IBM Storage Scale

Big Data and Analytics Guide

SC27-9284-14

Note

Before using this information and the product it supports, read the information in "Notices" on page 505.

This edition applies to Version 5 release 1 modification 9 of the following products, and to all subsequent releases and modifications until otherwise indicated in new editions:

- IBM Storage Scale Data Management Edition ordered through Passport Advantage® (product number 5737-F34)
- IBM Storage Scale Data Access Edition ordered through Passport Advantage (product number 5737-I39)
- IBM Storage Scale Erasure Code Edition ordered through Passport Advantage (product number 5737-J34)
- IBM Storage Scale Data Management Edition ordered through AAS (product numbers 5641-DM1, DM3, DM5)
- IBM Storage Scale Data Access Edition ordered through AAS (product numbers 5641-DA1, DA3, DA5)
- IBM Storage Scale Data Management Edition for IBM® ESS (product number 5765-DME)
- IBM Storage Scale Data Access Edition for IBM ESS (product number 5765-DAE)
- IBM Storage Scale Backup ordered through Passport Advantage® (product number 5900-AXJ)
- IBM Storage Scale Backup ordered through AAS (product numbers 5641-BU1, BU3, BU5)
- IBM Storage Scale Backup for IBM® Storage Scale System (product number 5765-BU1)

Significant changes or additions to the text and illustrations are indicated by a vertical line (|) to the left of the change.

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

Contents

Tables

About this information

This edition applies to IBM Storage Scale version 5.1.9 for AIX®, Linux®, and Windows.

IBM Storage Scale is a file management infrastructure, based on IBM General Parallel File System (GPFS) technology, which provides unmatched performance and reliability with scalable access to critical file data.

To find out which version of IBM Storage Scale is running on a particular AIX node, enter:

```
lslpp -l gpfs\*
```

To find out which version of IBM Storage Scale is running on a particular Linux node, enter:

```
rpm -qa | grep gpfs     (for SLES and Red Hat Enterprise Linux)
```

```
dpkg -l | grep gpfs     (for Ubuntu Linux)
```

To find out which version of IBM Storage Scale is running on a particular Windows node, open **Programs and Features** in the control panel. The IBM Storage Scale installed program name includes the version number.

Which IBM Storage Scale information unit provides the information you need?

The IBM Storage Scale library consists of the information units listed in Table 1 on page x.

To use these information units effectively, you must be familiar with IBM Storage Scale and the AIX, Linux, or Windows operating system, or all of them, depending on which operating systems are in use at your installation. Where necessary, these information units provide some background information relating to AIX, Linux, or Windows. However, more commonly they refer to the appropriate operating system documentation.

Note: Throughout this documentation, the term "Linux" refers to all supported distributions of Linux, unless otherwise specified.

Table 1. IBM Storage Scale library information units

Information unit	**Type of information**	**Intended users**
IBM Storage Scale: Concepts, Planning, and Installation Guide	This guide provides the following information: **Product overview** • Overview of IBM Storage Scale • GPFS architecture • Protocols support overview: Integration of protocol access methods with GPFS • Active File Management • AFM-based Asynchronous Disaster Recovery (AFM DR) • Introduction to AFM to cloud object storage • Introduction to system health and troubleshooting • Introduction to performance monitoring • Data protection and disaster recovery in IBM Storage Scale • Introduction to IBM Storage Scale GUI • IBM Storage Scale management API • Introduction to Cloud services • Introduction to file audit logging • Introduction to clustered watch folder • Understanding call home • IBM Storage Scale in an OpenStack cloud deployment • IBM Storage Scale product editions • IBM Storage Scale license designation • Capacity-based licensing • Dynamic pagepool	System administrators, analysts, installers, planners, and programmers of IBM Storage Scale clusters who are very experienced with the operating systems on which each IBM Storage Scale cluster is based

Table 1. IBM Storage Scale library information units (continued)

Information unit	Type of information	Intended users
IBM Storage Scale: Concepts, Planning, and Installation Guide	**Planning** • Planning for GPFS • Planning for protocols • Planning for cloud services • Planning for IBM Storage Scale on Public Clouds • Planning for AFM • Planning for AFM DR • Planning for AFM to cloud object storage • Planning for performance monitoring tool • Planning for UEFI secure boot	
IBM Storage Scale: Concepts, Planning, and Installation Guide	• Firewall recommendations • Considerations for GPFS applications • Security-Enhanced Linux support • Space requirements for call home data upload	

Table 1. IBM Storage Scale library information units (continued)

Information unit	Type of information	Intended users
IBM Storage Scale: Concepts, Planning, and Installation Guide	**Installing** • Steps for establishing and starting your IBM Storage Scale cluster • Installing IBM Storage Scale on Linux nodes and deploying protocols • Installing IBM Storage Scale on public cloud by using cloudkit • Installing IBM Storage Scale on AIX nodes • Installing IBM Storage Scale on Windows nodes • Installing Cloud services on IBM Storage Scale nodes • Installing and configuring IBM Storage Scale management API • Installing GPUDirect Storage for IBM Storage Scale • Installation of Active File Management (AFM) • Installing AFM Disaster Recovery • Installing call home • Installing file audit logging • Installing clustered watch folder • Installing the signed kernel modules for UEFI secure boot • Steps to permanently uninstall IBM Storage Scale **Upgrading** • IBM Storage Scale supported upgrade paths • Online upgrade support for protocols and performance monitoring • Upgrading IBM Storage Scale nodes	System administrators, analysts, installers, planners, and programmers of IBM Storage Scale clusters who are very experienced with the operating systems on which each IBM Storage Scale cluster is based

Table 1. IBM Storage Scale library information units (continued)

Information unit	Type of information	Intended users
IBM Storage Scale: Concepts, Planning, and Installation Guide	• Upgrading IBM Storage Scale non-protocol Linux nodes • Upgrading IBM Storage Scale protocol nodes • Upgrading IBM Storage Scale on cloud • Upgrading GPUDirect Storage • Upgrading AFM and AFM DR • Upgrading object packages • Upgrading SMB packages • Upgrading NFS packages • Upgrading call home • Upgrading the performance monitoring tool • Upgrading signed kernel modules for UEFI secure boot • Manually upgrading pmswift • Manually upgrading the IBM Storage Scale management GUI • Upgrading Cloud services • Upgrading to IBM Cloud Object Storage software level 3.7.2 and above • Upgrade paths and commands for file audit logging and clustered watch folder • Upgrading IBM Storage Scale components with the installation toolkit • Protocol authentication configuration changes during upgrade • Changing the IBM Storage Scale product edition • Completing the upgrade to a new level of IBM Storage Scale • Reverting to the previous level of IBM Storage Scale	System administrators, analysts, installers, planners, and programmers of IBM Storage Scale clusters who are very experienced with the operating systems on which each IBM Storage Scale cluster is based

Table 1. IBM Storage Scale library information units (continued)

Information unit	**Type of information**	**Intended users**
IBM Storage Scale: Concepts, Planning, and Installation Guide	• Coexistence considerations • Compatibility considerations • Considerations for IBM Storage Protect for Space Management • Applying maintenance to your IBM Storage Scale system • Guidance for upgrading the operating system on IBM Storage Scale nodes • Considerations for upgrading from an operating system not supported in IBM Storage Scale 5.1.x.x • Servicing IBM Storage Scale protocol nodes • Offline upgrade with complete cluster shutdown	

Table 1. IBM Storage Scale library information units (continued)

Information unit	Type of information	Intended users
IBM Storage Scale: Administration Guide	This guide provides the following information: **Configuring** • Configuring the GPFS cluster • Configuring GPUDirect Storage for IBM Storage Scale • Configuring the CES and protocol configuration • Configuring and tuning your system for GPFS • Parameters for performance tuning and optimization • Ensuring high availability of the GUI service • Configuring and tuning your system for Cloud services • Configuring IBM Power Systems for IBM Storage Scale • Configuring file audit logging • Configuring clustered watch folder • Configuring the cloudkit • Configuring Active File Management • Configuring AFM-based DR • Configuring AFM to cloud object storage • Tuning for Kernel NFS backend on AFM and AFM DR • Configuring call home • Integrating IBM Storage Scale Cinder driver with Red Hat OpenStack Platform 16.1 • Configuring Multi-Rail over TCP (MROT) • Dynamic pagepool configuration	System administrators or programmers of IBM Storage Scale systems

Table 1. IBM Storage Scale library information units (continued)

Information unit	Type of information	Intended users
IBM Storage Scale: Administration Guide	**Administering** • Performing GPFS administration tasks • Performing parallel copy with mmxcp command • Protecting file data: IBM Storage Scale safeguarded copy • Verifying network operation with the mmnetverify command • Managing file systems • File system format changes between versions of IBM Storage Scale • Managing disks	System administrators or programmers of IBM Storage Scale systems

Table 1. IBM Storage Scale library information units (continued)

Information unit	**Type of information**	**Intended users**
IBM Storage Scale: Administration Guide	• Managing protocol services • Managing protocol user authentication • Managing protocol data exports • Managing object storage • Managing GPFS quotas • Managing GUI users • Managing GPFS access control lists • Native NFS and GPFS • Accessing a remote GPFS file system • Information lifecycle management for IBM Storage Scale • Creating and maintaining snapshots of file systems • Creating and managing file clones • Scale Out Backup and Restore (SOBAR) • Data Mirroring and Replication • Implementing a clustered NFS environment on Linux • Implementing Cluster Export Services • Identity management on Windows / RFC 2307 Attributes • Protocols cluster disaster recovery • File Placement Optimizer • Encryption • Managing certificates to secure communications between GUI web server and web browsers • Securing protocol data • Cloud services: Transparent cloud tiering and Cloud data sharing • Managing file audit logging • RDMA tuning • Configuring Mellanox Memory Translation Table (MTT) for GPFS RDMA VERBS Operation • Administering cloudkit • Administering AFM • Administering AFM DR	System administrators or programmers of IBM Storage Scale systems

Table 1. IBM Storage Scale library information units (continued)

Information unit	Type of information	Intended users
IBM Storage Scale: Administration Guide	• Administering AFM to cloud object storage • Highly available write cache (HAWC) • Local read-only cache • Miscellaneous advanced administration topics • GUI limitations	System administrators or programmers of IBM Storage Scale systems

Table 1. IBM Storage Scale library information units (continued)

Information unit	**Type of information**	**Intended users**
IBM Storage Scale: Problem Determination Guide	This guide provides the following information: **Monitoring** • Monitoring system health by using IBM Storage Scale GUI • Monitoring system health by using the mmhealth command • Dynamic pagepool monitoring • Performance monitoring • Monitoring GPUDirect storage • Monitoring events through callbacks • Monitoring capacity through GUI • Monitoring AFM and AFM DR • Monitoring AFM to cloud object storage • GPFS SNMP support • Monitoring the IBM Storage Scale system by using call home • Monitoring remote cluster through GUI • Monitoring file audit logging • Monitoring clustered watch folder • Monitoring local read-only cache **Troubleshooting** • Best practices for troubleshooting • Understanding the system limitations • Collecting details of the issues • Managing deadlocks • Installation and configuration issues • Upgrade issues • CCR issues • Network issues • File system issues • Disk issues • GPUDirect Storage troubleshooting • Security issues • Protocol issues • Disaster recovery issues • Performance issues	System administrators of GPFS systems who are experienced with the subsystems used to manage disks and who are familiar with the concepts presented in the *IBM Storage Scale: Concepts, Planning, and Installation Guide*

Table 1. IBM Storage Scale library information units (continued)

Information unit	**Type of information**	**Intended users**
IBM Storage Scale: Problem Determination Guide	• GUI and monitoring issues • AFM issues • AFM DR issues • AFM to cloud object storage issues • Transparent cloud tiering issues • File audit logging issues • Cloudkit issues • Troubleshooting mmwatch • Maintenance procedures • Recovery procedures • Support for troubleshooting • References	

Table 1. IBM Storage Scale library information units (continued)		
Information unit	**Type of information**	**Intended users**
IBM Storage Scale: Command and Programming Reference Guide	This guide provides the following information: **Command reference** • cloudkit command • gpfs.snap command • mmaddcallback command • mmadddisk command • mmaddnode command • mmadquery command • mmafmconfig command • mmafmcosaccess command • mmafmcosconfig command • mmafmcosctl command • mmafmcoskeys command • mmafmctl command • mmafmlocal command • mmapplypolicy command • mmaudit command • mmauth command • mmbackup command • mmbackupconfig command • mmbuildgpl command • mmcachectl command • mmcallhome command • mmces command • mmchattr command • mmchcluster command • mmchconfig command • mmchdisk command • mmcheckquota command • mmchfileset command • mmchfs command • mmchlicense command • mmchmgr command • mmchnode command • mmchnodeclass command • mmchnsd command • mmchpolicy command • mmchpool command • mmchqos command • mmclidecode command	• System administrators of IBM Storage Scale systems • Application programmers who are experienced with IBM Storage Scale systems and familiar with the terminology and concepts in the XDSM standard

Table 1. IBM Storage Scale library information units (continued)		
Information unit	**Type of information**	**Intended users**
IBM Storage Scale: Command and Programming Reference Guide	• mmclone command • mmcloudgateway command • mmcrcluster command • mmcrfileset command • mmcrfs command • mmcrnodeclass command • mmcrnsd command • mmcrsnapshot command • mmdefedquota command • mmdefquotaoff command • mmdefquotaon command • mmdefragfs command • mmdelacl command • mmdelcallback command • mmdeldisk command • mmdelfileset command • mmdelfs command • mmdelnode command • mmdelnodeclass command • mmdelnsd command • mmdelsnapshot command • mmdf command • mmdiag command • mmdsh command • mmeditacl command • mmedquota command • mmexportfs command • mmfsck command • mmfsckx command • mmfsctl command • mmgetacl command • mmgetstate command • mmhadoopctl command • mmhdfs command • mmhealth command • mmimgbackup command • mmimgrestore command • mmimportfs command • mmkeyserv command	• System administrators of IBM Storage Scale systems • Application programmers who are experienced with IBM Storage Scale systems and familiar with the terminology and concepts in the XDSM standard

Table 1. IBM Storage Scale library information units (continued)

Information unit	Type of information	Intended users
IBM Storage Scale: Command and Programming Reference Guide	• mmlinkfileset command • mmlsattr command • mmlscallback command • mmlscluster command • mmlsconfig command • mmlsdisk command • mmlsfileset command • mmlsfs command • mmlslicense command • mmlsmgr command • mmlsmount command • mmlsnodeclass command • mmlsnsd command • mmlspolicy command • mmlspool command • mmlsqos command • mmlsquota command • mmlssnapshot command • mmmigratefs command • mmmount command • mmnetverify command • mmnfs command • mmnsddiscover command • mmobj command • mmperfmon command • mmpmon command • mmprotocoltrace command • mmpsnap command • mmputacl command • mmqos command • mmquotaoff command • mmquotaon command • mmreclaimspace command • mmremotecluster command • mmremotefs command • mmrepquota command • mmrestoreconfig command • mmrestorefs command • mmrestrictedctl command • mmrestripefile command	• System administrators of IBM Storage Scale systems • Application programmers who are experienced with IBM Storage Scale systems and familiar with the terminology and concepts in the XDSM standard

Table 1. IBM Storage Scale library information units (continued)

Information unit	Type of information	Intended users
IBM Storage Scale: Command and Programming Reference Guide	• mmrestripefs command • mmrpldisk command • mmsdrrestore command • mmsetquota command • mmshutdown command • mmsmb command • mmsnapdir command • mmstartup command • mmstartpolicy command • mmtracectl command • mmumount command • mmunlinkfileset command • mmuserauth command • mmwatch command • mmwinservctl command • mmxcp command • spectrumscale command **Programming reference** • IBM Storage Scale Data Management API for GPFS information • GPFS programming interfaces • GPFS user exits • IBM Storage Scale management API endpoints • Considerations for GPFS applications	• System administrators of IBM Storage Scale systems • Application programmers who are experienced with IBM Storage Scale systems and familiar with the terminology and concepts in the XDSM standard

Table 1. IBM Storage Scale library information units (continued)

Information unit	**Type of information**	**Intended users**
IBM Storage Scale: Big Data and Analytics Guide	This guide provides the following information: Summary of changes Big data and analytics support Hadoop Scale Storage Architecture • Elastic Storage Server • Erasure Code Edition • Share Storage (SAN-based storage) • File Placement Optimizer (FPO) • Deployment model • Additional supported storage features IBM Spectrum® Scale support for Hadoop • HDFS transparency overview • Supported IBM Storage Scale storage modes • Hadoop cluster planning • CES HDFS • Non-CES HDFS • Security • Advanced features • Hadoop distribution support • Limitations and differences from native HDFS • Problem determination IBM Storage Scale Hadoop performance tuning guide • Overview • Performance overview • Hadoop Performance Planning over IBM Storage Scale • Performance guide	• System administrators of IBM Storage Scale systems • Application programmers who are experienced with IBM Storage Scale systems and familiar with the terminology and concepts in the XDSM standard

Table 1. IBM Storage Scale library information units (continued)

Information unit	Type of information	Intended users
IBM Storage Scale: Big Data and Analytics Guide	Cloudera Data Platform (CDP) Private Cloud Base • Overview • Planning • Installing • Configuring • Administering • Monitoring • Upgrading • Limitations • Problem determination	• System administrators of IBM Storage Scale systems • Application programmers who are experienced with IBM Storage Scale systems and familiar with the terminology and concepts in the XDSM standard
IBM Storage Scale: Big Data and Analytics Guide	Cloudera HDP 3.X • Planning • Installation • Upgrading and uninstallation • Configuration • Administration • Limitations • Problem determination Open Source Apache Hadoop • Open Source Apache Hadoop without CES HDFS • Open Source Apache Hadoop with CES HDFS	• System administrators of IBM Storage Scale systems • Application programmers who are experienced with IBM Storage Scale systems and familiar with the terminology and concepts in the XDSM standard

Table 1. IBM Storage Scale library information units (continued)

Information unit	Type of information	Intended users
IBM Storage Scale Erasure Code Edition Guide	IBM Storage Scale Erasure Code Edition • Summary of changes • Introduction to IBM Storage Scale Erasure Code Edition • Planning for IBM Storage Scale Erasure Code Edition • Installing IBM Storage Scale Erasure Code Edition • Uninstalling IBM Storage Scale Erasure Code Edition • Creating an IBM Storage Scale Erasure Code Edition storage environment • Using IBM Storage Scale Erasure Code Edition for data mirroring and replication • Deploying IBM Storage Scale Erasure Code Edition on VMware infrastructure • Upgrading IBM Storage Scale Erasure Code Edition • Incorporating IBM Storage Scale Erasure Code Edition in an Elastic Storage Server (ESS) cluster • Incorporating IBM Elastic Storage Server (ESS) building block in an IBM Storage Scale Erasure Code Edition cluster • Administering IBM Storage Scale Erasure Code Edition • Troubleshooting • IBM Storage Scale RAID Administration	• System administrators of IBM Storage Scale systems • Application programmers who are experienced with IBM Storage Scale systems and familiar with the terminology and concepts in the XDSM standard

Table 1. IBM Storage Scale library information units (continued)

Information unit	Type of information	Intended users
IBM Storage Scale Container Native Storage Access	This guide provides the following information: • Overview • Planning • Installation prerequisites • Installing the IBM Storage Scale container native operator and cluster • Upgrading • Configuring IBM Storage Scale Container Storage Interface (CSI) driver • Using IBM Storage Scale GUI • Maintenance of a deployed cluster • Cleaning up the container native cluster • Monitoring • Troubleshooting • References	• System administrators of IBM Storage Scale systems • Application programmers who are experienced with IBM Storage Scale systems and familiar with the terminology and concepts in the XDSM standard
IBM Storage Scale Data Access Service	This guide provides the following information: • Overview • Architecture • Security • Planning • Installing and configuring • Upgrading • Administering • Monitoring • Collecting data for support • Troubleshooting • The **mmdas** command • REST APIs	• System administrators of IBM Storage Scale systems • Application programmers who are experienced with IBM Storage Scale systems and familiar with the terminology and concepts in the XDSM standard

Table 1. IBM Storage Scale library information units (continued)

Information unit	Type of information	Intended users
IBM Storage Scale Container Storage Interface Driver Guide	This guide provides the following information: • Summary of changes • Introduction • Planning • Installation • Upgrading • Configurations • Using IBM Storage Scale Container Storage Interface Driver • Managing IBM Storage Scale when used with IBM Storage Scale Container Storage Interface driver • Cleanup • Limitations • Troubleshooting	• System administrators of IBM Storage Scale systems • Application programmers who are experienced with IBM Storage Scale systems and familiar with the terminology and concepts in the XDSM standard

Prerequisite and related information

For updates to this information, see IBM Storage Scale in IBM Documentation.

For the latest support information, see the IBM Storage Scale FAQ in IBM Documentation.

Conventions used in this information

Table 2 on page xxix describes the typographic conventions used in this information. UNIX file name conventions are used throughout this information.

Note: Users of IBM Storage Scale for Windows must be aware that on Windows, UNIX-style file names need to be converted appropriately. For example, the GPFS cluster configuration data is stored in the `/var/mmfs/gen/mmsdrfs` file. On Windows, the UNIX namespace starts under the `%SystemDrive%\cygwin64` directory, so the GPFS cluster configuration data is stored in the `C:\cygwin64\var\mmfs\gen\mmsdrfs` file.

Table 2. Conventions

Convention	Usage
bold	Bold words or characters represent system elements that you must use literally, such as commands, flags, values, and selected menu options. Depending on the context, **bold** typeface sometimes represents path names, directories, or file names.
bold underlined	bold underlined keywords are defaults. These take effect if you do not specify a different keyword.

Table 2. Conventions (continued)

Convention	Usage
`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.

Note: CLI options that accept a list of option values delimit with a comma and no space between values. As an example, to display the state on three nodes use `mmgetstate -N` *NodeA*,*NodeB*,*NodeC*. Exceptions to this syntax are listed specifically within the command.

How to send your comments

Your feedback is important in helping us to produce accurate, high-quality information. If you have any comments about this information or any other IBM Storage Scale documentation, send your comments to the following e-mail address:

`mhvrcfs@us.ibm.com`

Include the publication title and order number, and, if applicable, the specific location of the information about which you have comments (for example, a page number or a table number).

To contact the IBM Storage Scale development organization, send your comments to the following e-mail address:

`scale@us.ibm.com`

Summary of changes

This topic summarizes changes to IBM Storage Scale Big Data and Analytics (BDA) support section.

For information about IBM Storage Scale changes, see the IBM Storage Scale Summary of changes.

For information about BDA feature support, see the *List of stabilized, deprecated, and discontinued features* section under the Summary of changes.

For information about the resolved IBM Storage Scale APARs, see IBM Storage Scale APARs Resolved.

For information about supported HDFS Transparency versions with IBM Storage Scale, see "HDFS Transparency support matrix" on page 27.

For information about supported Cloudera Data Platform (CDP) versions with IBM Storage Scale, see "Support Matrix" on page 294.

Summary of changes as updated, February 2024

Changes in IBM Storage Scale 5.1.9-2

- Includes HDFS Transparency 3.1.1-17 and HDFS Transparency 3.2.2-7.

Changes in HDFS Transparency 3.1.1-17 in IBM Storage Scale 5.1.9-2

- Updated several JavaScript files related to the NameNode and DataNode GUI.
- Fixed multiple issues that occurred while stopping or starting HDFS Transparency roles when the IBM Storage Scale file system was respectively unmounted or remounted.

Note: HDFS Transparency 3.2.2-7 supports an upgrade only from HDFS Transparency 3.2.2-5.

Summary of changes as updated, December 2023

Changes in IBM Storage Scale 5.1.9-1

- Includes HDFS Transparency 3.1.1-16 and HDFS Transparency 3.2.2-7.

Changes in HDFS Transparency 3.1.1-16 in IBM Storage Scale 5.1.9-1

- Fixed an issue where the reinstallation of the same HDFS Transparency `rpm` version failed and could not be recovered.
- Included `runLog4jV1Patcher.sh` in `/usr/lpp/mmfs/hadoop/scripts/` to patch a user provided `log4j` JAR.

Changes in HDFS Transparency 3.2.2-7 in IBM Storage Scale 5.1.9-1

- Fixed an issue where the reinstallation of the same HDFS Transparency `rpm` version failed and could not be recovered.
- Included `runLog4jV1Patcher.sh` in `/usr/lpp/mmfs/hadoop/scripts/` to patch a `log4j` JAR provided by a user.
- Fixed an issue where too many lookups and log entries for missing UID and GID would impact the HDFS Transparency performance.
- Improved **`hdfs dfs ls`** to use IBM Storage Scale **`ls`** as input, instead of caching the metadata and synchronizing with IBM Storage Scale regularly.

Changes in the documentation

- Restructured the "IBM Storage Scale support for Hadoop" chapter.
- Moved "Hadoop IBM Storage Scale Architecture" to "IBM Storage Scale support for Hadoop" > "Overview".

Note: HDFS Transparency 3.2.2-7 supports an upgrade only from HDFS Transparency 3.2.2-5.

Summary of changes as updated, November 2023

Changes in Cloudera Data Platform Private Cloud Base

- From IBM Storage Scale 5.1.8.0, CDP Private Cloud Base 7.1.9-CHF1 is certified with IBM Storage Scale on x86 and Power LE. For more information, see "Support Matrix" on page 294.

Changes in IBM Storage Scale 5.1.9-0

- Includes HDFS Transparency 3.1.1-15 and HDFS Transparency 3.2.2-6.

Changes in HDFS Transparency 3.1.1-15 in IBM Storage Scale 5.1.9-0

- Added the **`mmhdfs config dump`** subcommand. For more information, see **`mmhdfs command`**.
- Increased the performance for recursive deletions of snapshot-enabled directories by avoiding the **`mmlssnapshot`** dependency.
- Improved internal data structures to avoid directory lock contentions.
- Fixed an issue where the log includes many messages like `aclutil.cc get_file failed [No such file or directory]`.
- Fixed an issue where the **`getContentSummary`** returns inconsistent results if multiple files in the same directory are removed at the same time.
- Added buffered logging and log filtering, which increases HDFS Transparency I/O throughput. For more information, see "Buffered logging and filtering" on page 268.
- Changed the installation process to use self-provided JAR files. For more information, see "Installation prerequisites" on page 30.

Changes in HDFS Transparency 3.2.2-6 in IBM Storage Scale 5.1.9-0

- Rebranded scripts in HDFS Transparency 3.2.2.-6 from "IBM Spectrum Scale" to "IBM Storage Scale".
- Added the **`mmhdfs config dump`** subcommand. For more information, see **`mmhdfs command`**
- Increased the performance for recursive deletions of snapshot-enabled directories by avoiding the **`mmlssnapshot`** dependency.
- Improved internal data structures to avoid directory lock contentions.
- Fixed an issue where the log includes many messages like `aclutil.cc get_file failed [No such file or directory]`.
- Fixed an issue where the **`getContentSummary`** returns inconsistent results if multiple files in the same directory are removed at the same time.
- Added buffered logging and log filtering, which increases HDFS Transparency I/O throughput. For more information, see "Buffered logging and filtering" on page 268.
- Changed the installation process to use self-provided JAR files. For more information, see "Installation prerequisites" on page 30.

Note: In upcoming IBM Storage Scale versions, HDFS Transparency 3.2.2-x will be replaced by an HDFS Transparency 3.3.5-x version based on the correspondent Apache Hadoop 3.3.5 version. The Apache Hadoop version used as basis for the HDFS Transparency version will be supported by Apache Bigtop.

Summary of changes as updated, July 2023

Changes in IBM Storage Scale 5.1.8-1

- Includes HDFS Transparency 3.1.1-14, HDFS Transparency 3.2.2-5, and HDFS Transparency 3.3.0-2.

Changes in HDFS Transparency 3.1.1-14 in IBM Storage Scale 5.1.8-1

- Rebranding of documentation and scripts in HDFS Transparency 3.1.1-14

- Adding argument "validity" to `gpfs_tls_configuration.py` to define the time for which TLS certifications are valid.
- Previous default of 90 days TLS certification validity was changed to 1826 days if no "validity" argument is passed to `gpfs_tls_configuration.py`.
- Improved error handling for malconfigurations in `gpfs_tls_configuration.py`.
- Improvement of documentation around TLS certification setup. See "TLS" on page 129

Summary of changes as updated, April 2023

Changes in IBM Storage Scale 5.1.7-1

- Includes HDFS Transparency 3.1.1-13, HDFS Transparency 3.2.2-5, and HDFS Transparency 3.3.0-2.

Changes in HDFS Transparency 3.1.1-13 in IBM Storage Scale 5.1.7-1

- Fixed an issue where appending to an existing file in an encryption zone failed (APAR IJ45843).
- Improved parallel data access by reducing the locking scope on directory-level to avoid parent directory locking.
- Fixed an issue where the **rm** and **du** commands would fail with `NoSuchFileException`.
- Reduced exception to warning when a file lease cannot be found while creating a file, in order to prevent application-side failures.
- Improved overall performance by changing the update process for NameNode metadata and reducing the syncChildren calls.
- Fixed an issue where the NameNode crashes by failing to finalize the shared edit log on NameNode failover.
- Improved the listing performance by changing the way stat is called and avoiding stat oscillation behavior.
- Changed the RSA Key strength in the TLS enablement script from *1024* to *2048*.
- Added an **AccessControlException** in the **put** command if used for deleted users.
- Fixed an issue where the TLS script fails with **enable-tls** option if the **dfs.namenode.http-address** parameter is missing in the configuration.
- Realigned the usage text of **hdfs getconf**.
- Fixed an issue where the NameNode will not start because of a missing dependent jar. For resolution in the affected HDFS Transparency versions 3.1.1-11, 3.1.1-12, 3.2.2-2 and 3.2.2-3, see NameNode fails to start in HDFS Transparency 3.1.1-11, 3.1.1-12, 3.2.2-2 or 3.2.2-3.

Changes in HDFS Transparency 3.2.2.-5 in IBM Storage Scale 5.1.7-1

- Fixed an issue where parallel move or rename and listing operations on the same directory can lead to a deadlock situation.

Summary of changes as updated, March 2023

Changes in IBM Storage Scale 5.1.7-0

- Includes HDFS Transparency 3.1.1-12, HDFS Transparency 3.2.2-4, and HDFS Transparency 3.3.0-2.

Changes in HDFS Transparency 3.2.2-4 in IBM Storage Scale 5.1.7-0

- Fixed an issue where `gpfs_kerberos_configuration.py` fails to run.
- Improved parallel data access by reducing the locking scope on directory level to avoid parent directory locking.
- Fixed an issue where the **rm** and **du** commands fail with `NoSuchFileException`.
- Reduced exception to warning when a file lease cannot be found while creating a file, in order to prevent application side failures.

- Improved overall performance by changing the update process for NameNode metadata and reducing syncChildren calls.
- Fixed an issue where the NameNode crashes by failing to finalize the shared edit log on NameNode failover.
- Improved the listing performance by changing the way stat is called and avoiding stat oscillation behavior.
- Fixed an issue where the TLS script fails with `enable-tls` option if the **`dfs.namenode.http-address`** parameter is missing in the configuration.
- Changed TLS encryption value to *2048*.
- Fixed an issue where the NameNode will not start because of a missing dependent jar. For resolution in the affected HDFS Transparency versions 3.1.1-11, 3.1.1-12, 3.2.2-2 and 3.2.2-3, see NameNode fails to start in HDFS Transparency 3.1.1-11, 3.1.1-12, 3.2.2-2 or 3.2.2-3.

Summary of changes as updated, January 2023

Changes in IBM Storage Scale 5.1.6-1

- Includes HDFS Transparency 3.1.1-12, HDFS Transparency 3.2.2-3 and HDFS Transparency 3.3.0-2.

Changes in IBM Storage Scale 5.1.2-9

- Includes HDFS Transparency 3.1.1-12 and HDFS Transparency 3.3.0-2.

Changes in HDFS Transparency 3.1.1-12 in IBM Storage Scale 5.1.2-9 and IBM Storage Scale 5.1.6-1

- Added security fix for CVE-2022-25168.

Summary of changes as updated, December 2022

Changes in IBM Storage Scale 5.1.6-0

- Includes HDFS Transparency 3.1.1-11, HDFS Transparency 3.2.2-3, and HDFS Transparency 3.3.0-2.

Changes in HDFS Transparency 3.1.1-11 in IBM Storage Scale 5.1.6-0

- Fixed the issue where a ticket expiration in an AD Kerberos environment can lead to two active NameNodes.
- Included fine-grained read/write locking of file lease manager to improve the performance.
- Fixed the issue where **`mmhdfs config import`** ignored `ranger-hdfs-policymgr-ssl.xml`.
- Added general security fixes.

Changes in HDFS Transparency 3.2.2-3 in IBM Storage Scale 5.1.6-0

- Added general security fixes.
- Added a fix for the scripts in `/usr/lpp/mmfs/hadoop/scripts/` to run with Python 3.8.

Changes in IBM Storage Scale Big Data Analytics Integration Toolkit for HDFS Transparency (Toolkit for HDFS) 1.0.8-0 in IBM Storage Scale 5.1.6-0

- Fixed an issue that lets the IBM Storage Scale Install Toolkit fail if the file system configured for HDFS Transparency includes an underscore.

Changes in Cloudera Data Platform Private Cloud Base

- From IBM Storage Scale 5.1.4.0, CDP Private Cloud Base 7.1.8 is certified with IBM Storage Scale on Power. For more information, see “Support Matrix” on page 294.

  For Hue to work properly, Cloudera Manager 7.7.1+ requires Python version to be at v3.8 on the Hue nodes.

Summary of changes as updated, October 2022

Changes in IBM Storage Scale 5.1.5-1

- Includes HDFS Transparency 3.1.1-10, HDFS Transparency 3.2.2-2 and HDFS Transparency 3.3.0-2.

Changes in HDFS Transparency 3.2.2-2 in IBM Storage Scale 5.1.5-1

- Fixed the issue where a ticket expiration in an AD Kerberos environment can lead to two active NameNodes.
- Included fine-grained read/write locking of file lease manager to improve the performance.
- Added general security fixes.
- Added security fix for CVE-2022-25168.

Changes in the documentation

- Added "Remote mount at fileset level" on page 192.
- Added HDFS Transparency to IBM Storage Scale support matrix on "HDFS Transparency support matrix" on page 27.

Summary of changes as updated, September 2022

Changes in IBM Storage Scale 5.1.5

- Includes HDFS Transparency 3.1.1-10, HDFS Transparency 3.2.2-1 and HDFS Transparency 3.3.0-2.

Changes in Cloudera Data Platform Private Cloud Base

- From IBM Storage Scale 5.1.4.0, CDP Private Cloud Base 7.1.8 is certified with IBM Storage Scale on x86. For more information, see "Support Matrix" on page 294.

Summary of changes as updated, August 2022

Changes in IBM Storage Scale 5.1.2.6

- Includes HDFS Transparency 3.1.1-10 and HDFS Transparency 3.3.0-2.

  Note: IBM Storage Scale 5.1.3.0, IBM Storage Scale 5.1.3.1 and IBM Storage Scale 5.1.4.0 include earlier versions of HDFS Transparency and an upgrade must be considered to IBM Storage Scale 5.1.4.1 or later.

  Added support for Red Hat IPA Kerberos for HDFS Transparency.

Summary of changes as updated, July 2022

Changes in HDFS Transparency 3.1.1-10 in IBM Storage Scale 5.1.4.1

- Fixed the issue where a fast repetitive usage of **`mmces service stop hdfs`** and **`mmces service start hdfs`** can lead to two standby NameNodes.
- Added security fix for CVE-2022-23305, CVE-2022-23307, CVE-2022-23302 and CVE-2020-9488.

Changes in HDFS Transparency 3.3.0-2 in IBM Storage Scale 5.1.4.1

- Added security fix for CVE-2022-23305, CVE-2022-23307, CVE-2022-23302, CVE-2020-9488.

Summary of changes as updated, June 2022

Changes in HDFS Transparency 3.2.2-1 in IBM Storage Scale 5.1.4.0

- Supports CES HDFS Transparency 3.2.2-1 for Open Source Apache Hadoop 3.2.2 distribution on RH 7.9 on x86_64.

Changes in HDFS Transparency 3.1.1-9 in IBM Storage Scale 5.1.4.0

- Optimized the internal metadata data structures for the NameNode for improved memory efficiency. For more information, see "Recommended hardware resource configuration" on page 16.
- Fixed the parsing problem of `hadoop-env.sh` that used to skip the last line and therefore might miss configuration key-value pairs on the last line of the file.

Summary of changes as updated, May 2022

Changes in HDFS Transparency 3.2.2-0 in IBM Storage Scale 5.1.3.2

- IBM Storage Scale 5.1.3 PTF2 is a technology preview version specifically for Hadoop users who want to try out HDFS Transparency 3.2.2 for Open-source Apache Hadoop 3.2.2 during a limited download period in the Fix Central. This technology preview is only available for Data Management Edition on RHEL 7.9 on x86_64 with a limited-time period for nonproduction usage. IBM Storage Scale 5.1.3 PTF2 contains the additional HDFS Transparency 3.2.2 with the IBM Storage Scale 5.1.3 PTF1 content. Therefore, this technology preview cannot be installed if IBM Storage Scale 5.1.3 PTF1 is already installed.

Summary of changes as updated, April 2022

Changes in Cloudera Data Platform Private Cloud Base

- CDP Private Cloud Base 7.1.7 SP1 is certified with IBM Storage Scale starting from version 5.1.2.2. For more information, see "Support Matrix" on page 294.

Summary of changes as updated, March 2022

Changes in IBM Storage Scale Big Data Analytics Integration Toolkit for HDFS Transparency (Toolkit for HDFS) 1.0.5.0 in IBM Storage Scale 5.1.3

- Supports the parallel offline upgrade.

  The parallel offline upgrade support will change the current offline upgrade process from sequential to parallel. This will significantly reduce the upgrade time in the offline mode.

Changes in IBM Storage Scale file system core configuration in IBM Storage Scale 5.1.3

- For updates to the **`tscCmdAllowRemoteConnections`** parameter, see the *File system core improvements* section under the IBM Storage Scale Summary of changes documentation.

Summary of changes as updated, January 2022

Changes in HDFS Transparency 3.1.1-8 in IBM Storage Scale 5.0.5.12

Changes in HDFS Transparency 3.1.1-8 and 3.3.0-1 in IBM Storage Scale 5.1.2.2

- Added security fix for CVE-2021-4104 and CVE-2019-17571.

Changes in HDFS Transparency 3.1.0-10 in IBM Fix Central

- Added security fix for CVE-2021-4104 and CVE-2019-17571.
- Fixed the timing rename failures.

  Note that HDFS Transparency 3.1.0-10 is the last PTF in the 3.1.0.x stream.

For more information, see IBM Security Bulletin.

Summary of changes as updated, December 2021

Changes in HDFS Transparency 3.1.0-9

- Optimized the handling of the metadata for NameNode for improved memory efficiency.

To ensure that the data on IBM Storage Scale that is to be processed with HDFS Transparency is up to date, the IBM Storage Scale mount option **mtime** *-E: YES* (default value) must be set to always return the accurate file modification times.

- Optimized parallelism for DataNode request processing for the performance improvement. This includes the ports of HDFS-15150 and HDFS-15160 that introduces three DataNode configuration parameters. For more information, see "Configuration options for HDFS Transparency" on page 242.
- The IBM Storage Scale file system is now explicitly checked in mount and unmount callbacks during HDFS Transparency startup and shutdown. Unrelated IBM Storage Scale file systems no longer affect HDFS Transparency. This means that HDFS Transparency will start only if the relevant mount point is properly mounted and will stop if the relevant mount point is unmounted based on the HDFS Transparency status checking in the IBM Storage Scale event callback process.
- Fixed intermittent issues in date and size output when listing files.

Summary of changes as updated, November 2021

Changes in HDFS Transparency 3.1.1-7 in IBM Storage Scale 5.1.2.1

- Support added for Java 11.

Summary of changes as updated, October 2021

Changes in Mpack version 2.7.0.10

- The IBM Storage Scale service can now be deployed or upgraded in a single or multiple HDFS namespace configuration. This includes adding DataNode using Ambari in multiple HDFS namespaces.
- Decommissioning DataNodes using the Ambari HDFS service is now supported.
- Fixed NamenodeHAState init arguments after 1 retry failure during HDP upgrading with Ambari 2.7.5.17-6 and Mpack 2.7.0.9 at the HDFS service upgrade step.
- The IBM Storage Scale service can now be deployed in Ambari in remote cluster mount configuration for non-root Ambari and IBM Storage Scale environment.
- The MoveNameNodeTransparency.py script now supports moving the HDFS Transparency NameNode when Kerberos is enabled.

Changes in Cloudera Data Platform Private Cloud Base

- CDP Private Cloud Base 7.1.7 is certified with IBM Storage Scale from version 5.1.1.2 on Power LE platform. For more information, see "Support Matrix" on page 294.

Changes in IBM Storage Scale Big Data Analytics Integration Toolkit for HDFS Transparency (Toolkit for HDFS) 1.0.4-0 in IBM Storage Scale 5.1.2

- The **cleanup -n** option of installation toolkit will now clear only the configuration of a single HDFS cluster instead of clearing the configurations of all the HDFS clusters in a multi-HDFS cluster environment from the toolkit's metadata.

Changes in HDFS Transparency 3.1.1-6

- Optimized the handling of the metadata for the NameNode performance improvement.
- Optimized parallelism for DataNode request processing for performance improvement. This includes ports of HDFS-15150 and HDFS-15160 that introduces three DataNode configuration parameters. For more information, see "Configuration options for HDFS Transparency" on page 242.
- Fixed getListing RPC to handle the remaining files correctly when the block locations are requested that would cause higher-level services to get an incomplete directory listing.
- Support for decommissioned DataNodes is enabled. For more information, see "Decommissioning DataNodes" on page 77.
- Fixed metadata handling when a listing would not show the correct creation time.

Documentation update

- Added configuration parameters for `gpfs-site.xml` that describes the specific IBM Storage Scale parameters. For more information, see "Configuration parameters for gpfs-site.xml" on page 244.
- Moved the *Configuration options for HDFS Transparency* information to "Configuration parameters" on page 242.

Summary of changes as updated, August 2021

Changes in Cloudera Data Platform Private Cloud Base

- CDP Private Cloud Base 7.1.7 is certified with IBM Storage Scale 5.1.1.2 on x86_64 platform.
- CDP 7.1.7 supports the upgrade path from CDP 7.1.6 with CSD 1.1.0-0 on IBM Storage Scale 5.1.1.1 to CDP 7.1.7 with CSD 1.2.0-0 on IBM Storage Scale 5.1.1.2. For more information, see "Upgrading CDP" on page 341.

Summary of changes as updated, July 2021

Changes in HDFS Transparency 3.3.0-0 in IBM Storage Scale 5.1.1.2

- Supports CES HDFS Transparency 3.3 for Open Source Apache Hadoop 3.3 distribution on RH 7.9 on x86_64.

Changes in IBM Storage Scale Big Data Analytics Integration Toolkit for HDFS Transparency (Toolkit for HDFS) 1.0.3-2 in IBM Storage Scale 5.1.1.2

- Supports new installation of CES HDFS Transparency 3.3 through the IBM Storage Scale installation toolkit on RH 7.9 on x864_64 when the environment variable **`SCALE_HDFS_TRANSPARENCY_VERSION_33_ENABLE`**=*True* is exported. For more information, see "Steps for install toolkit" on page 32.

Changes in HDFS Transparency 3.1.0-8

- Optimized the handling of the metadata for the NameNode performance improvement.
- Fixed getListing RPC to handle the remaining files correctly when block locations are requested that would cause higher-level services to get an incomplete directory listing.
- Backported the fix for a race condition that caused parsing error of **`java.io.BufferedInputStream`** in `org.apache.hadoop.conf.Configuration` class (HADOOP-15331).
- Fixed the handling of the file listing so that the `java.nio.file.NoSuchFileException` warning messages do not occur.
- Fixed the handling of `getBlockLocation` RPC on the files that do not exist. This prevented the YARN ResourceManager to start after configuring node labels directory.
- Support for decommissioned DataNodes is enabled. For more information, see "Decommissioning DataNodes" on page 77.
- General security fixes and CVE-2020-9492 in IBM Support.

Changes in Cloudera HDP

- The **`--sync-hdp`** option used for upgrading HDP is now deprecated.

Summary of changes as updated, June 2021

Changes in Cloudera Data Platform Private Cloud Base

- CDP Private Cloud Base 7.1.6 is now certified on ppc64le.

Changes in HDFS Transparency 3.1.1-5 in IBM Storage Scale 5.1.1.1

- Fixed the handling of the file listing. Therefore, the `java.nio.file.NoSuchFileException` warning messages will no longer occur.
- Fixed the handling of getBlockLocation RPC on files that do not exist. This prevented the YARN ResourceManager to start after configuring the node labels directory.

- From HDFS Transparency 3.1.1-5, the `gpfs_tls_configuration.py` script automates the configuration of Transport Layer Security (TLS) on the CES HDFS Transparency cluster.

Changes in IBM Storage Scale Big Data Analytics Integration Toolkit for HDFS Transparency (Toolkit for HDFS) 1.0.3.1 in IBM Storage Scale 5.1.1.1

- From Toolkit version 1.0.3.1, creating multiple CES HDFS clusters using the IBM Storage Scale installation toolkit during the same deployment run is supported.

Summary of changes as updated, May 2021

Changes in Cloudera Data Platform Private Cloud Base

- From CDP Private Cloud Base 7.1.6, Impala is certified on IBM Storage Scale 5.1.1 on x86_64.

Summary of changes as updated, April 2021

Changes in Cloudera Data Platform Private Cloud Base

- CDP Private Cloud Base 7.1.6 is certified with IBM Storage Scale 5.1.1.0. This CDP Private Cloud Base version supports Transport Layer Security (TLS) and HDFS encryption.

Changes in HDFS Transparency 3.1.1-4

- Fixed the **`mmhdfs`** command to recognize short hostname configuration for NameNodes and Data Nodes. Therefore, `The node is not a namenode or datanode` error message will no longer occur.
- The IBM Storage Scale file systems are now explicitly checked in mount and unmount callbacks during HDFS Transparency startup and shutdown process. Unrelated IBM Storage Scale file systems no longer affect HDFS Transparency. This means that HDFS Transparency will start only if the relevant mount point is properly mounted and will stop if the relevant mount point is unmounted based on the HDFS Transparency status checking in the IBM Storage Scale event callback process.
- HDFS Transparency NameNode log now contains the HDFS Transparency full version information and the **`gpfs.encryption.enable`** value.
- Added general security fixes and CVE-2020-4851 in IBM Support.
- Added a new custom `json` file method for the Kerberos script. For more information, see "Configuring Kerberos using the Kerberos script provided with IBM Storage Scale" on page 117.

Changes in IBM Storage Scale Big Data Analytics Integration Toolkit for HDFS Transparency (Toolkit for HDFS) 1.0.3.0

- IBM Storage Scale installation toolkit now uses Ansible® configuration.
- Creating multiple CES HDFS clusters in the installation toolkit at the same deployment run is not supported under Ansible-based toolkit. For workaround, see Multi-HDFS cluster deployment through IBM Storage Scale 5.1.1.0 installation toolkit is not supported.

Summary of changes as updated, March 2021

Changes in IBM Storage Scale CES HDFS Transparency

- IBM Storage Scale CES HDFS Transparency now supports both the NameNode HA and non-HA options. Also, DataNode can now have Hadoop services colocated within the same node. For more information, see "Alternative architectures" on page 291.

Changes in Mpack version 2.7.0.9

- The Ambari maintenance mode for clusters is now supported by the IBM Storage Scale service on **`gpfs.storage.type`** with *shared* or *remote* environments. Earlier, when the user performed a **`Start all`** or **`Stop all`** operation from the Ambari GUI, the IBM Storage Scale service or its components that are used to start or stop respectively even when they were set to maintenance mode.

- The Mpack upgrade process does not reinitialize the following HDFS parameters to the Mpack's recommended settings:
  - **dfs.client.read.shortcircuit**
  - **dfs.datanode.hdfs-blocks-metadata.enabled**
  - **dfs.ls.limit**
  - **dfs.datanode.handler.count**
  - **dfs.namenode.handler.count**
  - **dfs.datanode.max.transfer.threads**
  - **dfs.replication**
  - **dfs.namenode.shared.edits.dir**

  Earlier any updates to these parameters by the end user were overwritten. As this issue is now fixed, any customized `hdfs-site.xml` configuration will not be changed during the upgrade process.
- In addition to **Check Integration Status** option in the Ambari service, you can now view the Mpack version/build information in `version.txt` in the `Mpack tar.gz` package.
- The hover message for the **GPFS Quorum Nodes** text field within the IBM Storage Scale service GUI has been updated. The hostnames to be entered for the Quorum Nodes should be from the IBM Storage Scale Admin network hostnames.
- The Mpack uninstaller script cleans up the IBM Storage Scale Ambari stale link that is no longer required. Therefore, the Ambari server restart will not fail because of the Mpack dependencies.
- The Mpack installation, upgrade, and uninstall script now supports the sudo root permission.
- The anonymous UID verification is checked only if **hadoop.security.authentication** is not set to *Kerberos*.
- The IBM Storage Scale service can now monitor the status of configured file system mount point (**gpfs.mnt.dir**).

  In earlier releases of Mpack, the IBM Storage Scale service was able to monitor only the status of the IBM Storage Scale runtime daemon.
  If any of the configured file system is not mounted on the IBM Storage Scale node, the status for the GPFS_NODE component for that node will now appear as down in the Ambari GUI.

Summary of changes as updated, January 2021

Changes in Cloudera Data Platform Private Cloud Base

Cloudera Data Platform Private Cloud Base with IBM Storage Scale is supported on Power®. For more information, see "Support Matrix" on page 294.

Changes in HDFS Transparency 3.1.0-7

- Fixed the `NullPointerException` error message that appeared in the NameNode logs.
- Fixed the JMX output to correctly report "open" operations when the **gpfs.ranger.enabled** parameter is set to *scale*.
- A vulnerability in IBM Storage Scale allows injecting malicious content into the log files. For the security fix information, see IBM Support.

Documentation update

Configuration options for using multiple threads to list a directory and load the metadata of its children are provided for HDFS Transparency 3.1.1-3 and 3.1.0-6. For more information, see the `list` option.

Summary of changes as updated, December 2020

Changes in HDFS Transparency 3.1.1-3

- HDFS Transparency implements performance enhancement by using fine-grained file system locking mechanism. After HDFS Transparency 3.1.1-3 is installed, ensure that the **gpfs.ranger.enabled** field is set to *scale* in /var/mmfs/hadoop/etc/hadoop/gpfs-site.xml. For more information, see "Setting configuration options in CES HDFS" on page 69.
- The create Hadoop users and groups script and the create Kerberos principals and keytabs script in IBM Storage Scale now reside in the /usr/lpp/mmfs/hadoop/scripts directory.
- Requires Python 3.6 or later.

Changes in IBM Storage Scale Big Data Analytics Integration Toolkit for HDFS Transparency (Toolkit for HDFS) 1.0.2-1

- The toolkit installation failure due to nodes that are not a part of the CES HDFS cluster and does not have JAVA installed and do not have JAVA_HOME set is now fixed.
- The following proxyuser configurations were added into core-site.xml by the installation toolkit to configure a CES HDFS cluster:

```
hadoop.proxyuser.livy.hosts=*
hadoop.proxyuser.livy.groups=*
hadoop.proxyuser.hive.hosts=*
hadoop.proxyuser.hive.groups=*
hadoop.proxyuser.oozie.hosts=*
hadoop.proxyuser.oozie.groups=*
```

Changes in IBM Storage Scale Cloudera Custom Service Descriptor (CDP CSD) 1.0.0-0

- Integrates IBM Storage Scale service into CDP Private Cloud Base Cloudera Manager.

Summary of changes as updated, November 2020

Changes in HDFS Transparency 3.1.1-2

- Supports CDP Private Cloud Base. For more information, see "Support Matrix" on page 294.
- Includes Hadoop sample scripts to create users and groups in IBM Storage Scale and set up the Kerberos principals and keytabs. Requires Python 3.6 or later.
- Summary operations (for example, du, count, and so on) in HDFS Transparency can be now done multi-threaded based on the number of files and subdirectories. It improves the performance when performing the operation on a path that contains numerous files and subdirectories. The performance improvement depends on the system environment. For more information, see Functional limitations.

Changes in IBM Storage Scale Big Data Analytics Integration Toolkit for HDFS Transparency (Toolkit for HDFS) 1.0.2-0

- Added support to deploy CES HDFS in SLES 15 and Ubuntu 20.04 on x86_64 platforms.
- Package was renamed from bda_integration-<version>.noarch.rpm to gpfs.bda-integration-<version>.noarch.rpm .
- Requires Python 3.6 or later.

Changes in IBM Storage Scale Cloudera Custom Service Descriptor (CDP CSD) 1.0.0-0 EA

- Integrates IBM Storage Scale service into CDP Private Cloud Base Cloudera Manager.

Summary of changes as updated, October 2020

Changes in HDFS Transparency 3.1.0-6

- HDFS Transparency now implements performance enhancement by using the fine-grained file system locking mechanism instead of using the Apache Hadoop global file system locking mechanism. From HDFS Transparency 3.1.0-6, set **gpfs.ranger.enabled** to *scale* from the HDP Ambari GUI under the

IBM Storage Scale service configuration page. If you are not using Ambari, set **gpfs.ranger.enabled** in /var/mmfs/hadoop/etc/hadoop/gpfs-site.xml as follows:

```
<property>
<name>gpfs.ranger.enabled</name>
<value>scale</value>
<final>false</final>
</property>
```

Note: The scale option replaces the original *true*/*false* values.

- Summary operations (for example, du, count, and so on) in HDFS Transparency can be now done multi-threaded based on the number of files and subdirectories. It improves the performance when performing the operation on a path that contains numerous files and subdirectories. The performance improvement depends on the system environment. For more information, see Functional limitations.

Summary of changes as updated, August 2020

Changes in Mpack version 2.7.0.8

For Mpack 2.7.0.7 and earlier, a restart of the IBM Storage Scale service would overwrite the IBM Storage Scale customized configuration if the **gpfs.storage.type** parameter was set to *shared*.

From Mpack 2.7.0.8, if the **gpfs.storage.type** parameter is set to *shared* or *shared,shared*, the IBM Storage Scale service will not set the IBM Storage Scale tunables, that are seen under the IBM Storage Scale service, back to the IBM Storage Scale cluster or file system.

Summary of changes as updated, July 2020

Changes in IBM Storage Scale Big Data Analytics Integration Toolkit for HDFS Transparency (Toolkit for HDFS) 1.0.1.1

- Supports rolling upgrade of HDFS Transparency through installation toolkit.

  Note: If the SMB protocol is enabled, all protocols are required to be offline for some time because the SMB does not support the rolling upgrade.
- Requires IBM Storage Scale 5.0.5.1 and HDFS Transparency 3.1.1-1. For more information, see CES HDFS "HDFS Transparency support matrix" on page 27.
- From IBM Storage Scale 5.0.5.1, only one CES-IP is needed for one HDFS cluster during installation toolkit deployment.

Changes in HDFS Transparency 3.1.0-5

- When **gpfs.replica.enforced** is set to *gpfs*, client replica setting is not honored. Convert the WARN namenode.GPFSFs (GPFSFs.java:setReplication(123)) - Set replication operation invalid when gpfs.replica.enforced is set to gpfs message to Debug, because this message can occur many times in the NameNode log.
- Fixed NameNode hangs when you are running the mapreduce jobs because of the lock synchronized issue.
- From IBM Storage Scale 5.0.5, the **gpfs.snap --hadoop** can access the HDFS Transparency logs from the user configured directories.
- From HDFS Transparency 3.1.0-5, the default value for **dfs.replication** is *3* and **gpfs.replica.enforced** is *gpfs*. Therefore, it uses the IBM Storage Scale file system replication and not the Hadoop HDFS replication. Also, increasing the **dfs.replication** value to *3* helps the hdfs client to tolerate the DataNode failures.

  Note: You need to have at least three DataNodes when you set the **dfs.replication** to *3*.
- Changed permission mode for editlog files to 640.
- For two file systems, HDFS Transparency ensures that the NameNodes and DataNodes are stopped before unmounting the second file system mount point.

Note: The local directory path for the second file system mount usage is not removed. Ensure this local directory path is empty before starting the NameNode.

- HDFS Transparency does not manage the storage. Therefore, the Apache Hadoop block function call used for native HDFS gives a false metric information. Therefore, HDFS Transparency does not run the Apache Hadoop block function calls.
- Delete operations in HDFS Transparency can be now done multi-threaded based on the number of files and subdirectories. It improves performance when deleting a path that contains numerous files and subdirectories. The performance improvement depends on the system environment. For more information, see Functional limitations.

Changes in Mpack version 2.7.0.7

- Supports HDP upgrade with Mpack 2.7.0.7 without unintegrating HDFS Transparency. .
- The Mpack 2.7.0.7 supports Ambari version 2.7.4 or later.
- The installation and upgrade scripts now support complex KDC password when Kerberos is enabled.
- You can now upgrade from older Mpacks (versions 2.7.0.x) to Mpack 2.7.0.7 if Kerberos is enabled without using the workaround .
- The upgrade postEU process is now simplified and can now automatically accept the user agreement license.
- The upgrade postEU option now requests the user inputs only once during the upgrade process.
- During the Mpack installation or upgrade process, the backup directory that is created by the Mpack installer now includes a date timestamp added to the directory name.
- The Check Integration Status UI action in IBM Storage Scale service now shows the unique Mpack build ID.
- If you are enabling Kerberos after integrating IBM Storage Scale service, ZKFC initialization used to fail because the `hdfs_jaas.conf` file was missing. A workaround is no longer required.
- Ambari now supports rolling restart for NameNodes and DataNodes.
- The configuration changes will be in effect after you restart the NameNodes and DataNodes and do not require all the HDFS Transparency nodes to be restarted.
- If the SSL is enabled, the upgrade script asks for the hostname instead of the IP address.
- The upgrade script requesting true/false inputs are no longer case sensitive.
- When deployment type is set to `gpfs.storage.type=shared`, a local GPFS cluster would be created even if the bidirectional passwordless ssh was not set up properly between the GPFS Master and the ESS contact node. This issue is now fixed. The deployment fails in such scenarios and an error message is displayed.
- If you are using IBM Storage Scale 4.2.3.2, Ambari service hangs because the **mmchconfig** would be prompting for an ENTER feedback for the **LogFileSize** parameter. From Mpack 2.7.0.7, the **LogFileSize** configuration cannot be modified. The **LogFileSize** parameter can be configured only through the command line by using the **mmchconfig** command.

Summary of changes as updated, May 2020

Changes in IBM Storage Scale Big Data Analytics Integration Toolkit for HDFS Transparency (Toolkit for HDFS) 1.0.1.0

- Supports offline upgrade of HDFS Transparency.
- Requires IBM Storage Scale 5.0.5 and HDFS Transparency 3.1.1-1. For more information, see CES HDFS "HDFS Transparency support matrix" on page 27.

Changes in HDFS Transparency 3.1.1-1

- A check is performed while you are running the **mmhdfs config upload** command to ensure that the `ces_group_name` is consistent with the HDFS Transparency `dfs.nameservices` values.

- From IBM Storage Scale 5.0.5, the **gpfs.snap --hadoop** can now access the HDFS Transparency logs from the user-configured directories.
- From HDFS Transparency 3.1.1-1, the default value for **dfs.replication** is *3* and **gpfs.replica.enforced** is *gpfs*. Therefore, it uses the IBM Storage Scale file system replication and not the Hadoop HDFS replication. Also, increasing the **dfs.replication** value to *3* helps the hdfs client to tolerate the DataNode failures.

  Note: You need to have at least three DataNodes when you set the **dfs.replication** to *3*.
- Fixed NameNode hangs when you are running the mapreduce jobs because of the lock synchronized issue.

CES HDFS changes

- From IBM Storage Scale 5.0.5, HDFS Transparency version 3.1.1-1 and Big Data Analytics Integration Toolkit for HDFS Transparency (Toolkit for HDFS) version 1.0.1.0, HDFS Transparency and Toolkit for HDFS packages are signed with a GPG (GNU Privacy Guard) key and can be deployed by the IBM Storage Scale installation toolkit.

  For more information, go to IBM Storage Scale documentation and see the following topics:

  – *Installation toolkit changes* subsection under the *Summary of changes* topic.

  – *Limitations of the installation toolkit* topic under the **Installing** > **Installing IBM Spectrum Scale on Linux nodes and deploying protocols** > **Installing IBM Spectrum Scale on Linux nodes with the installation toolkit**.

Summary of changes as updated, March 2020

Changes in IBM Storage Scale Big Data Analytics Integration Toolkit for HDFS Transparency (Toolkit for HDFS) 1.0.0.1

- Supports deployment on ESS.
- Supports remote mount file system only for CES HDFS protocol.
- Requires IBM Storage Scale 5.0.4.3 and HDFS Transparency 3.1.1-0. For more information, see CES HDFS "HDFS Transparency support matrix" on page 27.

Summary of changes as updated, January 2020

Changes in HDFS Transparency 3.1.1-0

- Integrates with CES protocol and IBM Storage Scale installation toolkit.
- Supports Open Source Apache Hadoop distribution and Red Hat Enterprise Linux operating systems.

Changes in HDFS Transparency 3.1.0-4

- Export NODE_HDFS_MAP_GPFS commented line into `hadoop-env.sh` file for **mmhadoopctl** multi-network usage.
- Fixed data replicate with AFM DR disk usage due to shrinkfit.
- Fixed Job will not fail if one DataNode failed when using gpfs.replica, enforced=gpfs, gpfs.storage.type and dfs.replication > 1 in shared mode.
- Change to log warning messages for outdated clusterinfo and diskinfo files.
- Fixed deleting a file issue on the 2nd file system when trash is enabled in a two file system configuration.
- Use the default community-defined port number for dfs.datanode (address, ipc.address, and http.address) to reduce port conflicts with ephemeral ports.
- Fixed **hadoop df** output that was earlier not consistent with the **POSIX df** output when 2 FS is configured.
- Fixed **dfs -du** that was earlier displaying wrong free space value.

Changes in Mpack version 2.7.0.6

- Supports HDP 3.1.5.

Summary of changes as updated, November 2019

Changes in Mpack version 2.7.0.5

- The Mpack Installation script `SpectrumScaleMPackInstaller.py` will no longer ask for the KDC credentials, even when the HDP Hadoop cluster is Kerberos enabled. The KDC credentials are only required to be setup before executing the IBM Storage Scale service Action "Unintegrated Transparency".
- If you are deploying the IBM Storage Scale service in a shared storage configuration (`gpfs.storage.type=shared`), the Mpack will check for consistency of UID, GID of the **anonymous** user only on the local GPFS nodes. The Mpack will not perform this check on the ESS nodes.
- If you are deploying the IBM Storage Scale service with two file system support with `gpfs.storage.type=shared,shared` or `gpfs.storage.type=remote,remote`, then the Block Replication in HDFS (**`dfs.blocksize`**) will default to 1.
- From Mpack 2.7.0.5, the issue of having all the nodes managed by Ambari to be set as GPFS nodes during deployment is fixed. For example, if you set some nodes as Hadoop client nodes and some nodes as GPFS nodes for HDFS Transparency NameNode and DataNodes, the deployment will succeed.
- In Mpack 2.7.0.4, if the **`gpfs.storage.type`** was set to *shared*, stopping the Scale service from Ambari would report a failure in the UI even if the operation had succeeded internally. This issue has been fixed in Mpack 2.7.0.5.
- IBM Storage Scale Ambari deployment can now support `gpfs.storage.type=shared,shared` mode.

Summary of changes as updated, October 2019

IBM Erasure Code Edition (ECE) is supported as shared storage mode for Hadoop with HDFS Transparency 3.1.0-3 and IBM Storage Scale 5.0.3.

Summary of changes as updated, September 2019

Changes in HDFS Transparency 3.1.0-3

- Validate open file limit when starting Transparency.
- **`mmhadoopctl`** supports dual network configuration when NODE_HDFS_MAP_GPFS is set in `/var/mmfs/hadoop/etc/hadoop/hadoop-env.sh`. See section "mmhadoopctl supports dual network" on page 204 for more details.

Changes in Mpack version 2.7.0.4

- For FPO clusters, the **`restripeOnDiskFailure`** value will be set to *NO* regardless of the original set value during the stopping of GPFS main components. After the GPFS main stop completes, the **`restripeOnDiskFailure`** value will be set back to its original value.
- The IBM Storage Scale service will do a graceful shutdown and will no longer do a force unmount of the GPFS file system via **`mmunmount -f`**.
- Seeing intermittent failure of one of the HDFS Transparency NameNodes at the startup due to the timing issue when both the NameNode HA and Kerberos are enabled has now been fixed.
- The HDFS parameter **`dfs.replication`** is set to the **`mmlsfs -r`** value (Default number of data replicas) of the GPFS file system for **`gpfs.storage.type`**=shared instead of the Hadoop replication value of 3.
- The Mpack installer (*.bin) file can now accept the license silently when the **`--accept-licence`** option is specified.

Summary of changes as updated, May 2019

Changes in HDFS Transparency 3.1.0-2

- Issue fixed when a map reduce task fails after running for one hour when the Ranger is enabled.
- Issue fixed when Hadoop permission settings do not work properly in a kerberized environment.

Documentation updates

- Updated the Migrating IOP to HDP for BI 4.2.5 and HDP 2.6 information.

Summary of changes as updated, March 2019

Changes in Mpack version 2.7.0.3

- Supports dual network configuration
- Issue fixed to look only at the first line in the `shared_gpfs_node.cfg` file to get the host name for shared storage so the deployment of shared file system would not hang.
- Removed **`gpfs_base_version`** and **`gpfs_transparency_version`** fields from the IBM Storage Scale service configuration GUI. This removes the restart all that is required after IBM Storage Scale is deployed.
- Mpack can now find the correct installed HDP version when multiple HDP versions are seen.
- IBM Storage Scale service is now able to handle hyphenated file system names so that the service will be able to start properly during file system mount.
- IBM Storage Scale entry into `system_action_definitions.xml` is fixed. Therefore, the IBM Storage Scale </actionDefinition> ending tag is not on the same line as the </actionDefinitions> tag. Otherwise, there is a potential installation issue when a new service is added after IBM Storage Scale service because the new service is added in between the IBM Storage Scale entry and the </actionDefinition></actionDefinitions> line.

HDFS Transparency 3.1.0-1

- Fixed Hadoop du to calculate all files under all subdirectories for the user even when the files have not been accessed.
- Supports ViewFS in HDP 3.1 with Mpack 2.7.0.3.

Summary of changes as updated, February 2019

Changes in Mpack version 2.7.0.2

- Supports HDP 3.1.
- SLES 12 SP3 support for new installs on x86 64 only.
- Upgrade the HDFS Transparency on all nodes in the IBM Storage Scale cluster instead of just upgrading it only on the NameNode and DataNodes.

Summary of changes as updated, December 2018

Changes in Mpack version 2.7.0.1

- Supports HDP 3.0.1.
- Supports preserving Kerberos token delegation during NameNode failover.
- IBM Storage Scale service Stop All/Start All service actions now support the best practices for IBM Storage Scale stop/start as per *Restarting a large IBM Storage Scale cluster* topic in the *IBM Storage Scale: Administration Guide*.
- The HDFS Block Replication parameter, **`dfs.replication`**, is automatically set to match the actual value of the IBM Storage Scale Default number of data replicas parameter, **`defaultDataReplicas`**, when adding the IBM Storage Scale service for remote mount storage deployment model.

HDFS Transparency 3.1.0-0

- Supports preserving Kerberos token delegation during NameNode failover.
- Fixed CWE/SANS security exposures in HDFS Transparency.
- Supports Hadoop 3.1.1

Summary of changes as updated, October 2018

Changes in Mpack version 2.4.2.7

- Supports preserving Kerberos token delegation during NameNode failover.
- IBM Storage Scale service Stop All/Start All service actions now support the best practices for IBM Storage Scale stop/start as per *Restarting a large IBM Storage Scale cluster* topic in the *IBM Storage Scale: Administration Guide*.

HDFS Transparency 2.7.3-4

- Supports preserving Kerberos token delegation during NameNode failover.
- Supports native HDFS encryption.
- Fixed CWE/SANS security exposures in HDFS Transparency.

Summary of changes as updated, August 2018

Changes in Mpack version 2.7.0.0

- Supports HDP 3.0.

Changes in HDFS Transparency version 3.0.0-0

- Supports HDP 3.0 and Mpack 2.7.0.0.
- Supports Apache Hadoop 3.0.x.
- Support native HDFS encryption.
- Changed IBM Storage Scale configuration location from `/usr/lpp/mmfs/hadoop/etc/` to `/var/mmfs/hadoop/etc/` and default log location for open source Apache from `/usr/lpp/mmfs/hadoop/logs` to `/var/log/transparency`.

New documentation sections

- Hadoop Scale Storage Architecture
- Hadoop Performance tuning guide
- Hortonworks Data Platform 3.X for HDP 3.0
- Open Source Apache Hadoop

Summary of changes as updated, July 2018

Changes in Mpack version 2.4.2.6

- HDP 2.6.5 is supported.
- Mpack installation resumes from the point of failure when the installation is re-run.
- The **Collect Snap Data** action in the IBM Storage Scale service in the Ambari GUI can capture the Ambari agents' logs in to a tar package under the `/var/log/ambari.gpfs.snap*` directory.
- Use cases where the Ambari server and the GPFS main are colocated on the same host but are configured with multiple IP addresses are handled within the IBM Storage Scale service installation.
- On starting IBM Storage Scale from Ambari, if a new kernel version is detected on the IBM Storage Scale node, the GPFS portability layer is automatically rebuilt on that node.

- On deploying the IBM Storage Scale service, the Ambari server restart is not required. However, the Ambari server restart is still required when running the **Service Action** > **Integrate Transparency** or **Unintegrate Transparency** from the Ambari UI.

Summary of changes as updated, May 2018

Changes in HDFS Transparency 2.7.3-3

- Non-root password-less login of contact nodes for remote mount is supported.
- When the Ranger is enabled, uid greater than 8388607 is supported.
- Hadoop storage tiering is supported.

Changes in Mpack version 2.4.2.5

- HDP 2.6.5 is supported.

Summary of changes as updated, February 2018

Changes in HDFS Transparency 2.7.3-2

- Snapshot from a remote-mounted file system is supported.
- IBM Storage Scale fileset-based snapshot is supported.
- HDFS Transparency and IBM Storage Scale Protocol SMB can coexist without the SMB ACL controlling the ACL for files or directories.
- HDFS Transparency rolling upgrade is supported.
- Zero shuffle for IBM ESS is supported.
- Manual update of file system configurations when root password-less access is not available for remote cluster is supported.

Changes in Mpack version 2.4.2.4

- HDP 2.6.4 is supported.
- IBM Storage Scale admin mode central is supported.
- The `/etc/redhat-release` file workaround for CentOS deployment is removed.

Summary of changes as updated, January 2018

Changes in Mpack version 2.4.2.3

- HDP 2.6.3 is supported.

Summary of changes as updated, December 2017

Changes in Mpack version 2.4.2.2

- The Mpack version 2.4.2.2 does not support migration from IOP to HDP 2.6.2. For migration, use the Mpack version 2.4.2.1.
- From IBM Storage Scale Mpack version 2.4.2.2, new configuration parameters have been added to the Ambari management GUI. These configuration parameters are as follows:

  `gpfs.workerThreads` defaults to *512*.

  `NSD threads per disk` defaults to *8*.

  For IBM Storage Scale version 4.2.0.3 and later, **`gpfs.workerThreads`** field takes effect and **`gpfs.worker1Threads`** field is ignored. For versions lower than 4.2.0.3, **`gpfs.worker1Threads`** field takes effect and **`gpfs.workerThreads`** field is ignored.

  `Verify if the disks are already formatted as NSDs` - defaults to *yes*
- The default values of the following parameters have changed. The new values are as follows:

gpfs.supergroup defaults to *hdfs,root* now instead of *hadoop,root*.

gpfs.syncBuffsPerIteration defaults to *100*. Earlier it was *1*.

Percentage of Pagepool for Prefetch defaults to *60* now. Earlier it was *20*.

gpfs.maxStatCache defaults to *512* now. Earlier it was *100000*.

- The default maximum log file size for IBM Storage Scale has been increased to 16 MB from 4 MB.

Summary of changes as updated, October 2017

Changes in Mpack version 2.4.2.1 and HDFS Transparency 2.7.3-1

- The GPFS Ambari integration package is now called the IBM Storage Scale Ambari management pack (in short, management pack or MPack).
- Mpack 2.4.2.1 is the last supported version for BI 4.2.5.
- IBM Storage Scale Ambari management pack version 2.4.2.1 with HDFS Transparency version 2.7.3.1 supports BI 4.2/BI 4.2.5 IOP migration to HDP 2.6.2.
- The remote mount configuration in Ambari is supported. (For HDP only)
- Support for two IBM Storage Scale file systems/deployment models under one Hadoop cluster/Ambari management. (For HDP only)

  This allows you to have a combination of IBM Storage Scale deployment models under one Hadoop cluster. For example, one file system with shared-nothing storage (FPO) deployment model along with one file system with shared storage (ESS) deployment model under single Hadoop cluster.
- Metadata operation performance improvements for Ranger enabled configuration.
- Introduction of Short circuit write support for improved performance where HDFS client and Hadoop DataNodes are running on the same node.

Chapter 1. Big data and analytics support

Analytics is defined as the discovery and communication of meaningful patterns in data. Big data analytics is the use of advanced analytic techniques against very large, diverse data sets (structured or unstructured) which can be processed through streaming or batch. Big data is a term applied to data sets whose size or type is beyond the ability of traditional data processing to capture, manage, and process the data.

Analyzing big data allows analysts, researchers, and business users to make better and faster decisions using data that was previously inaccessible or unusable. Using advanced analytics techniques such as text analytics, machine learning, predictive analytics, data mining, statistics, and natural language processing, businesses can analyze previously untapped data sources independent or together with their existing enterprise data to gain new insights resulting in significantly better and faster decisions.

IBM Storage Scale is an enterprise class software-defined storage for high performance, large scale workloads on-premises or in the cloud with flash, disk, tape, local, and remote storage in its storage portfolio. IBM Storage Scale unifies data silos, including those across multiple geographies and around the globe using Active File Management (AFM) to help ensure that data is always available in the right place at the right time with synchronous and asynchronous disaster recovery (AFM DR).

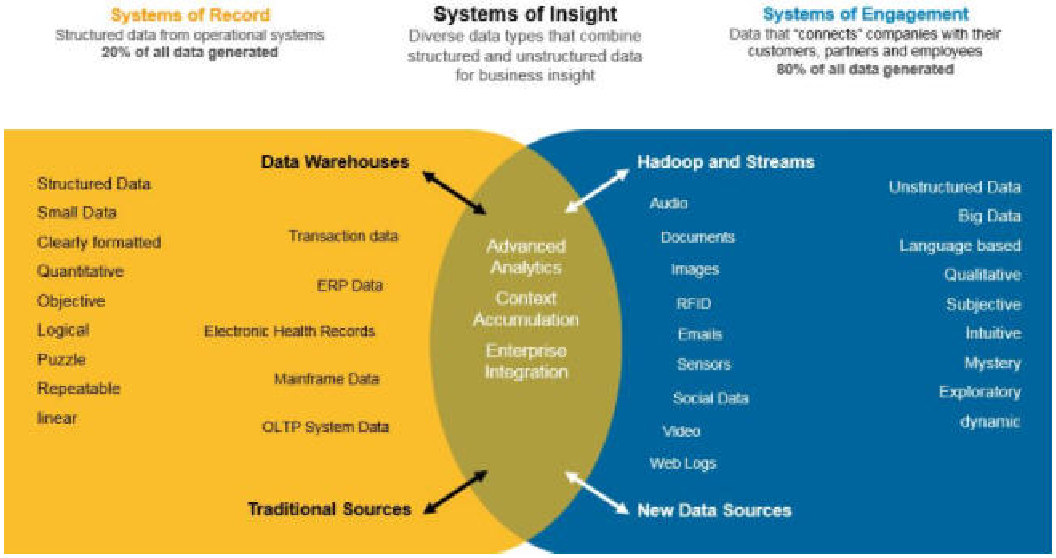
IBM Storage Scale is used for diverse workloads across every industry to deliver performance, reliability, and availability of data which are essential to the business.

This scale-out storage solution provides file, object and integrated data analytics for:

- Compute clusters (technical computing)
- Big data and analytics
- Hadoop Distributed File System (HDFS)
- Private cloud
- Content repositories
- Simplified data management and integrated information lifecycle management (ILM)

IBM Storage Scale enables you to build a data ocean solution to eliminate silos, improve infrastructure utilization, and automate data migration to the best location or tier of storage anywhere in the world to help lower latency, improve performance or cut costs. You can start small with just a few commodity servers fronting commodity storage devices and then grow to a data lake architecture or even an ocean of data. IBM Storage Scale is a proven solution in some of the most demanding environments with massive storage capacity under a single global namespace. Furthermore, your data ocean can store either files or objects and you can run analytics on the data in-place, which means that there is no need to copy the data to run your jobs. This design to provide anytime, anywhere access to data, enables files and objects to be managed together with standardized interfaces such as POSIX, OpenStack Swift, NFS, SMB/CIFS, and extended S3 API interfaces, delivering a true data without borders capability for your environments.

Decision making is a critical function in any enterprise. The decision-making process that is enhanced by analytics can be described as consuming and collecting data, detecting relationships and patterns, applying sophisticated analysis techniques, reporting, and automation of the follow-on action. The IT system that supports decision making is composed of the traditional "systems of record" and “systems of engagement” and IBM Storage Scale brings all the diverse data types of structured and unstructured data seamlessly to create a “systems of insight” for enterprise systems.



Evolving alongside Big Data Analytics, IBM Storage Scale can improve time to insight by supporting Hadoop and non-Hadoop application data sharing. Avoiding data replication and movement can reduce costs, simplify workflows, and add enterprise features to business-critical data repositories. Big Data Analytics on IBM Storage Scale can help reduce costs and increase security with data tiering, encryption, and support across multiple geographies.

Chapter 2. IBM Storage Scale support for Hadoop

IBM Storage Scale provides integration with Hadoop applications that use the Hadoop connector.

If you plan to use a Hadoop distribution with the Hadoop connector, see the chapter that corresponds to your Cloudera distribution (CDP Private Cloud Base) or the Chapter 6, "Apache Hadoop," on page 489 under the big data and analytics support documentation.

Different Hadoop connectors

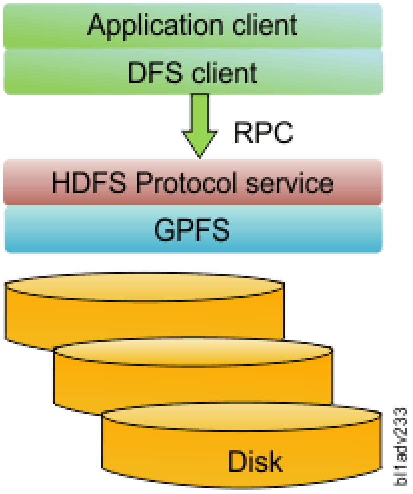
- Second generation HDFS Transparency
  - IBM Storage Scale HDFS Transparency (also known as, HDFS Protocol) offers a set of interfaces that allows applications to use HDFS Client to access IBM Storage Scale through HDFS RPC requests. HDFS Transparency implementation integrates both the NameNode and the DataNode services and responds to the request as if it were HDFS.
- First generation Hadoop connector
  - The IBM Storage Scale Hadoop connector implements Hadoop file system APIs and the FileContext class so that it can access the IBM Storage Scale.

Overview

All data transmission and metadata operations in HDFS are through the RPC mechanism and processed by the NameNode and the DataNode services within HDFS.

IBM Storage Scale HDFS protocol implementation integrates both the NameNode and the DataNode services and responds to the request as if it were HDFS. Advantages of the HDFS transparency are as follows:

- HDFS-compliant APIs or shell-interface command.
- Application client isolation from storage. Application client might access data in IBM Storage Scale file system without GPFS client installed.
- Improved security management by Kerberos authentication and encryption for RPCs.
- Simplified file system monitor by Hadoop Metrics2 integration.



In the following sections, DFS client is the node installed with HDFS client package. Hadoop Node is the node that is installed with any Hadoop-based components (such as Hive, Hbase, Pig, and Ranger). Hadoop service is the Hadoop-based application or components. HDFS Transparency node is the node running HDFS Transparency NameNode or DataNode.

Integration of Cluster Export Services (CES) protocol and deployment toolkit with HDFS Transparency are supported starting with HDFS Transparency 3.1.1 and IBM Storage Scale 5.0.4.2. For more information, see "HDFS Transparency overview" on page 10.

For information about downloading the HDFS Transparency package, see "HDFS Transparency download" on page 28.

Hadoop IBM Storage Scale Architecture

IBM Storage Scale allows Hadoop applications to access centralized storage or local storage data. All Hadoop nodes can access the storage as a GPFS™ client. You can share a cluster between Hadoop and any other application.

IBM Storage Scale has the following supported storage modes that Hadoop can access:

- Centralized Storage Mode:
  - IBM Elastic Storage® Server
  - IBM Erasure Code Edition
  - Shared Storage (SAN-based storage)
- Local Storage Mode:
  - File Placement Optimizer

Elastic Storage Server

IBM Elastic Storage Server is an optimized disk storage solution that is bundled with IBM hardware and innovative IBM Storage Scale RAID technology (based on erasure coding) that can be used to protect hardware failure instead of using data replication and offer better storage efficiency than the local storage.

It performs fast background disk rebuilds in minutes without affecting application performance. HDFS Transparency (2.7.0-1 and later) allows the Hadoop or Spark applications to access the data stored in IBM Elastic Storage Server, as illustrated in the following figure:

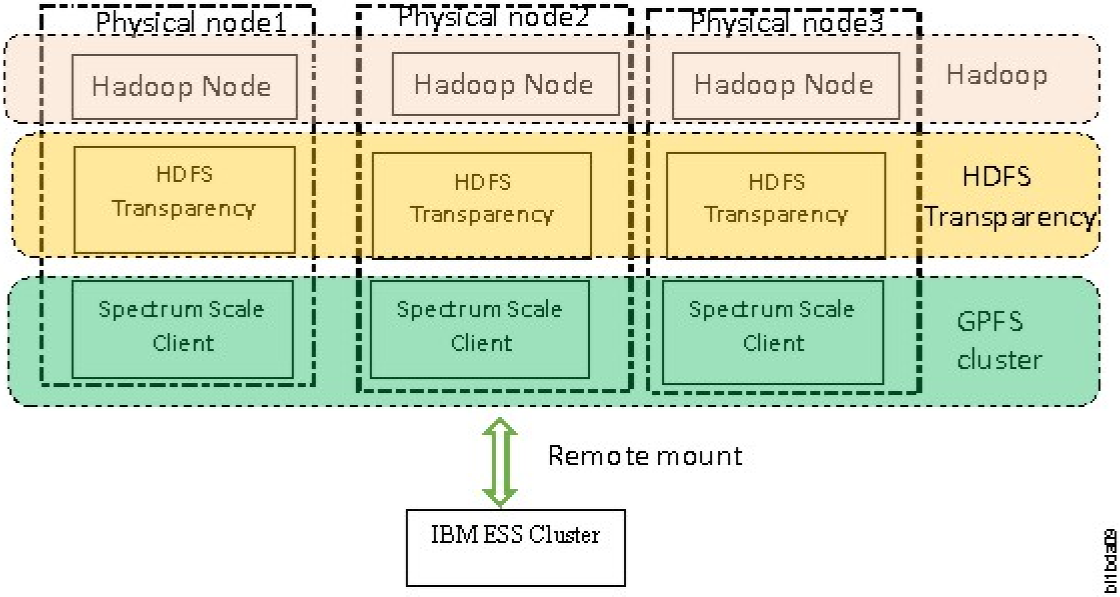


Figure 1. HDFS Transparency for IBM Elastic Storage Server

For more information, see Elastic Storage Server documentation.

Erasure Code Edition

IBM Storage Scale Erasure Code Edition (ECE) provides IBM Storage Scale RAID as software and it allows customers to create IBM Storage Scale clusters that use scale-out storage on any hardware that meets the minimum hardware requirements.

All the benefits of IBM Storage Scale and IBM Storage Scale RAID can be achieved by using existing commodity hardware.

IBM Storage Scale Erasure Code Edition provides the following features:

- Reed Solomon highly fault tolerant declustered Erasure Coding that protects against individual drive failures and node failures.
- Disk Hospital to identify issues before they become disasters.
- End-to-end checksum to identify and correct errors that are introduced by network, media, or both.
- Fast background disk rebuilds in minutes without affecting application performance.

HDFS Transparency version 3.1.0-3 and later allows the Hadoop or Spark applications to access the data stored in IBM ECE.

Note: The ECE storage must be configured as shared storage to be used in the Hadoop environment.

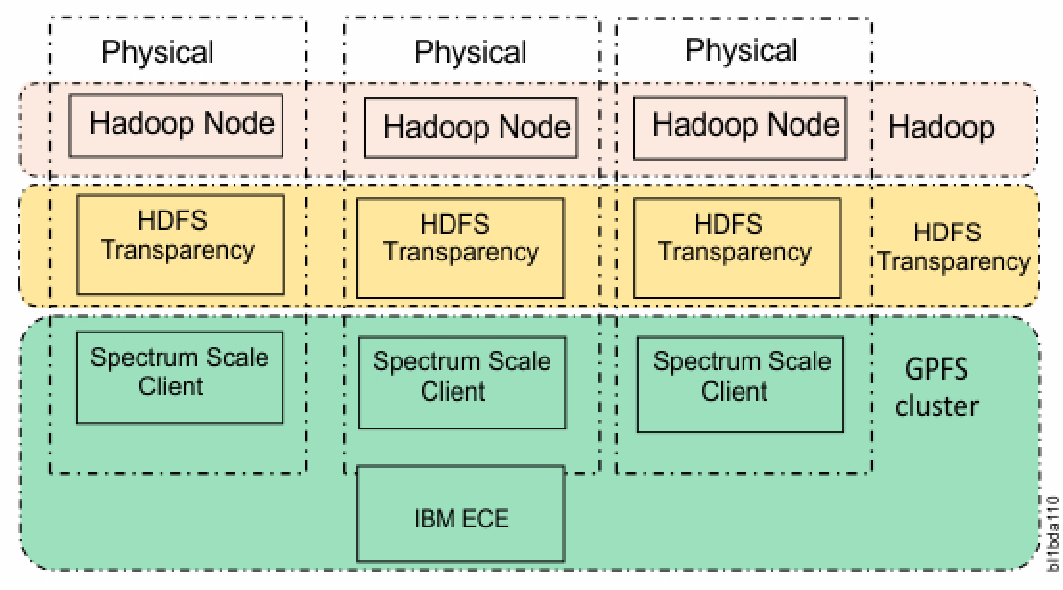


Figure 2. HDFS Transparency over ECE as shared storage mode

For more information, see the *IBM Storage Scale Erasure Code Edition* guide.

Share Storage (SAN-based storage)

HDFS Transparency (2.7.0-1 or later) allows Hadoop and Spark applications to access data stored in shared storage mode, such as IBM Storwize® V7000 etc.

This is illustrated in the following figure:

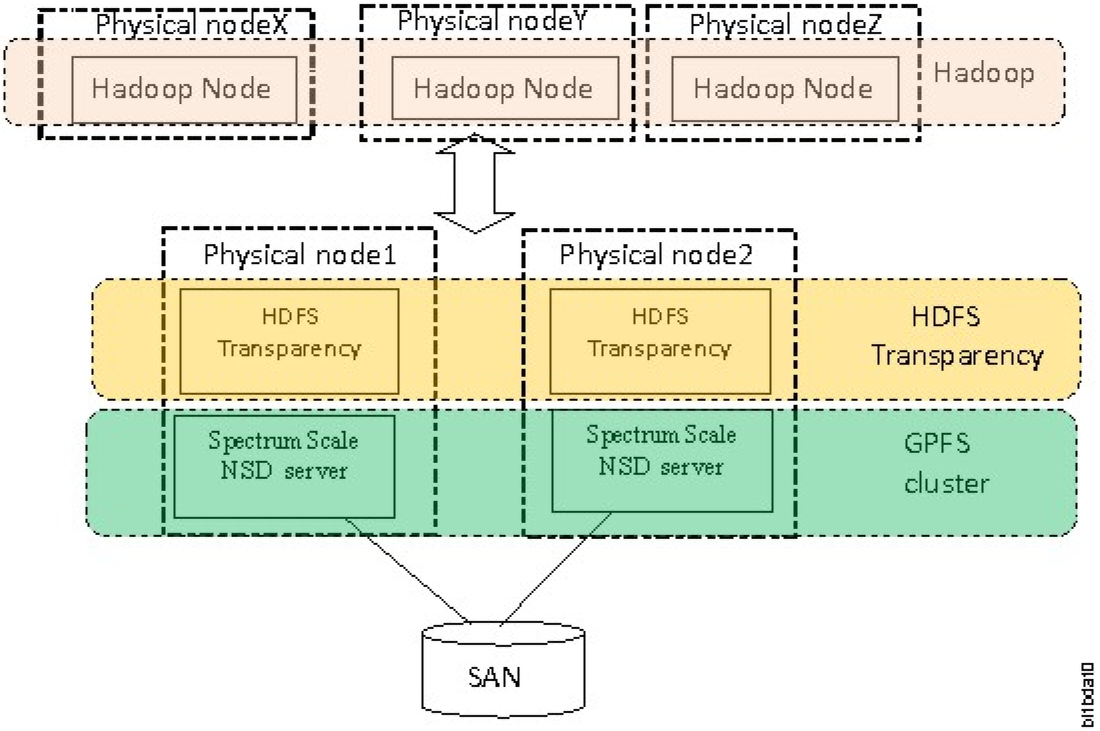


Figure 3. HDFS Transparency over IBM Storage Scale NSD for shared storage

File Placement Optimizer (FPO)

HDFS transparency allows big data applications to access IBM Storage Scale local storage - File Placement Optimizer (FPO) mode.

This is illustrated in the following figure:

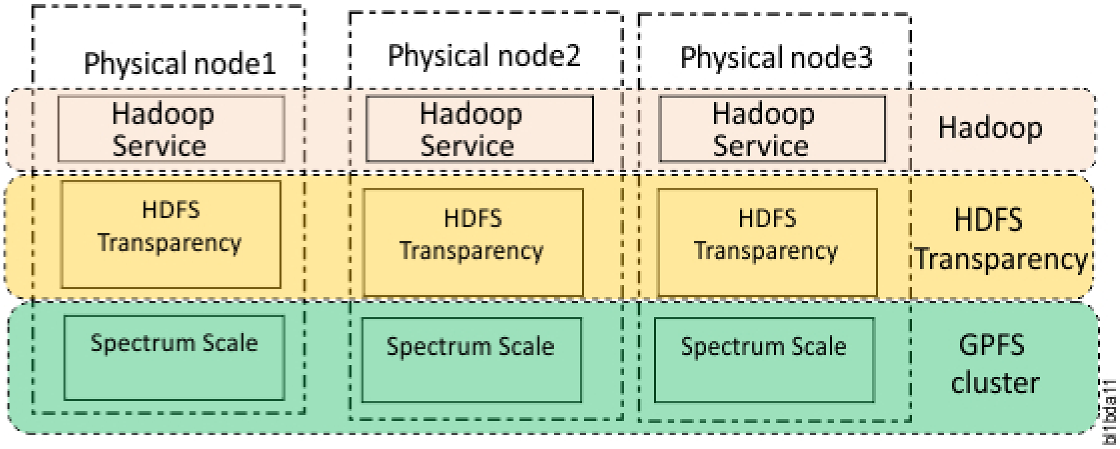


Figure 4. HDFS Transparency over IBM Storage Scale FPO

For more information, see the *File Placement Optimizer* topic in the *IBM Storage Scale: Administration Guide*.

Deployment model

A deployment model must be considered from two levels: IBM Storage Scale level and HDFS Transparency level.

From IBM Storage Scale level, the following two deployment models are available:

- Remote mount mode
- Single cluster mode

From HDFS Transparency level, the following two deployment models are available:

- All Hadoop nodes as IBM Storage Scale nodes
- Limited Hadoop nodes as IBM Storage Scale nodes

Note: Hadoop services or HDFS Transparency cannot be colocated with the ESS EMS, I/O nodes, or ECE nodes.

Model 1: Remote mount with all Hadoop nodes as IBM Storage Scale nodes

Use this model if you are using IBM Elastic Storage Server and you have small Hadoop node size, typically less than 50 Hadoop nodes.

This is illustrated in the following figure:

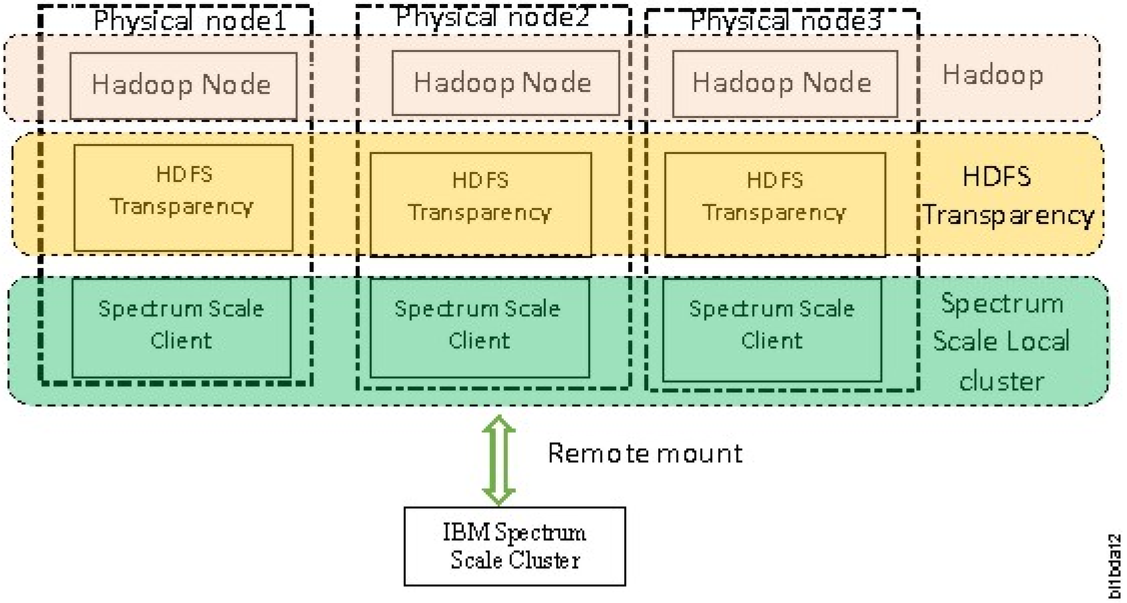


Figure 5. Remote mount with all Hadoop nodes as IBM Storage Scale nodes

This model consists of two IBM Storage Scale clusters. Configure Hadoop on the IBM Storage HDFS Transparency nodes. The Hadoop and HDFS Transparency node make up for one IBM Storage Scale cluster. The IBM Storage Scale local cluster is the IBM Storage Scale clients to the IBM Elastic Storage Server when remote mount is configured. The IBM Elastic Storage Server is the second IBM Storage Scale cluster. All the Hadoop and Spark services run on the IBM Storage Scale Hadoop local cluster.

With this model, one IBM Elastic Storage Server can be shared with different groups and the remote mount mode can isolate the storage management from the IBM Storage Scale local cluster. Some operations from local clusters (for example, **`mmshutdown -a`**) do not impact the storage side IBM Storage Scale. Meanwhile, one can enable Hadoop Short Circuit read/write to gain better I/O performance for Hadoop and Spark jobs.

Model 2: Remote mount with limited Hadoop nodes as IBM Storage Scale nodes

Use this model if you are using IBM Elastic Storage Server and you have huge Hadoop node size, typically more than 1000 Hadoop nodes.

