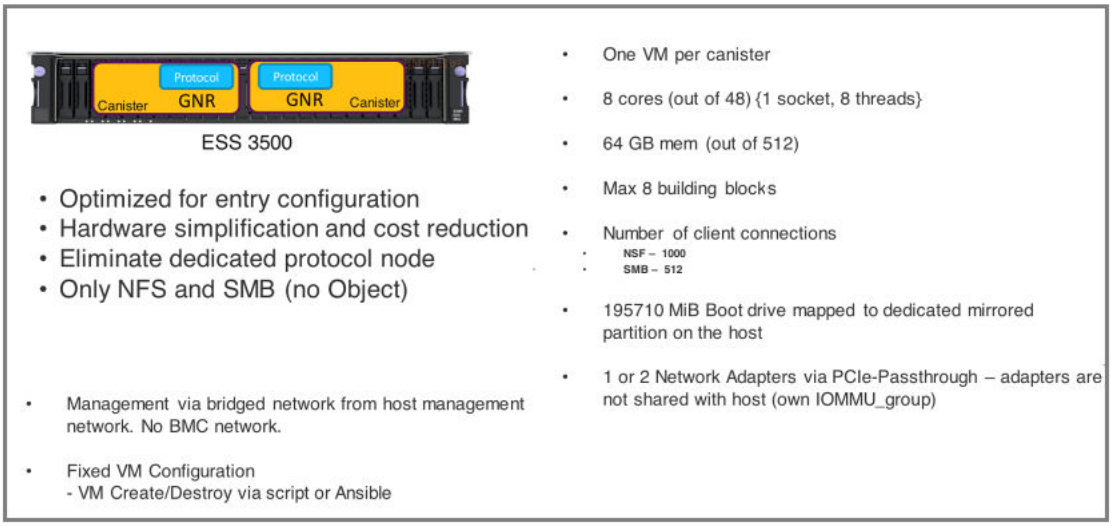


Figure 17. Hardware requirements

These requirements are provided out of the box with ESS 3500.

Protocol VM implementation

The deployment of protocol binaries and services remains the same by using the IBM Spectrum Scale install toolkit.

Ensure that the following prerequisites are met before you deploy and use a protocol VM in the ESS 3500 environment:

- ESS 3500 nodes are completely deployed.
- A CES shared root file system is created.

  See the *ESS Quick Deployment Guide* to deploy and create a CES shared root file system in the ESS 3500 or any ESS environment.

In the **essrun** command, an argument **cesvm** is introduced. The **cesvm** argument accepts the node name as a requirement for the `--create` and `--delete` options with several physical Mellanox cards, which the protocol VM uses.

Note: Only one protocol VM can be run on a single I/O node canister. Therefore, for one building block of ESS 3500, you can run two VMs per canister node.

The protocol VM implementation only supports full functional protocol deployment of NFS and SMB protocols. You cannot run the Object protocol in the protocol VM environment because of limited number of CPU cores and RAM assigned to a VM. There is no such restriction to run the Object protocol, but it might impact the performance of the VMs and produce unexpected results.

The protocol VM implementation does not support a highly scalable NFS and SMB workload. However, this implementation supports small NFS and SMB workloads.

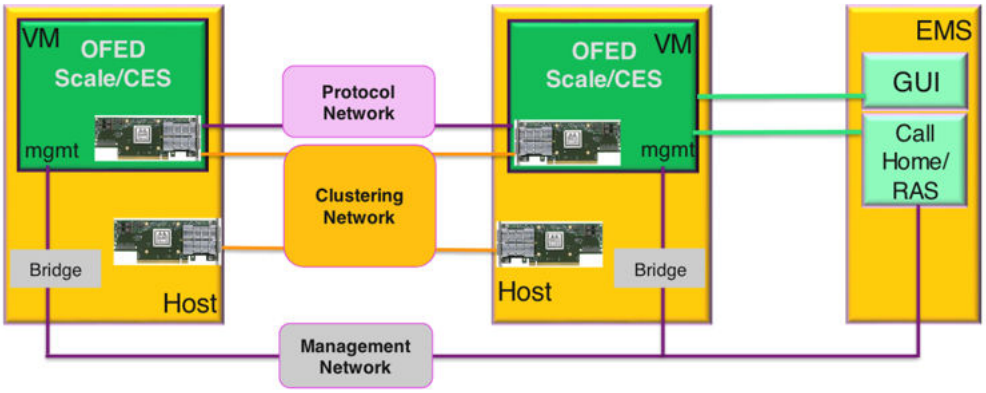
Tip: If you want to run a massive workload by using an NFS protocol and an SMB protocol, you can use the Power 64 LE version of the protocol node implementation instead the VM implementation.

For more information about POWER 64 node protocol node and implementation, contact IBM support.

Network designs

Before you set up the protocol VM, you must decide on its network aspect. Usually, ESS 3500 nodes come with two or three number of, two ports per adapter, Mellanox Ethernet or Mellanox Infiniband cards.

Figure 18. Protocol in a VM - logical network



The following figure contains slot mapping details:

Slot Number	Usage	VM Usage when Configured
Slot 1 – Network	Network: CX5 or CX6	CX5 or CX6– host IOMMU group
Slot 2 – Network	Network: CX5 or CX6	CX5 or CX6– host or guest IOMMU group
Slot 3 – Network	Network: CX5 or CX6	CX5 or CX6 – guest IOMMU group
Slot 4 – Network	Network: CX5 or CX6	CX5 or CX6 - host or guest IOMMU group

Figure 19. ESS 3500 slot placement summary

To run a protocol VM, you must assign one of the Mellanox network adapters (2 ports) to the VM that provides the following connections:

- CES (first port of the network adapter) connection
- GPFS admin or daemon (second port of the network adapter) connection

In the following figures, two types of network connections are shown:

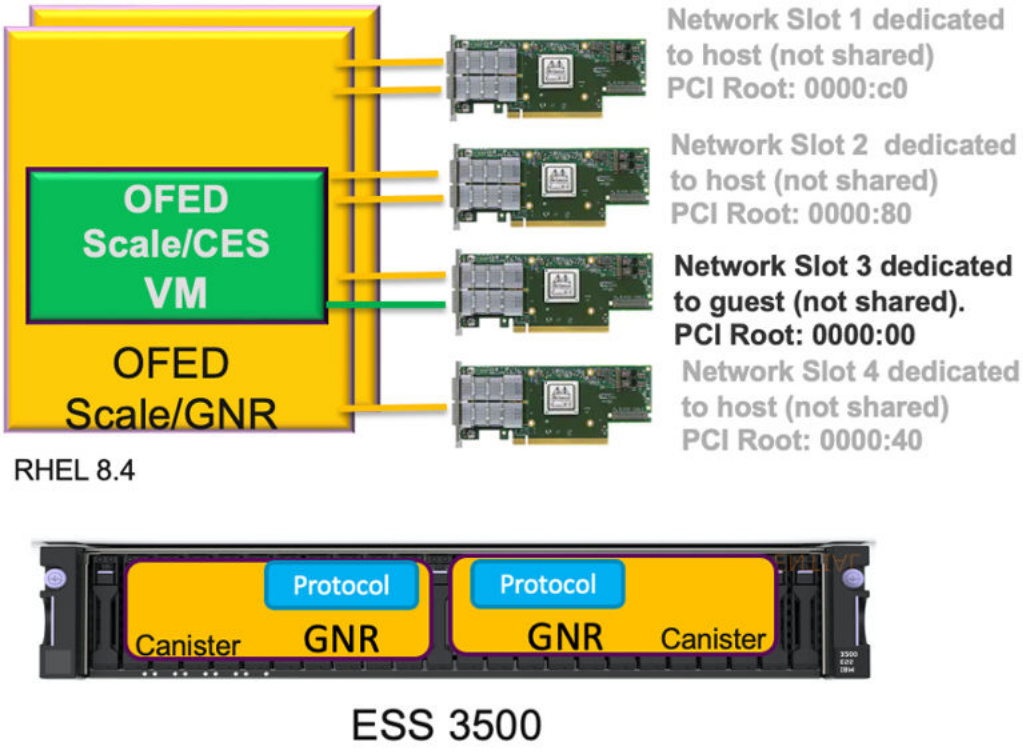


Figure 20. One dedicated card for VM

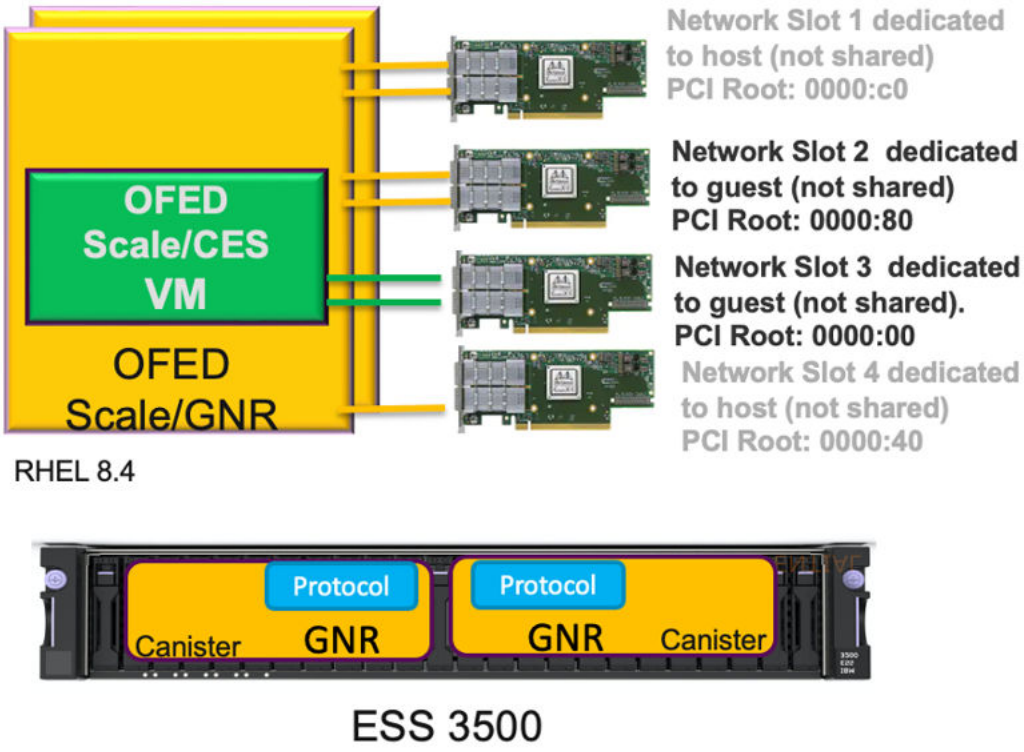


Figure 21. Two dedicated cards for VM

- Yellow is for GPFS admin or daemon network.
- Green is for CES network.

Apart from the new CES networks, the other network requirement of the ESS remains the same as the management and interlink network.

Important: When a network card is assigned to the VM, the network interfaces disappear from the host. You can see them in the VM.

The VM also needs Mellanox OFED and other requirements, such as physical I/O nodes to deploy a protocol VM. Complete the following steps before a VM protocol implementation:

1. Initialize the management network in the VM by using the **ess_instnic.essvm** command, to get the management network connectivity in the VM.
2. Update the VM by using the **yum update** command.
3. Reboot the I/O nodes after the VM operating system is updated.
4. Install OFED manually by using the **ess_ofed.essvm** command.

The essrun command

You can create a VM by using the **essrun** command.

```
# essrun cesvm --help
usage: essrun cesvm [-h] (--create | --delete) [--vm-name VMNAME] --number-of-mellanox-device-
passthru {1,2}
```

Where,

cesvm

The `cesvm` argument in the **essrun** command can either use the `--create` or `--delete` option to create or delete a VM.

-h, --help
Shows the help message and exit.

--create
Creates a CES VM on an I/O node.

--delete
Deletes a CES VM on an I/O node.

--vm-name
Shows a user-defined VM name. Default VM is *<Node Name>-essvm*. You can optionally specify the name of a VM by using the this option. However, do not provide a custom VM name unless required.

--number-of-mellanox-device-passthru 1, 2
Provides the number of Mellanox cards, minimum 1 card and maximum 2 cards, that you want to pass in a VM. Second and third cards are passed to the VM with respective number of card value.

This option pushes either 1 or 2 Mellanox network cards (each card has two network interfaces) to the VM depending on the use case. For more information about the network behavior when you pass 1 or 2 cards by using this switch, see Figure 20 on page 88 and Figure 21 on page 88. When this option is set, the network card on a PCI root device is picked up automatically and pushed to the VM.

PCI root addresses 0000:c0, 0000:80, and 0000:00 are reserved for network cards.

If you select the `--number-of-mellanox-device-passthru 1` option, the 0000:00 PCI root network card is added to the VM. Similarly, if you select the `--number-of-mellanox-device-passthru 2` option, 0000:00 and 0000:80 PCI root network cards are added to the VM. The 0000:c0 PCI root address is reserved for the host network card.

Example

This example shows how to create or delete a VM.

- Create a VM.

Before you create a VM, ensure that the */etc/hosts* have the entries for VM management network IP address and hostname mapping along with VM high-speed GPFS admin or daemon. All possible CES IP address must be in */etc/hosts*.

```
# essrun -N essio3,essio4 cesvm --create --number-of-mellanox-device-passthru 1
[INFO] Will continue with default order.
[INFO] Will continue with default order.

PLAY [Setup VM on IO node canister for CES Deployment.]
*********************************************************************************************
*********************************************************************************************
**********************************
Monday 11 April 2022  08:27:38 +0000 (0:00:00.051)       0:00:00.051 **********

TASK [Gathering Facts]
*********************************************************************************************
*********************************************************************************************
*****************************************************************
ok: [essio4]
ok: [essio3]
Monday 11 April 2022  08:27:40 +0000 (0:00:01.494)       0:00:01.546 **********
Monday 11 April 2022  08:27:40 +0000 (0:00:00.267)       0:00:01.813 **********

TASK [/opt/ibm/ess/deploy/ansible/roles/kvmCreation : Sync ESS facts | create dir]
*********************************************************************************************
*********************************************************************************************
*******
ok: [essio4]
ok: [essio3]
Monday 11 April 2022  08:27:40 +0000 (0:00:00.372)       0:00:02.186 **********
```

After a VM is created, complete the following steps to deploy a protocol node:

1. For the VM management, initialize the management network by attaching it to the VM console. Issue the following command to initialize the management network:

   ```
   virsh console essio3-essvm
   ```

   ```
   /opt/ibm/ess/tools/postscript/ess_instnic.essvm
   ```

2. Update the VM.

   ```
   yum update
   ```

3. Reboot the VM to install OFED.
4. Install Mellanox OFED.

   ```
   /opt/ibm/ess/tools/posctscripts/ess_ofed.essvm
   ```

5. Check whether the VM is ready to deploy a protocol by rebooting it.

If you want to use the CES or GPFS admin or daemon network bond in the VM, use the **essgennetwork** command to create the CES or GPFS admin or daemon network bond.