This is illustrated by the following figure:

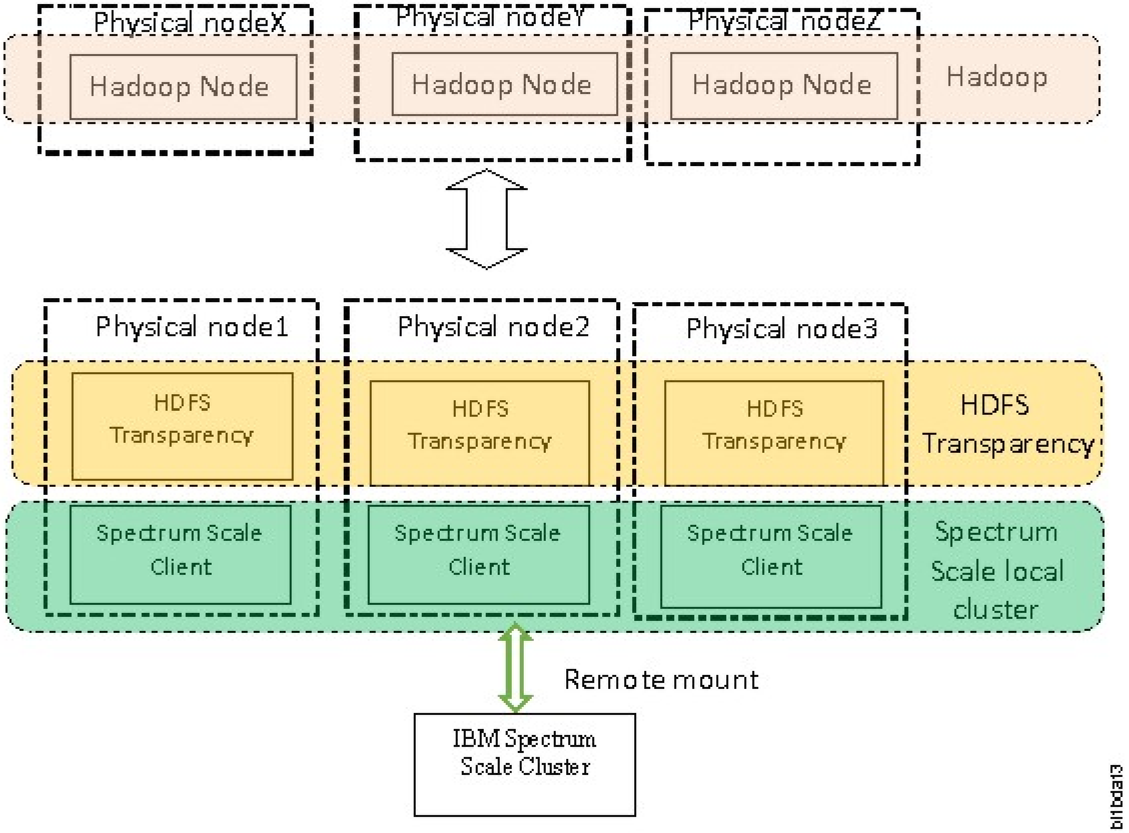


Figure 6. Remote mount with limited Hadoop nodes as IBM Storage Scale nodes

This deployment model is used for large number of nodes in the Hadoop cluster (for example, more than 1000 nodes). Creating a large IBM Storage Scale cluster requires careful planning and increased demands on the network. The deployment model in Figure 6 on page 8 limits the IBM Storage Scale deployment to just the nodes that are running the HDFS Transparency service rather than the entire Hadoop cluster. The data traffic goes from Hadoop nodes, network RPC, HDFS Transparency nodes and IBM Storage Scale Clients, network RPC, IBM Storage Scale NSD servers, and SAN storage. Short-circuit read/write configuration does not help the data reading performance.

Model 3: Single cluster with all Hadoop nodes as IBM Storage Scale nodes

Use this model if you are using IBM Storage Scale FPO.

This is illustrated in the following figure:

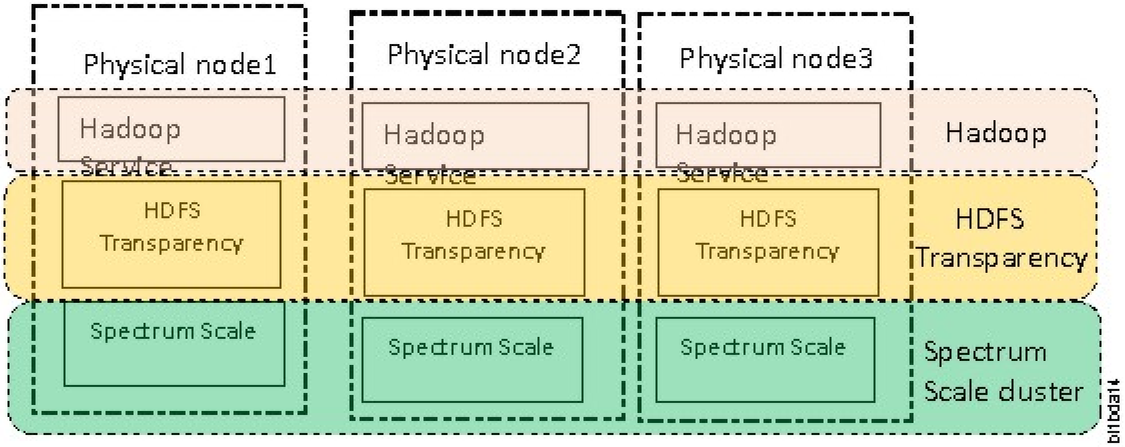


Figure 7. Single cluster with all Hadoop nodes as IBM Storage Scale nodes

In this deployment model, Hadoop/Spark jobs can leverage the data locality from IBM Storage Scale FPO.

If you are using IBM Elastic Storage Server storage, you can consider the model that is illustrated in the following figure:

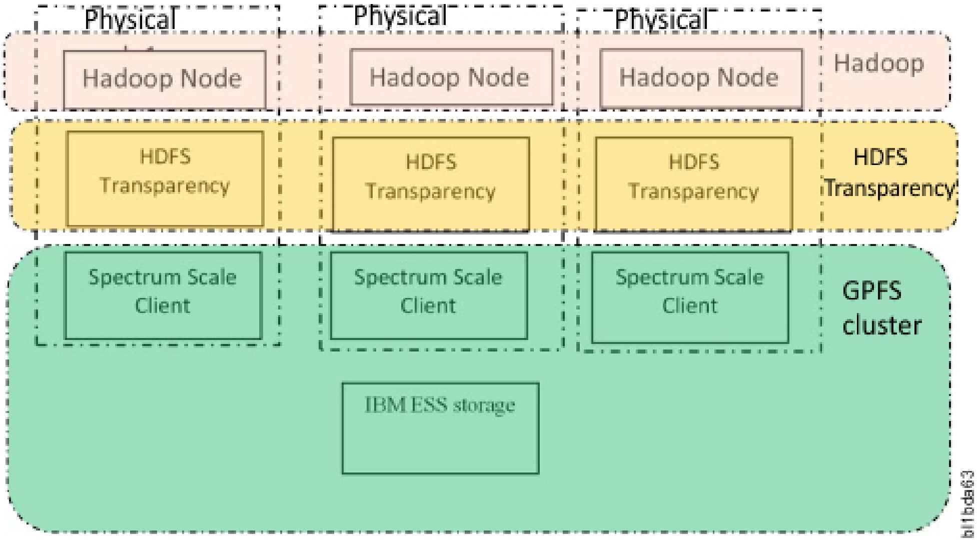


Figure 8. Single cluster with all Hadoop nodes as IBM Storage Scale nodes (Elastic Storage Server)

If you use SAN-based storage, you can consider the model that is illustrated in the following figure:

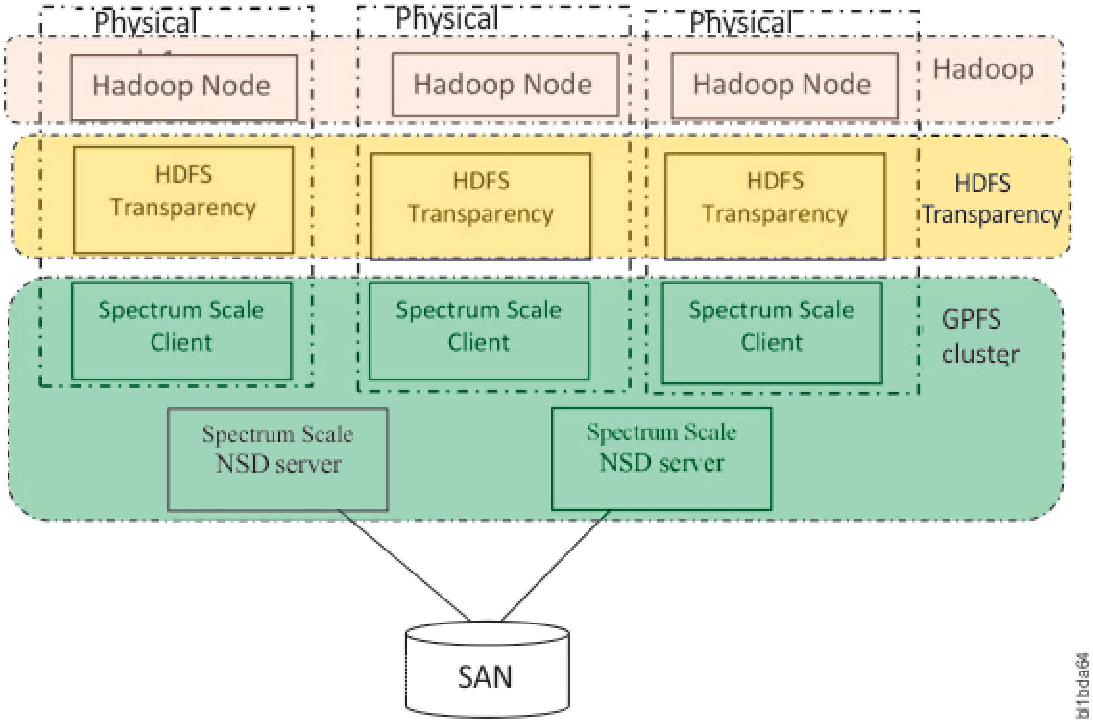


Figure 9. Single cluster with all Hadoop nodes as IBM Storage Scale nodes (SAN storage)

Model 4: Single cluster with limited Hadoop nodes as IBM Storage Scale nodes

Use this model if you are using a SAN-based storage.

This is illustrated in the following figure:

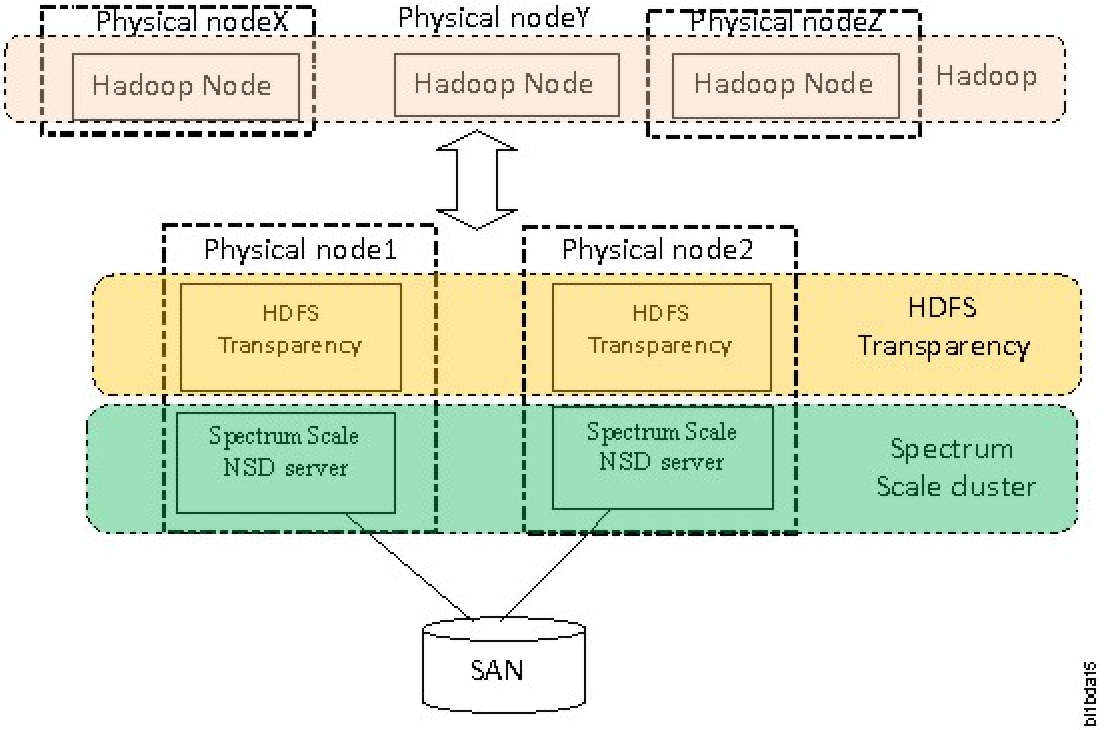


Figure 10. Single cluster with limited Hadoop nodes as IBM Storage Scale nodes

In this deployment model, HDFS Transparency services run on IBM Storage Scale NSD servers that have local connection path to SAN storages. All the other Hadoop/Spark services run on the Hadoop nodes and take network RPC to read/write data from or into IBM Storage Scale.

Additional supported storage features

This section describes the Hadoop Storage Tiering and Multiple IBM Storage Scale file system support features.

Hadoop Storage Tiering

Hadoop Storage Tiering setup can run jobs on the Hadoop cluster with native HDFS cluster and can read and write the data from IBM Storage Scale in real time. For more information, see Hadoop Storage Tiering.

Multiple IBM Storage Scale file system support

If you use multiple IBM Storage Scale clusters and you want to access them from the local IBM Storage Scale Hadoop cluster, see “Multiple IBM Storage Scale File System support” on page 191. If Ambari is available, see “Configuring multiple file system mount point access” on page 408.

HDFS Transparency overview

Starting from HDFS Transparency 3.1.1 and IBM Storage Scale 5.0.4.2, HDFS Transparency is integrated with the IBM Storage Scale installation toolkit and the Cluster Export Services (CES) protocol.

The installation toolkit automates the steps that are required to install GPFS, deploy protocols, and install updates and patches. CES provides highly available file and object services to a GPFS cluster like NFS, Object and SMB protocol support.

With the CES HDFS integration, the installation toolkit can now install HDFS Transparency as part of the CES protocol stack. The CES interface can now control and configure HDFS Transparency using the same interfaces as with the other protocols.

With the integration of HDFS into CES protocol, the use of the protocol server function requires extra licenses that need to be accepted.

For more information about the installation toolkit and CES protocol, see the *Overview of the installation toolkit* and *Protocols support overview: Integration of protocol access methods with GPFS* topic in the *IBM Storage Scale: Concepts, Planning, and Installation Guide*.

CES HDFS integration

- The installation toolkit can install and configure NameNodes and DataNodes.
- CES configures and manages only the NameNodes. A CES IP will be assigned for every CES HDFS cluster.
- Multiple HDFS clusters can be supported on the same IBM Storage Scale cluster.
- Each HDFS cluster requires to have its own CES group and cluster name where the CES group is the cluster name prefixed with “hdfs”.
- CES HDFS NameNode failover does not use ZKFailoverController. This is because CES will elect a new node to host the CES IP using its own failover mechanism. HDFS clients will always talk to the same CES IP. Therefore, NameNode failover happens transparently. The Hadoop clients require to be configured so that it only knows about one NameNode in order to work properly with the CES HDFS protocol failover functionality.
- CES HDFS protocol is installed only if it is enabled.

Planning

Learn about Hadoop distributions supported on IBM Storage Scale and aspects to consider while planning your integration with HDFS Transparency.

Hadoop cluster planning

In a Hadoop cluster that runs the HDFS protocol, a node can be a DFS Client, a NameNode, or a DataNode, or all of them. The Hadoop cluster might contain nodes that are all part of an IBM Storage Scale cluster or where only some of the nodes belong to the IBM Storage Scale cluster.

NameNode

You can specify a single NameNode or multiple NameNodes to protect against a single point of failure in the cluster. For more information, see “High availability configuration” on page 193. The NameNode must be a part of an IBM Storage Scale cluster and must have a robust configuration to reduce the chances of a single-node failure. The NameNode is defined by setting the `fs.defaultFS` parameter to the hostname of the NameNode in the `core-site.xml` file.

Note: The Secondary NameNode in native HDFS is not needed for HDFS Transparency because the HDFS Transparency NameNode is stateless and does not maintain an FSImage like state information.

DataNode

You can specify multiple DataNodes in a cluster. The DataNodes must be a part of an IBM Storage Scale cluster. The DataNodes are specified by listing their hostnames in the `workers` configuration file.

DFS Client

The DFS Client can be a part of an IBM Storage Scale cluster. When the DFS Client is a part of an IBM Storage Scale cluster, it can read data from IBM Storage Scale through an RPC or use the short-circuit mode. Otherwise, the DFS Client can access data from IBM Storage Scale only through an RPC. You can specify the NameNode address in DFS Client configuration so that DFS Client can communicate with the appropriate NameNode service.

In a production cluster, it is recommended to configure NameNode HA: one active NameNode and one standby NameNode. Active NameNode and standby NameNode must be located in two different nodes. For a small test or a POC cluster, such as 2-node or 3-node cluster, you can configure one node as

NameNode and DataNode. However, in a production cluster, it is not recommended to configure the same node as both NameNode and DataNode.

The purpose of cluster planning is to define the node roles: Hadoop node, HDFS transparency node, and GPFS node.

License planning

HDFS Transparency does not require additional license. If you have IBM Storage Scale license, you can get the HDFS Transparency package from the IBM Storage Scale package or see the HDFS Transparency download section.

As for IBM Storage Scale license, any IBM Storage Scale license works with HDFS Transparency. However, you should take IBM Storage Scale Standard Edition or IBM Storage Scale Advanced Edition or Data Management Edition so that you could leverage some advanced enterprise features (such as IBM Storage Scale storage pool, fileset, encryption or AFM) to power your Hadoop data platform.

First go through the "Hadoop IBM Storage Scale Architecture" on page 4 section, select the mode that you are planning to use and then refer the license requirements in the following table:

Table 3. IBM Storage Scale License requirement

Storage Category	**Deployment Mode**	**License requirement**
IBM Storage Scale FPO	Illustrated in Figure 7 on page 8	• 3+ IBM Storage Scale server license for manager/quorum (3 quorum tolerates 1 quorum node failure. If you want higher quorum node failure tolerance, you need to configure more quorum node/licenses, maximally up to 8 quorum nodes in one cluster). • All other nodes take the IBM Storage Scale FPO license. **Note:** If you purchase IBM Storage Scale capacity license, you do not need to purchase additional licenses mentioned above.

Table 3. IBM Storage Scale License requirement (continued)

Storage Category	Deployment Mode	License requirement
IBM Storage Scale + SAN storage	Illustrated in Figure 9 on page 9	• All NSD servers must be with IBM Storage Scale server license. • At least 2 NSD servers are required for HDFS Transparency (1 NameNode and 1 DataNode); it is recommended to take 4+ NSD servers for HDFS Transparency (1 active NameNode, 1 standby NameNode, 2 DataNodes). **Note:** If you purchase IBM Storage Scale capacity license, you do not need to purchase additional licenses mentioned above.
	Illustrated in Figure 8 on page 9 (configure one IBM Storage Scale cluster)	• 2+ IBM Storage Scale server license for quorum/NSD servers with tiebreak disks for quorum. If you want to tolerate more quorum node failure, configure more IBM Storage Scale NSD servers/quorum nodes. – All HDFS Transparency nodes should take IBM Storage Scale server license under this configuration. **Note:** If you purchase IBM Storage Scale capacity license, you do not need to purchase additional licenses mentioned above.
	Illustrated in Figure 6 on page 8 (configure IBM Storage Scale Multi-cluster)	• For home IBM Storage Scale cluster (NSD server cluster), 2+ NSD servers (IBM Storage Scale server license) configured with tiebreak disks for quorum. If you want to tolerate more quorum node failure, configure more IBM Storage Scale NSD servers/quorum nodes. • For local IBM Storage Scale cluster (all as IBM Storage Scale clients), 3+ IBM Storage Scale server license for quorum/manager (configure more IBM Storage Scale server license node to tolerate more quorum node failure); All HDFS Transparency nodes take IBM Storage Scale server license. other nodes could take IBM Storage Scale client license. **Note:** If you purchase IBM Storage Scale capacity license, you do not need to purchase additional licenses mentioned above.
	Illustrated in Figure 5 on page 7 (configure IBM Storage Scale Multi-cluster)	• For home IBM Storage Scale cluster (NSD server cluster), 2+ NSD servers (IBM Storage Scale server license) configured with tiebreak disks for quorum. If you want to tolerate more quorum node failure, configure more IBM Storage Scale NSD servers/quorum nodes. • For local IBM Storage Scale cluster (all as IBM Storage Scale clients), 3+ IBM Storage Scale server license for quorum/manager (configure more IBM Storage Scale server license node to tolerate more quorum node failure) is required. All other HDFS Transparency nodes need IBM Storage Scale server license. **Note:** If you purchase IBM Storage Scale capacity license, you do not need to purchase additional licenses mentioned above.

Table 3. IBM Storage Scale License requirement (continued)

Storage Category	Deployment Mode	License requirement
IBM ESS	Illustrated in Figure 8 on page 9 (configure one IBM Storage Scale cluster)	• If you take the ESS nodes as the quorum nodes, then you do not need to purchase the new IBM Storage Scale licenses. **Note:** Purchasing ESS will give you the license rights to use the nodes as quorum. • All other nodes take IBM Storage Scale server license. **Note:** If you purchase IBM ESS with IBM Storage Scale capacity license, you do not need to purchase additional licenses mentioned above.
	Illustrated in Figure 5 on page 7 (configure IBM Storage Scale Multi-cluster)	• Create ESS nodes as home cluster (you do not need to purchase new IBM Storage Scale license after you purchase IBM ESS). • For local IBM Storage Scale cluster (all as IBM Storage Scale clients), 3+ IBM Storage Scale server license for quorum/ manager (configure more IBM Storage Scale server license node to tolerate more quorum node failure); all other take IBM Storage Scale server license. **Note:** If you purchase IBM ESS with IBM Storage Scale capacity license, you do not need to purchase additional licenses mentioned above.
	Illustrated in Figure 6 on page 8 (configure IBM Storage Scale Multi-cluster)	• Create ESS nodes as home cluster (you do not need to purchase new IBM Storage Scale license after you purchase IBM ESS). • For local IBM Storage Scale cluster (all as IBM Storage Scale clients), 3+ IBM Storage Scale server license for quorum/ manager (configure more IBM Storage Scale server license node to tolerate more quorum node failure); all other HDFS Transparency nodes take IBM Storage Scale server license. **Note:** If you purchase IBM ESS with IBM Storage Scale capacity license, you do not need to purchase additional licenses mentioned above.

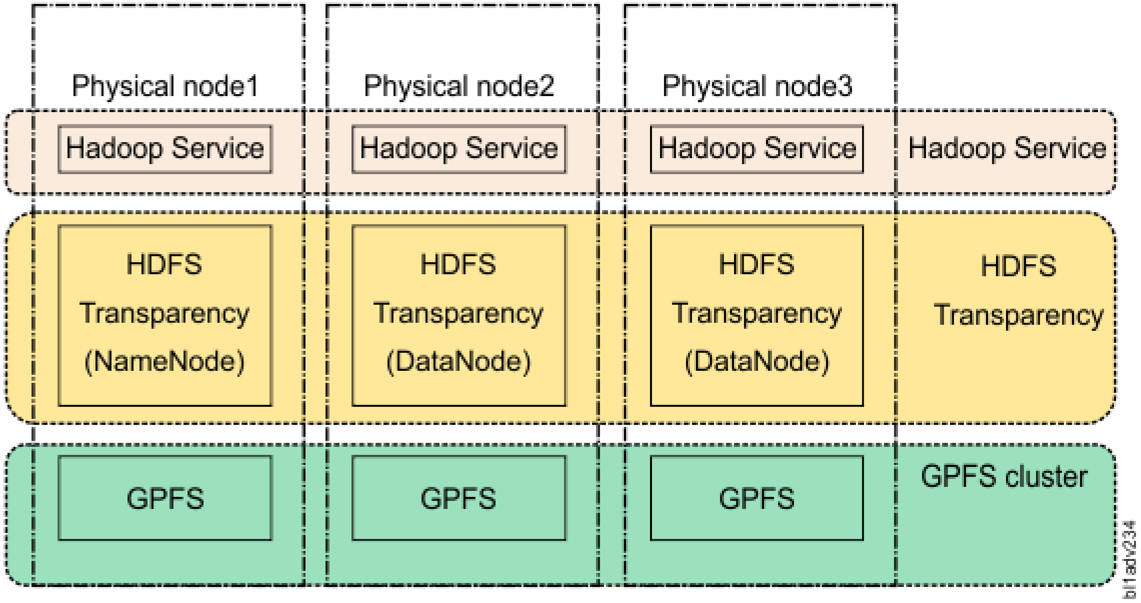
Note: If you plan to configure IBM Storage Scale protocol, you need to configure IBM Storage Scale services over nodes with IBM Storage Scale server license but no NSD disks in the file system. If you purchase IBM Storage Scale capacity license, you do not need to purchase additional licenses for IBM Storage Scale Protocol nodes.

Node roles planning

This section describes the node roles planning in FPO mode and shared storage mode and the integration with various hadoop distributions.

Node roles planning in FPO mode

In the FPO mode, all nodes are IBM Storage Scale nodes, Hadoop nodes, and HDFS Transparency nodes.



In this figure, one node is selected as the HDFS Transparency NameNode. All the other nodes are HDFS Transparency DataNodes. Also, the HDFS Transparency NameNode can be an HDFS Transparency DataNode. Any one node can be selected as HDFS Transparency HA NameNode. The administrator must ensure that the primary HDFS Transparency NameNode and the standby HDFS Transparency NameNode are not the same node.

In this mode, Hadoop cluster must be larger than or equal to the HDFS transparency cluster.

Note: The Hadoop cluster might be smaller than HDFS transparency cluster but this configuration is not typical and not recommended. Also, the HDFS transparency cluster must be smaller than or equal to IBM Storage Scale cluster because the HDFS transparency must read and write data to the local mounted file system. Usually, in the FPO mode, the HDFS transparency cluster is equal to the IBM Storage Scale cluster.

Note: Some nodes in the IBM Storage Scale (GPFS) FPO cluster might be GPFS clients without any disks in the file system.

The shared storage mode or IBM ESS

Among these nodes, you need to define at least one NameNode and one DataNode. If NameNode HA is configured, you need at least two nodes for NameNode HA and one DataNode.

In production, you need at least two DataNodes to tolerate one DataNode failure if your file system takes data replica 1. If your file system takes data replica 2 (for example, IBM Storage Scale over shared storage), you need at least three DataNodes to tolerate one DataNode failure.

After HDFS transparency nodes are selected, see "Installing" on page 29 and "Configuring" on page 52 to configure HDFS Transparency on these nodes.

Integration with Hadoop distributions

If you deploy HDFS transparency with a Hadoop distribution, such as IBM BigInsights® IOP or HortonWorks HDP, configure the native HDFS NameNode as the HDFS Transparency NameNode and configure native HDFS DataNodes as HDFS Transparency DataNodes. Add these nodes into IBM Storage

Scale cluster. This setup results in fewer configuration changes. Therefore, before installing Hadoop distribution, you need to plan the nodes as NameNode and the nodes as DataNodes.

If the HDFS Transparency NameNode is not the same as the native HDFS NameNode, some services might fail to start and can require additional configuration changes.

Hardware and software requirements

Hardware & OS matrix support

In addition to the normal operating system, IBM Storage Scale, and Hadoop requirements, the Transparency connector has minimum hardware requirements of 1 CPU (processor core) and 4 GB to 8 GB physical memory on each node where it is running. This is stated as a general guideline and actual configuration may vary.

For information about Hadoop distribution support, see "Hadoop distribution support" on page 24.

Recommended hardware resource configuration

10Gb Ethernet network is the minimum recommended configuration for Hadoop nodes. Higher speed networks, such as 25Gb/40Gb/100Gb/InfiniBand, can provide overall better performance. Hadoop nodes should have a minimum of 100GB memory and at least four physical cores. If Hadoop services are running with the same nodes as the HDFS Transparency service, a minimum of 8 physical cores is recommended. If an IBM Storage Scale FPO deployment pattern is used, 10-20 internal SAS/SATA disks per node are recommended.

In a production cluster, minimal node number for HDFS Transparency is 3. The first node as active NameNode, the second node as standby NameNode and the third node as DataNode. In testing cluster, one node is sufficient for HDFS Transparency cluster and the node could be configured as both NameNode and DataNode.

HDFS Transparency is a light-weight daemon and usually one logic modern processor (For example, 4-core or 8-core CPU with 2+GHz frequency).

For memory requirements, see the following tables:

Table 4. For HDFS Transparency 3.1.1-8 or earlier, and 3.3.0-0 and later

Ranger Support	HDFS Transparency NameNode	HDFS Transparency DataNode
Ranger support is off [1]	2GB or 4GB	2GB
Ranger support is on (by default)	Depends on the file number that the Hadoop applications will access [2]: 1024 bytes * inode number	2GB

Table 5. For HDFS Transparency 3.1.1-9 and later, and 3.2.2-0 and later

HDFS Transparency NameNode	HDFS Transparency DataNode
Depends on the file number that the Hadoop applications will access: 700 bytes * inode number.	2GB

Note: The file number means the total inode number under `/gpfs.mnt.dir/gpfs.data.dir` (refer `/usr/lpp/mmfs/hadoop/etc/hadoop/gpfs-site.xml` (for HDFS Transparency 2.7.3-x) or `/var/mmfs/hadoop/etc/hadoop/gpfs-site.xml` (for HDFS Transparency 3.0.x)).

As for SAN-based storage or IBM ESS, the number of Hadoop nodes required for scaling depends on the workload types. If the workload is I/O sensitive, you could calculate the Hadoop node number according to the bandwidth of ESS head nodes and the bandwidth of Hadoop node. For example, if the network bandwidth from your ESS head nodes is 100Gb and if your Hadoop node is configured with 10Gb network, for I/O sensitive workloads, 10 Hadoop nodes (100Gb/10Gb) will drive all network bandwidth for your

ESS head nodes. Considering that most Hadoop workloads are not pure I/O reading/writing workloads, you can take 10~15 Hadoop nodes in this configuration.

IBM ECE minimum hardware requirements

At a high level, it is required to have between 4 to 32 storage servers per recovery group (RG), and each server must be a x86_64 server running Red Hat® Enterprise Linux version 7.5 or 7.6. The storage configuration must be identical for all the storage servers. The supported storage types are SAS-attached HDD or SSD drives and using specified LSI adapters, or enterprise-class NVMe drives. Each storage server must have at least one SSD or NVMe drive. This is used for a fast write cache as well as user data storage.

For more information about hardware requirement, see *ECE Minimum hardware requirements* in the *IBM Storage Scale Erasure Code Edition* guide.

IBM Storage Scale software requirements

This section describes the software requirements for HDFS Transparency on IBM Storage Scale.

Ensure that the required packages needed by GPFS are installed on all the HDFS Transparency nodes. For more information, see the *Software Requirements* topic in the *IBM Storage Scale: Concepts, Planning, and Installation Guide*.

Kernel

IBM Storage Scale requires the Kernel packages.

Installation of Kernel packages

1. On all the IBM Storage Scale nodes, confirm that the output of **`rpm -qa |grep kernel`** includes the following:

   - kernel-headers
   - kernel-devel
   - kernel

   If any of the kernel RPM is missing, install it. If the kernel packages do not exist, run the following **`yum install`** command:

   ```
   yum -y install kernel kernel-headers kernel-devel
   ```

2. Check the installed kernel rpms. Unlike HDFS, IBM Storage Scale is a kernel-level file system that integrates with the operating system. This is a critical dependency. Ensure that the environment has the matching kernel, kernel-devel, and kernel-headers. The following example uses RHEL 7.4:

   ```
   [root@c902f05x01 ~]# uname -r
   3.10.0-693.11.6.el7.x86_64 <== Find kernel-devel and kernel-headers to match this

   [root@c902f05x01 ~]# rpm -qa | grep kernel
   kernel-tools-3.10.0-693.el7.x86_64
   kernel-headers-3.10.0-693.11.6.el7.x86_64 <== kernel-headers matches
   kernel-tools-libs-3.10.0-693.el7.x86_64
   kernel-debuginfo-3.10.0-693.11.6.el7.x86_64
   kernel-devel-3.10.0-693.11.6.el7.x86_64 <== kernel-devel matches
   kernel-3.10.0-693.el7.x86_64
   kernel-debuginfo-common-x86_64-3.10.0-693.11.6.el7.x86_64
   kernel-3.10.0-693.11.6.el7.x86_64
   kernel-devel-3.10.0-693.el7.x86_64
   [root@c902f05x01 ~]#
   ```

   ⚠ **Warning:** Kernels are updated after the original operating system installation. Ensure that the active kernel version matches the installed version of both kernel-devel and kernel-headers.

SELinux

This topic gives information about SELinux.

If you are using HDFS Transparency, from IBM Storage Scale 5.0.5, SELinux is supported in permissive or enforcing mode on Red Hat Enterprise.

If you are using Hortonworks Data Platform (HDP) in any IBM Storage Scale release, SELinux must be disabled.

If you are using Cloudera Data Platform (CDP) Private Cloud Base from IBM Storage Scale 5.1, SELinux is supported in permissive or enforcing mode on Red Hat ® Enterprise.

For more information, see:

- *Security-Enhanced Linux support* section in the *IBM Storage Scale: Concepts, Planning, and Installation Guide*.
- Cloudera HDP Disable SELinux and PackageKit and check the umask Value documentation.
- Cloudera CDP Private Cloud Base Setting SELinux Mode documentation.

NTP

This topic gives information about Network Time Protocol (NTP).

Configure NTP on all the nodes in your system to ensure that the clocks of all the nodes are synchronized.

On Red Hat Enterprise Linux nodes

```
# yum install -y ntp
# ntpdate <NTP_server_IP>
# systemctl enable ntpd
# systemctl start ntpd
# timedatectl list-timezones
# timedatectl set-timezone
# systemctl enable ntpd
```

Firewall recommendations for HDFS Transparency

Firewalls that are associated with open systems are specific to deployments, operating systems, and it varies from customer to customer. It is the responsibility of the system administrator or Lab Service (LBS) to set the firewall accordingly; similar to what Linux distributions do presently. For information on IBM Storage Scale firewall, see the *IBM Storage Scale system using firewall* section in the *IBM Storage Scale: Administration Guide*.

This section describes only the recommendations for HDFS Transparency firewall settings.

Table 6. Recommended port number settings for HDFS Transparency

HDFS Transparency Property	Port Number	Comments
dfs.namenode.rpc-address	nn-host1: 8020	RPC address that handles all clients requests. In the case of HA/Federation where multiple NameNodes exist, the name service id is added to the name. For example, dfs.namenode.rpc-address.ns1 dfs.namenode.rpc-address.EXAMPLENAMESERVICE . The value of this property will take the form of nn-host1:rpc-port. The NameNode's default RPC port is 8020.
dfs.namenode.http-address	0.0.0.0:9870	The address and the base port where the dfs NameNode web UI will listen on.

Table 6. Recommended port number settings for HDFS Transparency (continued)

HDFS Transparency Property	Port Number	Comments
dfs.datanode.address	0.0.0.0:9866	The DataNode server address and port for data transfer.
dfs.datanode.http.address	0.0.0.0:9864	The DataNode HTTP server address and port.
dfs.datanode.ipc.address	0.0.0.0:9867	The DataNode IPC server address and port.

Setting the firewall policies for HDFS Transparency

1. Run the **firewall-cmd** to add and reload the recommended ports.

   On each of the HDFS Transparency NameNodes, set the NameNode server port.

   The following example uses 8020:

   ```
   # firewall-cmd --add-port=8020/tcp --permanent
   ```

   On each of the HDFS Transparency NameNodes, set the NameNode webui port:

   ```
   # firewall-cmd --add-port=9870/tcp --permanent
   ```

   On each of the HDFS Transparency DataNodes, set the following ports:

   ```
   # firewall-cmd --add-port=9864/tcp --permanent
   # firewall-cmd --add-port=9866/tcp --permanent
   # firewall-cmd --add-port=9867/tcp --permanent
   ```

   For all HDFS Transparency that ran **--add-port**, run reload and check the ports:

   ```
   # firewall-cmd --reload
   # firewall-cmd --zone=public --list-ports
   ```

   For example:

   ```
   [root@c8f2n01 webhdfs]# firewall-cmd --zone=public --list-ports
   1191/tcp 60000-61000/tcp 8020/tcp 9870/tcp 9864/tcp 9866/tcp 9867/tcp
   ```

2. For the changes to reflect, restart HDFS Transparency.

   If HDFS Transparency is running, find the standby NameNode and restart the services followed by a failover.

   a. Get the standby NameNode.

   ```
   # /usr/lpp/mmfs/hadoop/bin/hdfs haadmin -getAllServiceState
   ```

   For example:

   ```
   [root@c8f2n01 webhdfs]# /usr/lpp/mmfs/hadoop/bin/hdfs haadmin -getAllServiceState
   c8f2n01:8020                                       active
   c8f2n05:8020                                       standby
   ```

   b. Restart the Standby NameNode (for example, on c8f2n05).

   For HDFS Transparency 3.1.0 or earlier, run the following command:

   ```
   # /usr/lpp/mmfs/hadoop/sbin/mmhadoopctl connector restart
   ```

   For HDFS Transparency 3.1.1 or later, run the following command:

```
# /usr/lpp/mmfs/bin/mmces service stop HDFS
# /usr/lpp/mmfs/bin/mmces service start HDFS
```

c. Transition standby to active NameNode.

   For example: nn1 is c8f2n01 and nn2 is c8f2n05.

   For HDFS Transparency 3.1.0 and earlier, run the following command:

   ```
   # /usr/lpp/mmfs/hadoop/bin/hdfs haadmin -transitionToActive nn2
   # /usr/lpp/mmfs/hadoop/bin/hdfs haadmin -getAllServiceState
   ```

   For HDFS Transparency 3.1.1 and later, run the following command:

   ```
   # /usr/lpp/mmfs/bin/mmces address move --ces-ip x.x.x.x --ces-node nn2
   # /usr/lpp/mmfs/hadoop/bin/hdfs haadmin -getAllServiceState
   ```

d. The original NameNode is now the standby NameNode.

   Restart the new Standby NameNode (for example, c8f2n01).

   For HDFS Transparency 3.1.0 and earlier, run the following command:

   ```
   # /usr/lpp/mmfs/hadoop/sbin/mmhadoopctl connector restart
   ```

   For HDFS Transparency 3.1.1 and later, run the following command:

   ```
   # /usr/lpp/mmfs/bin/mmces service stop HDFS
   # /usr/lpp/mmfs/bin/mmces service start HDFS
   ```

e. You can now transition back to the original NameNode by running the following command:

   For HDFS Transparency 3.1.0 and earlier, run the following command:

   ```
   # /usr/lpp/mmfs/hadoop/bin/hdfs haadmin -transitionToActive nn1
   # /usr/lpp/mmfs/hadoop/bin/hdfs haadmin -getAllServiceState
   ```

   For HDFS Transparency 3.1.1 and later, run the following command:

   ```
   # /usr/lpp/mmfs/bin/mmces address move --ces-ip x.x.x.x --ces-node nn1
   # /usr/lpp/mmfs/hadoop/bin/hdfs haadmin -getAllServiceState
   ```

3. Restart all Hadoop services on all the nodes.

   For example, on node with Yarn service:

   ```
   /opt/hadoop-3.1.3/sbin/stop-yarn.sh
   /opt/hadoop-3.1.3/sbin/start-yarn.sh
   ```

Hadoop service roles

In a Hadoop ecosystem, there are a lot of different roles for different components. For example, HBase Master Server, Yarn Resource Manager and Yarn Node Manager.

You need to plan to distribute these master roles over different nodes as evenly as possible. If you put all these master roles onto a single node, memory might become an issue.

When running Hadoop over IBM Storage Scale, it is recommended that up to 25% of the physical memory is reserved for GPFS pagepool with a maximum of 20 GB. If HBase is being used, it is recommended that up to 30% of the physical memory be reserved for the GPFS pagepool. If the node has less than 100 GB of physical memory, then the heap size for Hadoop Master services needs to be carefully planned. If HDFS transparency NameNode service and HBase Master service are resident on the same physical node, HBase workload stress may result in Out of Memory (OOM) exceptions.

Dual network interfaces

This section explains about the FPO mode and IBM ESS or SAN-based storage mode.

FPO mode

If the FPO cluster has a dual 10 Gb network, you have the following two configuration options:

- The first option is to bind the two network interfaces and deploy the IBM Storage Scale cluster and the Hadoop cluster over the bonded interface.
- The second option is to configure one network interface for the Hadoop services including the HDFS transparency service and configure the other network interface for IBM Storage Scale to use for data traffic. This configuration can minimize interference between disk I/O and application communication.

  To ensure that the Hadoop applications use data locality for better performance, perform the following steps:

  1. Configure the first network interface with one subnet address (for example, 192.0.2.0). Configure the second network interface as another subnet address (for example, 192.0.2.1).
  2. Create the IBM Storage Scale cluster and NSDs with the IP or hostname from the first network interface.
  3. Install the Hadoop cluster and HDFS transparency services by using IP addresses or hostnames from the first network interface.
  4. Run `mmchconfig subnets=192.0.2.1 -N all`.

     Note: 192.0.2.1 is the subnet used for IBM Storage Scale data traffic.

For Hadoop map/reduce jobs, the scheduler Yarn checks the block location. HDFS Transparency returns the hostname that is used to create the IBM Storage Scale cluster, as block location to Yarn. If the hostname is not found within the NodeManager list, Yarn cannot schedule the tasks according to the data locality. The suggested configuration can ensure that the hostname for block location can be found in Yarn's NodeManager list and therefore it can schedule the task according to the data locality.

For a Hadoop distribution like IBM BigInsights IOP, all Hadoop components are managed by Ambari™. In this scenario, all Hadoop components, HDFS transparency and IBM Storage Scale cluster must be created using one network interface. The second network interface must be used for GPFS data traffic.

Centralized Storage Modes (ESS, ECE, SAN-based)

For Centralized Storage, you have two configuration options:

- The first option is to configure the two adapters as bond adapter and then, deploy HortonWorks HDP and IBM Storage Scale over the bond adapters.
- The second option is to configure one adapter for IBM Storage Scale cluster and HortonWorks HDP and configure another adapter as subnets of IBM Storage Scale. For more information on subnets, see *GPFS and network communication* in the *IBM Storage Scale: Concepts, Planning, and Installation Guide*. Perform the following steps:

  1. Configure the first network interface with one subnet address (for example, 192.0.2.0). Configure the second network interface as another subnet address (for example, 192.0.2.1).
  2. Create the IBM Storage Scale cluster with the IP or hostname from the first network interface.
  3. Install the Hadoop cluster and HDFS transparency services by using the IP addresses or hostnames from the first network interface.
  4. Run `mmchconfig subnets=192.0.2.1 -N all`.

     Note: 192.0.2.1 is the subnet used for IBM Storage Scale data traffic.

Setting up local repository

Mirror repository server

IBM Storage Scale requires a local repository. Therefore, select a server to act as the mirror repository server. This server requires the installation of the Apache HTTP server or a similar HTTP server.

Every node in the Hadoop cluster must be able to access this repository server. This mirror server can be defined in the DNS, or you can add an entry for the mirror server in `/etc/hosts` on each node of the cluster.

- Create an HTTP server on the mirror repository server, such as Apache httpd. If the Apache httpd is not already installed, install it with the **`yum install httpd`** command. You can start the Apache httpd by running one of the following commands:
  - **`apachectl start`**
  - **`service httpd start`**
- [Optional]: Ensure that the http server starts automatically on reboot by running the following command:
  - **`chkconfig httpd on`**
- Ensure that the firewall settings allow inbound HTTP access from the cluster nodes to the mirror web server.
- On the mirror repository server, create a directory for your repositories, such as `<document root>/repos`. For Apache httpd with document root `/var/www/html`, type the following command:
  - **`mkdir -p /var/www/html/repos`**
- Test your local repository by browsing the web directory:
  - **`http://<yum-server>/repos`**

For example:

```
# rpm -qa | grep httpd
# service httpd start
# service httpd status
Active: active (running) □ Check to ensure is active
# systemctl enable httpd
```

Local OS repository

You must create the operating system repository because some of the IBM Storage Scale files, such as rpms have dependencies on all nodes.

1. Create the repository path:

   ```
   mkdir /var/www/html/repos/<rhel_OSlevel>
   ```

2. Synchronize the local directory with the current yum repository:

   ```
   cd /var/www/html/repos/<rhel_OSlevel>
   ```

   Note: Before going to the next step, ensure that you have registered your system. For instructions to register a system, refer to Get Started with Red Hat Subscription Manager. Once the server is subscribed, run the following command: **`subscription-manager repos --enable=<repo_id>`**

3. Run the following command:

   ```
   reposync --gpgcheck -l --repoid=rhel-7-server-rpms --download_path=/var/www/html/repos/
   <rhel_OSlevel>
   ```

4. Create a repository for this node:

   ```
   createrepo -v /var/www/html/repos/<rhel_OSlevel>
   ```

5. Ensure that all the firewalls are disabled or that you have the httpd service port open, because yum uses http to get the packages from the repository.
6. On all nodes in the cluster that require the repositories, create a file in `/etc/yum.repos.d called local_<rhel_OSlevel>.repo`.
7. Copy this file to all nodes. The contents of this file must look like the following:

```
[local_rhel_version]
name=local_rhel_version
enabled=1
baseurl=http://<internal IP that all nodes can reach>/repos/<rhel_OSlevel>
gpgcheck=0
```

8. Run the **`yum repolist`** and **`yum install rpms`** without external connections.

Local IBM Storage Scale repository

This section describes how to configure a local IBM Storage Scale repository for manual installation.

The following table lists the IBM Storage Scale 5.0.5 and later supported editions for the HDFS Transparency clusters:

IBM Storage Scale Edition	**Comments**
Data Management	See the *Capacity-based licensing* topic in the *IBM Storage Scale: Concepts, Planning, and Installation Guide*.
Data Access	See the *Capacity-based licensing* topic in the *IBM Storage Scale: Concepts, Planning, and Installation Guide*.
Advanced Edition	Legacy edition replaced by Data Management edition.
Standard Edition	Legacy edition replaced by Data Access edition.

The following example uses IBM Storage Scale 5.1.2.2:

1. On the repository web server, create a directory for your IBM Storage Scale repos, such as `<document root>/repos/GPFS`. For Apache httpd with document root `/var/www/html`, type the following command:

```
mkdir -p /var/www/html/repos/GPFS/5.1.2.2
```

2. Obtain the IBM Storage Scale software. If you have already installed IBM Storage Scale manually, skip this step. Download the IBM Storage Scale package. In this example, IBM Storage Scale 5.1.2.2 is downloaded from Fix Central, the package is unzipped, and the installer is extracted.

   For example, as root or a user with sudo privileges, run the installer to get the IBM Storage Scale packages into a user-specified directory via the `--dir` option:

```
chmod +x Spectrum_Scale_Data_Management-5.1.2.2-x86_64-Linux-install
./Spectrum_Scale_Data_Management-5.1.2.2-x86_64-Linux-install --silent --dir /var/www/html/
repos/GPFS/5.1.2.2
```

   Note: The `--silent` option is used to accept the software license agreement, and the `--dir` option places the IBM Storage Scale rpms into the directory `/var/www/html/repos/GPFS/5.1.2.2/gpfs_rpms`. Without specifying the `--dir` option, the default location is `/usr/lpp/mmfs/gpfs_rpms/5.1.2.2/gpfs_rpms`.

3. If the packages are extracted into the IBM Storage Scale default directory, `/usr/lpp/mmfs/5.1.2.2/gpfs_rpms`, copy all the IBM Storage Scale files that are required for your installation environment into the IBM Storage Scale repository path:

```
cd /usr/lpp/mmfs/5.1.2.2/gpfs_rpms

cp -R * /var/www/html/repos/GPFS/5.1.2.2/gpfs_rpms
```

4. Copy the HDFS Transparency package to the IBM Storage Scale repo path that you want to install manually.

   Note: The repo must contain only one HDFS Transparency package. Remove all old transparency packages.

   ```
   cp gpfs.hdfs-protocol-3.1.1-(version)  /var/www/html/repos/GPFS/5.1.2.2/gpfs_rpms
   ```

5. Create a yum repository:

   ```
   # cd /var/www/html/repos/GPFS/5.1.2.2/gpfs_rpms
   # createrepo .
   ```

6. Access the repository at http://<yum-server>/repos/GPFS/5.1.2.2/gpfs_rpms.

Hadoop distribution support

Cloudera distributions and Open Source Apache Hadoop are the officially supported Hadoop distributions.

For more information, contact scale@us.ibm.com.

OS and Arch support

HDFS Transparency supports a subset of the supported operating systems and the architecture platform that IBM Storage Scale supports.

IBM Storage Scale aligns with the OS vendor life cycle support statement. For RHEL, see Red Hat Enterprise Linux Life Cycle.

Java support

HDFS Transparency requires Java™ OpenJDK 8 or OpenJDK 11.

OpenJDK 11 is supported from HDFS Transparency 3.1.1-8.

CDP Private Cloud Base support

Table 7. CDP Private Cloud Base support

CDP Private Cloud Base version	HDFS Transparency version
See "Support Matrix" on page 294	3.1.1-X stream, 3.1.1-2 and later

Open Source Apache Hadoop support

- Open Source Apache Hadoop support is based on Cloudera's Hadoop supported versions. For more information, see CDP Private Cloud Base support matrix and HDP support matrix.
- IBM Storage Scale 5.1.3.2 technical preview release, the CES HDFS Transparency 3.2.2 is supported for a limited-time usage only on non-production clusters with Open Source Apache Hadoop 3.2.2 on RH 7.9 on x86_64.
- From IBM Storage Scale 5.1.4.0, CES HDFS Transparency 3.2.2-0 is supported for Open Source Apache Hadoop 3.2.2 on RH 7.9 on x86_64.
- From IBM Storage Scale 5.1.1.2, CES HDFS Transparency 3.3.x-x is supported for Open Source Apache Hadoop 3.3 on RH 7.9 on x86_64.

BigInsights IOP support

Support for IBM BigInsights is discontinued.

HDP support

Support for Cloudera HDP is discontinued.

Table 8. HDP support

HDP version	HDFS Transparency version
HDP 3.1	3.1.0-X stream

HDFS Transparency planning

The recommended configuration is to configure CES HDFS (NameNodes and DataNodes) as IBM Storage Scale client nodes remote mount to the centralized storage.

In each of the following architecture figures, these remote mounts are represented by the ESS blue boxes. For information about the centralized storage mode, see "Hadoop IBM Storage Scale Architecture" on page 4.

Note: File Placement Optimizer (FPO) is not a supported storage for the CES HDFS configuration.

The Hadoop master and clients are outside of the IBM Storage Scale cluster but the IBM Storage Scale NameNodes and DataNodes are part of the Hadoop cluster; in the following architecture diagrams, a Hadoop cluster is represented by a Hadoop cluster green box. The installer node does not need to be a part of the IBM Storage Scale cluster.

To add other protocols like SMB, NFS or OBJ to the cluster, ensure that the other protocol requirements are met. Before you add these protocols, see CES HDFS Limitations and recommendations.

For more information, see the following topics in the *IBM Storage Scale: Concepts, Planning, and Installation Guide*:

- *Planning for GPFS*
- *Preparing to use the installation toolkit*
- *Planning for Protocols*

Note:

- If you are using Cloudera CDP distribution, see "Support Matrix" on page 294, "Architecture" on page 288 and "Alternative architectures" on page 291.
- CES HDFS is not supported for Cloudera HDP distribution does not support CES HDFS.
- The NameNode cannot be colocated with the DataNode or with any other Hadoop services.

The following figures show the different architecture configuration layouts:

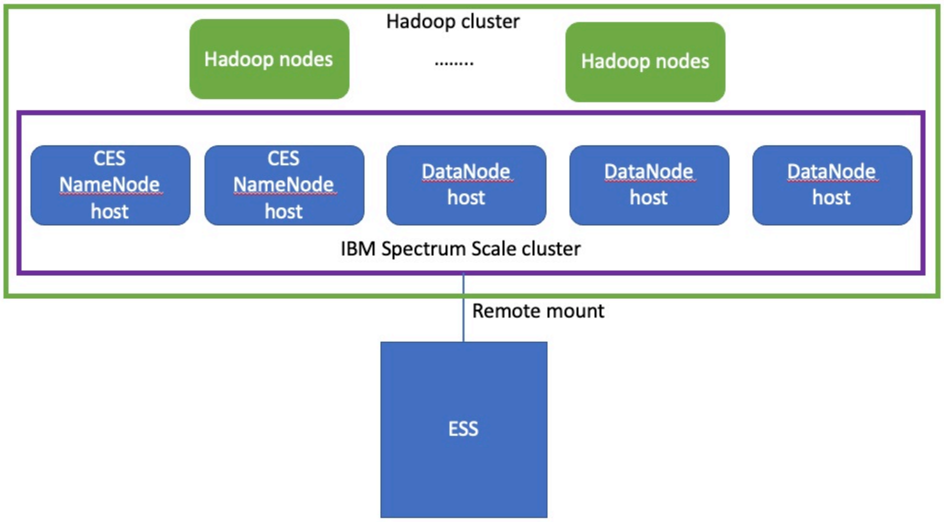


Figure 11. CES HDFS single HDFS configuration

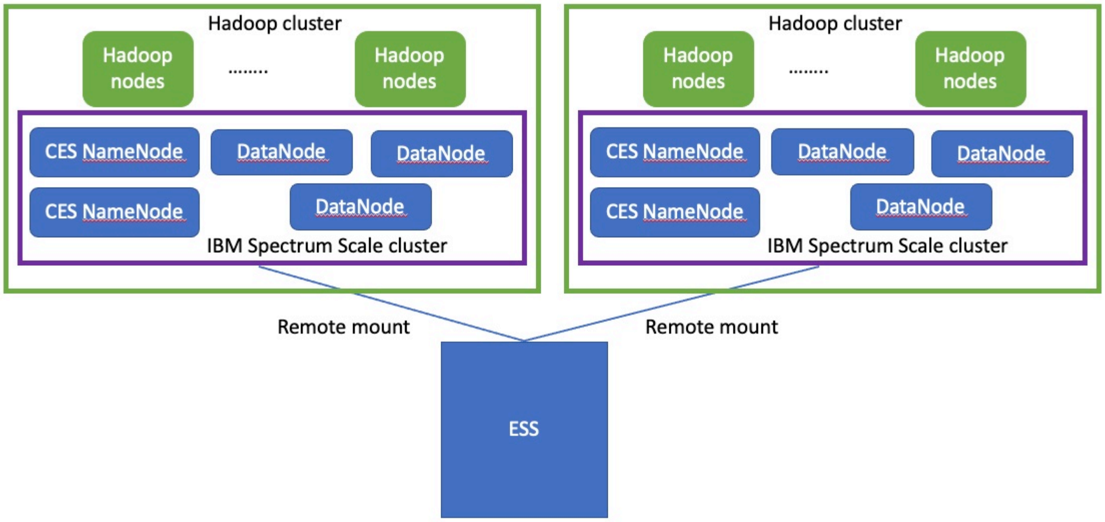


Figure 12. CES HDFS multiple HDFS configuration

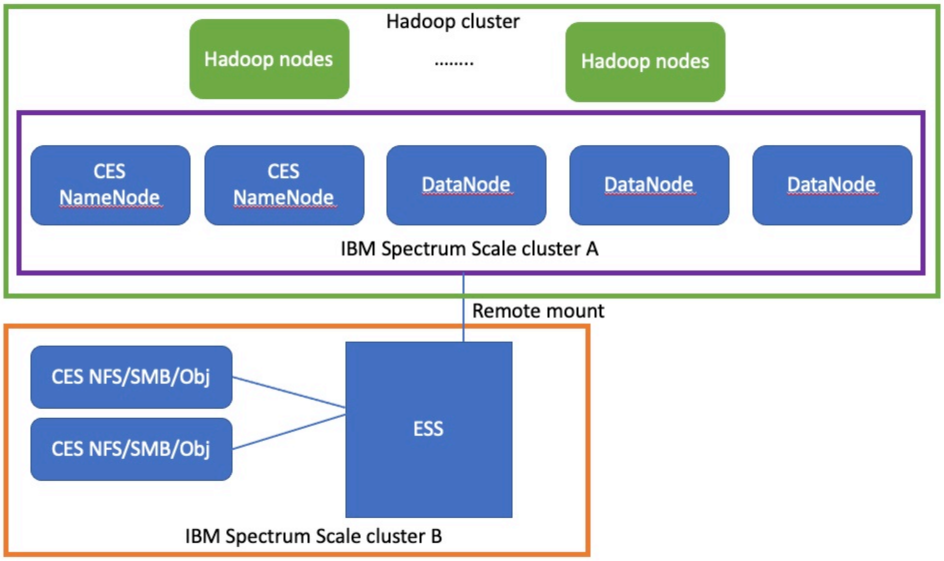


Figure 13. CES HDFS with other protocols configurations layout to the ESS

HDFS Transparency support matrix

The support matrix for HDFS Transparency and the IBM Storage Scale Big Data Analytics Integration Toolkit for HDFS Transparency (Toolkit for HDFS) is bundled together to work for the supported IBM Storage Scale release.

Table 9. HDFS Transparency support matrix

IBM Storage Scale version	**HDFS Transparency version**		
	3.1.1-x stream	**3.2.2-x stream**	**3.3.0-x stream**
5.1.9.2	3.1.1-17	3.2.2-7	N/A
5.1.9.1	3.1.1-16	3.2.2-7	N/A
5.1.9.0	3.1.1-15	3.2.2-6	N/A
5.1.8.1	3.1.1-14	3.2.2-5	3.3.0-2
5.1.7.1 - 5.1.8.0	3.1.1-13	3.2.2-5	3.3.0-2
5.1.7	3.1.1-12	3.2.2-4	3.3.0-2
5.1.6.1	3.1.1-12	3.2.2-3	3.3.0-2
5.1.6	3.1.1-11	3.2.2-3	3.3.0-2
5.1.5.1	3.1.1-10	3.2.2-2	3.3.0-2
5.1.5	3.1.1-10	3.2.2-1	3.3.0-2
5.1.4.1	3.1.1-10	3.2.2-1	3.3.0-2
5.1.4	3.1.1-9	3.2.2-1	3.3.0-1
5.1.3.2	3.1.1-8	3.2.2-0	3.3.0-1
5.1.3 - 5.1.3.1	3.1.1-8	-	3.3.0-1
5.1.2.9	3.1.1-12	-	3.3.0-2
5.1.2.6 - 5.1.2.8	3.1.1-10	-	3.3.0-2
5.1.2.2 - 5.1.2.5	3.1.1-8	-	3.3.0-1
5.1.2.1	3.1.1-7	-	3.3.0-0
5.1.2	3.1.1-6	-	3.3.0-0
5.1.1.2 - 5.1.1.4	3.1.1-5	-	3.3.0-0
5.1.1.1	3.1.1-5	-	-
5.1.1.0	3.1.1-4	-	-
5.1.0.1 - 5.1.0.3	3.1.1-3	-	-

Open-source Apache Hadoop support is certified on HDFS, Yarn, and MapReduce components with the following configurations:

Table 10. Open-source Apache Hadoop support matrix

Open-source Apache Hadoop version	**HDFS Transparency version**	**OS**	**Platform**
3.1.3	3.1.1-0 - 3.1.1-14	RHEL 7.9 and later	x86_64 ppc64le

Table 10. Open-source Apache Hadoop support matrix (continued)

Open-source Apache Hadoop version	HDFS Transparency version	OS	Platform
3.2.2	3.2.2-x	RHEL 7.9 RHEL9.x	x86_64
3.3.0	3.3.0-x	RHEL 7.9	x86_64

For more information about CDP Private Cloud Base support, see CDP Private cloud base support matrix.

Note:

- CES HDFS is not supported on Cloudera HDP distribution.
- Unlike previous versions of HDFS Transparency, HDFS Transparency 3.1.1-x, 3.2.2-x, and 3.3.0-x are tightly coupled with IBM Storage Scale. You need to upgrade the IBM Storage Scale package to get the correct supported versions for CES HDFS.
- Support for CES HDFS started from IBM Storage Scale 5.0.4.2 with HDFS Transparency 3.1.1-0 and Toolkit for HDFS 1.0.0.0.
- Support for CDP Private Cloud Base with CES HDFS started from IBM Storage Scale 5.1.0 with HDFS Transparency 3.1.1-2 package with bda_integration toolkit version 1.0.2.0.
- If the OS is not supported for a specific IBM Storage Scale release, then it is also not supported for HDFS Transparency. For more information, see "OS and Arch support" on page 24.
- Unlike previous versions of HDFS Transparency 3.1.1-x, HDFS Transparency 3.1.1-15+ is delivered without dependent JAR files. For more information about the installation process, see "Installation prerequisites" on page 30.
- Unlike previous versions of HDFS Transparency 3.2.2-x, HDFS Transparency 3.2.2-6+ is delivered without dependent JAR files. For more information about the installation process, see "Installation prerequisites" on page 30.

HDFS Transparency download

The download source and the contents of an installation package vary depending on the HDFS Transparency version.

1. Visit IBM Fix Central to download the HDFS Transparency package.
2. For HDFS Transparency 3.1.0 and earlier:

   a. Search for *Spectrum_Scale_HDFS_Transparency-<version>-<arch>-Linux* to find the correct package.

   b. Untar the downloaded package:

   ```
   tar zxvf Spectrum_Scale_HDFS_Transparency-<version>-<arch>-Linux.tgz
   ```

   For HDFS Transparency version 3.1.1 or later, the HDFS Transparency package is available through the IBM Storage Scale software. The IBM Storage Scale software is delivered in a self-extracting archive. This self-extracting archive can be downloaded from the Fix Central. For more information, see the *Extracting the IBM Storage Scale software on Linux nodes* topic in the *IBM Storage Scale: Concepts, Planning, and Installation Guide*.

   For example, `Spectrum_Scale_Advanced-5.1.0.0-x86_64-Linux` is the fix pack in the Fix Central and `Spectrum_Scale_Advanced-5.1.0.0-x86_64-Linux-install` is the self-extracting archive package.

For IBM Storage Scale 5.1.0.0 on RHEL7, the self-extracting installation package places the packages in the following default directory:

```
/usr/lpp/mmfs/5.1.0.0/hdfs_rpms/rhel7/hdfs_3.1.1.x
```

Packages information

- For HDFS Transparency 3.1.0 stream:
  - IBM Storage Scale HDFS Transparency

    For example, `gpfs.hdfs-protocol-3.1.0-5.x86_64.rpm`.
- For HDFS Transparency 3.1.1-0 and 3.1.1-1:
  - IBM Storage Scale Big Data Analytics Integration Toolkit for HDFS Transparency (Toolkit for HDFS)

    For example, `bda_integration-1.0.1-1.noarch.rpm`.
  - IBM Storage Scale HDFS Transparency

    For example, `gpfs.hdfs-protocol-3.1.1-1.x86_64.rpm`.
- For HDFS Transparency 3.1.1-2 and later:
  - IBM Storage Scale Big Data Analytics Integration Toolkit for HDFS Transparency (Toolkit for HDFS)

    For example, `gpfs.bda-integration-1.0.2-0.noarch.rpm`.
  - IBM Storage Scale HDFS Transparency

    For example, `gpfs.hdfs-protocol-3.1.1-2.x86_64.rpm`.

Note:

- From IBM Storage Scale 5.1.0, the BDA Toolkit for HDFS is named `gpfs.bda-integration`. In IBM Storage Scale 5.0.4 and 5.0.5, it was named `bda_integration`.
- In Fix Central, the fix pack name for HDFS Transparency version syntax is "x.x.x.x".

  For example, `Spectrum_Scale_HDFS_Transparency-3.1.0.5-x86_64-Linux`.
- The installation toolkit cannot be used if the HDFS Transparency package is not a part of the IBM Storage Scale software self-extracting archive package because of the signed repo checking (for example, as a patch/efix package). Therefore, you must use the manual installation method to install or upgrade.

To verify whether the HDFS Transparency can be used in your environment, see the following sections:

- The HDFS Transparency "Hardware & OS matrix support" on page 16
- HDFS Transparency matrix support
- "Hadoop distribution support" on page 24

Installing

This section will describe the steps to install the HDFS Transparency nodes (NameNode and DataNodes) as GPFS client nodes to be added to the centralized storage system to create a single GPFS cluster. All other Hadoop nodes (master and clients) are to be set up outside of the GPFS cluster.

Before you proceed, see the following sections:

- "HDFS Transparency planning" on page 25
- "HDFS Transparency support matrix" on page 27
- "HDFS Transparency limitations and recommendations" on page 250
- "Installation prerequisites" on page 30

Note: For Cloudera® HDP distribution, CES HDFS is not supported.

Note:

- The centralized storage file system needs to be available before setting up the CES HDFS protocol nodes.
- Required to create the CES shared root (`cesSharedRoot`) file system.
- Do not follow steps that deploy NSDs on the HDFS Transparency nodes because centralized storage mode is the only one supported currently.
- FPO is not supported.
- HDFS Transparency does not require to have the Hadoop distribution installed onto the IBM Storage Scale HDFS Transparency nodes. However, if the HDFS client is not installed on the CES HDFS NameNodes and DataNodes, then functions like distcp will not work because HDFS Transparency does not include the **`bin/hadoop`** command.
- When adding HDFS protocol into CES, the other protocols (NFS, SMB, Object) and GUI and performance monitor can be configured and deployed at the same time.
- SMB requires `NFSv4 ACL` permission while HDFS requires `ALL ACL` permission. Therefore, a warning will be seen if HDFS protocol is added to the protocol node and the ACL is not correct after the install toolkit deployment. The ACL should always be set to `ALL` if the HDFS protocol is used after deployment of the protocols.

Installation prerequisites

Set up the basic IBM Storage Scale installation prerequisites before installing CES HDFS.

See the *Installation prerequisites* section in the *IBM Storage Scale: Concepts, Planning, and Installation Guide* for base Scale installation requirements.

- NTP setup

  It is recommended that Network Time Protocol (NTP) must be configured on all the nodes in your system to ensure that the clocks of all the nodes are synchronized. Clocks that are not synchronized cause debugging issues and authentication problems with the protocols. Across all the HDFS Transparency and Hadoop nodes, follow the steps that are listed in "Configure NTP to synchronize the clock in HDFS Transparency" on page 56.

- SSH and network setup

  Set up passwordless SSH as follows:

  - From the admin node to the other nodes in the cluster.
  - From protocol nodes to other nodes in the cluster.
  - From every protocol node to the rest of the protocol nodes in the cluster.
  - On fresh Red Hat Enterprise Linux 8 installations, you must create passwordless SSH keys by using the **`ssh-keygen -m PEM`** command.

- CES public IP

  - A set of CES public IPs (or Export IPs) is required. These IPs are used to export data using the protocols. Export IPs are shared among all protocols and are organized in a public IP pool. See *Adding export IPs* section under *Deploying protocols* topic in the *IBM Storage Scale: Concepts, Planning, and Installation Guide*.
  - If you are using only the HDFS protocol, it is sufficient to have just one CES Public IP.
  - The CES IP/hostname used for CES HDFS must be resolved by the DNS service and not just by an entry in your `/etc/hosts` file. Otherwise, you might encounter errors when you add the Hadoop services.

    Note: This is a Java requirement.

- ACL

  In general, the recommendation is to configure the file system to support NFSv4 ACLs. NFSv4 ACL is a requirement for ACL usage with the SMB and NFS protocols. However, ALL ACL is requirement for

ACL usage with HDFS protocols. If the protocol node has multiple protocols, the final ACL setting after deployment should be set to `-k ALL` if you are using HDFS protocol.

For more information, see examples under the **mmchfs** command topic in the *IBM Storage Scale: Command and Programming Reference Guide*.

- Packages

  Corresponding kernel-header, kernel-devel, gcc, cpp, gcc-c++, instils, make must be installed.

  `yum install kernel-devel cpp gcc gcc-c++ binutils make`

  Note: If you are using CDP Private Cloud Base, you need to install Python 2.7 on Red Hat Enterprise Linux 8.0 nodes. By default, Python 3 might be installed on Red Hat Enterprise Linux 8.0 nodes. CDP Private Cloud Base with CES HDFS requires the nodes to have both Python 2.7 and Python 3.

- UID/GID consistency value under IBM Storage Scale

  Ensure that all the user IDs and group IDs used in the IBM Storage Scale cluster for running jobs, accessing the IBM Storage Scale file system or for the Hadoop services must be created and have the same values across all the IBM Storage Scale nodes. This is required for IBM Storage Scale.

  You can also use the `/usr/lpp/mmfs/hadoop/scripts/gpfs_create_hadoop_users_dirs.py` script that is provided with HDFS Transparency 3.1.1-3 and later. Any users or groups that are created with this script are guaranteed to have consistent UID/GID across all the nodes.

- Starting with HDFS Transparency 3.1.1-15 and HDFS Transparency 3.2.2-6, the dependent JAR files are not shipped with the HDFS Transparency `rpm`. The dependent JAR files need to be provided before an installation or upgrade.

  For 3.1.1-x on all HDFS Transparency nodes, complete the following steps:

  1. If it does not exist, create the path `/opt/hadoop/jars` by using the command:

     ```
     $ mkdir -p /opt/hadoop/jars
     ```

  2. Download `hadoop-3.1.4.tar.gz` from Apache by issuing the following commands:

     ```
     $ cd /opt/hadoop/jars
     $ wget https://archive.apache.org/dist/hadoop/core/hadoop-3.1.4/hadoop-3.1.4.tar.gz
     ```

  3. Extract the content of the `tar` files by using this command:

     ```
     $ tar -xvf hadoop-3.1.4.tar.gz
     ```

  4. Download additional JAR files from the maven repository and save them in `/opt/hadoop/jars`.

     The additional JAR files that are needed are:

     – `curator-client-2.12.0.jar`
     – `curator-framework-2.12.0.jar`
     – `curator-recipes-2.12.0.jar`
     – `guava-11.0.2.jar`
     – `hadoop-annotations-3.1.1.jar`
     – `hadoop-auth-3.1.1.jar`
     – `jsch-0.1.54.jar`
     – `jsr305-3.0.0.jar`
     – `xz-1.0.jar`

     Alternatively, download `hadoop-3.1.1.tar.gz` from Apache and extract it in `/opt/hadoop/jars`.

  5. Proceed with the installation or upgrade.

  For 3.2.2-x on all HDFS Transparency nodes, complete the following steps:

1. If it does not exist, create the path `/opt/hadoop/jars` by using the command:

   ```
   $ mkdir -p /opt/hadoop/jars
   ```

2. Download `hadoop-3.2.4.tar.gz` from Apache by issuing the following commands:

   ```
   $ cd /opt/hadoop/jars
   $ wget https://archive.apache.org/dist/hadoop/core/hadoop-3.2.4/hadoop-3.2.4.tar.gz
   ```

3. Extract the content of the `tar` files by using this command:

   ```
   $ tar -xvf hadoop-3.2.4.tar.gz
   ```

4. Download additional JAR files from the maven repository and save them in `/opt/hadoop/jars`.

   The additional JAR files that are needed are:

   - `accessors-smart-1.2.jar`
   - `hadoop-annotations-3.2.2.jar`
   - `hadoop-auth-3.2.2.jar`
   - `jetty-xml-9.4.20.v20190813.jar`
   - `jul-to-slf4j-1.7.25.jar`
   - `log4j-1.2.17.jar`
   - `slf4j-api-1.7.25.jar`
   - `slf4j-log4j12-1.7.25.jar`
   - `stax2-api-3.1.4.jar`

   Alternatively, download `hadoop-3.2.2.tar.gz` from Apache and extract it in `/opt/hadoop/jars`.

5. Proceed with the installation or upgrade.

The following sections are steps for installation with snips from the IBM Storage Scale installation documentation:

- If you are planning to use the installation toolkit, follow the "Steps for install toolkit" on page 32 section.
- If you are planning to install manually, follow the "Steps for manual installation" on page 33 section.

Steps for install toolkit

This section lists the steps for the installation of the toolkit.

Note: Ensure that the steps in the "Installation prerequisites" on page 30 section are completed before you proceed with the steps listed in this section.

1. Install the following packages for the installation toolkit:
   - python-2.7
   - net-tools
   - elfutils-libelf-devel [Only on Red Hat Enterprise Linux 8.0 nodes with kernel version 4.15 or later]
2. Install the JAVA openjdk-devel on all the nodes

   ```
   yum install java-1.8.0-openjdk-devel
   ```

3. Export Java home in root profile

   ```
   # vi ~/.bashrc
   export JAVA_HOME=/usr/lib/jvm/java-1.8.0-openjdk
   export PATH=$PATH:$JAVA_HOME/bin
   ```

4. Obtain and run the IBM Storage Scale self-extracting installation package.

Run the self-extracting installation package:

```
# ./Spectrum_Scale_Advanced-5.1.0.0-x86_64-Linux-install
```

After the IBM Storage Scale package is expanded, the hdfs_3.1.1-x folder will contain two packages required for the installation toolkit usage that will reside in the `/usr/lpp/mmfs/5.1.0.0/hdfs_rpms/rhel7` directory.

The HDFS Transparency and the IBM Storage Scale Big Data Analytics Integration Toolkit for HDFS Transparency (Toolkit for HDFS) are the two packages that are required for the installation toolkit usage based on the "HDFS Transparency support matrix" on page 27 versioning and should reside in the `hdfs_3.1.1-x` directory.

Installation Toolkit supports the deployment of the following versions of HDFS Transparency:

a. From IBM Storage Scale 5.1.1.2 through 5.1.3.1:

- HDFS Transparency 3.1.1-x
- HDFS Transparency 3.3.x-x

b. From IBM Storage Scale 5.1.3.2 (Technical preview release)/5.1.4:

- HDFS Transparency 3.1.1-x
- HDFS Transparency 3.2.2-x
- HDFS Transparency 3.3.x-x

By default, HDFS Transparency 3.1.1-x is deployed.

If you want to deploy HDFS Transparency 3.2.2-x, set the following environment variable before running the installation toolkit command:

```
# export SCALE_HDFS_TRANSPARENCY_VERSION_322_ENABLE=True
```

If you want to deploy HDFS Transparency 3.3.x-x, you need to set the following environment variable before running the installation toolkit command:

```
# export SCALE_HDFS_TRANSPARENCY_VERSION_33_ENABLE=True
```

You can set the *SCALE_HDFS_TRANSPARENCY_VERSION_<version>_ENABLE* variable in `~/.bashrc`. Here, *<version>* is 322 or 33 without any "." between the numbers.

For more information on the IBM Storage Scale software package, see *Extracting the IBM Storage Scale software on Linux nodes* in *IBM Storage Scale: Concepts, Planning, and Installation Guide* and "HDFS Transparency download" on page 28 section.

5. Install the required packages for Ansible toolkit deployment.

   From IBM Storage Scale 5.1.1, the IBM Storage Scale installation toolkit uses the Ansible deployment. For Red Hat 7 and 8, the installation toolkit installs the supported version of Ansible on the installer node when you run the **`./spectrumscale setup -s InstallNodeIP`** command.

   To manually install the correct Ansible for your environment, see the *Preparing to use the installation toolkit* topic in the *IBM Storage Scale: Concepts, Planning, and Installation Guide*.

6. After the setup is complete, see "Installing" on page 29 followed by "Using installation toolkit" on page 34.

Steps for manual installation

This section lists the steps for the manual installation of the toolkit.

Note: Ensure that the steps in the "Installation prerequisites" on page 30 section are completed before you proceed with the steps listed in this section.

1. On the nodes designated for CES HDFS, to extract the software, follow the steps in the *Preparing to install the GPFS software on Linux nodes* topic in the *IBM Storage Scale: Concepts, Planning, and Installation Guide*.
2. To install the packages, follow the steps listed in the *Installing IBM Storage Scale packages on Linux systems* topic in the *IBM Storage Scale: Concepts, Planning, and Installation Guide*.
3. After the GPFS packages are installed, run `/usr/lpp/mmfs/bin/mmbuildgpl` to build the portability layer on each node.
4. Install the JAVA openjdk-devel on all the nodes by executing the following command:

   ```
   yum install java-1.8.0-openjdk-devel
   ```
5. Export Java home in root profile:

   ```
   # vi ~/.bashrc
   export JAVA_HOME=/usr/lib/jvm/java-1.8.0-openjdk
   export PATH=$PATH:$JAVA_HOME/bin
   ```
6. After the setup is complete, see "Installing" on page 29, followed by "Manual installation" on page 42 to install and configure CES HDFS.

Using installation toolkit

This section describes how to install CES HDFS using the installation toolkit.

Run these steps after the Setup "Steps for install toolkit" on page 32 are completed.

The installation toolkit requires two packages to perform the installation for CES HDFS:

- HDFS Transparency
- IBM Storage Scale Big Data Analytics Integration Toolkit for HDFS Transparency (Toolkit for HDFS)

The installation toolkit must be run as the root user.

On the installer node, where the self-extracting IBM Storage Scale package resides, the installation toolkit default extraction path starting from IBM Storage Scale 5.1.1.x is `/usr/lpp/mmfs/package_code_version/ansible-toolkit`. For IBM Storage Scale version earlier than 5.1.1, the default extraction path is `/usr/lpp/mmfs/package_code_version/installer`.

There are two modes to setup CES HDFS nodes in the centralized file system:

- Adding CES HDFS nodes into the same GPFS cluster as the centralized file system.
- If the CES HDFS nodes are separate GPFS cluster from the centralized file system you need to first setup the remote mount configuration. For more information, see the *Mounting a remote GPFS file system* topic in the *IBM Storage Scale: Administration Guide*.

Adding CES HDFS nodes into the centralized file system

This topic lists the steps to add the CES HDFS nodes into the same GPFS cluster as the centralized file system.

1. Ensure that the centralized file system is already installed, configured and active. For example, the ESS.
2. Create the CES shared root file system which will be used by CES installation.

   Note: The recommendation for CES shared root is a dedicated file system. A dedicated file system can be created with the **mmcrfs** command. The CES shared root must reside on GPFS and must be available when it is configured through **mmchconfig** command.

   For more information, see the *Setting up Cluster Export Services shared root file system* topic in *IBM Storage Scale: Administration Guide*.
3. Change to the installer directory to run the **spectrumscale** commands:

For IBM Storage Scale 5.1.1 and later:

```
# cd /usr/lpp/mmfs/5.1.1.0/ansible-toolkit
```

For IBM Storage Scale 5.1.0 and earlier:

```
# cd /usr/lpp/mmfs/5.0.4.2/installer
```

4. Instantiate the installer node (chef zero server)

   To configure the installer node, issue the following command:

   ```
   ./spectrumscale setup -s InstallNodeIP -i SSHIdentity
   ```

   The -s argument identifies the IP that the nodes will use to retrieve their configuration. This IP will be the one associated with a device on the installer node. This is automatically validated during the setup phase.

   Optionally, you can specify a private SSH key to be used to communicate with the nodes in the cluster definition file, using the -i argument.

   In an Elastic Storage Server (ESS) cluster, if you want to use the installation toolkit to install GPFS and deploy protocols, you must specify the setup type as ess while setting up the installer node:

   ```
   ./spectrumscale setup -s InstallNodeIP -i SSHIdentity -st ess
   ```

5. Use the installation toolkit to populate the cluster definition file from the centralized storage.

   Re-populate the cluster definition file with the current cluster state by issuing the **`./spectrumscale config populate --node Node`** command.

   In a cluster containing ESS, you must specify the EMS node with the `config populate` command.

   For example:

   ```
   ./spectrumscale config populate --node EMSNode
   ```

6. Add the nodes that will be used for CES HDFS into the existing centralized file system. The additional nodes are added into the same GPFS cluster.

   ```
   ./spectrumscale node add FQDN
   ```

   Deployment of protocol services is performed on a subset of the cluster nodes that have been designated as protocol nodes using the **`./spectrumscale node add FQDN -p`** command.

   NameNodes are protocol nodes and requires the -p option during the node add operation.

   DataNodes are not protocol nodes.

   For example:

   For non-HA

   ```
   # NameNodes (Protocol node)
   ./spectrumscale node add c902f05x05.gpfs.net -p
   ```

   For HA

   ```
   # NameNodes (Protocol node)
   ./spectrumscale node add c902f05x05.gpfs.net -p
   ./spectrumscale node add c902f05x06.gpfs.net -p

   # DataNodes
   ./spectrumscale node add c902f05x07.gpfs.net
   ./spectrumscale node add c902f05x08.gpfs.net
   ./spectrumscale node add c902f05x09.gpfs.net
   ./spectrumscale node add c902f05x10.gpfs.net
   ```

7. If call home is enabled in the cluster definition file, specify the minimum call home configuration parameters.

   ```
   ./spectrumscale callhome config -n CustName -i CustID -e CustEmail -cn CustCountry
   ```

   For more information, see the *Enabling and configuring call home using the installation toolkit* topic in the *IBM Storage Scale: Concepts, Planning, and Installation Guide*.
8. Do environment checks before initiating the installation procedure.

   ```
   ./spectrumscale install -pr
   ```
9. Start the IBM Storage Scale installation and add the nodes into the existing cluster.

   ```
   ./spectrumscale install
   ```

Enable and deploy CES HDFS

Before you deploy the protocols, there must be a GPFS cluster that has GPFS started with at least one file system for the CES shared root file system. Protocol nodes requires at least two GPFS file systems to be mounted: one for CES shared root and one for data.

1. Enable HDFS.

   ```
   ./spectrumscale enable hdfs
   ```
2. Set the CES IPs.

   Data is served through these protocols from a pool of addresses designated as Export IP addresses or CES public IP addresses. This example uses 192.0.2.2 and 192.0.2.3.

   ```
   ./spectrumscale config protocols -e 192.0.2.2, 192.0.2.3
   ```

   Note: For IBM Storage Scale releases earlier to 5.0.5.1, a minimum of two CES IPs are required as input for configuring protocol when HDFS is enabled through the installation toolkit even though the HDFS protocol requires only one IP address.

   From IBM Storage Scale 5.0.5.1, only one CES-IP is needed for one HDFS cluster during installation toolkit deployment.
3. Configure the shared root directory.

   Get the CES shared root file system that was created from the step in "Adding CES HDFS nodes into the centralized file system" on page 34 and configure the protocols to point to a file system that will be used as the shared root using the following command:

   ```
   ./spectrumscale config protocols -f FS_Name -m FS_Mountpoint
   ```

   For example:

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

   For more information, see the *Defining a shared file system for protocols* section in *IBM Storage Scale: Concepts, Planning, and Installation Guide*.
4. Create the NameNodes and DataNodes for a new CES HDFS cluster.

   ```
   ./spectrumscale config hdfs new -n NAME -nn NAMENODES -dn DATANODES -f FILESYSTEM -d DATADIR
   ```

   The `-f` is the **`gpfs.mnt.dir`** value and `-d DATADIR` is the **`gpfs.data.dir`** value as seen in the HDFS Transparency configuration files. Therefore, each new HDFS Transparency cluster requires its own `-d DATADIR` value.

   For example:

For non-HA

```
# ./spectrumscale config hdfs new -n myhdfscluster -nn c902f05x05 -dn
c902f05x07,c902f05x08,c902f05x09,c902f05x10 -f gpfs -d gpfshdfs
```

For HA

```
# ./spectrumscale config hdfs new -n myhdfscluster -nn c902f05x05,c902f05x06 -dn
c902f05x07,c902f05x08,c902f05x09,c902f05x10 -f gpfs -d gpfshdfs
```

Where

```

  -n NAME,       --name NAME                          HDFS cluster name.
  -nn NAMENODES, --namenodes      NAMENODES           NameNode hostnames (comma separated).
  -dn DATANODES, --datanodes      DATANODES           DataNode hostnames (comma separated).
  -f FILESYSTEM, --filesystem     FILESYSTEM          Spectrum Scale file system name.
  -d DATADIR,    --datadir        DATADIR             Spectrum Scale data directory name.
```

Note: The *-n NAME* is the HDFS cluster name. The CES group contains the HDFS cluster name prefix with *hdfs*.

The *-d DATADIR* is a unique 32-character name required for each HDFS cluster to be created on the same centralized storage.

To configure multiple HDFS clusters, see "Adding a new HDFS cluster into existing HDFS cluster on the same GPFS cluster (Multiple HDFS clusters)" on page 73 section.

5. List the configured HDFS cluster by running the following command:

```
./spectrumscale config hdfs list
```

For example:

Single HDFS cluster list:

```
 Cluster Name  : mycluster
 NameNodesList : [c902f09x11kvm1],[c902f09x11kvm2]
 DataNodesList : [c902f09x11kvm3],[c902f09x11kvm4]
 FileSystem    : gpfs1
 DataDir       : datadir1
```

Multi-HDFS cluster list:

```

 Cluster Name  : mycluster1
 NameNodesList : [c902f09x11kvm1],[c902f09x11kvm2]
 DataNodesList : [c902f09x11kvm3],[c902f09x11kvm4]
 FileSystem    : gpfs1
 DataDir       : datadir1

 Cluster Name  : mycluster2
 NameNodesList : [c902f09x11kvm5],[c902f09x11kvm6]
 DataNodesList : [c902f09x11kvm7],[c902f09x11kvm8]
 FileSystem    : gpfs1
 DataDir       : datadir2
```

Note: Multi-HDFS cluster is not supported in IBM Storage Scale Big Data Analytics Integration Toolkit for HDFS Transparency (Toolkit for HDFS) version 1.0.3.0 under IBM Storage Scale 5.1.1.0.

6. Do environment checks before initiating the installation procedure.

```
./spectrumscale deploy --pr
```

7. Start the IBM Storage Scale installation and the creation of the CES HDFS nodes.

```
./spectrumscale deploy
```

8. Verify CES HDFS service after deployment is completed.

```
/usr/lpp/mmfs/bin/mmces service list -a
```

9. Check whether the CES HDFS protocol IPs values are configured properly.

   ```
   /usr/lpp/mmfs/bin/mmces address list
   ```

   For more information, see Listing CES HDFS IPs.

10. After the CES HDFS nodes are installed, create the HDFS client nodes manually. For more information, see Chapter 6, “Apache Hadoop,” on page 489.

    For information on the **spectrumscale**, **mmces**, and **mmhdfs** commands, see the *IBM Storage Scale: Command and Programming Reference Guide*.

    Note: If HDFS Transparency is a part of the protocols used in the cluster, ensure that the ACL for GPFS file system is set to -k ALL after all the protocols are installed.

    mmlsfs to check the -k value.

    mmchfs to change the -k value.

    Restart all services and IBM Storage Scale to pick up the -k changes.

Separate CES HDFS cluster remote mount into the centralized file system

This topic lists the steps to create the CES HDFS nodes in a separate GPFS cluster from the centralized file system. It is mandatory to setup the remote mount configuration between the CES HDFS GPFS cluster and the centralized file system GPFS cluster before you can deploy the CES HDFS configuration through the install toolkit.

Use the install toolkit to create a scale cluster for nodes that will be designated as CES HDFS NameNodes and DataNodes. This will be the local GPFS cluster and will be the accessing cluster. The local GPFS cluster requires to create NSDs to be used for the CES Shared root file system.

Preparing installer node on the local GPFS cluster

1. Change to the installer directory to run the **spectrumscale** commands.

   For IBM Storage Scale 5.1.1 and later:

   ```
   # cd /usr/lpp/mmfs/5.1.1.0/ansible-toolkit
   ```

   For IBM Storage Scale 5.1.0 and earlier:

   ```
   # cd /usr/lpp/mmfs/5.0.4.2/installer
   ```

2. Instantiate the installer node (chef zero server).

   To configure the installer node, issue the following command:

   ```
   ./spectrumscale setup -s InstallNodeIP -i SSHIdentity
   ```

   The -s argument identifies the IP that the nodes will use to retrieve their configuration. This IP will be the one associated with a device on the installer node. This is automatically validated during the setup phase.

   Optionally, you can specify a private SSH key to be used to communicate with the nodes in the cluster definition file, using the -i argument.

Configuring local GPFS cluster

1. Create the CES shared root file system on the local GPFS cluster that will be used by the CES installation.

The local GPFS cluster requires to have a minimum of 2 nodes with 1 disk per node to create the NSDs to be used for the CES shared root file system.

Set up a minimum of two nodes as NSD nodes using the `-n` option.

```
    ./spectrumscale node add <Nsd server Node1> -n
    ./spectrumscale node add <Nsd server Node2> -n

    ./spectrumscale nsd add -p Node1 -fs <local CES shared root filesystem name> -fg 1 <
device>
    ./spectrumscale nsd add -p Node2 -fs <local CES shared root filesystem name> -fg 2 <
device>
```

For example,

```
    ./spectrumscale node add c902f05x05.gpfs.net -n
    ./spectrumscale node add c902f05x06.gpfs.net -n

    ./spectrumscale nsd add -p c902f05x05.gpfs.net  -fs cesSharedRoot -fg 1 "/dev/sdk"
    ./spectrumscale nsd add -p c902f05x06.gpfs.net  -fs cesSharedRoot -fg 2 "/dev/sdl"
```

2. Add the NameNodes created in the local GPFS to set up as protocol nodes and add in the DataNodes.

   Deployment of protocol services is performed on a subset of the cluster nodes that have been designated as protocol nodes using the **`./spectrumscale node add FQDN -p`** command.

   NameNodes are protocol nodes and requires the `-p` option during the node add operation.

   DataNodes are not protocol nodes.

   For example:

   For non-HA:

```
# NameNodes (Protocol node)
./spectrumscale node add c902f05x05.gpfs.net -p
```

   For HA

```
# NameNodes (Protocol node)
./spectrumscale node add c902f05x05.gpfs.net -p
./spectrumscale node add c902f05x06.gpfs.net -p

# DataNodes
./spectrumscale node add c902f05x07.gpfs.net
./spectrumscale node add c902f05x08.gpfs.net
./spectrumscale node add c902f05x09.gpfs.net
./spectrumscale node add c902f05x10.gpfs.net
```

3. If call home is enabled in the cluster definition file, specify the minimum call home configuration parameters.

```
./spectrumscale callhome config -n CustName -i CustID -e CustEmail -cn CustCountry
```

   For more information, see the *Enabling and configuring call home using the installation toolkit* topic in the *IBM Storage Scale: Concepts, Planning, and Installation Guide*.

4. Perform the environment checks before initiating the installation procedure.

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

5. Start the IBM Storage Scale installation to create the local cluster with NameNodes and DataNodes.

```
./spectrumscale install
```

   For information on deploying a Scale cluster through the installation toolkit, see the *Using the installation toolkit to perform installation tasks: Explanations and examples* topic in the *IBM Storage Scale: Concepts, Planning, and Installation Guide*.

Setting up remote mount access

1. After the local GPFS cluster is installed, set up the remote mount file system on the local GPFS cluster to the owning cluster. The owning cluster is the centralized file system (for example, ESS).

   For more information, see the *Mounting a remote GPFS file system* topic in the *IBM Storage Scale: Administration Guide*.

Enabling and deploying CES HDFS

Before you deploy the protocols in a remote mount mode, the local GPFS cluster and the centralized file system GPFS cluster requires to be up and active. Remote mount access is required to be set up and configured. Protocol nodes requires at least 2 GPFS file systems to be mounted: one for CES shared root and one for data.

1. Enable HDFS.

   ```
   ./spectrumscale enable hdfs
   ```

2. Set the CES IPs.

   Data is served through these protocols from a pool of addresses designated as Export IP addresses or CES public IP addresses. This example uses 192.0.2.2 and 192.0.2.3.

   ```
   ./spectrumscale config protocols -e 192.0.2.2, 192.0.2.3
   ```

   Note: For IBM Storage Scale releases earlier to 5.0.5.1, a minimum of two CES IPs is required as input for configuring protocol when HDFS is enabled through the installation toolkit even though the HDFS protocol requires to use only one IP address.

   From IBM Storage Scale 5.0.5.1, only one CES-IP is needed for one HDFS cluster during installation toolkit deployment.

3. Configure the shared root directory.

   Get the CES shared root file system and configure the protocols to point to a file system that will be used as the shared root using the following command:

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

   For example:

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

   For more information, see the *Defining a shared file system for protocols* topic in the *IBM Storage Scale: Concepts, Planning, and Installation Guide*.

   Note: Use a dedicated file system for CES shared root. A dedicated file system can be created with the **mmcrfs** command. The CES shared root must reside on GPFS and must be available when it is configured through the **mmchconfig** command.

   For more information, see the *Setting up Cluster Export Services shared root file system* topic in *IBM Storage Scale: Administration Guide*.

4. Set up the NameNodes and DataNodes for a new CES HDFS cluster.

   ```
   ./spectrumscale config hdfs new -n NAME -nn NAMENODES -dn DATANODES -f FILESYSTEM -d DATADIR
   ```

   The `-f` is the file system name that belongs to the **gpfs.mnt.dir** mountpoint value and `-d DATADIR` option is the **gpfs.data.dir** value as seen in the HDFS Transparency configuration files. Therefore, each new HDFS Transparency cluster requires its own `-d DATADIR` value.

   From IBM Storage Scale 5.0.4.3, the `-f` option can take a remote mount file system only if the file system is already configured as remote mount and is shown in the **mmremotefs** command.

   For example:

```
# /usr/lpp/mmfs/bin/mmremotefs show
Local Name    Remote Name  Cluster name                  Mount Point        Mount Options    Automount  Drive  Priority
remotefs      gpfs504-FS2  c550f6u34.pok.stglabs.ibm.com /remoteFS2         rw               no           -        0
```

where, `remotefs` is the remote file system name.

For non-HA

```
# ./spectrumscale config hdfs new -n myhdfscluster -nn c902f05x05 -dn
c902f05x07,c902f05x08,c902f05x09,c902f05x10 -f remotefs -d gpfshdfs
```

For HA

```
# ./spectrumscale config hdfs new -n myhdfscluster -nn c902f05x05,c902f05x06 -dn
c902f05x07,c902f05x08,c902f05x09,c902f05x10 -f remotefs -d gpfshdfs
```

where,

```
  -n NAME,       --name NAME                           HDFS cluster name.
  -nn NAMENODES, --namenodes      NAMENODES            NameNode hostnames (comma separated).
  -dn DATANODES, --datanodes      DATANODES            DataNode hostnames (comma separated).
  -f FILESYSTEM, --filesystem     FILESYSTEM           Spectrum Scale file system name.
  -d DATADIR,    --datadir        DATADIR              Spectrum Scale data directory name.
```

Note: The -n NAME is the HDFS cluster name. The CES group contains the HDFS cluster name prefix with “hdfs”.

The -d DATADIR is a unique 32-character name required for each HDFS cluster to be created on the same centralized storage.

To configure multiple HDFS clusters, see the “Adding a new HDFS cluster into existing HDFS cluster on the same GPFS cluster (Multiple HDFS clusters)” on page 73 section.

5. List the configured HDFS cluster by running the following command:

   ```
   ./spectrumscale config hdfs list
   ```

6. From IBM Storage Scale 5.1.1.2, Installation Toolkit supports deployment of the following two versions of HDFS Transparency:

   - HDFS Transparency 3.1.1.x
   - HDFS Transparency 3.3.x

   By default, HDFS Transparency 3.1.1.x is deployed. If you want to deploy HDFS Transparency 3.3.x, you need to set the following environment variable before running the installation toolkit command:

   ```
   #export SCALE_HDFS_TRANSPARENCY_VERSION_33_ENABLE=True
   ```

   You can set the *SCALE_HDFS_TRANSPARENCY_VERSION_33_ENABLE* variable in `~/.bashrc`.

7. Perform the environment checks before initiating the installation procedure.

   ```
   ./spectrumscale deploy --pr
   ```

8. Start the IBM Storage Scale installation and the creation of the CES HDFS nodes.

   ```
   ./spectrumscale deploy
   ```

Verify cluster

1. Verify CES HDFS service after deployment is completed.

   ```
   /usr/lpp/mmfs/bin/mmces service list -a
   ```

2. Check if the CES HDFS protocol IPs values are configured properly.

```
/usr/lpp/mmfs/bin/mmces address list
```

For more information, see Listing CES HDFS IPs.

3. After the CES HDFS nodes are installed, create the HDFS client nodes manually. For more information, see Chapter 6, “Apache Hadoop,” on page 489.

   For information on the **spectrumscale**, **mmces** and **mmhdfs** commands, see the *IBM Storage Scale: Command and Programming Reference Guide* guide.

   Note: If HDFS Transparency is a part of the protocols used in the cluster, ensure that the ACL for GPFS file system is set to `-k ALL` after all the protocols are installed. Otherwise, the HDFS NameNodes would fail to start.

   mmlsfs to check the `-k` value.

   mmchfs to change the `-k` value.

   Restart all services and IBM Storage Scale to pick up the `-k` changes.

Manual installation

This section describes how to manually install and create the CES HDFS into a centralized file system.

Run these steps after the Steps in the “Steps for manual installation” on page 33 are completed.

Adding CES HDFS nodes into the centralized file system

1. Ensure that the centralized file system is already installed, configured and active. For example, the ESS.
2. Create a CES shared root file system which will be used by CES installation.

   Note: The recommendation for CES shared root is a dedicated file system. A dedicated file system can be created with the **mmcrfs** command. The CES shared root must reside on GPFS and must be available when it is configured through **mmchconfig**.

   For more information, see the *Setting up Cluster Export Services shared root file system* topic in the *IBM Storage Scale: Administration Guide*.
3. Add nodes designated for CES HDFS to the existing GPFS cluster.

   On a node that already belongs to the GPFS cluster issue the following command:

   ```
   mmaddnode -N c16f1n07.gpfs.net, c16f1n08.gpfs.net, c16f1n09.gpfs.net,
   c16f1n10.gpfs.net, c16f1n11.gpfs.net, c16f1n12.gpfs.net
   ```

   Run **mmlscluster** to ensure that the nodes are added.
4. After the CES HDFS nodes are added to the existing cluster follow the “Enable and Configure CES HDFS” on page 42 section to manually setup non-HA and HA HDFS Transparency cluster.

Enable and Configure CES HDFS

This section describes how to enable and configure CES HDFS manually using the IBM Storage Scale commands.

1. Install HDFS protocol packages on all the CES HDFS nodes.

   On Red Hat Enterprise Linux issue the following command:

   ```
   # rpm -ivh gpfs.hdfs-protocol-<version>.<arch>.rpm
   ```

   For example:

   ```
   rpm -ivh gpfs.hdfs-protocol-3.1.1-0.ppc64.rpm
   ```

2. Configure CES shared root.

   On the CES node, follow the steps in the *Setting up Cluster Export Services shared root file system* topic in the *IBM Storage Scale: Administration Guide* to configure the `cesShareRoot` directory.

   a. Create **`cesSharedRoot`** using the following command:

      ```
      mmchconfig cesSharedRoot=/gpfs/cessharedroot
      ```

      Note: The CES shared root must reside on GPFS and must be available when it is configured through **`mmchconfig`**.

3. Enable CES on the required nodes.

   Users must assign CES nodes belonging to one HDFS cluster to a CES group. If all the CES nodes (NameNodes) belong to a single HDFS cluster, they must be assigned to one CES group for that HDFS cluster. If different CES nodes belong to different HDFS clusters, they must be assigned to different CES groups accordingly in order to differentiate them.

   Note: Every HDFS cluster must have a CES group defined.

   The CES group name should be the hdfs cluster name with a 'hdfs' prefix and will be used as the name of the configuration tar in Clustered Configuration Repository (CCR). For example, if the CES group name is *hdfsmycluster*, the configuration tar in CCR will be *hdfsmycluster.tar* and the hdfs cluster name will be *mycluster*. For more information on CCR, see the *Clustered configuration repository* topic in the *IBM Storage Scale: Concepts, Planning, and Installation Guide*.

   ```
   mmchnode --ces-enable --ces-group=[clustername] -N [Namenode list]
   ```

   For non-HA

   ```
   mmchnode --ces-enable --ces-group [groupname] -N [NameNode]
   ```

   For HA

   ```
   mmchnode --ces-enable --ces-group [groupname] -N [NameNode1,NameNode2]
   ```

   For example, with HDFS HA cluster NameNodes as *c16f1n07* and *c16f1n08*, run the following command:

   ```
   mmchnode --ces-enable --ces-group hdfsmycluster -N c16f1n07,c16f1n08
   ```

4. Define CES IP for CES HA failover.

   A CES address that is associated with a group must be assigned only to a node that is also associated with the same group. For CES HDFS, NameNodes belonging to the same HDFS Transparency cluster belong to the same group.

   A CES HDFS group can be assigned a CES IP address so the HDFS clients can be configured using that CES IP to access the HDFS cluster. This is to configure IP failover provided by CES.

   For example, two HDFS Transparency clusters will have two CES groups (grp1, grp2).

   Each group has two CES nodes (NameNodes).

   Group grp1 will be assigned a CES IP address ip1 and group grp2 will be assigned a CES IP address ip2.

   If the CES node serving the CES IP ip1 fails, the CES IP ip1 will fail over to the other CES node in the group grp1 and the HDFS Transparency service on the 2nd CES node can continue to provide service for that group.

   You can run the following command to assign a CES IP to a CES group:

   ```
   mmces address add --ces-group [groupname] --ces-ip [ip]
   ```

   For non-HA and HA

```
mmces address add --ces-group [groupname] --ces-ip x.x.x.x
```

For example, create CES group named as hdfsmycluster for the HDFS HA cluster:

```
mmces address add --ces-group hdfsmycluster --ces-ip 192.0.2.4
```

5. Configure the HDFS configuration files settings: `core-site.xml`, `hdfs-site.xml`, `gpfs-site.xml` and `hadoop_env.sh` in `/var/mmfs/hadoop/etc/hadoop`. Ensure that the `fs.defaultFS` is configured without the hdfs prefix in the cluster name.

   a. `hadoop_env.sh`

      For non-HA and HA:

      First set the JAVA_HOME configuration to the correct JAVA home path on the node before executing any other **`mmhdfs`** commands. (Replace with your Java version)

      ```
      mmhdfs config set hadoop-env.sh -k JAVA_HOME=/usr/jdk64/jdk1.8.0_112
      ```

   b. `core-site.xml`

      For non-HA and HA

      ```
      mmhdfs config set core-site.xml -k fs.defaultFS=hdfs://mycluster
      ```

   c. `hdfs-site.xml`

      For non-HA

      ```
      mmhdfs config set hdfs-site.xml -k dfs.blocksize=134217728 -k
      dfs.nameservices=mycluster -k dfs.ha.namenodes.mycluster=nn1 -k
      dfs.namenode.rpc-address.mycluster.nn1=c16f1n07.gpfs.net:8020 -k
      dfs.namenode.http-address.mycluster.nn1=c16f1n07.gpfs.net:50070 -k
      dfs.client.failover.proxy.provider.mycluster=org.apache.hadoop.hdfs.server.namenode.ha.ConfiguredFail
      overProxyProvider -k
      dfs.namenode.rpc-bind-host=0.0.0.0 -k dfs.namenode.servicerpc-bind-host=0.0.0.0 -k
      dfs.namenode.lifeline.rpc-bind-host=0.0.0.0 -k dfs.namenode.http-bind-host=0.0.0.0
      -k gpfs.ranger.enabled=scale
      ```

      Note: For non-HA cluster, the property **`dfs.namenode.shared.edits.dir`** in `hdfs-site.xml` configuration file is not needed. Delete this property value otherwise the NameNode will fail to start.

      ```
      mmhdfs config del hdfs-site.xml -k dfs.namenode.shared.edits.dir
      ```

      For HA

      ```
      mmhdfs config set hdfs-site.xml -k dfs.blocksize=134217728 -k dfs.nameservices=cluster -k
      dfs.ha.namenodes.mycluster=nn1,nn2 -k dfs.namenode.rpc-address.mycluster.nn1=c16f1n07.gpfs.net:8020
      -k
      dfs.namenode.http-address.mycluster.nn1=c16f1n07.gpfs.net:50070 -k
      dfs.namenode.rpc-address.mycluster.nn2=c16f1n08.gpfs.net:8020 -k
      dfs.namenode.http-address.mycluster.nn2=c16f1n08.gpfs.net:50070 -k
      dfs.client.failover.proxy.provider.mycluster=org.apache.hadoop.hdfs.server.namenode.ha.ConfiguredFail
      overProxyProvider -k
      dfs.namenode.shared.edits.dir=file:///gpfs/HA-mycluster -k dfs.namenode.rpc-bind-host=0.0.0.0 -k
      dfs.namenode.servicerpc-bind-host=0.0.0.0 -k dfs.namenode.lifeline.rpc-bind-host=0.0.0.0 -k
      dfs.namenode.http-bind-host=0.0.0.0
      -k dfs.ha.fencing.methods='shell(/bin/true)' -k gpfs.ranger.enabled=scale
      ```

      Note: For IBM Storage Scale over shared storage or ESS, the recommended value for **`dfs.blocksize`** is *536870912*. For more information on tuning, see "HDFS Transparency Tuning" on page 262.

   d. `gpfs-site.xml`

      For non-HA and HA

      ```
      mmhdfs config set gpfs-site.xml -k gpfs.mnt.dir=/gpfs/fs0 -k gpfs.data.dir=cluster-data
      -k gpfs.storage.type=shared -k gpfs.replica.enforced=gpfs
      ```

6. Remove the localhost value from the DataNode list by running the following command:

```
mmhdfs worker remove localhost
```

If the localhost value is not removed, then **`mmhdfs hdfs status`** later will show the following errors:

```
c16f1n13.gpfs.net: This node is not a datanode
```

```
mmdsh: c16f1n13.gpfs.net remote shell process had return code 1.
```

7. Add DataNodes.

   Run the following command on the CES transparency node to add DataNodes into an HDFS Transparency cluster.

   ```
   mmhdfs worker add/remove [dn1,dn2,...dnN]
   ```

   For example:

   ```
   mmhdfs worker add c16f1n07.gpfs.net,c16f1n08.gpfs.net,c16f1n09.gpfs.net
   ```

8. Enable proxyuser settings for HDFS Transparency.

   If you are planning to use Hive, Livy or Oozie services with CDP Private Cloud Base, configure the proxyuser settings for those services by running the following commands:

   ```
   mmhdfs config set core-site.xml -k hadoop.proxyuser.hive.groups=*
   mmhdfs config set core-site.xml -k hadoop.proxyuser.hive.hosts=*
   mmhdfs config set core-site.xml -k hadoop.proxyuser.livy.hosts=*
   mmhdfs config set core-site.xml -k hadoop.proxyuser.livy.groups=*
   mmhdfs config set core-site.xml -k hadoop.proxyuser.oozie.hosts=*
   mmhdfs config set core-site.xml -k hadoop.proxyuser.oozie.groups=*
   ```

9. Upload the configuration into CCR.

   Run the following command on to upload the configuration into CCR.

   ```
   mmhdfs config upload
   ```

   Note: Remember to run this command after the HDFS transparency configuration is changed. Otherwise, the modified configuration will be overwritten when HDFS service restarts.

10. For HA environment, the Shared Edit log needs to be initialized from one of the NameNode by running the following command:

    ```
    /usr/lpp/mmfs/hadoop/bin/hdfs namenode -initializeSharedEdits
    ```

11. Enable HDFS and start NameNode service.

    Once the config upload completes, the configuration will be pushed to all the NameNodes and DataNodes when enabling the HDFS service.

    ```
    mmces service enable HDFS
    ```

    This command will start HDFS NameNode service on ALL the CES nodes.

    Note: If the configuration is not correct at this time, the command will print an error message for HDFS Transparency related important settings that are not set properly.

12. Start all the DataNodes by running the following command from one node in the new HDFS Transparency cluster.

    ```
    mmhdfs hdfs-dn start
    ```

13. Check the NameNodes and DataNodes status in the new HDFS Transparency cluster.

    ```
    mmhdfs hdfs status
    ```

14. Verify the HDFS Transparency cluster.

For HDFS CES NON-HA, run the **hdfs shell** command to check that the HDFS Transparency cluster is working.

```
/usr/lpp/mmfs/hadoop/bin/hdfs dfs -ls /
```

For HDFS CES HA, if the NameNodes is started then you need to verify that one NameNode is in Active status and the other is in Standby. Otherwise, HDFS Transparency is not in a healthy state.

Run the following command to retrieve the status of the all the HDFS NameNodes and check the state:

```
/usr/lpp/mmfs/hadoop/bin/hdfs haadmin -getAllServiceState
```

Run the **hdfs shell** command to confirm the HDFS HA cluster is working.

```
/usr/lpp/mmfs/hadoop/bin/hdfs dfs -ls /
```

15. Check if the CES HDFS protocol IPs values are configured properly.

    ```
    /usr/lpp/mmfs/bin/mmces address list
    ```

    For more information, see Listing CES HDFS IPs.

16. After the CES HDFS nodes are installed and verified, create the HDFS client nodes manually. For more information, see Chapter 6, “Apache Hadoop,” on page 489.

    For information about the *spectrumscale*, *mmces*, *mmhdfs* commands, see *IBM Storage Scale: Command and Programming Reference Guide*.

    Note: If HDFS Transparency is a part of the protocols used in the cluster, ensure that the ACL for GPFS file system is set to **-k ALL** after all protocols are installed.

    mmlsfs to check the -k value.

    mmchfs to change the -k value.

    Restart all services and IBM Storage Scale.

Uninstalling HDFS Transparency cluster

This section describes how to manually uninstall CES HDFS.

HDFS Transparency maintains various files that contain configuration and data that is related to the file system related. Because these files are critical for the proper functioning of HDFS Transparency and must be preserved across releases, they are not automatically removed when you uninstall HDFS Transparency.

Follow these steps if you do not intend to use HDFS Transparency on any of the nodes in your cluster.

1. Stop the HDFS Transparency cluster by using the following command.

   ```
   # mmhdfs hdfs stop
   ```

2. Disable the HDFS service from the CES protocols by issuing the next command:

   ```
   # mmces service disable hdfs
   ```

3. To remove the assigned CES IP for the HDFS Transparency cluster, use the following command:

   ```
   # mmces address remove --ces-ip <CES_IP>
   ```

4. To disable CES HDFS on the assigned CES HDFS node, issue the next command:

   ```
   # mmchnode --ces-disable -N <NameNode1,NameNode2>
   ```

5. If no other CES protocols exist, clear the **cesSharedRoot** configuration:

   ```
   #mmchconfig cesSharedRoot=DEFAULT
   ```

6. Uninstall the HDFS Transparency package:

```
#rpm -e gpfs.hdfs-protocol
```

Upgrading

This section describes the process to upgrade CES HDFS Transparency.

Note: Starting with HDFS Transparency 3.1.1-15 and HDFS Transparency 3.2.2-6, dependent JAR files need to be provided. This is also required as a prerequisite for an upgrade. For more information, see the instructions to provide dependent JAR files.

If Kerberos is enabled, see "Prerequisites for Kerberos" on page 64 before you proceed to the upgrade sections.

Installation toolkit upgrade process for HDFS Transparency

From IBM Storage Scale 5.0.5.0, the installation toolkit supports offline upgrade for HDFS protocol.

From IBM Storage Scale 5.0.5.1, the installation toolkit supports online upgrade for HDFS protocol.

Online installation toolkit upgrade

From IBM Storage Scale 5.0.5.1 and BDA integration 1.0.1.1, the installation toolkit supports the CES HDFS Transparency online upgrade.

1. After the IBM Storage Scale install package is extracted, starting from IBM Storage Scale 5.0.5.1 with Toolkit for HDFS at 1.0.1.1 and HDFS Transparency 3.1.1-1, the default location (`/usr/lpp/mmfs/5.0.5.1/`) for the files will contain the correct packages to do the online upgrade. Ensure that the HDFS Transparency and Toolkit for HDFS residing in `/usr/lpp/mmfs/<Scale version>/hdfs_rpms/rhel7/hdfs_3.1.1.x` (Default Red Hat location) have the support combination versions as stated in the CES HDFS "HDFS Transparency support matrix" on page 27 section.
2. For IBM Storage Scale 5.1.1 and later:

   ```
   # cd /usr/lpp/mmfs/5.1.1.0/ansible-toolkit
   ```

   For IBM Storage Scale 5.1.0 and earlier:

   ```
   # cd /usr/lpp/mmfs/5.0.5.1/installer
   ```

   Run the following command:

   ```
   # ./spectrumscale setup -s <Installer IP>
   ```

   For ESS:

   ```
   # ./spectrumscale setup -s <EMS Node> -st ess
   ```

3. Populate the existing configuration:

   ```
   # ./spectrumscale config populate -N < HDFS Node>
   Where HDFS NODE is any node in the HDFS Transparency cluster.
   For example:
   ./spectrumscale config populate -N c902f09x11.gpfs.net
   ```

4. The installation toolkit automatically updates only the HDFS package when HDFS protocol is enabled in the toolkit. To check if HDFS is enabled, run the following command:

   ```
   ./spectrumscale node list
   ```

5. Run upgrade precheck.

   ```
   # ./spectrumscale upgrade precheck
   ```

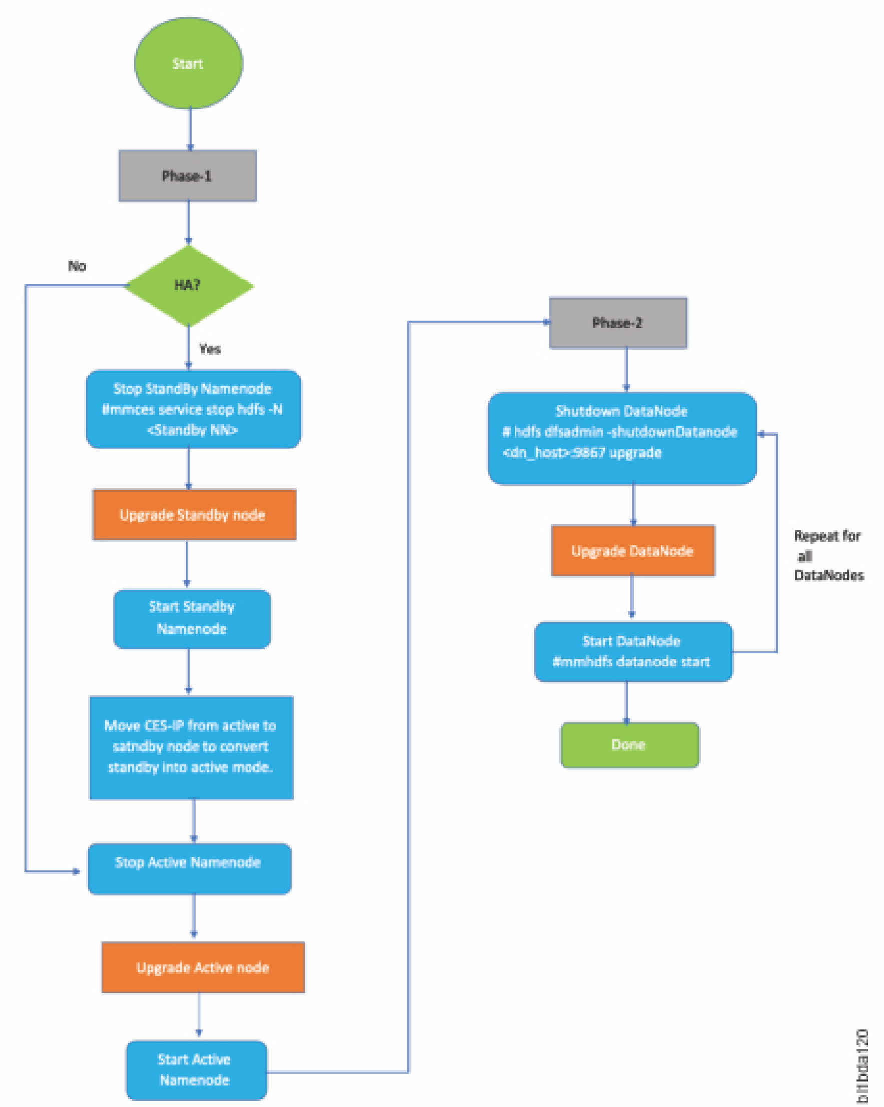
6. Deploy the upgrade if the precheck is successful.

```
# ./spectrumscale upgrade run
```

For more information about the installation toolkit online upgrade process, see the following topics in the *IBM Storage Scale: Concepts, Planning, and Installation Guide*:

- *Upgrading IBM Storage Scale components with the installation toolkit*
- *Upgrade process flow*
- *Performing online upgrade by using the installation toolkit*

For CES HDFS, the online upgrade flow is as follows:


Start
Phase-1
HA?
No
Yes
Stop StandBy Namenode #mmces service stop hdfs -N <Standby NN>
Upgrade Standby node
Start Standby Namenode
Move CES-IP from active to satndby node to convert standby into active mode.
Stop Active Namenode
Upgrade Active node
Start Active Namenode
Phase-2
Shutdown DataNode # hdfs dfsadmin -shutdownDatanode <dn_host>:9867 upgrade
Upgrade DataNode
Start DataNode #mmhdfs datanode start
Repeat for all DataNodes
Done
bl1bda120

Offline installation toolkit upgrade

From IBM Storage Scale 5.0.5, with Toolkit for HDFS at 1.0.1.0 and HDFS Transparency at 3.1.1-1, the installation toolkit supports only the offline upgrade process for HDFS protocol. If other protocols (for example, SMB, NFS) are also configured along with HDFS, those protocols will also be updated.

Ensure that you review the *Performing offline upgrade or excluding nodes from upgrade using installation toolkit* documentation in the *IBM Storage Scale: Concepts, Planning, and Installation Guide*.

The installation toolkit will update the nodes in the upgrade config offline node list only if those nodes have been shut down and are suspended. Ensure that there are sufficient quorum nodes available to run GPFS before shutting down the CES NameNodes and DataNodes.

Offline HDFS installation upgrade procedure

The following process uses IBM Storage Scale 5.0.5 as an example.

1. After the IBM Storage Scale installation package is extracted, starting from IBM Storage Scale 5.0.5 with Toolkit for HDFS at 1.0.1.0 and HDFS Transparency 3.1.1-1, the default location (`/usr/lpp/mmfs/5.0.5.0/`) for the files will contain the correct packages to do the offline upgrade. Ensure that the HDFS Transparency and Toolkit for HDFS residing in `/usr/lpp/mmfs/<Scale version>/hdfs_rpms/rhel7/hdfs_3.1.1.x` (Default Red Hat location) have the support combination versions as stated in the CES HDFS "HDFS Transparency support matrix" on page 27 section.
2. For IBM Storage Scale 5.1.1 and later:

   ```
   # cd /usr/lpp/mmfs/5.1.1.0/ansible-toolkit
   ```

   For IBM Storage Scale 5.1.0 and earlier:

   ```
   # cd /usr/lpp/mmfs/5.0.5.1/installer
   ```

   Run the following command:

   ```
   # ./spectrumscale setup -s <Installer IP>
   ```

   For ESS:

   ```
   # ./spectrumscale setup -s <EMS Node> -st ess
   ```

3. Populate the existing configuration:

   ```
   # ./spectrumscale config populate -N < HDFS Node>
   Where HDFS NODE is any node in the HDFS Transparency cluster.
   For example:
   ./spectrumscale config populate -N c902f09x11.gpfs.net
   ```

4. Shut down the NameNodes and DataNodes.

   ```

   # /usr/lpp/mmfs/bin/mmshutdown -N <NameNode and DataNodes list>

   For example,
   # /usr/lpp/mmfs/bin/mmshutdown -N c902f09x09,c902f09x10,c902f09x11,c902f09x12
         Fri Feb 21 00:09:11 EST 2020: mmshutdown: Starting force unmount of GPFS file systems
         Fri Feb 21 00:09:56 EST 2020: mmshutdown: Shutting down GPFS daemons
         Fri Feb 21 00:10:04 EST 2020: mmshutdown: Finished
   ```

   Note: Only the HDFS NameNodes and DataNodes need to be shut down. The other CES protocol nodes do not need to be shut down.

5. Suspend CES service on all the NameNodes.

   ```

   # /usr/lpp/mmfs/bin/mmces node suspend -N <NameNodes>

   For example,
   [root@c902f09x09 installer]# /usr/lpp/mmfs/bin/mmces node suspend -N c902f09x11,c902f09x12
   Node c902f09x11.gpfs.net now in suspended state.
   Node c902f09x12.gpfs.net now in suspended state.
   ```

   Note:

   - Shutting down GPFS using the **mmshutdown** command will stop the CES HDFS NameNodes and DataNodes.
   - For HDFS Transparency to be upgraded, all the CES NameNodes are required to be suspended.

6. The installation toolkit automatically updates only the HDFS package when HDFS protocol is enabled in the toolkit. To check if HDFS is enabled, run the following command:

   ```
   ./spectrumscale node list
   ```

7. Run upgrade configuration in offline mode for the NameNodes and DataNodes.

```
# ./spectrumscale upgrade config offline -N <List of NameNodes and DataNodes>

For example,
[root@c902f09x09 installer]# ./spectrumscale upgrade config offline -N
c902f09x09,c902f09x10,c902f09x11,c902f09x12
[ INFO  ] The node c902f09x09.gpfs.net is added as offline.
[ INFO  ] The node c902f09x10.gpfs.net is added as offline.
[ INFO  ] The node c902f09x11.gpfs.net is added as offline.
[ INFO  ] The node c902f09x12.gpfs.net is added as offline.
```

Note:

- This will only upgrade the HDFS Transparency nodes.
- Ensure that you list all the NameNodes and DataNodes in the HDFS Transparency cluster into the offline list.

8. Check the protocol configuration list to ensure that they are set to “offline".

```
# ./spectrumscale upgrade config list
```

For example:

The upgrade config list of NameNodes and other protocol nodes shows under `Phase2: Protocol Nodes Upgrade`. The upgrade config list of DataNodes and other nodes shows under `Phase1: Non Protocol Nodes Upgrade`.

In the following example, there are two NameNodes (c902f09x11.gpfs.net, c902f09x12.gpfs.net) and two DataNodes (c902f09x09.gpfs.net,c902f09x10.gpfs.net):

```
    # ./spectrumscale upgrade config list
    [ INFO  ] GPFS Node               SMB        NFS       OBJ      HDFS     GPFS
    [ INFO  ]
    [ INFO  ] Phase1: Non Protocol Nodes Upgrade
    [ INFO  ] c902f09x09.gpfs.net      -          -         -         -      offline
    [ INFO  ] c902f09x10.gpfs.net      -          -         -         -      offline
    [ INFO  ]
    [ INFO  ] Phase2: Protocol Nodes Upgrade
    [ INFO  ] c902f09x11.gpfs.net   offline    offline   offline   offline   offline
    [ INFO  ] c902f09x12.gpfs.net   offline    offline   offline   offline   offline
       [ INFO  ]
```

9. Run the upgrade precheck.

```
# ./spectrumscale upgrade precheck
```

10. Deploy the upgrade if the precheck is successful.

```
# ./spectrumscale upgrade run
```

11. After the upgrade completes successfully, start the HDFS Transparency cluster.

    a. Start the NameNodes.

    ```
    /usr/lpp/mmfs/bin/mmces service start hdfs -a
    ```

    b. Start the DataNodes.

    ```
    /usr/lpp/mmfs/hadoop/sbin/mmhdfs hdfs-dn start
    ```

Manual rolling upgrade for HDFS Transparency

HDFS Transparency 3.1.1-x is the version for CES HDFS integration. HDFS Transparency supports rolling upgrades when the commands are executed manually on the command line and not through the installation toolkit.

Manual rolling upgrade for CES HDFS Transparency NameNode

This topic lists the steps to manually perform a rolling upgrade for the NameNodes.

As root, follow the steps listed below to perform the rolling upgrade for the NameNode(s):

1. If NameNode HA is configured on the standby HDFS Transparency NameNode, stop the standby NameNode from the bash console with the following command:

   ```
   /usr/lpp/mmfs/bin/mmces service stop hdfs
   ```

   Note: If CES protocols such as SMB co-existed with HDFS, then the CES IP of SMB will failover from the standby NameNode to the active NameNode.

   If HDFS Transparency NameNode HA was not configured, then go to step 5.

   Note: When you upgrade the HDFS Transparency NameNode with non-HA configured, HDFS Transparency service gets interrupted.

2. Upgrade the standby HDFS Transparency NameNode.

   cd to the directory where the upgrade HDFS Transparency package resides.

   Run the following command from the bash console to update the HDFS Transparency package:

   ```
   rpm -Uvh gpfs.hdfs-protocol-3.1.1-<version>.<os>.rpm
   ```

3. Start the standby NameNode.

   Run the following command from the bash console:

   ```
   /usr/lpp/mmfs/bin/mmces service start hdfs
   ```

4. Move the CES IP of HDFS from the active NameNode to the standby NameNode.

   This does a failover of the current active NameNode to become the new standby NameNode.

   Run the following command from the bash console:

   ```
   /usr/lpp/mmfs/bin/mmces address move --ces-ip x.x.x.x --ces-node <standby_namenode_host>
   ```

   The original standby NameNode is now the active NameNode after the CES IP is moved successfully.

5. Check to see if the new active NameNode is active.

   Run the following commands from the bash console:

   ```
   /usr/lpp/mmfs/bin/mmces service list -a
   /usr/lpp/mmfs/hadoop/sbin/mmhdfs hdfs-nn status
   /usr/lpp/mmfs/hadoop/bin/hdfs haadmin -getAllServiceState
   ```

6. Stop the new standby HDFS Transparency NameNode, for which the status changed from active to standby, so that the HDFS Transparency package can be upgraded.

   Run the following command from the bash console:

   ```
   /usr/lpp/mmfs/bin/mmces service stop hdfs
   ```

   Note: If other CES protocols such as SMB co-existed with HDFS, then the CES IP of SMB will failover to the new active NameNode.

7. Upgrade the new standby HDFS Transparency NameNode.

Run the following command from the bash console to upgrade the HDFS Transparency package:

```
rpm -Uvh gpfs.hdfs-protocol-3.1.1-<version>.<os>.rpm
```

8. Start the new standby NameNode

   Run the following command from the bash console:

   ```
   /usr/lpp/mmfs/bin/mmces service start hdfs
   ```

   Note: If other CES protocols such as SMB co-existed with HDFS, then the CES IP of SMB will fail over back to the new standby NameNode.

Manual rolling upgrade for CES HDFS Transparency DataNode

This topic lists the steps to manually perform a rolling upgrade for the DataNodes.

Note: This is an online upgrade. Connected clients will wait for the DataNode to restart and continue the operation. The default timeout is 30 seconds and can be modified on client side by setting **`dfs.client.datanode-restart.timeout`** to a higher value. If the DataNode is not restarted within the specified time, the client considers the DataNode as dead (by default for 10 minutes, see **`dfs.client.write.exclude.nodes.cache.expiry.interval.millis`**) and will failover to another DataNode if **`dfs.replication`** is greater than 1.

1. Copy the latest gpfs.hdfs-protocol-<VERSION>.<ARCH> package to the DataNode that you want to upgrade.
2. Log in to the DataNode.
3. Upgrade the RPM by running the following command:

   ```
   rpm -Uvh gpfs.hdfs-protocol-<VERSION>.<ARCH>
   ```

4. Shut down the DataNode by running the following command:

   ```
   /usr/lpp/mmfs/hadoop/bin/hdfs dfsadmin -shutdownDatanode <HOST>:<IPC_PORT> upgrade
   ```

   Note: The default IPC port is 9867. You can see the IPC port value in **`dfs.datanode.ipc.address`** under the `/var/mmfs/hadoop/etc/hadoop/hdfs-site.xml` file.

5. Wait for the DataNode to shut down. Run the following command to check that the DataNode status is set to dead:

   ```
   /usr/lpp/mmfs/hadoop/sbin/mmhdfs datanode status
   ```

   For example:

   ```
   # /usr/lpp/mmfs/hadoop/sbin/mmhdfs datanode status

   c902f08x04.gpfs.net: cescluster1: datanode is dead, previous pid is 26077
   ```

6. Start the DataNode again by running the following command:

   ```
   /usr/lpp/mmfs/hadoop/sbin/mmhdfs datanode start
   ```

Configuring

The following configurations are for manually configuring HDFS Transparency. For example, set up HDFS Transparency for open-source Apache or Cloudera CDP stack.

For configuring, Hadoop distribution must be installed under `$YOUR_HADOOP_PREFIX` on each machine in the Hadoop cluster. The configurations for IBM Storage Scale HDFS transparency are located under `/usr/lpp/mmfs/hadoop/etc/hadoop` (for HDFS Transparency 2.7.3-x) or `/var/mmfs/hadoop/etc/hadoop` (for HDFS Transparency 3.0.x) for any Hadoop distribution. Configuration files for

Hadoop distribution are located in different locations. For example, `/etc/hadoop/conf` for Cloudera CDP.

The `core-site.xml` and `hdfs-site.xml` configuration files should be synchronized between all the nodes and kept identical for the IBM Storage Scale HDFS Transparency and Hadoop Distribution. The `log4j.properties` configuration file can differ between the IBM Storage Scale HDFS Transparency and the open-source Apache Hadoop distribution.

Password-less ssh access

Ensure that the root password-less ssh access does not prompt a response for the user. If the root password-less access configuration cannot be setup, HDFS transparency fails to start. The **`mmhadoopctl`** and **`mmhdfs`** commands require password-less ssh to all the nodes including itself.

If the IBM Storage Scale cluster is configured as `adminMode=central`, HDFS Transparency NameNodes can be configured on the management nodes of the IBM Storage Scale cluster. To check if the IBM Storage Scale cluster is configured as `adminMode=central`, run **`mmlsconfig adminMode`**.

If the IBM Storage Scale cluster is configured in sudo wrapper mode, IBM Storage Scale requires the user to have password-less root access to all the other nodes as a common user. To check if the IBM Storage Scale cluster is configured in sudo wrapper mode, log in as a root user in the node and execute **`ssh <non-root>@<other-node>`** in the password-less mode. With IBM Storage Scale in sudo wrapper mode, HDFS Transparency still requires the node to have root access to all the other nodes including itself to run the **`mmhadoopctl`** and **`mmhdfs`** commands.

HDFS Transparency provides the following options for root password-less requirement:

1. **Local cluster options**

   For the local cluster, follow one of the following options for the root password-less requirement:

   a. By default, HDFS Transparency requires root password-less access between any two nodes in the HDFS Transparency cluster.

   b. If the above option is not feasible, you need at least one node with root password-less access to all the other HDFS Transparency nodes and to itself. In such a case, **`mmhadoopctl/mmhdfs`** command can be run only on this node and this node should be configured as HDFS Transparency NameNodes. If NameNode HA is configured, all NameNodes should be configured with root password-less access to all DataNodes.

      Note:

      - If you configure the IBM Storage Scale cluster in admin central mode (**`mmchconfig adminMode=central`**), you can configure HDFS Transparency NameNodes on the IBM Storage Scale management nodes. Therefore, you have root password-less access from these management nodes to all the other nodes in the cluster.
      - If the file system is remotely mounted, HDFS Transparency requires two password-less access configurations: one is for the local cluster (configure HDFS Transparency according to this option for password-less access in the local cluster) and the other is for remote file system.

2. **Remote cluster options**

   For the remote file system, follow one of the following options for the root password-less requirement:

   a. By default, HDFS Transparency NameNodes require root password-less access to at least one of the contact nodes (the 1st contact node is recommended if you cannot configure all contact nodes as password-less access) from the remote cluster.

      For example, in the following cluster, `ess01-dat.gpfs.net` and `ess02-dat.gpfs.net` are contact nodes. `ess01-dat.gpfs.net` is the first contact node because it is listed first in the property **`Contact nodes`**:

      ```
      # /usr/lpp/mmfs/bin/mmremotecluster show all
      Cluster name: test01.gpfs.net
      Contact nodes: ess01-dat.gpfs.net,ess02-dat.gpfs.net
      ```

```
SHA digest: abe321118158d045f5087c00f3c4b0724ed4cfb8176a05c348ae7d5d19b9150d
File systems: latestgpfs (gpfs0)
```

Note: HDFS Transparency DataNodes do not require root password-less access to the contact nodes.

b. From HDFS Transparency 2.7.3-3, HDFS Transparency supports non-root password-less access to one of the contact nodes as a common user (instead of root user).

First, on HDFS Transparency NameNodes, configure password-less access for the root user as a non-privileged user to the contact nodes (at least one contact node and recommend the first contact node) from the remote cluster. Here, the *gpfsadm* user is used as an example.

Add the following into the `/usr/lpp/mmfs/hadoop/etc/hadoop/gpfs-site.xml` (for HDFS Transparency 2.7.3-x) or `/var/mmfs/hadoop/etc/hadoop/gpfs-site.xml` (for HDFS Transparency 3.1.x) file on HDFS Transparency NameNodes.

```

  <property>
    <name>gpfs.ssh.user</name>
    <value>gpfsadm</value>
  </property>
```

On one of the contact nodes (the first contact node is recommended), edit `/etc/sudoers` using `visudo` and add the following to the `sudoers` file.

```
gpfsadm  ALL=(ALL)       NOPASSWD: /usr/lpp/mmfs/bin/mmlsfs, /usr/lpp/mmfs/bin/
mmlscluster,
/usr/lpp/mmfs/bin/mmlsnsd, /usr/lpp/mmfs/bin/mmlsfileset, /usr/lpp/mmfs/bin/mmlssnapshot,
/usr/lpp/mmfs/bin/mmcrsnapshot, /usr/lpp/mmfs/bin/mmdelsnapshot, /usr/lpp/mmfs/bin/
tslsdisk
```

The `gpfsadm` user can run these IBM Storage Scale commands for any filesets in the file system using the sudo configurations above.

Note: Comment out `Defaults requiretty`. Otherwise, `sudo: sorry, you must have a tty to run sudo` error will occur.

```
#
# Disable "ssh hostname sudo <cmd>", because it will show the password in clear.
#         You have to run "ssh -t hostname sudo <cmd>".
#
#Defaults    requiretty
```

Note: Before you start HDFS Transparency, log in HDFS Transparency NameNodes as root and run **`ssh gpfsadmin@<the configured contact node> /usr/lpp/mmfs/bin/mmlsfs <fs-name>`** to confirm that it works.

c. Manually generate the internal configuration files from the contact node and copy them onto the local nodes so that you do not require root or user password-less ssh to the contact nodes.

From HDFS transparency 2.7.3-2, you can configure **`gpfs.remotecluster.autorefresh`** as *false* in `/usr/lpp/mmfs/hadoop/etc/hadoop/gpfs-site.xml` (for HDFS Transparency 2.7.3-x) or `/var/mmfs/hadoop/etc/hadoop/gpfs-site.xml` (for HDFS Transparency 3.1.x).

Manually copy the `/usr/lpp/mmfs/hadoop/sbin/initmap.sh` script from the NameNode to one of the contact nodes. The script can be copied to any directory.

Create the `/var/mmfs/hadoop/etc/`hadoop directory on the contact node and copy the contents of the `/var/mmfs/hadoop/etc/`hadoop directory from the NameNode to the directory created on the contact node.

Log on the contact node as root and run the **`initmap.sh`** command.

For example, to get the initmap files for two file systems on the contact node, run the following command:

```
/<savedir>/initmap.sh -i all <fs1>,<fs2>
```

Note: Do not use the **-d** option when running on the contact node.

Copy the generated internal configuration files to all the HDFS Transparency nodes.

The initmap.sh script requires to be re-run on the remote system if any of the following are changed:

- There are updates to the dataReplica configuration values for the filesystem.
- The gpfs cluster name (from the **mmlscluster** output) is changed through the **mmchcluster** command on the remote system.
- There are updates to the filesystem name in either the remote or local clusters.
- There are updates to the contact nodes information from the local cluster to the remote cluster.

For the **initmap.sh** script command syntax and generated internal configuration files, see "Cluster and file system information configuration" on page 62.

Note: If **gpfs.remotecluster.autorefresh** is configured as *false*, the snapshot from Hadoop interface is not supported against the remote mounted file system.

If the IBM Storage Scale cluster is configured as **adminMode**=*central* (check by executing **mmlsconfig adminMode**), HDFS Transparency NameNodes can be configured on the management nodes of the IBM Storage Scale cluster.

OS tuning for all nodes in HDFS Transparency

This topic describes the ulimit tuning.

ulimit tuning

For all nodes, `ulimit -n` and `ulimit -u` must be larger than or equal to 65536. Smaller value makes the Hadoop java processes report unexpected exceptions.

In Red Hat, add the following lines at the end of `/etc/security/limits.conf` file:

```
*       soft nofile 65536
*       hard nofile 65536

*       soft nproc 65536
*       hard nproc 65536
```

For other Linux distributions, see the relevant documentation.

After the above change, all the Hadoop services must be restarted for the change to take effect.

If you are using Ambari, ensure that you restart each ambari-agent and then restart HDFS Transparency in order to pick up the changes in the `/etc/security/limits.conf`.

Note:

- This must be done on all nodes including the Hadoop client nodes and the HDFS Transparency nodes.
- If the ambari agent is restarted using the command line (for example, **ambari-agent restart**), the ulimit inherited by the DataNode process is from the `/etc/security/limits.conf` file.

  If the ambari agent is started using systemd, (for example, server reboot), the ulimit inherited by the DataNode process is from the systemd config file (for example, LimitNOFILE value).

kernel.pid_max

Usually, the default value is 32K. If you see the `allocate memory` error or `unable to create new native thread` error, you can try to increase the **kernel.pid_max** by adding **kernel.pid_max**=*99999* at the end of `/etc/sysctl.conf` followed by running the **sysctl -p** command.

Configure NTP to synchronize the clock in HDFS Transparency

For distributed cluster, configure NTP to synchronize the clock in the cluster. If the cluster can access the internet, take the public NTP servers to synchronize the clock on the nodes. However, if the cluster does not have access to the internet, then configure one of the nodes as the NTP server so that all the other nodes will synchronize the clock according to that NTP server.

Refer to the CONFIGURING NTP USING NTPD to configure NTP on RHEL 7.

HDFS Transparency requires the clock to be synchronized on all nodes. Otherwise potential issues will occur.

Configure Hadoop nodes

On HortonWorks HDP, you could configure Hadoop on Ambari GUI. If you are not familiar with HDFS/Hadoop, set up the native HDFS first by seeing the Hadoop cluster setup guide. Setting up the HDFS Transparency to replace the native HDFS is easier after you set up HDFS/Hadoop.

Hadoop and HDFS Transparency must take the same `core-site.xml`, `hdfs-site.xml`, slaves (Hadoop 2.7.x) or workers (Hadoop 3.0.x+), `hadoop-env.sh` and `log4j.properties` for both Hadoop nodes and HDFS Transparency. This means, native HDFS in Hadoop and HDFS Transparency must take the same NameNodes and DataNodes.

Note:

1. For HortonWorks HDP, the configuration files above are located under `/etc/hadoop/conf`. For open source Apache Hadoop, the configuration files are located under `$YOUR_APACHE_HADOOP_HOME/etc/hadoop` and `/usr/lpp/mmfs/hadoop/etc/hadoop` for HDFS Transparency 2.7.3-x.
2. From HDFS Transparency 2.7.3-3, the configurations are located under `/usr/lpp/mmfs/hadoop/etc/hadoop`. From HDFS Transparency 3.0.0, the configurations are located under `/var/mmfs/hadoop/etc/hadoop`.

If your native HDFS NameNodes are different than HDFS Transparency NameNodes, you need to update `fs.defaultFS` in your Hadoop configuration (for HortonWorks HDP it is located under `/etc/Hadoop/conf`. If it is open source Apache Hadoop, it is located under `$YOUR_HADOOP_PREFIX/etc/hadoop/`.):

```
<property>
<name>fs.defaultFS</name>
<value>hdfs://hs22n44:8020</value>
</property>
```

For HDFS Transparency 2.7.0-x, 2.7.2-0, 2.7.2-1, do not export the Hadoop environment variables on the HDFS Transparency nodes because this can lead to issues when the HDFS Transparency uses the Hadoop environment variables to map to its own environment. The following Hadoop environment variables can affect HDFS Transparency:

- **HADOOP_HOME**
- **HADOOP_HDFS_HOME**
- **HADOOP_MAPRED_HOME**
- **HADOOP_COMMON_HOME**
- **HADOOP_COMMON_LIB_NATIVE_DIR**
- **HADOOP_CONF_DIR**
- **HADOOP_SECURITY_CONF_DIR**

For HDFS Transparency versions 2.7.2-3+, 2.7.3-x and 3.0.x+, the environmental variables listed above can be exported except for **HADOOP_COMMON_LIB_NATIVE_DIR**. This is because HDFS Transparency uses its own native .so library.

For HDFS Transparency versions 2.7.2-3+ and 2.7.3-x:

- If you did not export HADOOP_CONF_DIR, HDFS Transparency will read all the configuration files under `/usr/lpp/mmfs/hadoop/etc/hadoop` such as the `gpfs-site.xml` file and the `hadoop-env.sh` file.
- If you export HADOOP_CONF_DIR, HDFS Transparency will read all the configuration files under `$HADOOP_CONF_DIR`. As `gpfs-site.xml` is required for HDFS Transparency, it will only read the `gpfs-site.xml` file from the `/usr/lpp/mmfs/hadoop/etc/hadoop` directory.

For questions or issues with HDFS Transparency configuration, send an email to scale@us.ibm.com.

Configure HDFS Transparency nodes

This section provides information on configuring HDFS transparency nodes.

Hadoop configurations files

This topic lists the Hadoop configuration files.

By default, HDFS Transparency 2.7.3-x uses the following configuration files located under `/usr/lpp/mmfs/hadoop/etc/hadoop`:

- `core-site.xml`
- `hdfs-site.xml`
- `slaves`
- `log4j.properties`
- `hadoop-env.sh`

HDFS Transparency 3.0.0+ uses the following configuration files located under `/var/mmfs/hadoop/etc/hadoop`:

- `core-site.xml`
- `hdfs-site.xml`
- `workers`
- `log4j.properties`
- `hadoop-env.sh`

Configure the storage mode

Use this procedure to configure the storage mode.

Modify the `/var/mmfs/hadoop/etc/hadoop/gpfs-site.xml` file on the `hdfs_transparency_node1` node:

```
<property>
<name>gpfs.storage.type</name>
<value>local</value>
</property>
```

The property **`gpfs.storage.type`** is used to specify the storage mode: local or shared. Local is for IBM Storage Scale FPO file system and shared is for IBM Storage Scale over Centralized Storage or remote mounted file system. This is a required configuration parameter and the `gpfs-site.xml` configuration file must be synchronized with all the HDFS Transparency nodes after the modification.

Update other configuration files

Use this procedure to update the configuration files.

Note: To configure Hadoop HDFS, Yarn, etc. refer to the `hadoop.apache.org` website.

Configuring Apache Hadoop

Modify the `/var/mmfs/hadoop/etc/hadoop/gpfs-site.xml` file on the `hdfs_transparency_node1` node:

```
<property>
<name>gpfs.mnt.dir</name>
<value>/gpfs_mount_point</value>
</property>

<property>
<name>gpfs.data.dir</name>
<value>data_dir</value>
</property>

<property>
<name>gpfs.supergroup</name>
<value>hdfs,root</value>
</property>

<property>
<name>gpfs.replica.enforced</name>
<value>dfs</value>
</property>
```

In `gpfs-site.xml`, all the Hadoop data is stored under the `/gpfs_mount_point/data_dir` directory. You can have two Hadoop clusters over the same file system and these clusters are isolated from each other. When Hadoop operates the file, one limitation is that if there is a link under the `/gpfs_mount_point/data_dir` directory that points to a file outside the `/gpfs_mount_point/data_dir` directory, it reports an exception because that file is not accessible by Hadoop.

If you do not want to explicitly configure the **`gpfs.data.dir`** parameter, leave it as null. For example, keep its value as <value></value>.

Note: Do not configure it as <value>/</value>.

The `gpf.supergroup` must be configured according to your cluster. You need to add some Hadoop users, such as HDFS, yarn, hbase, hive, oozie, etc under the same group named Hadoop and configure `gpfs.supergroup` as Hadoop. You might specify two or more comma-separated groups as `gpfs.supergroup`. For example, `group1,group2,group3`.

Note: Users in `gpfs.supergroup` are super users and they can control all the data in `/gpfs_mount_point/data_dir` directory. This is similar to the user root in Linux. Since HDFS Transparency 2.7.3-1, `gpfs.supergroup` could be configured as `hdfs,root`.

The **`gpfs.replica.enforced`** parameter is used to control the replica rules. Hadoop controls the data replication through the **`dfs.replication`** parameter. When running Hadoop over IBM Storage Scale, IBM Storage Scale has its own replication rules. If you configure `gpfs.replica.enforced` as dfs, **`dfs.replication`** is always effective unless you specify **`dfs.replication`** in the command options when submitting jobs. If `gpfs.replica.enforced` is set to *gpfs*, all the data will be replicated according to IBM Storage Scale configuration settings. The default value for this parameter is *dfs*.

Usually, you must not change `core-site.xml` and `hdfs-site.xml` located under `/var/mmfs/hadoop/etc/hadoop/`. These two files must be consistent as the files used by Hadoop nodes.

You need to modify `/var/mmfs/hadoop/etc/hadoop/workers` to add all HDFS transparency DataNode hostnames and one hostname per line, for example:

```
# cat /var/mmfs/hadoop/etc/hadoop/workers
hs22n44
hs22n54
hs22n45
```

You might check `/var/mmfs/hadoop/etc/hadoop/log4j.properties` and modify it accordingly. This file might be different from the `log4j.properties` used by Hadoop nodes.

After you finish the configurations, use the following command to sync it to all IBM Storage Scale HDFS transparency nodes:

```
hdfs_transparency_node1#/usr/lpp/mmfs/hadoop/sbin/mmhadoopctl connector syncconf /var/mmfs/hadoop/etc/hadoop
```

Configure storage type data replication

To get the file system data replica values, run the **`mmlsfs <fsName> -r -R`** command to review the output values. The value of **`-r`** is the default number of data replicas and the value of **`-R`** is the maximum number of data replicas.

Important: The value of **`-R`** cannot be changed after the file system creation. Usually, the value *3* is the recommended values for **`-r`** and **`-R`** if you are using IBM Storage Scale FPO and the value *1* for **`-r`** and *2* for **`-R`** are recommended values for production when you are using Centralized Storage.

For different storage modes, refer to the following table for recommended combination for `dfs.replication`, `gpfs.replica.enforced` and file system data replica.

Table 11. Configurations for data replication

Storage mode	**`dfs.replication`**	**`gpfs.replica.enforced`**	**File system data replica**	**Comments**
#1 FPO (gpfs.storage.type=local)	3	gpfs or dfs	**`-r`** = 3 **`-R`** = 3	Other combinations are not recommended.
#2 ESS (gpfs.storage.type=shared)	1	dfs	**`-r`** = 1 **`-R`** = 2 **`-r`** = 1 **`-R`** = 3	Follow the HDFS protocol. But the job will fail if one DN is down after **`getBlockLocation`** is returned. Potential issue: Does not show the advantage that all DN can access the blocks. If you are using this configuration you must use the **`mmlsattr`** command to check the file replication value. If the set file replication value is less than the **`dfs.replication`** value, the HDFS interface cannot be used to check the file replication value because the NameNode returns at least the **`dfs.replication`** value in the shared storage mode.

Table 11. Configurations for data replication (continued)

Storage mode	`dfs.replication`	`gpfs.replica.enforced`	File system data replica	Comments
#3 ESS (gpfs.storage.type=shared)	2 or 3	gpfs	**-r** = 1 **-R** = 2 **-r** = 1 **-R** = 3	Follow the HDFS protocol (returns 2 or 3 DNs) but does not match the real storage usage on GPFS level. Job will not fail if one DN is down after **`getBlockLocation`** is returned. Potential risk: Upper-layer applications calculate the disk space consumption as replication * file size, thinking a file takes more storage space than it actually does. HDFS Transparency will still use the actual disk space correctly.
#4 ESS (gpfs.storage.type=shared)	1	gpfs	**-r** = 1 **-R** = 2 **-r** = 1 **-R** = 3	Do not use if the application wants to set the replication value from HDFS protocol.
#5 ESS (gpfs.storage.type=shared)	2 or 3	dfs	**-r** = 1 **-R** = 2 **-r** = 1 **-R** = 3	All the data will be set as replica 2 or 3 which will not take advantage of using IBM ESS or SAN storage. If you are using this configuration you must use the **`mmlsattr`** command to check the file replication value. If the set file replication value is less than the **`dfs.replication`** value, the HDFS interface cannot be used to check the file replication value because the NameNode returns at least the **`dfs.replication`** value in the shared storage mode.

Note:

- The **dfs.replication** is defined in the hdfs-site.xml file. The **gpfs.storage.type** and **gpfs.replica.enforced** are defined in the gpfs-site.xml file.
- Starting from HDFS Transparency version 3.1.1-1, the default value for **dfs.replication** is *3* in hdfs-site.xml and **gpfs.replica.enforced** is *gpfs* in gpfs-site.xml.
- The **dfs.replication** value should be smaller or equal to the DataNode count.

Update environment variables for HDFS transparency service

Use the following procedure to update the environment variables for HDFS transparency service.

The administrator might need to update some environment variables for the HDFS Transparency service. For example, change JVM options or Hadoop environment variables like **HADOOP_LOG_DIR**.

To update this, follow these steps:

1. On the HDFS Transparency NameNode, modify the /usr/lpp/mmfs/hadoop/etc/hadoop/hadoop-env.sh (for HDFS Transparency 2.7.3-x) or /var/mmfs/hadoop/etc/hadoop/hadoop-env.sh (for HDFS Transparency 3.0.0+) and other files as necessary.
2. Sync the changes to all the HDFS Transparency nodes. For information on synching the HDFS Transparency configurations, refer "Sync HDFS Transparency configurations" on page 61.

Sync HDFS Transparency configurations

Usually, all the HDFS Transparency nodes take the same configurations. So, if you change the configurations of HDFS Transparency on one node, you need to run /usr/lpp/mmfs/bin/mmhadoopctl on the node to sync the changed configurations into all other HDFS Transparency nodes.

For example, *hdfs_transparency_node1* is the node where you update your HDFS Transparency configurations:

For HDFS Transparency 2.7.3-x:

```
hdfs_transparency_node1#/usr/lpp/mmfs/hadoop/sbin/mmhadoopctl connector syncconf /usr/lpp/mmfs/
hadoop/etc/hadoop
```

For HDFS Transparency 3.0.0-x ~ 3.1.0-x:

```
hdfs_transparency_node1#/usr/lpp/mmfs/hadoop/sbin/mmhadoopctl connector syncconf /var/mmfs/
hadoop/etc/hadoop
```

For HDFS Transparency 3.1.1-x +:

```
hdfs_transparency_node1#/usr/lpp/mmfs/hadoop/sbin/mmhdfs config upload
```

Note: If you are using HDP with Mpack 2.4.2.1 or later, ensure you change configurations through Ambari only. If you change configurations by using **mmhadoopctl syncconf**, the changes get overwritten by Ambari integration after HDFS service restart.

Cluster and file system information configuration

After HDFS Transparency is started successfully for the first time, it executes a script called `initmap.sh` to automatically generate internal configuration files which contain the GPFS cluster information, disk-to-hostname map information and ip-to-hostname map information.

Table 12. Generating internal configuration files

HDFS Transparency version	**Generating internal configuration files**
HDFS Transparency 2.7.3-0 and earlier	If new disks are added in the file system or if the file systems are recreated, the `initmap.sh` script must be executed manually on the HDFS Transparency NameNode so that internal configuration files are updated with the new information.
HDFS Transparency 2.7.3-1 and later	The NameNode will run the `initmap.sh` script every time it starts so the script does not need to be run manually.
HDFS Transparency 2.7.3-2 and later	The internal configuration files will be generated automatically if they are not detected and will be synched to all other HDFS Transparency nodes when HDFS Transparency is started.

Table 13. initmap.sh script command syntax

HDFS Transparency version	**initmap.sh script command syntax**	
HDFS Transparency 2.7.3-1 and earlier	`/usr/lpp/mmfs/hadoop/sbin/initmap.sh <fsName> diskinfo nodeinfo clusterinfo`	
HDFS Transparency 2.7.3-2	`/usr/lpp/mmfs/hadoop/sbin/initmap.sh true <fsName> diskinfo nodeinfo clusterinfo`	
HDFS Transparency 2.7.3-3+ and 3.0.0+	Local cluster mode	`/usr/lpp/mmfs/hadoop/sbin/initmap.sh -d -i all [fsName]`
	Remote Mount mode	`/usr/lpp/mmfs/hadoop/sbin/initmap.sh -d -r -u [gpfs.ssh.user] -i all [fsName]`

Note:

- **`-d`** option propagates the generated files to all the nodes. This option should only be used when running on the NameNode.
- **`-r`** option requests to run the necessary commands on the contact nodes to generate the config files if this is a remote mounted file system.
- **`-u [gpfs.ssh.user]`** option is not necessary if **`gpfs.ssh.user`** is not set in `/usr/lpp/mmfs/Hadoop/etc/hadoop/gpfs-site.xml` (HDFS Transparency 2.7.3.3) or in `/var/mmfs/hadoop/etc/hadoop/gpfs-site.xml` (HDFS Transparency 3.0.0+).
- **`-i all`** option generates all the required configuration files.

Table 14. Internal configuration files and location information

HDFS Transparency version	Internal configuration files and location
HDFS Transparency 2.7.3-0 and earlier	Generated internal configuration files diskid2hostname, nodeid2hostname and clusterinfo4hdfs are under the /var/mmfs/etc/ directory.
HDFS Transparency 2.7.3-1 and later	Generated internal configuration files diskid2hostname.<fs-name>, nodeid2hostname.<fs-name> and clusterinfo4hdfs.<fs-name> are under the /var/mmfs/etc/hadoop for HDFS Transparency 2.7.3-x or under /var/mmfs/hadoop/init for HDFS Transparency 3.0.0+.

The following are examples of the internal configuration files:

```
# pwd
/var/mmfs/hadoop/init

# cat clusterinfo4hdfs.latestgpfs
clusterInfo:test01.gpfs.net:8833372820164647001
fsInfo:1:2:1048576:8388608
remoteInfo:gpfs0:ess01-dat.gpfs.net,ess02-dat.gpfs.net

# cat diskid2hostname.latestgpfs
1:ess01-dat.gpfs.net:30.1.1.11
2:ess01-dat.gpfs.net:30.1.1.11
3:ess01-dat.gpfs.net:30.1.1.11
4:ess01-dat.gpfs.net:30.1.1.11
5:ess02-dat.gpfs.net:30.1.1.12
6:ess02-dat.gpfs.net:30.1.1.12
7:ess02-dat.gpfs.net:30.1.1.12
8:ess02-dat.gpfs.net:30.1.1.12

# cat nodeid2hostname.latestgpfs
1:ess01-dat.gpfs.net:30.1.1.11
2:ess02-dat.gpfs.net:30.1.1.12
```

Note:

- The internal configuration files can be removed from all the nodes and regenerated when the HDFS Transparency is restarted or when the initmap.sh command is executed depending on the HDFS Transparency version you are at. If the internal configuration files are missing, HDFS transparency re-runs the script and will take longer to start.
- If only stopping and restarting the DataNode hits an error because the initmap files are outdated and failed to refresh configurations, on the DataNode, use the touch command on the initmap files so that the modification times are updated and it can come up properly. See Table 14 on page 63 for the initmap config file locations.

HDFS auditing

By default, the HDFS audit logs are not enabled in HDFS Transparency.

To enable the log4j-based HDFS audit logs, perform the following:

1. Log in to one of the CES HDFS NameNode hosts and change the value of **HDFS_AUDIT_LOGGER** from *INFO,NullAppender* to *INFO,RFAAUDIT* in the /var/mmfs/hadoop/etc/hadoop/hadoop-env.sh file.
2. Upload the configurations to IBM Storage Scale CCR by running the following command:

   ```
   /usr/lpp/mmfs/hadoop/sbin/mmhdfs config upload
   ```

3. Start the HDFS Transparency services.

4. The HDFS audit logs will be created in the default log directory location (/var/log/transparency) of HDFS Transparency.

   If you want to change the default location for the audit logs, update the **HADOOP_LOG_DIR** parameter in hadoop-env.sh to point to the new directory.

   For example:

   ```
   export HADOOP_LOG_DIR=/audit/logs
   ```

Administering

Different configurations, features, and tools are supported for managing, monitoring, or automating HDFS Transparency on IBM Storage Scale.

Managing HDFS Transparency cluster

Operations to administer and manage an HDFS Transparency cluster and its nodes.

Prerequisites for Kerberos

This topic lists the prerequisites for administering a Kerberos enabled CES HDFS cluster.

Note: Only MIT Kerberos is supported.

If you are adding a new NameNode or DataNode, execute step 1 and step 2. For all other administrative operations, go to step 3.

1. On the new node, create the Hadoop users and groups by following the instructions in "Configuring users, groups and file system access for IBM Storage Scale" on page 302.
2. Initialize Kerberos on the new node by running the Kerberos configuration script /usr/lpp/mmfs/hadoop/scripts/gpfs_create_hadoop_users_dirs.py as mentioned in "Configuring Kerberos using the Kerberos script provided with IBM Storage Scale" on page 117. This will create the principals and keytabs specific to the new node.
3. Obtain a Kerberos token for the hdfs user to administer CES HDFS when using either the installation toolkit method or the manual method. Run the following command:

   ```
   # kinit -kt /etc/security/keytab/hdfs.headless.keytab hdfs@<Realm Name>
   ```

   Note: The previous command needs to be executed on all the CES HDFS NameNodes.
4. Verify that there is a valid token by running the following command:

   ```
   # klist
   ```

Listing CES HDFS IPs

The **mmces address list** command displays all the currently configured protocol IPs.

For more information, see the *Protocol node IP further configuration* topic in the *IBM Storage Scale: Concepts, Planning, and Installation Guide*.

To list the CES HDFS protocol IPs, run the **mmces address list** command on a CES HDFS NameNode as root:

```
 root@c902f09x09# mmces address list

Address             Node                   Ces Group          Attributes
-------------   ---------------------    ---------------     ------------
192.0.2.5        c902f09x09.gpfs.net      hdfsmycluster1      hdfsmycluster1
192.0.2.6        c902f09x11.gpfs.net      hdfsmycluster2      hdfsmycluster2
192.0.2.7        c902f09x10.gpfs.net      none                none
192.0.2.8        c902f09x12.gpfs.net      none                none
```

Note: The CES group and Attributes values contain the HDFS cluster name prefixed with hdfs.

This example uses the CES Cluster name/group *mycluster1* for the first HDFS Transparency cluster and *mycluster2* for the second HDFS Transparency cluster where 192.0.2.5 and 192.0.2.6 CES IPs belong to mycluster1 and mycluster2 HDFS Transparency clusters respectively.

For installation toolkit, the **`spectrumscale config hdfs new -n <CLUSTER_NAME>`** command automatically adds the hdfs prefix to the input CLUSTER_NAME which then becomes the CES Group and Attribute values.

For example, **`spectrumscale config hdfs new -n mycluster1`**.

For more information, see "Adding CES HDFS nodes into the centralized file system" on page 34 or "Separate CES HDFS cluster remote mount into the centralized file system" on page 38.

For manual installation, you need to manually set the hdfs prefix.

Run **`mmchnode --ces-enable --ces-group hdfsmycluster1`** command to set the CES Cluster Name/Group with hdfs prefix.

Setting CES HDFS configuration files

This section describes the configuration files settings that will be changed in the "Enable and Configure CES HDFS" on page 42 section when using the **`mmhdfs`** command while you are manually trying to setup the CES HDFS cluster.

Before enabling HDFS Transparency, some configuration must be set. Some of them can be done automatically and some must be set manually.

Edit config fields

Use the following command on one CES transparency node to edit the config fields locally one at a time. After modifying the config fields, ensure that you upload to CCR (See Edit config files and upload section):

```
mmhdfs config set [config file] -k [key1=value] -k [key2=value] ... -k [keyX-value]
```

Edit config files and upload

Use the following command on one CES transparency node to download the configuration files, edit them and then upload the changes into CCR:

```
mmhdfs config import/export [a local config dir] [config_file1,config_file2,...]

mmhdfs config upload
```

Configuration file settings

The following configurations should be set to proper value to support CES IP failover:

For hadoop_env.sh:

JAVA_HOME: Set the correct java home path for the node.

For hdfs-site.xml:

- **`dfs.nameservices`**: Set to the logical name of the cluster. This must be equal to the CES group name without the hdfs prefix.

  In the following example, we use hdfscluster as the CES group name where hdfs is the prefix, and cluster is the cluster name:

  ```
  <property>
    <name>dfs.nameservices</name>
    <value>cluster</value>
  </property>
  ```

- **`dfs.ha.namenodes.[nameservice ID]`**: Set to a list of comma-separated NameNode IDs.

  For example:

```
<property>
  <name>dfs.ha.namenodes.cluster</name>
  <value>nn1,nn2</value>
</property>
```

If there is only one NameNode (Only one CES node which means no CES HA) the list should contain only one ID.

For example:

```
<property>
  <name>dfs.ha.namenodes.cluster</name>
  <value>nn1</value>
</property>
```

- **dfs.namenode.rpc-address.[nameservice ID].[namenode ID]**: Set to the fully qualified RPC address for each NameNode to listen on.

  For example:

```
<property>
  <name>dfs.namenode.rpc-address.cluster.nn1</name>
  <value>machine1.example.com:8020</value>
</property>
<property>
  <name>dfs.namenode.rpc-address.cluster.nn2</name>
  <value>machine2.example.com:8020</value>
</property>
```

- **dfs.namenode.http-address.[nameservice ID].[namenode ID]**: Set to the fully qualified HTTP address for each NameNode to listen on.

  For example:

```
<property>
  <name>dfs.namenode.http-address.hdfscluster.nn1</name>
  <value>machine1.example.com:50070</value>
</property>
<property>
  <name>dfs.namenode.http-address.hdfscluster.nn2</name>
  <value>machine2.example.com:50070</value>
</property>
```

- **dfs.namenode.shared.edits.dir**: Set to a directory which will be used to store shared editlogs for this HDFS HA cluster. The recommendation is to use a name like HA-[dfs.nameservices].

  For example:

```
<property>
  <name>dfs.namenode.shared.edits.dir</name>
  <value>file:///gpfs/HA-cluster</value>
</property>
```

  Note: If there is only one NameNode (Only one CES node which means no CES HA), do not set this property. Otherwise, NameNode will fail to start. The NameNode shared edit dir is used for HA.

- **dfs.client.failover.proxy.provider. [nameservice ID]**

```
<property>
   <name>dfs.client.failover.proxy.provider.cluster</name>
<value>org.apache.hadoop.hdfs.server.namenode.ha.ConfiguredFailoverProxyProvider</value>
</property>
```

- **dfs.namenode.rpc-bind-host**: This should be set to 0.0.0.0.

  For example:

```
<property>
   <name>dfs.namenode.rpc-bind-host</name>
   <value>0.0.0.0</value>
</property>
```

- **`dfs.namenode.servicerpc-bind-host`**: This should be set to 0.0.0.0.

  For example:

  ```
  <property>
     <name>dfs.namenode.servicerpc-bind-host</name>
     <value>0.0.0.0</value>
  </property>
  ```

- **`dfs.namenode.lifeline.rpc-bind-host`**: This should be set to 0.0.0.0.

  For example:

  ```
  <property>
     <name>dfs.namenode.lifeline.rpc-bind-host</name>
     <value>0.0.0.0</value>
  </property>
  ```

- **`dfs.namenode.http-bind-host`**: This should be set to 0.0.0.0.

  For example:

  ```
  <property>
     <name>dfs.namenode.http-bind-host</name>
     <value>0.0.0.0</value>
  </property>
  ```

For core-site.xml:

`fs.defaultFS`: This should be set to the value of the **`dfs.nameservices`**. For CES HDFS, this must be the CES HDFS group name without the hdfs prefix.

For example:

```
</property>
  <name>fs.defaultFS</name>
  <value>hdfs://cluster</value>
</property>
```

Follow the "Enable and Configure CES HDFS" on page 42 section to set the configuration values for non-HA and HA CES HDFS Transparency cluster.

Change CES HDFS NON-HA cluster into CES HDFS HA cluster

Changing CES HDFS Non-HA cluster into CES HDFS HA cluster using install toolkit

This topic lists the steps to change CES HDFS Non-HA cluster into CES HDFS HA cluster using install toolkit.

To add another NameNode to an existing CES HDFS cluster and to set it to HA configuration, follow the steps below:

1. If the new NameNode is not already a CES node, add it as a protocol node:

   a. Add the new NameNode by running the following command:

   ```
   /spectrumscale node add NAMENODE -p
   ```

   b. Before you initiate the installation procedure for the new NameNode, run the following command to perform the environment checks:

   ```
   /spectrumscale install -pr
   ```

   c. To set up the new NameNode, run the following command:

   ```
   /spectrumscale install
   ```

2. Add the new NameNode into the HDFS cluster by running the following command:

```
/spectrumscale config hdfs add -n CLUSTER_NAME -nn NAMENODE
```

3. Disable HDFS by running the following command:

   ```
   /usr/lpp/mmfs/bin/mmces service disable hdfs
   ```

4. Before you initiate the deployment, run the following command to perform the environment checks:

   ```
   /spectrumscale deploy -pr
   ```

5. To deploy the new configuration, run the following command:

   ```
   /spectrumscale deploy
   ```

   Note: After the deployment completes, HDFS is automatically enabled.

Manually change CES HDFS NON-HA cluster into CES HDFS HA cluster

This topic lists the steps to manually change CES HDFS NON-HA cluster into CES HDFS HA cluster.

1. If the new NameNode is already a part of your IBM Storage Scale cluster, go to the next step. Otherwise, install IBM Storage Scale on that node by following "Steps for manual installation" on page 33. Then add the new nodes into the existing IBM Storage Scale cluster by running the following command:

   ```
   mmaddnode -N
   ```

2. Log in to the new NameNode as root and install the HDFS Transparency package into the new NameNode. Issue the following command on RHEL:

   ```
   # rpm -ivh gpfs.hdfs-protocol-<version>.<arch>.rpm
   ```

3. Stop the existing HDFS Transparency NameNode.

   ```
   mmces service stop hdfs
   ```

4. Enable CES on the new NameNode by giving the same CES group name as that of the existing HDFS cluster.

   ```
   mmchnode --ces-enable --ces-group hdfscluster -N c16f1n08
   ```

5. For the new NameNode, add related property into the `hdfs-site.xml`.

   For example, the existing HDFS NON-HA cluster NameNode is c16f1n07, add another NameNode c16f1n08 to the cluster.

   ```
   mmhdfs config set hdfs-site.xml -k
   dfs.namenode.shared.edits.dir=file:///gpfs/HA-cluster -k
   dfs.ha.namenodes.cluster=nn1,nn2 -k
   dfs.namenode.rpc-address.cluster.nn2=c16f1n08.gpfs.net:8020 -k
   dfs.namenode.http-address.cluster.nn2=c16f1n08.gpfs.net:50070
   ```

6. Upload the configuration into CCR.

   ```
   mmhdfs config upload
   ```

7. Initialize the NameNode shared directory to store HDFS cluster HA info from one NameNode.

   ```
   /usr/lpp/mmfs/hadoop/bin/hdfs namenode -initializeSharedEdits
   ```

8. Start the existing HDFS Transparency NameNode.

   ```
   mmces service start hdfs -N c16f1n07,c16f1n08
   ```

   or

```
mmces service start hdfs -a
```

9. If the CES HDFS cluster is Kerberos enabled, ensure that you configure Kerberos for the new NameNode by following "Setting up Kerberos for HDFS Transparency nodes" on page 109.
10. Check the status of the added NameNodes in the HDFS cluster.

    ```
    mmces service list -a
    ```

11. Check the status of both the NameNodes. One should be Active and the other should be in standby.

    ```
    /usr/lpp/mmfs/hadoop/bin/hdfs haadmin -getAllServiceState
    ```

12. Restart DataNodes to take effect.

    ```
    mmhdfs hdfs-dn restart
    ```

13. Check the status of the DataNodes.

    ```
    mmhdfs hdfs-dn status
    ```

Setting configuration options in CES HDFS

This section lists the steps to set the configuration options in the CES HDFS.

To set configurations in the CES HDFS environment, run the following steps:

1. Stop HDFS Transparency.
2. Get the configuration file that you want to change.
3. Update the configuration file.
4. Import the file to CES HDFS.
5. Upload the changes to CES HDFS.
6. Start HDFS Transparency.

Setting up the `gpfs.ranger.enabled` field

From HDFS Transparency 3.1.1-3, ensure that the **`gpfs.ranger.enabled`** field is set to *scale*. The `scale` option replaces the original *true*/*false* values.

1. Stop HDFS Transparency.

   If you are using CDP Private Cloud Base, stop HDFS Transparency from the Cloudera Manager GUI. Otherwise, on the CES HDFS Transparency, run the following:

   ```
   /usr/lpp/mmfs/bin/mmces service stop hdfs -a
   /usr/lpp/mmfs/hadoop/sbin/mmhdfs hdfs-dn stop
   ```

2. After HDFS Transparency has completely stopped, on the CES HDFS node, run the following command:

   ```
   /usr/lpp/mmfs/hadoop/sbin/mmhdfs hdfs status
   ```

3. Update the HDFS Transparency configuration files and upload the changes. Get the config files by running the following commands:

   ```
   mkdir /tmp/hdfsconf
   /usr/lpp/mmfs/hadoop/sbin/mmhdfs config export /tmp/hdfsconf   gpfs-site.xml
   cd /tmp/hdfsconf/
   ```

4. Update the config files in `/tmp/hdfsconf` with the following changes:

   ```
   <property>
   <name>gpfs.ranger.enabled</name>
   <value>scale</value>
   ```

```
<final>false</final>
</property>
```

Note: From HDFS Transparency 3.1.0-6 and 3.1.1-3, ensure that the **`gpfs.ranger.enabled`** field is set to *scale*. The `scale` option replaces the original *true*/*false* values.

5. Import the files into the CES HDFS cluster by running the following command:

   ```
   /usr/lpp/mmfs/hadoop/sbin/mmhdfs config import /tmp/hdfsconf gpfs-site.xml
   ```

6. Upload the changes to the CES HDFS cluster by running the following command:

   ```
   /usr/lpp/mmfs/hadoop/sbin/mmhdfs config upload
   ```

7. Start HDFS Transparency.

   If you are using CDP Private Cloud Base, start HDFS Transparency from the Cloudera Manager GUI. Click **IBM Spectrum Scale** > **Actions** > **Start**.

   Otherwise, on the CES HDFS Transparency node, run the following:

   ```
   /usr/lpp/mmfs/bin/mmces service start hdfs -a
   /usr/lpp/mmfs/hadoop/sbin/mmhdfs hdfs-dn start
   ```

8. After HDFS Transparency has completely started, on the CES HDFS node, run the following command:

   ```
   /usr/lpp/mmfs/hadoop/sbin/mmhdfs hdfs status
   ```

Setting the Java heap size for NameNode/DataNode

HDFS Transparency does not set the Java heap size value in `hadoop_env.sh` for NameNode or DataNode. Therefore, the JVM autoscales based on the machine memory size.

If you need to set the Java heap size, perform the following:

1. Stop HDFS Transparency.
2. Ensure that HDFS Transparency has stopped by running the following command:

   ```
   mmhdfs hdfs status
   ```

3. Get the config file by running the following command:

   ```
   mkdir /tmp/hdfsconf /usr/lpp/mmfs/hadoop/sbin/mmhdfs config export /tmp/hdfsconf
   hadoop_env.sh cd /tmp/hdfsconf
   ```

4. In `/tmp/hdfsconf`, update the `hadoop_env.sh` to set the **`-Xmx`** and **`-Xms`** options for HDFS_NAMENODE_OPTS and/or HDFS_DATANODE_OPTS.

   For example:

   ```
   SHARED_HDFS_NAMENODE_OPTS=“-server -XX:ParallelGCThreads=8 -XX:+UseConcMarkSweepGC
   -XX:ErrorFile=/var/log/hadoop/$USER/hs_err_pid%p.log -XX:NewSize=1248m -XX:MaxNewSize=1248m
   -Xloggc:/var/log/hadoop/$USER/gc.log-`date +‘%Y%m%d%H%M’` -verbose:gc -XX:+PrintGCDetails
   -XX:+PrintGCTimeStamps -XX:+PrintGCDateStamps -XX:CMSInitiatingOccupancyFraction=70
   -XX:+UseCMSInitiatingOccupancyOnly -Xms9984m -Xmx9984m -Dhadoop.security.logger=INFO,DRFAS
   -Dhdfs.audit.logger=INFO,DRFAAUDIT”

   export HDFS_NAMENODE_OPTS=“${SHARED_HDFS_NAMENODE_OPTS}
   -XX:OnOutOfMemoryError=\“/usr/hdp/current/hadoop-hdfs-namenode/bin/kill-name-node\”
   -Dorg.mortbay.jetty.Request.maxFormContentSize=-1 ${HDFS_NAMENODE_OPTS}”

   export HDFS_DATANODE_OPTS=“-server -XX:ParallelGCThreads=4 -XX:+UseConcMarkSweepGC
   -XX:OnOutOfMemoryError=\“/usr/hdp/current/hadoop-hdfs-datanode/bin/kill-data-node\”
   -XX:ErrorFile=/var/log/hadoop/$USER/hs_err_pid%p.log -XX:NewSize=200m -XX:MaxNewSize=200m
   -Xloggc:/var/log/hadoop/$USER/gc.log-`date +‘%Y%m%d%H%M’` -verbose:gc -XX:+PrintGCDetails
   -XX:+PrintGCTimeStamps -XX:+PrintGCDateStamps -Xms1024m -Xmx1024m
   -Dhadoop.security.logger=INFO,DRFAS
   -Dhdfs.audit.logger=INFO,DRFAAUDIT ${HDFS_DATANODE_OPTS}
   -XX:CMSInitiatingOccupancyFraction=70 -XX:+UseCMSInitiatingOccupancyOnly”
   ```

Note: You can set the **-Xmx** and **-Xms** options directly in the HDFS_NAMENODE_OPTS and HDFS_DATANODE_OPTS export options.

5. Import the files into the CES HDFS cluster.
6. Upload the changes to the CES HDFS cluster.
7. Start HDFS Transparency.
8. Check the status of the HDFS Transparency cluster by running the following command:

   ```
   mmhdfs hdfs status
   ```

Enabling and disabling CES HDFS

This section lists the steps to enable and disable CES HDFS.

CES HDFS NameNodes are CES protocol nodes.

Enabling CES HDFS

The following steps are relevant only when CES HDFS is disabled and you want to re-enable CES HDFS.

To enable CES HDFS, run the following steps:

1. Check the CES information by running the following commands:

   ```
   # mmces node list
   # mmces service list
   # mmces address list
   ```

2. Reload the existing HDFS configuration by running the following command:

   ```
   # mmhdfs config upload
   ```

3. Enable HDFS by running the following command:

   ```
   # mmces service enable hdfs
   ```

   Note: Running this command will start the NameNodes.
4. Start the DataNodes by running the following command:

   ```
   # mmhdfs hdfs-dn start
   ```

5. Verify the status of CES HDFS by running the following command:

   ```
   # mmces node list
   # mmces service list
   # mmces address list
   # mmhdfs hdfs status
   ```

Disabling CES HDFS

The following steps are relevant only when CES HDFS is enabled and you want to disable CES HDFS.

Note that running the following steps will not delete but only disable the CES HDFS protocol.

To disable CES HDFS, run the following steps:

1. Check the CES information by running the following command:

   ```
   # mmces node list
   # mmces service list
   # mmces address list
   ```

2. Disable CES HDFS by running the following command:

   ```
   # mmces service disable hdfs
   ```

Note: Running this command will stop the NameNodes.

3. Stop the DataNodes by running the following command:

   ```
   # mmhdfs hdfs-dn stop
   ```

4. Verify the status of CES HDFS by running the following command:

   ```
   # mmces service list
   # mmces address list
   # mmces node list
   ```

   Note: In the output of the **`mmces node list`** command, the `Node Flags` column might be set to *Failed*. This output occurs because HDFS is disabled.

Removing a NameNode from existing HDFS HA cluster

Removing a NameNode from existing HDFS HA cluster using install toolkit

Removing the NameNodes and DataNodes using the install toolkit is not supported.

Manually remove a NameNode from existing HDFS HA cluster

This topic lists the steps to manually remove a NameNode from existing HDFS HA cluster.

1. Stop the existing HDFS Transparency NameNodes.

   ```
   mmces service stop hdfs -N c16f1n07,c16f1n08
   ```

   or

   ```
   mmces service stop hdfs -a
   ```

2. Disable the CES HDFS services on the NameNode that you want to remove.

   ```
   mmchnode --ces-disable -N c16f1n08
   ```

3. Remove the NameNode related property from the hdfs-site.xml.

   For example, the existing HDFS HA cluster NameNode is c16f1n07(nn1) and c16f1n08(nn2). The NameNode that will be removed is c16f1n08.

   ```
   mmhdfs config del hdfs-site.xml -k
   dfs.namenode.shared.edits.dir -k
   dfs.namenode.rpc-address.cluster.nn2 -k
   dfs.namenode.http-address.cluster.nn2
   ```

   ```
   mmhdfs config set hdfs-site.xml -k dfs.ha.namenodes.cluster=nn1
   ```

4. Upload the configuration into CCR.

   ```
   mmhdfs config upload
   ```

5. Start the existing HDFS Transparency NameNode.

   ```
   mmces service start hdfs
   ```

6. Check the HDFS NameNode status in the existing HDFS cluster.

   ```
   mmces service list -a
   ```

7. Restart DataNodes to take effective.

   ```
   mmhdfs hdfs-dn restart
   ```

8. Check DataNodes status.

```
mmhdfs hdfs-dn status
```

Adding a new HDFS cluster into existing HDFS cluster on the same GPFS cluster (Multiple HDFS clusters)

This section describes how to add a new HDFS Transparency cluster onto the same GPFS cluster that already has an existing HDFS Transparency cluster. This will create multiple HDFS clusters onto the same GPFS cluster.

Adding a new HDFS cluster into existing HDFS cluster on the same GPFS cluster using install toolkit

The “Using installation toolkit” on page 34 section describes how to add in a new HDFS cluster into the environment.

The difference when creating another HDFS cluster into an existing HDFS cluster on the same GPFS cluster is to create a different cluster name for the new HDFS cluster.

For example, use CLUSTER2 as the cluster name for the second HDFS cluster to be added into the existing 1st HDFS cluster:

1. Add the new 2nd HDFS cluster nodes into the GPFS cluster.

   Ensure that the nodes are new nodes and not a part of the existing HDFS cluster.

   ```
   # NameNodes (Protocol node)
   ./spectrumscale node add c902f09x01.gpfs.net -p
   ./spectrumscale node add c902f09x02.gpfs.net -p

   # DataNodes
   ./spectrumscale node add c902f09x03.gpfs.net
   ./spectrumscale node add c902f09x04.gpfs.net
   ./spectrumscale node add c902f09x05.gpfs.net
   ./spectrumscale node add c902f09x06.gpfs.net
   ```

2. Configure the 2nd cluster CES HDFS cluster.

   ```
   ./spectrumscale config hdfs new -n CLUSTER2 -nn NAMENODES -dn DATANODES -f FILESYSTEM -d
   DATADIR
   ./spectrumscale config hdfs new -n CLUSTER2 -nn c902f09x01.gpfs.net, c902f09x01.gpfs.net -dn
   c902f09x03.gpfs.net, c902f09x04.gpfs.net, c902f09x05.gpfs.net, c902f09x06.gpfs.net -f gpfs
   -d gpfshdfs2
   ```

3. Deploy the 2nd cluster.

   ```
   ./spectrumscale deploy -pr
   ./spectrumscale deploy
   ```

   Note:

   - Ensure that there are sufficient free CES-IPs available for usage.
   - Ensure that the new cluster NameNodes and DataNodes are not the same nodes as the existing HDFS cluster.
   - Ensure that the DATADIR is unique to host the second cluster’s data.

Manually adding a new HDFS cluster into existing HDFS cluster on the same GPFS cluster (Multiple HDFS clusters)

This topic lists the steps to manually add a new HDFS cluster into existing HDFS cluster on the same GPFS cluster (Multiple HDFS clusters).

1. Create different CES groups for different HDFS clusters and ensure that the existing HDFS cluster nodes are different than the new HDFS cluster nodes that will be added.

2. For the first HDFS cluster, HDFS configuration including core-site.xml, hdfs-site.xml, gpfs-site.xml and hadoop-env.sh must be executed on the NameNode belonging to that cluster. Run the following command to upload the configuration into CCR:

   ```
   /usr/lpp/mmfs/hadoop/sbin/mmhdfs config upload
   ```

3. Start the NameNodes service of the first HDFS cluster.

   ```
   mmces service start hdfs -a
   ```

4. Check that the NameNodes service status is running for the first HDFS cluster.

   ```
   mmces service list -a
   ```

5. For the second HDFS cluster, HDFS configuration including core-site.xml, hdfs-site.xml, gpfs-site.xml and hadoop-env.sh must be executed on the NameNode belonging to the second cluster.

   The value of **`dfs.nameservices`** should be set to the cluster name of the second HDFS cluster. Run the following command to upload the configuration into CCR. The configuration uploaded will be pushed to all the NameNodes of the new HDFS cluster.

   ```
   /usr/lpp/mmfs/hadoop/sbin/mmhdfs config upload
   ```

6. Enable CES for the NameNodes of the new HDFS cluster and set the CES group name to the CES group name of the new HDFS cluster.

   ```
   mmchnode --ces-enable --ces-group=[groupname_addedhdfscluster] -N
   NewClusterNameNode1,NewClusterNameNode2
   ```

7. Start the NameNodes service of the new HDFS cluster, if not started already.

   ```
   mmces service start hdfs -a
   ```

8. Check if the NameNodes service status is running for the new HDFS cluster.

   ```
   mmces service list -a
   ```

9. Log in to one of the newly added NameNodes and start the DataNodes of the new HDFS cluster.

   ```
   mmhdfs hdfs-dn start
   ```

Removing an existing HDFS cluster from multiple HDFS clusters of the same GPFS cluster

Removing an existing HDFS cluster from multiple HDFS clusters of the same GPFS cluster using install toolkit

Removing an existing HDFS cluster is not supported by the installation toolkit command line interface.

Manually remove an existing HDFS cluster from multiple HDFS clusters of the same GPFS cluster

This topic lists the steps to manually remove an existing HDFS Cluster from multiple HDFS clusters of the same GPFS cluster.

1. Stop all the Hadoop services and CES HDFS services.

   ```
   mmces service stop HDFS -N nn1,nn2
   mmhdfs hdfs-dn stop
   ```

2. Disable the CES HDFS service on the NameNodes of the removing HDFS cluster to stop the NameNode service.

```
mmchnode --noces-group [groupname_removedhdfscluster] -N removeNameNode1,removeNameNode2
```

3. Remove the configuration files from CCR. The [clustername] should be the value of **dfs.nameservices** that corresponds to the CES group name or hostname of the corresponding CES IP.

   ```
   mmccr fdel [clustername].tar
   ```

4. Check that the removed NameNode service is not running in the existing HDFS cluster.

   ```
   mmces service list -a
   ```

Adding DataNodes using installation toolkit

The CES HDFS NameNodes and DataNodes do not need to be stopped when adding or deleting DataNodes from the cluster.

The following are the two ways to add DataNodes into an existing CES HDFS cluster:

- Add new DataNodes into an existing CES HDFS cluster.
- Add existing IBM Storage Scale nodes that are already a part of the IBM Storage Scale cluster as new DataNodes, to the CES HDFS cluster.

Adding new DataNodes into an existing CES HDFS cluster

1. On the new nodes, ensure that the prerequisites are installed for the node to be able to be deployed by the installation toolkit.

   For more information on basic IBM Storage Scale requirements, see "Installation prerequisites" on page 30.

2. Log into the existing CES HDFS cluster installer node as root and change to the installer directory to run the **spectrumscale** commands.

   For IBM Storage Scale 5.1.1 and later:

   ```
   # cd /usr/lpp/mmfs/5.1.1.0/ansible-toolkit
   ```

   For IBM Storage Scale 5.1.0 and earlier:

   ```
   # cd /usr/lpp/mmfs/5.0.4.2/installer
   ```

3. Check the current CES HDFS cluster host information.

   ```
   # ./spectrumscale config hdfs list
   ```

4. Add the new nodes (DataNodes) into an IBM Storage Scale cluster.

   ```
   # ./spectrumscale node add <hostname>
   ```

5. Perform environment checks before initiating the installation procedure.

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

   Start the IBM Storage Scale installation and add the nodes into the existing cluster.

   ```
   # ./spectrumscale install
   ```

6. To add a new DataNode into an existing CES HDFS cluster, run the following command:

   ```
   # ./spectrumscale config hdfs add -n <Existing HDFS cluster name> -dn <new_DataNode_hostname>
   ```

7. Check the CES HDFS host list to ensure that the new hosts have been added.

   ```
   # ./spectrumscale config hdfs list
   ```

8. Perform environment checks before initiating the deployment procedure.

   ```
   # ./spectrumscale deploy -pr
   ```

9. Start the IBM Storage Scale installation and the creation of the new CES HDFS nodes.

   ```
   # ./spectrumscale deploy
   ```

Adding the existing IBM Storage Scale nodes to an existing CES HDFS cluster

1. Log in to the existing CES HDFS cluster installer node and change to the installer directory to run the **spectrumscale** commands.

   For IBM Storage Scale 5.1.1 and later:

   ```
   # cd /usr/lpp/mmfs/5.1.1.0/ansible-toolkit
   ```

   For IBM Storage Scale 5.1.0 and earlier:

   ```
   # cd /usr/lpp/mmfs/5.0.4.2/installer
   ```

2. Check the current CES HDFS cluster host information.

   ```
   # ./spectrumscale config hdfs list
   ```

3. Add a new DataNode into an existing CES HDFS cluster.

   ```
   # ./spectrumscale config hdfs add -n <Existing HDFS cluster name> -dn <new_DataNode_hostname>
   ```

4. Check the CES HDFS host list to ensure that the new hosts are added.

   ```
   # ./spectrumscale config hdfs list
   ```

5. Perform environment checks before initiating the deployment procedure.

   ```
   # ./spectrumscale deploy -pr
   ```

6. Start the IBM Storage Scale installation and the creation of the new CES HDFS nodes.

   ```
   # ./spectrumscale deploy
   ```

Adding DataNodes manually

The CES HDFS NameNodes and DataNodes do not need to be stopped when adding or deleting DataNodes from the cluster.

The following are the two ways to add DataNodes into an existing CES HDFS cluster:

- Add new DataNodes into an existing CES HDFS cluster.
- Add existing IBM Storage Scale nodes that are already a part of the IBM Storage Scale cluster as new DataNodes, to the CES HDFS cluster.

Adding new DataNodes into an existing CES HDFS cluster

1. If you have a new DataNode, install IBM Storage Scale by following the "Steps for manual installation" on page 33 topic and then add the new nodes into the existing IBM Storage Scale cluster by using the **mmaddnode -N** command.

   If you have existing IBM Storage Scale nodes that already have the IBM Storage Scale packages installed and configured, go to the next step.

2. Log in to the new DataNode as root.
3. Install HDFS Transparency package into the new DataNode.

On Red Hat Enterprise Linux, issue the following command:

```
   # rpm -ivh gpfs.hdfs-protocol-<version>.<arch>.rpm
```

4. On the NameNode as root, edit the worker configuration file to add in the new DataNode.

   ```
   # vi /var/mmfs/hadoop/etc/hadoop/workers
   ```

5. On the NameNode as root, upload the modified configuration.

   ```
   # mmhdfs config upload
   ```

6. On the NameNode as root, copy the `init` directory to the DataNode.

   ```
   # scp -r /var/mmfs/hadoop/init [datanode]:/var/mmfs/hadoop/
   ```

7. If the CES HDFS cluster is Kerberos enabled, ensure that you configure Kerberos for the new DataNode by following “Setting up Kerberos for HDFS Transparency nodes” on page 109.
8. On the DataNode as root, start the DataNode.

   ```
   # /usr/lpp/mmfs/hadoop/sbin/mmhdfs datanode start
   ```

9. On the NameNode, confirm if the DataNode is shown from the DataNode list with correct status by running the following command:

   ```
   # /usr/lpp/mmfs/hadoop/sbin/mmhdfs hdfs-dn status
   ```

Removing DataNodes manually

The CES HDFS NameNodes and DataNodes do not need to be stopped when adding or deleting DataNodes from the cluster.

To remove a DataNode from the CES HDFS cluster, perform the following steps:

1. On the DataNode as root, stop the DataNode service.

   ```
   # mmhdfs datanode stop
   ```

2. On the NameNode as root, edit the workers configuration file to remove the DataNode from the DataNode list.

   ```
   # vi /var/mmfs/hadoop/etc/hadoop/workers
   ```

3. On the NameNode as root, upload the modified configuration into CES.

   ```
   # mmhdfs config upload
   ```

4. On the NameNode, confirm that the DataNode is not shown from the DataNode list by running the following command:

   ```
   # mmhdfs hdfs-dn status
   ```

Note: The HDFS Transparency NameNodes must be restarted to fetch the information about the removed DataNode. Before this restart is completed, the removed DataNode is listed as "`dead datanode`" if you run the **`hdfs dfsadmin -report`** command.

Decommissioning DataNodes

This section lists the steps to decommission DataNodes.

To decommission a DataNode from the HDFS cluster, perform the following steps:

1. Modify the `dfs.exclude` file as specified under the **`dfs.hosts.exclude`** value in `hdfs-site.xml` by adding the nodes to be decommissioned.

   ```
   dfs.exclude file
   <hostname1>
   <hostname2>
   ```

2. Run the following command for the changes to take effect:

   ```
   hdfs dfsadmin -refreshNodes
   ```

3. Monitor the decommissioning process by running the following command:

   ```
   hdfs dfsadmin -report
   ```

   Note: When the DataNode is DECOMMISSIONED, you can stop the DataNode.

Frequently used commands

This section lists the commonly used commands and their options.

mmhdfs

- To check the status of the NameNodes and DataNodes in the HDFS Transparency cluster, run the following command:

  ```
  /usr/lpp/mmfs/hadoop/sbin/mmhdfs hdfs status
  ```

- To start the DataNode, run the following command:

  ```
  /usr/lpp/mmfs/hadoop/sbin/mmhdfs hdfs-dn start
  ```

- To stop the DataNode, run the following command:

  ```
  /usr/lpp/mmfs/hadoop/sbin/mmhdfs hdfs-dn stop
  ```

mmces

- To start the HDFS Transparency NameNodes, run the following command:

  ```
  /usr/lpp/mmfs/bin/mmces service start hdfs -a
  ```

  Note: Do not use the **`mmhdfs`** command.
- To stop the HDFS Transparency NameNodes, run the following command:

  ```
  /usr/lpp/mmfs/bin/mmces service stop hdfs -a
  ```

  Note: Do not use the **`mmhdfs`** command.
- To check the value of the CES HDFS protocol IPs, run the following command:

  ```
  /usr/lpp/mmfs/bin/mmces address list
  ```

- To verify the CES HDFS service, run the following command:

  ```
  /usr/lpp/mmfs/bin/mmces service list -a
  ```

mmhealth

- To show the health status of the node, run the following command:

  ```
  /usr/lpp/mmfs/bin/mmhealth node show
  ```

- To show the NameNode health status, run the following command on the NameNode:

  ```
  /usr/lpp/mmfs/bin/mmhealth node show HDFS_Namenode -v
  ```

- To show the DataNode health status, run the following command on the DataNode:

```
/usr/lpp/mmfs/bin/mmhealth node show HDFS_Datanode -v
```

hdfs

- To retrieve the status and check the state of all the HDFS NameNodes, run the following command:

```
/usr/lpp/mmfs/hadoop/bin/hdfs haadmin -getAllServiceState
```

- To manually trigger the HDFS Transparency cluster to failover to the standby NameNode, run the following commands:

```
/usr/lpp/mmfs/hadoop/bin/hdfs haadmin -failover nn1 nn2
/usr/lpp/mmfs/bin/mmces address move --ces-ip x.x.x.x --ces-node node.example.com
```

Note: Replace the **--ces-ip** parameter with the CES IP address of the HDFS Transparency cluster and the **--ces-node** parameter with the name of the new active NameNode.