You can deploy the protocol binaries by using the IBM Spectrum Scale Install Toolkit. Download and use the **x86_64** version of install-toolkit. Designate a protocol VM as an installer node though the setup type of install toolkit must be ESS. For more information about several types of installation, see the *IBM Spectrum Scale Install Toolkit*.

- Delete the VM.

```
# essrun -N essio3,essio4 cesvm –delete  --number-of-mellanox-device-passthru 1
[INFO] The node  essio3  is not registered in the Inventory
[INFO] Will continue with default order.
[INFO] The node  essio4  is not registered in the Inventory
[INFO] Will continue with default order.

PLAY [Setup VM on IO node canister for CES Deployment.]
*********************************************************************************************
*********************************************************************************************
**********************************
Monday 11 April 2022  08:27:38 +0000 (0:00:00.051)       0:00:00.051 **********
```

```
TASK [Gathering Facts]
*********************************************************************************************
*********************************************************************************************
*****************************************************************
ok: [essio4]
ok: [essio3]
Monday 11 April 2022  08:27:40 +0000 (0:00:01.494)       0:00:01.546 **********
Monday 11 April 2022  08:27:40 +0000 (0:00:00.267)       0:00:01.813 **********

TASK [/opt/ibm/ess/deploy/ansible/roles/kvmCreation : Sync ESS facts | create dir]
*********************************************************************************************
*********************************************************************************************
*******
ok: [essio4]
ok: [essio3]
Monday 11 April 2022  08:27:40 +0000 (0:00:00.372)       0:00:02.186 **********
```

When a VM is deleted by using **–delete** command, the network interfaces are returned to the host.

Note: After a VM is created or deleted, you can run the **virsh** command to manage the VM.

For more information about the **essrun** command, see this command in the ESS Command Reference.

Appendix K. Sample scenarios with mixed ESS types

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 I, “ESS protocol node deployment by using the IBM Spectrum Scale installation toolkit,” on page 83.

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 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: **ess_6.1.4.1_0919-18_dae.sh --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 -N 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 -N 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 status of the current ESS 3000 container should be `exited`. To confirm, use the **podman ps -a** command. For example:

```
# podman container attach cems0
ESS 3000 CONTAINER root@cems0:/ # exit
exit
# podman ps -a
CONTAINER ID  IMAGE                                          COMMAND               CREATED
STATUS                     PORTS  NAMES
bfc2980bdfc9  localhost/ess_6.1.4.1_dme_xcat:0919-18  /myStartupScript....  7 days ago  Exited
(0) 2 seconds ago          cems0
```

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: **ess_6.1.4.1_0919-18_dae.sh --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
```

Note: The full list of `nodetypes` you can add to any environment are (for usage with `--nodetype`):

- For EMS node, use *ems*.
- For Legacy ESS IO ppc64le Server node, use *ess5x*.
- For ESS 3000, use *ess3k*.
- For ESS 5000, use *ess5k*.
- For ESS 3200, use *ess3200*.
- Default is *ems*.

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 I, “ESS protocol node deployment by using the IBM Spectrum Scale installation toolkit,” on page 83.

Appendix L. ConnectX-5 or ConnectX-6 VPI support

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

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

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

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

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

--query
Queries the port type of the Mellanox interface.

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

--change {InfiniBand,Ethernet}
Changes the Mellanox port type to InfiniBand or Ethernet and vice versa.

--port {P1,P2}
Specifies the port number of the Mellanox VPI card.

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

1. Query the port type of all attached devices.

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

2. Start Mellanox Software Tools (MST).

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

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

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

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

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

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

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

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

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

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

Appendix M. 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.

Appendix N. Capacity upgrade flow in ESS 5000

For customers who are looking to add storage to an existing building block, this option is now supported. The goal of a capacity upgrade is to expand a customer's available storage, either by expanding an existing file system or adding new file systems, without the need to buy an entirely new building block. Capacity upgrade is designed to be an online operation and to not interrupt customer workloads.

Supported paths

SL models (5U92)

- SL1 -> SL2
- SL2 -> SL3
- SL2 -> SL4
- SL3 -> SL4
- SL3 -> SL5
- SL4 -> SL5
- SL4 -> SL6
- SL5 -> SL6
- SL5 -> SL7
- SL6 -> SL7

SC models (4U106)

- SC1 -> SC2
- SC2 -> SC3
- SC2 -> SC4
- SC3 -> SC4
- SC3 -> SC5
- SC4 -> SC5
- SC4 -> SC6
- SC5 -> SC6
- SC5 -> SC7
- SC6 -> SC7
- SC6 -> SC8
- SC7 -> SC8
- SC7 -> SC9
- SC8 -> SC9

Prerequisites

1. All new or existing building blocks must be at ESS 6.1.0.0 or later. If there are protocol nodes in the setup, they must also be upgraded to the matching ESS version.
2. If space needs to be made, for example for moving of the EMS, this has to be planned for accordingly.
3. LBS must wear an ESD wrist band when physically working on the hardware (like plugging in SAS cables).

Capacity upgrade considerations

- Do not try to configure call home before the capacity upgrade is complete, that is until resizing is done.
- You can perform additional capacity upgrades while the DA's are rebalancing.
- You can restripe the file system while the DA's are rebalancing.
- Although it is recommended, you do not have to rebalance the file system if NSDs are added during the capacity upgrade.
- For customers who have call home enabled, remember to re-configure call home after the capacity upgrade is complete. This is the last step in the following flow or you can use the instructions in the call home configuration appendix for guidance.

SAS cable plug-in tips

- Unlatch the cable arm from the I/O server node into which you will be plugging in the SAS cable.
- Remove the blue cap.
- Make sure the location code label from the cable matches the port location code and the port number.
- Remove the cap from the SAS cable connector and plug it into the port.
- You should hear a click when the cable is inserted correctly.

SSR tasks

SSR is responsible for the following tasks.

1. Code 20 of the new enclosures - replacing parts as needed.
2. Running or labeling the new SAS cable connections.
3. Potentially making space in the frame - Moving the EMS.

SSR is not responsible for checking system health using **`essutils`** like in a rackless or a rackful solution.

LBS tasks

LBS is responsible for the following tasks.

1. Upgrade of ESS 6.1.0.0 - prior to the capacity upgrade engagement.
2. Post capacity upgrade health checks.
3. Plugging the SAS cables into the adapters and enclosures.
4. Performing capacity upgrade software functions such as conversion and resizing.
5. New storage management functions such as adding new space to existing file system and creating a new file system.
6. Restriping the file system.
7. Replacing any bad parts such as disks or cables.
8. Pre and post engagement operations

Flow

TDA process ensures that the customer is prepared for the capacity upgrade. Considerations such as if there is enough room in the rack or usage of the file system space are planned out.

LBS

1. LBS performs normal ESS software upgrade. Customer must be at ESS 6.1.0.0 for the capacity upgrade. This upgrade is treated as a separate engagement than the future capacity upgrade operation.

=== Capacity upgrade begins ===

SSR

1. The SSR arrives at the customer site. If the EMS needs to be moved, the SSR shuts down GPFS and powers down the server to move. For more information, see *Shutting down and powering up ESS* in *ESS 5.3.x Quick Deployment Guide*.
2. The SSR places the new enclosures in the rack and establishes power connection. Based on the lights, the SSR performs a code 20 operation. If lights indicate any problem, they might need to take a service action.
3. The SSR runs the new SAS cable connections and labels in a bundle and hooks them to the frame. Later when LBS comes, they simply plug in the connections when required in the flow.
4. The SSR places the EMS (if required) back into the existing frame or a new frame. Network connections and power are restored. Once the server is powered on, the SSR (or customer) can start GPFS to return the EMS back into the cluster.

LBS

1. Power on the new enclosure(s).
   - For SLx or SCx, the power cord should be connected. Press the switch to turn on the enclosures.
2. Verify that the system is converted to mmvdisk.
   a. **`mmvdisk nodeclass list`** - This command shows if the **`mmvdisk`** node class exists.

      This command can be run on any node in the cluster.
3. Upon arrival LBS should first perform the normal upgrade-related system verification steps. Run the following from the EMS:
   a. **`gnrhealthcheck`** - This command determines if there are any issues in various areas of ESS. Any problems that show up must be addressed before capacity upgrade starts. Run this command from any node in the cluster. This command only needs to be run once.
   b. **`essinstallcheck -N localhost`** - This command checks the system to ensure all components match ESS 6.1.x.x levels. Run from each node in the cluster including protocol nodes.

      Note: For ESS 5000, a check is added that flags whether the WCE bit is enabled on any drive. If the WCE bit is enabled, refer the published flash for the recommended action.
   c. **`mmhealth node show -N all --verbose`** - This command shows any system health related issues to address. Run this command from any node in the cluster.
   d. Look for any events or tips that are open in the GUI. These also show up when you issue **`mmhealth`** but it is good to check in the GUI as well. The GUI is run from the EMS node.
   e. **`/usr/sbin/opal-elog-parse -s`** - Run this command from each node in the cluster to check for any serviceable events.
4. Verification steps:
   a. **`mmgetstate -a`** - Issue this command to ensure that all daemons are active.
   b. **`mmlsmount all -L`** - Issue this command to ensure that all mount points are still up. The file system must only be mounted on the EMS and protocol nodes (if applicable).

   After these issues are resolved, capacity upgrade can begin.
5. Start by moving both recovery groups to `essio2-hs`.

   Note: The following recovery group names are examples.

   Move the recovery group in the current I/O server node to the peer I/O server node in the same building block.
   a. To list the recovery groups and the current master server, run:

      ```
      mmvdisk recoverygroup list
      ```

b. To move the recovery group from the current active I/O server node (`rg_essio1-hs`) to the peer I/O server node (`essio2-hs`) in the same building block, run the following commands in the shown order:

```
mmvdisk recoverygroup change --recovery-group rg_essio1-hs --active essio2-hs
```

Running **mmvdisk recoverygroup list** should show both RGs actively managed by `essio2-hs`.

6. Plug in the SAS cables for essio1 on the server and enclosure ends. Shut down GPFS, only on the server just modified, and then reboot the I/O node. Wait for 5 minutes for the node to reboot and paths to be rediscovered. Run the following commands to ensure that essio1 has discovered the new enclosures.

   Note: Before shutting down GPFS, make sure that autoload is turned off (**mmchconfig autoload=no**).

   a. **essstoragequickcheck -N localhost**

   b. **essfindmissingdisks -N localhost**

   Both commands should return with no issues and recognize the new enclosure and disk counts. The paths should also be without error. After this is complete, start IBM Spectrum Scale on the node in question by using **mmstartup**. After determining that IBM Spectrum Scale is `active` by using **mmgetstate** proceed to the next step.

7. Move the recovery group ownership to `essio1-hs`. Use the same commands as used in this step but make sure to use the correct node name (`essio1-hs`).

   After the preceding steps are complete, new enclosures have been successfully cabled to both servers, proceed with the following final steps.

8. Rebalance both recovery groups by running from any node in the storage cluster.

   a. **mmvdisk rg list**

   b. **mmvdisk recoverygroup change --recovery-group** *rg1* **--active essio1-hs**

   c. **mmvdisk recoverygroup change --recovery-group** *rg2* **--active essio2-hs**

   d. Check that the ownership has changed using the **mmvdisk recoverygroup list** command.

9. Perform the system verification steps again before proceeding.

10. Update enclosure and drive firmware. If there are any issues, you should stop and replace any disks or enclosures that could not be updated for some reason.

    `CurrentIoServer` implies running the command from either of I/O server nodes in the building block.

    Note: It might take up to an hour for the firmware upgrade to complete. You might notice that the fan starts to run at high speed. This is a known issue.

    a. `CurrentIoServer$` **mmchfirmware --type storage-enclosure**

    b. `CurrentIoServer$` **mmchfirmware --type drive**

    c. **mmhealth node show -N all --verbose**- This command shows any system health related issues to address. (Run from any node in the storage cluster.)

    d. **gnrhealthcheck** - This command determines if there are any issues in various areas of ESS. Any problems that show up must be addressed before capacity upgrade starts.

11. Add new storage into recovery groups.

    ```
    mmvdisk rg resize --rg rg_essio1-hs,rg_essio2-hs -v no
    ```

12. Verify that the new storage is available and the DA is rebalancing.

```
mmvdisk recoverygroup list --recovery-group rg1 --all
mmvdisk recoverygroup list --recovery-group rg2 --all
```

Run for both recovery groups. Note that the additional free space available in the DA and the background task for each DA is showing as `'rebalance'`.

On the EMS node, update the component database.

```
mmaddcompspec default --replace
```

13. Start up the GUI and use **Edit rack components** to have the GUI discover the new topologies and make changes to the frame accordingly. Changes such as modify ESS model to consume more U space, move EMS, and so on.
14. Reconfigure call home.