For more information on **mmhdfs**, **mmces**, and **mmhealth** commands, see the respective command in the *IBM Storage Scale: Command and Programming Reference Guide* guide.

Monitoring HDFS Transparency status using the mmhealth command

The **mmhealth** command helps in monitoring the status of HDFS Transparency by using the PID.

To ensure that the **mmhealth** command picks up the correct HDFS Transparency PID, modify the **pidfilepath** and **slavesfile** fields in the `mmsysmonitor.conf` file to match your environment setup.

As root, perform the following steps on each HDFS Transparency node:

1. Under `/var/mmfs/mmsysmon/mmsysmonitor.conf` file, go to the [hadoopconnector] section. If the **pidfilepath** and **slavesfile** fields do not match your environment setup, modify these fields as follows:

```
[hadoopconnector]
monitorinterval = 30
monitoroffset = 0
clockalign = false
# Optional entries to override the current defaults in the HadoopConnector monitor
# Path to the PID file directory
# default is /var/run/hadoop/root
pidfilepath =                                          <---Edit this value and keep a blank
after the "=" sign before the actual value
# Path to hadoop binaries
# default is /usr/lpp/mmfs/hadoop/bin
binfilepath =
# Full path to the hadoop-env.shl file
# default is /var/mmfs/hadoop/etc/hadoop/hadoop-env.sh
envfile =
# Full path to the slaves file
# default is /var/mmfs/hadoop/etc/hadoop/slaves
slavesfile =                                           <---Edit this value if needed. The
filename can be "slaves" or "workers"
# Full path to the core-site.xml file
# default is /var/mmfs/hadoop/etc/hadoop/core-site.xml
coresitefile =
# Full path to the GPFS binary path (mmhadoopctl program)
# default is /usr/lpp/mmfs/bin
gpfsbinfilepath =
```

2. Restart the system health monitor by running the following command:

```
/usr/lpp/mmfs/bin/mmsysmoncontrol restart
```

For example:

a. Edit the `/var/mmfs/mmsysmon/mmsysmonitor.conf` file on all the HDFS Transparency nodes as follows:

For Open Source Apache:

```
pidfilepath = /tmp
binfilepath = /usr/lpp/mmfs/hadoop/bin
envfile = /var/mmfs/hadoop/etc/hadoop/hadoop-env.sh
slavesfile = /var/mmfs/hadoop/etc/hadoop/workers
coresitefile = /var/mmfs/hadoop/etc/hadoop/core-site.xml
gpfsbinfilepath = /usr/lpp/mmfs/bin
```

b. On all the HDFS Transparency nodes, run the following command:

```
mmsysmoncontrol restart
```

c. Check the HDFS Transparency status by running the following command:

```
mmhealth node show hadoopconnector -v
```

Monitoring HDFS Transparency status using IBM Storage Scale GUI

The IBM Storage Scale GUI can be used to monitor the state of the HDFS NameNodes and DataNodes. For CES HDFS integration with the IBM Storage Scale GUI, the GUI displays CES specific information like CES status, CES network states, CES node group, CES network address.

To access to CES HDFS information through GUI:

1. Log in to the IBM Storage Scale GUI to access the CES HDFS state information.
2. From the left hand side navigation, click **Services**.
3. Click **HDFS Transparency service**.

   The HDFS Transparency services contain the NameNodes, DataNodes and Events tab that contains the information about the HDFS Transparency cluster node states.

   Note: When a protocol is enabled, it will be on all the nodes that are configured to be a protocol node. However, only the nodes specified as the NameNode(s) will be enabled as CES HDFS nodes.

   Therefore, the HDFS Transparency NameNodes list might be less than the overall nodes listed in the **CES Nodes service** panel.

Recovering an HDFS Transparency cluster

Learn how to bring back online an HDFS Transparency cluster.

Disk failure or other unforeseen storage issues sometimes cause IBM Storage Scale file systems to get unmounted. In such cases, HDFS Transparency automatically shuts down itself, and workloads could report an exception. When the IBM Storage Scale cluster is back functioning, follow this recovery procedure for HDFS Transparency to bring the cluster back online.

1. Shut down HDFS Transparency NameNodes and Data Nodes by using the next commands.

   ```
   # mmces node suspend --stop -N <NameNode1_HOST>,<NameNode2_HOST>
   # mmhdfs hdfs-dn stop
   ```

   Use the **mmces node suspend** command to stop the NameNodes. Using this command is needed to ensure that the root directory shared with CES gets unlocked.

2. To retrieve the mount points that HDFS Transparency uses for the IBM Storage Scale file system, run the **mmhdfs config get** command as shown in the following example.

   Example:

   ```
   # mmhdfs config get gpfs-site.xml -k gpfs.mnt.dir
   gpfs.mnt.dir=/gpfs1,/gpfs2
   ```

3. If a secondary file system is configured, unmount that one first. Then, unmount the primary file system.

Example:

```
# mmumount gpfs2 -a
# mmumount gpfs1 -a
```

4. Check the status of HDFS Transparency to ensure that all the NameNodes and DataNodes are down.

```
# mmhdfs hdfs status
```

5. Remount the IBM Storage Scale file systems.

```
# mmmount gpfs1 -a
# mmmount gpfs2 -a
```

   Make sure that all the file systems are successfully mounted. Use the **mmlsmount** and **mount** commands to verify.

6. Start the HDFS Transparency NameNodes and DataNodes.

```
# mmces node resume --start -N <NameNode1_HOST>,<NameNode2_HOST>
# mmhdfs hdfs-dn start
```

7. Check the status of HDFS Transparency to corroborate that all NameNodes and DataNodes are in operation.

```
# mmhdfs hdfs status
```

Kerberos

This section describes how to set up Kerberos under CES HDFS.

Note:

- MIT Kerberos and Red Hat IPA Kerberos are supported.
- As per the prerequisites of Kerberos, the hostname of the cluster nodes that belong to the cluster must be in lowercase.
- If you need to set up more than one HDFS Transparency cluster by using a common KDC server, see If Kerberos was configured on multiple HDFS Transparency clusters using a common KDC server and the supplied `gpfs_kerberos_configuration.py` script, `kinit` with the hdfs user principal fails for all the clusters except the most recent one.

  This limitation has been fixed in HDFS Transparency 3.1.1.6.

Prerequisites

Learn the prerequisites that you must comply with before you enable Kerberos.

- Configure FQDN for all the hostname entries in your environment for consistency before you enable Kerberos. The **hostname** and **hostname -f** command outputs should have FQDN information.
- For all the hostname entries that are being replaced, ensure that you use FQDN hostnames from your environment.
- It is recommended having the hostname resolution through a working DNS setup.
- Synchronize clocks by using chronyd. Use the following command to check the time on all the IBM Storage Scale nodes:

```
# mmdsh -N all "date +%m%d%H%M%S%N"
```

- Change hostnames before you enable Kerberos. In case you must change the hostname after enabling Kerberos then re-create the principals and keytab also.
- If you need to set up more than one HDFS Transparency cluster by using a common KDC server, see Note.

- If you use Microsoft Active Directory based Kerberos with the 8u241 or a higher versions of Java Development Kit (JDK), for example OpenJDK 1.8.0u242+, you must disable referrals by making the following configuration:

  ```
  sun.security.krb5.disableReferrals=true
  ```

  Otherwise, HDFS Transparency services could fail to authenticate with each other through Kerberos.

  For more information about Java requirements, see Cloudera Docs.

Kerberos authentication with Active Directory (AD) support

Overview

Kerberos is a network authentication protocol. It is designed to provide strong authentication for client/server applications by using secret-key cryptography.

An active directory is a database that keeps track of all the user accounts and passwords in your organization. By using an active directory, you can store your user accounts and passwords in one protected location, which can improve the security of your organization. Network administrators can use active directories to allow or deny access to specific applications by end users through the trees in the network.

This section shows ways to setup a Kerberos authentication with Windows AD service on the HDP and HDFS Transparency cluster.

Prerequisites

Simplify Windows computer name

This topic lists the steps to simplify windows computer name.

Simplify the Windows computer name via the following sequence of operations (optional step):

1. On the **Start** screen, type Control Panel, and press ENTER.
2. Navigate to **System and Security**, and then click System.
3. Advanced system settings.
4. Under **Computer name** click Change settings.
5. On the **Computer Name** tab, add simple computer name such as “adverser” and click OK.
6. Restart computer.

Add Windows ip/hostname

Add the Windows ip and full hostname into the `/etc/hosts` file onto all the Hadoop nodes. This is required if all the nodes (Hadoop nodes and Windows computer) cannot be resolved via DNS.

This example adds the following Windows ip and hostname:

```
192.0.2.10 adserver.ad.gpfs.net
```

Synchronize Linux and Windows time

On all Hadoop and GPFS nodes perform the ntpdate command to sync the time across all nodes.

```
[root@c902f14x13 ~]# ntpdate adserver.ad.gpfs.net
13 May 12:52:32 ntpdate[3753]: step time server 192.0.2.10 offset 342.048227 sec
```

Add all Hadoop nodes into Windows hosts

On the Windows server, add all the Hadoop nodes in `C:\Windows\System32\drivers\etc\hosts` if the Hadoop node's IPs are not resolvable by the Windows server.

```
# localhost name resolution is handled within DNS itself.
#    192.0.2.22       localhost
#    ::1             localhost
```

```
192.0.2.11    c902f05x04.gpfs.net
192.0.2.12    c902f05x05.gpfs.net
192.0.2.13    c902f05x06.gpfs.net
192.0.2.14    c902f05x01.gpfs.net
192.0.2.15    c902f05x02.gpfs.net
192.0.2.16    c902f05x03.gpfs.net
192.0.2.17    c902f14x01.gpfs.net
192.0.2.18    c902f14x02.gpfs.net
192.0.2.19    c902f14x03.gpfs.net
```

Install and Configure Active Directory

A Domain Controller (DC) allows the creation of logical containers. These containers consist of users, computers and groups. The Domain Controllers also help in organizing and managing the Servers.

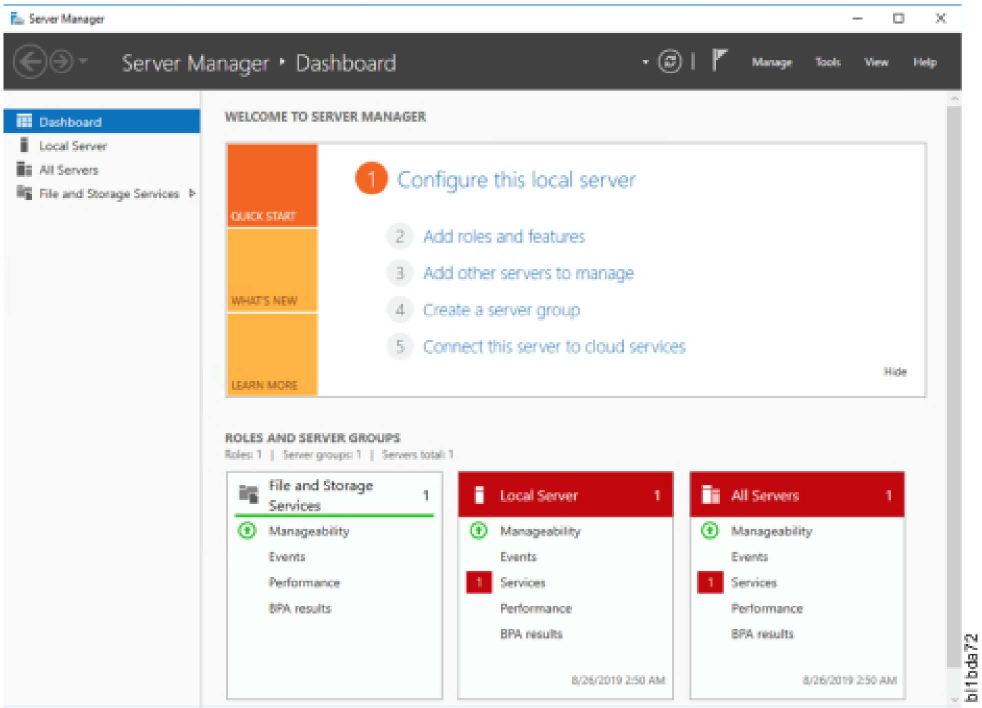
Active Directory is a service which runs on the Domain Controller. One uses this service to create logical containers.

Follow the steps below to setup the Active Directory services and promote it to a Domain Controller.

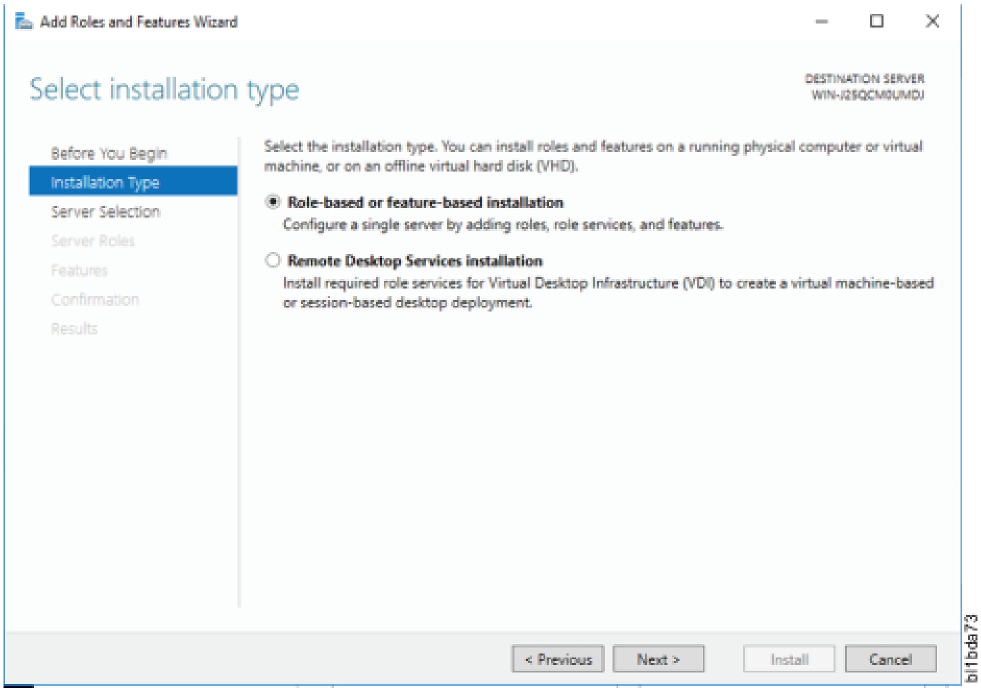
This example uses the following information:

```
Root domain name: AD.GPFS.NET
Password: Admin1234
NetBIOS domain name: AD0
```

1. Navigate to the Windows Server Manager.
2. Click **Add Roles and Features**.



3. It will open **Add Roles and Features** wizard. Click Next.



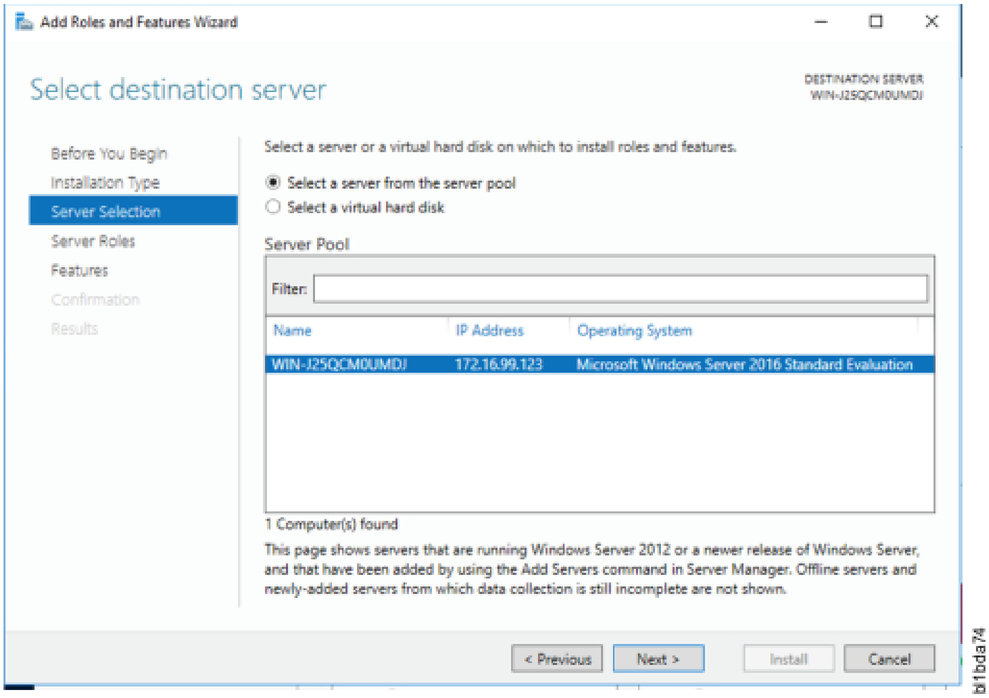
Select installation type

Select the installation type. You can install roles and features on a running physical computer or virtual machine, or on an offline virtual hard disk (VHD).

● **Role-based or feature-based installation**
Configure a single server by adding roles, role services, and features.

○ **Remote Desktop Services installation**
Install required role services for Virtual Desktop Infrastructure (VDI) to create a virtual machine-based or session-based desktop deployment.

4. Select the server from the server pool and click Next.



Select destination server

Select a server or a virtual hard disk on which to install roles and features.

● Select a server from the server pool
○ Select a virtual hard disk

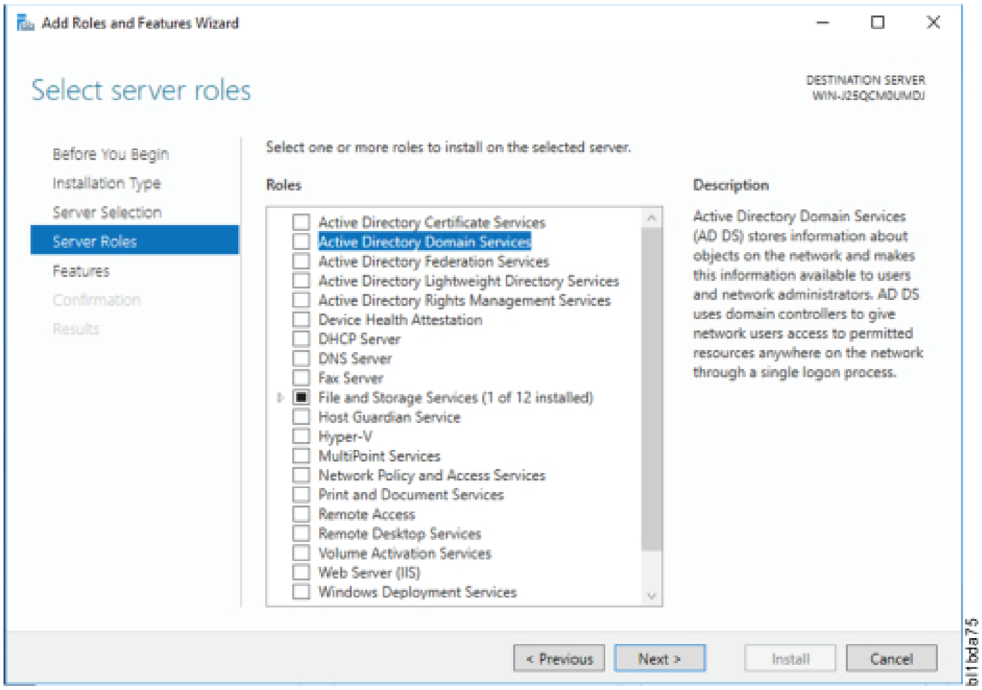
Server Pool

Name	IP Address	Operating System
WIN-J25QCM0UMDJ	172.16.99.123	Microsoft Windows Server 2016 Standard Evaluation

1 Computer(s) found

This page shows servers that are running Windows Server 2012 or a newer release of Windows Server, and that have been added by using the Add Servers command in Server Manager. Offline servers and newly-added servers from which data collection is still incomplete are not shown.

5. Click Checkbox to select Active Directory Domain Services.



Add Roles and Features Wizard

Select server roles

DESTINATION SERVER
WIN-J25QCM0UMDJ

- Before You Begin
- Installation Type
- Server Selection
- Server Roles
- Features
- Confirmation
- Results

Select one or more roles to install on the selected server.

Roles

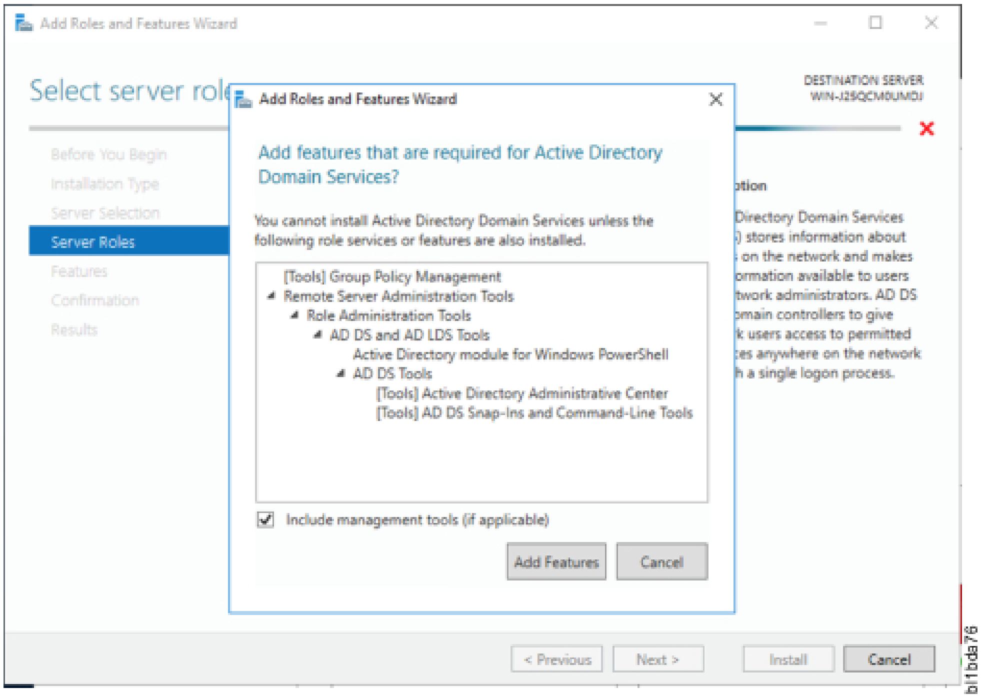
- ☐ Active Directory Certificate Services
- ☐ Active Directory Domain Services
- ☐ Active Directory Federation Services
- ☐ Active Directory Lightweight Directory Services
- ☐ Active Directory Rights Management Services
- ☐ Device Health Attestation
- ☐ DHCP Server
- ☐ DNS Server
- ☐ Fax Server
- ■ File and Storage Services (1 of 12 installed)
- ☐ Host Guardian Service
- ☐ Hyper-V
- ☐ MultiPoint Services
- ☐ Network Policy and Access Services
- ☐ Print and Document Services
- ☐ Remote Access
- ☐ Remote Desktop Services
- ☐ Volume Activation Services
- ☐ Web Server (IIS)
- ☐ Windows Deployment Services

Description

Active Directory Domain Services (AD DS) stores information about objects on the network and makes this information available to users and network administrators. AD DS uses domain controllers to give network users access to permitted resources anywhere on the network through a single logon process.

< Previous | Next > | Install | Cancel

6. On the popup Window, just click Add Features.



Add Roles and Features Wizard

Add features that are required for Active Directory Domain Services?

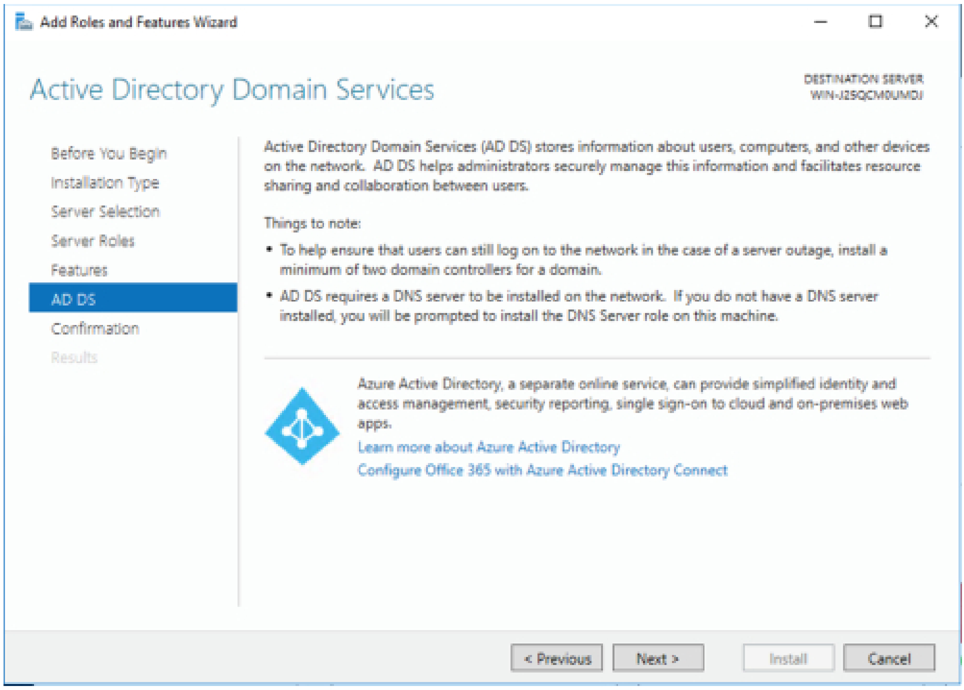
You cannot install Active Directory Domain Services unless the following role services or features are also installed.

```
[Tools] Group Policy Management
◢ Remote Server Administration Tools
   ◢ Role Administration Tools
      ◢ AD DS and AD LDS Tools
            Active Directory module for Windows PowerShell
         ◢ AD DS Tools
               [Tools] Active Directory Administrative Center
               [Tools] AD DS Snap-Ins and Command-Line Tools
```

☑ Include management tools (if applicable)

Add Features | Cancel

7. On the description window of Active Directory Domain Services, click Next.



Active Directory Domain Services

Active Directory Domain Services (AD DS) stores information about users, computers, and other devices on the network. AD DS helps administrators securely manage this information and facilitates resource sharing and collaboration between users.

Things to note:

- To help ensure that users can still log on to the network in the case of a server outage, install a minimum of two domain controllers for a domain.
- AD DS requires a DNS server to be installed on the network. If you do not have a DNS server installed, you will be prompted to install the DNS Server role on this machine.

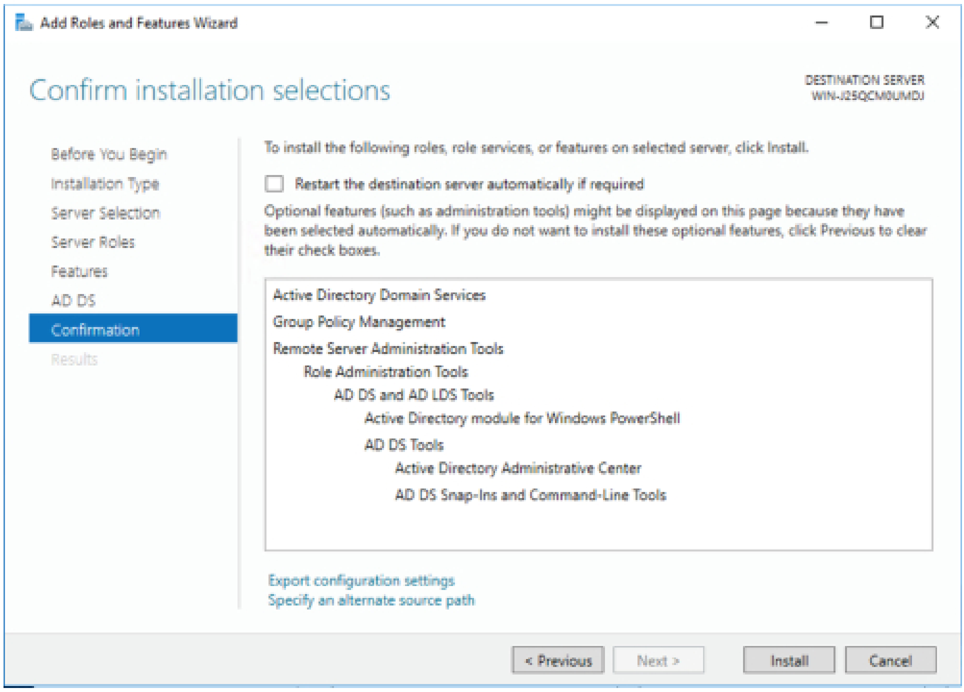
Azure Active Directory, a separate online service, can provide simplified identity and access management, security reporting, single sign-on to cloud and on-premises web apps.

Learn more about Azure Active Directory

Configure Office 365 with Azure Active Directory Connect

bl1bda77

8. Click Install on the Confirmation window.



Confirm installation selections

To install the following roles, role services, or features on selected server, click Install.

☐ Restart the destination server automatically if required

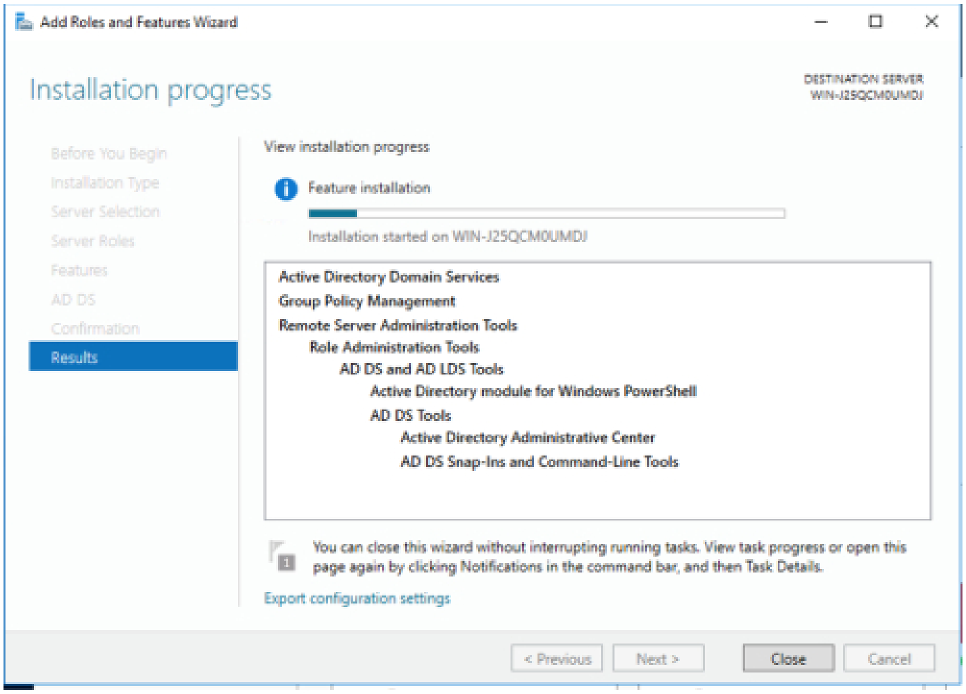
Optional features (such as administration tools) might be displayed on this page because they have been selected automatically. If you do not want to install these optional features, click Previous to clear their check boxes.

Active Directory Domain Services
Group Policy Management
Remote Server Administration Tools
    Role Administration Tools
        AD DS and AD LDS Tools
            Active Directory module for Windows PowerShell
            AD DS Tools
                Active Directory Administrative Center
                AD DS Snap-Ins and Command-Line Tools

Export configuration settings
Specify an alternate source path

bl1bda78

9. Installation process begins.



Installation progress

DESTINATION SERVER
WIN-J25QCM0UMDJ

View installation progress

Feature installation

Installation started on WIN-J25QCM0UMDJ

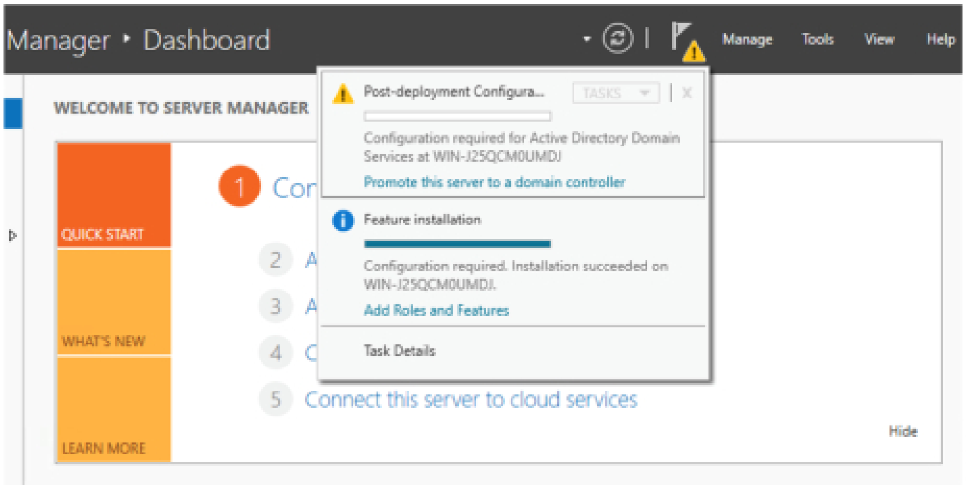
Active Directory Domain Services
Group Policy Management
Remote Server Administration Tools
    Role Administration Tools
        AD DS and AD LDS Tools
            Active Directory module for Windows PowerShell
            AD DS Tools
                Active Directory Administrative Center
                AD DS Snap-Ins and Command-Line Tools

You can close this wizard without interrupting running tasks. View task progress or open this page again by clicking Notifications in the command bar, and then Task Details.

Export configuration settings

bl1bda79

10. After installing AD DS Role, you can promote this Server to a Domain Controller.



Post-deployment Configura...

Configuration required for Active Directory Domain Services at WIN-J25QCM0UMDJ

Promote this server to a domain controller

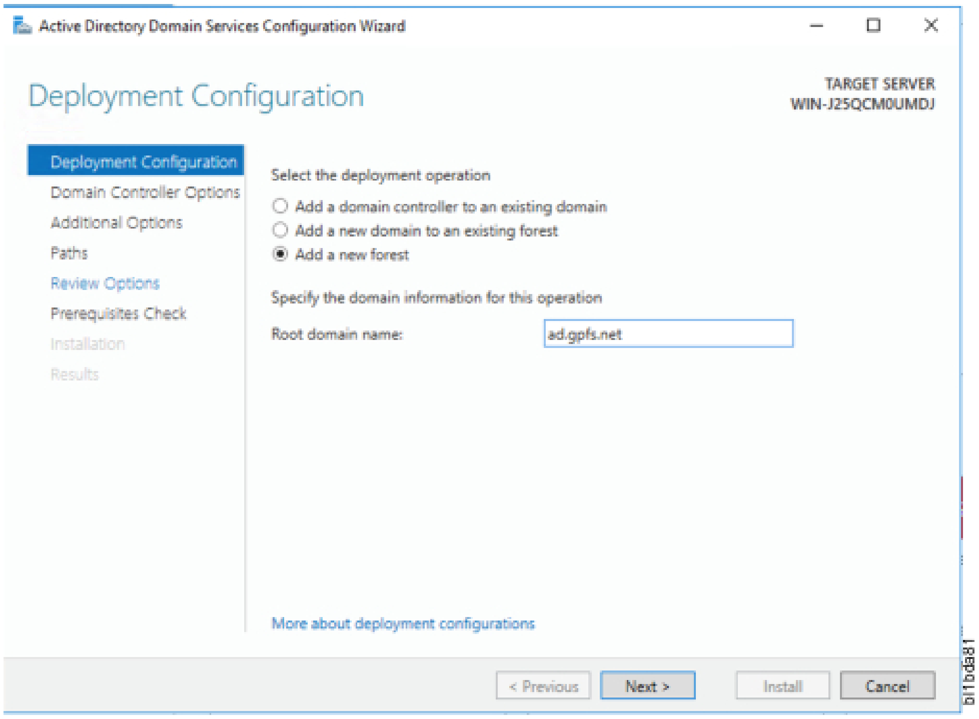
Feature installation

Configuration required. Installation succeeded on WIN-J25QCM0UMDJ.

Add Roles and Features

bl1bda80

11. Select Add a new forest and give the Root domain name, `ad.gpfs.net`. Click Next.



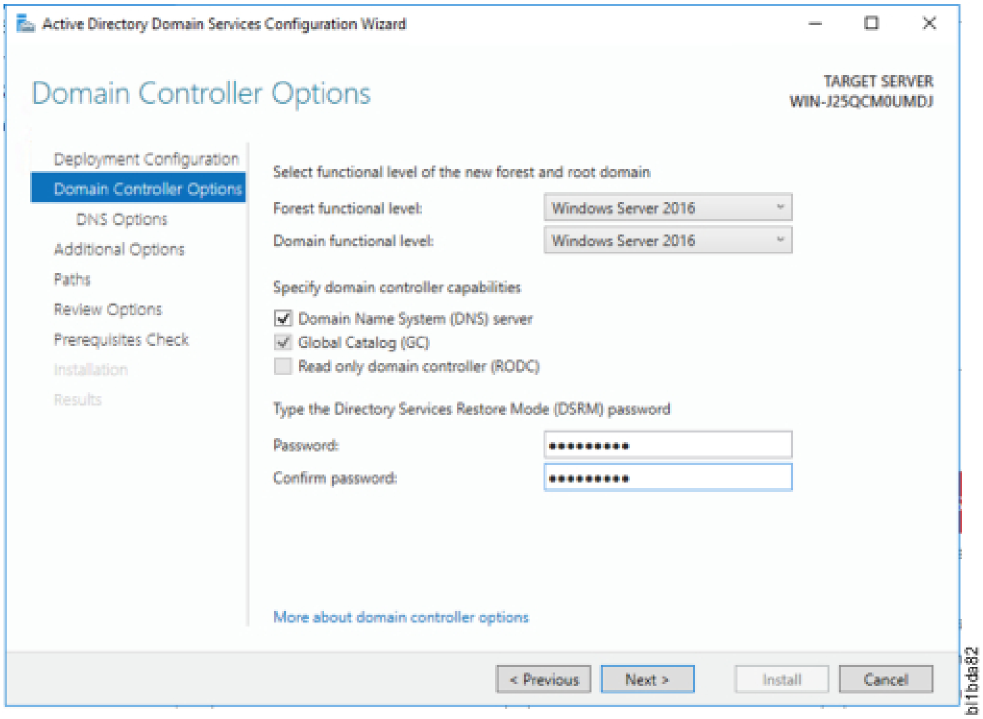
Select the deployment operation

○ Add a domain controller to an existing domain
○ Add a new domain to an existing forest
● Add a new forest

Specify the domain information for this operation

Root domain name: ad.gpfs.net

12. Enter the Directory Services Restore Mode password.



Select functional level of the new forest and root domain

Forest functional level: Windows Server 2016

Domain functional level: Windows Server 2016

Specify domain controller capabilities

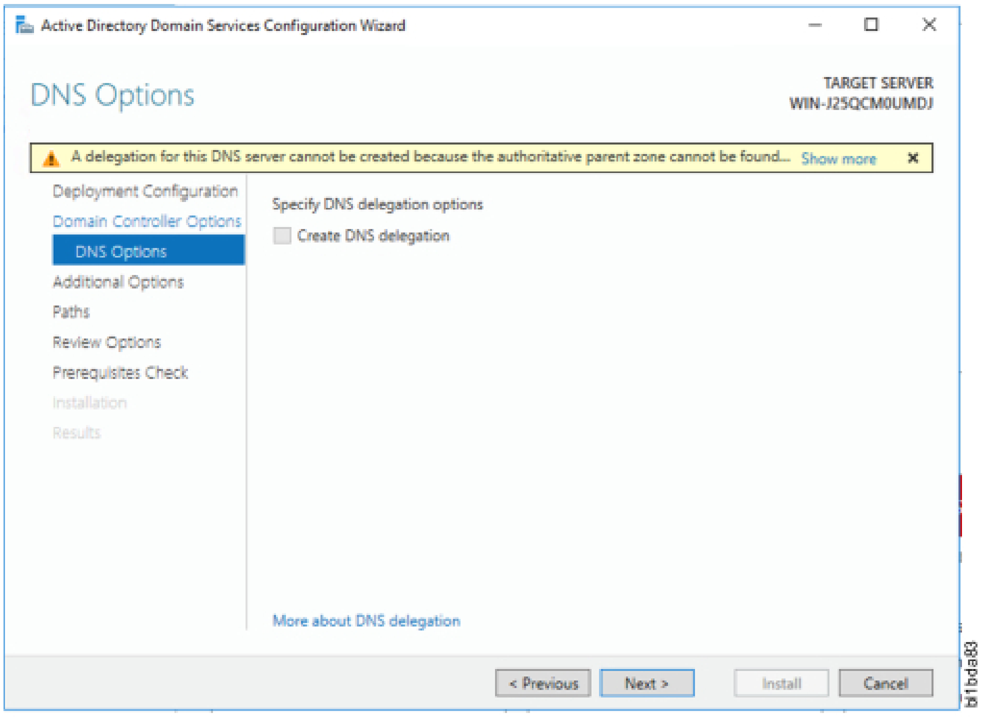
☑ Domain Name System (DNS) server
☑ Global Catalog (GC)
□ Read only domain controller (RODC)

Type the Directory Services Restore Mode (DSRM) password

Password: ••••••••••

Confirm password: ••••••••••

13. Ignore the warning message.


Active Directory Domain Services Configuration Wizard

DNS Options

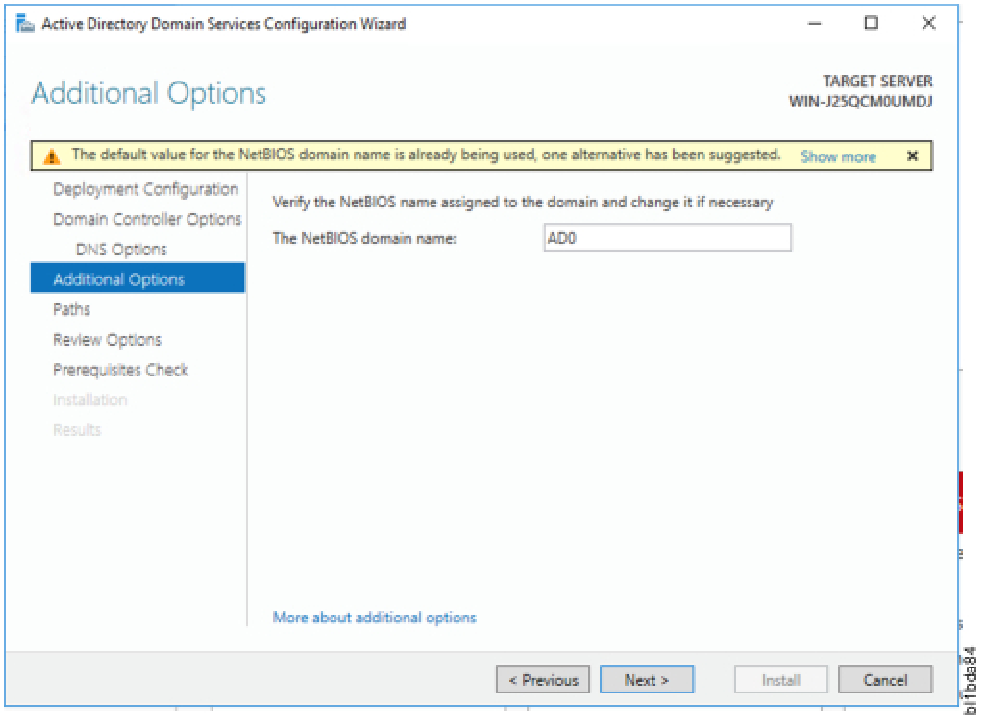
TARGET SERVER
WIN-J25QCM0UMDJ

⚠ A delegation for this DNS server cannot be created because the authoritative parent zone cannot be found... Show more ✕

Specify DNS delegation options

□ Create DNS delegation

14. Use the default NetBIOS domain name and click Next.


Active Directory Domain Services Configuration Wizard

Additional Options

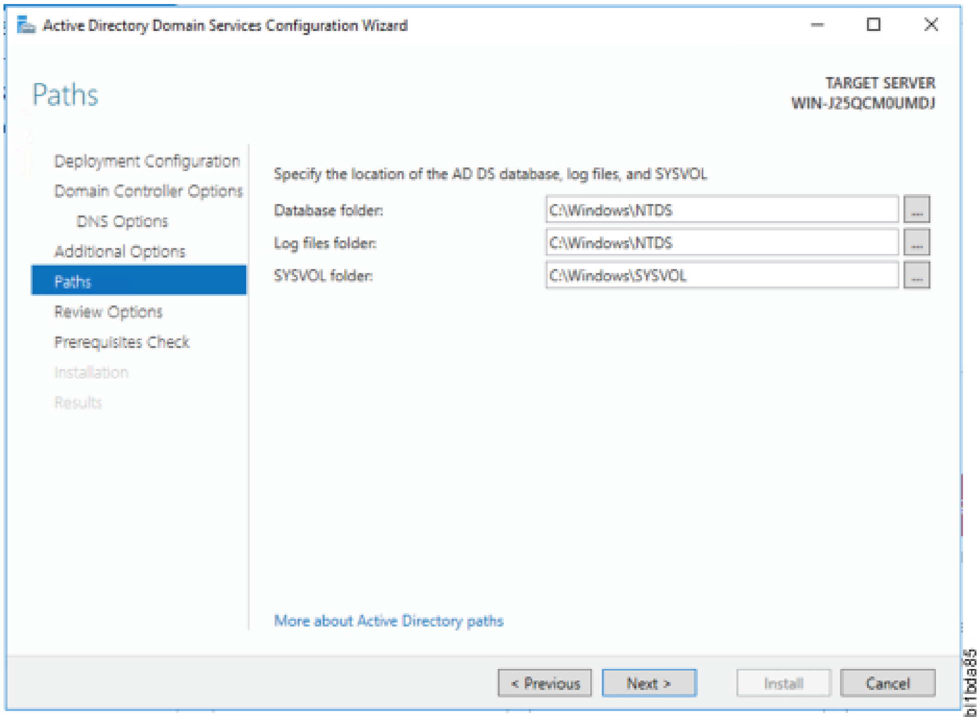
TARGET SERVER
WIN-J25QCM0UMDJ

⚠ The default value for the NetBIOS domain name is already being used, one alternative has been suggested. Show more ✕

Verify the NetBIOS name assigned to the domain and change it if necessary

The NetBIOS domain name: AD0

15. Use the default paths and click Next.



Paths

TARGET SERVER
WIN-J25QCM0UMDJ

Specify the location of the AD DS database, log files, and SYSVOL

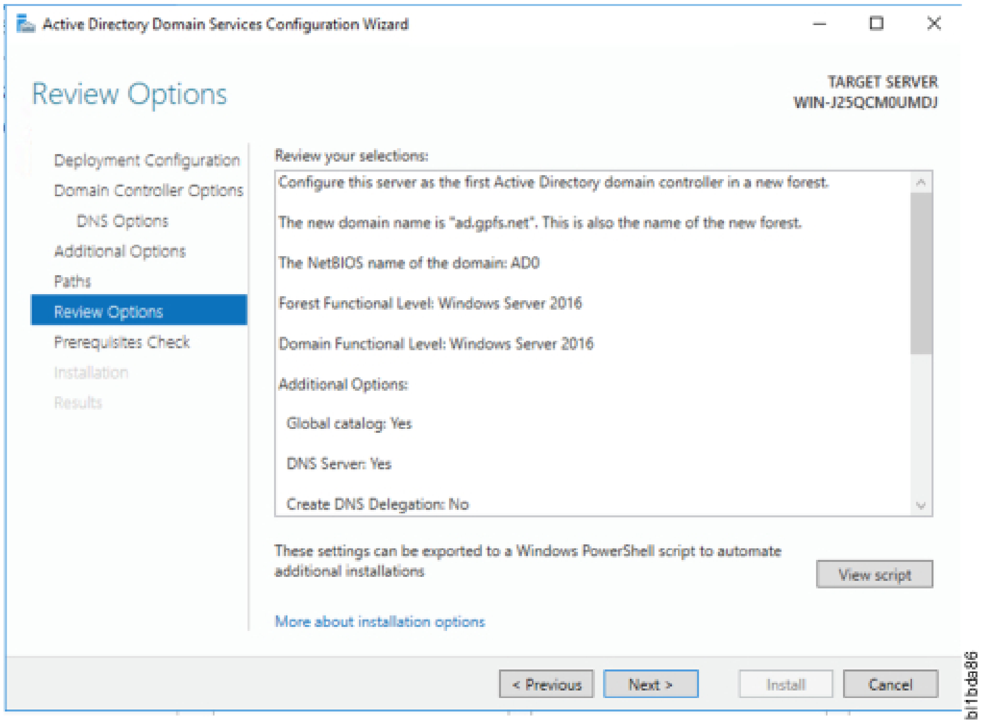
Database folder: C:\Windows\NTDS

Log files folder: C:\Windows\NTDS

SYSVOL folder: C:\Windows\SYSVOL

bl1bda85

16. Review and click Next if no errors.



Review Options

TARGET SERVER
WIN-J25QCM0UMDJ

Review your selections:

Configure this server as the first Active Directory domain controller in a new forest.

The new domain name is "ad.gpfs.net". This is also the name of the new forest.

The NetBIOS name of the domain: AD0

Forest Functional Level: Windows Server 2016

Domain Functional Level: Windows Server 2016

Additional Options:

Global catalog: Yes

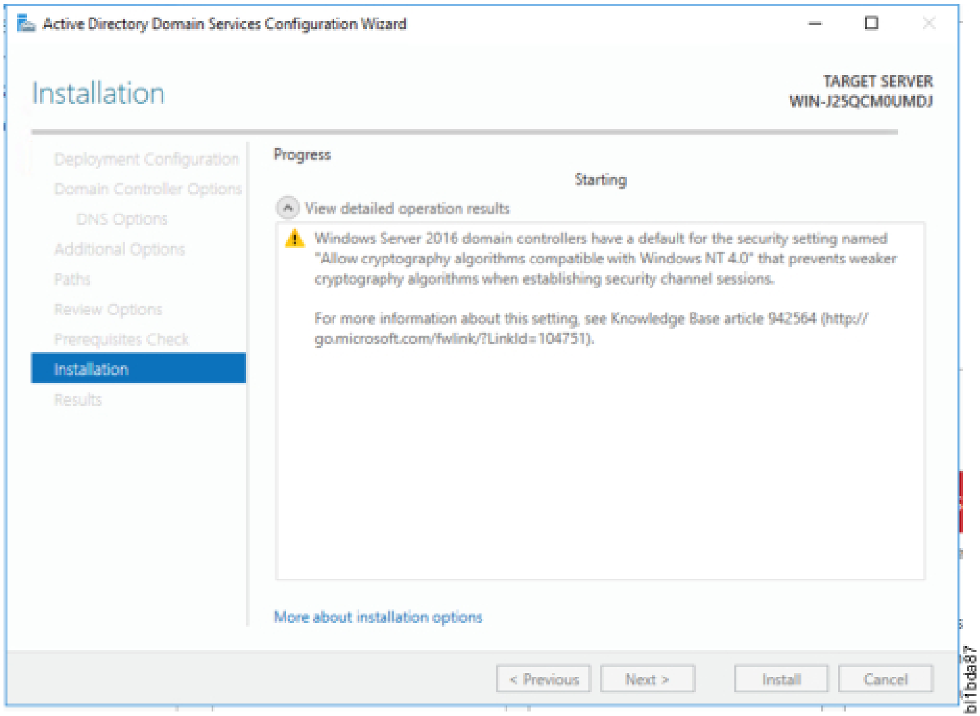
DNS Server: Yes

Create DNS Delegation: No

These settings can be exported to a Windows PowerShell script to automate additional installations

bl1bda86

17. Click Install and wait for the installation to finish.



Starting

Windows Server 2016 domain controllers have a default for the security setting named "Allow cryptography algorithms compatible with Windows NT 4.0" that prevents weaker cryptography algorithms when establishing security channel sessions.

For more information about this setting, see Knowledge Base article 942564 (http://go.microsoft.com/fwlink/?LinkId=104751).

bl1bda87

18. The Domain Controller is now set up.

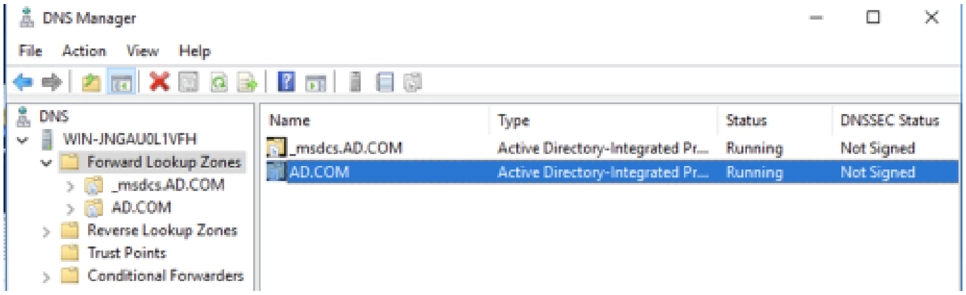
Enable Kerberos on existing Active Directory

This section lists the steps to enable Kerberos on existing Active Directory.

Installation and Configuration of Active Directory Certificate Services

Active Directory Certificate service is one of the essential services that is required for the certificate management within the organization.

If the Domain is up and running as shown in the picture below, then the ADCS is successfully installed and configured.



Name	Type	Status	DNSSEC Status
_msdcs.AD.COM	Active Directory-Integrated Pr...	Running	Not Signed
AD.COM	Active Directory-Integrated Pr...	Running	Not Signed

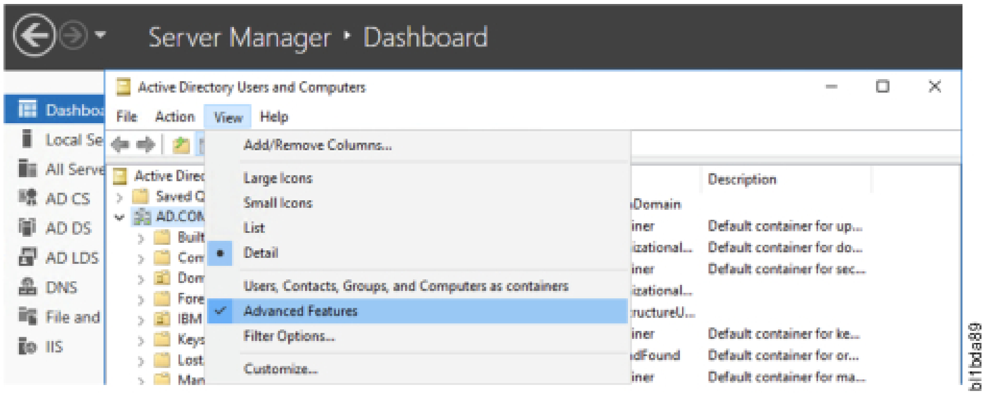
bl1bda88

Note: This is required only if you are generating your own certificates for Active Directory.

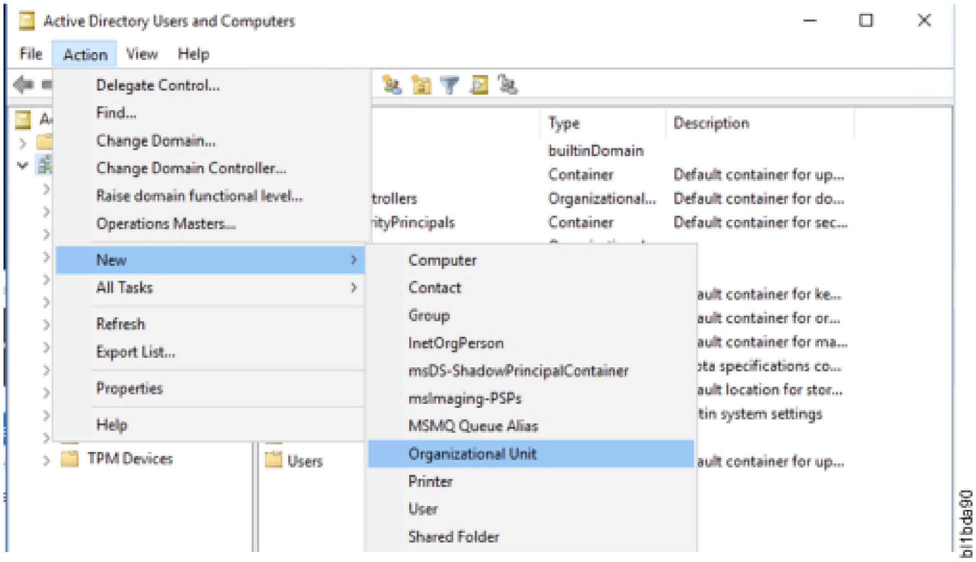
Create AD user and delegate control

Create a container, Kerberos admin, and set permissions for the cluster.

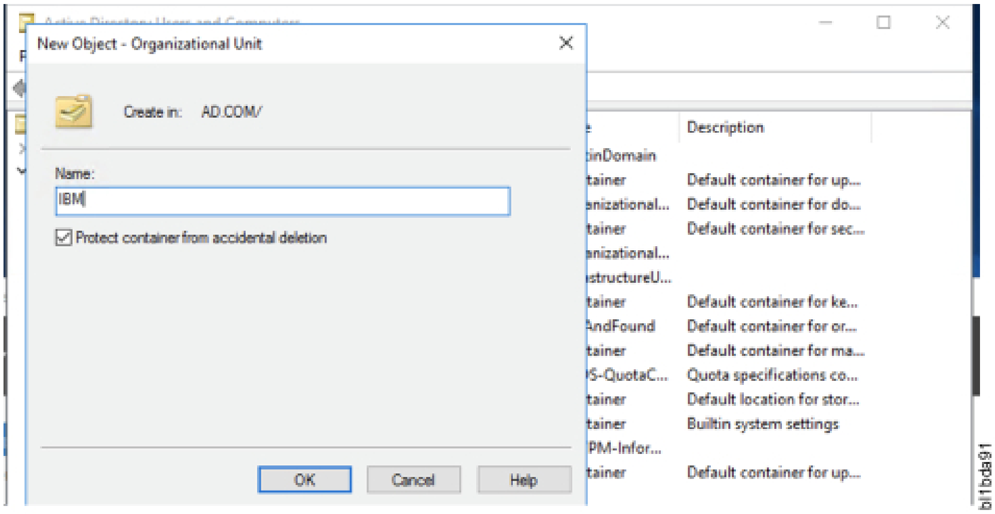
1. Navigate to **Server Manager** > **Tools** > **Active Directory Users and Computers**.
2. Click **View** and check **Advanced Features**.



3. Create a container. This example uses the name *IBM*. Navigate to **Action** > **New** > **Organizational Unit**.



4. Specify the container name (Example uses name “IBM”).

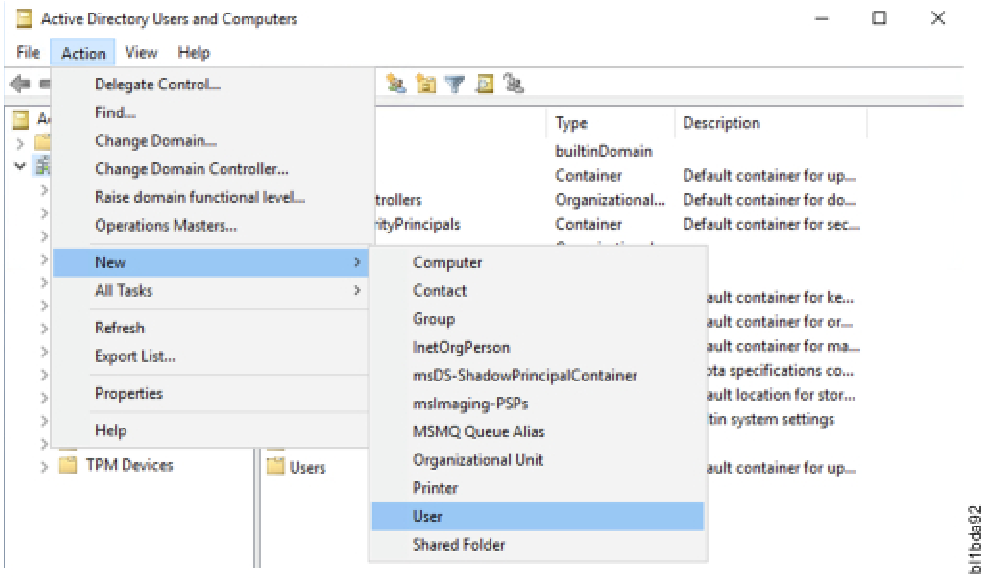


Create in: AD.COM/

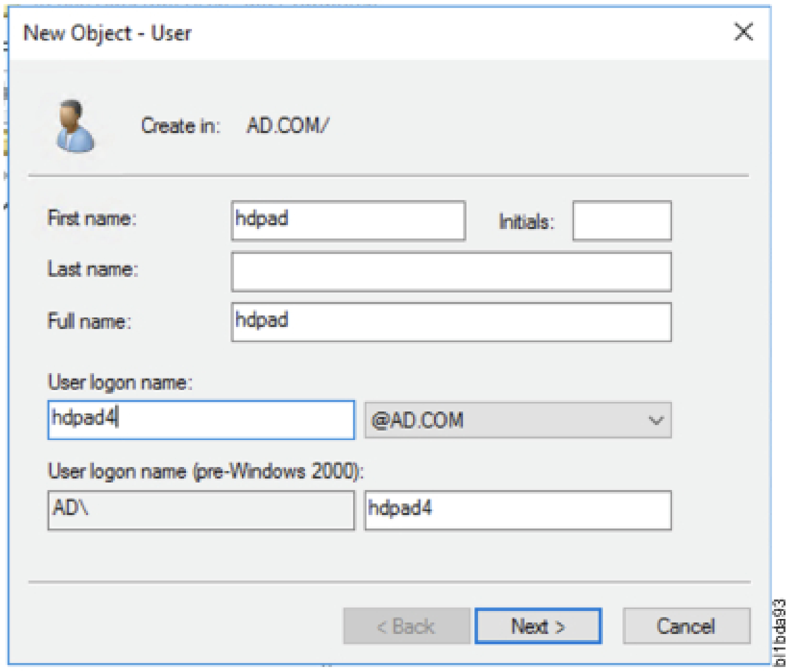
Name:
IBM

☑ Protect container from accidental deletion

5. Create a user named *hdpad*. Navigate to **Action** > **New** > **User**.



6. Specify the User logon name.



New Object - User

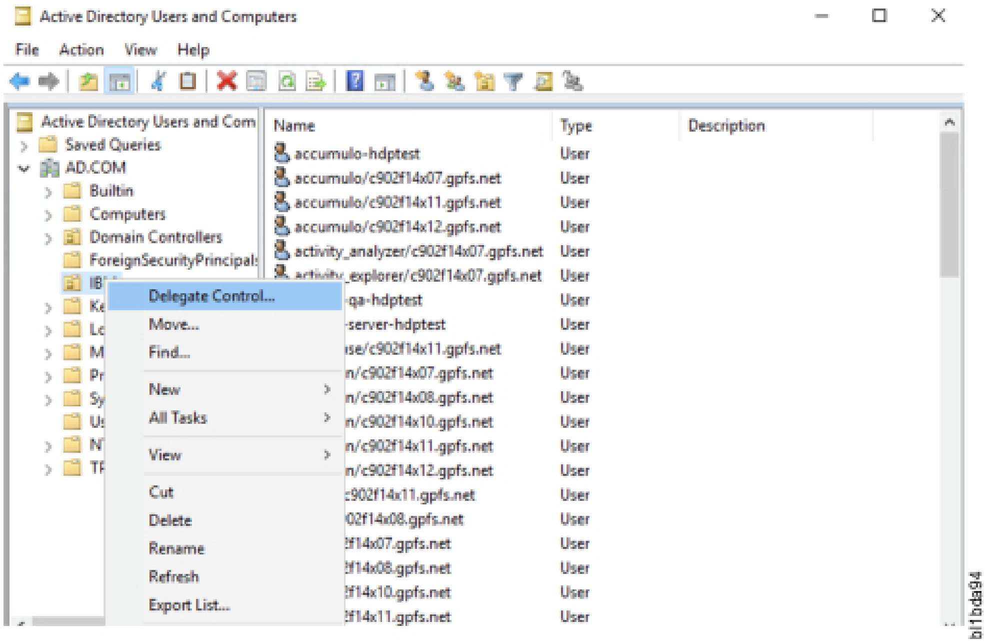
Create in: AD.COM/

First name: hdpad
Initials:
Last name:
Full name: hdpad

User logon name:
hdpad4 @AD.COM

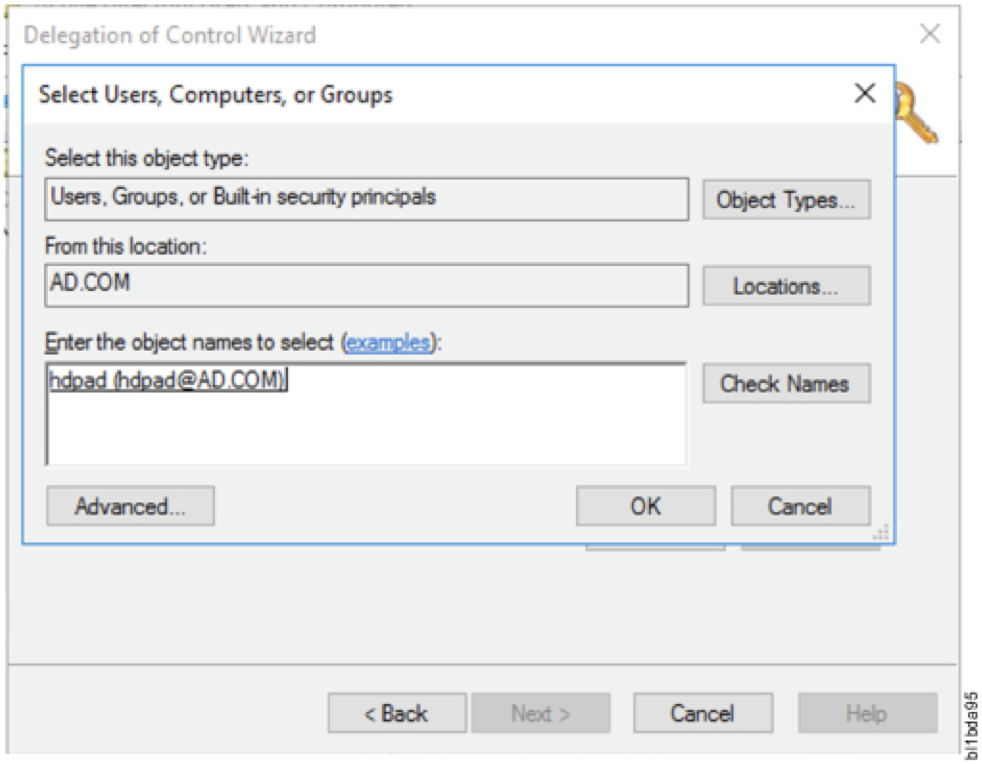
User logon name (pre-Windows 2000):
AD\ hdpad4

7. Delegate control of the container to *hdpad*. Right-click on the new container (IBM), and select **Delegate Control**.



Name	Type	Description
accumulo-hdptest	User	
accumulo/c902f14x07.gpfs.net	User	
accumulo/c902f14x11.gpfs.net	User	
accumulo/c902f14x12.gpfs.net	User	
activity_analyzer/c902f14x07.gpfs.net	User	
activity_explorer/c902f14x07.gpfs.net	User	
qa-hdptest	User	
server-hdptest	User	
se/c902f14x11.gpfs.net	User	
n/c902f14x07.gpfs.net	User	
n/c902f14x08.gpfs.net	User	
n/c902f14x10.gpfs.net	User	
n/c902f14x11.gpfs.net	User	
n/c902f14x12.gpfs.net	User	
c902f14x11.gpfs.net	User	
02f14x08.gpfs.net	User	
f14x07.gpfs.net	User	
f14x08.gpfs.net	User	
f14x10.gpfs.net	User	
f14x11.gpfs.net	User	

8. In the Delegation of Control Wizard, enter *hdpad* and click Check Names.

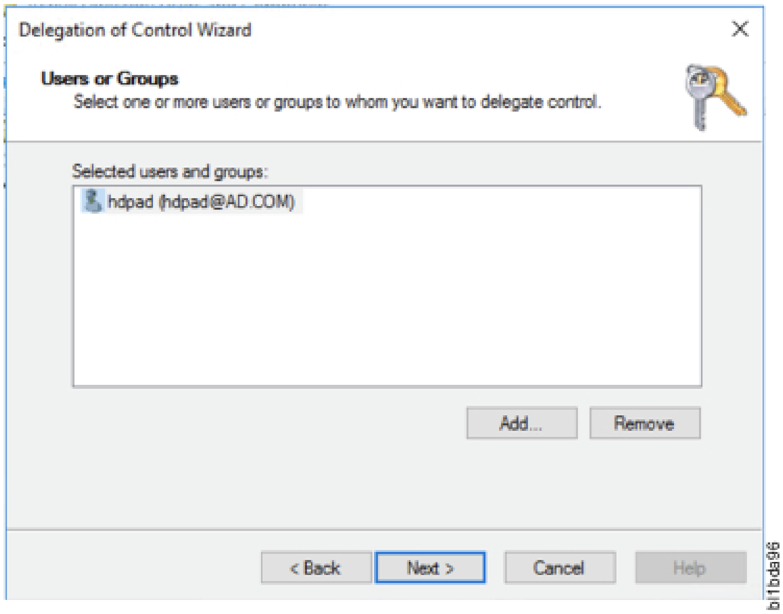


Select this object type:
Users, Groups, or Built-in security principals

From this location:
AD.COM

Enter the object names to select (examples):
hdpad (hdpad@AD.COM)

9. Confirm that the *hdpad* name is listed and click Next.



Delegation of Control Wizard

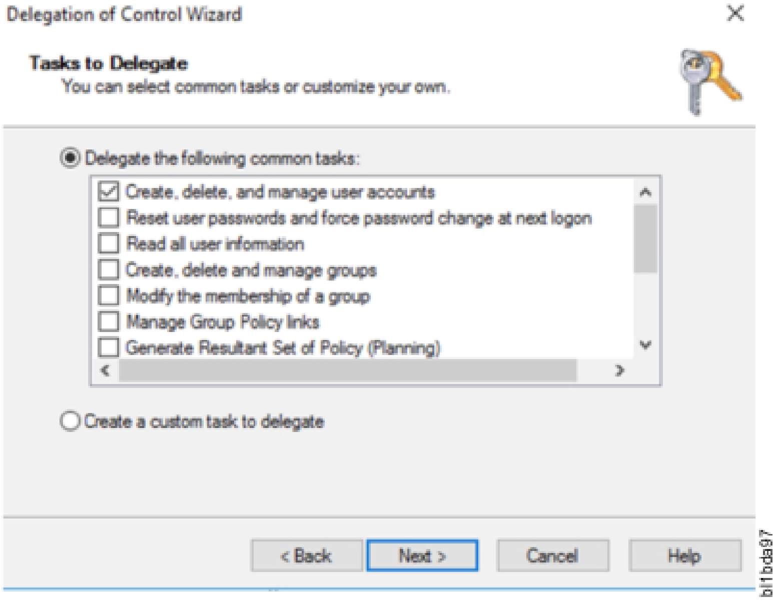
Users or Groups

Select one or more users or groups to whom you want to delegate control.

Selected users and groups:

hdpad (hdpad@AD.COM)

10. In the Tasks to Delegate field, check Create, delete, and manage user accounts.



Delegation of Control Wizard

Tasks to Delegate

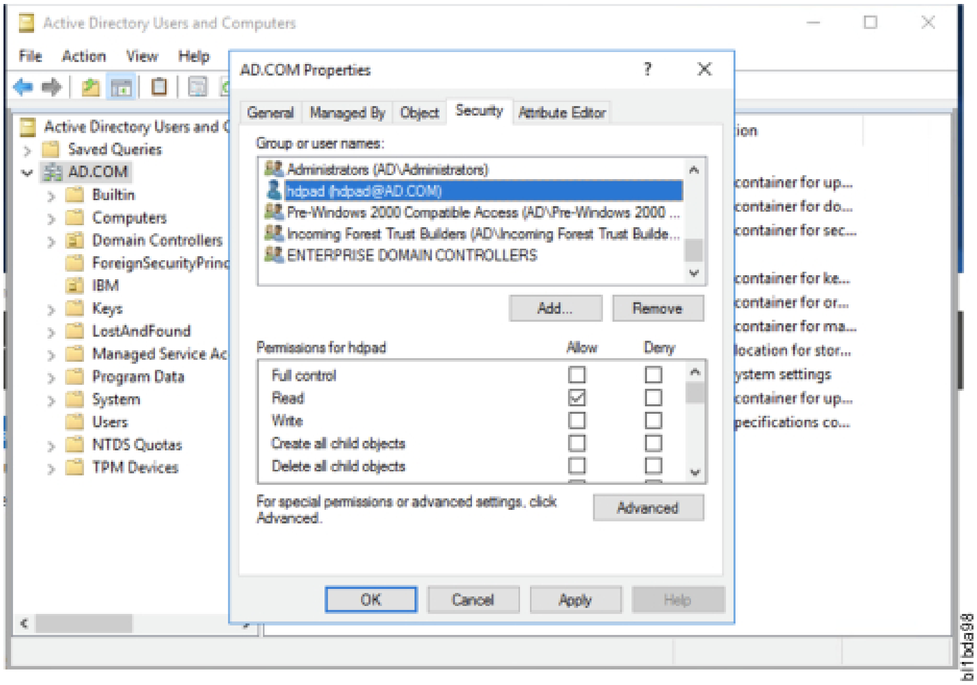
You can select common tasks or customize your own.

● Delegate the following common tasks:

☑ Create, delete, and manage user accounts
☐ Reset user passwords and force password change at next logon
☐ Read all user information
☐ Create, delete and manage groups
☐ Modify the membership of a group
☐ Manage Group Policy links
☐ Generate Resultant Set of Policy (Planning)

○ Create a custom task to delegate

11. Navigate to **AD.COM** > **Properties** > **Security** and add the *hdpad* user.



Group or user names:

- Administrators (AD\Administrators)
- hdpad (hdpad@AD.COM)
- Pre-Windows 2000 Compatible Access (AD\Pre-Windows 2000 ...
- Incoming Forest Trust Builders (AD\Incoming Forest Trust Builde...
- ENTERPRISE DOMAIN CONTROLLERS

Permissions for hdpad	Allow	Deny
Full control	□	□
Read	☑	□
Write	□	□
Create all child objects	□	□
Delete all child objects	□	□

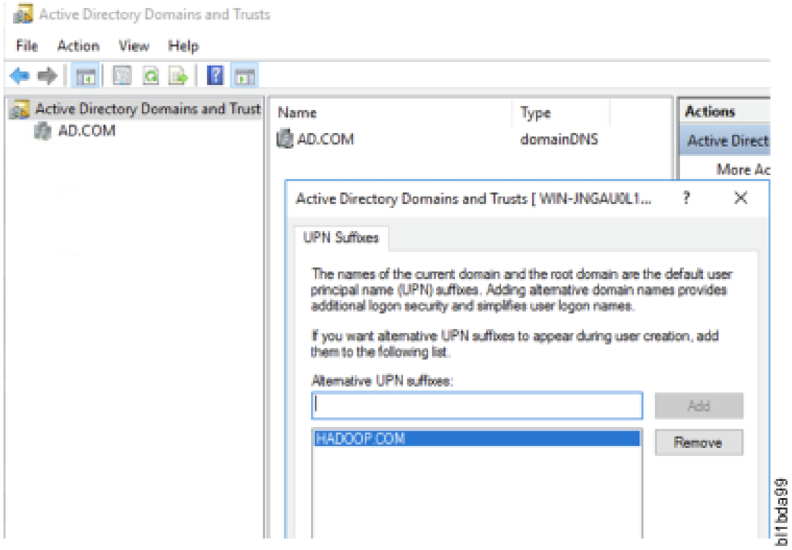
For special permissions or advanced settings, click Advanced.

Adding the domain of your Linux host(s) to be recognized by Active Directory

This topic lists the steps to add the domain of your Linux host(s) to be recognized by Active Directory.

Note: This step is required only if the domain of the Linux servers is different than the Active Directory.

1. On the Windows Host, navigate to **Server Manager** > **Tools** > **Active Directory Domains and Trusts**.
2. Click **Actions** > **Properties** > **UPN Suffixes**.
3. Add the alternative SPN. This is determined by running **`hostname -d`** on your Linux server.



Active Directory Domains and Trusts
File Action View Help
Active Directory Domains and Trust
AD.COM
Name
Type
AD.COM
domainDNS
Actions
Active Direct
More Ac
Active Directory Domains and Trusts [WIN-JNGAU0L1...
UPN Suffixes
Add
Remove
bl1bda99

The names of the current domain and the root domain are the default user principal name (UPN) suffixes. Adding alternative domain names provides additional logon security and simplifies user logon names.

If you want alternative UPN suffixes to appear during user creation, add them to the following list.

Alternative UPN suffixes:

HADOOP.COM

Configuring AD in Ambari

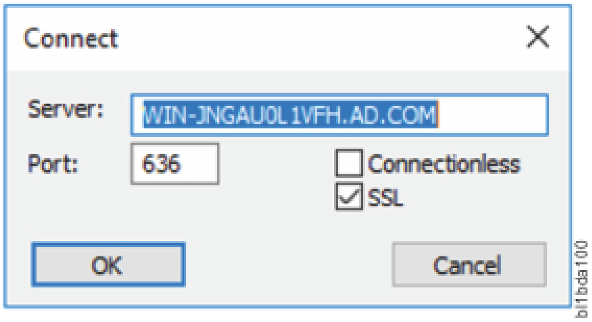
This section describes how to configure Kerberos with existing AD through the Ambari GUI.

Configuring Secure LDAP connection

The Lightweight Directory Access Protocol (LDAP) is used to read from and write to Active Directory. By default, LDAP traffic is transmitted unsecured. To make LDAP traffic confidential and secure, use Secure Sockets Layer (SSL) / Transport Layer Security (TLS) technology. Enable LDAP over SSL (LDAPS) by installing a properly formatted certificate from either a Microsoft certification authority (CA) or a non-Microsoft CA.

Follow the guide below to configure secure LDAP connection on Server 2016.

The picture shows a successfully configured Secure LDAP.

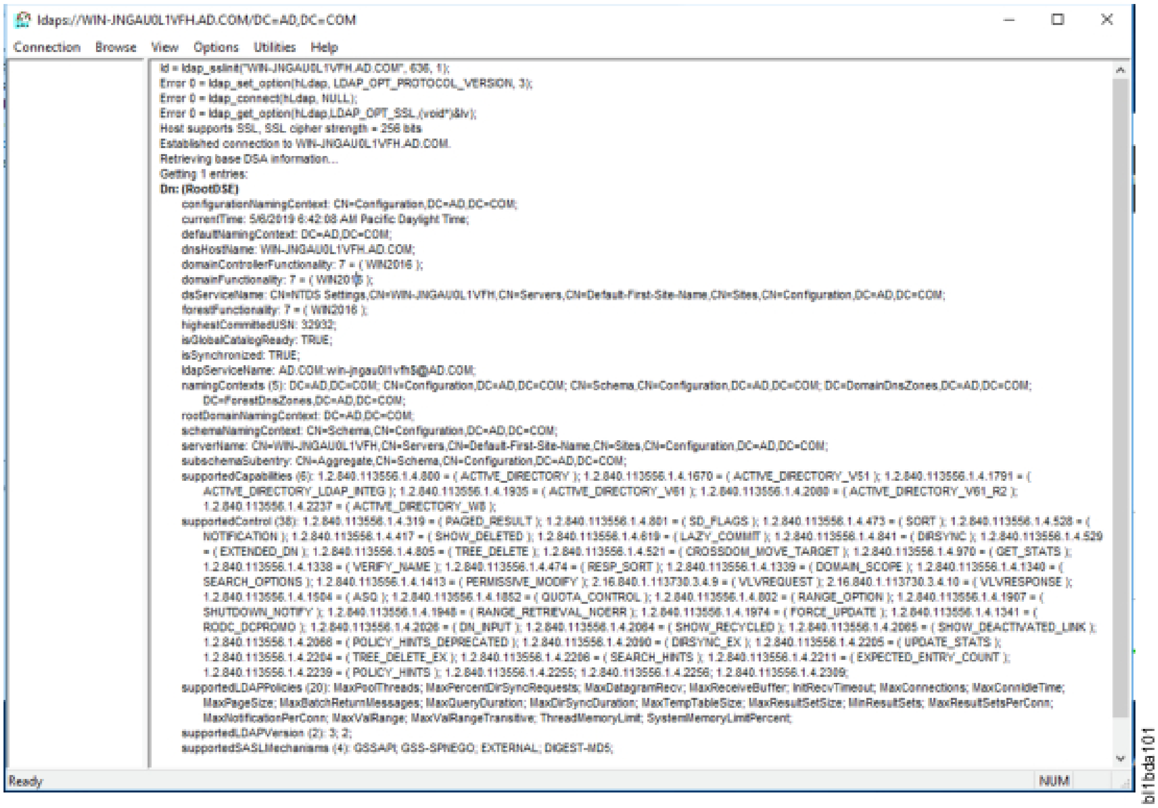


Server: WIN-JNGAU0L1VFH.AD.COM

Port: 636

□ Connectionless

☑ SSL



```
ld = ldap_sslinit("WIN-JNGAU0L1VFH.AD.COM", 636, 1);
Error 0 = ldap_set_option(hLdap, LDAP_OPT_PROTOCOL_VERSION, 3);
Error 0 = ldap_connect(hLdap, NULL);
Error 0 = ldap_get_option(hLdap,LDAP_OPT_SSL,(void*)&lv);
Host supports SSL, SSL cipher strength = 256 bits
Established connection to WIN-JNGAU0L1VFH.AD.COM.
Retrieving base DSA information...
Getting 1 entries:
Dn: (RootDSE)
    configurationNamingContext: CN=Configuration,DC=AD,DC=COM;
    currentTime: 5/6/2019 6:42:08 AM Pacific Daylight Time;
    defaultNamingContext: DC=AD,DC=COM;
    dnsHostName: WIN-JNGAU0L1VFH.AD.COM;
    domainControllerFunctionality: 7 = ( WIN2016 );
    domainFunctionality: 7 = ( WIN2016 );
    dsServiceName: CN=NTDS Settings,CN=WIN-JNGAU0L1VFH,CN=Servers,CN=Default-First-Site-Name,CN=Sites,CN=Configuration,DC=AD,DC=COM;
    forestFunctionality: 7 = ( WIN2016 );
    highestCommittedUSN: 32932;
    isGlobalCatalogReady: TRUE;
    isSynchronized: TRUE;
    ldapServiceName: AD.COM:win-jngau0l1vfh$@AD.COM;
    namingContexts (5): DC=AD,DC=COM; CN=Configuration,DC=AD,DC=COM; CN=Schema,CN=Configuration,DC=AD,DC=COM; DC=DomainDnsZones,DC=AD,DC=COM;
        DC=ForestDnsZones,DC=AD,DC=COM;
    rootDomainNamingContext: DC=AD,DC=COM;
    schemaNamingContext: CN=Schema,CN=Configuration,DC=AD,DC=COM;
    serverName: CN=WIN-JNGAU0L1VFH,CN=Servers,CN=Default-First-Site-Name,CN=Sites,CN=Configuration,DC=AD,DC=COM;
    subschemaSubentry: CN=Aggregate,CN=Schema,CN=Configuration,DC=AD,DC=COM;
    supportedCapabilities (6): 1.2.840.113556.1.4.800 = ( ACTIVE_DIRECTORY ); 1.2.840.113556.1.4.1670 = ( ACTIVE_DIRECTORY_V51 ); 1.2.840.113556.1.4.1791 = (
        ACTIVE_DIRECTORY_LDAP_INTEG ); 1.2.840.113556.1.4.1935 = ( ACTIVE_DIRECTORY_V61 ); 1.2.840.113556.1.4.2080 = ( ACTIVE_DIRECTORY_V61_R2 );
        1.2.840.113556.1.4.2237 = ( ACTIVE_DIRECTORY_W8 );
    supportedControl (38): 1.2.840.113556.1.4.319 = ( PAGED_RESULT ); 1.2.840.113556.1.4.801 = ( SD_FLAGS ); 1.2.840.113556.1.4.473 = ( SORT ); 1.2.840.113556.1.4.528 = (
        NOTIFICATION ); 1.2.840.113556.1.4.417 = ( SHOW_DELETED ); 1.2.840.113556.1.4.619 = ( LAZY_COMMIT ); 1.2.840.113556.1.4.841 = ( DIRSYNC ); 1.2.840.113556.1.4.529
        = ( EXTENDED_DN ); 1.2.840.113556.1.4.805 = ( TREE_DELETE ); 1.2.840.113556.1.4.521 = ( CROSSDOM_MOVE_TARGET ); 1.2.840.113556.1.4.970 = ( GET_STATS );
        1.2.840.113556.1.4.1338 = ( VERIFY_NAME ); 1.2.840.113556.1.4.474 = ( RESP_SORT ); 1.2.840.113556.1.4.1339 = ( DOMAIN_SCOPE ); 1.2.840.113556.1.4.1340 = (
        SEARCH_OPTIONS ); 1.2.840.113556.1.4.1413 = ( PERMISSIVE_MODIFY ); 2.16.840.1.113730.3.4.9 = ( VLVREQUEST ); 2.16.840.1.113730.3.4.10 = ( VLVRESPONSE );
        1.2.840.113556.1.4.1504 = ( ASQ ); 1.2.840.113556.1.4.1852 = ( QUOTA_CONTROL ); 1.2.840.113556.1.4.802 = ( RANGE_OPTION ); 1.2.840.113556.1.4.1907 = (
        SHUTDOWN_NOTIFY ); 1.2.840.113556.1.4.1948 = ( RANGE_RETRIEVAL_NOERR ); 1.2.840.113556.1.4.1974 = ( FORCE_UPDATE ); 1.2.840.113556.1.4.1341 = (
        RODC_DCPROMO ); 1.2.840.113556.1.4.2026 = ( DN_INPUT ); 1.2.840.113556.1.4.2064 = ( SHOW_RECYCLED ); 1.2.840.113556.1.4.2065 = ( SHOW_DEACTIVATED_LINK );
        1.2.840.113556.1.4.2066 = ( POLICY_HINTS_DEPRECATED ); 1.2.840.113556.1.4.2090 = ( DIRSYNC_EX ); 1.2.840.113556.1.4.2205 = ( UPDATE_STATS );
        1.2.840.113556.1.4.2204 = ( TREE_DELETE_EX ); 1.2.840.113556.1.4.2206 = ( SEARCH_HINTS ); 1.2.840.113556.1.4.2211 = ( EXPECTED_ENTRY_COUNT );
        1.2.840.113556.1.4.2239 = ( POLICY_HINTS ); 1.2.840.113556.1.4.2255; 1.2.840.113556.1.4.2256; 1.2.840.113556.1.4.2309;
    supportedLDAPPolicies (20): MaxPoolThreads; MaxPercentDirSyncRequests; MaxDatagramRecv; MaxReceiveBuffer; InitRecvTimeout; MaxConnections; MaxConnIdleTime;
        MaxPageSize; MaxBatchReturnMessages; MaxQueryDuration; MaxDirSyncDuration; MaxTempTableSize; MaxResultSetSize; MinResultSets; MaxResultSetsPerConn;
        MaxNotificationPerConn; MaxValRange; MaxValRangeTransitive; ThreadMemoryLimit; SystemMemoryLimitPercent;
    supportedLDAPVersion (2): 3; 2;
    supportedSASLMechanisms (4): GSSAPI; GSS-SPNEGO; EXTERNAL; DIGEST-MD5;
```

Trust the Active Directories certificate

Note: This is required for self-signed certificates. This step can be skipped if a purchased SSL certificate is in use.

On the Windows host:

1. Navigate to **Server Manager** > **Tools** > **Certificate Authority**.
2. Click **Action** > **Properties**.
3. Click **General Tab** > **View Certificate** > **Details** > **Copy to File**.
4. Choose the format: Base-64 encoded X.509 (.CER).
5. Choose a file name. For example, hdpad.cer, and save.
6. Open with Notepad and copy contents.

On the Linux host:

1. Create file `/etc/pki/ca-trust/source/anchors/hdpad.cer` and paste in the certificate contents.

2. Trust the CA certificate:

```
yum install openldap-clients ca-certificates
yum update-ca-trust enable
yum update-ca-trust extract
yum update-ca-trust check
```

3. Trust the CA certificate in Java:

```
/usr/jdk64/jdk1.8.0_112/bin/keytool -importcert -noprompt -storepass
changeit -file /etc/pki/ca-trust/source/anchors/hdpad.crt -alias ad
-keystore /etc/pki/java/cacerts
```

```
[root@c902f14x12 ~]# ambari-server setup-security
Using python  /usr/bin/python
Security setup options...
===========================================================================
Choose one of the following options:
  [1] Enable HTTPS for Ambari server.
  [2] Encrypt passwords stored in ambari.properties file.
  [3] Setup Ambari kerberos JAAS configuration.
  [4] Setup truststore.
  [5] Import certificate to truststore.
===========================================================================
Enter choice, (1-5): 4
Do you want to configure a truststore [y/n] (y)? y
TrustStore type [jks/jceks/pkcs12] (jks):jks
Path to TrustStore file :/etc/ambari-server/conf/hdpad.jks
Password for TrustStore:
Re-enter password:
Ambari Server 'setup-security' completed successfully.
```

```
[root@ c902f14x12 ~]# ambari-server setup-security
Using python  /usr/bin/python
Security setup options...
===========================================================================
Choose one of the following options:
  [1] Enable HTTPS for Ambari server.
  [2] Encrypt passwords stored in ambari.properties file.
  [3] Setup Ambari kerberos JAAS configuration.
  [4] Setup truststore.
  [5] Import certificate to truststore.
===========================================================================
Enter choice, (1-5): 5
Do you want to configure a truststore [y/n] (y)? y
Do you want to import a certificate [y/n] (y)? y
Please enter an alias for the certificate: ad
Enter path to certificate: /etc/pki/ca-trust/source/anchors/hdpad.crt
Ambari Server 'setup-security' completed successfully.
```

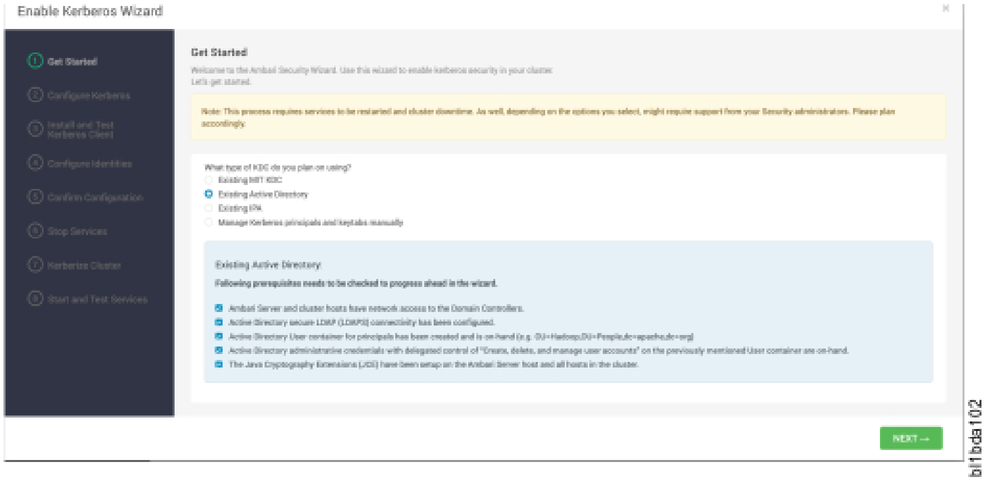
```
[root@c902f14x12 ~]# ambari-server restart
Using python  /usr/bin/python
Restarting ambari-server
Waiting for server stop...
Ambari Server stopped
Ambari Server running with administrator privileges.
Organizing resource files at /var/lib/ambari-server/resources...
Ambari database consistency check started...
Server PID at: /var/run/ambari-server/ambari-server.pid
Server out at: /var/log/ambari-server/ambari-server.out
Server log at: /var/log/ambari-server/ambari-server.log
Waiting for server start................
Server started listening on 8080

DB configs consistency check: no errors and warnings were found.
```

Enable Kerberos in Ambari

1. Open Ambari in your browser.
2. Ensure that all services are working before proceeding.
3. Click **Admin** > **Kerberos**.

4. On the Getting Started page, choose Existing Active Directory and make sure that all of the requirements are met.



Enable Kerberos Wizard

Get Started

Welcome to the Ambari Security Wizard. Use this wizard to enable kerberos security in your cluster.
Let's get started.

Note: This process requires services to be restarted and cluster downtime. As well, depending on the options you select, might require support from your Security administrators. Please plan accordingly.

What type of KDC do you plan on using?

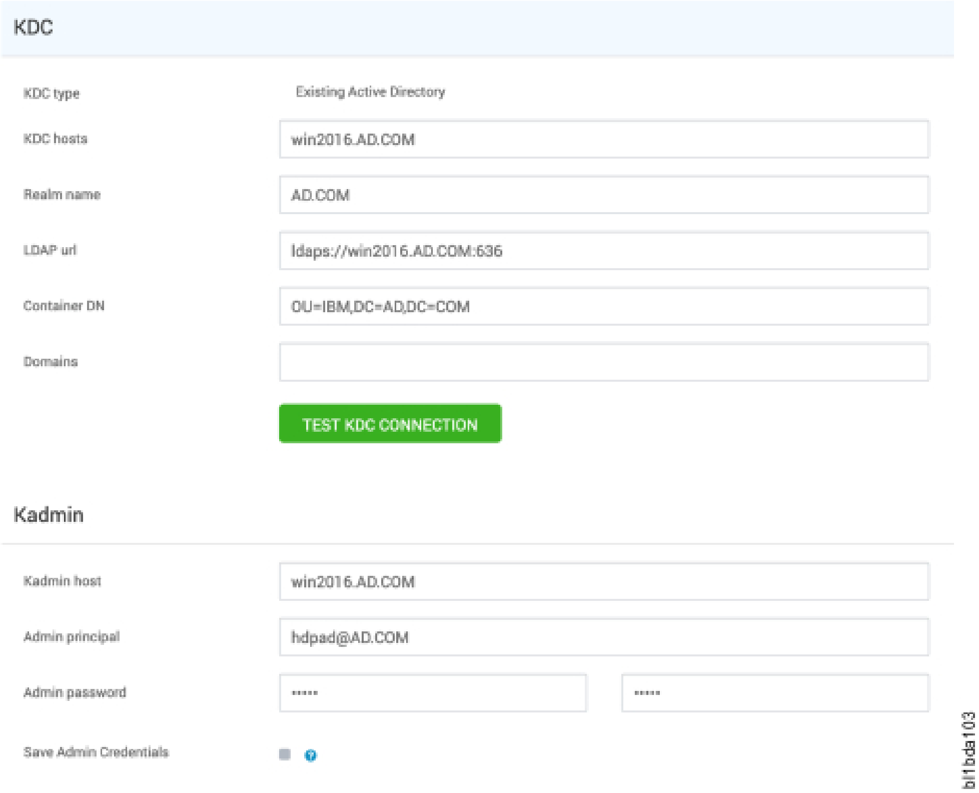
- ○ Existing MIT KDC
- ● Existing Active Directory
- ○ Existing IPA
- ○ Manage Kerberos principals and keytabs manually

Existing Active Directory

Following prerequisites needs to be checked to progress ahead in the wizard.