```
esscallhomeconf -E ems1 -N EMSNode,IONode1,IONode2 --suffix=-hs --register=all
```

At this point, discussions with the customers need to occur on what to do with the free space.

1. Add to the existing file system?
   a. See the add building block flow in *ESS 5.3.x Quick Deployment Guide* for tips on creating new NSDs and adding to an existing file system.
   b. See the add building block flow (Appendix B, “Adding additional nodes or building block(s),” on page 43) for tips on creating new NSDs and adding to an existing file system.
   c. Consider file system restripe at the end which might take time. (**mmrestripefs** *FileSystem* -b)
2. Create a new file system.
   - See the installation section on how to use **essrun** on creating a new file system from inside the container. You may also use **mmvdisk** commands directly to perform this operation.

Appendix O. Switch VLAN configuration instructions

This topic describes the instructions that are needed to configure an IBM Cumulus switch VLAN.

The IBM Cumulus switch would be preconfigured from manufacturing with proper VLAN that includes the following:

- Service/FSP/BMC VLAN
  - Blue network - Bottom ports
  - VLAN 101
- Management/Provisioning VLAN
  - Yellow network - Top ports
  - VLAN 102
- IBM Elastic Storage System special ports
  - Ports 1 - 12
  - Trunk ports
  - Default routes traffic to management VLAN
  - Packets with VLAN tag 101 routed to service network.

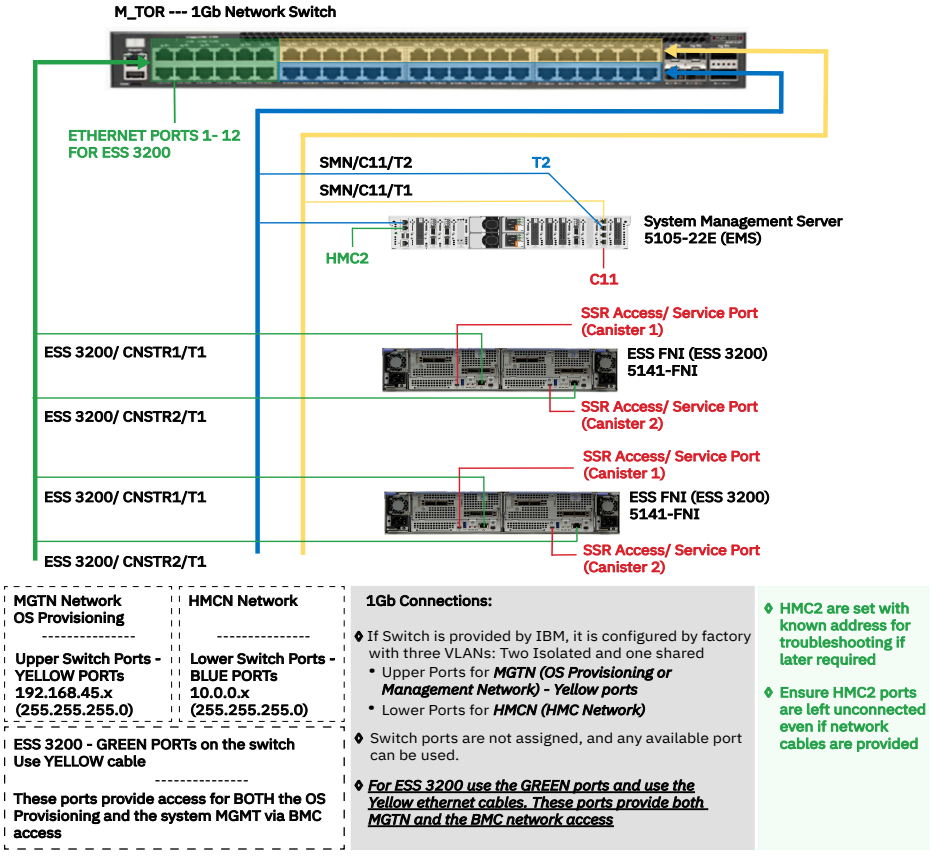


Figure 22. 1 Gb network switch

Procedure to change switch default password

Use the following steps to change switch default password.

1. Verify the 11S label at the back of the switch as shown in the following figure.

Figure 23. 11S label

Note: The required software update is cumulus-3.7.12a.

2. Log in to the switch by using the following default credentials and press the Enter key.
   - User ID: cumulus
   - Password: CumulusLinux!

   If the CumulusLinux1 default password does not work, to reset the password see the Password Recovery section in the NVIDIA documentation.
3. Use the following command to display the 11S serial number.

   ```
   cumulus@1Gsw:~$ decode-syseeprom | grep Serial | awk '{print $5}'| cut --complement -c -3
   ```

   The system displays the 11S serial number similar to the following:

   ```
   01FT690YA50YD7BGABX
   ```
4. Change the default password to the 11S password by using the following command:

   ```
   cumulus@accton-1gb-mgmt:~$ passwd
   ```

   ```
   current UNIX password: CumulusLinux!
   Enter new UNIX password: <<<Copy and paste the output provided in the 11S serial number
   display step.
   Retype new UNIX password: <<<Copy and paste the output provided in the 11S serial number
   display step.
   passwd: password updated successfully.
   ```
5. Log in through SSH or console and log in with the new 11S password to validate the changes.

   Note: The default password must be set to the 11S serial number **01FT690YA50YD7BGABX**. If not, the password must be **CumulusLinux!**.

Connect the PC to the switch console port

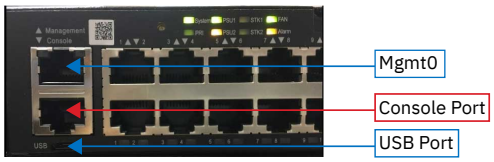


Figure 24. Switch port and switch markings

Connect the PC to the switch console port as follows:

- Connect to the switch by using RJ45 to serial cable.

Figure 25. RJ45 to serial cable and USB to serial cable

- Connect the serial end to the serial to USB cable.
- Connect the USB cable to the PC.

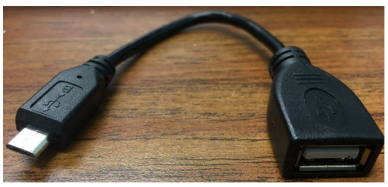


Figure 26. USB cable

Configure the host PC

Configure the host PC as follows:

1. Ensure that the driver for USB to serial cable is connected on the PC.
2. Open the device manager to verify that the COM port is used by the USB to serial cable.
3. Open `putty.exe` and use the COM port to connect to the switch.
4. Configure PuTTY to use as follows:
   a. Baud rate - 115200
   b. Parity - none
   c. Stop bits - 1
   d. Data bits - 8
   e. Flow control - none
5. Power on the switch and wait for the login prompt to show up.
6. Log in by using the following default credentials and press the Enter key.
   - User ID: cumulus
   - Password: <11S serial number>

   Note: If the switch has default Mellanox user ID and password, then proceed as follows:
   - User ID: cumulus
   - Password: CumulusLinux!
7. Download the VLAN configuration file `H48712_interfaces.rtf` from here.

   Note: If you do not have access to the above link, see "Full output of the interface file" on page 110.
8. Gain sudo rights by using the following command:

   ```
   sudo su -
   ```
9. Copy the contents of the interface file to the file name `/etc/network/interfaces` and save the file.

   Note: You can use **`vi`** or modify this file.
10. Reload the interfaces by using the following command:

    ```
    root@cumulus:/etc/network# ifreload -a
    root@cumulus:/etc/network# ifquery -a
    ```
11. Check VLAN setup.

    ```
    net show interface all
    ```
12. If required, set switch network. It is recommended to set a static IP to log remotely on the switch. For example, 192.168.45.0/24 network IP switch 192.168.45.60, gateway 192.168.45.1.
    - net add interface eth0 IP address 192.168.45.60/24
    - net add interface eth0 IP gateway 192.168.45.1
    - net pending

- net commit

13. Set the VLAN tag on each server canister. If this document is used, the tag must be 101.

```
# Set tag
/bin/ipmitool lan set 1 vlan id 101
# Confirm tag
/bin/ipmitool lan print 1 | grep -i 'VLAN ID'
```

Non-IBM Cumulus switches

If you have a non-IBM Cumulus switch, use the information above as a general reference on how to modify the switch. The key is to have a designated IBM Elastic Storage System trunk ports that are apart of both VLANs.

Modifying existing Cumulus switches

If you are converting a switch that has already non-ESS 3200 using the switch on any port in the range 1 - 12, you need to evacuate one by one those ports. If you are not using ports in the range 1 - 12, you need to apply the above process.

That means to move the cables on the upper ports in the range 1 - 12 to any free upper port that is not in the range ports 1 - 12. Equally any lower cable plugged to any port in the range 1 - 12 needs to be moved to any lower port not in the range of ports 1 - 12.

You must move one cable at the time and wait until the link LED on the destination port comes up. Once all ports in the range 1-12 are no longer cabled, you can apply the following procedure.

If an existing Cumulus switch must be modified to support IBM Elastic Storage System , the general guidance are as follows:

1. Free up at least two ports (1 IBM Elastic Storage System) on the existing switch. It is better if you can free up a block. Ideally, the current configuration is not scattered where it is easy to convert free ports for IBM Elastic Storage System usage.
2. Take the existing interfaces file from the switch and modify it for the chosen IBM Elastic Storage System ports.
3. Make the modifications to the interfaces file.

   ```
   auto swp10
   iface swp10
   bridge-pvid 102
   bridge-vids 101
   ```

   Any ports that you designate as IBM Elastic Storage System ports need to have this configuration. Consult the default IBM Elastic Storage System interfaces file for more information.
4. Copy the new interfaces file to the switch.
5. Reload and verify the interfaces.
6. Set the VLAN tags on the IBM Elastic Storage System canisters.

Full output of the interface file

`H48712_interfaces.rtf`:

```
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).
source /etc/network/interfaces.d/*.intf
# The loopback network interface
auto lo
iface lo inet loopback
# The primary network interface
auto eth0
iface eth0
    address 192.168.45.60/24
    gateway 192.168.45.1
```

```
# EVEN Ports/Lower ports PVID 101 for FSP network
auto swp14
iface swp14
bridge-access 101
auto swp16
iface swp16
bridge-access 101
auto swp18
iface swp18
bridge-access 101
auto swp20
iface swp20
bridge-access 101
auto swp22
iface swp22
bridge-access 101
auto swp24
iface swp24
bridge-access 101
auto swp26
iface swp26
bridge-access 101
auto swp28
iface swp28
bridge-access 101
auto swp30
iface swp30
bridge-access 101
auto swp32
iface swp32
bridge-access 101
auto swp34
iface swp34
bridge-access 101
auto swp36
iface swp36
bridge-access 101
auto swp38
iface swp38
bridge-access 101
auto swp40
iface swp40
bridge-access 101
auto swp42
iface swp42
bridge-access 101
auto swp44
iface swp44
bridge-access 101
auto swp46
iface swp46
bridge-access 101
auto swp48
iface swp48
bridge-access 101

# ODD Ports/Upper ports PVID 102 for management network
auto swp13
iface swp13
bridge-access 102
auto swp15
iface swp15
bridge-access 102
auto swp17
iface swp17
bridge-access 102
auto swp19
iface swp19
bridge-access 102
auto swp21
iface swp21
bridge-access 102
auto swp23
iface swp23
bridge-access 102
auto swp25
iface swp25
bridge-access 102
auto swp27
iface swp27
bridge-access 102
auto swp29
```

```
iface swp29
bridge-access 102
auto swp31
iface swp31
bridge-access 102
auto swp33
iface swp33
bridge-access 102
auto swp35
iface swp35
bridge-access 102
auto swp37
iface swp37
bridge-access 102
auto swp39
iface swp39
bridge-access 102
auto swp41
iface swp41
bridge-access 102
auto swp43
iface swp43
bridge-access 102
auto swp45
iface swp45
bridge-access 102
auto swp47
iface swp47
bridge-access 102


# ESS 3200 ports (1 to 12) FSP + OS on single physical port
auto swp1
iface swp1
bridge-pvid 102
bridge-vids 101
auto swp2
iface swp2
bridge-pvid 102
bridge-vids 101
auto swp3
iface swp3
bridge-pvid 102
bridge-vids 101
auto swp4
iface swp4
bridge-pvid 102
bridge-vids 101
auto swp5
iface swp5
bridge-pvid 102
bridge-vids 101
auto swp6
iface swp6
bridge-pvid 102
bridge-vids 101
auto swp7
iface swp7
bridge-pvid 102
bridge-vids 101
auto swp8
iface swp8
bridge-pvid 102
bridge-vids 101
auto swp9
iface swp9
bridge-pvid 102
bridge-vids 101
auto swp10
iface swp10
bridge-pvid 102
bridge-vids 101
auto swp11
iface swp11
bridge-pvid 102
bridge-vids 101
auto swp12
iface swp12
bridge-pvid 102
bridge-vids 101

# Bridge setup
```

```
auto bridge
iface bridge
bridge-vlan-aware yes
bridge-ports glob swp1-48
bridge-pvid 101
bridge-pvid 102
bridge-stp off}
```

Adding additional management switches

If the customer is out of ports on a single management switch, then a second one can be used to extend VLANs.

Connect the existing switch (Switch 1) port 45 to the new switch (Switch 2) port 47, and Switch 1 port 46 to Switch 2 port 48. It extends both the management and service VLANs ports to the new switch.

Appendix P. Dual 24 port (48 ports) MGMT switch ESS configuration

This topic describes how to configure a pair of ECS4100-28T to be the ESS top of the rack (TOR) switches.

For more information about the port, see ECS4100-28T.

This is intended for IBM manufacturing but can be also used by field engineers if needed.

When the management TOR is part of an order, IBM will deliver two of these switches as part of that order. The reason to deliver two instead of one is to keep similar number of ports available as with the 48 ports switch option.

Logical overview

From a logical perspective, the switches would look like the following figure:

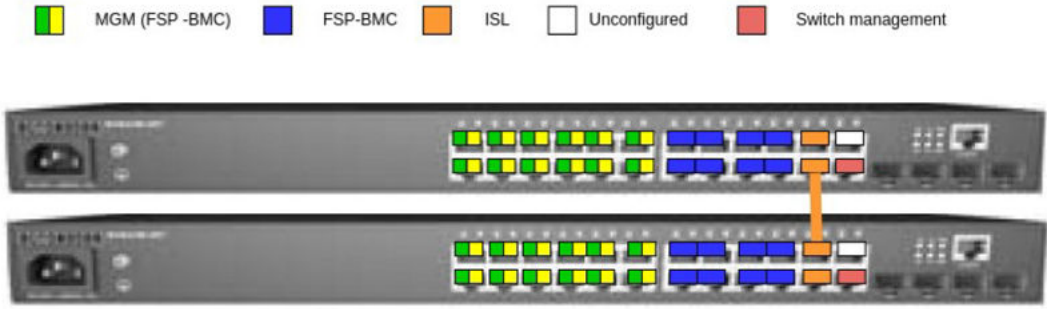


Figure 27. Logical view of two switches

The orange-colored cable shown in the figure must be connected between port 22 of the upper switch and 21 of the lower switch as a part of the configuration. That cable works as inter-switch link (ISL) between the two switches.

Ports definitions

The following are the ports definitions:

- Ports 1 through 12, named as MGM (FSP-BMC) in the green and yellow colors, are to be used for management connections to EMS and I/O nodes. These ports are used for all ESS models, regardless they have dual MAC ports (ESS 3200 and 3500) or not (ESS 5000). This is different from the previous IBM provided TOR rack where each logical color network had dedicated ports. Green and yellow ports share the same physical ports (1 through 12).
- Ports 13 through 20, named as FSP-BMC in the blue color, are to be used for systems that have dedicated FSPBMC connections. These are POWER9 EMS (both C11-T2 and HMC1 ports) and ESS 5000 I/O nodes HMC1 ports.
- Ports 21 and 22, named as ISL in the orange color, are to be used for ISL between switches only. IBM allows to extend this setup to more than two switches on a line topology, meaning the first and last switch can only have one ISL connection to the following/previous switch. The switches that are not on the edge of that line topology have both ISL ports used to their previous and following switch on the line.
- Port 23, named as Unconfigured in the white color, is not used and is shut down. This port might be used in the future.
- Port 24, named as Switch management in the red color, is to be used to access the management functions of the switch. It is intended for customer switch management network and it is set up to get an IP address via the DHCP protocol.

– The Switch management port is set as VLAN 1305 access port by default. It should work on any setup that provides an access port connected to it. If the field setup requires a different VLAN ID, change the following line:

```
VLAN 1305 name CUSTOMER media ethernet
```

– For the VLAN ID required. Match the VLAN ID with the ID in the following block:

```
interface vlan 1305
ip address dhcp
exit
```

– If you need to set a static IP address on the Switch management, replace the DHCP in the following block:

```
interface vlan 1305
ip address dhcp
exit
```

– For the IP and netmask required. In the following example, the IP address is set to 192.168.44.22 and the netmask is set to 255.255.255.0.

```
interface vlan 1305
ip address 192.168.44.22 255.255.255.0
exit
```

Personalization of the switch

To each switch, the following customization must be done. In this section, only one switch is customized but this customization must be repeated to the second and any other subsequent switch from any order.

It is assumed that the “Switch management” port is not used and the serial connection will be used. For this, you need the RJ-45 to DB-9 cable that comes with the switch. You might need extra adapters and/or converters to connect to your computer if you do not have a DB-9 connection on it.

The serial port RJ-45 is located on the top right of the switch, the serial configuration settings are 115200 bps, 8 characters, no parity, one stop bit and no flow control.

Note: The factory user is “admin” and the factory password is “admin”, if the switch is personalized already at manufacturing.

For more information, see *Quick Start Guide* in the https://www.edge-core.com/_upload/images/ECS4100_series_models_QSG_R03_20180418.pdf.

Once logged in, apply the switch configuration information to both switches.

- Copy and paste the following configuration to the switch and press Enter .
- Or, copy the contents of the file in the box (https://ibm.box.com/s/x3kzg5ykkfsu2t6536um5p0pmbg64xrl) to the switch and press Enter.
- Highlight all information and copy and paste to the switch command line after the login.

```
ip ssh crypto host-key generate
configure
ip ssh server
vlan database
VLAN 100 name ESS_MNG media ethernet
VLAN 101 name ESS_BMC media ethernet
VLAN 1305 name CUSTOMER media ethernet
exit
loopback-detection action none
no loopback-detection

interface ethernet 1/1-12
 switchport mode hybrid
 switchport native vlan 100
 switchport allowed vlan add 100 untagged
 switchport allowed vlan add 101 tagged
 switchport allowed vlan remove 1
```

```
 no spanning-tree loopback-detection
 no shutdown
 exit

interface ethernet 1/13-20
 switchport mode access
 switchport allowed vlan add 101 untagged
 switchport native vlan 101
 switchport allowed vlan remove 1
 no shutdown
 exit

interface ethernet 1/21-22
 switchport mode hybrid
 switchport native vlan 100
 switchport allowed vlan add 100 untagged
 switchport allowed vlan add 101 tagged
 spanning-tree spanning-disabled
 switchport allowed vlan remove 1
 no shutdown
 exit

interface ethernet 1/23
 shutdown
 no loopback-detection
 spanning-tree spanning-disabled
 exit

interface ethernet 1/24
 switchport allowed vlan add 1305 untagged
 switchport mode access
 switchport native vlan 1305
 switchport allowed vlan remove 1
 no shutdown
 exit

interface vlan 1305
 ip address dhcp
 exit

exit
copy running-config startup-config
```

Up to this point the configuration is always the same for every switch, the following lines are different for each switch. You need to know the serial number of the switch to continue. To get the serial, run show version command as shown in the following example:

```
Vty-1#show version
Unit 1
 Serial Number : EC2028001435
 Hardware Version : R02A
 Number of Ports : 28
 Main Power Status : Up
 Role : Master
 Loader Version : 1.0.1.9
 Linux Kernel Version : 2.6.19-g496f2361-di
 Operation Code Version : 1.2.71.203
```

In this example, the serial number EC2028001435 is used. You need to use the serial number of each switch on the commands.

```
configure
username guest password 0 EC2028001435
username admin password 0 EC2028001435
exit
copy running-config startup-config
exit
```

At this point, you are disconnected from the switch and the switch is personalized for ESS usage.

Appendix Q. Replacing all POWER8 nodes in an environment with POWER9 nodes in online mode

Replace all POWER8 nodes in your environment with POWER9 nodes in online mode by using this procedure. Cluster and file system remain up during this procedure.

Goal

The goal is to enable the customer or SLS to swap out all the POWER8 ESS nodes in the cluster with the new POWER9 (5105-22E) nodes without taking the cluster or the file system down.

Prerequisites and assumptions

- Existing POWER8 based ESS environment (GLxC – 4U106)
- Existing POWER8 EMS
- All POWER8 nodes (including EMS) will be swapped with POWER9 nodes.
- Existing cluster (at least storage nodes) must be fully updated to ESS 5.3.6.1 levels to match incoming POWER9 nodes.
- There are multiple building-blocks to properly drain BSDs, and maintain quorum, while keeping the cluster active and file system intact.
- There is enough space on the surviving NSDs to store existing data until new NSDs can be added to extend.
- Existing, supported ESS network and client infrastructure:
  - Management switch VLAN'd to spec
  - High-speed switch (Ethernet, InfiniBand, or both)
  - Client nodes with platforms supported by IBM Spectrum Scale
  - Optional POWER8 (5147-22L) protocol nodes in the same cluster as the ESS storage (2 or more protocol nodes)
  - Optional utility nodes, quorum servers, backup nodes, and so on.
- Supported ESS 5000 (POWER9 5105-22E) nodes that match 1-to-1 with the existing POWER8 nodes in the environment that will be swapped out.
- All POWER8 nodes must have the same high-speed adapter counts and port configuration as the POWER8 nodes that will be removed.
- The POWER9 servers should be able to be racked or powered without any changes to the existing infrastructure.

Examples target code levels:

- ESS 5.3.6.1 – For Legacy POWER8 environment
- ESS 6.0.1.1 – For New POWER9 environment

High-level flow

Example environment:

- 1 x POWER8 EMS
- 2 x POWER8 ESS GLxC building-blocks
  - Each building block is in its own failure group
  - Metadata replication between failure groups

- High-speed switch (IB or Ethernet)
- Remote client cluster
- POWER9 EMS (5105-22E) in a box
- 4 x POWER9 IO nodes (5105-22E) in a box

  (All P9components all have required power cables/brackets etc)
- EMS and all 4 IO nodes are quorum nodes
  - EMS is cluster and file system manager
  - Recovery groups are balanced initially per building blocks

Flow:

- Encourage customer or SLS to backup data offline as a precaution. Good practice if space to do it offcluster.
- Take a note of all IP addresses per server. LBS or SSR will need to restore when the POWER9 servers are in place. A new IP address and hostname will be needed for the POWER9 EMS

1. Ensure that the existing POWER8 environment is fully upgraded to ESS 5.3.6.1. Use the ESS 5.3.6.1 Quick Deployment Guide to perform this task.
2. **[Optional]** Upgrade client nodes to matching IBM Spectrum Scale level (5.0.5.3) and OFED (4.9x), if applicable. Move storage release=LATEST and file system format to FULL.
3. Exchange SSH keys between ESS and ESS 5000 (No xCAT running on POWER9 ESS 5000).

   Note: Shut down GUI services on the POWER8 EMS
4. Add the ESS 5000 system to existing ESS cluster.
   - Configure the recovery groups.
   - Configure the vdisk sets.
   - Configure the server lists.
5. Add ESS 5000 disks to the existing ESS file system but do not restripe.
6. Invoke **mmchdisk suspend** on all the disks coming from the ESS so that data will only be written to ESS 5000 disks.
7. Ensure that all GPFS nodes are active, quorum is maintained, NSDs are online, and file system is available. Ensure all storage hardware is healthy before proceeding.
8. Suspend and empty the NSDs from the first building block. Ensure that the data has been properly restriped to the surviving NSDs before continuing.
9. Delete the NSDs, vdisks, recovery groups, and cluster membership of drained NSD nodes (building block).
10. Verify that the old disks are empty with the **mmlsdisk** command.
11. Invoke **mmdeldisk** when all disks are emptied from ESS.
12. **[SSR]** Power off the servers and storage enclosures of the drained building block.
13. **[SSR]** Disconnect power and networking from the building block.
14. **[SSR]** Swap out the building block front the frame

    a. Insert the first pair of POWER9 nodes in their place.

    b. Cable the swapped POWER9 nodes to the ESS 5000 Spec (SAS cables, Ethernet, high speed, and so on).

    Note: POWER9 servers should go through the CSC process.
15. **[SSR]** Power on the new POWER9 building block and connected storage enclosures.
16. **[SSR]** Walk through the ESS 5000 Hardware Guide code 20 flow and set the original IP addresses per node.

17. **[SSR]** Install the POWER9 EMS and walk through the full code 20 flow. Set the new IP addresses provided by the customer.
18. **[Customer or SLS]** Log in to the POWER9 EMS and walk through the *ESS 5000 Quick Deployment Guide* to fully deploy the replaced building block.
19. **[Customer or SLS]** Create network bonds with same IP addresses removed from the POWER8 nodes and new IP address for the EMS itself.
20. **[Customer or SLS]** Add EMS and POWER9 nodes to the existing POWER8 cluster.
21. **[Customer or SLS]** Create NSDs from the POWER9 connected storage.
22. **[Customer or SLS]** Add NSDs to the existing POWER8 file system.
23. **[Customer or SLS]** Force restripe or rebalance.

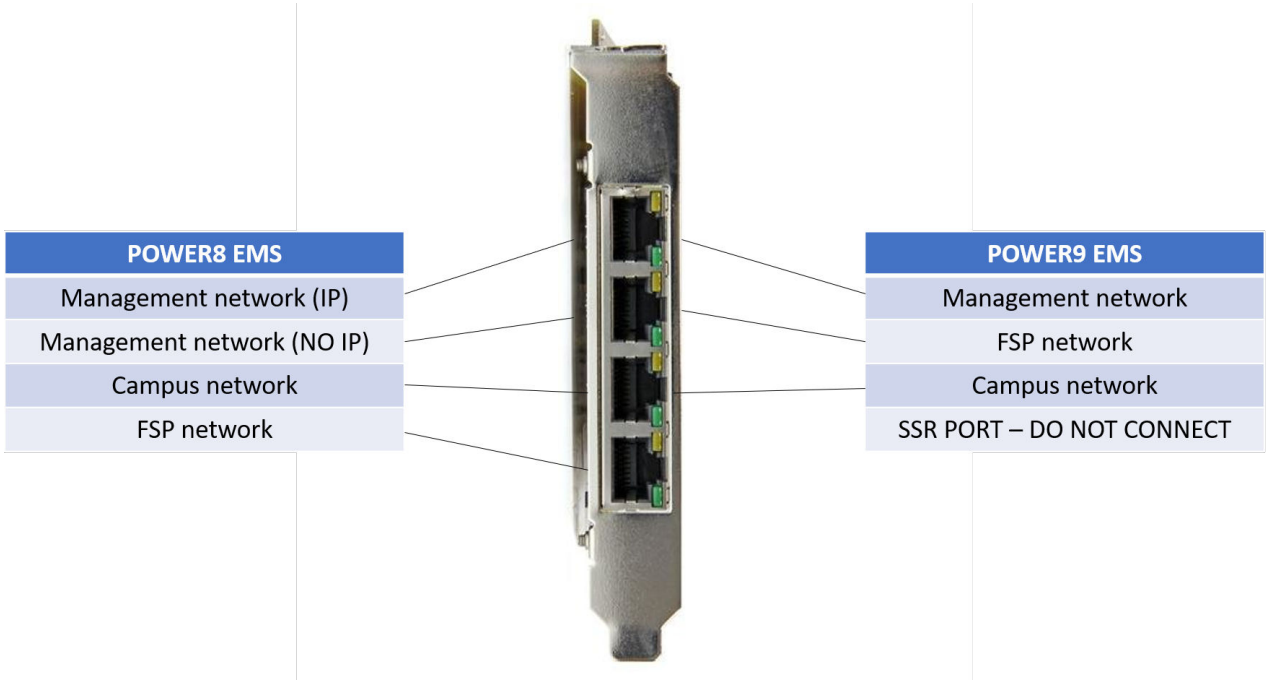
Repeat the preceding steps for the other POWER8 building blocks besides the POWER9 EMS rack and code 20 portion. When completed, all POWER8 nodes are swapped with POWER9 and a new set of NSDs in place; without losing file system or cluster access.

At the end of the procedure, the EMS node can be removed from the cluster and all POWER8 nodes can be repurposed, perhaps as client nodes.

The GUI, call home, and performance monitoring must be reconfigured to use the new EMS and cluster nodes, although IP addresses of the storage are the same. It is recommended to reconfigure all of these components afresh.

Appendix R. EMS network card port assignment

The ports on POWER8 and POWER9 EMS node network card need to be assigned as follows.



POWER8 EMS	POWER9 EMS
Management network (IP)	Management network
Management network (NO IP)	FSP network
Campus network	Campus network
FSP network	SSR PORT – DO NOT CONNECT

Appendix S. Tips for migrating from xCAT based (5.3.7.x)

The topic provides information on migrating from POWER8 xCAT based (5.3.7.x).

Note:

Refer the Support notes and rules section before proceeding with the guidance below.

A: Consider the following tips when you migrate from POWER8 xCAT based (5.3.7.x) to container based (6.1.x.x).

- You must add a new dedicated container connection to the management network to C10-T2. This connection is used by *mgmt_bridge*.
- The *fsp_bridge* uses the IP set for the FSP interface (C10-T4).
- The ESS 3000 and legacy containers both use the *fsp_bridge*.
- You must uninstall xCAT completely and the related dependencies.
- You must stop the dhcp server.
- It is recommended you reconfigure the GUI clean after conversion.
- You cannot run multiple containers at the same time.
- Only the ESS Legacy and ESS 3000 containers are supported on POWER8 EMS.
- If you have a POWER8 EMS and POWER9 EMS in the same environment, you must migrate all containers to the POWER9 EMS (and move to Scale 5.1.x.x or higher if possible).
- POWER8 GUI only officially supports hardware monitoring from POWER8 nodes (and x86 nodes).
- ESS 6.1.x.x upgrade

Table 3. RHEL kernels

ESS	Kernel
6.1.2.4	3.10.0-1160.71.1.el7
6.1.2.3	3.10.0-1160.62.1.el7
6.1.2.2	3.10.0-1160.49.1.el7
5.3.7.6	3.10.0-1160.62.1.el7
5.3.7.5	3.10.0-1160.59.1.el
5.3.7.4	3.10.0-1160.49.1.el7
5.3.7.3	3.10.0-1160.45.1.el7
5.3.7.2	3.10.0-1160.31.1
5.3.7.1	3.10.0-1160.24.1
5.3.7.0	3.10.0-1160.11.1.el7

  - It is not recommended to upgrade from ESS 5.3.7.x to ESS 6.1.1.2 anymore. Upgrade directly to ESS 6.1.2.3 or ESS 6.1.2.4. If you are updating from ESS 6.1.1.2, upgrade to 6.1.2.3 or higher (do not upgrade to 6.1.2.2).
  - For ESS 5.3.7.3, consider downgrading MOFED to MLNX_OFED_LINUX-4.9-3.1.5.3, and then convert to 6.1.2.3 or 6.1.2.4. This is to obtain full support for online upgrade when converting to RDMA core libs.

– When upgrading to 5.3.x.x, first upgrade to ESS 5.3.7.2 or ESS 5.3.7.3, and then upgrade to 6.1.2.3 or 6.1.2.4. This upgrade is to obtain full support for online upgrade when converting to RDMA core libs.

For more information about the ESS 6.1.x.x upgrade, see IBM Spectrum Scale Alert: Mellanox OFED 5.x considerations in IBM ESS V6.1.2.x+.

An output example after migrating the network from ESS 5.3.7.x before attempting to start the ESS 6.x.x.x container is as follows:

- Before creating the network bridges:

```
# ip a |grep "enP3\|bridge"
2: enP3p9s0f0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default
qlen 1000
    inet 192.168.45.20/24 brd 192.168.45.255 scope global noprefixroute enP3p9s0f0
3: enP3p9s0f1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default
qlen 1000
4: enP3p9s0f2: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default
qlen 1000
    inet 9.155.113.184/20 brd 9.155.127.255 scope global noprefixroute enP3p9s0f2
5: enP3p9s0f3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default
qlen 1000
```

```
# Sample essmgr.yml
CONTAINER:
  BKUP: /home/backup
  CONTAINER_DOMAIN_NAME: gpfs.net
  CONTAINER_HOSTNAME: cems0
  FSP_BRIDGE_IP: 10.0.0.2
  FSP_BRIDGE_NAME: fsp_bridge
  FSP_CONTAINER_IP: 10.0.0.5
  FSP_INTERFACE: enP3p9s0f3
  FSP_SUBNET: 10.0.0.0/24
  INSTALLER_HOSTNAME: ems1
  LOG: /home/log
  MGMT_BRIDGE_IP: 192.168.45.2
  MGMT_BRIDGE_NAME: mgmt_bridge
  MGMT_CONTAINER_IP: 192.168.45.80
  MGMT_INTERFACE: enP3p9s0f1
  MGMT_SUBNET: 192.168.45.0/24
```

- After creating the bridges (`./essmgr -n`):

```
# ip a |grep "enP3\|bridge"
2: enP3p9s0f0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default
qlen 1000
    inet 192.168.45.20/24 brd 192.168.45.255 scope global noprefixroute enP3p9s0f0
3: enP3p9s0f1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq master mgmt_bridge state
UP group default qlen 1000
4: enP3p9s0f2: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default
qlen 1000
    inet 9.155.113.184/20 brd 9.155.127.255 scope global noprefixroute enP3p9s0f2
5: enP3p9s0f3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq master fsp_bridge state UP
group default qlen 1000
65: mgmt_bridge: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group
default qlen 1000
    inet 192.168.45.2/24 brd 192.168.45.255 scope global noprefixroute mgmt_bridge
67: fsp_bridge: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group
default qlen 1000
    inet 10.0.0.2/24 brd 10.0.0.255 scope global noprefixroute fsp_bridge
```

B: Consider the following tips when you migrating from POWER8 EMS to POWER9 EMS.

- Running both POWER8 EMS and POWER9 EMS in same cluster is not recommended. Uninstall xCAT on POWER8 and recommission for alternative usage.
- POWER9 EMS uses C11-T1 (management) and C11-T2 (FSP).
- You must use a campus connection for POWER9 EMS (C11-T3).
- It is recommended to reconfigure the GUI and call home after conversion to POWER9 EMS.
- POWER9 EMS can see both POWER8 and POWER9 nodes (including x86) for hardware monitoring.

Appendix T. Enabling RoCE for IBM Elastic Storage Server

The topic describes setting up RDMA over Converged Ethernet (RoCE) to use within a Spectrum Scale cluster. RDMA over Converged Ethernet (RoCE) is a network protocol that allows Remote Direct Memory Access (RDMA) over an Ethernet network.

The current state of the OFED and firmware driver enables the optimization for performance and latency and simultaneously for better availability. It supports individual ports, multiple fabrics, and classical network bond to protect a service IP from network failures. With the current Mellanox OFED driver and adapter firmware, Mellanox now supports RoCEv2 over bonded interface ports from one physical NIC.

Remote Direct Memory Access (RDMA) provides a much faster access from one node's data to the other. Direct memory access from the memory of one node to the memory of another node without involving the CPU or system resources of the kernel allows more efficient use of the environment. The network and host performance improve significantly because of lower latency, lower CPU load, and higher bandwidth. In contrast, TCP/IP communications typically require copy operations, which add latency and use significant CPU and memory resources.

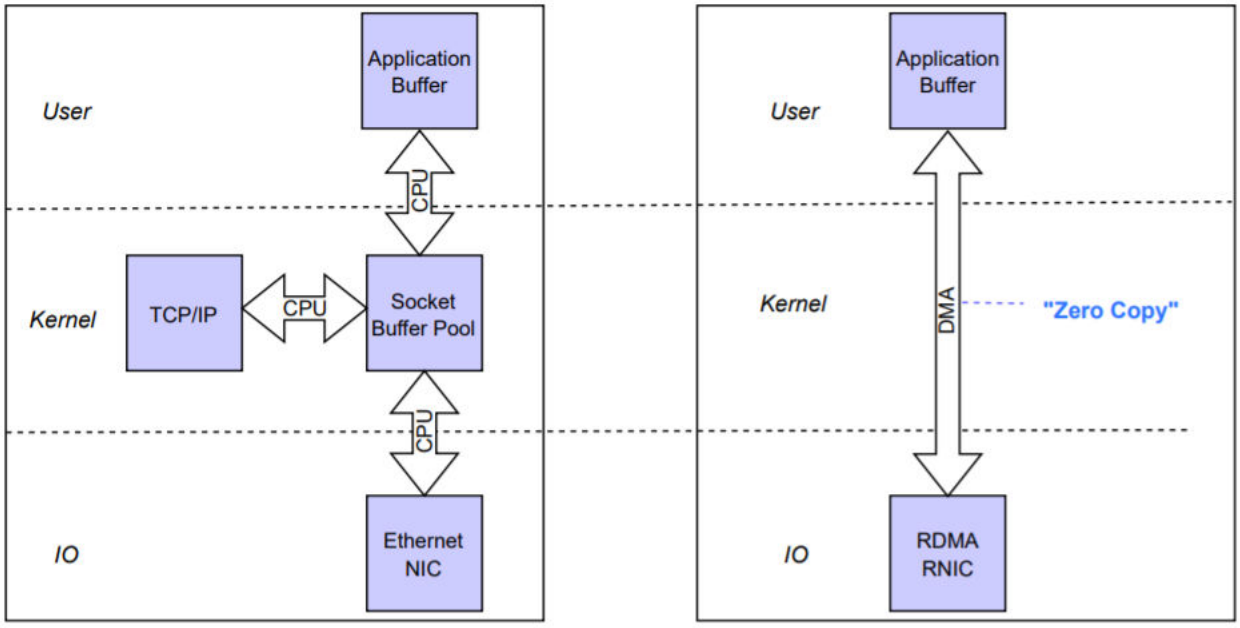


Figure 28. Comparing TCP/IP and RoCE communications

The resources, which would be typically needed to communicate by using TCP/IP are saved and can be used for real workload of the applications.

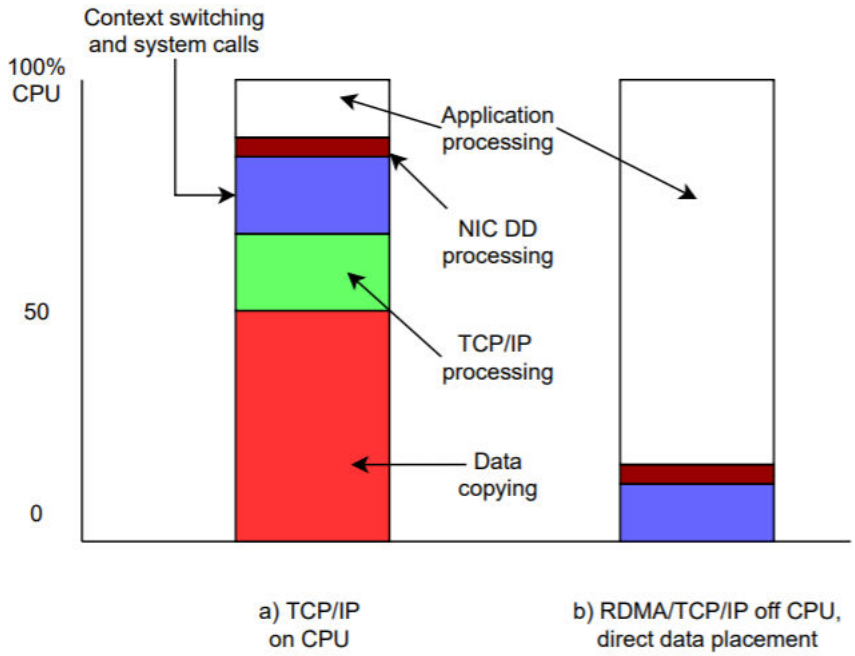


Figure 29. Comparing TCP/IP on CPU and RDMA/TCP/IP off CPU

Using RDMA has the following two major advantages.

- Ability to scale out bandwidth over multiple ports, which saves system resources.
- The higher network bandwidth results in more saving effects by using RDMA on the operating system side.

IBM Spectrum Scale support matrix

IBM Spectrum Scale supports RDMA on Red Hat Enterprise Linux only. IBM Spectrum Scale uses the VERBS programming interface to provide RDMA support. While IBM Spectrum Scale uses the VERBS programming interface for RDMA support, the underlying implementation of RDMA is vendor-specific.

IBM Spectrum Scale supports RDMA with RoCEv2 with in the following configurations:

- Supported on Red Hat Enterprise Linux only
- Supported on platform x86_64 and ppc64le
- Minimum Mellanox OFED Level 4.9 or later
- Minimum IBM Elastic Storage Server 6.1.4.1
- Minimum ESS models ESS 5000, ESS 3000, ESS 3200, and ESS 3500
- Minimum supported OS level is RHEL 8.4 for ESS 6.1.4.1

IBM Spectrum Scale RDMA support requirements and limitations

The following restrictions apply for IBM Spectrum Scale RDMA support:

- The protocols export over CES (NFS or SMB) does not use RDMA.
- Due to the TCP/IP implementation of current IBM Spectrum Scale releases, any additional network port apart from the **mmfsd** IP interface, needs to be on a different subnet. The **mmfsd** works itself for daemon-to-daemon communication.
- InfiniBand and Omni-Path partition keys are not supported by IBM Spectrum Scale.
- IPv6 must be enabled to use RoCE, if interfaces are selected by using the port name.
- IPv6 local link address needs to be conformed with `IPV6_ADDR_GEN_MODE=eui64`.

Mellanox requirements and restrictions

- RDMA is not supported on a node when both Mellanox HCAs and Intel Omni-Path HFIs are enabled for RDMA.
- You cannot use bonded ports across different adapters for use with RoCE.
- You need to have Connect X-6 adapters or later for using with RoCE.

RoCE Setup

You can deploy a RoCE environment by using one of the following methods:

- The simplest way is to use one adapter port per node that is connected to the network. It is achieved by using the same port for TCP/IP daemon communication and RoCE traffic concurrently.
- You can use multiple ports that are connected to a network. Use one port for TCP/IP daemon communication and RDMA in parallel and all the other ports for RDMA and RoCE traffic. You can scale out bandwidth capabilities of a node to the targeted numbers by using this configuration. It is more complex but powerful and flexible configuration method for deploying a RoCE environment.
- Mellanox and IBM together introduce a method to configure a network bond, consisting out of two ports from the same adapter to protect against cable, port, or switch issues.

Note: For the best performance result and reliability, configure your network as a lossless network.

Advantages and disadvantages to use or avoid bonded configuration are discussed later.

Network requirements

For running RDMA over Ethernet, you can get the best performance when the network is configured as a lossless network. Depending on the vendor, components, and the topology, the requirement for the lossless network can quickly become complex and is out of scope.

For more information, see Network configuration examples for RoCE deployment.

Make sure that all the nodes in your cluster have recent Mellanox OFED driver that is installed properly. The ESS IO server nodes are maintained by an IBM Spectrum Scale deployment. You can check the installed version by using the **ofed_info -s**command as shown in the following example:

```
# ofed_info -s
MLNX_OFED_LINUX-x.x.x.x.x
```

Note: The minimum level of OFED version is documented in the release notes of Elastic Storage Server.

Ensure that the IBM Spectrum Scale client nodes run the same MOFED level as the NSD and ESS. However, in many projects and environments, it is a challenge to maintain all nodes with the same MOFED level.

It is possible that the client nodes are running the OFED software, which is distributed by the operating system. Such configurations are simple to operate and to maintain in the client clusters. However, in cases of trouble and network glitches, such configurations can cause unexpected failures.

Network topology

The acronym host is used for an endpoint in the network. It can be an IBM Spectrum Scale client system or an ESS and NSD system.

A host's configuration depends on the number of network ports that are used or needed. The number of ports a node connects with the network depends on the expected bandwidth, the number of subnets that are needed to get access to other environments and high availability requirements.

In TCP/IP environments, scaling bandwidth with multiple ports is commonly acquired with bonding network ports, which are known as link aggregation. But bonding can have some challenging complexities. It makes the deployment complex from the network perspective.

Bonding is a commonly used technology in data centers for balancing the traffic in a network. However, bonding does not help single socket network traffic to use more than one cable. Further more, another major shortcoming of bonding is that the selection of network path is hard to predict or cannot be controlled from the application layer. Therefore, when you scale bandwidth over multiple ports, the use of RoCE is the most appropriate option.

To use a bond or not with an IBM Spectrum Scale system, the generic rule is to have at least one IP interface that is connected to the network for the daemon-to-daemon communication. A recommended network topology with bonding and RoCE is shown in the following figure.

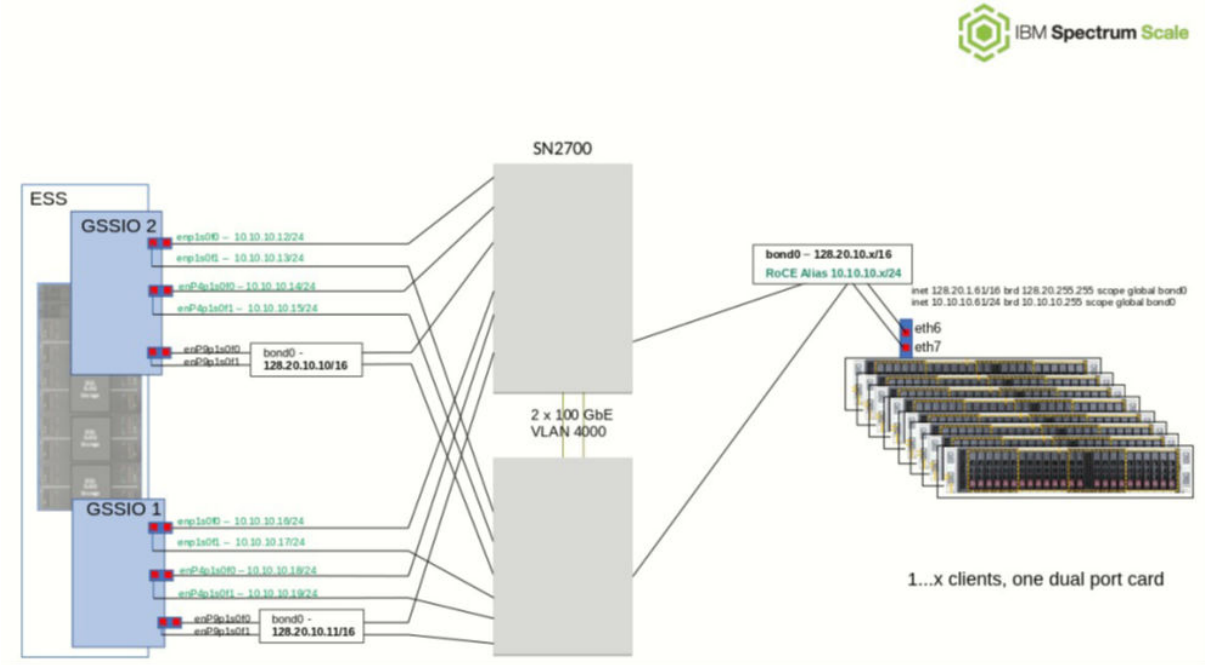


Figure 30. Displaying network topology with bonding

Depending on using a bond, it is considered that the LACP link aggregation needs to be set in the network. Alternatively, by using IBM Spectrum Scale and ESS building blocks, you can also rely on higher HA layers in IBM Spectrum Scale such as NSD and recovery group server fail over and can consider skipping bonds in your topology to make the network set up less complex. For better performance and less complex setup, it is recommended to use a configuration without bond.

For better availability, use bonded configuration. An example for a configuration without bond is shown in the following figure.

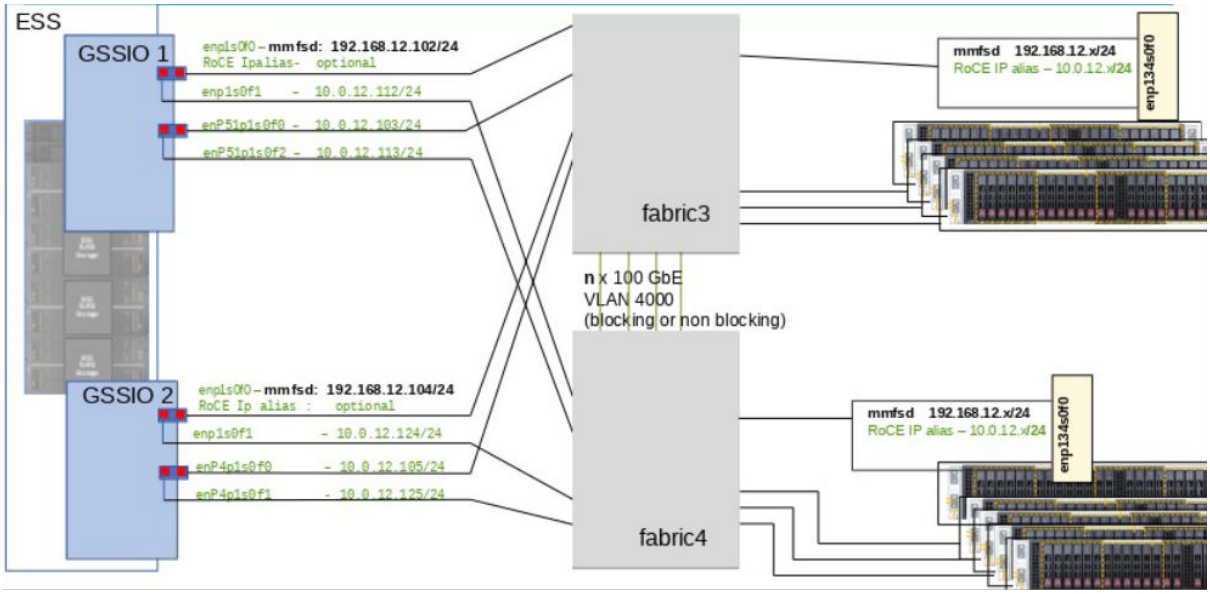


Figure 31. Showing configuration without bond

As shown in the figure, for network topology without bond, it is essential to have one IP address per network port. On the adapter that has the **mmfsd** IP address that is configured, additional RoCE IP address nor alias is needed until nodes in your cluster are able to communicate to this **mmfsd** IP address. You can configure as many aliases as the operating system version supports. You need one IP per adapter and RoCE can also use the existing IP, which is also used for TCP/IP traffic.

Running a configuration without bonds allows to later in the GPFS configuration, one can enhance the whole configuration by fabric numbers. By using such configuration, traffic on the ISL (inter-switch links) might be avoided.

Note: The adapters in a RoCE enabled environment can run TCP/IP traffic and RDMA traffic simultaneously.

MTU consideration

RDMA was first introduced on InfiniBand networks. RDMA over InfiniBand supports MTU (maximum transmission unit) sizes from 256 up to 4096 Bytes.

It is recommended to adjust the MTU to jumbo frames, which is 9000 Bytes. Adjust this setting on all adapters and all nodes, communicating in your clusters. If you need to communicate with external nodes in remote networks and you cannot be sure that the path through the network supports MTU 9000 Bytes end–to–end, make sure that `MTU Path Discovery` is enabled. The `MTU Path Discovery` is enabled by default on Red Hat Enterprise Linux.

Tip: MTU can be set for the bond only when it is being getting created by using the `--create-bond` switch. Once the bond has been created, the MTU cannot be modified by using the **essgennetwork** command. The **essgennetwork** command does not support to reconfigure the MTU for an existing bond or an interface. A user must use the **nmcli** command or manually edit the `ifcfg` network configuration file and change the MTU value. Once MTU changes are applied, reload the new connection configuration by using the **nmcli config load** command and restart the bond or the interface by using the **ifdown <interface_name>** command followed by **ifup <intreface_name>**.

CLI Reference

The CLI reference for **essgennetworks** is shown as follows:

```
# essgennetworks --help
usage: essgennetworks [-h] -N NODE-LIST [--prefix PREFIX] [--suffix SUFFIX]
                      [--interface INTERFACE] [--assignip ASSIGNIP]
                      [--create-bd | --delete-bond | --add-slave]
```

```
                      [--gateway GATEWAY] [--bond BONDNAME] [--miimon MIIMON]
                      [--vlan VLAN]
                      [--mode {balance-rr,active-backup,balance-xor,broadcast,802.3ad,balance-
tlb,balance-alb}]
                      [--hash-policy {layer2+3,layer3+4}] [--netmask CRID]
                      [--IPoIB] [--query] [--enableRDMA] [--enableRoCE]
                      [--configureRouteForRoCE]
                      [--roceRoutingTableId ROCEROUTINGTABLEID]
                      [--roceRoutingTableName ROCEROUTINGTABLENAME]
                      [--verbsPortsFabric VERBSPORTSFABRIC]
                      [--devices DEVICES] [--change {InfiniBand,Ethernet}]
                      [--port {P1,P2}] [--mtu {1500,2044,4092,9000}]
                      [--verbose]

optional arguments:
  -h, --help            show this help message and exit
  -N NODE-LIST          Provide a list of nodes for bonded network creation.
  --prefix PREFIX       Provide hostname prefix. Use = between --prefix and
                        value if the value starts with -.
  --suffix SUFFIX       Provide hostname suffix. Use = between --suffix and
                        value if the value starts with -.
  --interface INTERFACE
                        Provide list of interfaces for bond. Default to all
                        high speed interface if not provided.
  --assignip ASSIGNIP   Assign IP address to provide interface in --interface
                        switch.
  --create-bond         Create bonded interface.
  --delete-bond         Delete bonded interface.
  --add-slave           Add slave interfaces to bond.
  --gateway GATEWAY     Provide gateway for the network. By default it will
                        not configure any gateway on network interface until
                        specified.
  --bond BONDNAME       Provide name of the bond. Default bond0.
  --miimon MIIMON       Provide miimon value for bond. Default is miimon=100.
                        Consider miimon=1000 if you are planning to use bond
                        for RoCE.
  --vlan VLAN           Set VLAN_ID for the interface or bond.
  --mode {balance-rr,active-backup,balance-xor,broadcast,802.3ad,balance-tlb,balance-alb}
                        Provide bonding mode. Default is 802.3ad
                        (recommended).
  --hash-policy {layer2+3,layer3+4}
                        Provide xmit hash policy for 802.3ad and balanced-xor.
                        Default is layer2+3.
  --netmask CRID        Provide CIDR (netmask) for the interface default /24.
  --IPoIB               Enable IPoIB (IP over InfiniBand, in case InfiniBand
                        network present. By default False. It will also
                        enabler RDMA for InfiniBand.
  --query               Query the port type of the Mellanox Interface.
  --enableRDMA          Enable RDMA over the InfiniBand Network.
  --enableRoCE          Enable RoCE over the Ethernet Network.
  --configureRouteForRoCE
                        Configure routing for RoCE over the Ethernet
                        Network.Will be used if same subnet has been used for
                        different RoCE interfaces.
  --roceRoutingTableId ROCEROUTINGTABLEID
                        Routing table ID for the RoCE over the Ethernet
                        Network.
  --roceRoutingTableName ROCEROUTINGTABLENAME
                        Routing table Name for the RoCE over the Ethernet
                        Network.
  --verbsPortsFabric VERBSPORTSFABRIC
                        Name of the Mellanox verbs port fabric. For Example: 1
                        or 2. It will be automatically added to the verbs
                        port.
  --devices DEVICES     Name of the Mellanox device name. 'all' will query all
                        devices attached to node. Provide comman separated
                        device names to query mode than one device at a given
                        time.
  --change {InfiniBand,Ethernet}
                        change the Mellanox port type to InfiniBand or
                        Ethernet and vice versa.
  --port {P1,P2}        Port number of the Mellanox VPI card.
  --mtu {1500,2044,4092,9000}
                        Provide mtu of bond network. For Ethernet, 1500 or
                        9000 MTU allowed (Default: 1500). For InfiniBand, 2044
                        or 4092 MTU allowed (Default: 2044).
  --verbose             Provides more verbosity.

This program can be used to create a bonded network. Example: essgennetworks
-G ess_x86_64
```

Bond configuration

For RoCEv2 to work properly, all interfaces need an Ipv6 local link address and an IPv4 address.

Configuring bond interfaces is not needed to run RoCE. If you do not need a bonded TCP/IP interface, you can proceed with next section.

In case you want to have your IP interfaces protected against port failures, run the following steps to configure the bonds.

Note: If you do not need the daemon IP address to be protected by a bond, skip this step.

Note: To configure RoCE over bond, the bond must be created prior by using **essgennnetwork** command. If a bond is created by using more than two physical network interfaces, then the existing bond must be broken and recreated by using same physical card. For example, if a bond is created by using four ports and two physical cards, then the bond must be broken and a new bond must be created by using two ports with the same physical card.

Refer to the following example of enabling RoCE over bond.

1. Create a bond by using two interfaces by using the following command:

```
# essgens -N essio51 --suffix=-ce --interface enp1s0f0,enp1s0f0f1 --create-bond --hash-
policy layer3+4 --bond bond0 --mtu 9000 --miimon 1000
2021-10-25T03:30:24.978336 [INFO] Starting network generation...
2021-10-25T03:30:25.040319 [INFO] nodelist:  essio51
2021-10-25T03:30:25.040386 [INFO] suffix used for network hostname: -ce
2021-10-25T03:30:25.040430 [INFO] Bond will be created using --hash-policy layer3+4 which is
required to configure RoCE over Bond.
2021-10-25T03:30:25.040464 [INFO] Bond will be created using miimon 1000
2021-10-25T03:30:25.040495 [INFO] Bond will be created with --mtu 9000. Make sure switch is
configured to use 9000 MTU or set to auto negotiate.
2021-10-25T03:30:25.040526 [WARN] Make sure switch is configured to handle Jumbo frame
before creating bond.
2021-10-25T03:30:25.040556 [WARN] Without Jubmo frame configuration at switch may result to
network outage.
2021-10-25T03:30:25.040586 [WARN] Consider testing once configured. Example: ping -s 9000 -c
10 TARGET_NODE-hs
2021-10-25T03:30:50.356662 [INFO] Interface(s) available on node essio51-ce
2021-10-25T03:30:50.356748 [INFO] Considered interface(s) of node essio51-ce are
['enp1s0f0'] with RDMA Port ['mlx5_0'] for this operation
2021-10-25T03:30:50.360236 [INFO] Checking for IP address assignment on node(s)
2021-10-25T03:30:50.364861 [INFO] essio51-ce: Current IP Address: IP not assigned
2021-10-25T03:30:50.367996 [INFO] Creating network bond bond0 on node essio51-ce with IP
Address 192.168.2.51
2021-10-25T03:30:50.670300 [WARN] essio51-ce: Bond created with one slave interface
2021-10-25T03:30:51.734465 [INFO] Network bond bond0 on node essio51-ce with IP Address
192.168.2.51 has been created successfully.
2021-10-25T03:30:51.738006 [INFO] Network Bond with name bond0 is UP and connected at node
essio51-ce
2021-10-25T03:30:51.738055 [INFO] Bond creation complete. Reloading connections.
```

   Note: A bond is created by using the hash policy layer3+4.

   Tip: MTU can only be set for the bond while it's getting created using the `--create-bond` switch. Once the bond has been created, the MTU cannot be modified by using the **essgennetwork** command. The **essgennetwork** command does not support to reconfigure the MTU for an existing bond or an interface. A user must use the **nmcli** command or manually edit the `ifcfg` network configuration file and change the MTU value. Once MTU changes are applied, reload the new connection configuration by using the **nmcli config load** command and restart the bond or the interface by using the **ifdown <interface_name>** command followed by **ifup <intreface_name>**.

2. Once the bond is created, then the `bond0` is used to enable the RoCE by using the following command:

```
# essgennetworks -N ems5 --suffix=-ce --bond bond0 --enableRoCE
2021-10-25T02:13:32.989606 [INFO] Starting network generation...
2021-10-25T02:13:33.053310 [INFO] nodelist:  ems5
2021-10-25T02:13:33.053370 [INFO] suffix used for network hostname: -ce
2021-10-25T02:14:26.325569 [INFO] Interface(s) available on node ems5-ce
2021-10-25T02:14:26.325636 [INFO] Considered interface(s) of node ems5-ce are ['bond0'] with
RDMA Port ['mlx5_bond_0'] for this operation
2021-10-25T02:14:26.693602 [INFO] Supported Mellanox RoCE card found at node ems5
```