- ☑ Ambari Server and cluster hosts have network access to the Domain Controllers.
- ☑ Active Directory secure LDAP (LDAPS) connectivity has been configured.
- ☑ Active Directory User container for principals has been created and is on-hand (e.g. OU=Hadoop,OU=People,dc=apache,dc=org)
- ☑ Active Directory administrative credentials with delegated control of "Create, delete, and manage user accounts" on the previously mentioned User container are on-hand.
- ☑ The Java Cryptography Extensions (JCE) have been setup on the Ambari Server host and all hosts in the cluster.

5. On the Configure Kerberos page, set the following configurations:



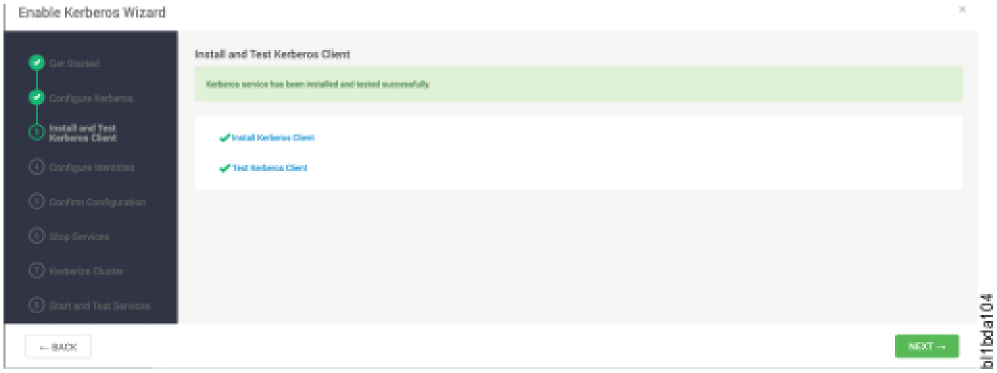
KDC

KDC type	Existing Active Directory
KDC hosts	win2016.AD.COM
Realm name	AD.COM
LDAP url	ldaps://win2016.AD.COM:636
Container DN	OU=IBM,DC=AD,DC=COM
Domains	

Kadmin

Kadmin host	win2016.AD.COM
Admin principal	hdpad@AD.COM
Admin password	••••• •••••
Save Admin Credentials	□

Follow the Ambari GUI to setup Kerberos.



Install and Test Kerberos Client

Kerberos service has been installed and tested successfully.

✓ Install Kerberos Client

✓ Test Kerberos Client

Create a one-way trust from an MIT KDC to Active Directory

Instead of using the KDC of Active Directory server to manage service principals, use a local MIT KDC in the Hadoop cluster to manage the service principals while using a one-way trust to allow AD users to utilize the Hadoop environment.

Prerequisites

Before setting up a one-way trust from an MIT KDC to Active Directory, ensure the following:

1. Existing HDP cluster has enabled Kerberos with an MIT KDC.
2. Existing AD server (or creating a new one) is running and promoted to a Domain Controller.

In this example, the following information is used:

```
MIT KDC realm name: IBM.COM
MIT KDC server name: c902f05x04.gpfs.net
AD domain/realm: AD.GPFS.NET
```

Configure the Trust in Active Directory

On the AD server, run the following command in a command window with Administrator privilege and create a definition for the KDC of the MIT realm.

```
ksetup /addkdc IBM.COM c902f05x04.gpfs.net
```

On the AD server, create an entry for the one-way trust.

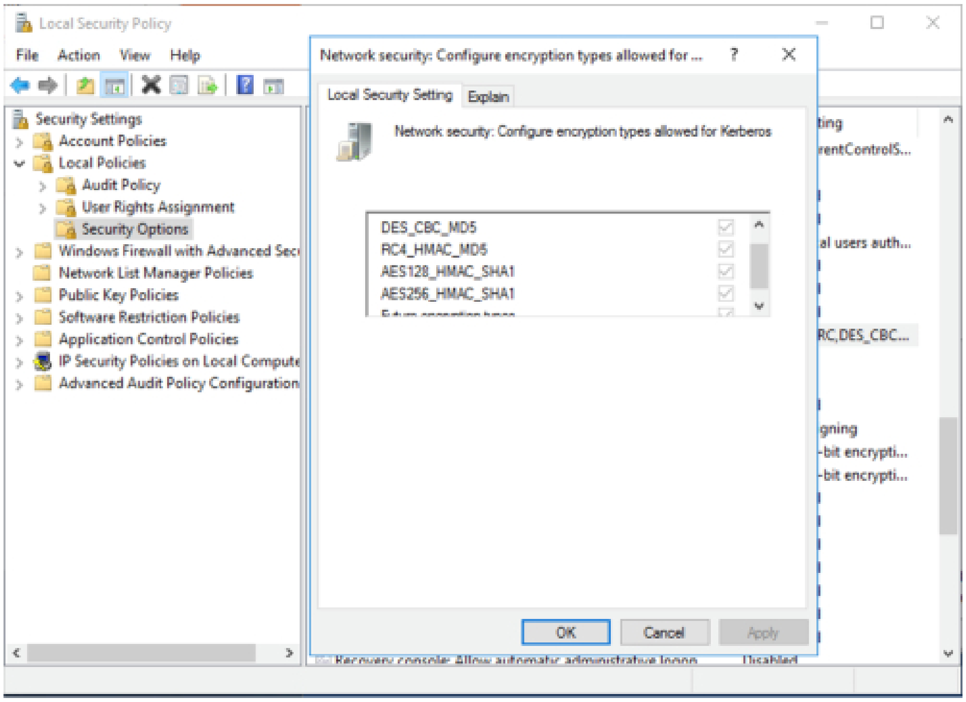
Note: The password used here will be used later in the MIT KDC configuration of the trust.

```
netdom trust IBM.COM /Domain:AD.GPFS.NET /add /realm /passwordt:Admin1234
```

Configure Encryption Types

The encryption types between both KDCs (AD KDC and MIT KDC) must be compatible, so that the tickets generated by AD KDC can be trusted by the MIT realm. There must be at least one encryption type that is accepted by both KDCs.

Review the encryption types in **Local Security Policy** > **Local Policies** > **Security Options** > **Network security**: Configure encryption types allowed for **Kerberos** > **Local Security Setting**.


Local Security Policy
File Action View Help
Security Settings
Account Policies
Local Policies
Audit Policy
User Rights Assignment
Security Options
Windows Firewall with Advanced Secu
Network List Manager Policies
Public Key Policies
Software Restriction Policies
Application Control Policies
IP Security Policies on Local Compute
Advanced Audit Policy Configuration
Network security: Configure encryption types allowed for ...
Local Security Setting
Explain
OK
Cancel
Apply
ting
rentControlS...
al users auth...
RC,DES_CBC...
gning
-bit encrypti...
-bit encrypti...
Recovery console: Allow automatic administrative logon
Disabled

Network security: Configure encryption types allowed for Kerberos

- ☑ DES_CBC_MD5
- ☑ RC4_HMAC_MD5
- ☑ AES128_HMAC_SHA1
- ☑ AES256_HMAC_SHA1
- ☑ Future encryption types

bl1bda105

On the AD server, specify which encryption types are acceptable for communication with the MIT realm.

```
ksetup /SetEncTypeAttr IBM.COM AES256-CTS-HMAC-SHA1-96
AES128-CTS-HMAC-SHA1-96 DES-CBC-MD5 DES-CBC-CRC RC4-HMAC-MD5
```

On the MIT KDC server, change the /etc/krb5.conf file to specify encryption types in MIT KDC. By default, all of the encryption types are accepted by the MIT KDC.

```
[libdefaults]
permitted_enctypes = aes256-cts-hmac-sha1-96 aes128-cts-hmac-sha1-96 des3-cbc-sha1 rc4 des-cbc-
md5
```

Enable Trust in MIT KDC
Add the trust to MIT KDC to complete the trust configuration.

In the `/etc/krb5.conf` file, add the AD domain.

In this example, domain AD.GPFS.NET is the added AD domain.

```
[realms]
  IBM.COM = {
  admin_server = c902f05x04.gpfs.net
  kdc = c902f05x04.gpfs.net
  }

  AD.GPFS.NET = {
    kdc = adserver.ad.gpfs.net
    admin_server = adserver.ad.gpfs.net
  }
```

On the MIT KDC server, create a principal that combines the realms in the trust.

Note: The password for this principal must be the same as the password used to create the trust on the AD server.

```
[root@c902f05x04 ~]# kadmin.local
Authenticating as principal nn/admin@AD.GPFS.NET with password.
kadmin.local:  addprinc krbtgt/IBM.COM@AD.GPFS.NET
WARNING: no policy specified for krbtgt/IBM.COM@AD.GPFS.NET; defaulting to no policy
Enter password for principal "krbtgt/IBM.COM@AD.GPFS.NET":
Re-enter password for principal "krbtgt/IBM.COM@AD.GPFS.NET":
```

```
add_principal: Principal or policy already exists while creating "krbtgt/IBM.COM@AD.GPFS.NET".
kadmin.local:
```

Configure AUTH_TO_LOCAL

In Ambari or in the core-site.xml file, add Auth_To_Local rules to properly convert the user principals from the AD domain to usable usernames in the Hadoop cluster.

```
    <property>
      <name>hadoop.security.auth_to_local</name>
      <value>
RULE:[1:$1@$0](^.*@AD.GPFS.NET$)s/^(.*)@AD.GPFS.NET$/$1/g
RULE:[2:$1@$0](^.*@AD.GPFS.NET$)s/^(.*)@AD.GPFS.NET$/$1/g
DEFAULT</value>
    </property>
```

Configure Transparency with Active Directory

This section describes how to configure HDFS Transparency (without HDP) with Active Directory.

To enable Kerberos with existing AD for Transparency, follow "Enable Kerberos in Ambari" on page 98 section.

Create Domain Users and export keytab file

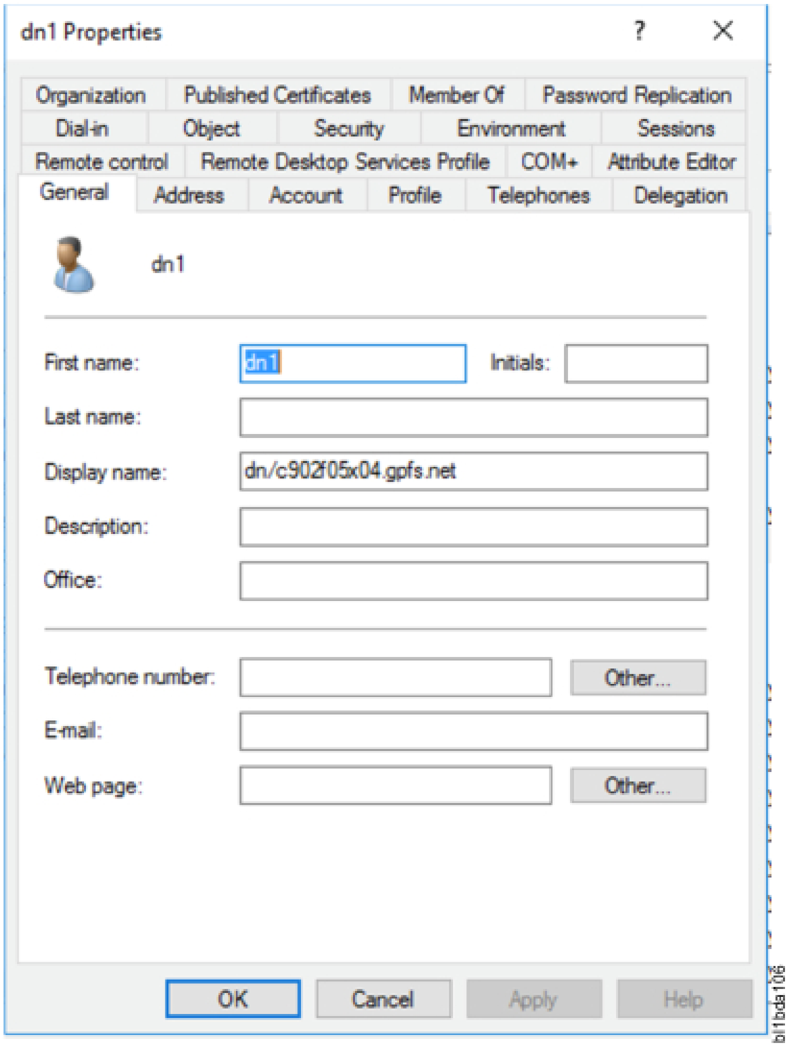
On the AD server, create Domain users for all the DataNodes and NameNodes for the HDFS Transparency cluster.

For example, the cluster contains:

```
Two namenodes: c902f08x06 and c902f08x07
Four datanodes, c902f08x05 – 08
```

Create the following users in AD for the HDFS Transparency cluster:

```
nn1 with Display name, nn/c902f08x06.gpfs.net, and User logon name, nn/
c902f08x06.gpfs.net@ad.gpfs.net
nn2 with Display name, nn/c902f08x07.gpfs.net, and User logon name, nn/
c902f08x07.gpfs.net@ad.gpfs.net
dn1 with Display name, dn/c902f08x05.gpfs.net, and User logon name, dn/
c902f08x05.gpfs.net@ad.gpfs.net
dn2 with Display name, dn/c902f08x06.gpfs.net, and User logon name, dn/
c902f08x06.gpfs.net@ad.gpfs.net
dn3 with Display name, dn/c902f08x07.gpfs.net, and User logon name, dn/
c902f08x07.gpfs.net@ad.gpfs.net
dn4 with Display name, dn/c902f08x08.gpfs.net, and User logon name, dn/
c902f08x08.gpfs.net@ad.gpfs.net
```



dn1 Properties

dn1

First name: dn1

Initials:

Last name:

Display name: dn/c902f05x04.gpfs.net

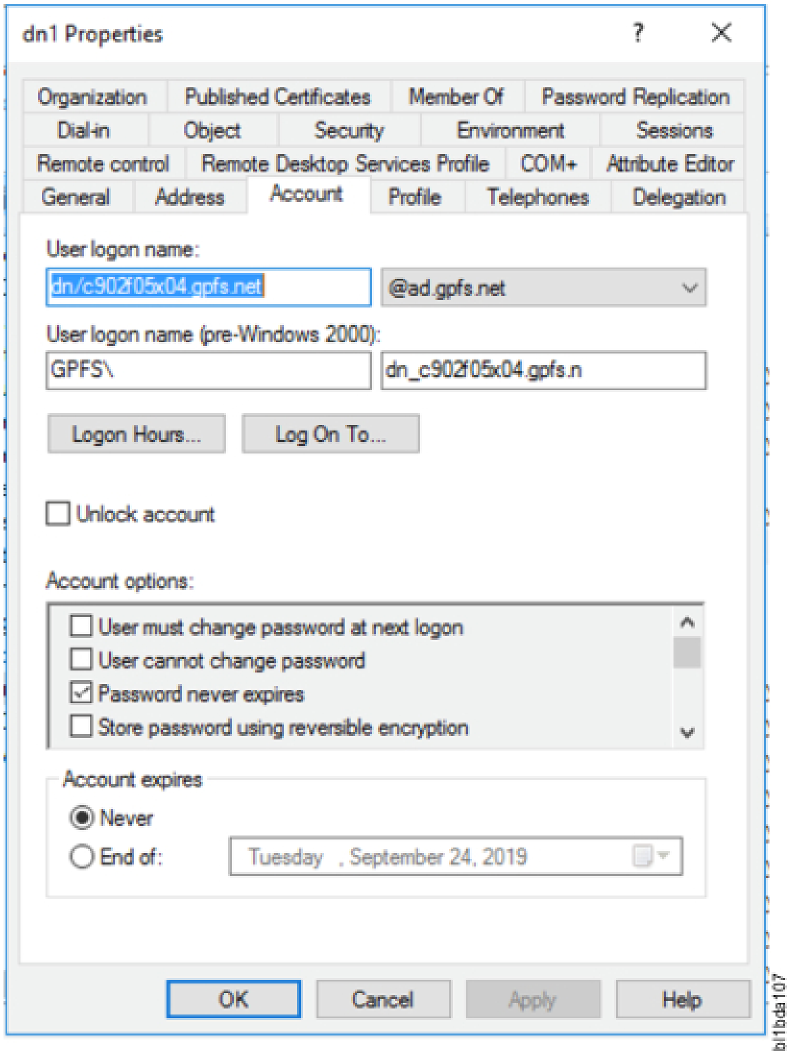
Description:

Office:

Telephone number:

E-mail:

Web page:



User logon name:
dn/c902f05x04.gpfs.net @ad.gpfs.net

User logon name (pre-Windows 2000):
GPFS\ dn_c902f05x04.gpfs.n

□ Unlock account

Account options:

- □ User must change password at next logon
- □ User cannot change password
- ☑ Password never expires
- □ Store password using reversible encryption

Account expires

● Never

○ End of: Tuesday , September 24, 2019

For the Account options, ensure to do the following:

- Un-select “User must change password at next logon”
- Select “Password never expires”
- Select “This account supports Kerberos AES 128 bit encryption”
- Select “This account supports Kerberos AES 256 bit encryption”

Export keytab files
Each node requires to generate its own corresponding key.tab.

This example below generates the keytab just for host c902f08x06.

To do another host, c902f08x07, will need to generate a new keytab with a new name like dn3.key.

On the Windows PowerShell, use the **ktpass** command to generate the principals and keytab files for all the Domain Users on the HDFS Transparency cluster.

```
PS C:\files> ktpass /princ dn/c902f08x06.gpfs.net@AD.GPFS.NET
/mapuser dn/c902f08x06.gpfs.net /pass Admin1234 /out dn2.key /ptype  KRB5_NT_SRV_INST /crypto
all
Targeting domain controller: adserver.ad.gpfs.net
Successfully mapped dn/c902f08x06.gpfs.net to dn_c902f08x06.gpfs.n.
Password successfully set!
WARNING: pType and account type do not match. This might cause problems.
Key created.
Output keytab to dn2.key:
Keytab version: 0x502
keysize 69 dn/c902f08x06.gpfs.net@AD.GPFS.NET ptype 2 (KRB5_NT_SRV_INST)
```

```
vno 3 etype 0x17 (RC4-HMAC) keylength 16 (0xdac3a2930fc196001f3aeab959748448)
PS C:\files>
```

Note: The /crypto specifies the keys that are generated in the keytab file. The default settings are based on older MIT versions. Therefore, /crypto should always be specified.

Distribute all the keytab files to the HDFS Transparency nodes and rename them to nn.service.keytab (for NameNode service) and dn.service.keytab (for DataNode service).

```
PS C:\files> .\pscp.exe dn2.key root@c902f08x06:/etc/security/keytabs/dn.service.keytab
```

Note: Ensure that the “dn2.key” exported corresponds to the host c902f08x06. Otherwise, the service will fail to start.

On the Linux nodes, change the owner and permissions for all the keytab files.

```
chown hdfs:hadoop /etc/security/keytabs/nn.service.keytab
chown hdfs:hadoop /etc/security/keytabs/dn.service.keytab
chmod 400 /etc/security/keytabs/nn.service.keytab
chmod 400 /etc/security/keytabs/dn.service.keytab
```

Install Kerberos clients and configure onto all the Linux nodes
On all nodes, run the command **`yum install krb5-workstation`** to install the Kerberos workstation.

Add the configuration information below to the `/etc/krb5.conf` file onto all the nodes.

```
[realms]
  AD.GPFS.NET = {
    kdc = adserver.ad.gpfs.net
    admin_server = adserver.ad.gpfs.net
  }
```

Configure Transparency to use Kerberos authentication
In `/usr/lpp/mmfs/hadoop/etc/hadoop/core-site.xml`, add or modify the configuration fields below based on your environment:

```
    <property>
      <name>hadoop.security.auth_to_local</name>
      <value>RULE:[2:$1@$0](dn@AD.GPFS.NET)s/.*/hdfs/
RULE:[2:$1@$0](nn@AD.GPFS.NET)s/.*/hdfs/
DEFAULT</value>
    </property>

    <property>
      <name>hadoop.security.authentication</name>
      <value>kerberos</value>
    </property>

    <property>
      <name>hadoop.security.authorization</name>
      <value>true</value>
    </property>
```

In `/usr/lpp/mmfs/hadoop/etc/hadoop/hdfs-site.xml`, add the configurations below based on your environment:

```
    <property>
      <name>dfs.datanode.kerberos.principal</name>
      <value>dn/_HOST@AD.GPFS.NET</value>
    </property>

    <property>
      <name>dfs.datanode.keytab.file</name>
      <value>/etc/security/keytabs/dn.service.keytab</value>
    </property>

    <property>
      <name>dfs.namenode.kerberos.principal</name>
      <value>nn/_HOST@AD.GPFS.NET</value>
    </property>

    <property>
```

```
      <name>dfs.namenode.keytab.file</name>
      <value>/etc/security/keytabs/nn.service.keytab</value>
    </property>
```

Use **mmhadoopctl connector syncconf /usr/lpp/mmfs/hadoop/etc/hadoop** (for HDFS Transparency 2.7.3-x) or **mmhadoopctl connector syncconf /var/mmfs/hadoop/etc/hadoop** (for HDFS Transparency 3.0.x/3.1.x) to sync the configuration files onto all cluster nodes.

Note:

- For HDFS Transparency 2.7.3-x, the configuration is stored in /usr/lpp/mmfs/hadoop/etc/hadoop.
- For HDFS Transparency 3.1.x, the configuration is stored in /var/mmfs/hadoop/etc/hadoop.

For more information, see the Sync HDFS Transparency section.

Start Transparency

As root on one of the HDFS Transparency node, run /usr/lpp/mmfs/bin/mmhadoopctl connector start to start HDFS Transparency.

Configure SSSD for Transparency

The System Security Services Daemon (SSSD) provides a set of daemons to manage access to remote directories and authentication mechanisms.

It provides Name Service Switch (NSS) and Pluggable Authentication Modules (PAM) interfaces toward the system and a pluggable back end system to connect to multiple different account sources.

SSSD installation and configuration

For Red Hat 7, install the following packages

```
yum -y install sssd realmd oddjob oddjob-mkhomedir adcli samba-common
```

Connect to an Active Directory Domain

Use the realmd to connect to an Active Directory Domain. The realmd system provides a clear and simple way to discover and join identity domains to achieve direct domain integration.

It configures underlying Linux system services, such as SSSD or Winbind, to connect to the domain.

```
realm join adserver.ad.gpfs.net -U Administrator
realm permit -g hadoop@AD.GPFS.NET
```

Configure the sudoers file to add below line:

```
%SudoNO@ad.gpfs.net  ALL=(ALL)   ALL
```

Configure /etc/sssd/sssd.conf file with the following changes:

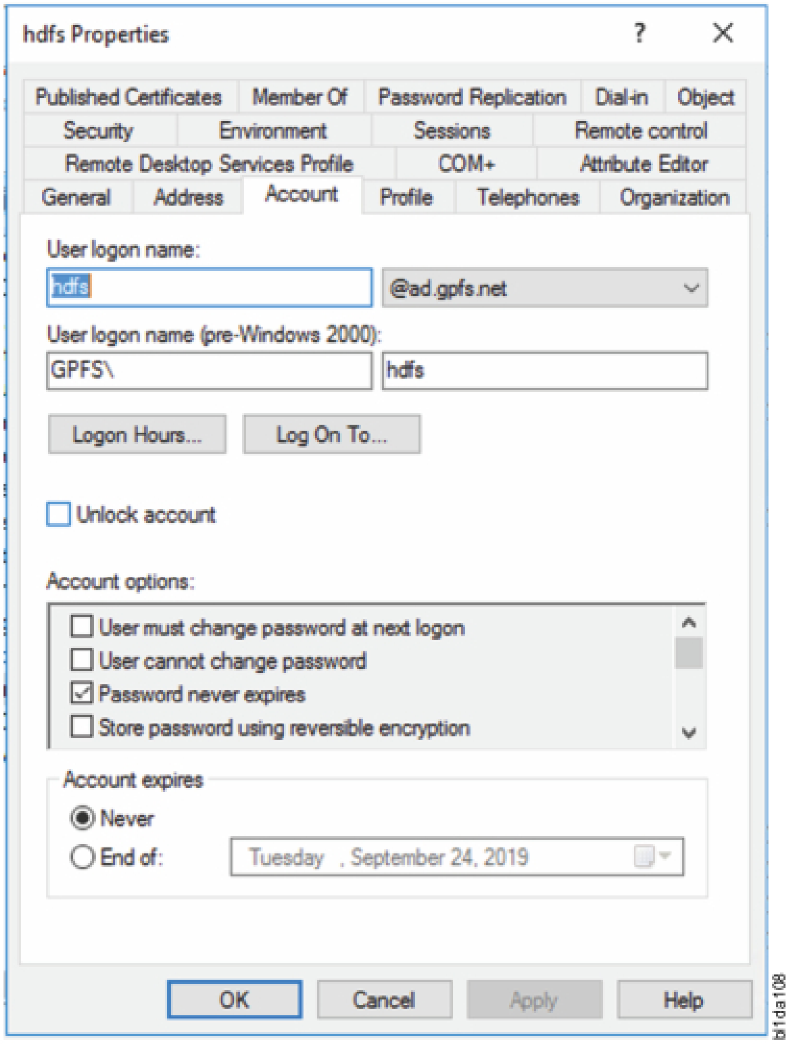
```
use_fully_qualified_names = False
fallback_homedir = /home/%u
```

Restart sssd service after changing the configuration file.

```
systemctl restart sssd
```

Test the integration of SSSD and AD

Create a User Account hdfs in AD and add it to group hadoop.



User logon name: hdfs @ad.gpfs.net

User logon name (pre-Windows 2000): GPFS\ hdfs

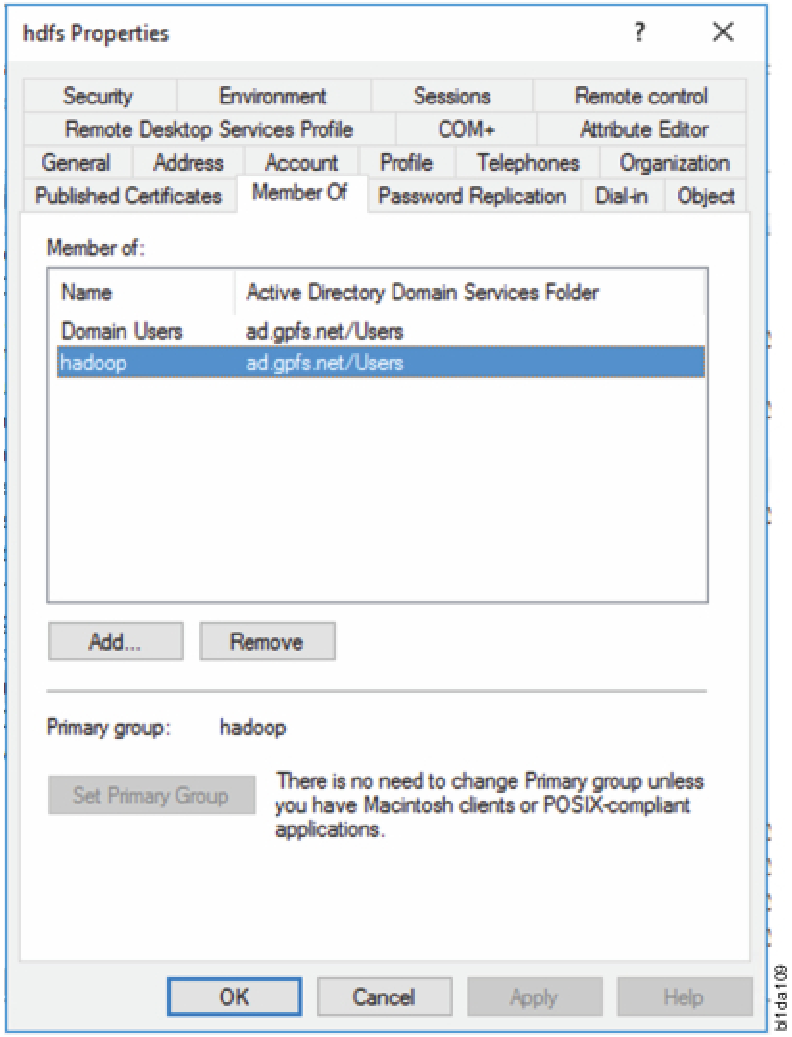
□ Unlock account

Account options:

- □ User must change password at next logon
- □ User cannot change password
- ☑ Password never expires
- □ Store password using reversible encryption

Account expires

- ● Never
- ○ End of: Tuesday , September 24, 2019



Member of:

Name	Active Directory Domain Services Folder
Domain Users	ad.gpfs.net/Users
hadoop	ad.gpfs.net/Users

Primary group: hadoop

There is no need to change Primary group unless you have Macintosh clients or POSIX-compliant applications.

On a Linux shell, run the command to verify the SSSD works properly.

```
[root@c902f05x04 ~]# id hdfs
uid=537601119(hdfs) gid=537601123(hadoop) groups=537601123(hadoop),537600513(domain users)
[root@c902f05x04 ~]#
```

MIT Kerberos

Manually configuring Kerberos

Setting up the Kerberos server

This topic lists the steps to set up the Kerberos server.

Before following these steps, see the “Prerequisites” on page 81 topic.

1. Install and configure the Kerberos server.

   ```
   yum install krb5-server krb5-libs krb5-workstation
   ```

2. Create `/etc/krb5.conf` with the following contents:

   ```
   [logging]
   default = FILE:/var/log/krb5libs.log
   kdc = FILE:/var/log/krb5kdc.log
   admin_server = FILE:/var/log/kadmind.log
   ```

```
[libdefaults]
default_realm = IBM.COM
dns_lookup_realm = false
dns_lookup_kdc = false
ticket_lifetime = 24h
renew_lifetime = 7d
forwardable = true
default_realm = IBM.COM

[realms]
IBM.COM =  {
    kdc = {KDC_HOST_NAME}
    admin_server = {KDC_HOST_NAME}
    }

[domain_realm]
    .ibm.com = IBM.COM
    ibm.com = IBM.COM
```

Note: The KDC_HOST_NAME, KDC_HOST_NAME and IBM.COM should reflect the correct host and REALM based on your environment.

3. Set up the server.

   ```
       kdb5_util create -s

       systemctl start krb5kdc
       systemctl start kadmin
       chkconfig krb5kdc on
       chkconfig kadmin on
   ```

4. Add the admin principal, and set the password.

   ```
   kadmin.local -q "addprinc root/admin"
   ```

   Check the `kadm5.acl` to ensure that the entry is correct.

   ```
   cat /var/kerberos/krb5kdc/kadm5.acl
   */admin@IBM.COM

   systemctl restart krb5kdc.service

   systemctl restart kadmin.service
   ```

5. Ensure that the password is working by running the following command:

   ```
   kadmin -p root/admin@IBM.COM
   ```

Setting up Kerberos for HDFS Transparency nodes

This topic lists the steps to set up the Kerberos clients on the HDFS Transparency nodes. These instructions work for both Cloudera Private Cloud Base and Apache Hadoop distributions.

Before following these steps, see the "Prerequisites" on page 81 topic.

1. Install the Kerberos clients package on all the HDFS Transparency nodes.

   ```
   yum install -y krb5-libs krb5-workstation
   ```

2. Copy the `/etc/krb5.conf` file to the Kerberos client hosts on the HDFS Transparency nodes.
3. Create a directory for the keytab directory and set the appropriate permissions on each of the HDFS Transparency node.

   ```
   mkdir -p /etc/security/keytabs/
   chown root:root /etc/security/keytabs
   chmod 755 /etc/security/keytabs
   ```

4. Create KDC principals for the components, corresponding to the hosts where they are running, and export the keytab files as follows:

Service	User:Group	Daemons	Principal	Keytab File Name
HDFS	root:root	NameNode	nn/<NN_Host_FQDN>@<REALM-NAME>	nn.service.keytab
		NameNode HTTP	HTTP/<NN_Host_FQDN>@<REALM-NAME>	spnego.service.keytab
		NameNode HTTP	HTTP/<CES_HDFS_Host_FQDN>@<REALM-NAME>	spnego.service.keytab
		DataNode	dn/<DN_Host_FQDN>@<REALM-NAME>	dn.service.keytab

Replace the < NN_Host_FQDN > with the HDFS Transparency NameNode hostname and the <DN_Host_FQDN> with the HDFS Transparency DataNode hostname. Replace the <CES_HDFS_Host_FQDN> with the CES hostname configured for your CES HDFS cluster.

You need to create one principal for each HDFS Transparency DataNode and two principals for each HDFS Transparency NameNode.

Note: If you are using CDP Private Cloud Base, Cloudera Manager creates the principals and keytabs for all the services except the IBM Storage Scale service. Therefore, you can skip the create service principals section below and go directly to step a.

If you are using Apache Hadoop, you need to create service principals for YARN and Mapreduce services as shown in the following table:

Service	User:Group	Daemons	Principal	Keytab File Name
YARN	yarn:hadoop	ResourceManager	rm/<Resource_Manager_FQDN>@<REALM-NAME>	rm.service.keytab
		NodeManager	nm/<Node_Manager_FQDN>@<REALM-NAME>	nm.service.keytab
Mapreduce	mapred:hadoop	MapReduce Job History Server	jhs/<Job_History_Server_FQDN>@<REALM-NAME>	jhs.service.keytab

Replace the <Resource_Manager_FQDN> with the Resource Manager hostname, the <Node_Manager_FQDN> with the Node Manager hostname and the <Job_History_Server_FQDN> with the Job History Server hostname.

a. Create service principals for each service. Refer to the sample table above.

```
kadmin.local -q "addprinc -randkey  -maxrenewlife 7d +allow_renewable {Principal}"
```

For example:

```
kadmin.local -q "addprinc -randkey -maxrenewlife 7d +allow_renewable nn/
nn01.gpfs.net@IBM.COM"
```

b. Create host principals for each Transparency host.

```
kadmin.local -q "addprinc -randkey host/{HOST_NAME}@<Realm Name>"
```

For example:

```
kadmin.local -q "addprinc -randkey host/nn01.gpfs.net@IBM.COM"
```

c. If you are using RHEL 9.1+ for Power LE, update the principals to include the `+requires_preauth` attribute.

For all the host and service principals created under the previous steps 4.a and 4.b, update the principals to include the `+requires_preauth` flag, as shown in the following example:

```
# kadmin.local: modify_principal +requires_preauth nn/nn01.gpfs.net@IBM.COM
Principal nn/nn01.gpfs.net@IBM.COM modified
```

d. For each service on each Transparency host, create a keytab file by exporting its service principal into a keytab file:

```
kadmin.local ktadd -k
/etc/security/keytabs/{SERVICE_NAME}.service.keytab {Principal}
```

For example:

DataNode:

```
kadmin.local ktadd -k /etc/security/keytabs/dn.service.keytab dn/dn01.gpfs.net@IBM.COM
```

NameNode:

```
kadmin.local ktadd -k /etc/security/keytabs/nn.service.keytab nn/nn01.gpfs.net@IBM.COM
```

NameNode HTTP:

The keytab for this service needs an additional step as it contains entries for two principals – one corresponding to the actual NameNode hostname and another for the CES IP hostname.

- First create the keytab file for HTTP service corresponding to the NameNode host.

  ```
  kadmin.local ktadd -k /etc/security/keytabs/spnego.service.keytab HTTP/
  nn01.gpfs.net@IBM.COM
  ```

- Create a temporary keytab file for HTTP service corresponding to the CES HDFS IP hostname.

  ```
  kadmin.local ktadd -norandkey -k /etc/security/keytabs/myceshdfs.service.keytab HTTP/
  myceshdfs.gpfs.net@IBM.COM
  ```

- Merge the above two keytabs with kutil utility to create an updated spnego.service.keytab:

  ```
  #ktutil
  ktutil: rkt /etc/security/keytabs/myceshdfs.service.keytab
  ktutil: wkt /etc/security/keytabs/spnego.service.keytab
  exit
  ```

  Note: **`myceshdfs.gpfs.net`** is an example of the CES IP hostname configured for your CES HDFS service.

- Repeat the "4.a" on page 110, "4.b" on page 111, and "4.d" on page 111 steps for every required keytab file.

Note:

- The filename for a service is common (for example, **`dn.service.keytab`**) across hosts but the contents would be different because every keytab would have a different host principal component.

- After a keytab is generated, move the keytab to the appropriate host immediately or move it into a different location to avoid the keytab from getting overwritten.

5. For CES HDFS NameNode HA, an HDFS admin user and its Kerberos user principal and keytab are required to be created and setup for the CES NameNodes. These credentials are used by the CES framework to elect an active NameNode.

   This principal should map to an existing OS user on the NameNode hosts.

   In this example, the OS user is *hdfs*. You will configure this principal/keytab into hadoop-env.sh in step 8.

   a. First create a Hadoop supergroup.

      Set the **dfs.permissions.superusergroup** parameter to *supergroup* by running the following command:

      ```
      /usr/lpp/mmfs/hadoop/sbin/mmhdfs config set hdfs-site.xml -k
      dfs.permissions.superusergroup=supergroup
      ```

   b. Create the *hdfs* user on all the HDFS Transparency nodes that belongs to the *supergroup* Hadoop super group by using the supplied **gpfs_create_hadoop_users_dirs.py** command.

      The command ensures that the custom user/group is created with consistent UID/GID across all the nodes.

      ```
      /usr/lpp/mmfs/hadoop/scripts/gpfs_create_hadoop_users_dirs.py --create-custom-hadoop-
      user-group hdfs:supergroup
      ```

      Note: If you are going to use CDP, you can skip this step. You will create this user as part of the CDP specific configuration workflow.

   c. Create the user principal.

      ```
      # kadmin.local "addprinc -randkey -maxrenewlife 7d +allow_renewable ces-
      <clustername>@IBM.COM"
      # kadmin.local "ktadd -k /etc/security/keytabs/ces-<clustername>.headless.keytab ces-
      <clustername>@IBM.COM"
      ```

      where, **<clustername>** is the name of your CES HDFS cluster. In case there are multiple CES HDFS clusters sharing a common KDC server, having the cluster name as part of the principal helps to create a user principal unique to each CES HDFS cluster.

   d. Copy the `/etc/security/keytabs/ces-<clustername>.headless.keytab` file to all the NameNodes and change the owner permission of the file to root:

      ```
      # chown root:root /etc/security/keytabs/ces-<clustername>.headless.keytab
      # chmod 400 /etc/security/keytabs/ces-<clustername>.headless.keytab
      ```

6. Copy the appropriate keytab file to each host. If a host runs more than one component (for example, both NameNode and DataNode), copy the keytabs for both these components.

7. Set the appropriate permissions for the keytab files.

   On the HDFS Transparency NameNode hosts:

   ```
   chown root:root /etc/security/keytabs/nn.service.keytab
   chmod 400 /etc/security/keytabs/nn.service.keytab
   chown root:root /etc/security/keytabs/spnego.service.keytab
   chmod 440 /etc/security/keytabs/spnego.service.keytab
   ```

   On the HDFS Transparency DataNode hosts:

   ```
   chown root:root /etc/security/keytabs/dn.service.keytab
   chmod 400 /etc/security/keytabs/dn.service.keytab
   ```

   On the Yarn resource manager hosts:

```
chown yarn:hadoop /etc/security/keytabs/rm.service.keytab
chmod 400 /etc/security/keytabs/rm.service.keytab
```

On the Yarn node manager hosts:

```
chown yarn:hadoop /etc/security/keytabs/nm.service.keytab
chmod 400 /etc/security/keytabs/nm.service.keytab
```

On Mapreduce job history server hosts:

```
chown mapred:hadoop /etc/security/keytabs/jhs.service.keytab
chmod 400 /etc/security/keytabs/jhs.service.keytab
```

8. Update the HDFS Transparency configuration files and upload the changes.

   - Get the config files

     ```
     mkdir /tmp/hdfsconf
     mmhdfs config export /tmp/hdfsconf core-site.xml,hdfs-site.xml,hadoop-env.sh
     ```

   - Configurations in `core-site.xml` and `hdfs-site.xml` are different for HDFS Transparency 3.1.x and HDFS Transparency 3.2.2-x/3.3.x. The configurations are as follows:

     – For HDFS Transparency 3.1.x use the following configurations in `core-site.xml` and `hdfs-site.xml`:

       `File: core-site.xml`

       ```
       <property>
          <name>hadoop.security.authentication</name>
          <value>kerberos</value>
       </property>

       <property>
          <name>hadoop.rpc.protection</name>
          <value>authentication</value>
       </property>
       ```

       If you are using Cloudera Private Cloud Base cluster, create the following rules:

       ```
       <property>
         <name>hadoop.security.auth_to_local</name>
         <value>
         RULE:[2:$1/$2@$0](nn/.*@.*IBM.COM)s/.*/hdfs/
         RULE:[2:$1/$2@$0](dn/.*@.*IBM.COM)s/.*/hdfs/
         RULE:[1:$1@$0](ces-<clustername>@IBM.COM)s/.*/hdfs/
         RULE:[1:$1@$0](.*@IBM.COM)s/@.*//
         DEFAULT
         </value>
       </property>
       ```

       Otherwise, if you are using Apache Hadoop, create the following rules:

       ```
       <property>
         <name>hadoop.security.auth_to_local</name>
         <value>
           RULE:[2:$1/$2@$0](nn/.*@.*IBM.COM)s/.*/hdfs/
           RULE:[2:$1/$2@$0](dn/.*@.*IBM.COM)s/.*/hdfs/
           RULE:[2:$1/$2@$0](nm/.*@.*IBM.COM)s/.*/yarn/
           RULE:[2:$1/$2@$0](rm/.*@.*IBM.COM)s/.*/yarn/
           RULE:[2:$1/$2@$0](jhs/.*@.*IBM.COM)s/.*/mapred/
           RULE:[1:$1@$0](ces-<clustername>@IBM.COM)s/.*/hdfs/
           DEFAULT
         </value>
       </property>
       ```

       In the above example, replace IBM.COM with your Realm name and *<clustername>* parameter with your actual CES HDFS cluster name.

`File: hdfs-site.xml`

```
<property>
  <name>dfs.data.transfer.protection</name>
  <value>authentication</value>
</property>

<property>
  <name>dfs.datanode.address</name>
  <value>0.0.0.0:1004</value>
</property>

<property>
  <name>dfs.datanode.data.dir.perm</name>
  <value>700</value>
</property>

<property>
  <name>dfs.datanode.http.address</name>
  <value>0.0.0.0:1006</value>
</property>

<property>
  <name>dfs.datanode.kerberos.principal</name>
  <value>dn/_HOST@IBM.COM</value>
</property>

<property>
  <name>dfs.datanode.keytab.file</name>
  <value>/etc/security/keytabs/dn.service.keytab</value>
</property>

<property>
  <name>dfs.encrypt.data.transfer</name>
  <value>false</value>
</property>

<property>
  <name>dfs.namenode.kerberos.internal.spnego.principal</name>
  <value>HTTP/_HOST@IBM.COM</value>
</property>

<property>
  <name>dfs.namenode.kerberos.principal</name>
  <value>nn/_HOST@IBM.COM</value>
</property>

<property>
  <name>dfs.namenode.keytab.file</name>
  <value>/etc/security/keytabs/nn.service.keytab</value>
</property>

<property>
  <name>dfs.web.authentication.kerberos.keytab</name>
  <value>/etc/security/keytabs/spnego.service.keytab</value>
</property>

<property>
  <name>dfs.web.authentication.kerberos.principal</name>
  <value>*</value>
</property>
```

– For HDFS Transparency 3.2.2-x and 3.3.x use the following configurations in `core-site.xml` and `hdfs-site.xml`:

`File: core-site.xml`

```
<property>
   <name>hadoop.security.authentication</name>
   <value>kerberos</value>
</property>

<property>
   <name>hadoop.rpc.protection</name>
   <value>authentication</value>
</property>

<property>
   <name>hadoop.http.authentication.type</name>
   <value>kerberos</value>
</property>

<property>
   <name>hadoop.http.authentication.kerberos.principal</name>
   <value>*</value>
</property>

<property>
   <name>hadoop.http.authentication.kerberos.keytab</name>
   <value>/etc/security/keytabs/spnego.service.keytab</value>
</property>
```

If you are using Cloudera Private Cloud Base cluster, create the following rules:

```
<property>
  <name>hadoop.security.auth_to_local</name>
  <value>
   RULE:[2:$1/$2@$0](nn/.*@.*IBM.COM)s/.*/hdfs/
   RULE:[2:$1/$2@$0](dn/.*@.*IBM.COM)s/.*/hdfs/
   RULE:[1:$1@$0](ces-<clustername>@IBM.COM)s/.*/hdfs/
   RULE:[1:$1@$0](.*@IBM.COM)s/@.*//
   DEFAULT
   </value>
</property>
```

Otherwise, if you are using Apache Hadoop, create the following rules:

```
<property>
  <name>hadoop.security.auth_to_local</name>
  <value>
    RULE:[2:$1/$2@$0](nn/.*@.*IBM.COM)s/.*/hdfs/
    RULE:[2:$1/$2@$0](dn/.*@.*IBM.COM)s/.*/hdfs/
    RULE:[2:$1/$2@$0](nm/.*@.*IBM.COM)s/.*/yarn/
    RULE:[2:$1/$2@$0](rm/.*@.*IBM.COM)s/.*/yarn/
    RULE:[2:$1/$2@$0](jhs/.*@.*IBM.COM)s/.*/mapred/
    RULE:[1:$1@$0](ces-<clustername>@IBM.COM)s/.*/hdfs/
    DEFAULT
  </value>
</property>
```

In the above example, replace IBM.COM with your Realm name and *<clustername>* parameter with your actual CES HDFS cluster name.

`File: hdfs-site.xml`

```
<property>
  <name>dfs.data.transfer.protection</name>
  <value>authentication</value>
</property>

<property>
  <name>dfs.datanode.address</name>
  <value>0.0.0.0:1004</value>
</property>

<property>
  <name>dfs.datanode.data.dir.perm</name>
  <value>700</value>
</property>

<property>
  <name>dfs.datanode.http.address</name>
  <value>0.0.0.0:1006</value>
</property>

<property>
  <name>dfs.datanode.kerberos.principal</name>
  <value>dn/_HOST@IBM.COM</value>
</property>

<property>
  <name>dfs.datanode.keytab.file</name>
  <value>/etc/security/keytabs/dn.service.keytab</value>
</property>

<property>
  <name>dfs.encrypt.data.transfer</name>
  <value>false</value>
</property>

<property>
  <name>dfs.namenode.kerberos.internal.spnego.principal</name>
  <value>HTTP/_HOST@IBM.COM</value>
</property>

<property>
  <name>dfs.namenode.kerberos.principal</name>
  <value>nn/_HOST@IBM.COM</value>
</property>

<property>
  <name>dfs.namenode.keytab.file</name>
  <value>/etc/security/keytabs/nn.service.keytab</value>
</property>

<property>
  <name>dfs.block.access.token.enable</name>
  <value>true</value>
</property>
```

- `File: hadoop-env.sh`

  ```
  KINIT_KEYTAB=/etc/security/keytabs/ces-<clustername>.headless.keytab
  KINIT_PRINCIPAL=ces-<clustername>@IBM.COM
  ```

  where, *<clustername>* is the name of your CES HDFS cluster.

9. Stop the HDFS Transparency services for the cluster.

   a. Stop the DataNodes.

      On any HDFS Transparency node, run the following command:

      ```
      mmhdfs hdfs-dn stop
      ```

   b. Stop the NameNodes.

      On any CES HDFS NameNode, run the following command:

      ```
      mmces service stop HDFS -N <NN1>,<NN2>
      ```

10. Import the files.

    ```
    mmhdfs config import /tmp/hdfsconf core-site.xml,hdfs-site.xml,hadoop-env.sh
    ```

11. Upload the changes.

    ```
    mmhdfs config upload
    ```

12. Start the HDFS Transparency services for the cluster.

    a. Start the DataNodes.

       On any HDFS Transparency node, run the following command:

       ```
       mmhdfs hdfs-dn start
       ```

    b. Start the NameNodes.

       On any CES HDFS NameNode, run the following command:

       ```
       mmces service start HDFS -N <NN1>,<NN2>
       ```

    c. Verify that the services have started.

       On any CES HDFS NameNode, run the following command:

       ```
       mmhdfs hdfs status
       ```

Configuring Kerberos using the Kerberos script provided with IBM Storage Scale

From HDFS Transparency 3.1.1-3, IBM Storage Scale provides a Kerberos configuration script `/usr/lpp/mmfs/hadoop/scripts/gpfs_kerberos_configuration.py` to help with setting up Kerberos for HDFS Transparency interactively.

From HDFS Transparency 3.1.1-4, a non-interactive version of the automation script is also supported. The input parameters can be specified through a customized json input file.

The output of the script is logged to `/var/log/kerberos_configuration_setup.log` file.

Note: If you need to set up more than one HDFS Transparency cluster using a common KDC server , see the Limitation in the "Kerberos" on page 81 topic.

Before following these steps, see the Prerequisites topic.

There are two methods to use the Kerberos script:

1. Interactive method
2. Custom json file method

Interactive method

You can perform the following using the interactive method:

1. Set up a new KDC server. If you already have a KDC server, go to step 2.

   Setting up a new KDC server helps with the following:

   a. Install and configure a new Kerberos server on the host being run. Create or update the `/var/kerberos/krb5kdc/kdc.conf` and `/etc/krb5.conf` files.

   b. By default, the principals are configured such that **`ticket_lifetime`** is set to *24h* and **`renew_lifetime`** is set to *7d*. If needed, update these default values.

2. Configure Kerberos for HDFS Transparency.

   Configuring Kerberos helps with the following:

   a. Install and configure Kerberos client on the HDFS Transparency nodes.

   b. Create host principals.

   c. Create NameNode and DataNode principals and keytabs for HDFS Transparency.

   d. Create hdfs user principal and keytab.

   e. Apply the Kerberos configurations for hdfs-site.xml, core-site.xml and hadoop-env.sh for HDFS Transparency.

3. Clear Kerberos configuration from HDFS Transparency.

   Clearing Kerberos configuration helps with the following:

   a. Disable the Kerberos configurations from HDFS Transparency.

   b. In case you want to re-enable Kerberos at a later time, the existing principals and keytabs created for NameNodes and DataNodes are retained.

Perform the following to run the `gpfs_kerberos_configuration.py` script:

- For HDFS Transparency-3.1.1-3:

```
# /usr/lpp/mmfs/hadoop/scripts/gpfs_kerberos_configuration.py
      MIT Kerberos configuration:
      1: Setup a new KDC server.
           [Run the script on the KDC server host]
      2: Configure Kerberos for HDFS Transparency.
           [Run the script on a CES-HDFS cluster node that has password-less SSH access to
the other HDFS Transparency nodes]
      3: Clear Kerberos configuration from HDFS Transparency.
           [This option will remove the Kerberos configurations from your HDFS Transparency
cluster.
            This will not remove the existing principals and keytabs for NameNodes and
DataNodes]

      Choose option 1/2/3:
```

- For HDFS Transparency-3.1.1-4:

```
# /usr/lpp/mmfs/hadoop/scripts/gpfs_kerberos_configuration.py
      MIT Kerberos configuration:
      1: Setup a new KDC server.
           [Run the script on the KDC server host]
      2: Configure Kerberos for HDFS Transparency.
           [Run the script on a CES-HDFS cluster node that has password-less SSH access to 
the other HDFS Transparency nodes]
      3: Clear Kerberos configuration from HDFS Transparency.
           [This option will remove the Kerberos configurations from your HDFS Transparency 
cluster.
            This will not remove the existing principals and keytabs for NameNodes and 
DataNodes]
      4: Exit.

      Choose option 1/2/3/4:
```

Custom json file method

For this method, the user needs to update the custom json file (`/usr/lpp/mmfs/hadoop/scripts/gpfs_kerberos_config_metadata.json`) with inputs specific to the environment. Then run the `gpfs_kerberos_configuration.py` script as follows:

```
[root@scripts]# ./gpfs_kerberos_configuration.py -h
usage: gpfs_kerberos_configuration.py [-h] [-c CONFIG]Create Kerberos configurationoptional 
arguments:
  -h, --help            show this help message and exit
  -c CONFIG, --config CONFIG
                        Provide 'gpfs_kerberos_config_metadata.json' config
                        path. Help: The sample config template file can be
                        found in '/usr/lpp/mmfs/hadoop/scripts/gpfs_kerberos_c
                        onfig_metadata.json'Example:
```

```
[root@scripts]#./gpfs_kerberos_configuration.py -c /usr/lpp/mmfs/hadoop/scripts/
gpfs_kerberos_config_metadata.json
```

Verifying Kerberos

For information about verifying Kerberos, see "Verifying Kerberos" on page 127.

Workaround for the Power LE platform

On RHEL 9.1+ for Power LE, you need to run a command to modify all MIT Kerberos principals that are generated by HDFS Transparency, so that a new and mandatory attribute `+requires_preauth` can be added to the principals.

This attribute is added by default when principals are created by using the **`addprinc`** command. However, `+requires_preauth` gets missed if the `+allow_renewable` flag is also passed. For example, the HDFS Transparency script `gpfs_kerberos_configuration.py` creates NameNode principals, which causes the `+requires_preauth` flag to get missed, as shown in this example:

```
addprinc -randkey -maxrenewlife 7d +allow_renewable {Principal}
```

Solution

Modify the principals as in the following example.

Make sure to repeat this solution for all these principals:

- `nn/host`
- `HTTP/host`
- `dn/host`

- `<hostname>/host`
- `HTTP/<cesip-host>`

Example:

```
# kadmin.local:  modify_principal +requires_preauth nn/nn01.gpfs.net@IBM.COM
Principal nn/nn01.gpfs.net@IBM.COM modified.

# kadmin.local:  getprinc nn/nn01.gpfs.net@IBM.COM
Principal: nn/nn01.gpfs.net@IBM.COM
Expiration date: [never]
Last password change: Tue Aug 08 09:45:21 EDT 2023
Password expiration date: [never]
Maximum ticket life: 1 day 00:00:00
Maximum renewable life: 7 days 00:00:00
Last modified: Wed Aug 09 01:18:00 EDT 2023 (root/admin@IBM.COM)
Last successful authentication: [never]
Last failed authentication: [never]
Failed password attempts: 0
Number of keys: 5
Key: vno 4, aes256-cts-hmac-sha1-96
Key: vno 4, aes128-cts-hmac-sha1-96
Key: vno 4, DEPRECATED:arcfour-hmac
Key: vno 4, camellia256-cts-cmac
Key: vno 4, camellia128-cts-cmac
MKey: vno 1
Attributes: REQUIRES_PRE_AUTH
Policy: [none]
kadmin.local:
```

Red Hat IPA Kerberos

Setting up the IPA Kerberos server

This topic lists the steps to set up the IPA Kerberos server.

Before following these steps, see the "Prerequisites" on page 81 topic.

For the complete procedure to setup your IPA environment, see the Red Hat documentation specific to your OS version. For example, Options for the ipa-server-install and ipa-replica-install commands.

1. IPA server installation and setup.

   RHEL7 and RHEL8 configure the IPA server differently.

   - Example of RHEL7:

     Install and configure the IPA server by running the following commands:

     ```
     # yum install ipa-server
     ```

     ```
     # ipa-server-install
     ```

   - Example of RHEL8:

     In RHEL8, there is no ipa-server package provided in its repo. For the setup steps, see Preparing the system for IdM server installation.

2. Set up the IPA server by running the **`ipa-server-install`** command as follows:
3. Verify that the IPA services are up by running the **`ipactl status`** command as follows:

   ```
   #  ipactl status
   Directory Service: RUNNING
   krb5kdc Service: RUNNING
   kadmin Service: RUNNING
   httpd Service: RUNNING
   ipa-custodia Service: RUNNING
   ntpd Service: RUNNING
   pki-tomcatd Service: RUNNING
   ipa-otpd Service: RUNNING
   ipa: INFO: The ipactl command was successful
   ```

4. Ensure that the Administrator (for example, admin) is able to obtain tickets by running the following command:

   ```
   # kinit admin
   ```

Setting up IPA Kerberos for HDFS Transparency nodes

This topic lists the steps to set up the IPA Kerberos clients on the HDFS Transparency nodes.

Before following these steps, see the "Prerequisites" on page 81 topic.

For the complete procedure to setup your IPA client environment, see the Red Hat documentation specific to your OS version. For example, Preparing the system for IdM client installation.

1. Install and setup the IPA Kerberos client on all the HDFS Transparency nodes by running the following commands:

   ```
   # yum install ipa-client
   # ipa-client-install --server=< IPA server FQDN> --domain=<IPA domain name> --realm=<IPA
   Realm> --hostname=<This hostname> --force-ntpd
   ```

   For example,

   ```
   # ipa-client-install --server=ipaserver.gpfs.net --domain=gpfs.net --realm=IBM.COM --
   hostname=dn01.gpfs.net --force-ntpd
   ```

2. Update the following configurations:

   Copy the `/etc/krb5.conf` file from the IPA server to one of IPA client nodes (i.e. HDFS Transparency nodes). Then update the local `/etc/krb5.conf` as follows:

   ```
   default_ccache_name = KEYRING:persistent:%{uid}
   ```

   to

   ```
   default_ccache_name = /tmp/krb5cc_%{uid}
   ```

   If the `/etc/krb5.conf.d/kcm_default_ccache` file exists, disable the KCM credential cache by commenting out the following lines in that file:

   ```
   [libdefaults] default_ccache_name = KCM:
   ```

3. Distribute the configuration files that were updated in the previous step to the remaining IPA client nodes.
4. Create a directory for the keytabs and set the appropriate permissions on each of the HDFS Transparency node by running the following commands:

   ```
   # mkdir -p /etc/security/keytabs/
   # chown root:root /etc/security/keytabs
   # chmod 755 /etc/security/keytabs
   ```

5. Create KDC principals for the components, corresponding to the hosts where they are running, and create the keytab files as follows:

Creating keytab files for HDFS				
Service	**User:Group**	**Daemons**	**Principal**	**Keytab File Name**
HDFS	root:root	NameNode	nn/<NN_Host_FQDN>@<REALM-NAME>	**`nn.service.keytab`**
		NameNode HTTP	HTTP/<NN_Host_FQDN>@<REALM-NAME>	**`spnego.service.keytab`**
		NameNode HTTP	HTTP/<CES_HDFS_Host_FQDN>@<REALM-NAME>	
		DataNode	dn/<DN_Host_FQDN>@<REALM-NAME>	**`dn.service.keytab`**

Replace the *< NN_Host_FQDN >* with the HDFS Transparency NameNode hostname and the *<DN_Host_FQDN>* with the HDFS Transparency DataNode hostname. Replace the *<CES_HDFS_Host_FQDN>* with the CES hostname configured for your CES HDFS cluster.

You need to create the following:

- One principal/keytab for each HDFS Transparency DataNode.
- Two principals/keytabs for each HDFS Transparency NameNode.
  - **`nn.service.keytab`** - For the *nn/<NameNode host>* principal.
  - **`spnego.service.keytab`** - This keytab contains entries for two principals (HTTP principal for the actual NameNode hostname and HTTP principal for the CES IP hostname).

If you are using Open source Apache Hadoop, you need to create service principals for the YARN and Mapreduce services as shown in the following table:

Creating service principals for YARN and Mapreduce				
Service	**User:Group**	**Daemons**	**Principal**	**Keytab File Name**
YARN	yarn:hadoop	ResourceManager	rm/<Resource_Manager_FQDN>@<REALM-NAME>	**`rm.service.keytab`**
		NodeManager	nm/<Node_Manager_FQDN>@<REALM-NAME>	**`nm.service.keytab`**
Mapreduce	mapred:hadoop	MapReduce Job History Server	jhs/<Job_History_Server_FQDN>@<REALM-NAME>	**`jhs.service.keytab`**

Replace the *<Resource_Manager_FQDN>* with the Resource Manager hostname, the *<Node_Manager_FQDN>* with the Node Manager hostname and the *<Job_History_Server_FQDN>* with the Job History Server hostname.

Run the following commands on the IPA server node:

a. Get a ticket for the IPA Admin.

   For example,

   ```
   kinit admin
   ```

b. For the *HTTP/<CES_HDFS_Host_FQDN>* principal, create a host entry for the CES IP hostname as follows:

   ```
   # ipa host-add /<CES_HDFS_Host_FQDN>
   ```

   For example,

   ```
   # ipa host-add myceshdfs.gpfs.net
   ```

   where, *myceshdfs.gpfs.net* is an example of the CES IP hostname configured for the CES HDFS service.

c. Create the service principals for each service mentioned in Table 1.

   ```
   # ipa service-add <Principal Name>
   ```

   For example:

   ```
   # ipa service-add nn/nn01.gpfs.net
   # ipa service-add nn/nn02.gpfs.net
   # ipa service-add HTTP/nn01.gpfs.net
   # ipa service-add HTTP/nn02.gpfs.net
   # ipa service-add dn/dn01.gpfs.net
   # ipa service-add HTTP/myceshdfs.gpfs.net
   ```

d. Create the following IPA rules:

   Rule to allow the IPA Admin to retrieve the **`HTTP/<CES HDFS hostname>`** principal.

   ```
   # ipa service-allow-retrieve-keytab HTTP/<CES HDFS hostname> --users=<IPA admin user>
   ```

   For example,

   ```
   # ipa service-allow-retrieve-keytab HTTP/myceshdfs.gpfs.net --users=admin
   ```

   Rule to allow the IPA Admin to retrieve the NameNode host principals. Repeat the command for all NameNode hosts.

   ```
   # ipa host-allow-retrieve-keytab --users=admin
   Host name:
   ```

   Enter the NameNode FQDN at the prompt. These rules are needed to create the `spnego.service.keytab` files properly in the next step.

e. For each service on each HDFS Transparency host, create a keytab file by exporting its service principal into a keytab file.

   Note:

   - If the file name for a service is common (for example, **`dn.service.keytab`**) across the hosts, the contents would be different because every keytab would have different principal components.
   - As soon as a keytab file is generated, move (**`scp`**) the keytab to the appropriate host immediately or move it into a different location to avoid the keytab from getting overwritten.

   ```
   # ipa-getkeytab -s <IPA server FQDN> -k /etc/security/keytabs/
   {SERVICE_NAME}.service.keytab -p {Principal}
   ```

   For example:

DataNode:

```
# ipa-getkeytab -s ipaserver.gpfs.net -k /etc/security/keytabs/dn.service.keytab -p dn/
dn01.gpfs.net
# ipa-getkeytab -s ipaserver.gpfs.net -k /etc/security/keytabs/dn.service.keytab -p host/
dn01.gpfs.net -r
```

Move (scp) the generated dn.service.keytab to the corresponding DataNode host under /etc/security/keytabs/ Then remove the local dn.service.keytab file immediately to avoid it from getting overwritten by the next DataNode's keytab.

NameNode1:

```
# ipa-getkeytab -s ipaserver.gpfs.net -k /etc/security/keytabs/nn.service.keytab -p nn/
nn01.gpfs.net
# ipa-getkeytab -s ipaserver.gpfs.net -k /etc/security/keytabs/nn.service.keytab -p host/
nn01.gpfs.net -r


# ipa-getkeytab -s ipaserver.gpfs.net -k /etc/security/keytabs/spnego.service.keytab -p
HTTP/nn01.gpfs.net
# ipa-getkeytab -s ipaserver.gpfs.net -k /etc/security/keytabs/spnego.service.keytab -p
HTTP/myceshdfs.gpfs.net
# ipa-getkeytab -s ipaserver.gpfs.net -k /etc/security/keytabs/spnego.service.keytab -p
host/nn01.gpfs.net -r
```

Move (**`scp`**) the generated **`nn.service.keytab`** and **`spnego.service.keytab`** to the corresponding NameNode host under `/etc/security/keytabs/`. Remove the local **`nn.service.keytab`** and **`spnego.service.keytab`** files immediately to avoid them from getting overwritten by the next NameNode's keytab.

NameNode 2:

```
# ipa-getkeytab -s ipaserver.gpfs.net -k /etc/security/keytabs/nn.service.keytab -p nn/
nn02.gpfs.net
# ipa-getkeytab -s ipaserver.gpfs.net -k /etc/security/keytabs/nn.service.keytab -p host/
nn02.gpfs.net -r

# ipa-getkeytab -s ipaserver.gpfs.net -k /etc/security/keytabs/spnego.service.keytab -p
HTTP/nn02.gpfs.net
# ipa-getkeytab -s ipaserver.gpfs.net -k /etc/security/keytabs/spnego.service.keytab -p
HTTP/myceshdfs.gpfs.net -r
ipa-getkeytab -s ipaserver.gpfs.net -k /etc/security/keytabs/spnego.service.keytab -p
host/nn02.gpfs.net -r
```

Move (**`scp`**) the generated **`nn.service.keytab`** and `spnego.service.keytab` to the corresponding NameNode host under `/etc/security/keytabs/`. Remove the local **`nn.service.keytab`** and **`spnego.service.keytab`** files immediately to avoid it from getting overwritten by the next NameNode's keytab.

Note: Notice the additional **`-r`** option used for the **`ipa-getkeytab`** commands, used to export the keytab files. This is needed if the keytab for a particular principal was already initialized. Otherwise, the existing keytab for that particular principal will get invalidated and can lead to authentication errors.

6. For CES HDFS NameNode HA, an HDFS admin user and its Kerberos user principal and keytab are required to be created and setup for the CES NameNodes. These credentials are used by the CES framework to elect an active NameNode.

   This principal should map to an existing OS user on the NameNode hosts.

   In this example, the OS user is *hdfs*. You will configuring this principal/keytab into `hadoop-env.sh` in step 8.

   a. Create a Hadoop supergroup.

Set the **dfs.permissions.superusergroup** parameter to *supergroup* by running the following command:

```
/usr/lpp/mmfs/hadoop/sbin/mmhdfs config set hdfs-site.xml -k
dfs.permissions.superusergroup=supergroup
```

b. Create the OS user *hdfs* on all the HDFS Transparency nodes that belongs to the Hadoop super group *supergroup* by using the supplied **gpfs_create_hadoop_users_dirs.py** command. This command ensures that the custom user and group is created with consistent UID/GID across all the nodes.

```
# /usr/lpp/mmfs/hadoop/scripts/gpfs_create_hadoop_users_dirs.py --create-custom-hadoop-
user-group hdfs:supergroup
```

Otherwise, if you use FreeIPA to manage all users/groups and SSSD, you should create the *hdfs* user and *supergroup* group in FreeIPA and ensure that you can see the *hdfs* user and *supergroup* group on all the CES HDFS nodes.

c. Create the IPA user principal and keytab for your HDFS Transparency cluster.

Create a unique user principal such as **ces-<clustername>** where, *<clustername>* is the name of your CES HDFS cluster. In case there are multiple CES HDFS clusters sharing a common IPA KDC server, having the cluster name as part of the principal helps to create a user principal unique to each CES HDFS cluster.

```
# ipa user-add ces-<clustername>
# ipa-getkeytab -s ipaserver.gpfs.net -p ces-<clustername>  -k /etc/security/keytabs/ces-
<clustername>.headless.keytab
```

For example,

```
# ipa user-add ces-scaleces
# ipa-getkeytab -s ipaserver.gpfs.net -p ces-scaleces  -k /etc/security/keytabs/ces-
scaleces.headless.keytab
```

d. Copy the `/etc/security/keytabs/ces-<clustername>.headless.keytab` file to all the NameNodes and change the owner permission of the file to *root*.

```
# chown root:root /etc/security/keytabs/ces-<clustername>.headless.keytab
# chmod 400 /etc/security/keytabs/ces-<clustername>.headless.keytab
```

7. Verify the keytabs. The contents should appear as follows:

```
# klist -kte /etc/security/keytabs/nn.service.keytab
Keytab name: FILE:nn.service.keytab
KVNO Timestamp           Principal
---- ------------------- ------------------------------------------------------
   1 07/06/2022 08:02:17 nn/nn01.gpfs.net@IBM.COM (aes256-cts-hmac-sha1-96)
   1 07/06/2022 08:02:17 nn/nn01.gpfs.net@IBM.COM (aes128-cts-hmac-sha1-96)
   7 07/06/2022 08:02:17 host/nn01.gpfs.net@IBM.COM (aes256-cts-hmac-sha1-96)
   7 07/06/2022 08:02:17 host/nn01.gpfs.net@IBM.COM (aes128-cts-hmac-sha1-96

# klist -kte /etc/security/keytabs/spnego.service.keytab
Keytab name: FILE:spnego.service.keytab
KVNO Timestamp           Principal
---- ------------------- ------------------------------------------------------
   1 07/06/2022 08:02:17 HTTP/nn01.gpfs.net@IBM.COM (aes256-cts-hmac-sha1-96)
   1 07/06/2022 08:02:17 HTTP/nn01.gpfs.net@IBM.COM (aes128-cts-hmac-sha1-96)
   1 07/06/2022 08:02:17 HTTP/myceshdfs.gpfs.net@IBM.COM (aes256-cts-hmac-sha1-96)
   1 07/06/2022 08:02:17 HTTP/myceshdfs.gpfs.net@IBM.COM (aes128-cts-hmac-sha1-96)
   7 07/06/2022 08:02:17 host/nn01.gpfs.net@IBM.COM (aes256-cts-hmac-sha1-96)
   7 07/06/2022 08:02:17 host/nn01.gpfs.net@IBM.COM (aes128-cts-hmac-sha1-96

# klist -kte /etc/security/keytabs/dn.service.keytab
Keytab name: FILE:dn.service.keytab
KVNO Timestamp           Principal
---- ------------------- ------------------------------------------------------
   1 07/06/2022 08:02:18 dn/dn01.gpfs.net@IBM.COM (aes256-cts-hmac-sha1-96)
   1 07/06/2022 08:02:18 dn/dn01.gpfs.net@IBM.COM (aes128-cts-hmac-sha1-96)
   7 07/06/2022 08:02:18 host/dn01.gpfs.net@IBM.COM (aes256-cts-hmac-sha1-96)
   7 07/06/2022 08:02:18 host/dn01.gpfs.net@IBM.COM (aes128-cts-hmac-sha1-96
# klist -kte /etc/security/keytabs/ces-scaleces.headless.keytab
Keytab name: FILE:ces-scaleces.headless.keytab
KVNO Timestamp           Principal
---- ------------------- ------------------------------------------------------
   1 07/05/2022 05:08:22 ces-scaleces@IBM.COM (aes256-cts-hmac-sha1-96)
   1 07/05/2022 05:08:22 ces-scaleces@IBM.COM (aes128-cts-hmac-sha1-96)
```

8. Copy the appropriate keytab files from the IPA server to each HDFS Transparency host.
9. Set the appropriate permissions for the keytab files.

   On the HDFS Transparency NameNode hosts, run the following command:

   ```
   # chown root:root /etc/security/keytabs/nn.service.keytab
   # chmod 400 /etc/security/keytabs/nn.service.keytab
   # chown root:root /etc/security/keytabs/spnego.service.keytab
   # chmod 440 /etc/security/keytabs/spnego.service.keytab
   ```

   On the HDFS Transparency DataNode hosts, run the following command:

   ```
   # chown root:root /etc/security/keytabs/dn.service.keytab
   # chmod 400 /etc/security/keytabs/dn.service.keytab
   ```

   On the Yarn resource manager hosts, run the following command:

   ```
   # chown yarn:hadoop /etc/security/keytabs/rm.service.keytab
   # chmod 400 /etc/security/keytabs/rm.service.keytab
   ```

   On the Yarn node manager hosts, run the following command:

   ```
   # chown yarn:hadoop /etc/security/keytabs/nm.service.keytab
   # chmod 400 /etc/security/keytabs/nm.service.keytab
   ```

   On Mapreduce job history server hosts, run the following command:

   ```
   # chown mapred:hadoop /etc/security/keytabs/jhs.service.keytab
   # chmod 400 /etc/security/keytabs/jhs.service.keytab
   ```

10. Update the HDFS Transparency configuration files and upload the changes.

    - Obtain the config files by running the following commands:

      ```
      # mkdir /tmp/hdfsconf
      # mmhdfs config export /tmp/hdfsconf core-site.xml,hdfs-site.xml,hadoop-env.sh
      ```

      Use the following configurations in core-site.xml and hdfs-site.xml, corresponding to HDFS Transparency 3.1.1.x.

      File: core-site.xml

      ```
      <property>
         <name>hadoop.security.authentication</name>
         <value>kerberos</value>
      </property>

      <property>
         <name>hadoop.rpc.protection</name>
         <value>authentication</value>
      </property>
      ```

      If you are using Cloudera Private Cloud Base cluster, create the following rules:

      ```
      <property>
        <name>hadoop.security.auth_to_local</name>
        <value>
        RULE:[2:$1/$2@$0](nn/.*@.*IBM.COM)s/.*/hdfs/
        RULE:[2:$1/$2@$0](dn/.*@.*IBM.COM)s/.*/hdfs/
        RULE:[1:$1@$0](ces-<clustername>@IBM.COM)s/.*/hdfs/
        RULE:[1:$1@$0](.*@IBM.COM)s/@.*//
        DEFAULT
        </value>
      </property>
      ```

Otherwise, if you are using Open source Apache Hadoop, create the following rules:

```
<property>
  <name>hadoop.security.auth_to_local</name>
  <value>
    RULE:[2:$1/$2@$0](nn/.*@.*IBM.COM)s/.*/hdfs/
    RULE:[2:$1/$2@$0](dn/.*@.*IBM.COM)s/.*/hdfs/
    RULE:[2:$1/$2@$0](nm/.*@.*IBM.COM)s/.*/yarn/
    RULE:[2:$1/$2@$0](rm/.*@.*IBM.COM)s/.*/yarn/
    RULE:[2:$1/$2@$0](jhs/.*@.*IBM.COM)s/.*/mapred/
    RULE:[1:$1@$0](ces-<clustername>@IBM.COM)s/.*/hdfs/
    DEFAULT
  </value>
</property>
```

In the above example, replace IBM.COM with your Realm name and *<clustername>* parameter with your actual CES HDFS cluster name.

File: hdfs-site.xml

```
<property>
  <name>dfs.data.transfer.protection</name>
  <value>authentication</value>
</property>

<property>
  <name>dfs.datanode.address</name>
  <value>0.0.0.0:1004</value>
</property>

<property>
  <name>dfs.datanode.data.dir.perm</name>
  <value>700</value>
</property>

<property>
  <name>dfs.datanode.http.address</name>
  <value>0.0.0.0:1006</value>
</property>

<property>
  <name>dfs.datanode.kerberos.principal</name>
  <value>dn/_HOST@IBM.COM</value>
</property>

<property>
  <name>dfs.datanode.keytab.file</name>
  <value>/etc/security/keytabs/dn.service.keytab</value>
</property>

<property>
  <name>dfs.encrypt.data.transfer</name>
  <value>false</value>
</property>

<property>
  <name>dfs.namenode.kerberos.internal.spnego.principal</name>
  <value>HTTP/_HOST@IBM.COM</value>
</property>

<property>
  <name>dfs.namenode.kerberos.principal</name>
  <value>nn/_HOST@IBM.COM</value>
</property>

<property>
  <name>dfs.namenode.keytab.file</name>
  <value>/etc/security/keytabs/nn.service.keytab</value>
</property>

<property>
  <name>dfs.web.authentication.kerberos.keytab</name>
  <value>/etc/security/keytabs/spnego.service.keytab</value>
</property>

<property>
  <name>dfs.web.authentication.kerberos.principal</name>
  <value>*</value>
</property>
```

```
<property>
  <name>dfs.block.access.token.enable</name>
  <value>true</value>
</property>
```

In the above example, replace IBM.COM with your Realm name.

File: hadoop-env.sh:

```
export KINIT_KEYTAB=/etc/security/keytabs/ces-<clustername>.headless.keytab
export KINIT_PRINCIPAL=ces-<clustername>@IBM.COM
```

where, *<clustername>* is the name of your CES HDFS cluster.

11. Stop the HDFS Transparency services for the cluster.

    a. Stop the DataNodes.

       On any HDFS Transparency node, run the following command:

       ```
       # mmhdfs hdfs-dn stop
       ```

    b. Stop the NameNodes.

       On any CES HDFS NameNode, run the following command:

       ```
       # mmces service stop HDFS -N <NN1>,<NN2>
       ```

12. Import the files by running the following command:

    ```
    # mmhdfs config import /tmp/hdfsconf core-site.xml,hdfs-site.xml,hadoop-env.sh
    ```

13. Upload the changes by running the following command:

    ```
    # mmhdfs config upload
    ```

14. Start the HDFS Transparency services for the cluster.

    a. Start the DataNodes.

       On any HDFS Transparency node, run the following command:

       ```
       # mmhdfs hdfs-dn start
       ```

    b. Start the NameNodes.

       On any CES HDFS NameNode, run the following command:

       ```
       # mmces service start HDFS -N <NN1>,<NN2>
       ```

    c. Verify that the services have started.

       On any CES HDFS NameNode, run the following command:

       ```
       # mmhdfs hdfs status
       ```

Verifying Kerberos

For information about verifying Kerberos, see "Verifying Kerberos" on page 127.

Verifying Kerberos

This topic lists the steps to verify Kerberos.

1. Verify that the HDFS Transparency has started.

   On CES HDFS cluster admin node, run the following command:

```
# /usr/lpp/mmfs/hadoop/sbin/mmhdfs hdfs status
```

2. Verify if Kerberos is enabled by running the following command on the CES HDFS cluster admin node:

   ```
   mmhdfs config get core-site.xml -k hadoop.security.authentication
   ```

   If Kerberos is enabled, the output will display `kerberos`, else it will display `simple`.

   `hadoop.security.authentication=kerberos`

3. Verify NameNode service principal

   The NameNode service principals configured for HDFS Transparency are internally used by Cloudera Manager to create home directories for Cloudera services. Follow these steps to ensure that these principals are configured properly and are functional:

   a. Log in to one of the CES HDFS NameNode hosts.

   b. Get a token for the NameNode service principal. For example,

   ```
   # kinit -kt /etc/security/keytabs/nn.service.keytab  nn/<NameNode hostname>@<Realm Name>
   ```

   c. Verify that you can access IBM Storage Scale file system by running the following command:

   ```
   # /usr/lpp/mmfs/hadoop/bin/hdfs dfs -ls /
   ```

   d. Ensure that the token can be renewed by running the following command:

   ```
   # kinit -R
   ```

   e. After you have verified, drop the token by running the following command:

   ```
   # kdestroy
   ```

   If possible, repeat this for all the NameNode principals.

4. Verify the hdfs admin user principal.

   The hdfs admin user principal (configured in hadoop-env.sh) is used by HDFS Transparency internally to run the HDFS administrative (**`hdfs haadmin`**) commands. Follow these steps to ensure that these principals are configured properly and are functional:

   a. Log in to the CES HDFS cluster admin node.

   b. Get a token for the hdfs admin user. For example,

   ```
   # kinit -kt /etc/security/keytabs/ces-<clustername>.headless.keytab ces-
   <clustername>@<Realm name> -c /var/mmfs/tmp/krb5cc_ces
   ```

   where, *<clustername>* is the cluster name of your CES HDFS and *<Realm>* is the Realm name of your Kerberos. For example, IBM.COM.

   Note: The non-default ticket cache location for this principal is `/var/mmfs/tmp/krb5cc_ces`. CES uses this location for ticket cache.

   c. Verify that you can access the IBM Storage Scale file system by running the following command:

   ```
   env KRB5CCNAME=/var/mmfs/tmp/krb5cc_ces /usr/lpp/mmfs/hadoop/bin/hdfs dfs -ls /
   ```

   d. Verify that there is one active NameNode.

   ```
   # env KRB5CCNAME=/var/mmfs/tmp/krb5cc_ces /usr/lpp/mmfs/hadoop/bin/hdfs haadmin
   -getAllServiceState
   Namenode1.gpfs.net:8020                           active
   Namenode2.gpfs.net:8020                           standby
   ```

   e. Run **`hdfs`** and **`webhdfs -ls/-put/-get/`** commands to successfully perform I/O to HDFS Transparency using the **`hdfs rpc`** and **`webhdfs`** protocols.

   f. Ensure that the hdfs token can be renewed by running the following command:

```
env KRB5CCNAME=/var/mmfs/tmp/krb5cc_ces kinit -R
```

TLS

Transport Layer Security (TLS)/Secure Sockets Layer (SSL) provides privacy and data integrity between applications communicating over a network by encrypting the packets transmitted between the endpoints. Configuring TLS/SSL for any system typically involves creating a private key and public key for use by server and client processes to negotiate an encrypted connection at runtime.

This section describes how to configure TLS for CES HDFS Transparency.

Prerequisites

This topic lists the prerequisites before enabling TLS for HDFS Transparency.

- Ensure that the webHDFS client is working properly before you enable TLS for HDFS Transparency by following the steps in the Verifying installation topic.
- `$JAVA_HOME/`**`keytool`** utility should be available on all the HDFS Transparency nodes.
- Ensure that both HDFS Transparency server and HDFS client are configured with JDK starting from `8u141` JDK.
- Before enabling TLS, confirm whether Kerberos is enabled and verified on the HDFS Transparency cluster by following the steps in "Kerberos" on page 81 and "Verifying Kerberos" on page 127 topics.
- Before enabling TLS, HDFS Transparency services must be stopped.

Note: For TLS to function properly, CES HDFS hostname must be DNS resolved. Remove any entries in the `/etc/hosts` file for the CES HDFS IP from all the cluster nodes.

Enabling TLS for HDFS Transparency using the automation script

This section lists the steps to enable TLS for CES HDFS Transparency by using the `gpfs_tls_configuration.py` script.

The `gpfs_tls_configuration.py` script is available from HDFS Transparency 3.1.1-5 under IBM Storage Scale 5.1.1.1. This script performs all the steps that are documented in "Manually enabling TLS for HDFS Transparency" on page 131. The actions performed are logged in the `logfile /var/log/transparency/tls_configuration.log` file.

Before following these steps, see the "Prerequisites" on page 129 topic. Passwordless ssh is required from the NameNode to all HDFS Transparency nodes for the script to run.

Note: In HDFS Transparency 3.1.1-13 and earlier, the TLS certificates created by the script `/usr/lpp/mmfs/hadoop/scripts/gpfs_tls_configuration.py` would have validity of 90 days only. From HDFS Transparency v3.1.1-14 onwards, default validity has been changed to five years and can be overridden by passing the **`--validity`** *NUMBER-OF-DAYS* flag.

The `gpfs_tls_configuration.py` automation script performs the following steps:

1. Creates key-pairs for NameNode(s) and DataNodes.
2. Distributes key-pairs to the HDFS Transparency nodes.
3. Creates a common truststore for IBM Storage Scale and distributes among the HDFS Transparency nodes.
4. Updates CES HDFS configurations and uploads it to CCR.

To enable TLS, run the following command as root from one of the NameNodes:

```
/usr/lpp/mmfs/hadoop/scripts/gpfs_tls_configuration.py enable-tls [--password-file
KEYSTORE_PASSWORD_FILE ] CES-HDFS-HOSTNAME DISTINGUISHED-NAME
```

where,

CES-HDFS-HOSTNAME

Is the fully qualified domain name of CES hostname. For example, *cesip09x10.gpfs.net*.

DISTINGUISHED-NAME

The script uses X.500 Distinguished Names (DN) to create the key-pairs. The following sub parts can be set for Distinguished Names:

- organizationUnit: (OU) - Department or division name. For example, "Systems".
- organizationName: (O) - Large organization name. For example, "IBM".
- localityName: (L) - Locality (city) name. For example, "SanJose".
- stateName: (S) - State or province name. For example, "California".
- country: (C) - Two letter country code. For example, "US".

  For example, --dname 'OU=Systems,O=IBM,L=SanJose,S=California,C=US'

  Note: Common Name (CN) cannot be set with this argument. CN is chosen based on the hostname of the server for which the key pair will be created.

--password-file KEYSTORE_PASSWORD_FILE

This is an optional argument. The password file that is denoted by *<KEYSTORE_PASSWORD_FILE>* must be created in the following JSON format to contain the passwords for the keystore and the truststore:

```
{
  "keystore-password": "CHANGE-ME",
  "keystore-keypassword": "CHANGE-ME",
  "truststore-password": "CHANGE-ME"
}
```

If the **`--password-file`** file is not specified or if one or more passwords are not specified in the JSON file, random passwords are generated against the fields that do not have the key:value pair. The random generated passwords can be found in the `ssl-server.xml` file.

Note: After the script has successfully run, the password file will be automatically deleted for security reason.

--validity NUMBER-OF-DAYS

This is an optional argument. Takes a number as input. The certificates will be created with the number of days passed. If this parameter is not used, certificates will be created with 5 years of validity from the current date.

For example:

```
/usr/lpp/mmfs/hadoop/scripts/gpfs_tls_configuration.py enable-tls cesip09x10.gpfs.net
OU=Systems,O=IBM,L=SanJose,S=California,C=US --password-file /tmp/passwordfile.txt
[ INFO  ] Performing validations
[ INFO  ] Keystore password file is provided, using passwords from '/tmp/passwordfile.txt'
[ INFO  ] Creating spectrum_scale_ces_hdfs_keystore.jks keystore with CES IP host-name:
'cesip09x10.gpfs.net'
[ INFO  ] Distributing spectrum_scale_ces_hdfs_keystore.jks containing CES host name key-pair
on NameNodes
[ INFO  ] Creating key-pairs for NameNode(s) and DataNodes hosts and adding it to IBM Spectrum
Scale common trust store file: spectrum_scale_ces_hdfs_truststore.jks
[ INFO  ] Distributing IBM Spectrum Scale trust store file:
spectrum_scale_ces_hdfs_truststore.jks to all the NameNode(s) and DataNodes
[ INFO  ] Performing configuration changes
[ INFO  ] Uploading configurations to CCR
[ INFO  ] Enablement of TLS on CES HDFS Transparency is completed
```

Disabling TLS for HDFS Transparency using the automation script

This section lists the steps to disable TLS for HDFS Transparency cluster using the `gpfs_tls_configuration.py` script.

Note: TLS cannot be disabled on Cloudera Manager. It can only be disabled on HDFS Transparency cluster.

1. Stop the HDFS Transparency services by running the following commands:

   a. Stop the DataNodes by running the following command as root on any HDFS Transparency node:

   ```
   mmhdfs hdfs-dn stop
   ```

   b. Stop the NameNodes by running the following command as root on a CES HDFS NameNode:

   ```
   mmces service stop HDFS -N <NN1>,<NN2>
   ```

2. Run the following command on a CES HDFS NameNode to disable TLS for the HDFS Transparency cluster:

   ```
   /usr/lpp/mmfs/hadoop/scripts/gpfs_tls_configuration.py disable-tls
   ```

   Running this script performs the following:

   - Clears the configuration changes that were performed while enabling TLS.
   - Deletes the TLS certificate files created for HDFS Transparency on all the HDFS Transparency nodes.
   - Uploads the configuration to IBM Storage Scale CCR.

3. Delete the IBM Storage Scale Trust store file manually from all the Cloudera managed nodes. On all the Cloudera managed nodes run the following command:

   ```
   rm -f /var/lib/cloudera-scm-agent/agent-cert/spectrum_scale_ces_hdfs_truststore.jks
   ```

4. Start the HDFS Transparency services by running the following commands:

   a. Start the NameNodes by running the following command as root on a CES HDFS NameNode:

   ```
   mmces service start HDFS -N <NN1>,<NN2>
   ```

   b. Start the DataNodes by running the following command as root on any HDFS Transparency node:

   ```
   mmhdfs hdfs-dn start
   ```

Manually enabling TLS for HDFS Transparency

This section lists the steps to manually enable TLS for HDFS Transparency.

Overview:

For enabling TLS on the CES HDFS Transparency cluster, the following must be created on the CES HDFS nodes:

- Truststores
- Certificates
- Keystores

You must then verify that the IBM Storage Scale HDFS clients can securely access the IBM Storage Scale file system.

Before following these steps, see the Prerequisites topic.

Procedure:

1. On all the HDFS Transparency nodes, as root, create a directory `/etc/security/serverKeys/` where the TLS keys and certificates can be stored. On all the HDFS Transparency nodes, run the following command:

   ```
   # mkdir -p /etc/security/serverKeys/
   ```

2. Log in to one of the CES HDFS NameNode as root (for example, NameNode1) and run all the following commands from that one node:

a. Create a keystore specific to CES HDFS IP, using the **keytool -genkey** command.

```
# keytool -genkey -alias <CES_HOSTNAME_FQDN> -keyalg RSA -keysize 2048 -validity 1800
-keystore /etc/security/serverKeys/spectrum_scale_ces_hdfs_keystore.jks
```

where, *<CES_HOSTNAME_FQDN>* is the FQDN Hostname corresponding to the CES IP configured for your CES HDFS cluster.

After you run the command, enter a password for the keystore, a password for the key, your first and last name, and your organization and location details.

Note:

- Your first name and last name must be same as your CES IP hostname.
- The keystore password and the key password will be needed for later configuration steps. Therefore, keep the passwords in a safe place.

For example,

```
# keytool -genkey -alias cesip09x15.gpfs.net -keyalg RSA -keysize 2048 -validity 1800
-keystore /etc/security/serverKeys/spectrum_scale_ces_hdfs_keystore.jks

Enter keystore password:
Re-enter new password:
What is your first and last name?
  [Unknown]:  cesip09x15.gpfs.net
What is the name of your organizational unit?
  [Unknown]:  IBM
What is the name of your organization?
  [Unknown]:  IBM
What is the name of your City or Locality?
  [Unknown]:  Poughkeepsie
What is the name of your State or Province?
  [Unknown]:  New York
What is the two-letter country code for this unit?
  [Unknown]:  US
Is CN=cesip09x15.gpfs.net, OU=IBM, O=IBM, L=Poughkeepsie, ST=New York, C=US correct?
  [no]:  yes

Enter key password for <cesip09x15.gpfs.net>
        (RETURN if same as keystore password):

Warning:
The JKS keystore uses a proprietary format. It is recommended to migrate to PKCS12 which
is an industry standard format using "keytool -importkeystore -srckeystore /etc/security/
serverKeys/spectrum_scale_ces_hdfs_keystore.jks -destkeystore /etc/security/serverKeys/
spectrum_scale_ces_hdfs_keystore.jks -deststoretype pkcs12".
#
```

b. For the keystore that is created in the above step, export the certificate public key to a certificate file.

```
# keytool -export -alias <CES_HOSTNAME_FQDN> -keystore  /etc/security/
serverKeys/spectrum_scale_ces_hdfs_keystore.jks -rfc -file  /etc/security/serverKeys/
<CES_HOSTNAME_FQDN>.pem
```

where, *<CES_HOSTNAME_FQDN>* is the FQDN hostname corresponding to the CES IP configured for your CES HDFS cluster.

For example,

```
# keytool -export -alias cesip09x15.gpfs.net -keystore  /etc/security/
serverKeys/spectrum_scale_ces_hdfs_keystore.jks -rfc -file  /etc/security/serverKeys/
cesip09x15.gpfs.net.pem
Enter keystore password:
Certificate stored in file </etc/security/serverKeys/cesip09x15.gpfs.net.pem
```

c. Distribute the generated Keystore to all the HDFS Transparency NameNodes. This keystores will be updated in the next step.

Run the following command for each *<HDFS Transparency NameNode>*:

```
# scp /etc/security/serverKeys/spectrum_scale_ces_hdfs_keystore.jks  root@<HDFS
Transparency NameNode>:/etc/security/serverKeys/
```

Note: Do not copy the Keystore to DataNodes. DataNodes have their own Keystores that will be created in step 4.

3. On each CES HDFS NameNode, run the following commands as a root user:

   a. Update the keystore specific to this NameNode hostname.

   ```
   # keytool -genkey -alias <NN_FQDN_HOSTNAME> -keyalg RSA -keysize 2048 -validity 1800
   -keystore /etc/security/serverKeys/spectrum_scale_ces_hdfs_keystore.jks
   ```

   where, *<NN_FQDN_HOSTNAME>* is the FQDN hostname of this NameNode. When you are prompted, use the keystore password from step 2.a. This command will append the generated keystore to the already existing keystore created specific to the CES HDFS IP hostname.

   b. For the keystore that is created in the above step, export the certificate public key to a certificate file.

   ```
   # keytool -export -alias <NN_FQDN_HOSTNAME> -keystore  /etc/security/
   serverKeys/spectrum_scale_ces_hdfs_keystore.jks -rfc -file  /etc/security/serverKeys/
   <NN_FQDN_HOSTNAME>.pem
   ```

   where, *<NN_FQDN_HOSTNAME>* is the FQDN hostname of this NameNode. When you are prompted, use the keystore password from step 2.a.

4. On each HDFS Transparency DataNode, run the following commands as a root user:

   a. Create a keystore specific to this DataNode.

   ```
   # keytool -genkey -alias <DN_HOSTNAME_FQDN> -keyalg RSA -keysize 2048 -validity 1800
   -keystore /etc/security/serverKeys/spectrum_scale_ces_hdfs_keystore.jks
   ```

   where, *<DN_FQDN_HOSTNAME>* is the FQDN hostname of this DataNode. When prompted, assign a password for the keystore.

   b. For the keystore that is created in the above step, export the certificate public key to a certificate file specific to this DataNode.

   ```
   # keytool -export -alias <DN_HOSTNAME_FQDN> -keystore  /etc/security/
   serverKeys/spectrum_scale_ces_hdfs_keystore.jks -rfc -file  /etc/security/serverKeys/
   <DN_HOSTNAME_FQDN>.pem
   ```

   where, *<DN_FQDN_HOSTNAME>* is the FQDN Hostname of this DataNode.

   When you are prompted, use the keystore password from step 4.a.

5. Create a Master Truststore for HDFS Transparency. Log into the same CES HDFS NameNode as in step 2 and run the following commands from that node:

   a. Create a `/etc/security/serverKeys/trust_store/` directory where the created Truststore .jks files corresponding to all NameNodes, DataNodes, and the CES IP will be merged together to create a master Truststore.

   ```
   # mkdir -p /etc/security/serverKeys/trust_store/
   ```

   b. From the NameNode and DataNodes hosts, copy all the .pem files from the `/etc/security/serverKeys/` directory to the `/etc/security/serverKeys/trust_store/` directory on this NameNode.

   For each *<HDFS Transparency node>*, run the following command:

   ```
   # scp <HDFS Transparency node>:/etc/security/serverKeys/*.pem  /etc/security/serverKeys/
   trust_store/
   ```

Ensure that all the .pem files corresponding to every NameNode, DataNode hostname as well as the CES IP hostname are copied over to `/etc/security/serverKeys/trust_store/`.

c. For each <.pem file> in the `/etc/security/serverKeys/trust_store/` directory, run the following command:

```
# keytool -import -noprompt -alias <FQDN Hostname Corresponding the .pem file>
-file /etc/security/serverKeys/trust_store/<.pem file> -keystore /etc/security/serverKeys/
spectrum_scale_ces_hdfs_truststore.jks
```

This will import the certificates into a Master Truststore `/etc/security/serverKeys/spectrum_scale_ces_hdfs_truststore.jks` for HDFS Transparency:

For example,

If `namenode1.gpfs.net.pem` is the certificate file corresponding to the host `namenode1.gpfs.net`, run the following command:

```
# keytool -import -noprompt -alias namenode1.gpfs.net -file /etc/security/
serverKeys/trust_store/namenode1.gpfs.net.pem  -keystore /etc/security/serverKeys/
spectrum_scale_ces_hdfs_truststore.jks
```

Note: When you import the first .pem file, you need to set a password for the Master Truststore. Use this password for importing the remaining .pem files.

d. Repeat the **`keytool -import`** command for all the .pem files.