```
2021-10-25T02:14:26.697197 [INFO] Supported version of Mellanox OFED found at node ems5-ce
2021-10-25T02:14:40.007004 [INFO] Bond validation passed and found bonds bond0 has been
created using same physical network adapter at node ems5-ce
2021-10-25T02:14:40.011375 [INFO] Bond MTU validation passed and found bonds MTU set to 9000
at node ems5-ce
2021-10-25T02:14:40.015525 [INFO] Interface bond0 have the IPv4 Address assigned at node
ems5-ce
2021-10-25T02:14:40.019593 [INFO] Interface bond0 have the IPv6 Address assigned at node
ems5-ce
2021-10-25T02:14:40.019648 [INFO] Enabling RDMA for Ports ['mlx5_bond_0']
2021-10-25T02:14:48.370397 [INFO] Enabled RDMA i.e. RoCE over Ethernet using bond bond0
2021-10-25T02:14:48.370446 [INFO] Please recycle the GPFS daemon on those nodes where RoCE
has been enabled.
```

Note: GPFS daemon must be recycled to make the RoCE configuration working. Once daemon recycled, you can run the **mmdiag –network** command to check whether the RDMA is enabled over Ethernet.

Advantages and disadvantages of using bond with RoCE:

- A bonded interface protects against port and cable failures.
- For running RDMA, all ports of the bonded interface need to be on one, the same, physical PCI adapter. So, an adapter failure is not covered such configurations.
- Creating bonds over multiple switches makes MLAG configuration mandatory in a network, which can cause unbalanced network utilization in the fabric.

Regular IP interface configuration

For RoCEv2 to work properly, all interfaces need an Ipv6 local link address and an IPv4 address. Configure all interfaces intended to use for RoCE communication, as shown in the following example.

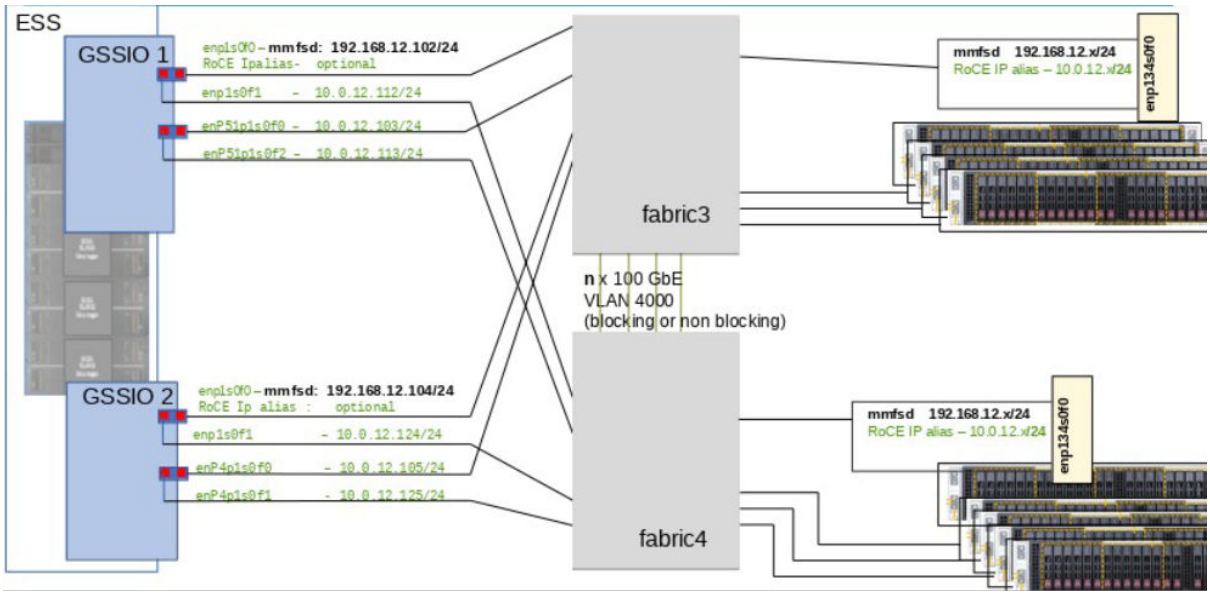


Figure 32. Configuring interfaces

As highlighted in black, make sure that only one IP interface has an IP address for the **mmfs** daemon communication. All other interfaces need to be on a different subnet.

1. Assign IPv4 to high-speed interface by using the following command:

```
# essgennetworks -N essio51 --suffix=-ce --interface enP48p1s0f0 --assignip 192.168.4.51
2021-10-25T04:40:58.779195 [INFO] Starting network generation...
2021-10-25T04:40:58.842575 [INFO] nodelist:  essio51
2021-10-25T04:40:58.842642 [INFO] suffix used for network hostname: -ce
2021-10-25T04:41:22.178448 [INFO] Interface(s) available on node essio51-ce
2021-10-25T04:41:22.178533 [INFO] Considered interface(s) of node essio51-ce are
['enP48p1s0f0'] with RDMA Port ['Inbuilt'] for this operation

2021-10-25T04:41:22.178591 [INFO] Checking for IP address assignment on node(s)
```

```
2021-10-25T04:41:22.182983 [INFO] essio51-ce: Current IP Address: IP not assigned

2021-10-25T04:41:22.183057 [INFO] Interface name enP48p1s0f0 consider for IP assignment with
IP address 192.168.4.51 for node essio51-ce
2021-10-25T04:41:22.308265 [INFO] Successfully assigned IP Address to interface enP48p1s0f0
at node essio51-ce
2021-10-25T04:41:22.506151 [INFO] IP Assignment successful. Reloading connections.
[PASS] essgennetworks passed successfully
```

2. If you are using only one subnet, no need to enable routing. For multiple networks in same subnet, must configure routing to route the RoCE traffic by using all subnets. To enable routing for a subnet, see the “Routing configurations” on page 135 section. In the following example, only one subnet 192.168.4.0/24 is used. Therefore, you can enable RoCE by using the **–enableRoCE** command with the **–interface** switch. However, it is safe to enable routing even only one subnet is in use, by issuing the following command:

```
# essgennetworks -N essio51 --suffix=-ce --interface enP48p1s0f0 --enableRoCE
2021-10-25T04:43:15.983035 [INFO] Starting network generation...
2021-10-25T04:43:16.045174 [INFO] nodelist:  essio51
2021-10-25T04:43:16.045243 [INFO] suffix used for network hostname: -ce
2021-10-25T04:43:39.040751 [INFO] Interface(s) available on node essio51-ce
2021-10-25T04:43:39.040829 [INFO] Considered interface(s) of node essio51-ce are
['enP48p1s0f0'] with RDMA Port ['mlx5_2'] for this operation
2021-10-25T04:43:39.348322 [WARN] MTU is NOT set to 9000 for interface enP48p1s0f0 at node
essio51. It's advisible to set MTU as 9000 for better performance.
2021-10-25T04:43:39.348381 [WARN] Interface MTU NOT set to 9000 at node essio51-ce
2021-10-25T04:43:39.348413 [WARN] Please re-configure the interface with --mtu 9000 and make
sure switch is also configured to use --mtu 9000 i.e. Jumbo frame for better performance at
node essio51-ce
2021-10-25T04:43:39.352619 [INFO] Interface enP48p1s0f0 have the IPv4 Address assigned at
node essio51-ce
2021-10-25T04:43:39.356763 [INFO] Interface enP48p1s0f0 have the IPv6 Address assigned at
node essio51-ce
2021-10-25T04:43:39.356817 [INFO] Enabling RDMA for Ports ['mlx5_2']
2021-10-25T04:43:45.409405 [INFO] Enabled RDMA i.e. RoCE over Ethernet using Ethernet
interfaces enP48p1s0f0
2021-10-25T04:43:45.409464 [INFO] Please recycle the GPFS daemon on those nodes where RoCE
has been enabled.
```

Note: If you want to enable RoCE for bond and interfaces both, then use **–interface** and **–bond** switch with the **–enabelRoCE** to enable it.

GPFS daemon must be recycled to make the RoCE configuration working. Once daemon recycled you can run the **mmdiag –network** command to check whether the RDMA is enabled over Ethernet.

Tip: The multiple `verbsPortsFabric` configuration is not supported as of now by using the **essgennnetwork** command. For example, if a user wants to configure fabric for bond as "/1/1" and for interfaces "/1/2", then the **essgennetwork** command does not have the option to configure it. The user needs to use the **mmchconfig verbsPorts="mlx5_bond_0/1/1 mlx5_2/1/2 mlx5_3/1/2** command to enable the multiple fabrics for RoCE. GPFS daemon must be recycled after the **mmchconfig verbsPorts** parameter is used.

Routing configurations

In Figure 32 on page 134, the **mmfsd** communication runs in the 192.168.12.0/24 network. All other Mellanox cards interfaces are highlighted in green and configured to be in the subnet 10.10.10.x/24. All interfaces are intended to use for RDMA communication, while only the IP addresses in 192.168.12.0/24 network are used for TCP/IP communication.

With current IBM Spectrum Scale releases, only one IP address per node is supported for communication to the **mmfs** daemon's network.

To scale out over multiple ports, RoCE can be used. But according to the definition of OFED standards, each RDMA interface needs to have an IP address to maintain the connection.

Theoretically, to use multiple interfaces, need to configure multiple subnets. It is complex for larger environments and technically not needed. It is possible to configure all interfaces that are intended to be used for RDMA into one separate subnet.

As a limitation of the Red Hat Enterprise Linux kernel, multiple interfaces to the same subnets are complex. When you have more than one interface per subnet on a node, answering arp requests, selecting rules for outgoing IP traffic, and other complexities need to be reconsidered. To resolve such issues, you must add some special routing tables, described in routing table entries. You need to specify routing entries for each interface and for all subnets, where you have more than one interface connected.

In example, each IO server node has four interfaces that are connected to the same physical network. These interfaces need to get an IP address in a different IP-range(subnet) than the **mmfsd** but they are all in the same network. Refer to the following figure for topology overview.

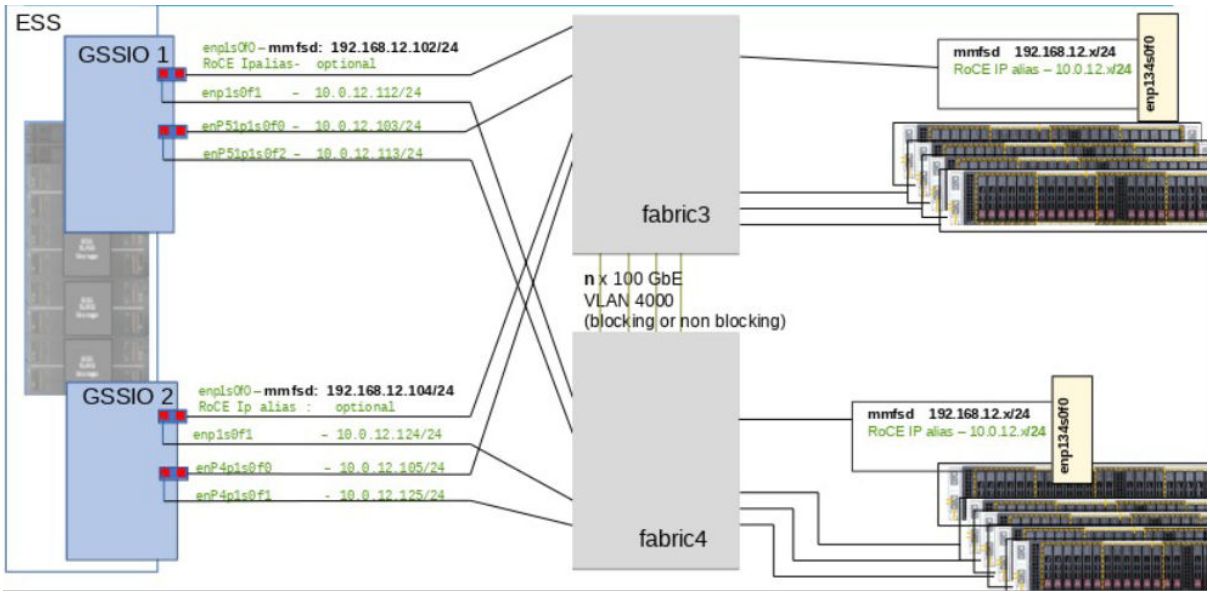


Figure 33. Displaying topology overview

For each green marked interface, a routing entry is needed.

1. To enable routing for each RoCE interface, issue the following command:

```
# essgennetworks -N essio51 --configureRouteForRoCE --roceRoutingTableId 101 --
roceRoutingTableName t1 --interface enP48p1s0f0
2021-10-25T04:51:53.048486 [INFO] Starting network generation...
2021-10-25T04:51:53.111121 [INFO] nodelist:  essio51
2021-10-25T04:52:13.897913 [INFO] Interface(s) available on node essio51
2021-10-25T04:52:13.897991 [INFO] Considered interface(s) of node essio51 are
['enP48p1s0f0'] with RDMA Port ['mlx5_2'] for this operation
2021-10-25T04:52:14.202172 [WARN] MTU is NOT set to 9000 for interface enP48p1s0f0 at node
essio51. It's advisible to set MTU as 9000 for better performance.
2021-10-25T04:52:14.202235 [WARN] Interface MTU NOT set to 9000 at node essio51
2021-10-25T04:52:14.202268 [WARN] Please re-configure the interface with --mtu 9000 and make
sure switch is also configured to use --mtu 9000 i.e. Jumbo frame for better performance at
node essio51
2021-10-25T04:52:14.206459 [INFO] Interface enP48p1s0f0 have the IPv4 Address assigned at
node essio51
2021-10-25T04:52:14.210723 [INFO] Interface enP48p1s0f0 have the IPv6 Address assigned at
node essio51
2021-10-25T04:52:14.225432 [INFO] RoCE routing validation passes.
2021-10-25T04:52:14.225488 [INFO] Setting UP routing for interface RDMA for Interface
['enP48p1s0f0'] at node essio51
2021-10-25T04:52:14.228225 [INFO] Successfully added routing id and it's table name
inside /etc/iproute2/rt_tables at node essio51
2021-10-25T04:52:14.672155 [INFO] Successfully configured route for interface enP48p1s0f0 at
node essio51
2021-10-25T04:52:14.672219 [INFO} Routing configuration for subnet has configured
successfully.
```

2. Following is the routing table entry after previous step:

```
# ip r
10.111.222.100/30 dev enP1p8s0f3 proto kernel scope link src 10.111.222.101 metric 101
linkdown
192.168.2.0/24 dev bond0 proto kernel scope link src 192.168.2.51 metric 300
```

```
192.168.4.0/24 dev enP48p1s0f0 proto kernel scope link src 192.168.4.51 metric 104
192.168.15.0/24 dev enP1p8s0f0 proto kernel scope link src 192.168.15.51 metric 100
```

Repeat same steps for all the interface to configure multiple RoCE interfaces.

Note: `sysctl` settings:

In addition to the interface scripts, you need system-wide settings that are managed by `sysctl`. Depending on the ESS version in use, a customized `sysctl` setting is available by a tuned profile, which is named as scale. You need to edit the file `/etc/tuned/scale/tuned.conf`. For any other client node in your cluster, you can deploy the same `sysctl` configuration file.

Make sure the following `sysctl` settings are applied:

```
net.ipv6.conf.all.disable_ipv6=0
net.ipv6.conf.default.disable_ipv6=0
```

Note: The IPv6 configuration maybe enabled by the default ESS deployment process.

Validation:

Run **`mmlsconfig`** to validate the `verbsPorts` configuration:

```
# mmlsconfig
[ems5-ce,essio51-ce,essio52-ce]
verbsRdma enable
verbsRdmaCm enable
verbsRdmaSend yes
[ems5-ce,essio52-ce]
verbsPorts mlx5_bond_0
[essio51-ce]
verbsPorts mlx5_2
```

In addition, you can run the **`mmdiag –network`** and check for the `verbsRDMA` connection.

Appendix U. Enabling goconserver for ESS 5000

In ESS 6.1.4.1, IPMI over LAN console is centralized on the EMS system. However, you still need to configure it manually in this version.

To configure the console server, you need to perform the following steps on the EMS (that must be on 6.1.4.1 or later):

1. To configure `goconserver`, issue the following command:

   ```
   # esscongocfg --task configure
   ```

2. To check whether nodes are configured, issue the following command on EMS:

   ```
   # esscongocfg --task list
   ```

3. After a system is added to the console server, do not use the IPMI console to connect the serial over LAN (SOL) to configured nodes. Access the nodes by issuing the following command:

   ```
   # congo console node_name
   ```

In addition to access the console from EMS, the EMS keeps a file for each node with all the output of each server on the `/var/log/goconserver/nodes` directory. By issuing the following commands, you can get log files:

```
# ls -ltr /var/log/goconserver/nodes
```

A sample output is as follows:

```
total 28
-rw------- 1 root root 3818 Aug 12 08:11 gssems2-new.log
-rw------- 1 root root 3984 Aug 12 08:11 gssio3-new.log
-rw------- 1 root root 4150 Aug 12 08:11 gssio4-new.log
-rw------- 1 root root 7755 Aug 12 08:21 essio3.log
-rw------- 1 root root  100 Aug 12 08:22 essio4.log
```

For more information, see Configuring goconserver.

Appendix V. Enabling goconserver for ESS x86 nodes

In ESS 6.1.2.4, IPMI over LAN console is centralized on the EMS system. However, you still need to configure it manually in for x86 nodes.

To configure the console server, complete the following steps on the EMS:

1. To configure goconserver, issue the following command:

```
# /bin/congo create essio4 driver=cmd --params cmd="/usr/bin/ipmitool -I lanplus -H
172.16.0.6 -U admin -P admin sol activate"
```

2. To check whether nodes are configured, issue the following command on EMS:

```
# esscongocfg --task list
```

3. After a system is added to the console server, do not use the IPMI console to connect the serial over LAN (SOL) to configured nodes. Access the nodes by issuing the following command:

```
# congo console node_name
```

In addition to access the console from EMS, the EMS keeps a file for each node with all the output of each server on the /var/log/goconserver/nodes directory. By issuing the following commands, you can get log files:

```
# ls -ltr /var/log/goconserver/nodes
```

A sample output is as follows:

```
total 28
-rw------- 1 root root 3818 Aug 12 08:11 gssems2-new.log
-rw------- 1 root root 3984 Aug 12 08:11 gssio3-new.log
-rw------- 1 root root 4150 Aug 12 08:11 gssio4-new.log
-rw------- 1 root root 7755 Aug 12 08:21 essio3.log
-rw------- 1 root root  100 Aug 12 08:22 essio4.log
```

For more information, see Configuring goconserver.

Appendix W. POWER8 to POWER9 EMS container conversion

The topic describes the process of moving from the POWER8 to the POWER9 EMS regarding consolidation of the containers. The current requirement is to decommission the POWER8 EMS if you are in a Scale 5.1.x.x environment. It is suggested to consolidate to a single POWER9 EMS. It reduces complexity (single GUI, single deployment environment, more simple networking) and is the tested environment.

Requirements for POWER8 to the POWER9 EMS container conversion.

- Maintain Quorum by initially adding the POWER9 EMS to the POWER8 EMS environment.
- Cleanup and shutdown the POWER8 EMS.
- Reconfigure the GUI and collector.
  For stretch clustering assign two or more collectors, one for each cluster to set up GUI correctly. For more information about collectors, see the *Configuring multiple collectors* section in IBM Spectrum Scale: Problem Determination Guide.

Note: Therefore, in the most cases, for larger environments EMS is needed for quorum add the POWER9 EMS first.

Refer to the following steps for POWER8 to the POWER9 EMS container conversion.

Note: These steps are the broad level steps, for full commands, see *Quick Deployment Guide*.

POWER9 EMS steps.

1. Code 20 the POWER9 EMS (clean hardware, management, and FSP networks set). POWER9 EMS can talk over the management network to the other nodes in the cluster with high-speed cables run.
2. Using a supported POWER9 container (any can work) deployment the container, run config load, config check, and EMS update by using the **(essrun config load | check | update --offline)**.
3. Create high-speed network bonds by using (**essrun network**).
4. Add EMS to cluster by using the **essrun cluster --add-ems** command.
5. The POWER9 EMS is now part of the cluster and can be a quorum node. Use the **mmlscluster** to confirm.
6. Set up node as a second collector node, as shown in the following example:

```
mmperfmon config show
         mmperfmon config update { [--collectors Collector-Node[,CollectorNode...] ]
               [ --config-file InputFile]
```

POWER8 EMS steps.

1. Remove collector attribute from the node.
2. Shutdown ESA; shutdown GUI.
3. Remove node from cluster.
4. Shutdown GPFS.
5. Remove OFED.
6. Remove GPFS.
7. Remove xCAT.
8. Run cleanup script (removes keys, logs, and config files).
9. Clear `ifcfg` files.

After you complete these steps, POWER8 EMS is no longer in use.

Optional: Can use script to secure wipe boot drives.

Final POWER9 steps.

1. Reenter the container and run GUI config setup (sets up `hosts.py`).
   a. To configure GUI hardware monitoring from the container, issue the following command:

      ```
      essrun -N <nodes separated by comma:ems1,essio1,essio2> gui --configure
      ```

   b. To create GUI user (from EMS host, not from the container), issue the following command from EMS host:

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

   c. Issue the following command to start GUI:

      ```
      systemctl restart gpfsgui
      ```

      For more information about GUI configuration, see Chapter 4, “ESS new deployment instructions,” on page 29.
2. Rerun GUI config (can need to wipe the GUI database clean prior).

   ```
   systemctl start gpfsgui
   ```

   Log in to GUI and reconfigure.

   Note: Ensure that you have the rack elevation of the hardware installed.
3. Exit the container and re ESA config for call home. For more information about configuring call home, see Configuring call home.

The following example lists the steps of cleaning the GUI database:

```
#  cleanup GUI db
#!/bin/bash
echo "Stopping GUI..."
systemctl stop gpfsgui
echo "Cleaning database..."
psql postgreruns postgres -c "drop schema fscc cascade;"
echo "Cleaning CCR files..."
mmccr fdel _gui.settings
mmccr fdel _gui.user.repo
mmccr fdel _gui.keystore_settings
mmccr fdel _gui.policysettings
mmccr fdel _gui.dashboards
mmccr fdel _gui.notification
mmccr fdel gui_jobs
mmccr fdel gui
echo "Cleaning local CCR files..."
rm -f /var/lib/mmfs/gui/*.json*
echo "Cleaning logs..."
rm -rf /var/log/cnlog/mgtsrv/*
echo "Starting GUI..."
systemctl start gpfsgui
echo "Finished"
```

Appendix X. Summary of ESS deployment scenarios

Starting a container

1. Populate /etc/hosts correctly.
2. Fix passwordless between all clusters or ESS nodes (manually).
3. Extract the installation package.
4. Start the container.
5. Issue the following command on all ESS nodes:

```
# essrun -N allESSNodes(management hostnames) config load
```

Ensure that ESS nodes do not have any issues.

Deploying a new ESS node

1. Complete steps in the “Starting a container” on page 145 section.
2. Check whether any ESS nodes has any issues. If any issue is found, fix it.

```
# essrun -N allESSNodes(management hostnames) config check
```

3. Create network bonds.

```
# essrun -N allESSNodes(management hostnames) network --suffix=YourSuffix
```

You can create network bonds manually from any ESS node by using by the **essgennetworks** command. For more information, see the **essgennetworks command**, in the Elastic Storage System: Command Reference.

4. Create a GPFS cluster.

```
# essrun -N allESSIONodes(management hostnames) cluste
```

- Ensure that you use only I/O nodes (ESS 5000, ESS 3000, ESS 3200, or ESS 3500). Do not use EMS or protocol nodes.
- You can use the `--suffix=YourSuffix` option in the command.

5. Check whether any nodes have any issues and fix them.

```
# essrun -N allESSNodes(management hostnames) config check
```

6. Add EMS to the GPFS cluster.

```
# essrun -N oneESSIONodeInCluster(management hostname) cluster --add-ems EMSNode(management
hostname)
```

- Ensure that you use only one node in the cluster for the `-N` option.
- You can use the `--suffix=YourSuffix` option in the command.

7. Create a file system.

```
# essrun -N allESSIONodes(management hostnames) filsystem
```

- Enure that you use only I/O nodes (ESS 5000, ESS 3000, ESS 3200, or ESS 3500).
- You can use the `--suffix=YourSuffix` option in the command.

For more information, see the **essrun command**, in the Elastic Storage System: Command Reference.

Adding a new building block to an existent cluster

1. Complete steps in the “Starting a container” on page 145 section.
   a. Include a new building block in /etc/hosts.
   b. Include the new building block in the **allESSNOdes** commands.
2. Check whether any ESS nodes have any issues. If any issues is found, fix it.

   ```
   # essrun -N allESSNodes(management hostnames) config check
   ```

   Include the new building block or blocks in the **allESSNOdes** commands.
3. Add a building block or blocks to the cluster.

   ```
   # essrun -N oneESSIONodeInCluster(management hostname) cluster --add-nodes
   newESSIONodes(management hostname)
   ```

   - You can use the `--suffix=YourSuffix` option in the command.
   - Use only one type of building block in the `--add-nodes` option. You can use more than two nodes but both must have the same node type.
   - Issue this command, if you are adding different node types.
4. Decide how you want to use the new building block or blocks.
   a. Add a new building block to an existing file system.

      ```
      # essrun -N newESSIONodes vdisk --name someVdiskName(default is essfs)
      ```

      You can use multiple flags in the command. For example, `--bs/--code/--suffix`.

      Ensure that you use the same BlockSize and RaidCode as an existent file system.

      You can use `--extra-vars “--nsd-usage dataOnly --storage-pool system”` or any other `--nsd-usage/--storage-pool` combination.

      Ensure that only one building block is supported to create vdisk.
   b. Execute in `oneESSIONodeInCluster`:

      ```
      mmvdisk filesystem add --fs YourExistentFSName --vdisk NewVdiskCreated
      ```

      Check the output of the previous command.

   Repeat these steps, you want to add more than one building blocks.
5. Create a file system in the new building block or blocks.

   ```
   # essrun -N allNewESSIONodes(management hostnames) filsystem --name someFSName(default is
   essfs)
   ```

   - Ensure that you use only I/O nodes (ESS 5000, ESS 3000, ESS 3200, or ESS 3500).
   - You can use the `--suffix=YourSuffix` option in the command.

   For more information, see the **essrun command**, in the Elastic Storage System: Command Reference.
6. Create recovery groups only in the new building block(s).

   ```
   # essrun -N  allNewESSIONodes(management hostnames) filsystem --rg-only
   ```

   - Ensure that you use only I/O nodes (ESS 5000, ESS 3000, ESS 3200, or ESS 3500).
   - You can use the `--suffix=YourSuffix` option in the command.

   For more information, see the **essrun command**, in the Elastic Storage System: Command Reference.

Updating ESS online

1. Complete steps in the “Starting a container” on page 145 section.
2. Check whether any nodes have any issues.

```
# essrun -N allESSNodes(management hostnames) config check
```

3. Update ESS.

```
# essrun -N allESSIONodes(management hostnames) update
```

Complete steps in the “Starting a container” on page 145 section.

Online update requires Cluster and RecoveryGroups to be created, otherwise use Offline update.

--serial option
The `--serial 1` option is default behavior for online updates. When this option is selected online update is done one by one from one “side” of the building block(s), and then one by one of the other “side” of the building block(s).

--serial n
If more than one building block is useful, it will do n by n from one “side” of the building blocks, and then n by n of the other “side” of the building blocks.

--serial all
If you have more than one building block, it will do ALL nodes from one “side” of the building blocks, and then ALL nodes from the other “side” of the building blocks.

--no-fw-update

- Skips the firmware level update, which updated by using the **mmchfirmware** command.
- OS, Kernel, GPFS, and OFED will be updated.
- Run the **mmchfirmware --type drives/storage-enclosure** command on nodes that require firmware updates.

--no-check
This option will avoid to stop between “sides”. When this option is selected, you are not asked whether you are ready to continue with “side B”, which is the second half of the building block(s).

Usually there is a stop in the middle for safety checks, you can manually run some health checks in the updated nodes.

--ofed-only
This option is not supported when you do online update.

Updating ESS offline

1. Complete steps in the “Starting a container” on page 145 section.
2. Check whether any ESS nodes have any issues.

```
# essrun -N allESSNodes(management hostnames) config check
```

3. Update ESS offline.

```
# essrun -N allESSIONodes(management hostnames) update --offline
```

- Offline update does not require Cluster and RecoveryGroups to be created.
- If GPFS is active on some nodes, shut down those nodes. This might create quorum issues.

--no-fw-update
Skip firmware level update by using the **mmchfirmware** command only if GPFS cluster is created.

OS, Kernel, GPFS, and OFED will be updated.

Run the **mmchfirmware --type drives/storage-enclosure** command on nodes that require firmware updates.

--ofed-only
forces installation or update of OFED.

OS, Kernel, GPFS are not updated.

--serial
This option is ignored in offline update.

- The default behavior for offline update is to update ALL nodes that are defined in the `-N` option.

For more information, see the **essrun command**, in the Elastic Storage System: Command Reference.

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 Documentation, and its related publications, are accessibility-enabled.

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.

The registered trademark Linux is used pursuant to a sublicense from the Linux Foundation, the exclusive licensee of Linus Torvalds, owner of the mark on a worldwide basis.

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

Red Hat and Ansible are trademarks or registered trademarks of Red Hat, Inc. or its subsidiaries in the United States and other countries.

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

Terms and conditions for product documentation

Permissions for the use of these publications are granted subject to the following terms and conditions.

IBM Privacy Policy

At IBM we recognize the importance of protecting your personal information and are committed to processing it responsibly and in compliance with applicable data protection laws in all countries in which IBM operates.

Visit the IBM Privacy Policy for additional information on this topic at https://www.ibm.com/privacy/details/us/en/.

Applicability

These terms and conditions are in addition to any terms of use for the IBM website.

Personal use

You can reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You cannot distribute, display, or make derivative work of these publications, or any portion thereof, without the express consent of IBM.

Commercial use

You can reproduce, distribute, and display these publications solely within your enterprise provided that all proprietary notices are preserved. You cannot make derivative works of these publications, or reproduce, distribute, or display these publications or any portion thereof outside your enterprise, without the express consent of IBM.

Rights

Except as expressly granted in this permission, no other permissions, licenses, or rights are granted, either express or implied, to the Publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions that are granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or as determined by IBM, the above instructions are not being properly followed.

You cannot download, export, or reexport this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.

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)*.

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.

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)*.

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 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.

Index

Product Number: 5765-DME
5765-DAE

SC27-9873-02