6. Distribute the master Truststore and master certificate to all the HDFS Transparency nodes (NameNodes and DataNodes).

   Run the following command specific to each *<HDFS Transparency node>*:

```
# scp /etc/security/serverKeys/spectrum_scale_ces_hdfs_truststore.jks  root@<HDFS
Transparency node>:/etc/security/serverKeys/
```

7. Create and update the CES HDFS Transparency configuration files.

   a. Create a directory to host the configuration files. This directory will host existing and newly created configuration files.

```
# mkdir /tmp/hdfsconf
```

   b. Get the existing configuration files from CCR into that directory.

```
# /usr/lpp/mmfs/hadoop/sbin/mmhdfs config export /tmp/hdfsconf core-site.xml,hdfs-
site.xml,ssl-server.xml,ssl-client.xml
```

   c. Update the existing config files with the following changes based on your environment:

   File: core-site.xml

```
<property>
    <name>hadoop.ssl.require.client.cert</name>
    <value>false</value>
  </property>

<property>
    <name>hadoop.ssl.hostname.verifier</name>
    <value>DEFAULT</value>
  </property>

  <property>
    <name>hadoop.ssl.keystores.factory.class</name>
    <value>org.apache.hadoop.security.ssl.FileBasedKeyStoresFactory</value>
  </property>

 <property>
    <name>hadoop.ssl.server.conf</name>
    <value>ssl-server.xml</value>
  </property>

<property>
    <name>hadoop.ssl.client.conf</name>
    <value>ssl-client.xml</value>
  </property>
```

File: hdfs-site.xml

```
<property>
    <name>dfs.http.policy</name>
    <value>HTTPS_ONLY</value>
  </property>
  <property>
    <name>dfs.client.https.need-auth</name>
    <value>false</value>
  </property>
  <property>
    <name>dfs.namenode.https-bind-host</name>
    <value>0.0.0.0</value>
  </property>
<property>
   <name>dfs.namenode.https-address.<cluster name>.nn1</name>
   <value><NameNode 1 hostname>:50470</value>
</property>
<property>
   <name>dfs.namenode.https-address.<cluster name>.nn2</name>
   <value><NameNode 2 hostname>:50470</value>
</property>
```

where,

<NameNode 1 hostname> and *<NameNode 2 hostname>* are the actual CES HDFS NameNode FQDN hostnames.

<cluster name> is the name of your CES HDFS cluster that is also your HDFS Namespace.

If you want both the secure and unsecure http connections, set **`dfs.http.policy`** to *HTTP_AND _HTTPS*.

File: ssl-server.xml

```
<property>
    <name>ssl.server.truststore.location</name>
    <value>/etc/security/serverKeys/spectrum_scale_ces_hdfs_truststore.jks</value>
  </property>
  <property>
    <name>ssl.server.truststore.password</name>
    <value><truststore_password></value>
  </property>
  <property>
    <name>ssl.server.truststore.type</name>
    <value>jks</value>
  </property>
  <property>
    <name>ssl.server.truststore.reload.interval</name>
    <value>10000</value>
  </property>
  <property>
    <name>ssl.server.keystore.location</name>
    <value>/etc/security/serverKeys/spectrum_scale_ces_hdfs_keystore.jks</value>
  </property>
  <property>
    <name>ssl.server.keystore.password</name>
    <value><keystore_password></value>
  </property>
  <property>
    <name>ssl.server.keystore.keypassword</name>
    <value><key_password></value>
  </property>
  <property>
    <name>ssl.server.keystore.type</name>
    <value>jks</value>
  </property>
```

where,

<keystore_password> and *<key_password>* are the corresponding actual passwords from "2.a" on page 132.

<truststore_password> is the corresponding actual password from "5.c" on page 134.

File: ssl-client.xml

```
<property>
    <name>ssl.client.truststore.location</name>
    <value>/etc/security/serverKeys/spectrum_scale_ces_hdfs_truststore.jks</value>
  </property>

  </property>
  <property>
    <name>ssl.client.truststore.password</name>
    <value><truststore_password></value>
  </property>
  <property>
    <name>ssl.client.truststore.type</name>
    <value>jks</value>
  </property>
```

where, *<truststore_password>* is the corresponding actual password from “5.c” on page 134.

8. Update the CES HDFS configuration to IBM Storage Scale CCR repository and restart the HDFS Transparency services.

   a. Stop HDFS Transparency services for the cluster.

      i) On any CES HDFS NameNode, run the following commands:

      ```
      # mmhdfs hdfs-dn stop
      # mmces service stop HDFS -N <NN1>,<NN2>
      ```

   b. Import the existing and new configuration files to `/var/mmfs/hadoop/etc/hadoop` by running the following command:

      ```
      # mmhdfs config import /tmp/hdfsconf core-site.xml,hdfs-site.xml,ssl-server.xml,ssl-
      client.xml
      ```

   c. Upload the changes to CCR repository.

      ```
      # mmhdfs config upload
      ```

   d. Start the HDFS Transparency services for the cluster.

      i) On any CES HDFS NameNode, run the following command:

      ```
      # mmhdfs hdfs-dn start
      # mmces service start HDFS -N <NN1>,<NN2>
      ```

   e. Verify that the CES HDFS services have started.

      On any CES HDFS NameNode, run the following command:

      ```
      # mmhdfs hdfs status
      ```

Verifying TLS for HDFS Transparency

This section describes the steps to verify TLS security on the HDFS Transparency nodes.

Run **`kinit`** with a valid keytab to obtain a Kerberos ticket. For more information, see “Verifying installation” on page 309.

To list the files under IBM Storage Scale Hadoop root directory, run the following commands:

1. Verify the secure HDFS Java (swebhdfs) client by running the following command:

   ```
   # echo “hello world” > /tmp/hello
   # /usr/lpp/mmfs/hadoop/bin/hdfs dfs -ls swebhdfs://<HDFS HA Namespace>/
   # /usr/lpp/mmfs/hadoop/bin/hdfs dfs -put /tmp/hello swebhdfs://<HDFS HA Namespace>/tmp
   # /usr/lpp/mmfs/hadoop/bin/hdfs dfs -cat swebhdfs://<HDFS HA Namespace >/tmp/hello
   ```

   where, **`<HDFS HA Namespace>`** is defined by the **`fs.defaultFS`** parameter in your `/var/mmfs/hadoop/etc/hadoop/core-site.xml`.

2. Verify the https client by running the following command:

   ```
   #curl  --cacert /var/lib/cloudera-scm-agent/agent-cert/cm-auto-global_cacerts.pem --
   negotiate -u: https://<CES_HOSTNAME>:50470/webhdfs/v1/?op=LISTSTATUS
   ```

   where, *<CES_HOSTNAME>* is the FQDN hostname corresponding to the CES IP configured for your CES HDFS cluster.

   The following command may also be used to verify. However, it bypasses the CA certificate checking. Therefore, it is not recommended other than for troubleshooting purposes.

   ```
   # curl -ku: --negotiate https://<CES_HOSTNAME>:50470/webhdfs/v1/?op=LISTSTATUS
   ```

   Note:

   - For Non-HA CES HDFS clusters, use the <CES_HOSTNAME>:<port> format instead of Namespace for the **hdfs** commands.
   - For **curl** commands, always use the <CES_HOSTNAME>:<port> format. For Kerberos enabled clusters, substituting <CES_HOSTNAME> with <CES-IP> will fail with HTTP 401 (Auth) error, as the Kerberos principal is created only for the CES hostname.

Rotating TLS certificates

You may rotate TLS certificates that were created for IBM Storage Scale HDFS Transparency if your existing certificates are going to expire or have already expired.

To rotate the TLS certificates created for IBM Storage Scale HDFS Transparency, ensure that you complete the following steps:

- Stop all HDFS Transparency services.
- Follow the steps under the "Disabling TLS for HDFS Transparency using the automation script" on page 130 topic to disable TLS for HDFS Transparency by using the automation script. This action will remove all the existing certificates, keystores, and truststores created for HDFS Transparency.
- Follow the steps under the "Enabling TLS for HDFS Transparency using the automation script" on page 129 topic to enable TLS for HDFS Transparency by using the automation script again. This action regenerates keystores, truststores, and certificates for HDFS Transparency.
- If Cloudera CDP is integrated, exchange the certificates between Cloudera and IBM Storage Scale truststores by following the Update Cloudera and IBM Storage Scale truststores step. As a result, following actions are completed:
  - Cloudera Manager public certificates are imported to IBM Storage Scale truststore.
  - The older IBM Storage Scale certificates are removed from Cloudera Manager truststore.
  - The new IBM Storage Scale certificates are imported to Cloudera Manager truststore.
- If Ranger is enabled, the Ranger TLS plugin file `rangerpluginssl.jceks` needs to be re-created. For more information, see the Additional configurations when TLS is enabled step.
- Start HDFS Transparency services:
  - If Ranger is enabled together with Cloudera, start the NameNodes by using the workaround in the Ranger issues with TLS enabled step.

Note: To see the validity of your current certificates, use the **keytool -list** command with the `-v` option. For example:

```
# keytool -list -keystore /etc/security/serverKeys/spectrum_scale_ces_hdfs_truststore.jks -v
```

The `Valid from:` field in the output shows the validity information.

Apache Ranger

Learn how to enable Apache Ranger plug-in for HDFS Transparency.

Make sure to meet the following prerequisites before you enable the Apache Ranger plug-in for HDFS Transparency:

- Set up Apache Ranger according to its installation instructions.
- Install a relational database management system (RDBMS) supported by Apache Ranger, such as MySQL or MariaDB.
- Verify that Ranger Admin, Ranger Usersync, and Ranger TagSync are successfully installed and without errors.
- Even though not mandatory for installing and using Apache Ranger, it is strongly recommended to enable Kerberos in your Hadoop. This data security tool ensures that all requests are authenticated, which is very important for authorization and auditing. Without Kerberos, the users would be able to impersonate other users and workaround any authorization policies.
- Make sure that Apache Solr is working well for Apache Ranger. When properly configured, Apache Solr is used by Apache Ranger to store audit logs; Apache Solr also provides a search capability of the audit logs through the **Ranger Admin GUI**.

1. Stop the HDFS Transparency by using the following command:

   ```
   # mmhdfs hdfs stop
   ```

2. To enable Apache Ranger for HDFS Transparency, log in to one of the HDFS Transparency nodes and change the configuration as described in this step.

   For `hadoop-env.sh`, set the following configuration:

   Note: Based on your environment, substitute the right path to the Apache Ranger `ranger-hdfs-plugin` library.

   ```
   for f in <ranger_hdfs_plugin_directory>/lib/*.jar; do
     export HADOOP_CLASSPATH=$HADOOP_CLASSPATH:$f
   done

   for f in /usr/share/java/mysql-connector-java.jar; do
     export HADOOP_CLASSPATH=$HADOOP_CLASSPATH:$f
   done
   ```

   For `core-site.xml`, set the following configuration:

   ```
   <property>
     <name>hadoop.security.auth_to_local</name>
     <value>
   RULE:[2:$1@$0](rangeradmin@<REALM_NAME>)s/(.*)@<REALM_NAME>/ranger/
   RULE:[2:$1@$0](rangertagsync@<REALM_NAME>)s/(.*)@<REALM_NAME>/rangertagsync/
   RULE:[2:$1@$0](rangerusersync@<REALM_NAME>)s/(.*)@<REALM_NAME>/rangerusersync/
   ……
   DEFAULT
     </value>
     <final>false</final>
   </property>
   ```

   For `hdfs-site.xml`, set the following configuration:

   ```
   <property>
     <name>dfs.namenode.inode.attributes.provider.class</name>
     <value>org.apache.ranger.authorization.hadoop.RangerHdfsAuthorizer</value>
      <final>false</final>
   </property>
   ```

3. Copy the following configuration files from the Apache Ranger installation directory to an HDFS Transparency node configuration directory (`/var/mmfs/hadoop/etc/hadoop`). These configuration files are generated by the `enable-hdfs-plugin.sh` script when the Apache Ranger plug-in is enabled.

- `ranger-hdfs-audit.xml`
- `ranger-hdfs-security.xml`
- `ranger-policymgr-ssl.xml`

4. To synchronize the configuration in all the HDFS Transparency nodes, issue the following command:

```
# mmhdfs config upload
```

5. Create **`ranger`**, **`rangertagsync`**, and **`rangerusersync`** using the `gpfs_create_hadoop_users_dirs.py` script.

   Log in to a CES HDFS NameNode and run the following commands:

```
# /usr/lpp/mmfs/hadoop/scripts/gpfs_create_hadoop_users_dirs.py --create-custom-hadoop-user-group ranger

# /usr/lpp/mmfs/hadoop/scripts/gpfs_create_hadoop_users_dirs.py --create-custom-hadoop-user-group rangertagsync

# /usr/lpp/mmfs/hadoop/scripts/gpfs_create_hadoop_users_dirs.py --create-custom-hadoop-user-group rangerusersync
```

6. To ensure that changes are effective, start the HDFS Transparency by using the following command:

```
# mmhdfs hdfs start
```

Verifying Apache Ranger policy for HDFS Transparency

Learn how to verify the Apache Ranger policy for HDFS Transparency.

After the Apache Ranger HDFS plug-in for HDFS Transparency is configured, you must verify that the Ranger-based resource control is working properly.

The following example is Kerberos-enabled and it shows how to verify an HDFS resource policy by using the testuser (operating system's user ID).

1. Log in to the **Ranger Admin GUI**, at http://*<Ranger Admin host>*:6080.
2. Create a new policy and set the testuser's **READ** permission as `deny` for HDFS Resource Path `/tmp`.
3. Log in to any CDP node and obtain a Kerberos token for the testuser principal.

```
# kinit -kt /etc/security/keytabs/testuser.headless.keytab testu@<REALM_NAME>
```

4. Make sure that the testuser cannot read from the `/tmp` directory. The output must be similar to the following example:

```
# testuser can write data into /tmp
# hdfs dfs -put /etc/hosts /tmp/rangertest
# testuser can not read data from /tmp
# hdfs dfs -cat /tmp/rangertest
cat: Permission denied: user=testuser, access=READ, inode="/tmp/rangertest"
```

Hadoop Storage Tiering with IBM Storage Scale HDFS Transparency

Overview

IBM Storage Scale HDFS Transparency, also known as HDFS Protocol, offers a set of interfaces that allows applications to use HDFS clients to access IBM Storage Scale through HDFS RPC requests.

For more information about HDFS Transparency, see Chapter 2, “IBM Storage Scale support for Hadoop,” on page 3.

Currently, if the jobs running on the native HDFS cluster plan to access data from IBM Storage Scale, the option is to use **`distcp`** or Hadoop Storage Tiering mode with native HDFS federation.

Using Hadoop **distcp** requires the data to be copied between the native HDFS and the IBM Storage Scale HDFS Transparency cluster and this must be done before accessing. There are two copies of the same data consuming the storage space. For more information, see "Hadoop distcp support" on page 189.

If you are using Hadoop Storage Tiering mode with native HDFS federation to federate native HDFS and IBM Storage Scale HDFS Transparency, the jobs running on the Hadoop cluster with native HDFS can read and write the data from IBM Storage Scale in real time. There would be only one copy of the data. However, the ViewFs schema used in federation with native HDFS is not certified by the Hive community and HDFS federation is not supported by Hortonworks HDP 2.6.

Note: Hadoop Storage Tiering mode with native HDFS federation is not supported in HDFS Transparency.

Hadoop Storage Tiering mode without native HDFS federation

This topic shows how to architect and configure a Hadoop Storage Tiering solution with a suite of test cases executed based on this configuration.

The Hadoop Storage Tiering with IBM Storage Scale architecture is shown in Figure 14 on page 140 and Figure 15 on page 140:

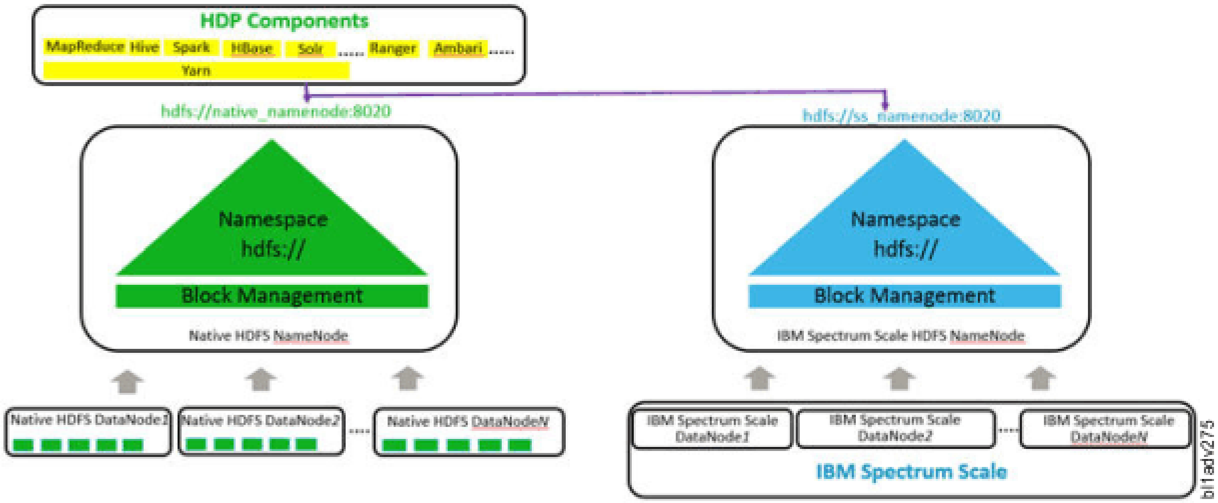


Figure 14. Hadoop Storage Tiering with IBM Storage Scale without HDP cluster

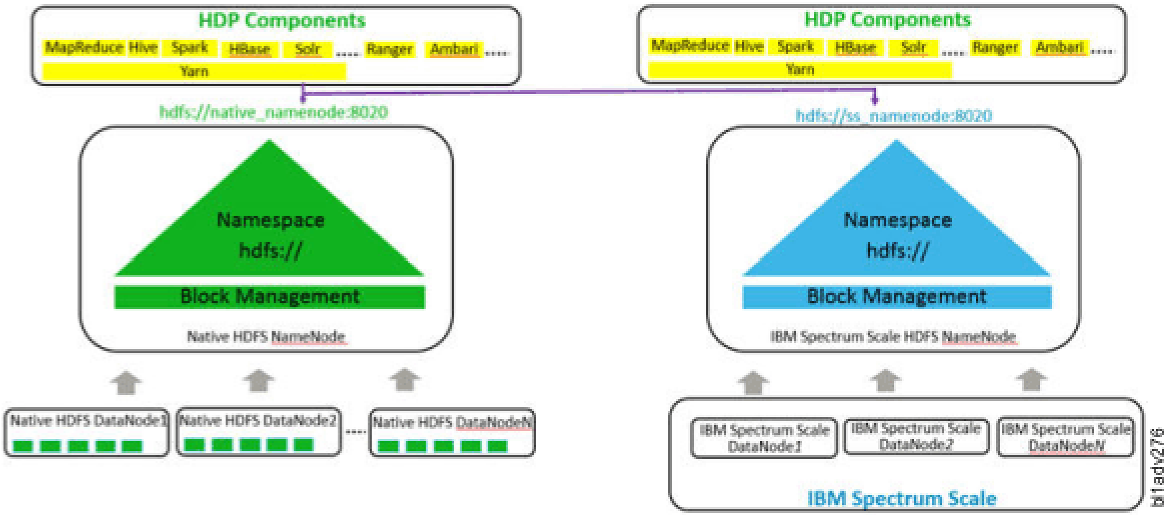


Figure 15. Hadoop Storage Tiering with IBM Storage Scale with HDP clusters

The architecture for the Hadoop Storage Tiering has a native HDFS cluster (local cluster), seen on the left hand side, and an IBM Storage Scale HDFS Transparency cluster (remote cluster), seen on the right hand side. The jobs running on the native HDFS cluster can access the data from the native HDFS or from the IBM Storage Scale HDFS Transparency cluster according to the input or output data path or from the metadata path. For example, Hive job from Hive metadata path.

Note: The Hadoop cluster deployed on the IBM Storage Scale HDFS Transparency cluster side is not a requirement for Hadoop Storage Tiering with IBM Storage Scale solution. This Hadoop cluster deployed on the IBM Storage Scale HDFS Transparency cluster side shows that a Hadoop cluster can access data via HDFS or POSIX from the IBM Storage Scale file system.

This documentation configuration setup was done without the HDP components on the remote cluster.

This document used the following software versions for testing:

Clusters	Stack	Version
HDP cluster	Ambari	2.6.1.0
	HDP	2.6.4.0
	HDP-Utils	1.1.0.22
IBM Storage Scale & HDFS Transparency cluster	IBM Storage Scale	5.0.0
	HDFS Transparency	2.7.3-2
	IBM Storage Scale Ambari management pack	2.4.2.4

Common configuration

Setup local native HDFS cluster

To setup the local native HDFS cluster:

- Follow the HDP guide from Hortonworks to set up the native HDFS cluster.
- Refer to Enable Kerberos section to setup Kerberos and to Enable Ranger section to setup Ranger in a Hadoop Storage Tiering configuration.

Setup remote HDFS Transparency cluster
This topic lists the steps to setup remote HDFS Transparency cluster.

To setup the remote IBM Storage Scale HDFS Transparency cluster, follow the steps listed below:

Option 1: IBM Storage Scale and HDFS Transparency cluster

This configuration is just for storage and does not have any Hadoop components.

1. Follow “Installing” on page 29 and “Configuring” on page 52 to set up the IBM Storage Scale HDFS Transparency cluster.
2. Refer to “Enable Kerberos” on page 143 section to setup Kerberos and “Enable Ranger” on page 147 section to setup Ranger in a Hadoop Storage Tiering configuration.

Option 2: HDP with IBM Storage Scale and HDFS Transparency integrated cluster

1. Follow the “Installation” on page 359 topic to setup HDP and IBM Storage Scale HDFS Transparency cluster.
2. Refer to “Enable Kerberos” on page 143 section to setup Kerberos and “Enable Ranger” on page 147 section to setup Ranger in a Hadoop Storage Tiering configuration.

Fixing Hive schema on local Hadoop cluster

After the local native HDFS cluster and the remote IBM Storage Scale HDFS Transparency cluster are deployed, follow these steps on the local native HDFS cluster to avoid issues with the Hive schema being changed after restarting the Hive Server2. Hive Server2 is a component from the Hive service.

Replace the following <> with your cluster specific information:

- `<ambari-user>:<ambari-password>` - Login and password used for Ambari
- `<ambari-server>:<ambari-port>` - The URL used to access the Ambari UI
- `<cluster-name>` Refers to the cluster name. The cluster name is located at the top left side of the Ambari panel in between the Ambari logo and the Background Operations (ops) icon.

On the local Hadoop cluster:

1. Get the cluster environment tag version.

   ```
   curl -u <ambari-user>:<ambari-password> -H "X-Requested-By: ambari" -X
   GET http://<ambari-server>:<ambari-port>/api/v1/clusters/<cluster-name>/configurations?
   type=cluster-env
   ```

   Note: By default, the **`cluster-env`** tag is at *version1* if the **`cluster-env`** was never updated. However, if the **`cluster-env`** was updated, you need to check manually the latest version to use.

2. Save the specific tag version cluster environment into the cluster_env.curl file by running the following command:

   ```
   curl -u <ambari-user>:<ambari-password> -H "X-Requested-By: ambari" -X
   GET http://<ambari-server>:<ambari-port>/api/v1/clusters/<cluster-name>/configurations?
   type=cluster-env&
   tag=<tag_version_found> > cluster_env.curl
   ```

   For example, running the command on the Ambari server host:

   ```
   [root@c16f1n07 ~]# curl  -u admin:admin -H "X-Requested-By: ambari" -X
   GET "http://localhost:8080/api/v1/clustershdfs264/configurations?type=cluster-
   env&tag=version1"
   ```

3. Copy the `cluster_env.curl` file into `cluster_env.curl_new` and modify the `cluster_env.curl_new` with the following information:

   a. Set the **`manage_hive_fsroot`** field to *false*.

   b. If Kerberos is enabled, set the **`security_enabled`** field to *true*.

   c. Modify the beginning of the `cluster_env.curl_new`

      From:

      ```
      {
       "href" : "http://localhost:8080/api/v1/clusters/hdfs264/configurations?type=cluster-
      env&tag=version1",
       "items" : [
         {
           "href" : "http://localhost:8080/api/v1/clusters/hdfs264/configurations?type=cluster-
      env&tag=version1",
           "tag" : "version1",
           "type" : "cluster-env",
           "version" : 1,
           "Config" : {
             "cluster_name" : "hdfs264",
             "stack_id" : "HDP-2.6"
           },
      ```

      To:

      ```
      {
          "tag" : "version2",
           "type" : "cluster-env",
           "version" : 2,
           "Config" : {
             "cluster_name" : "hdfs264",
      ```

```
        "stack_id" : "HDP-2.6"
      },
```

Note: Ensure that the tag and version are bumped up accordingly based on the last numeric value if the `cluster-env` was updated from the default value of *1*.

d. Remove the last symbol] and } at the end of the `cluster_env.curl_new` file.

4. Run the following command after replacing the “/path/to” with the real path to the cluster_env.curl_new file to POST the update:

   ```
   curl   -u <ambari-user>:<ambari-password> -H "X-Requested-By: ambari" -X
   POST "http://<ambari-server>:<ambari-port>/api/v1/clusters/<cluster-name>/configurations"
   --data @/path/to/cluster_env.curl_new
   ```

   For example, on the Ambari server host, run:

   ```
   [root@c16f1n07 ~]# curl   -u admin:admin -H "X-Requested-By: ambari" -X
   POST "http://localhost:8080/api/v1/clusters/hdfs264/configurations" --data
   @cluster_env.curl_new
   ```

5. Run the following command to PUT the update:

   ```
   curl -u <ambari-user>:<ambari-password> -H "X-Requested-By: ambari" -X
   PUT "http://<ambari-server>:<ambari-port>/api/v1/clusters/<cluster-name>" -d $'{
    "Clusters": {
      "desired_config": {
        "type": "cluster-env",
        "tag": "version2"
      }
    }
   }'
   ```

   For example, on the Ambari server host, run:

   ```
   [root@c16f1n07 ~]# curl -u admin:admin -H "X-Requested-By: ambari" -X
   PUT "http://localhost:8080/api/v1/clusters/hdfs264" -d $'{
   >  "Clusters": {
   >    "desired_config": {
   >      "type": "cluster-env",
   >      "tag": "version2"
   >    }
   >  }
   > }'
   ```

Verifying environment
Refer to the “Hadoop test case scenarios” on page 150 on how to test and leverage Hadoop Storage Tiering with IBM Storage Scale.

Enable Kerberos

To enable Kerberos on the native HDFS cluster, the native HDFS cluster and the remote IBM Storage Scale HDFS Transparency cluster requires to have the same Kerberos principals for the HDFS service.

After setting up the local native HDFS cluster and the remote HDFS Transparency cluster based on the Common configuration section, following these additional steps to configure Kerberos:

1. Enable Kerberos on the local native HDFS/HDP cluster by installing a new MIT KDC by following the Hortonworks documentation for Configuring Ambari and Hadoop for Kerberos.
2. Perform the following configuration changes on the remote HDFS Transparency cluster:

   For cluster with Ambari:

   a. Follow the “Setting up KDC server and enabling Kerberos” on page 425, using the MIT KDC server already setup in the above so as to manage the same test user account (such as hdp-user1 in below examples) Principal/Keytab on both local native HDFS cluster and remote IBM Storage Scale HDFS Transparency cluster.
   b. By default, HDP configures the service principals followed by the cluster name. If the remote HDFS Transparency cluster has the same cluster name as the local native HDFS/HDP cluster, the default

principal values on either of these clusters need to manually configure its Kerberos principals to have a different service principal name.

For example, if the remote HDFS Transparency cluster's cluster name is `REMOTE`, the default principal for the remote HDFS Transparency HDFS service is set as `hdfs-REMOTE@{REALMNAME}`. If the local native HDFS/HDP cluster also have the same cluster name (`REMOTE`), the HDFS service on the native HDFS/HDP cluster will fail to start.

If you do not change the remote HDFS Transparency cluster service principals, one can change the local native HDFS/HDP cluster default service principal to another value like `hdfs@{REALMNAME}`.

c. If the remote HDFS Transparency cluster has a different cluster name than the local native HDFS/HDP cluster, Kerberos can be enabled by following the "Setting up KDC server and enabling Kerberos" on page 425 section. After enabling Kerberos, add all the service principal rules from the local native cluster to the remote HDFS Transparency cluster.

For example, go to **Ambari** > **Services** > **HDFS** > **CONFIGS** > **Advanced core-site** > **hadoop.security.auth_to_local** to find all the service principals and copy them to the remote Transparency cluster.

```
RULE:[1:$1@$0](accumulo-hdptest@IBM.COM)s/.*/accumulo/
RULE:[1:$1@$0](ambari-qa-hdptest@IBM.COM)s/.*/ambari-qa/
RULE:[1:$1@$0](druid-hdptest@IBM.COM)s/.*/druid/
RULE:[1:$1@$0](hbase-hdptest@IBM.COM)s/.*/hbase/
RULE:[1:$1@$0](hdfs-hdptest@IBM.COM)s/.*/hdfs/
RULE:[1:$1@$0](spark-hdptest@IBM.COM)s/.*/spark/
RULE:[1:$1@$0](tracer-hdptest@IBM.COM)s/.*/accumulo/
RULE:[1:$1@$0](yarn-ats-hdptest@IBM.COM)s/.*/yarn-ats/
RULE:[1:$1@$0](zeppelin-hdptest@IBM.COM)s/.*/zeppelin/
```

For cluster without Ambari:

Note: From HDFS Transparency version 3.x, the HDFS Transparency configuration directory is changed from `/usr/lpp/mmfs/hadoop/etc/hadoop` to `/var/mmfs/hadoop/etc/hadoop`. Ensure that the correct directory paths are used with the corresponding changes when manually configuring HDFS Transparency.

a. Do not copy the `hadoop-env.sh` from the local native HDFS/HDP cluster to the HDFS Transparency cluster.

b. If **`dfs.client.read.shortcircuit`** is *true*, run the following command on one of the HDFS Transparency nodes. Otherwise, the HDFS Transparency DataNode fails to start.

```
/usr/lpp/mmfs/bin/mmdsh -N all “chown root:root -R /var/lib/hadoop-hdfs”
```

No change is required on the HDFS Transparency cluster if the **`dfs.client.read.shortcircuit`** is set to *false* in the `hdfs-site.xml` on the local native HDFS cluster.

c. Copy the configuration files, `core-site.xml` and `hdfs-site.xml`, located in `/etc/hadoop/conf` from the local native HDFS cluster to `/usr/lpp/mmfs/hadoop/etc/hadoop` on one of node from the HDFS Transparency cluster.

d. Change the NameNode value from the local native HDFS cluster NameNode to the HDFS Transparency NameNode on the HDFS Transparency node selected in "2.c" on page 144 for both the `core-site.xml` and `hdfs-site.xml` files.

e. Remove the property **`net.topology.script.file.name`** in `/usr/lpp/mmfs/hadoop/etc/hadoop/core-site.xml` and remove the property `dfs.hosts.exclude` and secondary NameNode related properties **`dfs.namenode.secondary.http-address`**, **`dfs.namenode.checkpoint.dir`**, **`dfs.secondary.namenode.kerberos.internal.spnego.principal`**, **`dfs.secondary.namenode.kerberos.principal`**, **`dfs.secondary.namenode.keytab.file`** in `/usr/lpp/mmfs/hadoop/etc/hadoop/hdfs-site.xml` on the HDFS Transparency node selected in "2.c" on page 144.

f. On the HDFS Transparency node selected in "2.c" on page 144, run **/usr/lpp/mmfs/bin/mmhadoopctl connector syncconf /usr/lpp/mmfs/hadoop/etc/hadoop** to sync all these changes into the other HDFS Transparency nodes.

3. Enable Kerberos on the remote HDFS Transparency cluster.

   For cluster with Ambari

   a. Follow the "Enabling Kerberos when the IBM Spectrum Scale service is integrated" on page 425 to enable Kerberos on IBM Storage Scale HDFS Transparency cluster.

   b.

   For cluster without Ambari:

   a. Ensure the HDFS Transparency cluster is not in running status.

   ```
   /usr/lpp/mmfs/bin/mmhadoopctl connector status
   ```

   b. Using the same KDC server with the local native HDFS/HDP cluster.

   c. Install the Kerberos clients package on all the HDFS Transparency nodes.

   ```
   yum install -y krb5-libs krb5-workstation
   ```

   d. Sync the KDC Server config, /etc/krb5.conf, to the Kerberos clients (All the HDFS Transparency nodes).

   HDFS Transparency principals and keytabs list information:

Component	Principal name	Keytab File Name
NameNode	nn/ $NN_Host_FQDN@REALMS	nn.service.keytab
NameNode HTTP	HTTP/ $NN_Host_FQDN@REALMS	spnego.service.keytab
DataNode	dn/ $DN_Host_FQDN@REALMS	dn.service.keytab

   Note: Replace the NN_Host_FQDN with your HDFS Transparency NameNode hostname and replace the DN_Host_FQDN with your HDFS Transparency DataNode hostname. If HDFS Transparency NameNode HA is configured, you need to have two principals for both NameNodes. It is required to have one principal for each HDFS Transparency DataNode.

   e. Add the principals above to the Kerberos database on the KDC Server.

   ```
   #kadmin.local
   #kadmin.local:  add_principal -randkey nn/$NN_Host_FQDN@REALMS
   #kadmin.local:  add_principal -randkey HTTP/$NN_Host_FQDN@REALMS
   #kadmin.local:  add_principal -randkey dn/$DN_Host_FQDN@REALMS
   ```

   Note: Replace the NN_Host_FQDN and DN_Host_FQDN with your cluster information. It is required to have one principal for each HDFS Transparency DataNode.

   f. Create a directory for the keytab directory and set the appropriate permissions on each of the HDFS Transparency node.

   ```
   mkdir -p /etc/security/keytabs/
   chown root:root /etc/security/keytabs
   chmod 755 /etc/security/keytabs
   ```

   g. Generate the keytabs for the principals.

   ```
   #xst -norandkey -k /etc/security/keytabs/nn.service.keytab  nn/$NN_Host_FQDN@REALMS

   #xst -norandkey -k /etc/security/keytabs/spnego.service.keytab  HTTP/$NN_Host_FQDN@REALMS
   ```

```
#xst -norandkey -k /etc/security/keytabs/dn.service.keytab  dn/$DN_Host_FQDN@REALMS
```

Note: Replace the NN_Host_FQDN and DN_Host_FQDN with your cluster information. It is required to have one principal for each HDFS Transparency DataNode.

h. Copy the appropriate keytab file to each host. If a host runs more than one component (for example, both NameNode and DataNode), copy the keytabs for both components.

i. Set the appropriate permissions for the keytab files.

On the HDFS Transparency NameNode host(s):

```
chown root:hadoop /etc/security/keytabs/nn.service.keytab
chmod 400 /etc/security/keytabs/nn.service.keytab
chown root:hadoop /etc/security/keytabs/spnego.service.keytab
chmod 440 /etc/security/keytabs/spnego.service.keytab
```

On the HDFS Transparency DataNode hosts:

```
chown root:hadoop /etc/security/keytabs/dn.service.keytab
chmod 400 /etc/security/keytabs/dn.service.keytab
```

j. Start the HDFS Transparency service from any one of the HDFS Transparency node with root passwordless ssh access to all the other HDFS Transparency nodes:

```
/usr/lpp/mmfs/bin/mmhadoopctl connector start
```

4. Validate the local native HDFS cluster when Kerberos is enabled by running a MapReduce wordcount workload.

a. Create user such as *hdp-user1* and *hdp-user2* on all the nodes of the local native HDFS cluster and the remote HDFS Transparency cluster (For example, *c16f1n07.gpfs.net* is the local native HDFS cluster NameNode, *c16f1n03.gpfs.net* is the remote HDFS Transparency cluster NameNode).

```
kinit -k -t /ect/security/keytabs/hdptestuser.headless.keytab hdp-user1@IBM.COM
```

b. The MapReduce wordcount workload by hdp-user1 and hdp-user2 will failed on the local native HDFS cluster node.

```
[root@c16f1n07 ~]# su hdp-user2
[hdp-user2@c16f1n07 root]$ klist
klist: Credentials cache file '/tmp/krb5cc_11016' not found
[hdp-user2@c16f1n07 root]$ yarn jar /usr/hdp/current/hadoop-mapreduce-client/hadoop-
mapreduce-examples.jar
wordcount hdfs://c16f1n07.gpfs.net:8020/user/hdp-user1/redhat-release
hdfs://c16f1n03.gpfs.net:8020/user/hdp-user1/redhat-release-wordcount
18/03/05 22:29:26 INFO client.RMProxy: Connecting to ResourceManager at c16f1n08.gpfs.net/
192.0.2.1:8050
18/03/05 22:29:27 INFO client.AHSProxy: Connecting to Application History server at
c16f1n08.gpfs.net/192.0.2.1:10200
18/03/05 22:29:27 WARN ipc.Client: Exception encountered while connecting to the server :
javax.security.sasl.SaslException: GSS initiate failed [Caused by GSSException: No valid
credentials
provided (Mechanism level: Failed to find any Kerberos tgt)]
java.io.IOException: Failed on local exception: java.io.IOException:
javax.security.sasl.SaslException:
GSS initiate failed [Caused by GSSException: No valid credentials provided (Mechanism
level:
Failed to find any Kerberos tgt)]; Host Details : local host is: "c16f1n07/192.0.2.0";
destination host is: "c16f1n03.gpfs.net":8020;
    at org.apache.hadoop.net.NetUtils.wrapException(NetUtils.java:785)
    at org.apache.hadoop.ipc.Client.getRpcResponse(Client.java:1558)
    at org.apache.hadoop.ipc.Client.call(Client.java:1498)
    at org.apache.hadoop.ipc.Client.call(Client.java:1398)
    at org.apache.hadoop.ipc.ProtobufRpcEngine$Invoker.invoke(ProtobufRpcEngine.java:233)
    at com.sun.proxy.$Proxy10.getDelegationToken(Unknown Source)
    at
org.apache.hadoop.hdfs.protocolPB.ClientNamenodeProtocolTranslatorPB.getDelegationToken
(ClientNamenodeProtocolTranslatorPB.java:985)
```

c. To fix the MapReduce wordcount workload error, generate the principal and keytab for user *hdp-user1* on the KDC server.

```
# kadmin.local
#kadmin.local:  add_principal -randkey hdp-user1
WARNING: no policy specified for hdp-user1@IBM.COM; defaulting to no policy
Principal "hdp-user1@IBM.COM" created.
kadmin.local:  xst -norandkey -k /etc/security/keytabs/hdptestuser.headless.keytab hdp-
user1@IBM.COM
Entry for principal hdp-user1@IBM.COM with kvno 1, encryption type aes256-cts-hmac-
sha1-96
added to keytab WRFILE:/etc/security/keytabs/hdptestuser.headless.keytab.
Entry for principal hdp-user1@IBM.COM with kvno 1, encryption type aes128-cts-hmac-
sha1-96
added to keytab WRFILE:/etc/security/keytabs/hdptestuser.headless.keytab.
Entry for principal hdp-user1@IBM.COM with kvno 1, encryption type des3-cbc-sha1 added to
keytab WRFILE:/etc/security/keytabs/hdptestuser.headless.keytab.
Entry for principal hdp-user1@IBM.COM with kvno 1, encryption type arcfour-hmac added to
keytab WRFILE:/etc/security/keytabs/hdptestuser.headless.keytab.
kadmin.local:
```

d. Copy the hdp-user1 keytab to all the nodes of the local native HDFS cluster and the remote HDFS Transparency cluster and change the permission for the *hdp-user1* keytab file.

```
[root@c16f1n07 keytabs]#pwd
/etc/security/keytabs
[root@c16f1n07 keytabs]# chown hdp-user1 /etc/security/keytabs/hdptestuser.headless.keytab
[root@c16f1n07 keytabs]# chmod 400 /etc/security/keytabs/hdptestuser.headless.keytab
```

e. Re-run the MapReduce wordcount workload by user *hdp-user1* to ensure that no errors are seen.

Enable Ranger

To enable Ranger on the native HDFS cluster, use the Ranger from the native HDFS cluster to control the policy for both the local native HDFS cluster and remote IBM Storage Scale HDFS Transparency cluster.

After setting up the local native HDFS cluster and the remote HDFS Transparency cluster based on the Common configuration section, following these additional steps to configure Ranger:

1. Install Ranger by following the Hortonworks documentation for Installing Ranger Using Ambari.
2. Perform the following configuration changes on the remote HDFS Transparency cluster:

   For cluster with Ambari:

   a. To enable Ranger on IBM Storage Scale HDFS Transparency cluster, see "Enabling Ranger" on page 417.

   For cluster without Ambari:

   Note: From HDFS Transparency version 3.x, the HDFS Transparency configuration directory is changed from `/usr/lpp/mmfs/hadoop/etc/hadoop` to `/var/mmfs/hadoop/etc/hadoop`. Ensure that the correct directory paths are used with the corresponding changes when manually configuring HDFS Transparency.

   a. Do not copy the hadoop-env.sh from HDP cluster to the HDFS Transparency cluster.
   b. If **`dfs.client.read.shortcircuit`** is *true*, run the following command on one of the HDFS Transparency nodes. Otherwise, the HDFS Transparency DataNode fails to start.

   ```
   /usr/lpp/mmfs/bin/mmdsh -N all "chown root:root /var/lib/hadoop-hdfs"
   ```

   No change is required on the HDFS Transparency cluster if the **`dfs.client.read.shortcircuit`** is set to **`false`** in the `hdfs-site.xml` on the local native HDFS cluster.

   c. Copy the configuration files, `core-site.xml` and `hdfs-site.xml`, located in `/etc/hadoop/conf` from the local native HDFS cluster to `/usr/lpp/mmfs/hadoop/etc/hadoop` on one of node from the HDFS Transparency cluster.

d. Change the NameNode value from the local native HDFS cluster NameNode to the HDFS Transparency NameNode on the HDFS Transparency node selected in step 2.c for both the `core-site.xml` and `hdfs-site.xml` files.

e. Remove the property **`net.topology.script.file.name`** in `/usr/lpp/mmfs/hadoop/etc/hadoop/core-site.xml` and remove the property **`dfs.hosts.exclude`** in `/usr/lpp/mmfs/hadoop/etc/hadoop/hdfs-site.xml` and secondary NameNode related properties **`dfs.namenode.secondary.http-address`**, **`dfs.namenode.checkpoint.dir`** on the HDFS Transparency node selected in step 2.c.

f. On the HDFS Transparency node selected in step 2.c, run **`/usr/lpp/mmfs/bin/mmhadoopctl connector syncconf /usr/lpp/mmfs/hadoop/etc/hadoop`** to sync all these changes into the other HDFS Transparency nodes.

3. Enable Ranger on the remote HDFS Transparency cluster.

   For cluster with Ambari:

   After ranger configured, ensure that all services from Ambari GUI are started successfully and Run Service Check to ensure that no issue is caused by enabling ranger.

   For cluster without Ambari:

   a. Ensure that the HDFS Transparency cluster is not in running status.

   ```
   /usr/lpp/mmfs/bin/mmhadoopctl connector status
   ```

   b. From the `/etc/hadoop/conf` directory, copy the `ranger-hdfs-audit.xml`, `ranger-hdfs-security.xml`, `ranger-policymgr-ssl.xml` and `ranger-security.xml` from the local native HDFS cluster into the `/usr/lpp/mmfs/hadoop/etc/hadoop` directory on all the nodes in the HDFS Transparency cluster.

   c. Check the value **`gpfs.ranger.enabled`** on **`gpfs-site.xml`**. The default value is set to true even if it is not configured in the `/usr/lpp/mmfs/hadoop/etc/hadoop/gpfs-site.xml` file. If it is *false*, set it to *true*.

   Note: From HDFS Transparency 3.1.0-6 and 3.1.1-3, ensure that the **`gpfs.ranger.enabled`** field is set to *scale*. The `scale` option replaces the original *true*/*false* values.

   d. Add the following to the `hadoop_env.sh` on the HDFS Transparency NameNode:

   For HDP 2.6.x:

   ```
   for f in /usr/hdp/2.6.4.0-65/ranger-hdfs-plugin/lib/*.jar; do
     export HADOOP_CLASSPATH=$HADOOP_CLASSPATH:$f
   done

   for f in /usr/share/java/mysql-connector-java.jar; do
     export HADOOP_CLASSPATH=$HADOOP_CLASSPATH:$f
   done
   ```

   For HDP 3.x:

   ```
   for f in /usr/hdp/<your-HDP-version>/ranger-hdfs-plugin/lib/*.jar;
   do
     export HADOOP_CLASSPATH=$HADOOP_CLASSPATH:$f
   done

     export HADOOP_CLASSPATH=$HADOOP_CLASSPATH:/usr/share/java/mysql-connector-java.jar

   for f in /usr/hdp/<your-HDP-version>/hadoop/client/jersey-client.jar;
   do
     export HADOOP_CLASSPATH=$HADOOP_CLASSPATH:$f
   done
   ```

   e. As root, create a directory for the above command on the HDFS Transparency NameNode.

   ```
   mkdir -p /usr/hdp/2.6.4.0-65/ranger-hdfs-plugin/lib
   mkdir -p /usr/share/java/
   ```

   Note: Change the version string `2.6.4.0-65` value based on your HDP stack version.

   f. Copy the Ranger enablement dependency path from any one node in the local native HDFS cluster node to the HDFS Transparency NameNode:

   ```
   scp -r {$NATIVE_HDFS_NAMENODE} /usr/share/java/* {$HDFS_Trans_NAMENODE}:/usr/share/java/
   scp -r {$ NATIVE_HDFS_NAMENODE} /usr/hdp/2.6.4.0-65/ranger-hdfs-plugin/lib/*
   {$HDFS_Trans_NAMENODE}:/usr/hdp/2.6.4.0-65/ranger-hdfs-plugin/lib/
   ```

   Note: Replace the **`NATIVE_HDFS_NAMENODE`** with your hostname of the local native HDFS NameNode.

   Replace the **`HDFS_Trans_NAMENODE`** with your hostname of the HDFS Transparency NameNode.

   g. To start the HDFS Transparency cluster, issue the **`/usr/lpp/mmfs/bin/mmhadoopctl connector start`** command.

4. Validate the local native HDFS and the HDFS Transparency cluster when Ranger is enabled.

   a. Create user such as *hdp-user1* on all nodes of the local native HDFS cluster and the HDFS Transparency cluster (For example, *c16f1n07.gpfs.net* is the local native HDFS cluster NameNode, *c16f1n03.gpfs.net* is the remote HDFS Transparency cluster NameNode).

   b. The `/user/hive` directory in the remote HDFS Transparency cluster is created with `rwxr-xr-x` permission for the *hdp-user1* user.

   ```
   [hdp-user1@c16f1n07 root]$ hadoop fs -ls -d hdfs://c16f1n03.gpfs.net:8020/user/hive
   drwxr-xr-x   - root root          0 2018-03-05 04:23 hdfs://c16f1n03.gpfs.net:8020/user/
   hive
   ```

   c. The Hive CLI command fails to write data to the local native HDFS cluster or to the HDFS Transparency cluster due to permission error.

   ```
   hive> CREATE DATABASE remote_db_gpfs_2 COMMENT 'Holds the tables data in remote location GPFS cluster'
   LOCATION
   'hdfs://c16f1n03.gpfs.net:8020/user/hive/remote_db_gpfs_2';
   FAILED: Execution Error, return code 1 from org.apache.hadoop.hive.ql.exec.DDLTask.
   MetaException(message:java.security.AccessControlException:
   Permission denied: user=hdp-user1, access=WRITE, inode="/user/hive":root:root:drwxr-xr-x
       at org.apache.hadoop.hdfs.server.namenode.FSPermissionChecker.check(FSPermissionChecker.java:319)
       at
   org.apache.hadoop.hdfs.server.namenode.FSPermissionChecker.checkPermission(FSPermissionChecker.java:219
   )
       at
   org.apache.hadoop.hdfs.server.namenode.GPFSPermissionChecker.checkPermission(GPFSPermissionChecker.java
   :86)
       at
   org.apache.ranger.authorization.hadoop.RangerHdfsAuthorizer$RangerAccessControlEnforcer.checkDefaultEnf
   orcer
   (RangerHdfsAuthorizer.java:428)
       at org.apache.ranger.authorization.hadoop.RangerHdfsAuthorizer$RangerAccessControlEnforcer.
   checkPermission(RangerHdfsAuthorizer.java:304)
   ```

   d. Log in to the Ranger admin web URL to create the policy to assign the RWX on the `/user/hive` directory for the *hdp-user1* user on the local native HDFS cluster and the HDFS Transparency cluster.

   e. Re-run the Hive CLI command to ensure that no errors are seen.

Hadoop test case scenarios

This section describes test cases ran on the local Hadoop cluster with Hadoop Storage Tiering configuration.

MapReduce cases without Kerberos

Test case name	Step	Description
Word count	1	Put the local file `/etc/redhat-release` into native HDFS.
	2	Put the local file `/etc/redhat-release` into IBM Storage Scale HDFS Transparency cluster
	3	Run the MapReduce WordCount job with input from the native HDFS and generate output to IBM Storage Scale HDFS Transparency cluster.
	4	Run the MapReduce WordCount job with input from the IBM Storage Scale HDFS Transparency cluster and generate output to the native HDFS.

Running MapReduce without Kerberos test

1. Run a MapReduce WordCount job with input from the local native HDFS cluster and generate the output to the remote HDFS Transparency cluster.

```
sudo -u hdfs yarn jar /usr/hdp/current/hadoop-mapreduce-client/hadoop-mapreduce-examples.jar
wordcount hdfs://c16f1n07.gpfs.net:8020/tmp/mr/passwd hdfs://c16f1n03.gpfs.net:8020/tmp/mr/

sudo -u hdfs hadoop fs -ls -R hdfs://c16f1n03.gpfs.net:8020/tmp/mr
-rw-r--r--   3 hdfs root          0 2018-03-11 23:13 hdfs://c16f1n03.gpfs.net:8020/tmp/mr/
_SUCCESS
-rw-r--r--   1 hdfs root       3358 2018-03-11 23:13 hdfs://c16f1n03.gpfs.net:8020/tmp/mr/
part-r-00000
```

2. Run a MapReduce WordCount job with input from the remote HDFS Transparency cluster and generate output to the local native HDFS cluster.

```
sudo -u hdfs yarn jar /usr/hdp/current/hadoop-mapreduce-client/hadoop-mapreduce-examples.jar
wordcount hdfs://c16f1n03.gpfs.net:8020/tmp/mr/passwd hdfs://c16f1n07.gpfs.net:8020/tmp/mr/

hadoop fs -ls -R hdfs://c16f1n07.gpfs.net:8020/tmp/mr/
-rw-r--r--   3 hdfs hdfs          0 2018-03-11 23:30 hdfs://c16f1n07.gpfs.net:8020/tmp/mr/
_SUCCESS
-rw-r--r--   3 hdfs hdfs         68 2018-03-11 23:30 hdfs://c16f1n07.gpfs.net:8020/tmp/mr/
part-r-00000
```

Spark cases without Kerberos cases

Test case name	Step	Description
Line count and word count	1	Put the local file `/etc/passwd` into native HDFS.
	2	Put the local file `/etc/passwd` into IBM Storage Scale HDFS Transparency cluster.
	3	Run the Spark LineCount/ WordCount job with input from the native HDFS and generate output to the IBM Storage Scale HDFS Transparency cluster.
	4	Run the Spark LineCount/ WordCount job with input from the IBM Storage Scale HDFS Transparency cluster and generate output to the native HDFS.

Running Spark test
Run the Spark shell to perform a word count with input from the local native HDFS and generate output to the remote HDFS Transparency cluster.

This example uses the Spark Shell (spark-shell).

1. Read the text file from the local native HDFS cluster.

```
val lines = sc.textFile("hdfs://c16f1n07.gpfs.net:8020/tmp/passwd")
```

2. Split each line into words and flatten the result.

```
val words = lines.flatMap(_.split("\\s+"))
```

3. Map each word into a pair and count them by word (key).

```
val wc = words.map(w => (w, 1)).reduceByKey(_ + _)
```

4. Save the result in text files on the remote HDFS Transparency cluster.

```
wc.saveAsTextFile("hdfs://c16f1n03.gpfs.net:8020/tmp/passwd_sparkshell")
```

5. Review the contents of the `README.count` directory.

```
hadoop fs -ls -R hdfs://c16f1n03.gpfs.net:8020/tmp/passwd_sparkshell
-rw-r--r--   3 hdfs root          0 2018-03-11 23:58 hdfs://c16f1n03.gpfs.net:8020/tmp/passwd_sparkshell/
_SUCCESS
-rw-r--r--   1 hdfs root       1873 2018-03-11 23:58 hdfs://c16f1n03.gpfs.net:8020/tmp/passwd_sparkshell/
part-00000
-rw-r--r--   1 hdfs root       1679 2018-03-11 23:58 hdfs://c16f1n03.gpfs.net:8020/tmp/passwd_sparkshell/
part-00001
```

Hive-MapReduce/Tez without Kerberos cases

Test case name	Step	Descriptions
DDL operations 1. LOAD data local inpath 2. INSERT into table 3. INSERT Overwrite TABLE	1	Drop remote database if EXISTS cascade.
	2	Create *remote_db* with Hive warehouse on the IBM Storage Scale HDFS Transparency cluster.
	3	Create internal nonpartitioned table on *remote_db*.
	4	LOAD data local inpath into table created in the above step.
	5	Create internal nonpartitioned table on the remote IBM Storage Scale HDFS Transparency cluster.
	6	LOAD data local inpath into table created in the above step.
	7	Create internal transactional table on *remote_db*.
	8	INSERT into table from internal nonpartitioned table.
	9	Create internal partitioned table on *remote_db*.
	10	INSERT OVERWRITE TABLE from internal nonpartitioned table.
	11	Create external nonpartitioned table on *remote_db*.
	12	Drop local database if EXISTS cascade.
	13	Create *local_db* with, Hive warehouse on local DAS Hadoop cluster.
	14	Create internal nonpartitioned table on *local_db*.
	15	LOAD data local inpath into table created in preceding step.
	16	Create internal nonpartitioned table into the local native HDFS cluster.
	17	LOAD data local inpath into table created in the above step.
	18	Create internal transactional table on *local_db*.
	19	INSERT into table from internal nonpartitioned table.

Test case name	Step	Descriptions
	20	Create internal partitioned table on *local_db*.
	21	INSERT OVERWRITE TABLE from internal nonpartitioned table.
	22	Create external nonpartitioned table on *local_db*.
DML operations 1. Query local database tables 2. Query remote database tables	1	Query data from local external nonpartitioned table.
	2	Query data from local internal nonpartitioned table.
	3	Query data from local nonpartitioned remote data table.
	4	Query data from local internal partitioned table.
	5	Query data from local internal transactional table.
	6	Query data from remote external nonpartitioned table.
	7	Query data from remote internal nonpartitioned table.
	8	Query data from remote nonpartitioned remote data table.
	9	Query data from remote internal partitioned table.
	10	Query data from remote internal transactional table.
JOIN tables in local database	1	JOIN external nonpartitioned table with internal nonpartitioned table.
	2	JOIN internal nonpartitioned table with internal nonpartitioned remote table.
	3	JOIN internal nonpartitioned remote table with internal partitioned table.
	4	JOIN internal partitioned table with internal transactional table.
	5	JOIN internal transactional table with external nonpartitioned table.

Test case name	Step	Descriptions
JOIN tables in remote database	1	JOIN external nonpartitioned table with internal nonpartitioned table.
	2	JOIN internal nonpartitioned table with internal nonpartitioned remote table.
	3	JOIN internal nonpartitioned remote table with internal partitioned table.
	4	JOIN internal partitioned table with internal transactional table.
	5	JOIN internal transactional table with external nonpartitioned table.
JOIN tables between *local_db* and *remote_db*	1	JOIN *local_db* external nonpartitioned table with *remote_db* internal nonpartitioned table.
	2	JOIN *local_db* internal nonpartitioned table with *remote_db* internal nonpartitioned remote table.
	3	JOIN *local_db* internal nonpartitioned remote table with *remote_db* internal partitioned table.
	4	JOIN *local_db* internal partitioned table with *remote_db* internal transactional table.
	5	JOIN *local_db* internal transactional table with *remote_db* external nonpartitioned table.
Local temporary table from *remote_db* table	1	Create temporary table on *local_db* AS select query from *remote_db* table.
	2	Query data from temporary table.

Test case name	Step	Descriptions
IMPORT and EXPORT operations	1	EXPORT *local_db* internal partitioned table to the remote IBM Storage Scale HDFS Transparency cluster.
	2	List the directory/file created on the remote HDFS Transparency cluster by EXPORT operation.
	3	IMPORT table to create table in *local_db* from the EXPORT data from the above step into the remote HDFS Transparency cluster.
	4	List the directory/file created on the local native HDFS cluster by IMPORT operation.
	5	Query data from *local_db* table created by IMPORT operation.
	6	EXPORT *remote_db* external table to the local native HDFS cluster location.
	7	List the directory/file created on the local Hadoop cluster by EXPORT operation.
	8	IMPORT table to create table on *remote_db* from the EXPORT data from the preceding step on the local native HDFS cluster.
	9	List directory/file created on the remote HDFS Transparency cluster by preceding IMPORT operation.
	10	Query data from *remote_db* table created by preceding IMPORT operation.
Table-level and column-level statistics	1	Run table-level statistics command on external nonpartitioned table.
	2	Run DESCRIBE EXTENDED to check the statistics of the nonpartitioned table.
	3	Run column-level statistics command on internal partitioned table.
	4	Run DESCRIBE EXTENDED command to check the statics of the partitioned table.

Running Hive on MapReduce execution engine test

This section lists the steps to run Hive on MapReduce execution engine test.

Note: Apache Tez replaces MapReduce as the default Hive execution engine in Apache Hive 3. MapReduce is no longer supported when using Apache Hive 3.

1. Open the MapReduce execution engine interface.

   ```
   hive -hiveconf hive.execution.engine=mr
   ```

2. Create a local database location on the local native HDFS and create an internal nonpartitioned remote table.

   ```
   Hive> CREATE database local_db COMMENT 'Holds all the tables data in local Hadoop cluster'
   LOCATION 'hdfs://c16f1n07.gpfs.net:8020/user/hive/local_db'
   OK
   Time taken: 0.066 seconds

   hive> USE local_db;
   OK
   Time taken: 0.013 seconds
   hive> CREATE TABLE passwd_int_nonpart_remote (user_name STRING, password STRING, user_id
   STRING,
   group_id STRING,
   user_id_info STRING, home_dir STRING, shell STRING) ROW FORMAT DELIMITED FIELDS TERMINATED
   BY ':'
   LOCATION 'hdfs://c16f1n07.gpfs.net:8020/user/hive/local_db/passwd_int_nonpart_remote'
   OK
   Time taken: 0.075 seconds
   ```

3. Create an external nonpartitioned table on the local native HDFS cluster.

   ```
   hive> CREATE EXTERNAL TABLE passwd_ext_nonpart (user_name STRING, password STRING,
   user_id STRING, group_id STRING, user_id_info STRING, home_dir STRING, shell STRING) ROW
   FORMAT DELIMITED FIELDS TERMINATED BY ':' LOCATION
   'hdfs://c16f1n07.gpfs.net:8020/user/hive/local_db/passwd_in t_nonpart_remote'
   OK
   Time taken: 0.066 seconds
   ```

Running Hive on Tez execution engine test

This section lists the steps to run Hive on Tez execution engine test.

1. Open the Tez execution engine interface.

   ```
   hive -hiveconf hive.execution.engine=tez
   ```

2. Create a remote database location on the remote HDFS Transparency cluster and create an internal partitioned table.

   ```
   Hive> CREATE database remote_db COMMENT 'Holds all the tables data in remote HDFS
   Transparency cluster'
   LOCATION 'hdfs://c16f1n03.gpfs.net:8020/user/hive/remote_db'
   OK
   Time taken: 0.08 seconds
   hive> USE remote_db;
   OK
   Time taken: 0.238 seconds
   hive> CREATE TABLE passwd_int_part (user_name STRING, password STRING, user_id STRING,
   user_id_info STRING,
   home_dir STRING, shell STRING) PARTITIONED BY (group_id STRING) ROW FORMAT DELIMITED FIELDS
   TERMINATED BY ':';
   OK
   Time taken: 0.218 seconds
   ```

3. Create a local database location on the local native HDFS cluster and create an internal transactional table.

   ```
   hive> CREATE database local_db COMMENT 'Holds all the tables data in local Hadoop cluster'
   LOCATION 'hdfs://c16f1n07.gpfs.net:8020/user/hive/local_db';
   OK
   Time taken: 0.035 seconds
   hive> USE local_db ;
   OK
   ```

```
Time taken: 0.236 seconds
hive> CREATE TABLE passwd_int_trans (user_name STRING, password STRING, user_id STRING,
group_id STRING,
user_id_info STRING, home_dir STRING, shell STRING) CLUSTERED by(user_name) into 3 buckets
stored as orc
tblproperties ("transactional"="true");
OK
Time taken: 0.173 seconds
```

Running Hive import and export operations test

This section lists the steps for running Hive import and export operations test.

1. On the local HDFS cluster, EXPORT local_db internal partitioned table to the remote HDFS Transparency cluster.

```
hive> EXPORT TABLE local_db.passwd_int_part TO
'hdfs://c16f1n03.gpfs.net:8020/user/hive/remote_db/passwd_int_part_export';
OK
Time taken: 0.986 seconds
```

2. On the local HDFS cluster, list the directory/file that was created on the remote HDFS Transparency cluster using the EXPORT operation.

```
hive> dfs -ls hdfs://c16f1n03.gpfs.net:8020/user/hive/remote_db/passwd_int_part_export;
Found 2 items
-rw-r--r--   1 hdp-user1 root    2915 2018-03-19 21:43
hdfs://c16f1n03.gpfs.net:8020/user/hive/remote_db/passwd_int_part_export/_metadata
drwxr-xr-x   - hdp-user1 root       0 2018-03-19 21:43
hdfs://c16f1n03.gpfs.net:8020/user/hive/remote_db/passwd_int_part_export/group_id=2011-12-14
```

3. On the local HDFS cluster, IMPORT table to create a table in the local_db from the EXPORT data from the above step on the remote HDFS Transparency cluster.

```
hive> IMPORT TABLE local_db.passwd_int_part_import FROM
'hdfs://c16f1n03.gpfs.net:8020/user/hive/remote_db/passwd_int_part_export'
LOCATION 'hdfs://c16f1n07.gpfs.net:8020/user/hive/local_db/passwd_int_part_import';
Copying data from hdfs://c16f1n03.gpfs.net:8020/user/hive/remote_db/passwd_int_part_export/group_id=2011-12-14
Copying file: hdfs://c16f1n03.gpfs.net:8020/user/hive/remote_db/passwd_int_part_export/group_id=2011-12-14/lt101.sorted.txt
Loading data to table local_db.passwd_int_part_import partition (group_id=2011-12-14)
OK
Time taken: 1.166 seconds
```

4. List the directory/file created on the local native HDFS cluster by using the IMPORT operation.

```
hive> dfs -ls hdfs://c16f1n07.gpfs.net:8020/user/hive/local_db/passwd_int_part_import;
Found 1 items
drwxr-xr-x   - hdp-user1 hdfs    0 2018-03-19 21:59
hdfs://c16f1n07.gpfs.net:8020/user/hive/local_db/passwd_int_part_import/group_id=2011-12-14
```

5. Query data from the local_db table created by the IMPORT operation.

```
hive> select * from local_db.passwd_int_part_import;
OK
0  val_0     NULL    NULL    NULL    NULL    NULL    2011-12-14
0  val_0     NULL    NULL    NULL    NULL    NULL    2011-12-14
0  val_0     NULL    NULL    NULL    NULL    NULL    2011-12-14
10 val_10    NULL    NULL    NULL    NULL    NULL    2011-12-14
11 val_11    NULL    NULL    NULL    NULL    NULL    2011-12-14
12 val_12    NULL    NULL    NULL    NULL    NULL    2011-12-14
15 val_15    NULL    NULL    NULL    NULL    NULL    2011-12-14
17 val_17    NULL    NULL    NULL    NULL    NULL    2011-12-14
18 val_18    NULL    NULL    NULL    NULL    NULL    2011-12-14
24 val_24    NULL    NULL    NULL    NULL    NULL    2011-12-14
35 val_35    NULL    NULL    NULL    NULL    NULL    2011-12-14
35 val_35    NULL    NULL    NULL    NULL    NULL    2011-12-14
37 val_37    NULL    NULL    NULL    NULL    NULL    2011-12-14
…...
Time taken: 0.172 seconds, Fetched: 84 row(s)
```

TPC-DS cases

Test case name	Step	Description
Prepare Hive- testbench	1	Download latest Hive-testbench from Hortonworks github repository.
	2	Run **tpcds-build.shto** build TPC-DS data generator.
	3	Run **tpcds-setup** to set up the testbench database and load the data into created tables.
Database on remote Hadoop cluster and load data	1	Create LLAP database on remote IBM Storage Scale HDFS Transparency cluster.
	2	Create 24 tables in LLAP database required to run the Hive test benchmark queries.
	3	Check the Hadoop file system location for the 24 table directories created on the remote IBM Storage Scale HDFS Transparency cluster.
TPC-DS benchmarking	1	Switch from default database to LLAP database.
	2	Run **query52.sqlscript**.
	3	Run **query55.sqlscript**.
	4	Run **query91.sqlscript**.
	5	Run **query42.sql script**.
	6	Run **query12.sqlscript**.
	7	Run **query73.sqlscript**.
	8	Run **query20.sqlscript**.
	9	Run **query3.sqlscript**.
	10	Run **query89.sqlscript**.
	11	Run **query48.sqlscript**.

Running TPC-DS test
This topic lists the steps to run a TPC-DS test.

1. Prepare Hive-testbench by running the tpcdc-build.sh script to build the TPC-DS and the data generator. Run the tpcds-setup to set up the testbench database and load the data into the created tables.

```
cd ~/hive-testbench-hive14/

./tpcds-build.sh

./tpcds-setup.sh 2 (A map reduce job runs to create the data and load the data into hive.
This will take some time to complete. The last line in the script is: Data loaded into
database tpcds_bin_partitioned_orc_2.)
```

2. Create a new remote Low Latency Analytical Processing (LLAP) database on the remote HDFS Transparency cluster.

```
hive> DROP database if exists llap CASCADE;
hive> CREATE database if not exists llap LOCATION 'hdfs://c16f1n03.gpfs.net:8020/user/hive/
llap.db';
```

3. Create 24 tables and load data from the tables.

```
hive> DROP table if exists llap.call_center;
hive> CREATE table llap.call_center stored as orc as select * from tpcds_text_2.call_center;
```

4. Run the benchmark queries on the tables that you created on the remote LLAP database.

```
hive> use llap;
hive> source query52.sql;
hive> source query55.sql;
hive> source query91.sql;
hive> source query42.sql;
hive> source query12.sql;
hive> source query73.sql;
hive> source query20.sql;
hive> source query3.sql;
hive> source query89.sql;
hive> source query48.sql;
```

For more information, refer to the Apache Hive SQL document.

Kerberos security cases

Test case name	**Step**	**Description**
Kerberos user setup and testing	1	Create user hdp-user1 on all the nodes of HDP (local native HDFS) cluster.
	2	Add hdp-user1 principal in the Kerberos KDC server and assign password.
	3	Create home directory and assign permission for hdp-user1 in local native HDFS and IBM Storage Scale HDFS Transparency cluster with `hadoop dfs` interface.
	4	Switch to hdp-user1 in Hadoop client node and query data from the local native HDFS cluster and the remote IBM Storage Scale HDFS Transparency cluster.
	5	Put local file `/etc/redhat-release` on HDP (local native HDFS) file system with **`hadoop dfs -put`**.
	6	Put local file `/etc/redhat-release` on IBM Storage Scale HDFS Transparency cluster with **`hadoop dfs -put`**.
	7	Run MapReduce WordCount job with input from HDP (local native HDFS) and generate output to IBM Storage Scale HDFS Transparency cluster.
	8	Run MapReduce WordCount job with input from IBM Storage Scale HDFS Transparency cluster and generate output to HDP (local native HDFS).

Test case name	Step	Description
Non-Kerberos user setup and testing	1	Create user hdp-user2 in Hadoop client node.
	2	Switch to hdp-user2 in Hadoop client node and query data from the local native HDFS and the remote IBM Storage Scale HDFS Transparency cluster.
	3	Create home directory and assign permission for hdp-user2 in the local native and the remote IBM Storage Scale HDFS Transparency cluster.
	4	Put local file `/etc/redhat-release` on HDP (local native HDFS) file system.
	5	Put local file `/etc/redhat-release` on IBM Storage Scale HDFS Transparency cluster.
	6	Run MapReduce WordCount job with input from HDP (local native HDFS) and generate output to the IBM Storage Scale HDFS Transparency cluster.
	7	Run MapReduce WordCount job with input from IBM Storage Scale HDFS Transparency cluster and generate output to HDP (local native HDFS).

Ranger policy cases

Test case name	Step	Description
Access and restriction policy	1	Create directory GRANT_ACCESS on remote IBM Storage Scale HDFS Transparency cluster.
	2	Create directory RESTRICT_ACCESS on remote IBM Storage Scale HDFS Transparency cluster.
	3	Create hdp-user1 on all the nodes of both the Hadoop cluster (HDP local HDFS) and IBM Storage Scale.
	4	Assign RWX access for the hdp-user1 on GRANT_ACCESS from Ranger UI under hdp3_hadoop Service Manager.
	5	Put local file `/etc/redhat-release` into GRANT_ACCESS folder.
	6	Put local file `/etc/redhat-release` into RESTRICT_ACCESS folder.
	7	Assign only read/write access for hdp-user1 on RESTRICT_ACCESS folder from Ranger UI.
	8	Copy file from GRANT_ACCESS to RESTRICT_ACCESS folder.
	9	Assign only read access for hdp-user1 on RESTRICT_ACCESS folder from Ranger UI.
	10	Delete GRANT_ACCESS and RESTRICT_ACCESS folders.

Ranger policy cases with Kerberos security cases

Test case name	**Step**	**Description**
MapReduce (word count)	1	Create hdp-user1 home directory on HDP (local HDFS) and HDFS Transparency IBM Storage Scale.
	2	Assign RWX on `/user/hdp-user1` directory for hdp-user1 on HDP (local HDFS) and IBM Storage Scale HDFS Transparency cluster using Ranger UI.
	3	Put local file `/etc/redhat-release` on HDP (local HDFS) file system.
	4	Put local file `/etc/redhat-release` on IBM Storage Scale HDFS Transparency cluster.
	5	Run MapReduce WordCount job with input from HDP (local HDFS), and generate output to the remote IBM Storage Scale HDFS Transparency cluster.
	6	Run MapReduce WordCount job with input from IBM Storage Scale HDFS Transparency cluster and generate output to HDP (local native HDFS).
Spark (line count and word count)	1	Put local file `/etc/passwd` into HDP (local native HDFS) file system.
	2	Put local file `/etc/passwd` into IBM Storage Scale HDFS Transparency cluster.
	3	Run Spark LineCount/WordCount job with input from primary HDP (local HDFS) and generate output to the IBM Storage Scale HDFS Transparency cluster.
	4	Run Spark LineCount/WordCount job with input from remote IBM Storage Scale HDFS Transparency cluster and generate output to the primary HDP (local native HDFS) HDFS.

Ranger policy with Kerberos security on Hive warehouse cases

Test case name	Step	Description
Hive data warehouse Ranger policy setup	1	Assign RWX on `/user/hivedirectory` for hdp-user1 on HDP (local native HDFS) and IBM Storage Scale HDFS Transparency cluster using Ranger UI.
DDL operations 1. LOAD data local inpath 2. INSERT into table 3. INSERT Overwrite TABLE	1	Drop remote database if EXISTS cascade.
	2	Create `remote_db` with hive warehouse on remote IBM Storage Scale HDFS Transparency cluster.
	3	Create internal nonpartitioned table on `remote_db`.
	4	LOAD data local inpath into table created in the above step.
	5	Create internal nonpartitioned table on remote IBM Storage Scale HDFS Transparency cluster.
	6	LOAD data local inpath into table created in the above step.
	7	Create internal transactional table on `remote_db`.
	8	INSERT into table from internal nonpartitioned table.
	9	Create internal partitioned table on `remote_db`.
	10	INSERT OVERWRITE TABLE from internal nonpartitioned table.
	11	Create external nonpartitioned table on `remote_db`.

Test case name	**Step**	**Description**
DDL operations 1. LOAD data local inpath 2. INSERT into table 3. INSERT Overwrite TABLE	12	Drop local database if EXISTS cascade.
	13	Create `local_db` with hive warehouse on local native HDFS cluster.
	14	Create internal nonpartitioned table on `local_db`.
	15	LOAD data local inpath into table created in the above step.
	16	Create internal nonpartitioned table on local native HDFS cluster.
	17	LOAD data local inpath into table created in the above step.
	18	Create internal transactional table on `local_db`.
	19	INSERT into table from internal nonpartitioned table.
	20	Create internal partitioned table on `local_db`.
	21	INSERT OVERWRITE TABLE from internal nonpartitioned table.
	22	Create external nonpartitioned table on `local_db`.

Test case name	Step	Description
DML operations 1. Query local database tables 2. Query remote database tables	1	Query data from local external nonpartitioned table.
	2	Query data from local internal nonpartitioned table.
	3	Query data from local nonpartitioned remote data table.
	4	Query data from local internal partitioned table.
	5	Query data from local internal transactional table.
	6	Query data from remote external nonpartitioned table.
	7	Query data from remote internal nonpartitioned table.
	8	Query data from remote nonpartitioned remote data table.
	9	Query data from remote internal partitioned table.
	10	Query data from remote internal transactional table.
JOIN tables in local database	1	JOIN external nonpartitioned table with internal nonpartitioned table.
	2	JOIN internal nonpartitioned table with internal nonpartitioned remote table.
	3	JOIN internal nonpartitioned remote table with internal partitioned table.
	4	JOIN internal partitioned table with internal transactional table.
	5	JOIN internal transactional table with external nonpartitioned table.

Test case name	Step	Description
JOIN tables in remote database	1	JOIN external nonpartitioned table with internal nonpartitioned table.
	2	JOIN internal nonpartitioned table with internal nonpartitioned remote table.
	3	JOIN internal nonpartitioned remote table with internal partitioned table.
	4	JOIN internal partitioned table with internal transactional table.
	5	JOIN internal transactional table with external nonpartitioned table.
JOIN tables between local_db and remote_db	1	JOIN `local_db` external nonpartitioned table with `remote_db` internal nonpartitioned table.
	2	JOIN `local_db` internal nonpartitioned table with `remote_db` internal nonpartitioned remote table.
	3	JOIN `local_db` internal nonpartitioned remote table with `remote_db` internal partitioned table.
	4	JOIN `local_db` internal partitioned table with `remote_db` internal transactional table.
	5	JOIN `local_db` internal transactional table with `remote_db` external nonpartitioned table.
Local temporary table from remote_db table	1	Create temporary table on `local_db` AS select query from `remote_db` table.
	2	Query data from temporary table.

Test case name	Step	Description
IMPORT and EXPORT operations	1	EXPORT `local_db` internal partitioned table to remote IBM Storage Scale HDFS Transparency cluster location.
	2	List the directory/file created on remote Hadoop cluster by EXPORT operation.
	3	IMPORT table to create table in `local_db` from the EXPORT data on remote HDFS Transparency cluster.
	4	List the directory/file created on the local Hadoop cluster by IMPORT operation.
	5	Query data from `local_db` table created by IMPORT operation.
	6	EXPORT `remote_db` external table to the local HDP (local native HDFS) Hadoop cluster location.
	7	List the directory/file created on the local Hadoop cluster by EXPORT operation.
	8	IMPORT table to create table on `remote_db` from the EXPORT data on the local Hadoop cluster.
	9	List directory/file created on remote HDFS Transparency cluster by IMPORT operation.
	10	Query data from `remote_db` table created by the IMPORT operation.
Table-level and column-level statistics	1	Run **`table-level statistics`** command on external nonpartitioned table.
	2	Run **`DESCRIBE EXTENDED`** to check the statics of the nonpartitioned table.
	3	Run **`column-level statistics`** command on internal partitioned table.
	4	Run **`DESCRIBE EXTENDED`** to check the statics of the partitioned table.

DistCp in Kerberized and non-Kerberized cluster cases

Test Case	Step	Description
Distcp	1	Use **distcp** to copy sample file from native HDFS to remote IBM Storage Scale/HDFS Transparency cluster.
	2	Use **distcp** to copy sample file from remote HDFS Transparency cluster to local native HDFS cluster.

Running distcp in Kerberized and non-Kerberized cluster test

1. Run **distcp** to copy a sample file from the local native HDFS to the remote HDFS Transparency in a non-Kerberized cluster:

```
[hdfs@c16f1n07 root]$ hadoop distcp -skipcrccheck -update 
hdfs://c16f1n07.gpfs.net/tmp/redhat-release hdfs://c16f1n03.gpfs.net:8020/tmp

[hdfs@c16f1n07 root]$ hadoop fs -ls -R hdfs://c16f1n03.gpfs.net:8020/tmp
-rw-r--r--   1 hdfs      root         52 2018-03-19 23:26 hdfs://c16f1n03.gpfs.net:8020/tmp/
redhat-release
```

2. Run **distcp** to copy a sample file from the remote HDFS Transparency to the local native HDFS in a Kerberized cluster:

```
[hdp-user1@c16f1n07 root]$ klist
Ticket cache: FILE:/tmp/krb5cc_11015
Default principal: hdp-user1@IBM.COM
Valid starting       Expires              Service principal
03/19/2018 22:54:03  03/20/2018 22:54:03  krbtgt/IBM.COM@IBM.COM

[hdp-user1@c16f1n07 root]$ hadoop distcp -pc 
hdfs://c16f1n03.gpfs.net:8020/tmp/redhat-release hdfs://c16f1n07.gpfs.net:8020/tmp

[hdp-user1@c16f1n07 root]$ hadoop fs -ls hdfs://c16f1n07.gpfs.net:8020/tmp/redhat-release
-rw-r--r--   3 hdp-user1 hdfs         52 2018-03-20 01:30 hdfs://c16f1n07.gpfs.net:8020/tmp/
redhat-release
```

Hadoop Storage Tiering mode with native HDFS federation

The Hadoop ViewFs support is available from HDP 3.0. HDFS Transparency support of Hadoop ViewFs is available from HDP 3.1.

Note:

- Hadoop Storage Tiering mode with native HDFS federation is not supported in HortonWorks HDP 2.6.x.
- ViewFs does not support Hive.
- For CES HDFS, see Limitations and Recommendations.

For information on how to architect and configure a Hadoop Storage Tiering solution with a suite of test cases executed based on this configuration, see Managing and Monitoring a Cluster.

The Hadoop Storage Tiering with IBM Storage Scale architecture is shown in the following figures:

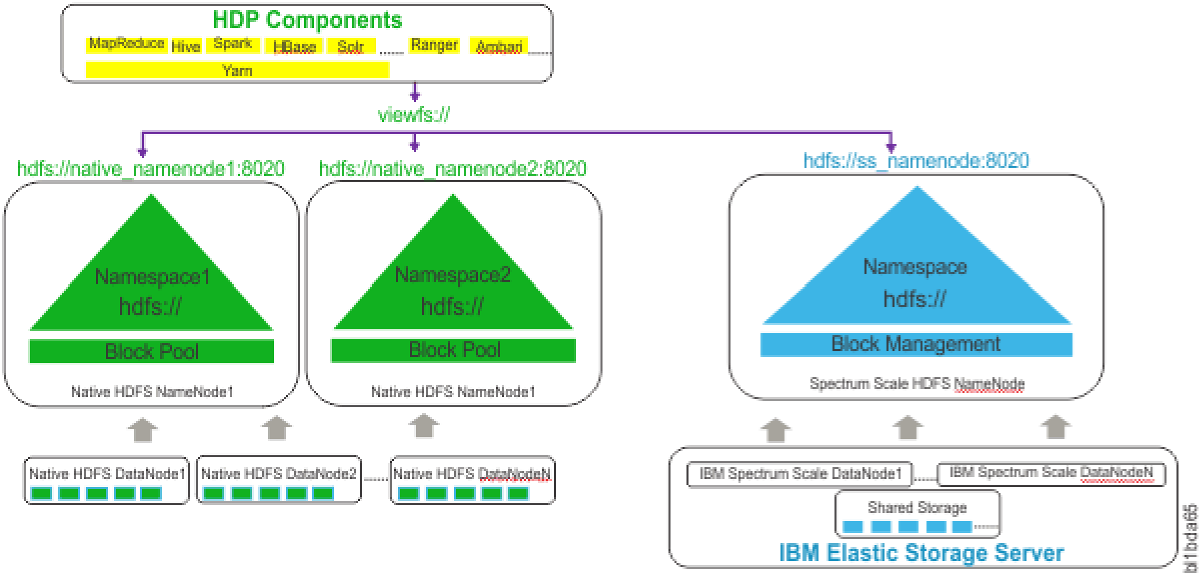


Figure 16. Hadoop Storage Tiering with IBM Storage Scale w/o HDP cluster

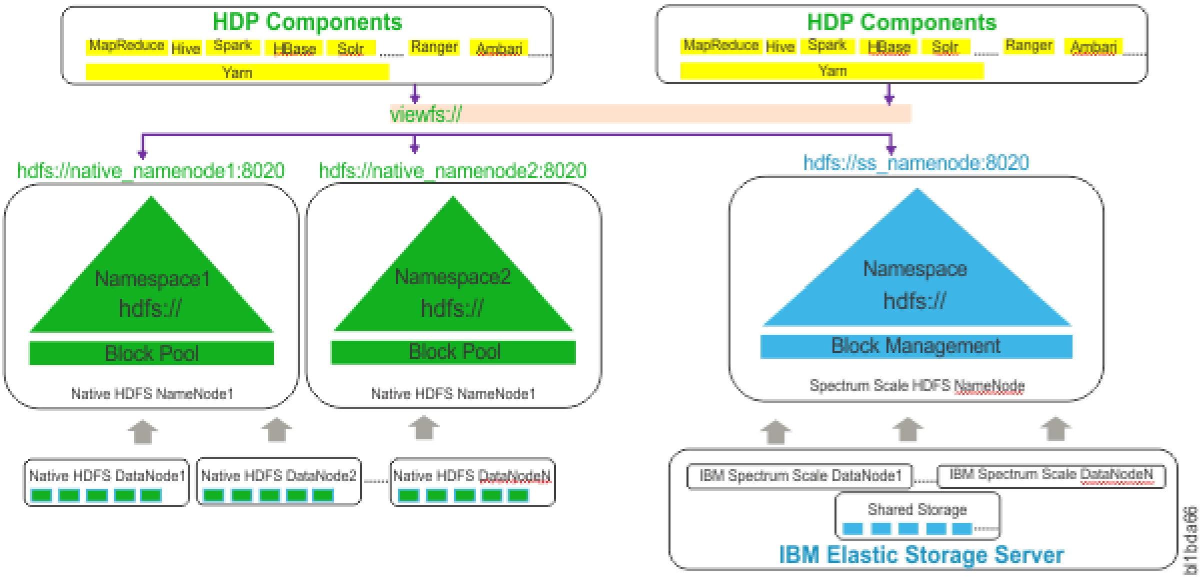


Figure 17. Hadoop Storage Tiering with IBM Storage Scale with HDP cluster

The architecture for the Hadoop Storage Tiering has one or more native HDFS clusters (local cluster) as shown on the left side of these figures. The IBM Storage Scale HDFS Transparency cluster (remote cluster), is shown on the right side of the figures. The jobs running on the native HDFS cluster can access the data from the native HDFS or from the IBM Storage Scale HDFS Transparency cluster according to the input or output data path or from the metadata path. For example, Hive job using the Hive metadata path.

Note: The Hadoop cluster deployed on the IBM Storage Scale HDFS Transparency cluster side is not a requirement for Hadoop Storage Tiering with IBM Storage Scale solution as shown in Figure 17 on page 170. This Hadoop cluster deployed on the IBM Storage Scale HDFS Transparency cluster side shows that a Hadoop cluster can access data via HDFS or POSIX from the IBM Storage Scale file system.

The following section setup was done without the HDP components on the remote cluster as shown in Figure 16 on page 170.

This section used the following software versions for testing:

Clusters	Stack	Version
HDP cluster	Ambari	2.7.3.0
	HDP	3.1.0.0
	HDP-UTILS	1.1.0.22
IBM Storage Scale & HDFS Transparency cluster	IBM Storage Scale	5.0.2.2
	HDFS Transparency	3.1.0-0
	IBM Storage Scale Ambari management pack	2.7.0.2

Common configuration

Setup local native HDFS cluster
This section lists the steps to setup local native HDFS cluster.

To setup the local native HDFS cluster:

- Follow the HDP guide from Hortonworks to set up the native HDFS cluster.
- Refer to the "Enable Kerberos" on page 143 section to setup Kerberos and the "Enable Ranger" on page 147 section to setup Ranger in a Hadoop Storage Tiering configuration.

Configure the ViewFs on Native HDFS cluster by following the "Configuring ViewFs on HDFS cluster with HA" on page 173 section.

Setup remote HDFS Transparency cluster
This section lists the steps to setup remote HDFS Transparency cluster.

Follow and one set of steps to setup the remote IBM Storage Scale HDFS Transparency cluster:

- Option 1: Configure IBM Storage Scale and HDFS Transparency cluster

  This configuration does not have any Hadoop components. This is a storage configuration setup.

  - Follow "Installing" on page 29 and "Configuring" on page 52 to set up the IBM Storage Scale HDFS Transparency cluster.
  - Refer to the "Enable Kerberos" on page 143 section to setup Kerberos and the "Enable Ranger" on page 147 section to setup Ranger in a Hadoop Storage Tiering configuration.
- Option 2: Configure HDP with IBM Storage Scale and HDFS Transparency integrated cluster
  - Follow the "Installation" on page 359 to setup HDP and IBM Storage Scale HDFS Transparency cluster.
  - Refer to the "Enable Kerberos" on page 143 section to setup Kerberos and the "Enable Ranger" on page 147 section to setup Ranger in a Hadoop Storage Tiering configuration.

Verify environment
Refer to the Hadoop test case scenarios on how to test and leverage Hadoop Storage Tiering with IBM Storage Scale.

Configure ViewFs on HDFS clusters without HA

The local native HDFS cluster is setup through Ambari.

The local HDFS and remote HDFS do not have HA setup. Either cluster can have Scale service added or not. This section describes how to configure ViewFs on the local HDFS cluster through Ambari.

Setup instructions

1. Go to **Ambari** > **HDFS** > **CONFIGS** > **ADVANCED** > **Advanced core-site** and change the **`fs.defaultFS`** from `hdfs://<namenode_host>:<port>` to `viewfs://<federation_cluster_name>`.

For example, the property **`fs.defaultFS`** field is changed to `viewfs://federationcluster`.

2. Go to **Ambari** > **HDFS** > **CONFIGS** > **Custom hdfs-site** and add or change all the following properties:

   For example, the **`c16f1n03.gpfs.net`** is the local native HDFS cluster and **`c16f1n10.gpfs.net`** is the remote IBM Storage Scale cluster.

   Note: Configure the key=value pair properties based on your cluster information.

   ```
   dfs.namenode.http-address.nn1=c16f1n03.gpfs.net:50070
   dfs.namenode.http-address.nn2=c16f1n10.gpfs.net:50070
   dfs.namenode.rpc-address.nn1=c16f1n03.gpfs.net:8020
   dfs.namenode.rpc-address.nn2=c16f1n10.gpfs.net:8020
   dfs.nameservices=nn1, nn2
   ```

3. Go to **Ambari** > **HDFS** > **CONFIGS** > **ADVANCED** > **Advanced viewfs-mount-table** and specify the ViewFs mount table entries. You must define the mount table entries to map to the physical locations to the corresponding mount points from the federated cluster. You must consider factors such as data access, mount levels, and application requirements while defining ViewFs mount table entries.

   Note: Place the `<configuration> </configuration>` entries with the corresponding `<property></property>` values based on your environment into the Advanced views-mount-table field using the xml format as shown below:

   ```
   <configuration>

     <property>
       <name>fs.viewfs.mounttable.federationcluster.link./app-logs</name>
       <value>hdfs://c16f1n03.gpfs.net:8020/app-logs</value>
     </property>
     <property>
       <name>fs.viewfs.mounttable.federationcluster.link./apps</name>
       <value>hdfs://c16f1n03.gpfs.net:8020/apps</value>
     </property>
     <property>
       <name>fs.viewfs.mounttable.federationcluster.link./ats</name>
       <value>hdfs://c16f1n03.gpfs.net:8020/ats</value>
     </property>

     <property>
       <name>fs.viewfs.mounttable.federationcluster.link./atsv2</name>
       <value>hdfs://c16f1n03.gpfs.net:8020/atsv2</value>
     </property>

     <property>
       <name>fs.viewfs.mounttable.federationcluster.link./hdp</name>
       <value>hdfs://c16f1n03.gpfs.net:8020/hdp</value>
     </property>

     <property>
       <name>fs.viewfs.mounttable.federationcluster.link./livy2-recovery</name>
       <value>hdfs://c16f1n03.gpfs.net:8020/livy2-recovery</value>
     </property>

     <property>
       <name>fs.viewfs.mounttable.federationcluster.link./mapred</name>
       <value>hdfs://c16f1n03.gpfs.net:8020/mapred</value>
     </property>

     <property>
       <name>fs.viewfs.mounttable.federationcluster.link./mr-history</name>
       <value>hdfs://c16f1n03.gpfs.net:8020/mr-history</value>
     </property>

     <property>
       <name>fs.viewfs.mounttable.federationcluster.link./services</name>
       <value>hdfs://c16f1n03.gpfs.net:8020/services</value>
     </property>

     <property>
       <name>fs.viewfs.mounttable.federationcluster.link./spark2-history</name>
       <value>hdfs://c16f1n03.gpfs.net:8020/spark2-history</value>
     </property>

     <property>
       <name>fs.viewfs.mounttable.federationcluster.link./tmp</name>
       <value>hdfs://c16f1n03.gpfs.net:8020/tmp</value>
   ```

```
  </property>

  <property>
    <name>fs.viewfs.mounttable.federationcluster.link./user</name>
    <value>hdfs://c16f1n03.gpfs.net:8020/user</value>
  </property>

  <property>
    <name>fs.viewfs.mounttable.federationcluster.link./warehouse</name>
    <value>hdfs://c16f1n03.gpfs.net:8020/warehouse</value>
  </property>

  <property>
    <name>fs.viewfs.mounttable.federationcluster.link./gpfs</name>
    <value>hdfs://c16f1n10.gpfs.net:8020/gpfs</value>
  </property>

</configuration>
```

4. Save the configuration and restart all the required services.

Configuring ViewFs on HDFS cluster with HA

The local native HDFS HA cluster is setup through Ambari. The local HDFS and remote HDFS both have HA setup. Either cluster can have Scale service added. This section describes how to configure ViewFs on the local HDFS cluster with HA setup to access the data on the remote HDFS HA cluster through Ambari.

Suppose there are two clusters, ClusterX and ClusterY. Each cluster is configured with NameNode HA. If applications running on ClusterX needs to access the data from ClusterY, then configure ViewFs on ClusterX. If applications running on ClusterY do not need access to the data on ClusterX, do not configure ViewFs on ClusterY. In the following example, we configure ViewFs on ClusterX and all configurations are done through ClusterX's Ambari GUI.

Setup instructions

1. Go to **Ambari** > **HDFS** > **CONFIGS** > **ADVANCED** > **Advanced core-site** and change the **`fs.defaultFS`** from `hdfs://<fs_name>` to `viewfs://<federation_name>`.

   For example, the property **`fs.defaultFS`** field is changed to `viewfs://federationcluster`.

2. Go to **Ambari** > **HDFS** > **CONFIGS** > **Custom hdfs-site** and add or change all the properties as shown below.

   Note: Configure the key=value pair properties based on your cluster information.

```
Check/modify existing Cluster X information
-----------------------------------------------
dfs.client.failover.proxy.provider.ClusterX=org.apache.hadoop.hdfs.server.namenode.ha.ConfiguredFailoverPr
oxyProvider
dfs.ha.namenodes.ClusterX=nn1,nn2
dfs.internal.nameservices=ClusterX
dfs.namenode.http-address.ClusterX.nn1=c902f14x01.gpfs.net:50070
dfs.namenode.http-address.ClusterX.nn2=c902f14x03.gpfs.net:50070
dfs.namenode.rpc-address.ClusterX.nn1=c902f14x01.gpfs.net:8020
dfs.namenode.rpc-address.ClusterX.nn2=c902f14x03.gpfs.net:8020
dfs.nameservices=ClusterX, ClusterY

ADD new ClusterY key=value pair information
----------------------------------------------
dfs.client.failover.proxy.provider.ClusterY=org.apache.hadoop.hdfs.server.namenode.ha.ConfiguredFailoverPr
oxyProvider
dfs.ha.namenodes.ClusterY=nn1,nn2
dfs.namenode.http-address.ClusterY.nn1=c902f14x04.gpfs.net:50070
dfs.namenode.http-address.ClusterY.nn2=c902f14x06.gpfs.net:50070
dfs.namenode.rpc-address.ClusterY.nn1=c902f14x04.gpfs.net:8020
dfs.namenode.rpc-address.ClusterY.nn2=c902f14x06.gpfs.net:8020
```

3. Go to **Ambari** > **HDFS** > **CONFIGS** > **ADVANCED** > **Advanced viewfs-mount-table** and specify the ViewFs mount table entries. You must define the mount table entries to map to the physical locations to the corresponding mount points from the ViewFs cluster. You must consider factors such as data access, mount levels, and application requirements while defining ViewFs mount table entries.

   Note: Place the `<configuration> </configuration>` entries with the corresponding `<property></property>` values based on your environment into the Advanced views-mount-table field using the xml format.

For example, in ClusterX add the following properties into the viewfs-mount-table tab:

```
<configuration>

  <property>
    <name>fs.viewfs.mounttable.federationcluster.link./app-logs</name>
    <value>hdfs://ClusterX/app-logs</value>
  </property>
  <property>
    <name>fs.viewfs.mounttable.federationcluster.link./apps</name>
    <value>hdfs://ClusterX/apps</value>
  </property>
  <property>
    <name>fs.viewfs.mounttable.federationcluster.link./ats</name>
    <value>hdfs://ClusterX/ats</value>
  </property>

  <property>
    <name>fs.viewfs.mounttable.federationcluster.link./atsv2</name>
    <value>hdfs://ClusterX/atsv2</value>
  </property>

  <property>
    <name>fs.viewfs.mounttable.federationcluster.link./hdp</name>
    <value>hdfs://ClusterX/hdp</value>
  </property>

  <property>
    <name>fs.viewfs.mounttable.federationcluster.link./livy2-recovery</name>
    <value>hdfs://ClusterX/livy2-recovery</value>
  </property>

  <property>
    <name>fs.viewfs.mounttable.federationcluster.link./mapred</name>
    <value>hdfs://ClusterX/mapred</value>
  </property>

  <property>
    <name>fs.viewfs.mounttable.federationcluster.link./mr-history</name>
    <value>hdfs://ClusterX/mr-history</value>
  </property>

  <property>
    <name>fs.viewfs.mounttable.federationcluster.link./services</name>
    <value>hdfs://ClusterX/services</value>
  </property>

  <property>
    <name>fs.viewfs.mounttable.federationcluster.link./spark2-history</name>
    <value>hdfs://ClusterX/spark2-history</value>
  </property>

  <property>
    <name>fs.viewfs.mounttable.federationcluster.link./tmp</name>
    <value>hdfs://ClusterX/tmp</value>
  </property>

  <property>
    <name>fs.viewfs.mounttable.federationcluster.link./user</name>
    <value>hdfs://ClusterX/user</value>
  </property>

  <property>
    <name>fs.viewfs.mounttable.federationcluster.link./warehouse</name>
    <value>hdfs://ClusterX/warehouse</value>
  </property>

  <property>
    <name>fs.viewfs.mounttable.federationcluster.link./gpfs</name>
    <value>hdfs://ClusterY/gpfs</value>
  </property>

</configuration>
```

4. Save the configuration and restart all the required services.

Configure Spark2 for ViewFs support on local Hadoop cluster

The Spark History server provides application history from event logs stored in the file system.

It periodically checks in the background for applications that have completed and renders a UI to show the history of applications by parsing the associated event logs. Therefore, you should configure your installation to enable the Spark History service for monitoring. Otherwise, the Spark2 history will fail to start with the following exception:

```
19/01/07 03:16:04 INFO FsHistoryProvider: History server ui acls disabled; users with admin
permissions:
; groups with admin permissions
Exception in thread "main" java.lang.reflect.InvocationTargetException
        at sun.reflect.NativeConstructorAccessorImpl.newInstance0(Native Method)
        at
sun.reflect.NativeConstructorAccessorImpl.newInstance(NativeConstructorAccessorImpl.java:62)
        at
sun.reflect.DelegatingConstructorAccessorImpl.newInstance(DelegatingConstructorAccessorImpl.ja
va:45)
        at java.lang.reflect.Constructor.newInstance(Constructor.java:423)
        at org.apache.spark.deploy.history.HistoryServer$.main(HistoryServer.scala:280)
        at org.apache.spark.deploy.history.HistoryServer.main(HistoryServer.scala)
Caused by: java.io.IOException: Incomplete HDFS URI, no host: hdfs:///spark2-history
        at
org.apache.hadoop.hdfs.DistributedFileSystem.initialize(DistributedFileSystem.java:171)
        at org.apache.hadoop.fs.FileSystem.createFileSystem(FileSystem.java:3303)
        at org.apache.hadoop.fs.FileSystem.access$200(FileSystem.java:124)
        at org.apache.hadoop.fs.FileSystem$Cache.getInternal(FileSystem.java:3352)
        at org.apache.hadoop.fs.FileSystem$Cache.get(FileSystem.java:3320)
        at org.apache.hadoop.fs.FileSystem.get(FileSystem.java:479)
        at org.apache.hadoop.fs.Path.getFileSystem(Path.java:361)
        at org.apache.spark.deploy.history.FsHistoryProvider.<init>(FsHistoryProvider.scala:115)
        at org.apache.spark.deploy.history.FsHistoryProvider.<init>(FsHistoryProvider.scala:84)
```

Setup instructions

1. Go to **Ambari** > **Services** > **Spark2** > **CONFIGS** > **Advanced** > **Advanced spark2-defaults** and change the two properties to ViewFs from HDFS schema.

   ```
   Spark Eventlog directory = viewfs:///spark2-history/
   Spark History FS Log directory = viewfs:///spark2-history/
   ```

2. Restart Spark2 service for the changes to take effect.

Enable Kerberos

Refer to "Enable Kerberos" on page 143 section to enable Kerberos on the local Native HDFS cluster and the remote IBM Storage Scale HDFS Transparency cluster.

Enable Ranger

Refer to "Enable Ranger" on page 147 section to enable Ranger on the local Native HDFS cluster and the remote IBM Storage Scale HDFS Transparency cluster.

Ranger policy will only be effective on its own HDP cluster, so it is required to enable Ranger individually on each cluster to control the directory and file authorization and authentication.

Hadoop test case scenarios

This section describes the test cases that were run on the local Hadoop cluster with Hadoop Storage Tiering configuration.

Please refer to the "Hadoop test case scenarios" on page 150 section on how to test and leverage Hadoop Storage Tiering with IBM Storage Scale.

Note: If the remote IBM Storage Scale cluster shared path is configured in the ViewFs mount table of the Local Native HDFS, then you do not need to give the full schema path. One just requires specifying the directory path value.

For example, running a MapReduce WordCount job with input from the local native HDFS cluster to generate output to the remote HDFS Transparency cluster will just need to specify the `/path` instead of `hdfs://<namenode_host>:<port>/path` full schema path.

```
sudo -u hdfs yarn jar /usr/hdp/current/hadoop-mapreduce-client/hadoop-
mapreduce-examples.jar wordcount /tmp/redhat-release /gpfs/mapred/wordcount_hdfs

[root@c16f1n03 ~]# sudo -u hdfs hdfs dfs -ls -R
hdfs://c16f1n10.gpfs.net:8020/gpfs/mapred/wordcount_hdfs
-rw-r--r--   3 hdfs root          0 2019-01-07 02:23
hdfs://c16f1n10.gpfs.net:8020/gpfs/mapred/wordcount_hdfs/_SUCCESS
-rw-r--r--   3 hdfs root         68 2019-01-07 02:23
hdfs://c16f1n10.gpfs.net:8020/gpfs/mapred/wordcount_hdfs/part-r-00000
```

Known limitation

In a Kerberos enabled environment, the MapReduce job will fail when trying to create tables on the remote HDFS Transparency cluster when selecting data from the local native HDFS cluster.

Job invoked by the Mapreduce job will fail as follows:

```
hive> CREATE database if not exists gpfsdb LOCATION 'hdfs://c16f1n03.gpfs.net:8020/tmp/hdp-
user1/gpfsdb';
OK
Time taken: 0.099 seconds
hive> describe database gpfsdb;
OK
gpfsdb        hdfs://c16f1n03.gpfs.net:8020/tmp/hdp-user1/gpfsdb    hdp-user1    USER
Time taken: 0.157 seconds, Fetched: 1 row(s)

hive> create table gpfsdb.call_center stored as orc as select * from tpcds_text_5.call_center;
Query ID = hdp-user1_20180319044819_f3d3f976-5d30-4bce-9b7b-bcb6fd5c8e00
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_1520923434038_0020,
Tracking URL = http://c16f1n08.gpfs.net:8088/proxy/application_1520923434038_0020/
Kill Command = /usr/hdp/2.6.4.0-65/hadoop/bin/hadoop job  -kill job_1520923434038_0020
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 0
2018-03-19 04:48:34,360 Stage-1 map = 0%,  reduce = 0%
2018-03-19 04:49:01,733 Stage-1 map = 100%,  reduce = 0%
Ended Job = job_1520923434038_0020 with errors
Error during job, obtaining debugging information...
Examining task ID: task_1520923434038_0020_m_000000 (and more) from job job_1520923434038_0020

Task with the most failures(4):
-----
Task ID:
  task_1520923434038_0020_m_000000

URL:
  http://c16f1n08.gpfs.net:8088/taskdetails.jsp?
jobid=job_1520923434038_0020&tipid=task_1520923434038_0020_m_000000
-----
Diagnostic Messages for this Task:
Error: java.lang.RuntimeException: org.apache.hadoop.hive.ql.metadata.HiveException: Hive
Runtime Error while
processing row
{"cc_call_center_sk":1,"cc_call_center_id":"AAAAAAAABAAAAAAA","cc_rec_start_date":"1998-01-01",
"cc_rec_end_date":"","cc_closed_date_sk":null,"cc_open_date_sk":2450952,"cc_name":"NY Metro",
"cc_class":"large","cc_employees":135,"cc_sq_ft":76815,"cc_hours":"8AM-4PM","cc_manager":"Bob
Belcher",
"cc_mkt_id":6,"cc_mkt_class":"More than other authori","cc_mkt_desc":"Shared others could not
count fully
dollars. New members ca","cc_market_manager":"Julius
Tran","cc_division":3,"cc_division_name":"pri",
"cc_company":6,"cc_company_name":"cally","cc_street_number":"730","cc_street_name":"Ash Hill",
"cc_street_type":"Boulevard","cc_suite_number":"Suite
0","cc_city":"Fairview","cc_county":"Williamson County",
"cc_state":"TN","cc_zip":"35709","cc_country":"United
States","cc_gmt_offset":-5.0,"cc_tax_percentage":0.11}
    at org.apache.hadoop.hive.ql.exec.mr.ExecMapper.map(ExecMapper.java:172)
    at org.apache.hadoop.mapred.MapRunner.run(MapRunner.java:54)
    at org.apache.hadoop.mapred.MapTask.runOldMapper(MapTask.java:453)
    at org.apache.hadoop.mapred.MapTask.run(MapTask.java:343)
    at org.apache.hadoop.mapred.YarnChild$2.run(YarnChild.java:170)
    at java.security.AccessController.doPrivileged(Native Method)
    at javax.security.auth.Subject.doAs(Subject.java:422)
    at org.apache.hadoop.security.UserGroupInformation.doAs(UserGroupInformation.java:1866)
    at org.apache.hadoop.mapred.YarnChild.main(YarnChild.java:164)
Caused by: org.apache.hadoop.hive.ql.metadata.HiveException: Hive Runtime Error while
processing row
{"cc_call_center_sk":1,"cc_call_center_id":"AAAAAAAABAAAAAAA","cc_rec_start_date":"1998-01-01",
"cc_rec_end_date":"","cc_closed_date_sk":null,"cc_open_date_sk":2450952,"cc_name":"NY Metro",
"cc_class":"large","cc_employees":135,"cc_sq_ft":76815,"cc_hours":"8AM-4PM","cc_manager":"Bob
Belcher",
"cc_mkt_id":6,"cc_mkt_class":"More than other authori","cc_mkt_desc":"Shared others could not
count
fully dollars. New members ca","cc_market_manager":"Julius
Tran","cc_division":3,"cc_division_name":
"pri","cc_company":6,"cc_company_name":"cally","cc_street_number":"730","cc_street_name":"Ash
Hill",
"cc_street_type":"Boulevard","cc_suite_number":"Suite 0","cc_city":"Fairview","cc_county":
"Williamson County","cc_state":"TN","cc_zip":"35709","cc_country":"United
States","cc_gmt_offset":
-5.0,"cc_tax_percentage":0.11}
    at org.apache.hadoop.hive.ql.exec.MapOperator.process(MapOperator.java:565)
    at org.apache.hadoop.hive.ql.exec.mr.ExecMapper.map(ExecMapper.java:163)
    ... 8 more
Caused by: org.apache.hadoop.hive.ql.metadata.HiveException:
org.apache.hadoop.hive.ql.metadata.HiveException:
java.io.IOException: Failed on local exception: java.io.IOException:
org.apache.hadoop.security.AccessControlException: Client cannot authenticate via:[TOKEN,
KERBEROS];
Host Details : local host is: "c16f1n07.gpfs.net/192.0.2.0"; destination host is:
"c16f1n03.gpfs.net":8020;
    at
org.apache.hadoop.hive.ql.exec.FileSinkOperator.createBucketFiles(FileSinkOperator.java:582)
    at org.apache.hadoop.hive.ql.exec.FileSinkOperator.process(FileSinkOperator.java:680)
    at org.apache.hadoop.hive.ql.exec.Operator.forward(Operator.java:841)
    at org.apache.hadoop.hive.ql.exec.SelectOperator.process(SelectOperator.java:88)
    at org.apache.hadoop.hive.ql.exec.Operator.forward(Operator.java:841)
    at org.apache.hadoop.hive.ql.exec.TableScanOperator.process(TableScanOperator.java:133)
    at org.apache.hadoop.hive.ql.exec.MapOperator$MapOpCtx.forward(MapOperator.java:170)
    at org.apache.hadoop.hive.ql.exec.MapOperator.process(MapOperator.java:555)
    ... 9 more
Caused by: org.apache.hadoop.security.AccessControlException: Client cannot authenticate via:
[TOKEN, KERBEROS]
    at org.apache.hadoop.security.SaslRpcClient.selectSaslClient(SaslRpcClient.java:172)
    at org.apache.hadoop.security.SaslRpcClient.saslConnect(SaslRpcClient.java:396)
    at org.apache.hadoop.ipc.Client$Connection.setupSaslConnection(Client.java:595)
    at org.apache.hadoop.ipc.Client$Connection.access$2000(Client.java:397)
    at org.apache.hadoop.ipc.Client$Connection$2.run(Client.java:762)
    at org.apache.hadoop.ipc.Client$Connection$2.run(Client.java:758)
    at java.security.AccessController.doPrivileged(Native Method)
    at javax.security.auth.Subject.doAs(Subject.java:422)
    at org.apache.hadoop.security.UserGroupInformation.doAs(UserGroupInformation.java:1866)
    at org.apache.hadoop.ipc.Client$Connection.setupIOstreams(Client.java:758)
    ... 40 more

Container killed by the ApplicationMaster.
Container killed on request. Exit code is 143
Container exited with a non-zero exit code 143.


FAILED: Execution Error, return code 2 from org.apache.hadoop.hive.ql.exec.mr.MapRedTask
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1   HDFS Read: 0 HDFS Write: 0 FAIL
Total MapReduce CPU Time Spent: 0 msec
```

However, it will work if creating a table on the native HDFS by selecting data from the remote HDFS Transparency cluster when Kerberos is enabled.

In this example, the database `local_db_hdfs_ranger` is stored on the local HDFS cluster and the database `remote_db_gpfs_ranger` is stored on the remote HDFS Transparency cluster.

```
hive> create table local_db_hdfs_ranger.localtbl1 as select * from
remote_db_gpfs_ranger.passwd_int_part;
Query ID = hdp-user1_20180319000550_ba08afa2-bbc3-4636-a6dd-c2c9564bfaf3
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1520923434038_0018)

--------------------------------------------------------------------------------
       VERTICES      STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 1 ..........   SUCCEEDED      1          1        0        0       0       0
--------------------------------------------------------------------------------
```

```
VERTICES: 01/01  [==========================>>] 100%  ELAPSED TIME: 4.07 s
--------------------------------------------------------------------------------
Moving data to directory hdfs://c16f1n07.gpfs.net:8020/user/hive/local_db_hdfs_ranger/localtbl1
Table local_db_hdfs_ranger.localtbl1 stats: [numFiles=1, numRows=84, totalSize=3004,
rawDataSize=2920]
OK
Time taken: 6.584 seconds

hive> create table local_db_hdfs_ranger.localtbl2 as select * from
local_db_hdfs_ranger.passwd_int_part;
Query ID = hdp-user1_20180319000658_90631a73-e34e-4919-a30a-05a66769ab41
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1520923434038_0018)

--------------------------------------------------------------------------------
       VERTICES      STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 1 ..........   SUCCEEDED      1          1        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 01/01  [==========================>>] 100%  ELAPSED TIME: 6.16 s
--------------------------------------------------------------------------------
Moving data to directory hdfs://c16f1n07.gpfs.net:8020/user/hive/local_db_hdfs_ranger/localtbl2
Table local_db_hdfs_ranger.localtbl2 stats: [numFiles=1, numRows=84, totalSize=3004,
rawDataSize=2920]
OK
Time taken: 7.904 seconds
```

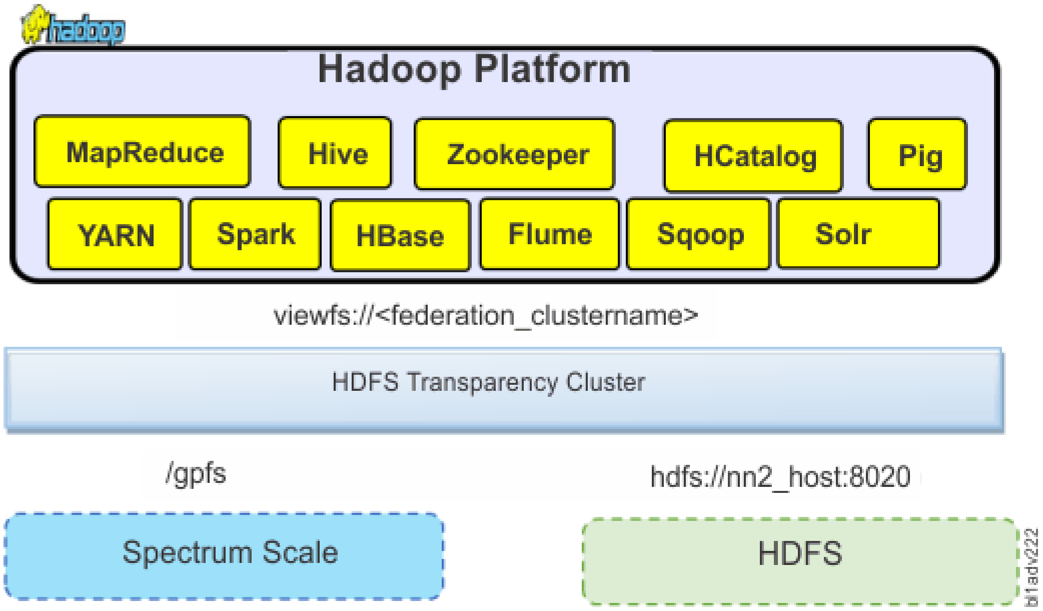
Apache Hadoop ViewFs support

NameNode federation feature in Hadoop 3.x enables adding multiple native HDFS namespaces under single Hadoop cluster. This feature was added to improve the HDFS NameNode horizontal scaling. Instances where IBM Storage Scale file system is used with HDFS transparency can also use ViewFS to have more than one IBM Storage Scale file system along with native HDFS file system under a single Hadoop cluster. The Hadoop applications can get input data from the native HDFS, analyze the input, and write the output to the IBM Storage Scale file system. This feature is available from HDFS transparency version 2.7.0-2 (gpfs.hdfs-protocol-2.7.0-2) and later.

Note: If you want your applications running in clusterA to process the data in clusterB, only update the configuration for federation in clusterA. This is call federating clusterB with clusterA. If you want your applications running in clusterB to process data from clusterA, you need to update the configuration for federation in clusterB. This is called federating clusterA with clusterB.

Single viewfs namespace between IBM Storage Scale and Native HDFS

Before configuring viewfs support, ensure that you have configured the HDFS Transparency cluster (see “Hadoop cluster planning” on page 11, “Installing” on page 29, and “Configuring” on page 52).



See the following sections to configure viewfs for IBM Storage Scale and Native HDFS.

Single viewfs namespace between IBM Storage Scale and native HDFS –Part I
This topic describes the steps to get a single namespace by federating HDFS transparency namespace into native HDFS namespace. If you take HortonWorks HDP, you could change the configurations from the **Ambari GUI** > **HDFS** > **Configs**.

In this mode, `nn1_host` is HDFS Transparency NameNode. You have another native HDFS cluster. After you federate native HDFS into HDFS Transparency, your applications can access the data from both HDFS Transparency and native HDFS with the schema `fs.defaultFS` defined in the HDFS Transparency cluster. All configuration changes are done on the HDFS Transparency side and your Hadoop client nodes. This mode does not need configurations change on native HDFS cluster.

1. Shut down the HDFS Transparency cluster daemon by running the following command from one of the HDFS transparency nodes in the cluster:

   ```
   # mmhadoopctl connector stop
   ```

2. On the *nn1_host*, add the following configuration settings in `/usr/lpp/mmfs/hadoop/etc/hadoop/core-site.xml` (for HDFS Transparency 2.7.3-x) or `/var/mmfs/hadoop/etc/hadoop/core-site.xml` (for HDFS Transparency 3.0.x).:

   ```
   <configuration>
   <property>
       <name>fs.defaultFS</name>
       <value>viewfs://<viewfs_clustername></value>
       <description>The name of the namespace</description>
   </property>

   <property>
       <name>fs.viewfs.mounttable.<viewfs_clustername>.link./<viewfs_dir1></name>
       <value>hdfs://nn1_host:8020/<mount_dir></value>
       <description>The name of the Spectrum Scale file system</description>
   </property>

   <property>
       <name>fs.viewfs.mounttable.<federation_clustername>.link./<viewfs_dir2></name>
       <value>hdfs://nn2_host:8020/<mount_dir></value>
       <description>The name of the hdfs file system</description>
   </property>
   </configuration>
   ```

Note: Change <viewfs_clustername> and <mount_dir> according to your cluster configuration. In this example, the *nn1_host* refers to the HDFS transparency NameNode and the *nn2_host* refers to the native HDFS NameNode.

Once the federation configuration changes are in effect on the node, the node will only see the directories that are specified in the core-site.xml file. For the above configurations, you can only see the two directories /<viewfs_dir1> and /<viewfs_dir2>.

3. On *nn1_host*, add the following configuration settings in /var/mmfs/hadoop/etc/hadoop/hdfs-site.xml (for HDFS Transparency 3.0.x).

```
<configuration>
<property>
    <name>dfs.nameservices</name>
    <value>nn1,nn2</value>
</property>

<property>
    <name>dfs.namenode.rpc-address.nn1</name>
    <value>nn1-host:8020</value>
</property>

<property>
    <name>dfs.namenode.rpc-address.nn2</name>
    <value>nn2-host:8020</value>
</property>

<property>
    <name> dfs.namenode.http-address.nn1</name>
    <value>nn1-host:50070</value>
</property>

<property>
    <name>dfs.namenode.http-address.nn2</name>
    <value>nn2-host:50070</value>
</property>
</configuration>
```

4. On *nn1_host*, synchronize the configuration changes with the other HDFS transparency nodes by running the following command:

For HDFS Transparency 2.7.3-x:

```
# mmhadoopctl connector syncconf /usr/lpp/mmfs/hadoop/etc/hadoop/
```

For HDFS Transparency 3.0.x:

```
# mmhadoopctl connector syncconf /var/mmfs/hadoop/etc/hadoop/
```

Note: The following output messages from the above command for the native HDFS NameNode, *nn2-host*, can be seen:

```
scp: /usr/lpp/mmfs/hadoop/etc/hadoop//: No such file or directory
scp: /usr/lpp/mmfs/hadoop/etc/hadoop//: No such file or directory
scp: /usr/lpp/mmfs/hadoop/etc/hadoop//: No such file or directory
scp: /usr/lpp/mmfs/hadoop/etc/hadoop//: No such file or directory
scp: /usr/lpp/mmfs/hadoop/etc/hadoop//: No such file or directory
scp: /usr/lpp/mmfs/hadoop/etc/hadoop//: No such file or directory
scp: /usr/lpp/mmfs/hadoop/etc/hadoop//: No such file or directory
scp: /usr/lpp/mmfs/hadoop/etc/hadoop//: No such file or directory
```

The output messages above are seen because during the synchronization of the configuration to all the nodes in the cluster, the /usr/lpp/mmfs/Hadoop/etc/hadoop directory does not exist in the nn2-host native HDFS NameNode. This is because the HDFS Transparency is not installed on the native HDFS NameNode. Therefore, these messages for the native HDFS NameNode can be ignored.

Another way to synchronize the configuration files is by using the **scp** command to copy the following files under /usr/lpp/mmfs/hadoop/etc/hadoop/ into all the other nodes in HDFS Transparency cluster: workers, log4j.properties, hdfs-site.xml, hadoop-policy.xml, hadoop-metrics.properties, hadoop-metrics2.properties, core-site.xml, and gpfs-site.xml.

For HDFS Transparency 3.0.x:

```
#mmhadoopctl connector syncconf /var/mmfs/hadoop/etc/hadoop
```

5. On *nn1_host*, start all the HDFS transparency cluster nodes by running the following command:

   `# mmhadoopctl connector start`

   Note: The following warning output messages from the above command for the native HDFS NameNode, nn2-host can be seen:

   ```
   nn2-host: bash: line 0: cd: /usr/lpp/mmfs/hadoop: No such file or directory
   nn2-host: bash: /usr/lpp/mmfs/hadoop/sbin/hadoop-daemon.sh: No such file or directory
   ```

   These messages are displayed because HDFS Transparency is not installed on the native HDFS NameNode. Therefore, these messages can be ignored.

   To avoid the above messages, run the following commands:

   a. On nn1-host, run the following command as root to start the HDFS Transparency NameNode:

   ```
   # cd /usr/lpp/mmfs/hadoop; /usr/lpp/mmfs/hadoop/sbin/hadoop-daemon.sh
   --config /usr/lpp/mmfs/hadoop/etc/hadoop
   --script /usr/lpp/mmfs/hadoop/sbin/gpfs start namenode
   ```

   b. On nn1-host, run the following command as root to start the HDFS Transparency DataNode:

   ```
   # cd /usr/lpp/mmfs/hadoop; /usr/lpp/mmfs/hadoop/sbin/hadoop-daemons.sh
   --config /usr/lpp/mmfs/hadoop/etc/hadoop
   --script /usr/lpp/mmfs/hadoop/sbin/gpfs start datanode
   ```

   Note: If you deployed IBM BigInsights IOP, the IBM Storage Scale Ambari integration module (gpfs.hdfs-transparency.ambari-iop_4.1-0) does not support viewfs configuration in Ambari. Therefore, starting the HDFS Transparency service or other services will regenerate the `core-site.xml` and `hdfs-site.xml` from the Ambari database and will overwrite the changes that were done from Step 1 to Step 4. HDFS Transparency and all other services will have to be started in the command mode.

6. Update the configuration changes in Step 2 and Step 3 in your Hadoop client configurations so that the Hadoop applications can view all the directories in viewfs.

   Note: If you deployed IBM BigInsights IOP, update the `core-site.xml` and the `hdfs-site.xml` in Step 2 and Step 3 accordingly from the `/etc/hadoop/conf` directory on each of the node so that the Hadoop applications can see the directories in viewfs.

   If you deployed Open Source Apache Hadoop, then update the `core-site.xml` and the `hdfs-site.xml` according to the Apache Hadoop location configured in your site.

7. From one of the Hadoop clients, verify that the viewfs directories are available by running the following command:

   ```
   hadoop dfs -ls /
   ```

Single viewfs namespace between IBM Storage Scale and native HDFS –Part II

This topic describes the steps to get a single namespace by joining HDFS transparency namespace with native HDFS namespace.

1. Stop the hadoop applications and the native HDFS services on the native HDFS cluster.

   The detailed command is dependent on the Hadoop distro. For example, for IBM BigInsights IOP, stop all services from the Ambari GUI.

2. Perform Step 2 and Step 3 in the section "Single viewfs namespace between IBM Storage Scale and native HDFS –Part I" on page 179 on the node *nn2-host* with the correct path for `core-site.xml` and the `hdfs-site.xml` according to the Hadoop distribution.

   If running with the open source Apache Hadoop, the location of the `core-site.xml` and the `hdfs-site.xml` is in `$YOUR_HADOOP_PREFIX/etc/hadoop/`. The `$YOUR_HADOOP_PREFIX` is the location of the Hadoop package. If running with IBM BigInsights IOP, then Ambari currently does not support viewfs configuration. You will have to manually update the configurations under `/etc/hadoop/conf/`.

Note: If you want to see all the directories from the native HDFS shown up in viewfs, define all the native HDFS directories in the core-site.xml.

If you have a secondary NameNode configured in native HDFS, update the following configuration in the `hdfs-site.xml`:

```
<property>
 <name>dfs.namenode.secondary.http-address.nn2-host</name>
 <value>secondaryNameNode-host:50090</value>
</property>

<property>
 <name>dfs.secondary.namenode.keytab.file.nn2-host</name>
  <value>/etc/security/keytabs/nn.service.keytab</value>
</property>
```

Note: If you have deployed IBM BigInsights IOP, it will generate the key `dfs.namenode.secondary.http-address` and `dfs.secondary.namenode.keytab.file` by default. For viewfs, modify the `hdfs-site.xml` file with the correct values according to your environment.

3. Synchronize the updated configurations from the *nn2-host* node to all the other native HDFS nodes and start the native HDFS services.

   If running with open source Apache Hadoop, you need to use the **scp** command to synchronize the `core-site.xml` and the `hdfs-site.xml` from the host *nn2-host* to all the other native HDFS nodes. Start the native HDFS service by running the following command:

   ```
   $YOUR_HOME_PREFIX/sbin/start-dfs.sh
   ```

   If IBM BigInsights IOP is running, synchronize the updated configurations manually to avoid the updated viewfs configurations overwritten by Ambari.

   Note: Check the configurations under `/etc/hadoop/conf` to ensure that all the changes have been synchronized to all the nodes.

4. Start the native HDFS service.

   If you are running open source Hadoop, start the native HDFS service on the command line:

   ```
   $YOUR_HADOOP_PREFIX/bin/start-dfs.sh
   ```

   If you deployed IBM BigInsights IOP, Ambari does not support viewfs configuration. Therefore, you must start the native HDFS services manually.

   a. Start native HDFS NameNode.

      Log in to nn2-host as root, run **`su - hdfs`** to switch to the hdfs UID and then run the following command:

      ```
      /usr/iop/current/hadoop-client/sbin/hadoop-daemon.sh --config
      /usr/iop/current/hadoop-client/conf start namenode
      ```

   b. Start the native HDFS DataNode.

      Log in to the DataNode, run **`su - hdfs`** to switch to the hdfs UID and then run the following command:

      ```
      /usr/iop/current/hadoop-client/sbin/hadoop-daemon.sh --config
      /usr/iop/current/hadoop-client/conf start datanode
      ```

      Note: Run the above command on each DataNode.

Log in to the Secondary NameNode, run **su - hdfs** to switch to the hdfs UID and run the following command to start Secondary NameNode:

```
/usr/iop/current/hadoop-client/sbin/hadoop-daemon.sh --config
/usr/iop/current/hadoop-client/conf start secondarynamenode
```

5. Update the `core-site.xml` and `hdfs-site.xml` used by the Hadoop clients on which the Hadoop applications will run over viewfs.

   If the open source Apache Hadoop is running, the location of `core-site.xml` and `hdfs-site.xml` is in $YOUR_HADOOP_PREFIX/etc/hadoop/. The $YOUR_HADOOP_PREFIX is the location of the Hadoop package. If another Hadoop distro is running, see "Known limitations" on page 186.

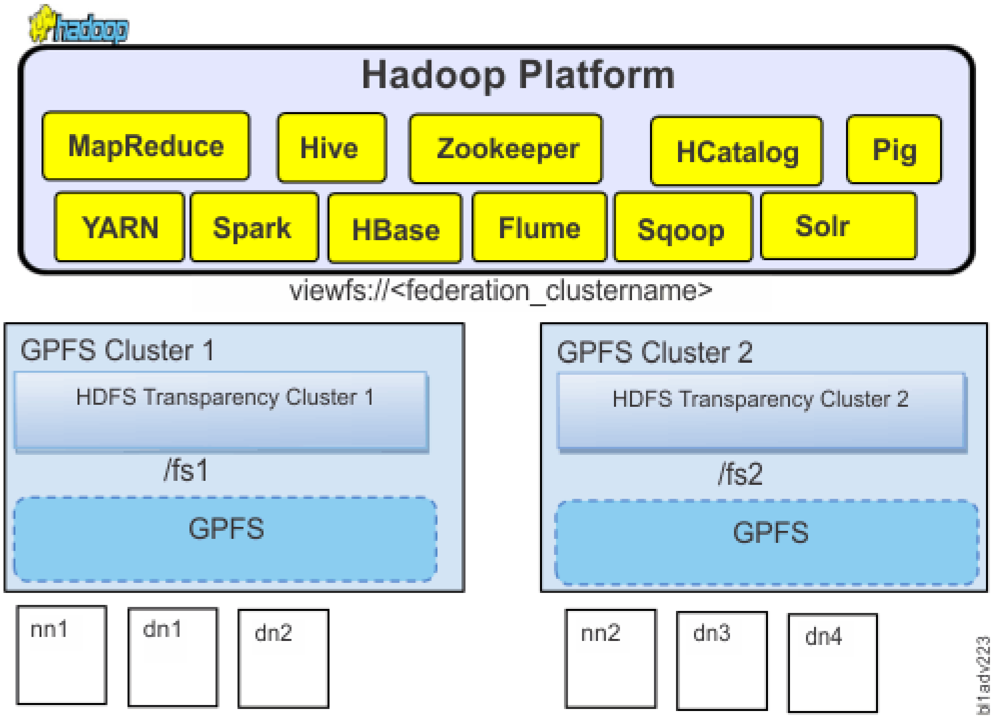
   If IBM BigInsights IOP is running, `core-site.xml` and `hdfs-site.xml` are located at `/etc/hadoop/conf/`.
6. To ensure that the viewfs file system is functioning correctly, run the following command:

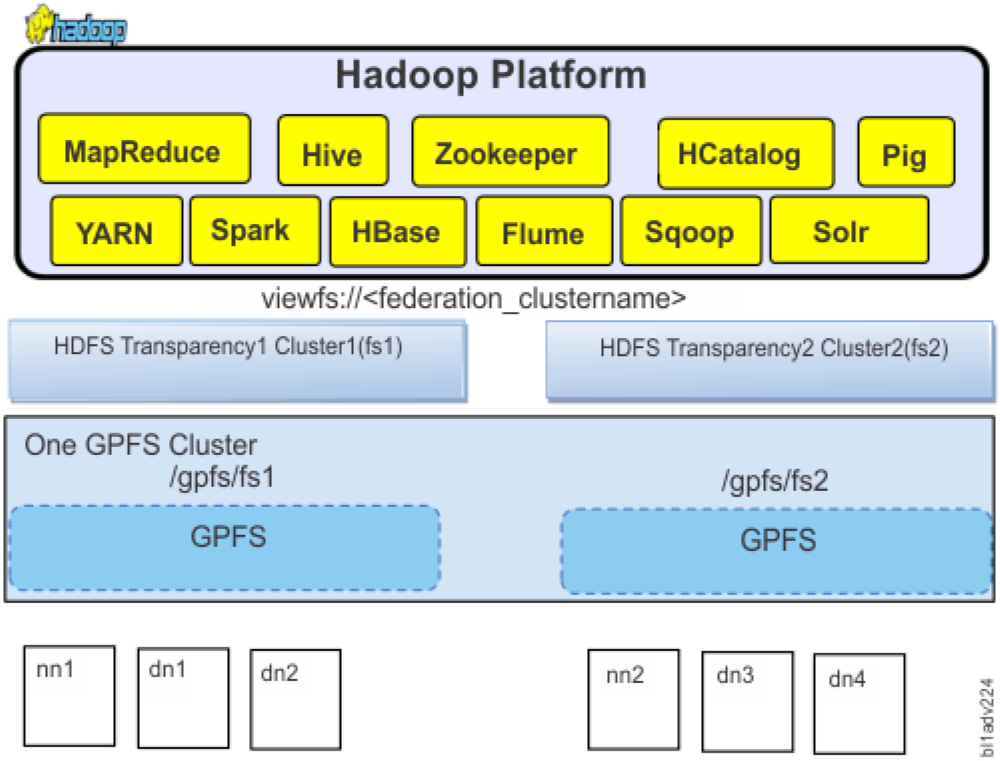
```
hadoop fs -ls /
```

Single viewfs namespace between two IBM Storage Scale file systems

You can get a single viewfs namespace joining two IBM Storage Scale file systems from different clusters or from the same cluster.

Irrespective of the mode that you select, configure one HDFS transparency cluster for each IBM Storage Scale file system (see "Hadoop cluster planning" on page 11, "Installing" on page 29, and "Configuring" on page 52), and then join the two HDFS transparency clusters together.





To join two file systems from the same cluster, select nodes that can provide HDFS transparency services for the first file system and the second file system separately.

Configuration
This topic describes the steps to configure viewfs between two IBM Storage Scale file systems.

Before configuring the viewfs, see "Hadoop cluster planning" on page 11, "Installing" on page 29, and "Configuring" on page 52 to configure HDFS transparency cluster 1 and HDFS transparency cluster 2 for each file system.

1. To stop the HDFS transparency services, run the **mmhadoopctl connector stop** on both HDFS transparency clusters.
2. On the *nn1* host, add the following configuration settings in `/usr/lpp/mmfs/hadoop/etc/hadoop/core-site.xml` (for HDFS Transparency 2.7.3-x) or `/var/mmfs/hadoop/etc/hadoop/core-site.xml` (for HDFS Transparency 3.0.x):

```
<configuration>
<property>
    <name>fs.defaultFS</name>
    <value>viewfs://<viewfs_clustername></value>
    <description>The name of the viewfs file system</description>
</property>

<property>
    <name>fs.viewfs.mounttable.<viewfs_clustername>.link./<mount_dir></name>
    <value>hdfs://nn1_host:8020/<mount_dir></value>
    <description>The name of the gpfs file system</description>
</property>


<property>
    <name>fs.viewfs.mounttable.<viewfs_clustername>.link./<mount_dir></name>
    <value>hdfs://nn2_host:8020/<mount_dir></value>
    <description>The name of the hdfs file system</description>
</property>
</configuration>
```

Note: Change `<viewfs_clustername>` and `<mount_dir>` according to your cluster. Change *nn1_host* and *nn2_host* accordingly.

3. On *nn1_host*, add the following configuration settings in `hdfs-site.xml`.

   ```
   <configuration>
   <property>
       <name>dfs.nameservices</name>
       <value>nn1,nn2</value>
   </property>

   <property>
       <name>dfs.namenode.rpc-address.nn1</name>
       <value>nn1:8020</value>
   </property>

   <property>
       <name>dfs.namenode.rpc-address.nn2</name>
       <value>nn2:8020</value>
   </property>

   <property>
       <name> dfs.namenode.http-address.nn1</name>
       <value>nn1:50070</value>
   </property>

   <property>
       <name>dfs.namenode.http-address.nn2</name>
       <value>nn2:50070</value>
   </property>
   </configuration>
   ```

4. On *nn1_host*, take `mmhadoopctl connector syncconf /usr/lpp/mmfs/hadoop/etc/hadoop` (for HDFS Transparency 2.7.3-x) or `mmhadoopctl connector syncconf /var/mmfs/hadoop/etc/hadoop` (for HDFS Transparency 3.0.x) to synchronize the configurations from `nn1_host` to all other HDFS Transparency nodes.

   Note: The above must be done for each node in the HDFS Transparency Cluster 1. For example, change the hostX accordingly and run it for each node in the HDFS Transparency Cluster1.

5. On *nn2_host*, perform Step 1 through Step 4.

   Note: If you only want to federate HDFS Transparency Cluster2 into HDFS Transparency Cluster1, Step 5 is not needed.

6. On *nn1_host*, start the HDFS transparency cluster:

   a. On nn1, run the following command as root to start HDFS Transparency Cluster1 NameNode:

   ```
   #cd /usr/lpp/mmfs/hadoop; /usr/lpp/mmfs/hadoop/sbin/hadoop-daemon.sh
   --config /usr/lpp/mmfs/hadoop/etc/hadoop --script /usr/lpp/mmfs/hadoop/sbin/gpfs start
   namenode
   ```

   b. On nn1, run the following command as root to start HDFS Transparency Cluster1 DataNode:

   ```
   cd /usr/lpp/mmfs/hadoop; /usr/lpp/mmfs/hadoop/sbin/hadoop-daemons.sh
   --config /usr/lpp/mmfs/hadoop/etc/hadoop --script /usr/lpp/mmfs/hadoop/sbin/gpfs start
   datanode
   ```

   Note: If you deployed IBM BigInsights IOP, IBM Storage Scale Ambari integration package `gpfs.hdfs-transparency.ambari-iop_4.1-0` does not support federation configuration on Ambari. Therefore, starting the HDFS Transparency service will re-generate the `core-site.xml` and `hdfs-site.xml` from the Ambari database and overwrite the changes you made from Step1 to Step4. Repeat Step 6.1 and Step 6.2 to start HDFS Transparency in the command mode.

7. On *nn2*, start the other HDFS transparency cluster:

   a. On nn2, run the following command as root to start the HDFS Transparency Cluster2 NameNode:

   ```
   #cd /usr/lpp/mmfs/hadoop; /usr/lpp/mmfs/hadoop/sbin/hadoop-daemon.sh
   --config /usr/lpp/mmfs/hadoop/etc/hadoop --script /usr/lpp/mmfs/hadoop/sbin/gpfs start
   namenode
   ```

   b. On nn2, run the following command as root to start the HDFS Transparency Cluster2 DataNode:

```
cd /usr/lpp/mmfs/hadoop; /usr/lpp/mmfs/hadoop/sbin/hadoop-daemons.sh
--config /usr/lpp/mmfs/hadoop/etc/hadoop --script /usr/lpp/mmfs/hadoop/sbin/gpfs start
datanode
```

Note: If you deployed IBM BigInsights IOP, IBM Storage Scale Ambari integration package `gpfs.hdfs-transparency.ambari-iop_4.1-0` does not support viewfs configuration on Ambari. Therefore, starting the HDFS Transparency service will re-generate the `core-site.xml` and `hdfs-site.xml` from the Ambari database and overwrite the changes you made from Step1 to Step4. Repeat the Step 7.1 and Step 7.2 to start HDFS Transparency in the command mode.

8. Update `core-site.xml` and `hdfs-site.xml` for the Hadoop clients on which the Hadoop applications run over viewfs.

   If you take open source Apache Hadoop, the location of `core-site.xml` and `hdfs-site.xml` is `$YOUR_HADOOP_PREFIX/etc/hadoop/`. The `$YOUR_HADOOP_PREFIX` is the location of the Hadoop package. If you take another Hadoop distro, see "Known limitations" on page 186.
9. Restart the Hadoop applications on both clusters.

   Note: You should always keep the native HDFS service non-functional if you select HDFS Transparency.
10. To ensure that the viewfs is functioning correctly, run the **`hadoop fs -ls /`** command.

Known limitations

This topic lists the known limitations for viewfs support.

- All the changes at `/usr/lpp/mmfs/hadoop/etc/hadoop/core-site.xml` and `/usr/lpp/mmfs/hadoop/etc/hadoop/hdfs-site.xml` must be updated in the configuration file that is used by the Hadoop distributions. However, Hadoop distributions occasionally manage their configuration, and the management interface might not support the key used for viewfs. For example, IBM BigInsights IOP takes Ambari and Ambari GUI does not support some property names.

  Similarly, HortonWorks HDP 2.6.x does not support the key used for viewfs.
- The native HDFS and HDFS transparency cannot be run over the same node because of the network port number conflict.
- If you select to join both the native HDFS and HDFS transparency, configure the native HDFS cluster and make the native HDFS service function. Configure the viewfs for native HDFS and HDFS transparency.

  For a new native HDFS cluster, while starting the service for the first time, DataNode registers itself with the NameNode. The HDFS Transparency NameNode does not accept any registration from the native HDFS DataNode. Therefore, an exception occurs if you configure a new native HDFS cluster, federate it with HDFS transparency, and then try to make both clusters (one native HDFS cluster and another HDFS Transparency cluster) function at the same time.
- Start and stop the native HDFS cluster or the HDFS Transparency cluster separately if you want to maintain them.

Problem determination

1. ERROR: `Requested user hdfs is banned` while running MapReduce jobs as user *hdfs* in native HDFS cluster.

   Solution:

   For solution, see https://my.cloudera.com/knowledge/LinuxTaskController-job-fails-with-error-Requested-user-hdfs?id=275909.
2. `IOException: javax.security.sasl.SaslException`: GSS initiate failed [Caused by GSSException: No valid credentials provided (Mechanism level: Failed to find any Kerberos tgt)] when running any **`hadoop fs`** command as a specified user.

   Solution:

   You must change to the appropriate principal and keytab for the specified user.

```
kinit -k -t /usr/lpp/mmfs/hadoop/tc/hadoop/keytab/hdptestuser.headless.keytab hdp-
user1@IBM.COM
```

3. hive> CREATE database remote_db2 COMMENT 'Holds all the tables data in remote HDFS Transparency cluster' LOCATION `hdfs://c16f1n13.gpfs.net:8020/user/hive/remote_db2`;

   FAILED: Execution Error, return code 1 from `org.apache.hadoop.hive.ql.exec.DDLTask`. MetaException (message:org.apache.hadoop.ipc.RemoteException(org.apache.hadoop.security.authorize.AuthorizationException): Unauthorized connection for super-user: hive/c16f1n08.gpfs.net@IBM.COM from IP 192.0.2.1)

   Solution:

   Change the below custom core-site properties on all the nodes of the remote HDFS Transparency cluster:

   `hadoop.proxyuser.hive.hosts`=*

   `hadoop.proxyuser.hive.groups`=*

4. Hive Import and Export cases are not supported in ViewFS schema. The following exception will be thrown:

   ```
   0: jdbc:hive2://c16f1n03.gpfs.net:2181,c16f1n> EXPORT TABLE local_db_hdfs.passwd_int_part
   TO 'viewfs://federationcluster/gpfs/hive/remote_db_gpfs/passwd_int_part_export';
   Error: Error while compiling statement: FAILED: SemanticException Invalid path only the
   following file systems accepted for export/import : hdfs,pfile,file,s3,s3a,gs
   (state=42000,code=40000)
   ```

   Solution:

   Change the schema from `ViewFS://xx` to `hdfs://xx`.

   ```
   0: jdbc:hive2://c16f1n03.gpfs.net:2181,c16f1n> EXPORT TABLE local_db_hdfs.passwd_int_part
   TO 'hdfs://c16f1n10:8020/gpfs/hive/remote_db_gpfs/passwd_int_part_export';
   INFO  : Compiling command(queryId=hive_20190110021038_7f5d37d6-f6e6-488a-b7ee-99261fc946e3):
   EXPORT
   TABLE local_db_hdfs.passwd_int_part TO 'hdfs://c16f1n10:8020/gpfs/hive/remote_db_gpfs/
   passwd_int_part_export'
   INFO  : Semantic Analysis Completed (retrial = false)
   INFO  : Returning Hive schema: Schema(fieldSchemas:null, properties:null)
   INFO  : Completed compiling command(queryId=hive_20190110021038_7f5d37d6-f6e6-488a-
   b7ee-99261fc946e3);
   Time taken: 0.125 seconds
   INFO  : Executing command(queryId=hive_20190110021038_7f5d37d6-f6e6-488a-b7ee-99261fc946e3):
   EXPORT
   TABLE local_db_hdfs.passwd_int_part TO 'hdfs://c16f1n10:8020/gpfs/hive/remote_db_gpfs/
   passwd_int_part_export'
   INFO  : Starting task [Stage-0:REPL_DUMP] in serial mode
   INFO  : Completed executing command(queryId=hive_20190110021038_7f5d37d6-f6e6-488a-
   b7ee-99261fc946e3);
   Time taken: 0.19 seconds
   INFO  : OK
   No rows affected (0.343 seconds)
   ```

5. Hive LOAD DATA INPATH cases failed in ViewFS schema with the following exception:

   ```
   0: jdbc:hive2://c16f1n03.gpfs.net:2181,c16f1n> LOAD DATA INPATH '/tmp/2012.txt'
   INTO TABLE db_bdpbase.Employee PARTITION(year=2012);
   INFO  : Compiling command(queryId=hive_20190110024717_b6a0b5a0-d8a1-42e3-a6bb-40ca297d97dd):
   LOAD DATA INPATH '/tmp/2012.txt' INTO TABLE db_bdpbase.Employee PARTITION(year=2012)
   INFO  : Semantic Analysis Completed (retrial = false)
   INFO  : Returning Hive schema: Schema(fieldSchemas:null, properties:null)
   INFO  : Completed compiling command(queryId=hive_20190110024717_b6a0b5a0-d8a1-42e3-a6bb-40ca297d97dd);
   Time taken: 0.168 seconds
   INFO  : Executing command(queryId=hive_20190110024717_b6a0b5a0-d8a1-42e3-a6bb-40ca297d97dd):
   LOAD DATA INPATH '/tmp/2012.txt' INTO TABLE db_bdpbase.Employee PARTITION(year=2012)
   INFO  : Starting task [Stage-0:MOVE] in serial mode
   INFO  : Loading data to table db_bdpbase.employee partition (year=2012) from
   viewfs://federationcluster/tmp/2012.txt
   ERROR : FAILED: Execution Error, return code 1 from org.apache.hadoop.hive.ql.exec.MoveTask.
   org.apache.hadoop.hive.ql.metadata.HiveException: Unable to move source
   viewfs://federationcluster/tmp/2012.txt to destination
   viewfs://federationcluster/warehouse/tablespace/managed/hive/db_bdpbase.db/employee/year=2012/delta_0000011_0000011_0000
   INFO  : Completed executing command(queryId=hive_20190110024717_b6a0b5a0-d8a1-42e3-a6bb-40ca297d97dd); Time taken: 0.117
   seconds
   Error: Error while processing statement: FAILED: Execution Error,
   return code 1 from org.apache.hadoop.hive.ql.exec.MoveTask. org.apache.hadoop.hive.ql.metadata.HiveException:
   Unable to move source viewfs://federationcluster/tmp/2012.txt to destination
   ```

```
viewfs://federationcluster/warehouse/tablespace/managed/hive/db_bdpbase.db/employee/year=2012/delta_0000011_0000011_0000
(state=08S01,code=1)
```

Solution:

Change the load path to use the same mount point directory. Here the table Employee is located at `/warehouse/tablespace/managed/hive/db_bdpbase/Employee`, so the data inpath is located under the ViewFS same mount point `/warehouse`.

```
0: jdbc:hive2://c16f1n03.gpfs.net:2181,c16f1n> LOAD DATA INPATH
'/warehouse/2012.txt' INTO TABLE db_bdpbase.Employee PARTITION(year=2012);
INFO  : Compiling command(queryId=hive_20190110024734_b5c59f01-9f08-4e47-8884-bb7dd8e131ca):
LOAD DATA INPATH '/warehouse/2012.txt' INTO TABLE db_bdpbase.Employee PARTITION(year=2012)
INFO  : Semantic Analysis Completed (retrial = false)
INFO  : Returning Hive schema: Schema(fieldSchemas:null, properties:null)
INFO  : Completed compiling command(queryId=hive_20190110024734_b5c59f01-9f08-4e47-8884-
bb7dd8e131ca);
Time taken: 0.118 seconds
INFO  : Executing command(queryId=hive_20190110024734_b5c59f01-9f08-4e47-8884-bb7dd8e131ca):
LOAD DATA INPATH '/warehouse/2012.txt' INTO TABLE db_bdpbase.Employee PARTITION(year=2012)
INFO  : Starting task [Stage-0:MOVE] in serial mode
INFO  : Loading data to table db_bdpbase.employee partition (year=2012) from
viewfs://federationcluster/warehouse/2012.txt
INFO  : Starting task [Stage-1:STATS] in serial mode
INFO  : Completed executing command(queryId=hive_20190110024734_b5c59f01-9f08-4e47-8884-
bb7dd8e131ca);
Time taken: 0.358 seconds
INFO  : OK
No rows affected (0.501 seconds)
```

6. See Permission denied: Principal [name=hive, type=USER] does not have following privileges for operation DFS [ADMIN] in Hive Beeline console when run HiveQL.

```
0: jdbc:hive2://c16f1n03.gpfs.net:2181,c16f1n> load data local inpath
'/tmp/hive/kv2.txt' into table local_db_hdfs.passwd_ext_nonpart;
Error: Error while compiling statement: FAILED: HiveAccessControlException
Permission denied: Principal [name=hive, type=USER] does not have following
privileges for operation LOAD [ADMIN] (state=42000,code=40000)
0: jdbc:hive2://c16f1n03.gpfs.net:2181,c16f1n> dfs -ls /gpfs;
Error: Error while processing statement: Permission denied: Principal [name=hive, type=USER]
does not have following privileges for operation DFS [ADMIN] (state=,code=1)
0: jdbc:hive2://c16f1n03.gpfs.net:2181,c16f1n>
```

Solution:

Go to **Ambari** > **Hive** > **CONFIGS** > **ADVANCED** > **Custom hive-site** and add **`hive.users.in.admin.role`** to the list of comma-separated users who require admin role authorization (such as the user hive). Restart the Hive services for the changes to take effect.

The permission denied error is fixed after adding **`hive.users.in.admin.role`**`=hive`.

```
0: jdbc:hive2://c16f1n03.gpfs.net:2181,c16f1n> dfs -ls /gpfs;
+----------------------------------------------------+
|                      DFS Output                    |
+----------------------------------------------------+
| drwxr-xr-x   - nobody root          0 2019-01-08 02:32 /gpfs/passwd_sparkshell |
| -rw-r--r--   2 hdfs   root         52 2019-01-08 02:37 /gpfs/redhat-release |
+----------------------------------------------------+
25 rows selected (0.123 seconds)
0: jdbc:hive2://c16f1n03.gpfs.net:2181,c16f1n> load data local inpath
'/tmp/hive/kv2.txt' into table local_db_hdfs.passwd_ext_nonpart;
INFO  : Compiling command(queryId=hive_20190111020239_bb71b8c0-1b00-4f96-bec2-e0e899de62df):
load data local inpath '/tmp/hive/kv2.txt' into table local_db_hdfs.passwd_ext_nonpart
INFO  : Semantic Analysis Completed (retrial = false)
INFO  : Returning Hive schema: Schema(fieldSchemas:null, properties:null)
INFO  : Completed compiling command(queryId=hive_20190111020239_bb71b8c0-1b00-4f96-bec2-
e0e899de62df);
Time taken: 0.285 seconds
INFO  : Executing command(queryId=hive_20190111020239_bb71b8c0-1b00-4f96-bec2-e0e899de62df):
load data local inpath '/tmp/hive/kv2.txt' into table local_db_hdfs.passwd_ext_nonpart
INFO  : Starting task [Stage-0:MOVE] in serial mode
INFO  : Loading data to table local_db_hdfs.passwd_ext_nonpart from file:/tmp/hive/kv2.txt
INFO  : Starting task [Stage-1:STATS] in serial mode
INFO  : Completed executing command(queryId=hive_20190111020239_bb71b8c0-1b00-4f96-bec2-
e0e899de62df);
Time taken: 0.545 seconds
```

```
INFO  : OK
No rows affected (0.947 seconds)
```

7. The OOZIE Service check failed with error: Error: E0904: Scheme [viewfs] not supported in uri [viewfs://hdpcluster/user/ambari-qa/examples/apps/no-op]

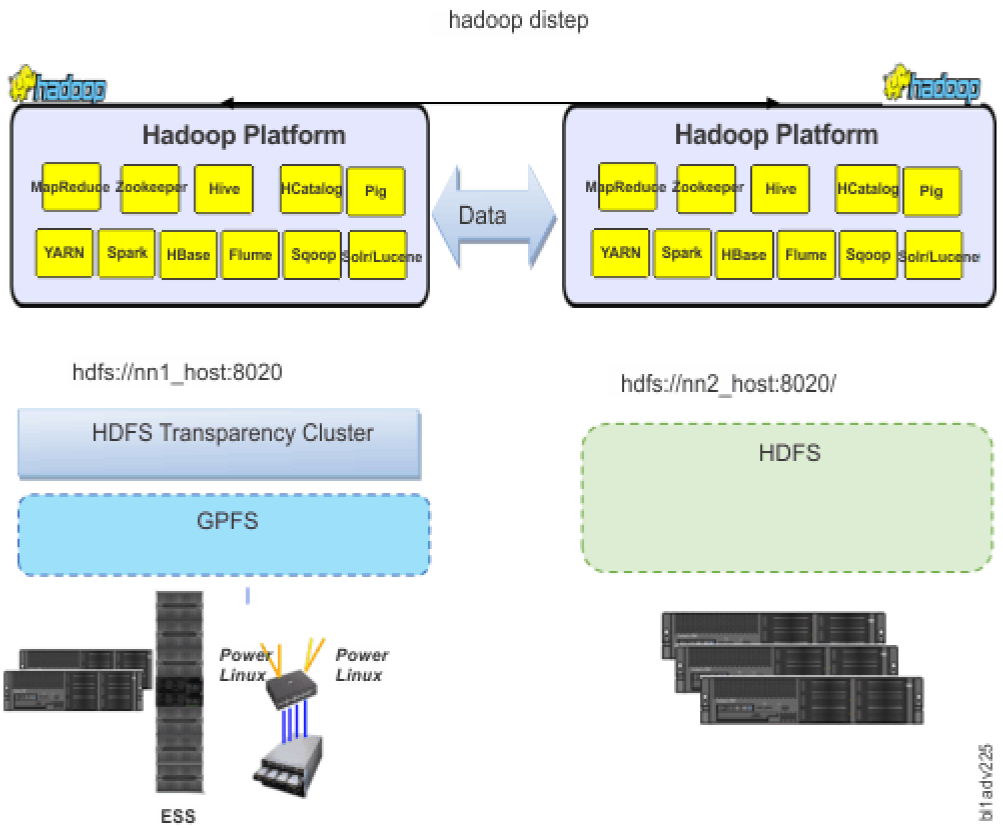
   Solution:

   Go to **Ambari** > **Oozie** > **CONFIGS** > **ADVANCED** > **Custom oozie-site** and add the following property:

```
<property>
   <name>oozie.service.HadoopAccessorService.supported.filesystems</name>
   <value>hdfs,viewfs</value>
</property>
```

Hadoop distcp support

The **hadoop distcp** command is used for data migration from HDFS to the IBM Storage Scale file system and between two IBM Storage Scale file systems.



There are no additional configuration changes. The **hadoop distcp** command is supported in HDFS transparency 2.7.0-2 (gpfs.hdfs-protocol-2.7.0-2) and later.

```
hadoop distcp hdfs://nn1_host:8020/source/dir
hdfs://nn2_host:8020/target/dir
```

Known Issues and Workaround

Issue 1: Permission is denied when the hadoop `distcp` command is run with the root credentials.

The super user root in Linux is not the super user for Hadoop. If you do not add the super user account to **`gpfs.supergroup`**, the system displays the following error message:

```
org.apache.hadoop.security.AccessControlException: Permission denied: user=root, access=WRITE,
inode="/user/root/.staging":hdfs:hdfs:drwxr-xr-x
```

at `org.apache.hadoop.hdfs.server.namenode.FSPermissionChecker.check(FSPermissionChecker.java:319)`.

Workaround

Configure root as a super user. Add the super user account to `gpfs.supergroup` in `gpfs-site.xml` to configure the root as the super user or run the related **`hadoop distcp`** command with the super user credentials. This is applicable only for HDFS Transparency 2.7.3-x. From HDFS Transparency 3.0.x, the configuration `gpfs.supergroup` has been removed from HDFS Transparency.

Issue 2: Access time exception while copying files from IBM Storage Scale to HDFS with the `-p` option

```
[hdfs@c8f2n03 conf]$ hadoop distcp -overwrite -p
hdfs://c16f1n03.gpfs.net:8020/testc16f1n03/
hdfs://c8f2n03.gpfs.net:8020/testc8f2n03
```

Error: `org.apache.hadoop.ipc.RemoteException(java.io.IOException)`: Access time for HDFS is not configured. Set the **`dfs.namenode.accesstime.precision`** configuration parameter at `org.apache.hadoop.hdfs.server.namenode.FSDirAttrOp.setTimes(FSDirAttrOp.java:101)`

Workaround

Change the `dfs.namenode.accesstime.precision` value from 0 to a value such as 3600000 (1 hour) in `hdfs-site.xml` for the HDFS cluster.

Issue 3: The `distcp` command fails when the src director is root.

```
[hdfs@c16f1n03 root]$ hadoop distcp hdfs://c16f1n03.gpfs.net:8020/
hdfs://c8f2n03.gpfs.net:8020/test5

16/03/03 22:27:34 ERROR tools.DistCp: Exception encountered

java.lang.NullPointerException
```

at `org.apache.hadoop.tools.util.DistCpUtils.getRelativePath(DistCpUtils.java:144)`

at `org.apache.hadoop.tools.SimpleCopyListing.writeToFileListing(SimpleCopyListing.java:353)`

Workaround

Specify at least one directory or file at the source directory.

Issue 4: The distcp command throws NullPointerException when the target directory is root in the federation configuration but the job is completed.

This is not a real issue. For more information, see https://issues.apache.org/jira/browse/HADOOP-11724.

Note: This will not impact your data copy.

Multiple IBM Storage Scale File System support

This feature is available from HDFS Transparency version 2.7.3-1.

NameNode federation feature in Hadoop 3.x enables adding multiple native HDFS namespaces under single Hadoop cluster. This feature was added to improve the HDFS NameNode horizontal scaling. If you want to use native HDFS namespace and IBM Storage Scale namespaces under the same Hadoop cluster, you can either use Hadoop Storage Tiering with IBM Storage Scale or ViewFS support. You can have multiple IBM Storage Scale file systems under single Hadoop cluster with ViewFS support. However, federation is not officially supported by HortonWorks HDP nor is it certificated with Hive in the Hadoop community. The multiple IBM Storage Scale file system support is designed to help resolve the federation issue.

If you want to enable this feature, you could configure the following properties in the `/usr/lpp/mmfs/hadoop/etc/hadoop/gpfs-site.xml` (for HDFS Transparency 2.7.3-x) or `/var/mmfs/hadoop/etc/hadoop/gpfs-site.xml` (for HDFS Transparency 3.0.x).

Configure the HDFS Transparency on the primary file system as the 1st file system defined in the **`gpfs.mnt.dir`** field.

Once the primary file system configuration is working properly, enable the multiple IBM Storage Scale file system support.

Note:

- Currently multiple IBM Storage Scale file system only supports 2 file systems.
- Do not put the NameNode and DataNode onto the same node when you are using multiple file system configuration. The reason for this being when either the NameNode or DataNode is stopped, the second file system gets unmounted and that leaves the other (still running) daemon non-functional.

Example of the `gpfs-site.xml` configuration for multiple file systems:

```
<property>
  <name>gpfs.mnt.dir</name>
  <value>/path/fpo,/path/ess1</value>
</property>
<property>
  <name>gpfs.storage.type</name>
  <value>local,shared</value>
</property>
<property>
  <name>gpfs.replica.enforced</name>
  <value>dfs,gpfs</value>
</property>
```

The `gpfs.mnt.dir` is a comma delimited string used to define the mount directory for each file system. In the above configuration, we have two file systems with mount point `/path/fpo` and `/path/ess1`. The first file system, `/path/fpo` will be considered as the primary file system.

The `gpfs.storage.type` is a comma delimited string used to define the storage type for each file system defined by `gpfs.mnt.dir`. Currently, only support 2 file systems mount access as `local,shared` or as `shared,shared`. The `local` means the file system with the same index in **`gpfs.mnt.dir`** is the IBM Storage Scale FPO file system with locality. The `shared` means the file system with the same index in **`gpfs.mnt.dir`** is the SAN-based file system or IBM ESS. You need to configure the **`gpfs.storage.type`** values correctly; otherwise, performance will be impacted. To check if the file system is `local` or `shared`, run **`mmlspool <fs-name> all -L`** to see whether the **`allowWriteaAffinity`** of the file system datapool is *yes* or *no*. If the value is *yes*, configure `local` for this file system. If the value is *no*, configure `shared` for this file system.

The **`gpfs.replica.enforced`** is a comma delimited string used to define the replica enforcement policy for all file systems defined by **`gpfs.mnt.dir`**.

Sync the above changes to all the HDFS Transparency nodes and restart HDFS Transparency. HDFS Transparency will mount all the non-primary file systems with bind mode into the primary file system.

In the above example, the primary file system is /path/fpo. The /path/fpo/<gpfs.data.dir> is the root directory for HDFS Transparency. The secondary file system /path/ess1 will be mapped as /path/fpo/<gpfs.data.dir>/ess1 directory virtually. Therefore, if you have /path/fpo/<gpfs.data.dir>/file1, /path/fpo/<gpfs.data.dir>/file2, /path/fpo/<gpfs.data.dir>/dir1, after mapping, the **hadoop dfs -ls /** will see /file1, /file2, /dir1 and /ess1. Use the **hadoop dfs -ls /ess1** to list all files/directories under /path/ess1.

Note:

1. The **gpfs.data.dir** is a single directory and it is always configured for the primary file system.
2. In the example, if the directory /path/fpo/<gpfs.data.dir>/ess1 exists and is not empty, on starting HDFS Transparency, an exception will be reported about the /path/fpo/<gpfs.data.dir>/ess1 directory is not empty and will fail to start. To resolve this issue, rename the directory /path/fpo/<gpfs.data.dir>/ess1 or remove all the files under the /path/fpo/<gpfs.data.dir>/ess1/ directory so that the directory does not contain any contents.
3. When HDFS Transparency is stopped, the second file system that is not the primary file system (1st file system) gets unmounted. The unmount will remove the GPFS file system but not the local directory name used for the mount. Ensure that this local directory used for the mounting of the 2nd file system does not contain any data when you start HDFS Transparency. Otherwise, HDFS Transparency will not be able to restart.

HDFS encryption

IBM Storage Scale already offers in-built encryption support. HDFS level encryption for IBM Storage Scale HDFS transparency connector is also supported.

It is important to understand the difference between HDFS level encryption and in-built encryption with IBM Storage Scale. HDFS level encryption is per user based whereas in-built encryption is per node based. Therefore, if the use case demands more fine-grained control at the user level, use HDFS level encryption. However, if you enable HDFS level encryption, you will not be able to get in-place analytics benefits such as accessing the same data with HDFS and POSIX/NFS.

This is supported since HDFS Transparency 3.0.0-0 and 2.7.3-4. This requires Ranger and Ranger KMS and this has only been tested over HortonWorks stack. If you plan to enable this for open source Apache, you should enable it on the native HDFS first and confirm it is working before you switch native HDFS into HDFS Transparency.

To enable native HDFS encryption, configure **gpfs.ranger.enabled**=*true* in gpfs-site.xml and configure the following value for gpfs-site.xml from Ambari GUI:

Configuration	Value (default)
gpfs.encryption.enabled	true (false)
gpfs.ranger.enabled	true (true)

Note: From HDFS Transparency 3.1.0-6 and 3.1.1-3, ensure that the **gpfs.ranger.enabled** field is set to *scale*. The scale option replaces the original *true*/*false* values.

Known limits

- CES HDFS does not support **hdfs crypto -listZones**. For a workaround with the GPFS policy engine, see the step 31 in the Second generation HDFS Transparency Protocol troubleshooting topic.
- Files in HDFS snapshot against encryption zone files are not supported for read/write.

Remote mount at fileset level

This topic lists the steps to configure HDFS Transparency to mount a remote fileset.

To configure HDFS Transparency to mount a remote fileset, perform the following:

1. From the owning cluster, add the remote fileset into the allowed fileset list. For more information, see Fileset access control for remote clusters.

   ```
   [root@owningcluster ~]# mmauth grant accessingcluster -f gpfs --fileset fset1,root
   mmauth: [I] Collecting fileset information ...
   mmauth: Granting cluster accessingcluster access to file system gpfs:
           access type rw; root credentials will not be remapped.
   mmauth: Propagating the cluster configuration data to all affected nodes.
   mmauth: Command successfully completed
   ```

   Note: For the command to run successfully, ensure that the root fileset is included in the allow list when you are enabling the remote fileset access control.

2. Ensure that only the granted filesets are visible from the accessing cluster.

   ```
   [root@accessingcluster ~]# mmlsfileset remotefs
   Filesets in file system 'remotefs':
   Name                     Status    Path
   root                       Linked    /remotefs
   indepfset1            Linked    /remotefs/indepfset1
   depfset1               Linked    /remotefs/depfset1
   ```

3. Stop the HDFS Transparency cluster.

   ```
   mmhdfs hdfs stop
   ```

4. For remote mount, set the granted fileset as the **`gpfs.data.dir`** parameter in `gpfs-site.xml`. Both independent and dependent filesets are supported. If you are operating with 2fs mode, no configuration changes are needed.

   ```
     <property>
       <name>gpfs.mnt.dir</name>
       <value>/remotefs</value>
       <description>gpfs mount point</description>
     </property>

   <property>
       <name>gpfs.data.dir</name>
       <value>indepfset1</value>
       <description>Set this to a sub directory in gpfs mount point will make this sub
   directory as the root directory from hadoop client point of view. Leave it empty to make
   whole gpfs file system visible to hadoop client. When specify the sub directory, gpfs mount
   point should not be included in the string.</description>
   </Property>
   ```

5. Upload and sync the changes across all the transparency nodes.

   ```
   mmhdfs config upload
   ```

6. Start all the transparency nodes and check the status of namenode.

   ```
   mmhdfs hdfs start; hdfs haadmin -getAllServiceState
   ```

7. Change the fileset permissions from *700* to *755*.

   ```
   hdfs dfs -chmod 755 /
   ```

High availability configuration

For HortonWorks HDP, you could configure HA from Ambari GUI directly. For open source Apache Hadoop, refer the following sections.

Manual HA switch configuration

High Availability (HA) is implemented in HDFS Transparency by using a shared directory in the IBM Storage Scale file system.

If your HDFS Transparency version is 2.7.0-2+, configure automatic HA according to "Automatic NameNode service HA" on page 196.

In the following configuration example, the HDFS `nameservice` ID is `mycluster` and the NameNode IDs are nn1 and nn2.

1. Define the nameservice ID in the `core-site.xml` file that is used by the Hadoop distribution. If you are using IBM BigInsights IOP or Hortonworks HDP, change this configuration in the Ambari GUI and restart the HDFS services to synchronize it with all the Hadoop nodes.

```
<property>
<name>fs.defaultFS</name>
<value>hdfs://mycluster</value>
</property>
```

2. Configure the `hdfs-site.xml` file that is used by the Hadoop distro. If you are using IBM BigInsights IOP or Hortonworks HDP, change these configurations in the Ambari GUI and restart the HDFS services to synchronize it with all the Hadoop nodes.

```
<property>
<!--define dfs.nameservices ID-->
<name>dfs.nameservices</name>
<value>mycluster</value>
</property>
```

```
<property>
<!--define NameNodes ID for HA-->
<name>dfs.ha.namenodes.mycluster</name>
<value>nn1,nn2</value>
</property>
```

```
<property>
<!--Actual hostname and rpc address of NameNode ID-->
<name>dfs.namenode.rpc-address.mycluster.nn1</name>
<value>c8f2n06.gpfs.net:8020</value>
</property>
```

```
<property>
<!--Actual hostname and rpc address of NameNode ID-->
<name>dfs.namenode.rpc-address.mycluster.nn2</name>
<value>c8f2n07.gpfs.net:8020</value>
</property>
```

```
<property>
<!--Actual hostname and http address of NameNode ID-->
<name>dfs.namenode.http-address.mycluster.nn1</name>
<value>c8f2n06.gpfs.net:50070</value>
</property>
```

```
<property>
<!--Actual hostname and http address of NameNode ID-->
<name>dfs.namenode.http-address.mycluster.nn2</name>
<value>c8f2n07.gpfs.net:50070</value>
</property>
```

```
<property>
<!--Shared directory used for status sync up-->
<!--Shared directory should be under gpfs.mnt.dir but not gpfs.data.dir-->
<name>dfs.namenode.shared.edits.dir</name>
<value>file:///<gpfs.mnt.dir>/HA</value>
</property>
```

```
<property>
<name>dfs.ha.standby.checkpoints</name>
<value>false</value>
</property>
```

```
<property>
<name>dfs.client.failover.proxy.provider.mycluster</name>
<value>org.apache.hadoop.hdfs.server.namenode.ha.ConfiguredFailoverProxyProvider</value>
</property>
```

The native HDFS uses the directory specified by **dfs.namenode.shared.edits.dir** configuration parameter to save information shared between the active NameNode and the standby NameNode. HDFS Transparency 2.7.3-4 and 3.1.0-0 uses this directory to save Kerberos delegation token related information. If you revert from HDFS Transparency back to the native HDFS, please revert **dfs.namenode.shared.edits.dir** configuration parameter back to the one used for the native HDFS. In Ambari Mpack 2.4.2.7 and Mpack 2.7.0.1, the **dfs.namenode.shared.edits.dir** parameter is set automatically when integrating or unintegrating IBM Storage Scale service.

The **dfs.namenode.shared.edits.dir** configuration parameter must be consistent with gpfs.mnt.dir defined in /usr/lpp/mmfs/hadoop/etc/hadoop/gpfs-site.xml. You can create the directory /<gpfs.mnt.dir>/HA and change the ownership to hdfs:hadoop before starting the HDFS transparency services.

For HDFS Transparency releases earlier than 2.7.3-4 and also the 3.0.0 release, the **dfs.ha.standby.checkpoints** must be set as *false*. If not, you will see a log of exceptions in the standby NameNode logs. However, for HDFS Transparency 2.7.3-4 and 3.1.0-0, the **dfs.ha.standby.checkpoints** must be set as *true*.

For example,

```
ERROR ha.StandbyCheckpointer (StandbyCheckpointer.java:doWork(371)) - Exception in
doCheckpoint
```

If you are using Ambari, search under **HDFS** > **Configs** to see if the **dfs.ha.standby.checkpoints** field is set to *false*. If the field is not found, add the field to the **Custom hdfs-site** > **Add property** and set it to *false*. Save Config, and restart HDFS and any services that you might see with restart icon in Ambari.

Note: This disables the exception written to the log.

To not show the alert from Ambari, click on the alert and disable the alert from Ambari GUI.

HDFS transparency does not have fsImage and editLogs. Therefore, do not perform checkpoints from the standby NameNode service.

After 2.7.3-4 of 2.x and 3.1.0-0 of 3.x, HDFS transparency has minimal EditLogs support. Therefore, need checkpoints (dfs.ha.standby.checkpoints=true setting) to clean up outdated EditLogs. The 3.0.0 release of HDFS transparency does not support EditLog (dfs.ha.standby.checkpoints=false setting).

The **dfs.client.failover.proxy.provider.mycluster** configuration parameter must be changed according to the name service ID. In the above example, the name service ID is configured as mycluster in core-site.xml. Therefore, the configuration name is dfs.client.failover.proxy.provider.mycluster.

Note: If you enable Short Circuit Read in the Short Circuit Read Configuration section, the value of the configuration parameter must be **org.apache.hadoop.gpfs.server.namenode.ha.ConfiguredFailoverProxyProvider**.

3. Follow the guide in the Sync Hadoop configurations section to synchronize core-site.xml and hdfs-site.xml from the Hadoop distribution to any one node that is running HDFS transparency services. For example, HDFS_Transparency_node1.
4. For HDFS Transparency 2.7.0-x, on **HDFS_Transparency_node1**, modify /usr/lpp/mmfs/hadoop/etc/hadoop/hdfs-site.xml:

```
 <property>
<name>dfs.client.failover.proxy.provider.mycluster</name>
<value>org.apache.hadoop.gpfs.server.namenode.ha.ConfiguredFailoverProxyProvider</value>
</property>
```

With this configuration, WebHDFS service functions correctly when NameNode HA is enabled.

Note: On HDFS transparency nodes, the configuration value of the key **dfs.client.failover.proxy.provider.mycluster** in hdfs-site.xml is different from that in Step2.

Note: This step should not be performed from HDFS Transparency 2.7.2-x.

5. On **HDFS_Transparency_node1**, run the command as the root user to synchronize the HDFS Transparency configuration to all the HDFS transparency nodes:

   ```
   # mmhadoopctl connector syncconf /usr/lpp/mmfs/hadoop/etc/hadoop
   ```

6. If you are using HDFS Transparency version 2.7.3-4 or 3.1.0-0, before you start HDFS transparency for the first time, run the `/usr/lpp/mmfs/hadoop/bin/hdfs namenode -initializeSharedEdits` command to initialize the shared edits directory. Run this command only after configuring HA and before starting the service.
7. Start the HDFS transparency service by running the **mmhadoopctl** command:

   ```
   # mmhadoopctl connector start
   ```

8. After the service starts, both NameNodes are in the standby mode by default. You can activate one NameNode by using the following command so that it responds to the client:

   ```
   # /usr/lpp/mmfs/hadoop/bin/gpfs haadmin -transitionToActive --forceactive
   [namenode ID]
   ```

   For example, you can activate the nn1 NameNode by running the following command:

   ```
   # /usr/lpp/mmfs/hadoop/bin/gpfs haadmin -transitionToActive -forceactive nn1
   ```

   If the nn1 NameNode fails, you can activate another NameNode and relay the service by running the following command:

   ```
   # /usr/lpp/mmfs/hadoop/bin/gpfs haadmin -transitionToActive -forceactive nn2
   ```

   Note: The switch must be done manually. Automatic switch will be supported in the future releases.

   Use the following command to view the status of the NameNode:

   ```
   # /usr/lpp/mmfs/hadoop/bin/gpfs haadmin -getServiceState [namenode ID]
   ```

   You could check your `/usr/lpp/mmfs/hadoop/etc/hadoop/hdfs-site.xml` or run the following commands to figure out the [NameNode ID]:

   ```
   #/usr/lpp/mmfs/hadoop/bin/gpfs getconf -confKey fs.defaultFS
   hdfs://mycluster
   #hdfs getconf -confKey dfs.ha.namenodes.mycluster
   nn1,nn2
   ```

   After one NameNode becomes active, you can start the other Hadoop components, such as hbase and hive and run your Hadoop jobs.

   Note: When HA is enabled for HDFS transparency, you might see the following exception in the logs: Get corrupt file blocks returned error: Operation category READ is not supported in state standby.

   These are known HDFS issues: HDFS-3447 and HDFS-8910.

Automatic NameNode service HA

Automatic NameNode Service HA is supported in gpfs.hdfs-protocol 2.7.0-2 and later. The implementation of high availability (HA) is the same as NFS-based HA in native HDFS. The only difference is that except for the NFS shared directory in native HDFS, HA is not needed for HDFS transparency.

The prerequisite to configure automatic NameNode HA is to have zookeeper services running in the cluster.

Configuring Automatic NameNode Service HA

If you take a Hadoop distro, such as Hortonworks Data Platform, the zookeeper service is deployed by default.

If you select open-source Apache Hadoop, you must set up the Zookeeper service by following the instruction on the Zookeeper website.

After you set up the Zookeeper service, perform the following steps to configure automatic NameNode HA.

Note: In the following configuration example, HDFS Transparency NameNode service ID is *mycluster* and NameNode IDs are *nn1* and *nn2*. Zookeeper server `zk1.gpfs.net, zk2.gpfs.net` and `zk3.gpfs.net` are configured to support automatic NameNode HA. The ZooKeeper servers must be started before starting the HDFS Transparency cluster.

1. Define the NameNode service ID in the core-site.xml that is used by your Hadoop distribution.

   Note: If you are using IBM BigInsights IOP or Hortonworks HDP, you can change this configuration in Ambari GUI and restart the HDFS services to synchronize it with all the Hadoop nodes.

   ```
   <property>
   <name>fs.defaultFS</name>
   <value>hdfs://mycluster</value>
   </property>
   ```

2. Configure the `hdfs-site.xml` file used by your Hadoop distribution:

   Note: If you are using IBM BigInsights IOP or Hortonworks HDP, you can change this configuration in Ambari GUI and restart the HDFS services to synchronize it with all the Hadoop nodes.

   ```
   <property>
   <!--define dfs.nameservices ID-->
   <name>dfs.nameservices</name>
   <value>mycluster</value>
   </property>

   <property>
   <!--define NameNodes ID for HA-->
   <name>dfs.ha.namenodes.mycluster</name>
   <value>nn1,nn2</value>
   </property>

   <property>
   <!--Actual hostname and rpc address of NameNode ID-->
   <name>dfs.namenode.rpc-address.mycluster.nn1</name>
   <value>c8f2n06.gpfs.net:8020</value>
   </property>

   <property>
   <!--Actual hostname and rpc address of NameNode ID-->
   <name>dfs.namenode.rpc-address.mycluster.nn2</name>
   <value>c8f2n07.gpfs.net:8020</value>
   </property>

   <property>
   <!--Actual hostname and http address of NameNode ID-->
   <name>dfs.namenode.http-address.mycluster.nn1</name>
   <value>c8f2n06.gpfs.net:50070</value>
   </property>

   <property>
   <!--Actual hostname and http address of NameNode ID-->
   <name>dfs.namenode.http-address.mycluster.nn2</name>
   <value>c8f2n07.gpfs.net:50070</value>
   </property>


   <property>
   <!--Shared directory used for status sync up-->
   <!--Shared directory should be under gpfs.mnt.dir but not gpfs.data.dir-->
   <name>dfs.namenode.shared.edits.dir</name>
   <value>file:///<gpfs.mnt.dir>/HA</value>
   </property>


   <property>
   <name>dfs.ha.standby.checkpoints</name>
   <value>false</value>
   </property>

   <property>
   <name>dfs.client.failover.proxy.provider.mycluster</name>
   <value>org.apache.hadoop.hdfs.server.namenode.ha.ConfiguredFailoverProxyProvider</value>
   </property>
   ```

```
<property>
<name>dfs.ha.fencing.methods</name>
<value>shell(/bin/true)</value>
</property>

<property>
<name>dfs.ha.automatic-failover.enabled</name>
<value>true</value>
</property>

<property>
<name>ha.zookeeper.quorum</name>
<value>zk1.gpfs.net:2181,zk2.gpfs.net:2181,zk3.gpfs.net:2181</value>
</property>
```

The native HDFS uses the directory specified by **dfs.namenode.shared.edits.dir** configuration parameter to save information shared between the active NameNode and the standby NameNode. HDFS Transparency 2.7.3-4 and 3.1.0-0 uses this directory to save Kerberos delegation token related information. If you revert from HDFS Transparency back to the native HDFS, please revert **dfs.namenode.shared.edits.dir** configuration parameter back to the one used for the native HDFS. In Ambari Mpack 2.4.2.7 and Mpack 2.7.0.1, the **dfs.namenode.shared.edits.dir** parameter is set automatically when integrating or unintegrating IBM Storage Scale service.

The configuration **dfs.namenode.shared.edits.dir** must be consistent with gpfs.mnt.dir defined in /usr/lpp/mmfs/hadoop/etc/hadoop/gpfs-site.xml (for HDFS Transparency 2.7.x) or /var/mmfs/hadoop/etc/hadoop/gpfs-site.xml (for HDFS Transparency 3.x). You could create the /<gpfs.mnt.dir>/HA directory and change the ownership to hdfs:hadoop before starting the HDFS transparency services.

For HDFS Transparency releases earlier than 2.7.3-4 of HDP 2.x and also the 3.0.0 of HDP 3.x release, the **dfs.ha.standby.checkpoints** must be set as *false*. If not, you will see a log of exceptions in the standby NameNode logs. However, for the 2.7.3-4 of HDP 2.x and 3.1.0-0 of HDP 3.x, the **dfs.ha.standby.checkpoints** must be set as *true*.

For example,

```
ERROR ha.StandbyCheckpointer (StandbyCheckpointer.java:doWork(371)) - Exception in
doCheckpoint.
```

HDFS transparency does not have fsImage and editLogs. Therefore, do not perform checkpoints from the standby NameNode service.

After 2.7.3-4 for HDFS Transparency 2.7.x release or 3.1.0-0 for HDFS Transparency 3.x release, HDFS transparency has minimal EditLogs support. Therefore, need checkpoints (dfs.ha.standby.checkpoints=true setting) to clean up outdated EditLogs. The HDFS transparency release earlier than 2.7.3-4 or 3.0.0-0 do not support EditLog (dfs.ha.standby.checkpoints=false setting).

The configuration name dfs.client.failover.proxy.provider.mycluster must be changed according to the nameservice ID. In the above example, the nameservice ID is configured as *mycluster* in core-site.xml. Therefore, the configuration name is dfs.client.failover.proxy.provider.mycluster.

Note: If you enable Short Circuit Read in the "Short-circuit read configuration" on page 200, the value of this configuration must be *org.apache.hadoop.gpfs.server.namenode.ha.ConfiguredFailoverProxyProvider*.

3. To synchronize core-site.xml with hdfs-site.xml from your Hadoop distribution to any one node that is running HDFS transparency services, you could run **scp** to copy them into one of your HDFS Transparency node, assuming it is HDFS_Transparency_node1.
4. For HDFS Transparency 2.7.0-x, on HDFS_Transparency_node1, modify the /usr/lpp/mmfs/hadoop/etc/hadoop/hdfs-site.xml:

```
<property>
<name>dfs.client.failover.proxy.provider.mycluster</name>
```

```
<value>org.apache.hadoop.gpfs.server.namenode.ha.ConfiguredFailoverProxyProvider</value>
</property>
```

The WebHDFS service functions properly when NameNode HA is enabled.

Note: On HDFS transparency nodes, the above configuration value in `hdfs-site.xml` is different as that in Step2.

Note: This step should not be performed from HDFS Transparency 2.7.2-0 and later.

5. On HDFS_Transparency_node1, run the following command as the root user to synchronize HDFS Transparency configuration with all HDFS transparency nodes:

   For HDFS Transparency 2.7.3-x, run the following command:

   ```
   mmhadoopctl connector syncconf /usr/lpp/mmfs/hadoop/etc/hadoop
   ```

   For HDFS Transparency 3.0.x or 3.1.x, run the following command:

   ```
   mmhadoopctl connector syncconf /var/mmfs/hadoop/etc/hadoop
   ```

6. Before you start HDFS transparency for the first time, you should run the `/usr/lpp/mmfs/hadoop/bin/hdfs namenode -initializeSharedEdits` command.

   This initializes the shared edits directory.

   Note: Run this command just for once after configuring HA and before staring the service.

7. Start the HDFS Transparency service by running the **`mmhadoopctl`** command:

   ```
   mmhadoopctl connector start
   ```

8. Format the zookeeper data structure:

   For HDFS Transparency 2.7.3-x:

   ```
   /usr/lpp/mmfs/hadoop/bin/gpfs --config /usr/lpp/mmfs/hadoop/etc/hadoop/ zkfc -formatZK
   ```

   For HDFS Transparency 3.0.x or 3.1.x:

   ```
   /usr/lpp/mmfs/hadoop/bin/hdfs --config /var/mmfs/hadoop/etc/hadoop/ zkfc -formatZK
   ```

   This step is only needed when you start HDFS Transparency service for the first time. After that, this step is not needed when restarting HDFS Transparency service.

9. Start the zkfc daemon:

   ```
   /usr/lpp/mmfs/hadoop/sbin/hadoop-daemon.sh start zkfc -formatZK
   ```

   Run jps on the nn1 and nn2 NameNodes to check if the DFSZKFailoverController process has been started.

   Note: If the `-formatZK` option is not added, the system displays the following exception: `FATAL org.apache.hadoop.ha.ZKFailoverController: Unable to start failover controller. Parent znode does not exist`

10. Check the state of the services

    Run the following command to check that all NameNode services and DataNode services are up:

    ```
    # mmhadoopctl connector getstate
    ```

11. Run the following command to check the state of NameNode services:

    ```
    /usr/lpp/mmfs/hadoop/bin/gpfs haadmin -getServiceState [namenode ID]
    ```

You could check your /usr/lpp/mmfs/hadoop/etc/hadoop/hdfs-site.xml (for HDFS Transparency 2.7.3-x) or /var/mmfs/hadoop/etc/hadoop/hdfs-site.xml (for HDFS Transparency 3.0.x) or run the following commands to figure out the [namenode ID]:

```
#/usr/lpp/mmfs/hadoop/bin/gpfs getconf -confKey fs.defaultFS
hdfs://mycluster
#hdfs getconf -confKey dfs.ha.namenodes.mycluster
nn1,nn2
```

Note: When HA is enabled for HDFS transparency, the following exception might be logged: `Get corrupt file blocks returned error: Operation category READ is not supported in state standby`. These are unfixed HDFS issues: HDFS-3447 and HDFS-8910.

Short-circuit read configuration

In HDFS, read requests go through the DataNode. When the client requests the DataNode to read a file, the DataNode reads that file off the disk and sends the data to the client over a TCP socket. The short-circuit read obtains the file descriptor from the DataNode, allowing the client to read the file directly.

Short-circuit read feature works only when Hadoop client and HDFS Transparency DataNode are colocated on the same node. For example, if the Yarn's NodeManagers and HDFS Transparency DataNodes are on the same nodes, short circuit read will be effective when running the Yarn's jobs.

Note: Admin must enable the short circuit read in advance. For HortonWorks and IBM Storage Scale Mpack (Ambari integration), you could enable or disable short circuit read from Ambari GUI. For Apache Hadoop, refer the following sections.

Short-circuit reads provide a substantial performance boost to many applications.

For HDFS Transparency version 2.7.0-x

Short-circuit local read can only be enabled on Hadoop 2.7.0. HDFS Transparency versions 2.7.0-x does not support this feature in Hadoop 2.7.1/2.7.2. IBM BigInsights IOP 4.1 uses Hadoop version 2.7.1. Therefore, short circuit cannot be enabled over IBM BigInsights IOP 4.1 if HDFS Transparency 2.7.0-x is used. For more information on how to enable short-circuit read on other Hadoop versions, contact scale@us.ibm.com.

Configuring short-circuit local read

To configure short-circuit local reads, enable libhadoop.so and use the DFS Client shipped by the IBM Storage Scale HDFS transparency. The package name is gpfs.hdfs-protocol. You cannot use standard HDFS DFS Client to enable the short-circuit mode over the HDFS transparency.

To enable libhadoop.so, compile the native library on the target machine or use the library shipped by IBM Storage Scale HDFS transparency. To compile the native library on the specific machine, do the following steps:

1. Download the Hadoop source code from Hadoop community. Unzip the package and **cd** to that directory.
2. Build by mvn: **$ mvn package -Pdist,native -DskipTests -Dtar**
3. Copy hadoop-dist/target/hadoop-2.7.1/lib/native/libhadoop.so.* to $YOUR_HADOOP_PREFIX/lib/native/

   To use the libhadoop.so delivered by the HDFS transparency, copy /usr/lpp/mmfs/hadoop/lib/native/libhadoop.so to $YOUR_HADOOP_PREFIX /lib/native/libhadoop.so.

   The shipped libhadoop.so is built on x86_64, ppc64 or ppc64le respectively.

   Note: This step must be performed on all nodes running the Hadoop tasks.

Enabling DFS Client

To enable DFS Client, perform the following procedure:

1. On each node that accesses IBM Storage Scale in the short-circuit mode, back up **`hadoop-hdfs-2.7.0.jar`** using `$ mv $YOUR_HADOOP_PREFIX/share/hadoop/hdfs/hadoop-hdfs-2.7.0.jar $YOUR_HADOOP_PREFIX/share/hadoop/hdfs/hadoop-hdfs-2.7.0.jar.backup`.
2. Link **`hadoop-gpfs-2.7.0.jar`** to classpath using `$ln -s /usr/lpp/mmfs/hadoop/share/hadoop/hdfs/hadoop-gpfs-2.7.0.jar $YOUR_HADOOP_PREFIX/share/hadoop/hdfs/hadoop-gpfs-2.7.0.jar`
3. Update the `core-site.xml` file with the following information:

   ```
   <property>
     <name>fs.hdfs.impl</name>
     <value>org.apache.hadoop.gpfs.DistributedFileSystem</value>
   </property>
   ```

Short-circuit reads make use of a UNIX domain socket. This is a special path in the file system that allows the client and the DataNodes to communicate. You need to set a path to this socket. The DataNode needs to be able to create this path. However, users other than the HDFS user or root must not be able to create this path. Therefore, paths under `/var/run` or `/var/lib` folders are often used.

The client and the DataNode exchange information through a shared memory segment on the `/dev/shm` path. Short-circuit local reads need to be configured on both the DataNode and the client. Here is an example configuration.

```
<configuration>
<property>
<name>dfs.client.read.shortcircuit</name>
<value>true</value>
</property>
<property>
<name>dfs.domain.socket.path</name>
<value>/var/lib/hadoop-hdfs/dn_socket</value>
</property>
</configuration>
```

Synchronize all these changes on the entire cluster and if needed, restart the service.

Note: The `/var/lib/hadoop-hdfs` and `dfs.domain.socket.path` must be created manually by the root user before running the short-circuit read. The `/var/lib/hadoop-hdfs` must be owned by the root user. If not, the DataNode service fails when starting up.

```
#mkdir -p  /var/lib/hadoop-hdfs
#chown root:root /var/lib/hadoop-hdfs
#touch   /var/lib/hadoop-hdfs/${dfs.dome.socket.path}
#chmod 666 /var/lib/hadoop-hdfs/${dfs.dome.socket.path}
```

The permission control in short-circuit reads is similar to the common user access in HDFS. If you have the permission to read the file, then you can access it through short-circuit read.

For HDFS Transparency version 2.7.2-x/2.7.3-x/3.x

The short-circuit read configuration described in this section is only applicable to Apache Hadoop 2.7.1+. Therefore, if you are using Apache Hadoop, you can follow the steps below to enable it. For HortonWorks Data Platform (HDP), you could enable/disable short circuit read from the Ambari GUI. The following steps describe how to manually enable the native library (libhadoop.so), how to replace the client JARs with the versions provided by HDFS Transparency and how to change the configuration in hdfs-site.xml with the correct values for **`dfs.client.read.shortcircuit`** and **`dfs.domain.socket.path`**.

Note: For configuring short-circuit read, glibc version must be at least version 2.14.

To configure short-circuit read, the libhadoop.so library must be enabled and the HDFS client must use the HDFS client JAR file shipped with IBM Storage Scale HDFS Transparency.

To configure short-circuit read, the libhadoop.so library must be enabled and the HDFS client must use the HDFS client JAR file shipped with IBM Storage Scale HDFS Transparency.

To enable libhadoop.so, you can:

1. Use the pre-compiled libhadoop.so shipped with IBM Storage Scale HDFS transparency.
2. Compile libhadoop.so manually.

To use the pre-compiled libhadoop.so, follow the steps listed below:

Note: This example uses Hadoop 3.1.3. If you are using any other version, change the paths accordingly.

1.
   ```
   HADOOP_DISTRO_HOME=/opt/hadoop/hadoop-3.1.3/
   TRANSPARENCY_HOME=/usr/lpp/mmfs/hadoop/
   ```

   Create a backup of any existing libhadoop.so:

   ```
   mv $HADOOP_DISTRO_HOME/lib/native/libhadoop.so $HADOOP_DISTRO_HOMElib/native/
   libhadoop.so.backup
   ```

2. Link to the libhadoop.so library shipped with HDFS Transparency:

   ```
   ln -s $TRANSPARENCY_HOME/lib/native/libhadoop.so $HADOOP_DISTRO_HOME/lib/native/libhadoop.so
   ```

3. Repeat the same for libhadoop.so1.0.0

   ```
   mv $HADOOP_DISTRO_HOME/lib/native/libhadoop.so.1.0.0 $HADOOP_DISTRO_HOME/lib/native/
   libhadoop.so.1.0.0.backup
   ln -s $TRANSPARENCY_HOME/lib/native/libhadoop.so.1.0.0 $HADOOP_DISTRO_HOME/lib/native/
   libhadoop.so.1.0.0
   ```

   Note: These steps must be performed on all nodes that are DataNodes and HDFS Clients (for example, NodeManagers).

To manually compile libhadoop.so, follow the steps listed below:

Note: This example uses Hadoop 2.7.2. If you are using another version, change the paths accordingly.

1.
   ```
   HADOOP_DISTRO_HOME=/opt/hadoop/hadoop-2.7.2/
   ```

   Download the Hadoop source code from Hadoop community. Unzip the package under <target-hadoop-path>.

2.
   ```
   cd <target-hadoop-path>/hadoop-2.7.2-src/hadoop-common-project/hadoop-common/
   ```

3.
   ```
   mvn package -Pdist,native -DskipTests -Dtar
   ```

4.
   ```
   cp <target-hadoop-path>/hadoop-common-2.7.2/lib/native/libhadoop.so.* to
   $HADOOP_DISTRO_HOME/lib/native/
   ```

Using the HDFS Transparency Client JAR in version 2.7.2-x/2.7.3-x

Note: The following example uses Hadoop 2.7.2. If you are using another version, change the paths accordingly.

1.
   ```
   HADOOP_DISTRO_HOME=/opt/hadoop/hadoop-2.7.2/
   HADOOP_DISTRO_VERSION=2.7.2
   TRANSPARENCY_HOME=/usr/lpp/mmfs/hadoop/
   TRANSPARENCY_VERSION=2.7.2
   ```

   Create a backup of the existing client JAR:

   ```
   mv $HADOOP_DISTRO_HOME/share/hadoop/hdfs/hadoop-hdfs-$HADOOP_DISTRO_VERSION.jar
   $HADOOP_DISTRO_HOME/share/hadoop/hdfs/hadoop-hdfs-$HADOOP_DISTRO_VERSION.jar.backup
   ```

2. Link to the client JAR shipped with HDFS Transparency:

```
ln -s $TRANSPARENCY_HOME/share/hadoop/hdfs/hadoop-hdfs-$TRANSPARENCY_VERSION.jar
$HADOOP_DISTRO_HOME/share/hadoop/hdfs/hadoop-hdfs-$HADOOP_DISTRO_VERSION.jar
```

Note: These steps must be performed on all nodes that are DataNodes and HDFS Clients (for example, NodeManagers).

Using the HDFS Transparency Client JAR in version 3.1.0-x/3.1.1-x

Note: The following example uses Apache Hadoop 3.1.3. If you are using another version, change the paths accordingly.

1.
   ```
   HADOOP_DISTRO_HOME=/opt/hadoop/hadoop-3.1.3/
   HADOOP_DISTRO_VERSION=3.1.3
   TRANSPARENCY_HOME=/usr/lpp/mmfs/hadoop/
   TRANSPARENCY_VERSION=3.1.1
   ```

   Create a backup of the existing client JAR:

   ```
   mv $HADOOP_DISTRO_HOME/share/hadoop/hdfs/hadoop-hdfs-client-$HADOOP_DISTRO_VERSION.jar
   $HADOOP_DISTRO_HOME/share/hadoop/hdfs/hadoop-hdfs-client-$HADOOP_DISTRO_VERSION.jar.backup
   ```

2. Link to the client JAR shipped with HDFS Transparency:

   ```
   ln -s $TRANSPARENCY_HOME/share/hadoop/hdfs/hadoop-hdfs-client-$TRANSPARENCY_VERSION.jar
   $HADOOP_DISTRO_HOME/share/hadoop/hdfs/hadoop-hdfs-client-$HADOOP_DISTRO_VERSION.jar
   ```

Note: These steps must be performed on all nodes that are DataNodes and HDFS Clients (for example, NodeManagers).

Enable short-circuit read in hdfs-site.xml

If you are running HDP, use Ambari to enable short-circuit read. For more information, see HDP 3.X "Short-circuit read (SSR)" on page 427.

Enabling short-circuit read manually:

Note:

1. These steps must be performed on all nodes that are DataNodes and HDFS Clients (for example, NodeManagers).
2. Short-circuit reads make use of a UNIX domain socket. This is a special path in the file system that allows the client and the DataNodes to communicate. You need to set a path to this socket. The DataNode needs to be able to access this path. However, users other than the root user should not be able to access this path. Therefore, paths under `/var/run` or `/var/lib` are often used.
3. The directory `/var/lib/hadoop-hdfs` and socket file must be created manually by the root user before enabling short-circuit read in the configuration. The directory must be owned by the root user. Otherwise the DataNode service fails when starting up.

   ```
   mkdir -p  /var/lib/hadoop-hdfs
   chown root:root /var/lib/hadoop-hdfs
   touch   /var/lib/hadoop-hdfs/dn_socket
   chmod 666 /var/lib/hadoop-hdfs/dn_socket
   ```

In addition, the client and the DataNode exchange information through a shared memory segment on the `/dev/shm` path.

1. Add or change the following parameters in `/var/mmfs/hadoop/etc/hadoop/hdfs-client.xml` and the `hdfs-client.xml` used by your Apache Hadoop distro:

   ```
   <configuration>
   <property>
   <name>dfs.client.read.shortcircuit</name>
   <value>true</value>
   </property>
   <property>
   <name>dfs.domain.socket.path</name>
   <value>/var/lib/hadoop-hdfs/dn_socket</value>
   ```

```
</property>
</configuration>
```

2. Synchronize the changes in the entire cluster and restart the Hadoop client cluster to ensure that all the services are aware of this configuration change.
3. Restart the HDFS Transparency cluster or follow the "Automatic Configuration Refresh " on page 207 section to refresh the configuration without interrupting the HDFS Transparency service.

   The permission control in short-circuit reads is similar to the common user access in HDFS. If you have the permission to read the file, then you can access it through short-circuit read.

Note: For Apache Hadoop, if you run other components, such as Oozie, Solr, Spark, these components will also be packaged with a `hadoop-hdfs-<version>.jar` in Apache Hadoop 2.7.x or `hadoop-hdfs-client-<version>.jar` in Apache Hadoop 3.x. When enabling short circuit read/write for these additional clients, one needs to re-package these as well with the `hadoop-hdfs-2.7.3.jar` from HDFS Transparency 2.7.3-x or `hadoop-hdfs-client-<version>.jar` from HDFS Transparency 3.x.

When you disable short circuit read/write, you need to re-package these again with the original `hadoop-hdfs-2.7.3.jar` from Apache Hadoop 2.7.x or `hadoop-hdfs-client-<version>.jar` from Apache Hadoop 3.x.

Verify that short-circuit read/write is working

After enabling short-circuit read, short-circuit write is also enabled. This is because short-circuit write is enabled by default internally with the **`gpfs.short-circuit-write.enabled`** field set to *yes*. For more information, see "Short circuit write" on page 205.

To verify that short circuit read and write is working properly, run the following commands on all the DataNodes:

```
grep "Listening on UNIX domain socket" /var/log/transparency/*
```

When a workload is running:

```
grep "REQUEST_SHORT_CIRCUIT_FDS" /var/log/transparency/*
```

Note: The location of the DataNode log files differ depending on your HDFS Transparency version and Hadoop distribution (for example, HDP).

mmhadoopctl supports dual network

The HDFS Transparency **`mmhadoopctl`** command now supports dual network configuration.

The **`mmhadoopctl`** for dual network setup is not used in Ambari. Therefore, if you are using Ambari, see "Dual-network deployment" on page 396 section for setup.

HDFS Transparency **`mmhadoopctl`** command requires the NameNode and DataNode to have password-less ssh access setup for the network.

The HDFS Transparency dual network setup is for the case when HDFS Transparency node names are on a private network and cannot configure password-less ssh access.

For HDFS Transparency **`mmhadoopctl`** to work properly using network without password-less ssh access configured, the following export variable, NODE_HDFS_MAP_GPFS, will need to be set in order to convert the HDFS Transparency node names set in the HDFS Transparency config files to use the IBM Storage Scale admin node name that has password-less ssh setup.

Scenario:

IBM Storage Scale admin network is configured on network 1.

HDFS Transparency NameNode and DataNode and all the Hadoop nodes are configured to use network 2.

Note:

- IBM Storage Scale requires only the admin network to have password-less ssh access.

- It is required to use the export command to export the NODE_HDFS_MAP_GPFS variable in the `hadoop-env.sh` file to generate the mapping file correctly.
- Delete the `/var/mmfs/hadoop/init/nodemap` mapping file on all nodes if needed to regenerate this file when HDFS Transparency restarts.
- Ensure that you delete the nodemap file on all the nodes before doing a **`syncconf`**.
- In order to run the **`mmhadoopctl connector start/stop`** command on the node in a dual network environment, the export NODE_HDFS_MAP_GPFS=yes is required to be set so that the nodemap file is created for the node.

Steps:

1. Edit configuration.

   Manually add the export line 'export NODE_HDFS_MAP_GPFS=yes' in the `/var/mmfs/hadoop/etc/hadoop/hadoop-env.sh` file.

   ```
   # cat hadoop-env.sh | tail -2
   export NODE_HDFS_MAP_GPFS=yes
   ```

   This will generate a request for HDFS Transparency to convert the node names used in HDFS Transparency config files to the IBM Storage Scale admin node names.

   A mapping file `/var/mmfs/hadoop/init/nodemap` will be created.

   If the Hadoop configuration hosts is changed (add/delete), then the mapping file `/var/mmfs/hadoop/init/nodemap` will need to be deleted so that restarting the HDFS Transparency can re-create a new mapping file with the correct host configuration entries.
2. Sync the configuration.
   - Ensure to remove all existing `/var/mmfs/hadoop/init/nodemap` files from all the nodes.
   - Run **`mmhadoopctl syncconf`** to sync the configuration files in the cluster. For syncconf syntax, see “Sync HDFS Transparency configurations” on page 61.
3. Start Transparency.

   The **`mmhadoopctl`** will now be set to use the Scale admin node names.

Short circuit write

Short circuit write is supported since HDFS Transparency 2.7.3-1.

If HDFS Client and HDFS Transparency DataNode are located on the same node, when writing file from HDFS client, Short circuit write will write data directly into IBM Storage Scale file system instead of writing data through RPC. This could reduce the RPC latency through the local loop network adapter and thus enhance the write performance.

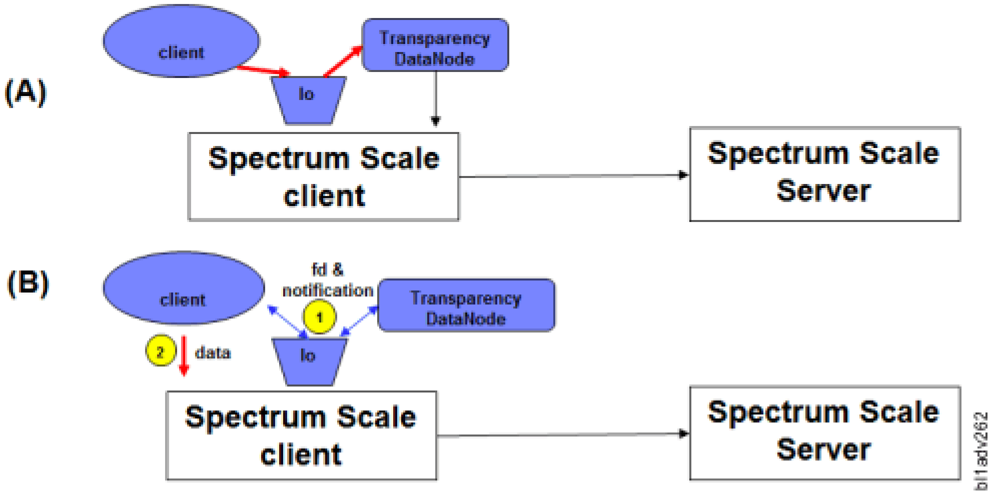


Figure 18. Short Circuit Write Logic

In Figure 18 on page 206, (A) is for the original logic for data write. With short circuit write enabled, the data write logic will be shown as (B). The data will be written directly into IBM Storage Scale file system.

If you want to enable this feature, refer "Short-circuit read configuration" on page 200 to enable short circuit read first. By default, when short circuit read is enabled, short circuit write is also enabled. When short circuit read is disabled, short circuit write is also disabled.

If you want to disable short circuit write when short circuit read is enabled:

1. Add the following configuration in `hdfs-site.xml` for Hadoop client.

   If you take HortonWorks HDP, change this on `Ambari/HDFS/configs` and restart HDFS service. If you take open source Apache Hadoop, change this in `<Hadoop-home-dir>/etc/hadoop/hdfs-site.xml` on all Hadoop nodes.

   ```
   <property>
   <name>gpfs.short-circuit-write.enabled</name>
   <value>false</value>
   </property>
   ```

2. Add the same configuration into `gpfs-site.xml`.

   If you take HortonWorks HDP, change this on `Ambari/Spectrum Scale/Configs/custom gpfs-site` and restart IBM Storage Scale service from Ambari.

   If you take open source Apache Hadoop, change this in `/usr/lpp/mmfs/hadoop/etc/hadoop/gpfs-site.xml` (for HDFS Transparency 2.7.3-x) or `/var/mmfs/hadoop/etc/hadoop/gpfs-site.xml` (for HDFS Transparency 3.0.x) and run `/usr/lpp/mmfs/bin/mmhadoopctl connector syncconf /usr/lpp/mmfs/hadoop/etc/hadoop` (for HDFS Transparency 2.7.3-x) or `/usr/lpp/mmfs/bin/mmhadoopctl connector syncconf /var/mmfs/hadoop/etc/hadoop` (for HDFS Transparency 3.0.x) to sync the change to all HDFS Transparency nodes.

Multiple Hadoop clusters over the same file system

By using HDFS transparency, you can configure multiple Hadoop clusters over the same IBM Storage Scale file system. For each Hadoop cluster, you need one HDFS transparency cluster to provide the file system service.

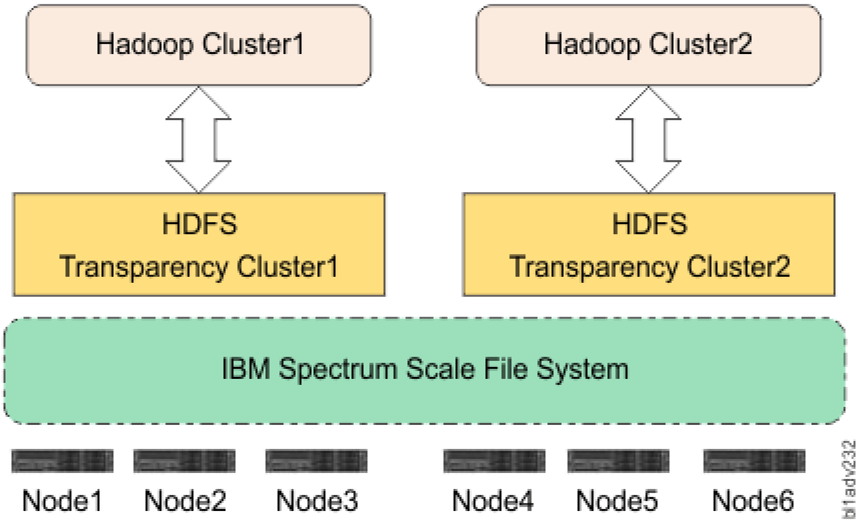

Hadoop Cluster1
Hadoop Cluster2
HDFS Transparency Cluster1
HDFS Transparency Cluster2
IBM Spectrum Scale File System
Node1
Node2
Node3
Node4
Node5
Node6
bl1adv232

Figure 19. Two Hadoop Clusters over the same IBM Storage Scale file system

You can configure Node1 to Node6 as an IBM Storage Scale cluster (FPO or shared storage mode). Then configure Node1 to Node3 as one HDFS transparency cluster and Node4 to Node6 as another HDFS transparency cluster. HDFS transparency cluster1 and HDFS transparency cluster2 take different configurations by changing `/usr/lpp/mmfs/hadoop/etc/hadoop/gpfs-site.xml` (for HDFS Transparency 2.7.3-x) or `/var/mmfs/hadoop/etc/hadoop/gpfs-site.xml` (for HDFS Transparency 3.0.x):

1. Change the `gpfs-site.xml` for HDFS transparency cluster1 to store the data under `/<gpfs-mount-point>/<hadoop1>` (**`gpfs.data.dir`**=*hadoop1* in `gpfs-site.xml`).
2. Run `mmhadoopctl connector syncconf /usr/lpp/mmfs/hadoop/etc/hadoop` (for HDFS Transparency 2.7.3-x) or `mmhadoopctl connector syncconf /var/mmfs/hadoop/etc/hadoop` (for HDFS Transparency 3.0.x) to synchronize the `gpfs-site.xml` from Step1 to all other nodes in HDFS transparency cluster1.
3. Change the `gpfs-site.xml` for HDFS transparency cluster2 to store the data under `/<gpfs-mount-point>/<hadoop2>` (**`gpfs.data.dir`**=*hadoop2* in `gpfs-site.xml`).
4. Run `mmhadoopctl connector syncconf /usr/lpp/mmfs/hadoop/etc/hadoop` (for HDFS Transparency 2.7.3-x) or `mmhadoopctl connector syncconf /var/mmfs/hadoop/etc/hadoop` (for HDFS Transparency 3.0.0) to synchronize the `gpfs-site.xml` from Step3 to all other nodes in HDFS transparency cluster2.
5. Restart the HDFS transparency services.

Automatic Configuration Refresh

The Automatic configuration refresh feature is supported in `gpfs.hdfs-protocol` 2.7.0-2 and later.

After making configuration changes in `/usr/lpp/mmfs/hadoop/etc/hadoop` (for HDFS Transparency 2.7.3-x) or `/var/mmfs/hadoop/etc/hadoop` (for HDFS Transparency 3.0.0) or in the IBM Storage Scale file system, such as maximum number of replica and NSD server, run the following command to refresh HDFS transparency without restarting the HDFS transparency services:

```
/usr/lpp/mmfs/hadoop/bin/gpfs dfsadmin -refresh
<namenode_hostname>:<port> refreshGPFSConfig
```

Run the command on any HDFS transparency node and change *<namenode_hostname>:<port>* according to the HDFS transparency configuration. For example, if *fs.defaultFS* is `hdfs://c8f2n03.gpfs.net:8020` in `/usr/lpp/mmfs/hadoop/etc/hadoop/core-site.xml`, replace *<namenode_hostname>* with `c8f2n03.gpfs.net` and *<port>* with `8020`. HDFS transparency synchronizes the configuration changes with the HDFS transparency services running on the HDFS transparency nodes and makes it immediately effective.

Rack locality support for shared storage

HDFS Transparency 2.7.2-0, rack locality is supported for shared storage including IBM ESS.

If your cluster meets the following conditions, you can enable this feature:

- There is more than one rack in the IBM Storage Scale cluster.
- Each rack has its own ToR (Top of Rack) Ethernet switch and there are rack-to-rack switches between the two racks.

Otherwise, enabling this feature will not benefit your Hadoop applications. The key advantage of the feature is to reduce the network traffic over the rack-to-rack Ethernet switch and make as many map/reduce tasks as possible to read data from the local rack.

The typical topology is shown by the following figure:

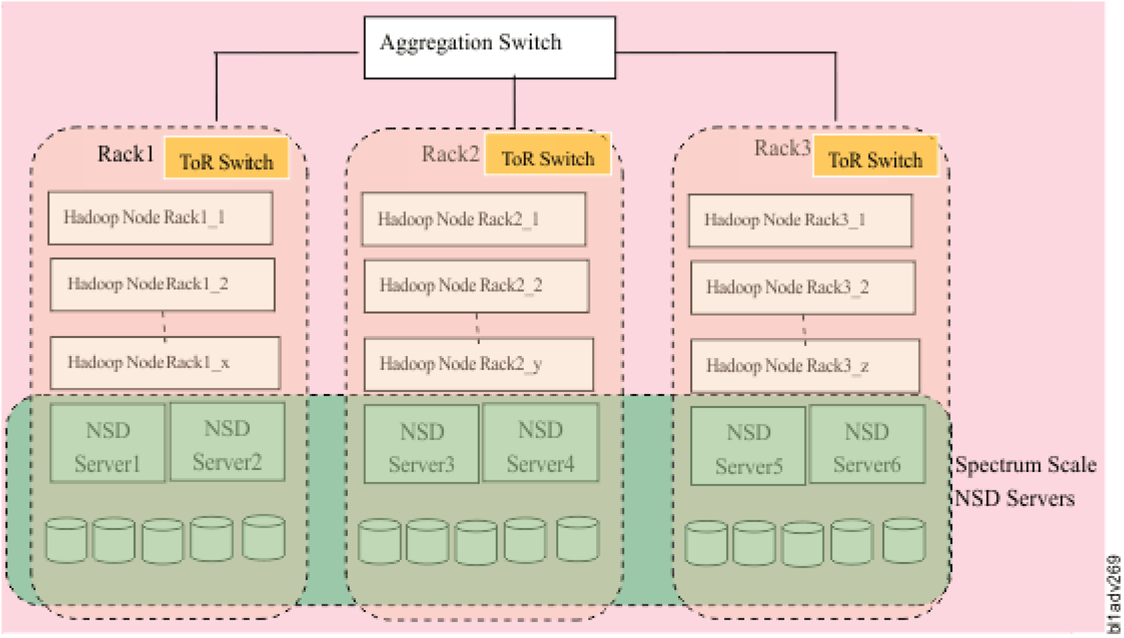


Figure 20. Topology of rack awareness locality for shared storage

For IBM Storage Scale over shared storage or IBM ESS, there is no data locality in the file system. The maximal file system block size from IBM Storage Scale file system is 16M bytes. However, on the Hadoop level, the **`dfs.blocksize`** is 128M bytes by default. The **`dfs.blocksize`** on the Hadoop level will be split into multiple 16MB blocks stored on the IBM Storage Scale file system. After enabling this feature, HDFS Transparency will consider the location of 8 blocks (16Mbytes * 8 = 128M bytes) including replica (if you take replica 2 for your file system) and will return the hostname with most of the data from the blocks to the applications so that the application can read most of the data from the local rack to reduce the rack-to-rack switch traffic. If there are more than one HDFS Transparency DataNodes in the selected rack, HDFS Transparency randomly returns one of them as the DataNode of the block location for that replica.

Enabling rack-awareness locality for shared storage

1. Select the HDFS Transparency nodes from the Hadoop node in Figure 20 on page 208. You can select all of the Hadoop nodes as the HDFS Transparency nodes, or part of them as the HDFS Transparency nodes.

   All of the selected HDFS Transparency nodes must be installed with IBM Storage Scale and can mount the file system locally. Select at least one of the Hadoop node from each of the rack for HDFS Transparency.

   Select all Hadoop Yarn Node Managers as the HDFS Transparency nodes to avoid data transfer delays from the HDFS Transparency node to the Yarn Node Manager node for Map/Reduce jobs.

2. On the HDFS Transparency NameNode, modify the `/usr/lpp/mmfs/hadoop/etc/hadoop/core-site.xml` (for HDFS Transparency 2.7.x) or `/var/mmfs/hadoop/etc/hadoop/core-site.xml` (for HDFS Transparency 3.0.x):

   ```
   <property>
       <name>net.topology.table.file.name</name>
       <value>/usr/lpp/mmfs/hadoop/etc/hadoop/topology.data</value>
     </property>
     <property>
       <name>net.topology.node.switch.mapping.impl</name>
       <value>org.apache.hadoop.net.TableMapping</value>
     </property>
   ```

3. On the HDFS Transparency NameNode, create the topology in `/usr/lpp/mmfs/hadoop/etc/hadoop/topology.data` (for HDFS Transparency2.7.x) or `/var/mmfs/hadoop/etc/hadoop/topology.data` (for HDFS Transparency 3.0.x):

   ```
   # vim topology.data

   192.0.2.0         /dc1/rack1
   192.0.2.1         /dc1/rack1
   192.0.2.2          /dc1/rack1
   192.0.2.3          /dc1/rack1
   192.0.2.4           /dc1/rack2
   192.0.2.5           /dc1/rack2
   192.0.2.6           /dc1/rack2
   192.0.2.7          /dc1/rack2
   ```

   Note: The topology.data file uses IP addresses. To configure two IP addresses, see the "Dual network interfaces" on page 21 section. The IP addresses here must be the IP addresses used for Yarn services and the IBM Storage Scale NSD server.

   Also, it is required to specify the IP addresses for the IBM Storage Scale NSD servers. For Figure 20 on page 208, specify the IP and corresponding rack information for NSD Server 1/2/3/4/5/6.

4. On the HDFS Transparency NameNode, modify the `/usr/lpp/mmfs/hadoop/etc/hadoop/gpfs-site.xml` (for HDFS Transparency 2.7.x) or `/var/mmfs/hadoop/etc/hadoop/gpfs-site.xml` (for HDFS Transparency 3.0.x):

   ```
     <property>
       <name>gpfs.storage.type</name>
       <value>rackaware</value>
     </property>
   ```

5. On the HDFS Transparency NameNode, run the `mmhadoopctl connector syncconf /usr/lpp/mmfs/hadoop/etc/hadoop` (for HDFS Transparency 2.7.x) or `mmhadoopctl connector syncconf /var/mmfs/hadoop/etc/hadoop` (for HDFS Transparency 3.0.x) command to synchronize the configurations to all the HDFS Transparency nodes.

   Note: If you have HDP with Ambari Mpack 2.4.2.1 and later, the **`connector syncconf`** cannot be executed. Ambari manages the configuration syncing through the database.

6. **(optional)**: To configure multi-cluster between IBM Storage Scale NSD servers and an IBM Storage Scale HDFS Transparency cluster, you must configure password-less access from the HDFS Transparency NameNode to at least one of the contact nodes from the remote cluster. For 2.7.3-2, HDFS Transparency supports only the root password-less ssh access. From 2.7.3-3, support of non-root password-less ssh access is added.

If password-less ssh access configuration cannot be set up, starting from HDFS transparency 2.7.3-2, you can configure **`gpfs.remotecluster.autorefresh`** as *false* in the `/usr/lpp/mmfs/hadoop/etc/hadoop/gpfs-site.xml`. This prevents Transparency from automatically accessing the remote cluster to retrieve information.

a. If you are using Ambari, add the **`gpfs.remotecluster.autorefresh=false`** field in **IBM Spectrum Scale service** > **Configs tab** > **Advanced** > **Custom gpfs-site**.

b. Stop and Start all the services.

c. Manually generate the mapping files and copy them to all the HDFS Transparency nodes. For more information, see option 3 under the "Password-less ssh access" on page 53 section.

Accumulo support

Native HDFS and HDFS Transparency

Apache Accumulo is fully tested over HDFS Transparency. See the Installing Apache Accumulo for Accumulo configuration information.

The Hadoop community addressed the NameNode bottleneck issue with the HDFS federation section that allows a DataNode to serve up blocks for multiple NameNodes. Additionally, **`ViewFS`** allows clients to communicate with multiple NameNodes by using a client-side mount table.

Multi-Volume support (MVS™), included in 1.6.0, includes the changes that allow Accumulo to work across multiple clusters such as Native HDFS and IBM Storage Scale HDFS Transparency (called volumes in Accumulo) while you continue to use a single HDFS directory. A new property, **`instance.volumes`**, can be configured with multiple HDFS nameservices. Accumulo uses them to balance out the NameNode operations.

You can include multiple NameNode namespaces into Accumulo for greater scalability of Accumulo instances by using federation.

Federation ViewFS has its own configuration settings to put in `core-site.xml` and `hdfs-site.xml`. You must also specify the namespaces in Accumulo configuration that has its setting in `$ACCUMULO_HOME/conf/accumulo-site.xml`:

```
instance.volumes=hdfs://nn1:port1/path/accumulo/data1, hdfs://nn2:port2/path/accumulo/data2
instance.namespaces=hdfs://nn1:port1,hdfs://nn2:port2
```

Following is an example:

```
    <property>
      <name>instance.namespaces</name>
      <value>hdfs://c16f1n10.gpfs.net:8020,hdfs://c16f1n13.gpfs.net:8020</value>
    </property>

    <property>
      <name>instance.volumes</name>
      <value>hdfs://c16f1n10.gpfs.net:8020/apps/accumulo/data1,
             hdfs://c16f1n13.gpfs.net:8020/apps/accumulo/data2
      </value>
    </property>
```

The **`instance.volumes`** need to specify the separated namespace full path but not the `viewfs://` schema trace by the https://issues.apache.org/jira/browse/ACCUMULO-3006.

After you start the federated multiple clusters, start the accumulo service. Run **`accumulo init`** on the accumulo client during the accumulo start if the following error occurred.

```
2017-11-02 05:46:49,954 [fs.VolumeManagerImpl]
WARN : dfs.datanode.synconclose set to false in hdfs-site.xml:
data loss is possible on hard system reset or power loss
2017-11-02 05:46:49,955 [fs.VolumeManagerImpl] WARN : dfs.datanode.synconclose
set to false in hdfs-site.xml: data loss is possible on hard
```

```
system reset or power loss
2017-11-02 05:46:50,038 [zookeeper.ZooUtil] ERROR:
unable obtain instance id at hdfs://c16f1n13.gpfs.net:8020/apps/accumulo/data/instance_id
2017-11-02 05:46:50,039 [start.Main] ERROR: Thread
'org.apache.accumulo.server.util.ZooZap' died.
java.lang.RuntimeException: Accumulo not initialized, there is no instance
id at hdfs://c16f1n13.gpfs.net:8020/apps/accumulo/data/instance_id
    at org.apache.accumulo.core.zookeeper.ZooUtil.getInstanceIDFromHdfs(ZooUtil.java:66)
    at org.apache.accumulo.core.zookeeper.ZooUtil.getInstanceIDFromHdfs(ZooUtil.java:51)
    at
org.apache.accumulo.server.client.HdfsZooInstance._getInstanceID(HdfsZooInstance.java:137)
    at org.apache.accumulo.server.client.HdfsZooInstance.getInstanceID(HdfsZooInstance.java:121)
    at org.apache.accumulo.server.util.ZooZap.main(ZooZap.java:76)
    at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
    at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)
    at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
    at java.lang.reflect.Method.invoke(Method.java:498)
    at org.apache.accumulo.start.Main$2.run(Main.java:130)
    at java.lang.Thread.run(Thread.java:745)
```

After the Accumulo is configured correctly, run the following command to ensure that the multiple volumes set-up successfully.

```
$ accumulo  admin volumes --list
2017-11-07 22:40:09,043 [fs.VolumeManagerImpl] WARN : dfs.datanode.synconclose
set to false in hdfs-site.xml: data loss is possible on hard
system reset or power loss
2017-11-07 22:40:09,044 [fs.VolumeManagerImpl] WARN : dfs.datanode.synconclose
set to false in hdfs-site.xml: data loss is possible on hard
system reset or power loss
Listing volumes referenced in zookeeper
    Volume : hdfs://c16f1n13.gpfs.net:8020/apps/accumulo/data2

Listing volumes referenced in accumulo.root tablets section
    Volume : hdfs://c16f1n10.gpfs.net:8020/apps/accumulo/data1
    Volume : hdfs://c16f1n13.gpfs.net:8020/apps/accumulo/data2
Listing volumes referenced in accumulo.root deletes section (volume replacement occurs at
deletion time)
    Volume : hdfs://c16f1n10.gpfs.net:8020/apps/accumulo/data1
    Volume : hdfs://c16f1n13.gpfs.net:8020/apps/accumulo/data2

Listing volumes referenced in accumulo.metadata tablets section
    Volume : hdfs://c16f1n10.gpfs.net:8020/apps/accumulo/data1
    Volume : hdfs://c16f1n13.gpfs.net:8020/apps/accumulo/data2
Listing volumes referenced in accumulo.metadata deletes section (volume replacement occurs at
deletion time)
    Volume : hdfs://c16f1n10.gpfs.net:8020/apps/accumulo/data1
```

Special configuration on IBM Storage Scale

By default, the property **tserver.wal.blocksize** is not configured and its default value is 0. Accumulo will calculate the block size accordingly and set the block size of the file in the distributed file system. For IBM Storage Scale, the valid block size could only be integral multiple of 64KB, 128KB, 256KB, 512KB, 1MB, 2MB, 4MB, 8MB and 16MB. Otherwise, HDFS Transparency will throw an exception.

To avoid this exception, configure **tserver.wal.blocksize** as the file system data block size. Use the **mmlspool <fs-name> all -L** command to check the value.

Zero shuffle support

Zero Shuffle is the ability for the map tasks to write data into the file system and the reduce tasks read data from the file system directly without doing the data transfers between the map tasks and reduce tasks first.

Do not use this feature if you are using IBM Storage Scale FPO mode (internal disk-based deployment). For IBM ESS or SAN-based storage, the recommendation is to take local disks on the computing nodes to store the intermediate shuffle data.

Zero shuffle should be used only for IBM ESS or SAN-based customers who cannot have local disks available for shuffle. For these customers, the previous solution is to store the shuffle data in IBM Storage Scale file system with replica 1. If you are taking zero shuffle, the Map/Reduce jobs will store shuffled data into IBM Storage Scale file system and read them directly during the reduce phase. This is supported from HDFS Transparency 2.7.3-2.

To enable zero shuffle, you need to configure the following values for `mapred-site.xml` from the Ambari GUI:

Configuration	Value
`mapreduce.task.io.sort.mb`	*<=1024*
`mapreduce.map.speculative`	*false*
`mapreduce.reduce.speculative`	*false*
`mapreduce.job.map.output.collector.class`	*org.apache.hadoop.mapred.SharedFsPlugins$MapOutputBuffer*
`mapreduce.job.reduce.shuffle.consumer.plugin.class`	*org.apache.hadoop.mapred.SharedFsPlugins$Shuffle*

Also, enable short circuit read for HDFS from the Ambari GUI.

If you take open source Apache Hadoop, you should put the `/usr/lpp/mmfs/hadoop/share/hadoop/hdfs/hadoop-hdfs-<version>.jar` (for HDFS Transparency 2.7.3-x) or `/usr/lpp/mmfs/hadoop/share/hadoop/hdfs/hadoop-hdfs-client-<version>.jar` (for HDFS Transparency 3.0.x) into your mapreduce class path.

Important:

- Zero shuffle does not impact teragen-like workloads because this kind of workloads do not involve using shuffle.
- **`mapreduce.task.io.sort.mb`** should be *<=1024*. Therefore, the data size for each map task must not be larger than 1024MB.
- Zero shuffle creates one file from each map task for each reduce task. Assuming your job has 1000 map tasks and 300 reduce tasks, it will create at least 300K intermediate files. Considering spilling, it might create around one million intermediate inodes and remove them after the job is done. Therefore, if the *reduce-task-number*map-task-number* is more than 300,000, it is not recommended to use zero shuffle.

Troubleshooting

Consult solutions or workarounds for HDFS Transparency protocol issues, learn about limitations and differences among HDFS distributions, or see limitations and recommendations for CES HDFS.

HDFS Transparency protocol troubleshooting

This topic contains information on troubleshooting the Second generation HDFS Transparency protocol issues.

Note:

- For HDFS Transparency 3.1.0 and earlier, use the **`mmhadoopctl`** command.
- For CES HDFS (HDFS Transparency 3.1.1 and later), use the corresponding **`mmhdfs`** and **`mmces`** commands.
- **`gpfs.snap --hadoop`** is used for all HDFS Transparency versions.
- From HDFS Transparency 3.1.0-6 and 3.1.1-3, ensure that the **`gpfs.ranger.enabled`** field is set to *scale*. The `scale` option replaces the original *true/false* values.

1. Enable Debugging

   Gather the following for problem determination:

- IBM Storage Scale and HDFS Transparency version
- NameNodes (Primary & HA), DataNodes and application service logs
- **gpfs.snap** with the **--hadoop [-a]** option (The **--hadoop** option is only available for IBM Storage Scale version 4.2.2 and later). For more information, see the *Data gathered for hadoop on Linux* topic under **Troubleshooting** > **Collecting details of the issues** > **CLI commands for collecting issue details** > **Using the gpfs.snap command** > **Data gathered by gpfs.snap on Linux for protocols** path in IBM Storage Scale documentation.

  Note:

  - The **gpfs.snap --hadoop** captures only the logs in the default log settings as specified in the *Data gathered for hadoop on Linux* topic in the *IBM Storage Scale: Problem Determination Guide*.
  - From IBM Storage Scale 5.0.5, **gpfs.snap --hadoop** is able to capture the HDFS Transparency logs from the user configured directories.

To enable the debug information for HDFS Transparency version 3.0.0-x/2.7.3-x/2.7.2-x, set the following fields to DEBUG as seen in the Fields box below into the `log4j.properties` file.

If you are using Ambari and you want to enable the log fields, add the fields through the Ambari GUI HDFS service and restart the HDFS service.

If you are not using Ambari and you want to enable the log fields, change the fields in the `<GPFS_CONFIG_PATH>/log4j.properties` file and run the `/usr/lpp/mmfs/bin/mmhadoop connector syncconf <GPFS_CONFIG_PATH>` command and then restart the HDFS transparency. Restart the HDFS Transparency by running the following commands:

```
/usr/lpp/mmfs/bin/mmhadoopctl connector stop;
/usr/lpp/mmfs/bin/mmhadoopctl connector start
```

For HDFS Transparency 2.7.3-x, the **<GPFS_CONFIG_PATH>** is `mmhadoopctl connector syncconf /usr/lpp/mmfs/hadoop/etc/hadoop`.

For HDFS Transparency 3.0.x, the **<GPFS_CONFIG_PATH>** is `mmhadoopctl connector syncconf /var/mmfs/hadoop/etc/hadoop`.

Log and configuration location:

For HDFS Transparency version 3.0.0-x:

- With Hortonworks HDP 3.0, the HDFS Transparency logs are located at `/var/log/hadoop/root` by default.
- For Open Source Apache, the HDFS Transparency logs are located at `/var/log/transparency`.
- Configuration is moved from `/usr/lpp/mmfs/hadoop/etc` to `/var/mmfs/hadoop/etc`.

For HDFS Transparency version 2.7.3-x with HortonWorks HDP 2.6.x or BI IOP 4.2.5, the HDFS Transparency logs are located at `/var/log/hadoop/root` by default.

For HDFS Transparency version 2.7.2-x with IBM BigInsights IOP 4.0/4.1/4.2.x, the HDFS Transparency logs are located at `/usr/lpp/mmfs/hadoop/logs` by default.

For HDFS Transparency version 2.7.x-x with Open Source Apache, the HDFS Transparency logs are located at `/usr/lpp/mmfs/hadoop/logs`.

Fields:

```
log4j.logger.BlockStateChange=DEBUG

log4j.logger.org.apache.hadoop.hdfs.StateChange=DEBUG

log4j.logger.org.apache.hadoop.hdfs.server.namenode.GPFSNative=DEBUG

log4j.logger.org.apache.hadoop.hdfs.server.namenode=DEBUG

log4j.logger.org.apache.hadoop.hdfs.protocol.datatransfer=DEBUG

log4j.logger.org.apache.hadoop.hdfs.server.namenode.top.metrics=ERROR
```

```
log4j.logger.org.apache.hadoop.hdfs.server.namenode.top.window=ERROR

log4j.logger.org.apache.hadoop.hdfs.server.blockmanagement.DatanodeManager=INFO

log4j.logger.org.apache.hadoop.hdfs.server.blockmanagement.BlockPlacementPolicyDefault=INFO

log4j.logger.org.apache.hadoop.hdfs.server.datanode.fsdataset.impl.FsDatasetImpl=DEBUG

log4j.logger.org.apache.hadoop.hdfs.server=DEBUG

log4j.logger.org.apache.hadoop.hdfs.DFSClient=DEBUG

log4j.logger.org.apache.hadoop.ipc=DEBUG

log4j.logger.org.apache.hadoop.fs=DEBUG

log4j.logger.org.apache.hadoop.conf.Configuration.deprecation=WARN
```

The logs are located at `/usr/lpp/mmfs/hadoop/logs`.

To enable the debug information for HDFS Transparency version 2.7.0-x, set the following fields in the `/usr/lpp/mmfs/hadoop/etc/hadoop/log4j.properties` file to DEBUG:

```
log4j.logger.org.apache.hadoop.gpfs.server=DEBUG

log4j.logger.org.apache.hadoop.ipc=DEBUG

log4j.logger.org.apache.hadoop.hdfs.protocol=DEBUG

log4j.logger.org.apache.hadoop.hdfs.protocol.datatransfer.DataTransferProtocol=DEBUG

log4j.logger.org.apache.hadoop.hdfs.StateChange=DEBUG
```

Dynamically changing debug log settings

Find the active NameNode and port to set the daemon logging level dynamically. The setting will be in effect until HDFS restarts or you reset the Debug Log Level to INFO.

Run the following command:

```
hadoop daemonlog -setlevel <Active Namenode>:<Active Namenode port> <Daemon to set Log
level> <Debug Log Level>
```

For example:

- To get the debug level:

  ```
  # hadoop daemonlog -getlevel c902f08x01.gpfs.net:50070
  org.apache.hadoop.hdfs.server.namenode
  ```

  Connecting to *http://c902f08x01.gpfs.net:50070/logLevel?log=org.apache.hadoop.hdfs.server.namenode*.

  Submitted Class Name: `org.apache.hadoop.hdfs.server.namenode`

  Log Class: `org.apache.commons.logging.impl.Log4JLogger`

  Effective Level: INFO

- To set the debug level:

  ```
  # hadoop daemonlog -setlevel c902f08x01.gpfs.net:50070
  org.apache.hadoop.hdfs.server.namenode DEBUG
  ```

  Connecting to *http://c902f08x01.gpfs.net:50070/logLevel?log=org.apache.hadoop.hdfs.server.namenode&level=DEBUG*.

  Submitted Class Name: `org.apache.hadoop.hdfs.server.namenode`

  Log Class: `org.apache.commons.logging.impl.Log4JLogger`

  Submitted Level: DEBUG

Setting Level to DEBUG ...

Effective Level: DEBUG

Get jps and jstack information

Run jps to get the PID of the daemon and run jstack PID of daemon and pipe to a file to get the output.

For example:

```
# jps | grep NameNode
15548 NameNode

# jstack -l 15548 > jstack_15548_NameNode.output
```

Debug Time issues

Set the debug flags from DEBUG to ALL. Do not change the other flags.

Debugging HDFS NameNode using failover

If you cannot stop NameNode in a running cluster and you are not able to dynamically change the debug setting for the NameNode in the cluster due to the security settings, then change the xml file and manually perform the NameNode failover.

For example, the security settings were set:

```
hadoop.http.authentication.simple.anonymous.allowed=true
hadoop.http.authentication.type=simple
```

Note: Manually editing the xml files in HDP with Mpack environment will be void after HDFS restarts because Ambari saves the configuration in the database.

For example, during Ranger debugging:

- Checked if user group is in **`dfs.permissions.superusergroup`**
- Checked **`xasecure.add-hadoop-authorization`** = *true*
- Checked **`dfs.namenode.inode.attributes.provider.class`** = *org.apache.ranger.authorization.hadoop.RangerHdfsAuthorizer* and **`dfs.permissions.enabled`** = *true*
- GPFS directory is set to 700
- Not able to dynamically set the debug command:

  ```
  hadoop daemonlog -setlevel [active namenode]:50070
  org.apache.ranger.authorization.hadoop.RangerHdfsAuthorizer DEBUG
  ```

Edit `/var/mmfs/hadoop/etc/hadoop/log4j.properties` to add in the DEBUG flag on the standby NameNode.

```
log4j.logger.org.apache.hadoop.security.UserGroupInformation=DEBUG
log4j.logger.org.apache.ranger.authorization.hadoop.RangerHdfsAuthorizer=DEBUG
```

#On the standby NameNode, stop

```
/usr/lpp/mmfs/hadoop/bin/hdfs --config /var/mmfs/hadoop/etc/hadoop --daemon stop namenode
```

#Check is stopped

```
jps
```

#Start standby NameNode to pick up the debug changes

```
/usr/lpp/mmfs/hadoop/bin/hdfs --config /var/mmfs/hadoop/etc/hadoop --daemon start namenode
```

#Failover primary NameNode to standby

```
hdfs haadmin -failover nn2 nn1
```

#Recreate issue and look into the NameNode log

```
2020-02-17 07:47:06,331 ERROR util.RangerRESTClient ….
```

2. A lot of "topN size for comm" in NameNode logs

   The NameNode log contains a lot of topN size entries as shown below:

   ```
   2016-11-20 10:02:27,259 INFO  window.RollingWindowManager
   (RollingWindowManager.java:getTopUsersForMetric(247)) - topN size for command rename is: 0

   2016-11-20 10:02:27,259 INFO  window.RollingWindowManager
   (RollingWindowManager.java:getTopUsersForMetric(247)) - topN size for command mkdirs is: 1

   ......
   ```

   Solution:

   In the `log4j.properties` file, set the following field:

   ```
   log4j.logger.org.apache.hadoop.hdfs.server.namenode.top.window=WARN
   ```

3. **`webhdfs`** is not working

   Solution:

   On the node running the NameNode service, check whether the port (50070 by default) is up. If it is up, check if the **`dfs.webhdfs.enabled`** is set to *true* in your configuration. If not, configure it to *true* in your hadoop configuration and sync it to the HDFS transparency nodes.

4. Could not find or load main class `org.apache.hadoop.gpfs.tools.GetConf`

   Solution:

   Check all the nodes to see whether any of the bash variables (HADOOP_HOME, HADOOP_HDFS_HOME, HADOOP_MAPRED_HOME, HADOOP_COMMON_HOME, HADOOP_COMMON_LIB_NATIVE_DIR, HADOOP_CONF_DIR, HADOOP_SECURITY_CONF_DIR) were exported. If it was exported, unexport or change it to another name. Setting these variables will result in the failure of some of the HDFS Transparency commands.

5. `org.apache.hadoop.hdfs.server.namenode.GPFSRunTimeException: java.io.IOException`: Invalid argument: anonymous

   If **`hive.server2.authentication`** is configured as LDAP or Kerberos enabled, then the anonymous user is not used by Hive. However, the default setting for **`hive.server2.authentication`** is set to NONE. Therefore, no authentication will be done for Hive's requests to the Hiveserver2 (meta data). This means that all the requests will be done as the anonymous user.

   This exception is only seen when you run HIVE (HIVE supports anonymous authentication):

   ```
   2016-08-09 22:53:07,782 WARN  ipc.Server (Server.java:run(2068)) -
   IPC Server handler 143 on 8020, call org.apache.hadoop.hdfs.protocol.ClientProtocol.mkdirs
   from 192.0.2.10:37501 Call#19 Retry#0

   org.apache.hadoop.hdfs.server.namenode.GPFSRunTimeException: java.io.IOException: Invalid argument: anonymous

           at org.apache.hadoop.hdfs.server.namenode.SerialNumberManager.getUserSerialNumber(SerialNumberManager.java:59)

           at org.apache.hadoop.hdfs.server.namenode.INodeWithAdditionalFields$PermissionStatusFormat.toLong(INodeWithAdditionalFields.java:64)

           at org.apache.hadoop.hdfs.server.namenode.INodeWithAdditionalFields.<init>(INodeWithAdditionalFields.java:116)

           at org.apache.hadoop.hdfs.server.namenode.INodeDirectory.<init>(INodeDirectory.java:77)

           at org.apache.hadoop.hdfs.server.namenode.FSDirMkdirOp.unprotectedMkdir(FSDirMkdirOp.java:234)

           at org.apache.hadoop.hdfs.server.namenode.FSDirMkdirOp.createSingleDirectory(FSDirMkdirOp.java:191)

           at org.apache.hadoop.hdfs.server.namenode.FSDirMkdirOp.createChildrenDirectories(FSDirMkdirOp.java:166)

           at org.apache.hadoop.hdfs.server.namenode.FSDirMkdirOp.mkdirs(FSDirMkdirOp.java:97)

           at org.apache.hadoop.hdfs.server.namenode.FSNamesystem.mkdirs(FSNamesystem.java:3928)

           at org.apache.hadoop.hdfs.server.namenode.GPFSNamesystem.mkdirs(GPFSNamesystem.java:1254)

           at org.apache.hadoop.hdfs.server.namenode.NameNodeRpcServer.mkdirs(NameNodeRpcServer.java:993)

           at
   org.apache.hadoop.hdfs.protocolPB.ClientNamenodeProtocolServerSideTranslatorPB.mkdirs(ClientNamenodeProtocolServerSideTranslatorPB.java:622)

           at
   ```

```
org.apache.hadoop.hdfs.protocol.proto.ClientNamenodeProtocolProtos$ClientNamenodeProtocol$2.callBlockingMethod(ClientNamenodeProtocolProtos.java)
        at org.apache.hadoop.ipc.ProtobufRpcEngine$Server$ProtoBufRpcInvoker.call(ProtobufRpcEngine.java:616)
        at org.apache.hadoop.ipc.RPC$Server.call(RPC.java:969)
        at org.apache.hadoop.ipc.Server$Handler$1.run(Server.java:2049)
        at org.apache.hadoop.ipc.Server$Handler$1.run(Server.java:2045)
        at java.security.AccessController.doPrivileged(Native Method)
        at javax.security.auth.Subject.doAs(Subject.java:422)
        at org.apache.hadoop.security.UserGroupInformation.doAs(UserGroupInformation.java:1657)
        at org.apache.hadoop.ipc.Server$Handler.run(Server.java:2043)
Caused by: java.io.IOException: Invalid argument: anonymous
        at org.apache.hadoop.hdfs.server.namenode.GPFSNative.getUid(Native Method)
        at org.apache.hadoop.hdfs.server.namenode.SerialNumberManager.getUserSerialNumber(SerialNumberManager.java:51)
        ... 20 more
```

Solutions:

There are two solutions to fix this issue:

a. Create the user and group "anonymous" with the same uid/gid on all the nodes.
b. Configure HIVE hive.server2.authentication as LDAP or Kerberos assuming cluster is already LDAP/Kerberos enabled.

6. **`javax.security.sasl.SaslException`** when Kerberos is enabled

```
'''

16/09/21 01:40:30 WARN ipc.Client: Exception encountered while connecting to the server :
javax.security.sasl.SaslException: GSS initiate failed [Caused by GSSException: No valid
credentials provided (Mechanism level: Failed to find any Kerberos tgt)]


Operation failed: Failed on local exception: java.io.IOException:
javax.security.sasl.SaslException:
 GSS initiate failed [Caused by GSSException: No valid credentials provided (Mechanism
level:
Failed to find any Kerberos tgt)]; Host Details : local host is: "c8f2n13.gpfs.net/
192.0.2.11";
destination host is: "c8f2n13.gpfs.net":8020;

'''

'''

16/09/21 01:42:20 WARN ipc.Client: Exception encountered while connecting to the server :
javax.security.sasl.SaslException: GSS initiate failed [Caused by GSSException:
No valid credentials provided (Mechanism level: Failed to find any Kerberos tgt)]

'''
```

Solution:

For IBM BigInsights IOP 4.0/4.1/4.2, run the `kinit -kt /etc/security/keytabs/nn.service.keytab nn/c8f2n13.gpfs.net@IBM.COM` command.

Note: Replace the hostname `c8f2n13.gpfs.net` with the node on which you will run the knit command.

For other Hadoop distro, you must check the configuration **`dfs.namenode.kerberos.principal`** value.

7. NameNode failed to start because the null pointer was encountered when SSL was configured with HDFS Transparency version 2.7.2-0.

The NameNode will fail to start when SSL is configured because a null pointer exception is encountered due to missing ssl files for HDFS Transparency version 2.7.2-0.

Hadoop NameNode log:

```
STARTUP_MSG: Starting NameNode
.....
2016-11-29 18:51:45,572 INFO  namenode.NameNode (LogAdapter.java:info(47)) -
registered UNIX signal handlers for [TERM, HUP, INT]

2016-11-29 18:51:45,575 INFO  namenode.NameNode (NameNode.java:createNameNode(1438)) -
createNameNode []
```

```
...

2016-11-29 18:51:46,417 INFO  http.HttpServer2 (NameNodeHttpServer.java:initWebHdfs(86)) -
Added filter 'org.apache.hadoop.hdfs.web.AuthFilter' (class=org.apache.hadoop.hdfs.web.AuthFilter)

2016-11-29 18:51:46,418 INFO  http.HttpServer2 (HttpServer2.java:addJerseyResourcePackage(609)) -
addJerseyResourcePackage:
packageName=org.apache.hadoop.hdfs.server.namenode.web.resources;org.apache.hadoop.hdfs.web.resources,
pathSpec=/webhdfs/v1/*

2016-11-29 18:51:46,434 INFO  http.HttpServer2 (HttpServer2.java:openListeners(915)) -
Jetty bound to port 50070

2016-11-29 18:51:46,462 WARN  mortbay.log (Slf4jLog.java:warn(76)) -
java.lang.NullPointerException

2016-11-29 18:51:46,462 INFO  http.HttpServer2 (HttpServer2.java:start(859)) -
HttpServer.start() threw a non Bind IOException

java.io.IOException: !JsseListener: java.lang.NullPointerException

at org.mortbay.jetty.security.SslSocketConnector.newServerSocket(SslSocketConnector.java:516)

at org.apache.hadoop.security.ssl.SslSocketConnectorSecure.newServerSocket(SslSocketConnectorSecure.java:46)

at org.mortbay.jetty.bio.SocketConnector.open(SocketConnector.java:73)

at org.apache.hadoop.http.HttpServer2.openListeners(HttpServer2.java:914)

at org.apache.hadoop.http.HttpServer2.start(HttpServer2.java:856)

at org.apache.hadoop.hdfs.server.namenode.NameNodeHttpServer.start(NameNodeHttpServer.java:142)

at org.apache.hadoop.hdfs.server.namenode.NameNode.startHttpServer(NameNode.java:773)

at org.apache.hadoop.hdfs.server.namenode.NameNode.initialize(NameNode.java:647)

at org.apache.hadoop.hdfs.server.namenode.NameNode.<init>(NameNode.java:832)

at org.apache.hadoop.hdfs.server.namenode.NameNode.<init>(NameNode.java:816)

at org.apache.hadoop.hdfs.server.namenode.NameNode.createNameNode(NameNode.java:1509)

at org.apache.hadoop.hdfs.server.namenode.NameNode.main(NameNode.java:1575)

2016-11-29 18:51:46,465 INFO  impl.MetricsSystemImpl (MetricsSystemImpl.java:stop(211)) -
Stopping NameNode metrics system...

2016-11-29 18:51:46,466 INFO  impl.MetricsSinkAdapter (MetricsSinkAdapter.java:publishMetricsFromQueue(141)) -
timeline thread interrupted.

2016-11-29 18:51:46,466 INFO  impl.MetricsSystemImpl (MetricsSystemImpl.java:stop(217)) -
NameNode metrics system stopped.

2016-11-29 18:51:46,466 INFO  impl.MetricsSystemImpl (MetricsSystemImpl.java:shutdown(607)) -
NameNode metrics system shutdown complete.

2016-11-29 18:51:46,466 ERROR namenode.NameNode (NameNode.java:main(1580)) - Failed to start namenode.

java.io.IOException: !JsseListener: java.lang.NullPointerException

at org.mortbay.jetty.security.SslSocketConnector.newServerSocket(SslSocketConnector.java:516)

at org.apache.hadoop.security.ssl.SslSocketConnectorSecure.newServerSocket(SslSocketConnectorSecure.java:46)

at org.mortbay.jetty.bio.SocketConnector.open(SocketConnector.java:73)

at org.apache.hadoop.http.HttpServer2.openListeners(HttpServer2.java:914)

at org.apache.hadoop.http.HttpServer2.start(HttpServer2.java:856)

at org.apache.hadoop.hdfs.server.namenode.NameNodeHttpServer.start(NameNodeHttpServer.java:142)

at org.apache.hadoop.hdfs.server.namenode.NameNode.startHttpServer(NameNode.java:773)

at org.apache.hadoop.hdfs.server.namenode.NameNode.initialize(NameNode.java:647)

at org.apache.hadoop.hdfs.server.namenode.NameNode.<init>(NameNode.java:832)

at org.apache.hadoop.hdfs.server.namenode.NameNode.<init>(NameNode.java:816)

at org.apache.hadoop.hdfs.server.namenode.NameNode.createNameNode(NameNode.java:1509)

at org.apache.hadoop.hdfs.server.namenode.NameNode.main(NameNode.java:1575)

2016-11-29 18:51:46,468 INFO  util.ExitUtil (ExitUtil.java:terminate(124)) - Exiting with status 1

2016-11-29 18:51:46,469 INFO  namenode.NameNode (LogAdapter.java:info(47)) - SHUTDOWN_MSG:

/************************************************************

SHUTDOWN_MSG: Shutting down NameNode at <NameNodeHost>/<IPADDRESS>

************************************************************/
```

Solution:

The workaround solution for HDFS Transparency version 2.7.2-0 with SSL configured is to copy the `/etc/hadoop/conf/ssl-client.xml` and the `/etc/hadoop/conf/ssl-server.xml` files into `/usr/lpp/mmfs/hadoop/etc/hadoop` on all the nodes.

8. `org.apache.hadoop.ipc.RemoteException(java.io.IOException): blocksize(xxxxx)` should be an integral mutiple of `dataBlockSize(yyyyy)`

   HDFS Transparency + IBM Storage Scale, the blocksize of file system could only be 64KB, 128KB, 256KB, 512KB, 1MB, 2MB, 4MB, 8MB and 16MB. Therefore, the `dataBlockSize(yyyyy)` must be the blocksize of your IBM Storage Scale file system. If your hadoop workloads take blocksize which is not integral multiple of dataBlockSize, you will see this issue. Typically, such as Accumulo:

   Accumulo TServer failed to start with the below error.

   ```
   Caused by: org.apache.hadoop.ipc.RemoteException(java.io.IOException): blocksize(1181115904) should be an integral mutiple of 
   dataBlockSize(1048576)

           at org.apache.hadoop.hdfs.server.namenode.GPFSDetails.verifyBlockSize(GPFSDetails.java:230)

           at org.apache.hadoop.hdfs.server.namenode.GPFSNamesystem.startFile(GPFSNamesystem.java:254)

           at org.apache.hadoop.hdfs.server.namenode.NameNodeRpcServer.create(NameNodeRpcServer.java:632)

           at 
   org.apache.hadoop.hdfs.protocolPB.ClientNamenodeProtocolServerSideTranslatorPB.create(ClientNamenodeProtocolServerSideTranslatorPB.java:397)

           at 
   org.apache.hadoop.hdfs.protocol.proto.ClientNamenodeProtocolProtos$ClientNamenodeProtocol$2.callBlockingMethod(ClientNamenodeProtocolProtos.java)

           at org.apache.hadoop.ipc.ProtobufRpcEngine$Server$ProtoBufRpcInvoker.call(ProtobufRpcEngine.java:616)

           at org.apache.hadoop.ipc.RPC$Server.call(RPC.java:969)

           at org.apache.hadoop.ipc.Server$Handler$1.run(Server.java:2049)

           at org.apache.hadoop.ipc.Server$Handler$1.run(Server.java:2045)

           at java.security.AccessController.doPrivileged(Native Method)

           at javax.security.auth.Subject.doAs(Subject.java:422)
   ```

   Solution:

   For Accumulo, add the configuration in **Accumulo** > **Configs** > **Custom accumulo-site** and set **`tserver.wal.blocksize`** = `<value-of-your-gpfs-filesystem>` and then restart the service.

9. Got exception: `java.io.IOException No FileSystem for scheme: hdfs`

   Running a service failed.

   The service log shows `Got exception: java.io.IOException No FileSystem for scheme: hdfs` message.

   The issue might occur because of a Maven-assembly issue (refer to hadoop No FileSystem for scheme: file) with duplicated file system in the classpath.

   Solution:

   If this exception is seen, add **`fs.hdfs.impl`** = *org.apache.hadoop.hdfs.DistributedFileSystem* into the core-site.xml to resolve the issue.

10. java.io.IOException: Couldn't clear parent znode /hadoop-ha/<HA-cluster-ID> when NameNode HA is enabled.

    This exception can be seen when you start the HDFS Transparency NameNode through the Ambari GUI or from the command line which indicates that there are some files or directories under `/hadoop-ha/<HA-cluster-ID>` that are preventing the NameNode from starting up.

    Solution:

    Manually remove the directory by running `<zookeeper-home-dir>/bin/zkCli.sh -server <one-zookeeper-server-hostname>:<network-port-number> rmr /hadoop-ha` from any one Hadoop node. For example, the command will be **`/usr/iop/4.2.0.0/zookeeper/bin/zkCli.sh -server c16f1n02:2181 rmr /hadoop-ha`** for IOP 4.2 with hostname c16f1n02 and default zookeeper network port number 2181.

11. datanode.DataNode (BPServiceActor.java:run(840)) - ClusterIds are not matched, existing.

    If a DataNode did not come up and the DataNode's log (located under `/usr/lpp/mmfs/hadoop/logs`) contains the above error message, then the HDFS Transparency DataNode might have been at one time used for a different HDFS Transparency cluster.

    Solution:

Run the **/usr/lpp/mmfs/hadoop/sbin/initmap.sh <your-file-system> diskmap nodemap clusterinfo** command on the node and try to start it again. The command will update the following three files and will place the node into the current HDFS Transparency cluster:

- /var/mmfs/etc/clusterinfo4hdfs
- /var/mmfs/etc/diskid2hostname
- /var/mmfs/etc/nodeid2hostname

12. All blocks to be queried must be in the same block pool

    When you run workloads (such as Impala or IBM BigSQL), you might hit the following exception:

    ```
    All blocks to be queried must be in the same block pool:
    BP-gpfs_data,/gpfs_data/CDH5/user/root/POC2/2014-07.txt:blk_318148_0 and
    LocatedBlock{BP-gpfs_data,/gpfs_data/CDH5/user/root/POC2/2014-09.txt:blk_318150_0;
    getBlockSize()=134217728; corrupt=false; offset=0;
    locs=[DatanodeInfoWithStorage[192.0.2.12:50010,,DISK]]} are from different pools.
    ```

    Solution:

    Change the **dfs.datanode.hdfs-blocks-metadata.enabled** field to *false* in hdfs-site.xml and restart HDFS Transparency and Impala/BigSQL.

13. Exception: Failed to load the configurations of Core GPFS (/<gpfs mount point>)

    When you start the HDFS Transparency NameNode or DataNode, the daemon is not up and you get the following exceptions:

    ```
    Exception: Failed to load the configurations of Core GPFS (/<gpfs mount point>) :
    org.apache.hadoop.hdfs.server.namenode.GPFSDetails.refreshCoreGPFSConfig()
    ```

    Solution:

    Check the /var/mmfs/etc/clusterinfo4hdfs, /var/mmfs/etc/diskid2hostname and /var/mmfs/etc/nodeid2hostname on all the Transparency nodes. If they are of 0 size, run /usr/lpp/mmfs/hadoop/sbin/initmap.sh <your-file-system-name> diskmap nodemap clusterinfo on all the Transparency nodes.

14. UID/GID failed with illegal value “Illagal value: USER = xxxxx > MAX = 8388607"

    Solution:

    If you have installed Ranger and need to leverage Ranger capabilities, you need to make the UID/GID less than 8388607.

    If you do not need Ranger, then follow these steps to disable Ranger from HDFS Transparency:

    a. Stop HDFS Transparency

       ```
       mmhadoopctl connector stop -N all
       ```

    b. On the NameNode, set the **gpfs.ranger.enabled** = *false* in /usr/lpp/mmfs/hadoop/etc/hadoop/gpfs-site.xml

       ```
       <property>
         <name>gpfs.ranger.enabled</name>
         <value>false</value>
       </property>
       ```

    c. Sync the HDFS Transparency configuration

       ```
       mmhadoopctl connector syncconf /usr/lpp/mmfs/hadoop/etc/hadoop
       ```

    d. Start HDFS Transparency

       ```
       mmhadoopctl connector start -N all
       ```

       Note: For HDFS Transparency 2.7.3-3 and later, when Ranger is enabled, uid greater than 8388607 is supported.

15. Issues that can be encountered if the IBM Storage Scale ACL is not set properly

    a. **hadoop fs -ls** command failed with `Operation not supported` message.

       Even when the Hadoop command failed, the POSIX command can be executed successfully.

       ```
       # hadoop fs -ls /
       ```

       ```
       ls: Operation not supported: /bigpfs/mapred
       ```

    b. **hdfs dfs -mkdir** and **hdfs dfs -ls** commands are not working after installation.

       The **hdfs dfs -mkdir** command fails with `NullPointerExecption` and the **hdfs dfs -ls /** command fails with `No such file or directory` message.

       ```
       # /usr/lpp/mmfs/hadoop/bin/hdfs dfs -mkdir -p /user
       ```

       ```
       mkdir: java.lang.NullPointerException
       ```

       ```
       # /usr/lpp/mmfs/hadoop/bin/hdfs dfs -ls /
       ```

       ```
       ls: `/': No such file or directory
       ```

       Solution:

       These issues occur when the ACL mode on the IBM Storage Scale file system is set to **nfs4** instead of **all**.

       If IBM Storage Scale CES (SMB and NFS) is used, ACL semantics are required to be set to **nfsv4**.

       Note: When HDFS is used do not set the ACL semantics to **nfsv4** because HDFS and IBM Storage Scale HDFS Transparency support only POSIX ACLs. Therefore, the **-k** option can be set to value `posix` or `all`.

       To display the type of authorization currently set on the file system, issue the following command:

       ```
       # /usr/lpp/mmfs/bin/mmlsfs bdafs -k
       flag                  value                    description
       ------------------- ------------------------ -----------------------------------
       -k                     nfs4                       ACL semantics in effect
       ```

       If the value is **nfs4**, change it to **all**.

       To change from **nfs4** to **all**, issue the following command:

       ```
       # /usr/lpp/mmfs/bin/mmlsfs bdafs -k
       flag                value                    description
       ------------------- ------------------------ -----------------------------------
       -k                   all                         ACL semantics in effect
       ```

       Note: `-k all` means that any supported ACL type is permitted. This includes traditional GPFS (posix) and NFS V4 and Windows ACLs (nfsv4).

       When you are in a HDFS Transparency environment, ensure that the **-k** value is set to `all`.

       Restart HDFS Transparency after ACL is set to ALL.

16. User permission denied when Ranger is disabled

    If Kerberos is enabled and Ranger flag is disabled, then the user might get permission denied errors when accessing the file system.

    Solution:

    Check the Kerberos principal mapping **hadoop.security.auth_to_local** field in `/usr/lpp/mmfs/hadoop/etc/hadoop/core-site.xml` or in Ambari under HDFS Config to ensure that the NameNode and DataNode are mapped to root instead of hdfs.

    For example, change

    From:

```
RULE:[2:$1@$0](dn@COMPANY.DIV.COM)s/.*/hdfs/
RULE:[2:$1@$0](nn@COMPANY.DIV.COM)s/.*/hdfs/
```

To:

```
RULE:[2:$1@$0](dn@COMPANY.DIV.COM)s/.*/root/
RULE:[2:$1@$0](nn@COMPANY.DIV.COM)s/.*/root/
```

Then restart the HDFS service in Ambari or HDFS Transparency by executing the following commands:

```
/usr/lpp/mmfs/bin/mmhadoopctl connector stop

/usr/lpp/mmfs/bin/mmhadoopctl connector start
```

17. IBM Storage Scale NSD is not able to be recovered in FPO.

    In an FPO environment, the IBM Storage Scale NSD is not able to be recovered after the node got expelled or the NSD went down and the auto recovery failed.

    Note: This can also occur while performing a STOP ALL/Start ALL from Ambari.

    If you see the `Attention: Due to an earlier configuration change the file system is no longer properly replicated` message on executing the **`mmlsdisk <fs>`** command, **`mmrestripefs -R`** command is needed to re-replicate the files after all the disks are brought back.

    IBM Storage Scale will also do a special check of file system recovery log file. To guarantee the consistency of the file system, if the number of failure groups with down disk is greater than or equal to the current log file replication, IBM Storage Scale will also panic the file system.

    Solution:

    a. Run **`mmrestripefs <fs> -R`** to restore all the files (including FS recovery log files) to their designated degree of replication as they were not properly replicated when there were too many down disks. This will also help to avoid the re-occurrence that FS panic on one single metadata disk failure.
    b. After the command completes its execution, run the following command to double check the log file's replica restored to the expected value 3. (FPO replica is set to 3 usually)

       ```
       # mmdsh -N all mmfsadm dump log | egrep "Dump of LogFile|nReplicas"
       ```

       Dump of LogFile at 0x3FFEE900A990 stripe group mygpfs state 'open for append' type 'striped': nReplicas 3 logGroupInconsistent 0 whichPartial 0 whichDesc 0

       Note:

       - The example shows that the nReplicas is set to 3.
       - Upgrade to GPFS 5.0.1.2 or later. In the newer release, GPFS will not trigger **`mmrestripefs -r`** during the auto recovery if there is not enough FGs to satisfy the needed replica. This can help in avoiding the same issue as here. A side effect is it will not try the auto-recovery (includes the attempts to start disk) if not enough FGs are left. Therefore, if there are disk down after many NSD servers restart, you might need to manually run the **`mmchdisk start`** to start the disks when the NSD servers are back.
       - Do not execute **`mmstripefs -r`** manually if there are not enough FG left to satisfy the desired replica. If you have already executed it, then run **`mmrestripefs -R`** after the down disks are brought back up.

    To avoid this issue, follow the FPO Maintenance procedure.

    If you are using Ambari, do not perform a STOP ALL, START ALL or shutdown/restart of the IBM Storage Scale service from the Ambari GUI.

    See the following topics in the IBM Storage Scale KC under the *File Placement Optimizer* subsection under the *Administering* section, to perform the maintenance of FPO cluster. Temporarily disable auto recovery before the planned maintenance. Especially do not let the disks fail and the automatic recovery gets initiate.

- *Restarting a large IBM Storage Scale cluster*
- *Upgrade FPO*
- *Rolling upgrade*
- *Handling disk failures*
- *Handling node failures*

18. DataNode reports exceptions after Kerberos is enabled on RHEL7.5

    If your Kerberos KDC or your Kerberos client are on RHEL7.5 at version 1.15.1-18 (default version shipped in RHEL7.5), you might hit the following exceptions when you start the DataNodes:

    ```
    2018-08-06 14:10:56,812 WARN  ipc.Client (Client.java:run(711)) -
    Couldn't setup connection for dn/140.bd@EXAMPLE.COM to 139.bd/**.**.**.**:8020
    javax.security.sasl.SaslException: GSS initiate failed [Caused by GSSException:
    No valid credentials provided (Mechanism level: Ticket expired (32) - PROCESS_TGS)]
        at com.sun.security.sasl.gsskerb.GssKrb5Client.evaluateChallenge(GssKrb5Client.java:211)
    ```

    Solution:

    Upgrade the Kerberos KDC (krb5-libs and krb5-server) from version 1.15.1-18 to version 1.15.1-19 or upgrade the Kerberos client (krb5-workstation and krb5-libs) from version 1.15.1-18 to version 1.15.1-19.

19. NameNode fails to start - Cannot find **`org.apache.ranger.authorization.hadoop.RangerHdfsAuthorizer`**

    With the Ranger enabled, the NameNode fails to start and HDFS Transparency shows the `cannot find org.apache.ranger.authorization.hadoop.RangerHdfsAuthorizer` error.

    Solution:

    The `hadoop-env.sh` file did not have the proper Ranger setup. See "Apache Ranger" on page 138 on how to add the Apache Ranger files properly within the hadoop-env.sh file as well as ensure that the 4 ranger-*.xml files are copied to the proper HDFS Transparency directory.

    If you are using Ambari, restart the HDFS service. Otherwise, restart HDFS Transparency by executing the **`/usr/lpp/mmfs/bin/mmhadoopctl connector stop/start`** command.

    For example:

    The `/usr/hdp/current/ranger-hdfs-plugin` path was not used and the `/usr/hdp/<your-HDP-version>/ranger-hdfs-plugin` path was used. Therefore, set up the hadoop-env.sh file to use the path in your environment.

20. Read from HDFS interface hangs (for example, `hadoop dfs -get`, `hadoop dfs -copyToLocal`)

    When you are reading data from the HDFS interface, the `read` command will hang. If you are using Ambari, the service checks from the Ambari GUI will be timed out.

    Debug output from the command shows timeout after 60s:

    ```
    DEBUG hdfs.DFSClient: Connecting to datanode x.x.x.x:50010
    DEBUG sasl.SaslDataTransferClient: SASL encryption trust check:
    localHostTrusted = false, remoteHostTrusted = false
    DEBUG sasl.SaslDataTransferClient: SASL client skipping handshake in
    unsecured configuration for addr = /x.x.x.x, datanodeId = DatanodeInfoWithStorage[xxxxx]
    WARN hdfs.DFSClient: Exception while reading from .....

    java.net.SocketTimeoutException: 60000 millis timeout while waiting for channel
    to be ready for read. ch : java.nio.channels.SocketChannel[connected local=/x.x.x.x:37058
    remote=/x.x.x.x:50010]
            at org.apache.hadoop.net.SocketIOWithTimeout.doIO(SocketIOWithTimeout.java:164)
    ```

However, when the HDFS Transparency debug is set, no exceptions are seen in the logs. Increasing the timeout value of the HDFS client socket does not resolve the issue.

Note: Writing in the HDFS interface works. The issue is only while reading from the HDFS interface.

Solution:

Configure **`dfs.datanode.transferTo.allowed`** as *false* in the `hdfs-site.xml` and restart HDFS Transparency. When **`dfs.datanode.transferTo.allowed`** is *true* by default, some transfers on the socket might hang on some platforms (OS/JVM).

If you are using Ambari, ensure that you have set **`dfs.datanode.transferTo.allowed`** to *false* in the HDFS service configuration. If the field does not exist, add a new field and restart the HDFS service.

21. Hive LLAP queries failed

    By default, LLAP uses inode paths to access data from the file system using the `hive.llap.io.user.fileid.path=true` setting which will not work on the IBM Storage Scale file system.

    Solution:

    Configure `hive.llap.io.use.fileid.path=false` to have LLAP access the file from file path instead of inode number.

    If you are using Ambari, then go to **Ambari GUI** > **Hive** > **Configs - Advanced Tab**.

    a. Search for **`hive.llap.io.use.fileid.path`** field.
    b. If you find it and it is not set to *false*, change it to *false*. If you do not find it, add `hive.llap.io.use.fileid.path=false` under Custom hive-site Add Property.
    c. Save configuration and restart Hive.

22. DataNode failed with error `Failed to refresh configuration`

    The DataNode will not be able to come up if the `initmap.sh` internal generated configuration files are stale or incorrect.

    The following error can be seen:

```
 ERROR datanode.DataNode (DataNode.java:secureMain(2922)) - Exception in secureMain
java.io.IOException: Failed to refresh configurations
       at org.apache.hadoop.hdfs.server.namenode.GPFSDetails.refreshCoreGPFSConfig(GPFSDetails.java:613)
       at org.apache.hadoop.hdfs.server.namenode.GPFSDetails.init(GPFSDetails.java:175)
       at org.apache.hadoop.hdfs.server.datanode.DataNode.<init>(DataNode.java:477)
       at org.apache.hadoop.hdfs.server.datanode.DataNode.makeInstance(DataNode.java:2821)
       at org.apache.hadoop.hdfs.server.datanode.DataNode.instantiateDataNode(DataNode.java:2713)
       at org.apache.hadoop.hdfs.server.datanode.DataNode.createDataNode(DataNode.java:2766)
       at org.apache.hadoop.hdfs.server.datanode.DataNode.secureMain(DataNode.java:2915)
       at org.apache.hadoop.hdfs.server.datanode.DataNode.main(DataNode.java:2939)
Caused by: java.io.IOException: /var/mmfs/etc/hadoop/clusterinfo4hdfs.fpo is outdated because initmap.sh
failed probably.
       at org.apache.hadoop.hdfs.server.namenode.GPFSFs.collectClusterInfo(GPFSFs.java:414)
       at org.apache.hadoop.hdfs.server.namenode.GPFSFs.collectInfo(GPFSFs.java:551)
       at org.apache.hadoop.hdfs.server.namenode.GPFSDetails.refreshCoreGPFSConfig(GPFSDetails.java:608)
       ... 7 more
```

    Solution:

    The DataNode internal configuration files need to be regenerated. If possible, restart the NameNode, or restart the Standby NameNode if HA is configured, or touch those internal configuration files and then restart the DataNode from the Ambari GUI or from the command line.

    For more information, see "Cluster and file system information configuration" on page 62.

23. Viewfs `hadoop dfs -copyFromLocal -l` fails

    With ViewFs configuration, running the `copyFromLocal -l` will generate failure.

```
$ hadoop fs -copyFromLocal -f -l /etc/hosts
/TestDir/Test_copyFromLocal.1545110085.28040/copyFromLocal -copyFromLocal: Fatal internal error

org.apache.hadoop.fs.viewfs.NotInMountpointException: getDefaultBlockSize on empty path is invalid
at org.apache.hadoop.fs.viewfs.ViewFileSystem.getDefaultBlockSize(ViewFileSystem.java:695)
at org.apache.hadoop.fs.FilterFileSystem.getDefaultBlockSize(FilterFileSystem.java:420)
```

```
at
org.apache.hadoop.fs.shell.CommandWithDestination$TargetFileSystem.create(CommandWithDestination.java:505
)
at
org.apache.hadoop.fs.shell.CommandWithDestination$TargetFileSystem.writeStreamToFile(CommandWithDestinati
on.java:484)
at org.apache.hadoop.fs.shell.CommandWithDestination.copyStreamToTarget(CommandWithDestination.java:407)
at org.apache.hadoop.fs.shell.CommandWithDestination.copyFileToTarget(CommandWithDestination.java:342)
at org.apache.hadoop.fs.shell.CopyCommands$CopyFromLocal.copyFile(CopyCommands.java:357)
at org.apache.hadoop.fs.shell.CopyCommands$CopyFromLocal.copyFileToTarget(CopyCommands.java:365)
at org.apache.hadoop.fs.shell.CommandWithDestination.processPath(CommandWithDestination.java:277)
at org.apache.hadoop.fs.shell.CommandWithDestination.processPath(CommandWithDestination.java:262)
at org.apache.hadoop.fs.shell.Command.processPathInternal(Command.java:367)
at org.apache.hadoop.fs.shell.Command.processPaths(Command.java:331)
at org.apache.hadoop.fs.shell.Command.processPathArgument(Command.java:304)
at org.apache.hadoop.fs.shell.CommandWithDestination.processPathArgument(CommandWithDestination.java:257)
at org.apache.hadoop.fs.shell.Command.processArgument(Command.java:286)
at org.apache.hadoop.fs.shell.Command.processArguments(Command.java:270)
at org.apache.hadoop.fs.shell.CommandWithDestination.processArguments(CommandWithDestination.java:228)
at org.apache.hadoop.fs.shell.CopyCommands$Put.processArguments(CopyCommands.java:295)
at org.apache.hadoop.fs.shell.CopyCommands$CopyFromLocal.processArguments(CopyCommands.java:385)
at org.apache.hadoop.fs.shell.FsCommand.processRawArguments(FsCommand.java:120)
at org.apache.hadoop.fs.shell.Command.run(Command.java:177)
at org.apache.hadoop.fs.FsShell.run(FsShell.java:328)
at org.apache.hadoop.util.ToolRunner.run(ToolRunner.java:76)
at org.apache.hadoop.util.ToolRunner.run(ToolRunner.java:90)
at org.apache.hadoop.fs.FsShell.main(FsShell.java:391)
```

Solution:

HDP SPEC-56 has a fix For IBM to distribute to customer. The fix is to replace the jar files in the environment.

Contact IBM service if you want the fix for SPEC-56/RTC Defect 20924.

24. `numNode=0` due to time synchronization issue

    When numNode is 0, the number of DataNode available for use is 0.

    The NameNode Log will show the following error:

    ```
    Caused by: org.apache.hadoop.hdfs.server.namenode.GPFSRunTimeException:
    110259f803894787b8d364019aa4106f fileid=259093 block=blk_259093_0, numNodes=0
      at
    org.apache.hadoop.hdfs.server.blockmanagement.GPFSBlockManager.getStorages(GPFSBlockManager.java:102)
      at
    org.apache.hadoop.hdfs.server.blockmanagement.GPFSBlockManager.createLocatedBlock(GPFSBlockManager.java:2
    20)
    ```

    Solution:

    Check if the Scale cluster time are in sync.

    Run: `mmdsh -N all "date +%m%d:%H%M%S-%s"`

    If not, ensure that the time are in sync.

25. How to enable audit logging in HDFS Transparency

    Solution:

    a. Manually enable the audit log by editing the `hadoop-env.sh` file.

       - Edit the `/var/mmfs/hadoop/etc/hadoop/ hadoop-env.sh` file to include the following entry:

         ```
         export HDFS_AUDIT_LOGGER=INFO,RFAAUDIT
         ```

       - Synchronize the configuration files by using the **mmhadoopctl** command (for HDFS Transparency 3.1.0-x and earlier) or upload the configuration files into CCR by using the **mmhdfs** command (for CES HDFS Transparency 3.1.1-x).
       - Restart HDFS Transparency.
       - Verify looking into the `${hadoop.log.dir}/hdfs-audit.log` file for audit information such as:

         ```
         INFO FSNamesystem.audit: allowed=true
         cmd=getfileinfo src=
         ```

```
cmd=listStatus  src
(auth:KERBEROS)
```

b. Manually enable the audit log by editing the `log4j.properties` file.

   - Edit the `/var/mmfs/hadoop/etc/hadoop/log4j.properties` file to change the following entry:

     From

     ```
     hdfs.audit.logger=INFO,NullAppender
     ```

     To

     ```
     hdfs.audit.logger=INFO,RFAAUDIT
     ```

   - Synchronize the configuration files by using the **`mmhadoopctl`** command (for HDFS Transparency 3.10-x and earlier) or upload the configuration files into CCR by using the **`mmhdfs`** command (for CES HDFS Transparency 3.1.1-x).
   - Restart HDFS Transparency.
   - Verify by looking into the `${hadoop.log.dir}/hdfs-audit.log` file for the audit information such as:

     ```
     INFO FSNamesystem.audit: allowed=true
     cmd=getfileinfo src=
     cmd=listStatus  src
     (auth:KERBEROS)
     ```

c. If you are using Ambari, then go to **HDFS service** > **Configs** > **Advanced tab** > **Advanced hdfs-log4j** and set the hdfs.audit.logger=INFO,RFFAAUDIT, save the configuration and restart the HDFS service.

   Advanced hdfs-log4j

   ```
   #
   # hdfs audit logging
   #
   hdfs.audit.logger=INFO,RFAAUDIT
   log4j.logger.org.apache.hadoop.hdfs.server.namenode.FSNamesystem.audit=$
   {hdfs.audit.logger}
   log4j.additivity.org.apache.hadoop.hdfs.server.namenode.FSNamesystem.audit=false
   ```

26. Ulimit issues

a. Java exception: All datanodes DatanodeInfoWithStorage are bad error

   Solution:

   If you see similar exception errors of `java.io.IOException: All datanodes DatanodeInfoWithStorage are` bad when running the map/reduce jobs, you must increase your **`ulimit -n`** and **`ulimit -u`** to *64K*.

   ```
   15/12/30 07:09:16 INFO mapreduce.Job: Task Id : attempt_1450797405784_0281_m_000005_0, Status : FAILED

   c902f10x09: Error: java.io.IOException: All datanodes DatanodeInfoWithStorage
   [192.0.2.13:50010,DS-1ca06221-194c-47d2-82d0-8b602a64921b,DISK] are bad. Aborting...

   c902f10x09:     at
   org.apache.hadoop.hdfs.DFSOutputStream$DataStreamer.setupPipelineForAppendOrRecovery(DFSOutputStream.java:1218)

   c902f10x09:     at org.apache.hadoop.hdfs.DFSOutputStream$DataStreamer.processDatanodeError(DFSOutputStream.java:1016)

   c902f10x09:     at org.apache.hadoop.hdfs.DFSOutputStream$DataStreamer.run(DFSOutputStream.java:560)
   ```

b. Job getting IOException BlockMissingException could not obtain block error and cannot read the file using HDFS until HDFS Transparency restarts. The file can be read using POSIX.

   Error:

   ```
   org.apache.hadoop.hive.ql.metadata.HiveException: java.io.IOException:
   org.apache.hadoop.hdfs.BlockMissingException: Could not obtain block:
   ...
   ```

```
Caused by: org.apache.hadoop.hdfs.BlockMissingException: Could not obtain block
...
```

Solution

Check the DataNode for errors:

```
IOException "don't match block error", then search previous messages for that
block blk_<value> under sees "FileNotFoundException" and "Too many open files" error
previously, then this means the DataNode does not have enough open files limit value.

2020-03-17 12:22:46,638 INFO  datanode.DataNode (DataXceiver.java:writeBlock(1023))
- opWriteBlock BP-XXXXXXXX:blk_5079204_0 received exception
java.io.FileNotFoundException: /dev/null (Too many open files)

2020-03-17 13:06:50,384 ERROR datanode.DataNode (DataXceiver.java:run(326)) -
<HOST>.com:1019:DataXceiver error processing READ_BLOCK operation  src: /<IP>:49771
dst: /<IP>:1019
java.io.IOException:  Offset 0 and length 60986628 don't match block BP-
XXXXXXXX:blk_5079204_0 ( blockLen 0 )
  at org.apache.hadoop.hdfs.server.datanode.BlockSender.<init>(BlockSender.java:429)
  at org.apache.hadoop.hdfs.server.datanode.DataXceiver.readBlock(DataXceiver.java:684)
  at org.apache.hadoop.hdfs.protocol.datatransfer.Receiver.opReadBlock(Receiver.java:163)
  at org.apache.hadoop.hdfs.protocol.datatransfer.Receiver.processOp(Receiver.java:111)
  at org.apache.hadoop.hdfs.server.datanode.DataXceiver.run(DataXceiver.java:293)
  at java.lang.Thread.run(Thread.java:748)
```

The issue occurs when there are not enough open file limit set. The `FileNotFoundException` exception is thrown and the internal data structure is left in an uninitiated state. The block length is not initialized to the correct value and is still at 0. Therefore, when the block was read later, the sanity check failed in HDFS Transparency.

To fix this issue, increase the open file limit.

To increase the open file limit, see "OS tuning for all nodes in HDFS Transparency" on page 55.

27. HDFS Transparency fails to start if the Java version is upgraded.

    When you are starting HDFS Transparency in CES HDFS or non-CES HDFS you get the following error:

    `ERROR: JAVA_HOME /usr/lib/jvm/java-1.8.0-openjdk-1.8.0.242.b08-0.el7_7.x86_64 does not exist.`

    Solution

    If the Java version was upgraded, or if a kernel patch or OS upgrade, upgraded the Java version, then the Java path is changed to the updated version path. Therefore, the JAVA_HOME setting for the user profile (.bashrc) and the JAVA_HOME in `/var/mmfs/hadoop/etc/hadoop/hadoop-env.sh` need to be updated to reflect the updated JAVA directory.

    The updated Java directory can be seen under `/etc/alternatives/java`.

    ```
    # ls -l /etc/alternatives/jre lrwxrwxrwx 1 root root 62 Jan 21 10:05 /etc/alternatives/jre
    ->
    /usr/lib/jvm/java-1.8.0-openjdk-1.8.0.252.b09-2.el7_8.x86_64/jre
    ```

    This path has to be set in `/var/mmfs/hadoop/etc/hadoop/hadoop-env.sh` and then the user profile (.bashrc):

    ```
    export JAVA_HOME=/usr/lib/jvm/java-1.8.0-openjdk-1.8.0.252.b09-2.el7_8.x86_64/jre
    ```

    Instead of using the versioned path to the JRE, you can also use the version agnostic symbolic link that is automatically updated with every upgrade:

    ```
    export JAVA_HOME=/etc/alternatives/jre
    ```

28. In the NameNode log, when you are appending a file, you get **`NullPointerException`** on the file operations. The exception stack trace includes the following:

    ```
    java.lang.NullPointerException
            at
    ```

```
org.apache.hadoop.hdfs.server.blockmanagement.BlockPlacementPolicyDefault.chooseTarget()
        at
org.apache.hadoop.hdfs.server.blockmanagement.BlockPlacementPolicyDefault.chooseTarget()
        at
org.apache.hadoop.hdfs.server.blockmanagement.BlockManager.chooseTarget4AdditionalDatanode()
        at org.apache.hadoop.hdfs.server.namenode.GPFSNamesystemV0.getAdditionalDatanode()
        at org.apache.hadoop.hdfs.server.namenode.NameNodeRpcServer.getAdditionalDatanode()
        at
org.apache.hadoop.hdfs.protocolPB.ClientNamenodeProtocolServerSideTranslatorPB.getAdditional
Datanode()
(...)
```

Solution

The problem occurs when there are fewer DataNodes available than the value of the specified replication factor (**dfs.replication**). Add more DataNodes to the cluster or reduce the value of the **dfs.replication** parameter to resolve the error.

To get the existing replication value, run the following command:

```
mmhdfs config get hdfs-site.xml -k dfs.replication
```

The output for the above command is displayed as follows:

```
dfs.replication=3
```

To reduce the replication value, run the following command:

```
mmhdfs config set hdfs-site.xml -k dfs.replication=2
```

29. Write operations (put/copyFromLocal) fail with the following error message:

    ```
    File [FILENAME] could only be written to 0 of the X minReplication nodes.
    There are Y datanode(s) running and no node(s) are excluded in this
    operation.
    ```

    Solution

    The problem occurs when the **dfs.datanode.du.reserved** value is set too large. Ensure that it is set to the recommended value or reduce the value to resolve the error.

30. NameNode fails to start in HDFS Transparency 3.1.1-11, 3.1.1-12, 3.2.2-2 or 3.2.2-3 with the following stack trace:

    ```
    2023-01-25 17:58:22,926 ERROR namenode.NameNode (NameNode.java:main(1828)) - Failed to
    start namenode.
    java.lang.NoClassDefFoundError: org/codehaus/jackson/Versioned
    ...
    Caused by: java.lang.ClassNotFoundException: org.codehaus.jackson.Versioned
    ```

    Solution

    Add `jackson-core-asl-1.9.13.jar` from the Maven repository, or from HDFS Transparency version 3.1.1-10 or 3.2.2-1, to the following paths on every HDFS node in the cluster.

    a. `/usr/lpp/mmfs/hadoop/share/hadoop/hdfs/lib/`
    b. `/usr/lpp/mmfs/hadoop/share/hadoop/common/lib/`

    The `jackson-core-asl-1.9.13.jar` has an MD5 checksum of *319c49a4304e3fa9fe3cd8dcfc009d37*.

31. To list all the HDFS encryption zones in an IBM Storage Scale file system using the GPFS policy engine, use the following steps to create a policy rule that matches the encryption zone directory structure and prints the names of the directories containing the **system.raw.hdfs.crypto.encryption.zone** attribute. These steps are a workaround as the **hdfs crypto -listZones** command is not supported in HDFS Transparency.

    It is important to note that these steps should be executed on the node where the file system is locally mounted:

a. Create a policy rule file that matches the encryption zone directory structure (for example, `list_encryption_zones.pol`):

```
RULE 'dirsRule' LIST 'dirs'
DIRECTORIES_PLUS
SHOW(varchar(mode) || ' ' || varchar(XATTR('system.raw.hdfs.crypto.encryption.zone')))
where (XATTR('system.raw.hdfs.crypto.encryption.zone') IS NOT NULL)
```

In this rule, the **`system.raw.hdfs.crypto.encryption.zone`** extended attribute matches any directory in IBM Storage Scale to identify encryption zones.

b. To run this policy rule, save it to a file (follow step 1) and apply the policy to the file system using the following command:

```
#/usr/lpp/mmfs/bin/mmapplypolicy GPFS -P list_encryption_zones.pol -I defer -f /tmp/
encryption_zone
```

Note: It is important to keep in mind that the GPFS policy engine applies policies to the local file system on the node where the file system is mounted. To ensure accurate and consistent policy application, it is recommended to run policies on the node where the file system is locally mounted.

```
[root@]# cat list_encryption_zones.pol
RULE 'dirsRule' LIST 'dirs'
DIRECTORIES_PLUS
SHOW(varchar(mode) || ' ' || varchar(XATTR('system.raw.hdfs.crypto.encryption.zone')))
where (XATTR('system.raw.hdfs.crypto.encryption.zone') IS NOT NULL)

[root@]# mmapplypolicy GPFS -P list_encryption_zones.pol -I defer  -f /tmp/
encryption_zone
[I] GPFS Current Data Pool Utilization in KB and %
Pool_Name                  KB_Occupied           KB_Total     Percent_Occupied
system                        393849856         1992294400        19.768657484%
[I] 199168 of 307968 inodes used: 64.671654%.
[W] Attention: In RULE 'dirsRule' LIST name 'dirs' appears but there is no corresponding
"EXTERNAL LIST 'dirs' EXEC ... OPTS ..." rule to specify a program to process the
matching files.
[I] Loaded policy rules from list_encryption_zones.pol.
Evaluating policy rules with CURRENT_TIMESTAMP = 2023-05-04@12:07:36 UTC
Parsed 1 policy rules.
RULE 'dirsRule' LIST 'dirs'
DIRECTORIES_PLUS
SHOW(varchar(mode) || ' ' || varchar(XATTR('system.raw.hdfs.crypto.encryption.zone')))
where (XATTR('system.raw.hdfs.crypto.encryption.zone') IS NOT NULL)
[I] 2023-05-04@12:07:37.298 Directory entries scanned: 195159.
[I] Directories scan: 191829 files, 3294 directories, 36 other objects, 0 'skipped'
files and/or errors.
[I] 2023-05-04@12:07:39.739 Parallel-piped sort and policy evaluation. 195159 files
scanned.
[I] 2023-05-04@12:07:39.772 Piped sorting and candidate file choosing. 2 records scanned.
[I] Summary of Rule Applicability and File Choices:
Rule#        Hit_Cnt         KB_Hit         Chosen      KB_Chosen         KB_Ill    Rule
     0              2              0              2              0              0
RULE 'dirsRule' LIST 'dirs' DIRECTORIES_PLUS SHOW(.) WHERE(.)

[I] Filesystem objects with no applicable rules: 195138.

[I] GPFS Policy Decisions and File Choice Totals:
Chose to list 0KB: 2 of 2 candidates;
Predicted Data Pool Utilization in KB and %:
Pool_Name                  KB_Occupied           KB_Total     Percent_Occupied
system                        393879552         1992294400        19.770148026%
[I] 2023-05-04@12:07:39.776 Policy execution. 0 files dispatched.
[I] A total of 0 files have been migrated, deleted or processed by an EXTERNAL EXEC/
script;
    0 'skipped' files and/or errors.
```

Encryption zones directories list will be collected in `/tmp/encryption_zone.list.dirs` directory.

```
[root@]# cat /tmp/encryption_zone.list.dirs
10041 1253024653 0  drwxr-xr-xkey -- /gpfs/datadir_regr32-01/zone
179202 968077442 0  drwxr-xr-xkey -- /gpfs/datadir_regr32-01/test_zone
```

32. To address any failures that may occur during the installation or upgrade of HDFS versions 3.1.1-15 or 3.2.2-6, follow these steps before a retry.

    a. Execute one the following cleanup commands to remove the RPM package from all nodes.

       To remove `gpfs.hdfs-protocol-3.1.1-15.x86_64` (HDFS 3.1.1-15), issue the next command:

       ```
       mmdsh -N all "rpm -e gpfs.hdfs-protocol-3.1.1-15.x86_64"
       ```

       To remove `gpfs.hdfs-protocol-3.2.2-6.x86_64` (HDFS 3.2.2-6), issue the next command:

       ```
       mmdsh -N all "rpm -e gpfs.hdfs-protocol-3.2.2-6.x86_64"
       ```

    b. Delete the Hadoop directory at the specified location on all nodes by using the next command:

       ```
       mmdsh -N all "rm -rf /usr/lpp/mmfs/hadoop/share/hadoop"
       ```

33. Debug, trace, and logs.

    Solution:

    To check the state of the CES HDFS cluster, see the *mmhealth* command documentation in *IBM Storage Scale: Command and Programming Reference Guide* guide.

    To determine the status of the CES HDFS NameNodes state, run the following command:

    ```
    /usr/lpp/mmfs/hadoop/bin/hdfs haadmin -checkHealth -scale -all
    ```

    For more information, see the "hdfs haadmin" on page 241 command.

    For HDFS Transparency, see "HDFS Transparency protocol troubleshooting" on page 212 on how to Enable Debugging.

34. CES HDFS Transparency cluster failed to start.

    ```

    mmces service enable HDFS
    or
    mmces service start hdfs -a
    ```

    Solution:

    Note: Run `/usr/lpp/mmfs/hadoop/bin/hdfs namenode -initializeSharedEdits`, if the NameNode failed to start with the following exception:

    ```
    2019-11-22 01:02:01,925 ERROR namenode.FSNamesystem (FSNamesystem.java:<init>(911)) -
    GPFSNamesystem initialization failed.
    java.io.IOException: Invalid configuration: a shared edits dir must not be specified if HA
    is not enabled.
            at org.apache.hadoop.hdfs.server.namenode.FSNamesystem.<init>(FSNamesystem.java:789)
            at
    org.apache.hadoop.hdfs.server.namenode.GPFSNamesystemBase.<init>(GPFSNamesystemBase.java:49)
            at
    org.apache.hadoop.hdfs.server.namenode.GPFSNamesystem.<init>(GPFSNamesystem.java:74)
            at
    org.apache.hadoop.hdfs.server.namenode.FSNamesystem.loadFromDisk(FSNamesystem.java:706)
            at org.apache.hadoop.hdfs.server.namenode.NameNode.loadNamesystem(NameNode.java:669)
            at org.apache.hadoop.hdfs.server.namenode.NameNode.initialize(NameNode.java:731)
            at org.apache.hadoop.hdfs.server.namenode.NameNode.<init>(NameNode.java:968)
            at org.apache.hadoop.hdfs.server.namenode.NameNode.<init>(NameNode.java:947)
            at
    org.apache.hadoop.hdfs.server.namenode.NameNode.createNameNode(NameNode.java:1680)
            at org.apache.hadoop.hdfs.server.namenode.NameNode.main(NameNode.java:1747)
    ```

35. Mapreduce container job exit with return code 1.

    Solution:

If `Container exited with a non-zero exit code 1. Error file: prelaunch.err` occurred while running the Mapreduce workloads, add the following property into the `mapred-site.xml` to resolve the issue:

```
<property>
   <name>mapreduce.application.classpath</name>
   <value>/usr/hadoop-3.1.2/share/hadoop/mapreduce/*, /usr/hadoop-3.1.2/share/hadoop/
mapreduce/lib/*</value>
</property>
```

36. **`mmhdfs hdfs status`** shows node is not a DataNode.

    The command **`mmhdfs hdfs status`** shows the following errors:

    ```
    c16f1n13.gpfs.net:  This node is not a datanode
    mmdsh: c16f1n13.gpfs.net remote shell process had return code 1.
    ```

    Solution:

    Remove the localhost value from the host.

    On the worker node, run:

    ```
    mmhdfs worker remove localhost
    ```

37. All the NameNodes status shows standby after **`mmhdfs start/stop/restart`** commands.

    Solution:

    Use the **`mmces service`** command to start/stop NameNodes so that the proper state is reflected for the NameNodes.

    If the **`mmhdfs start/stop/restart`** command was executed against the NameNodes, run the **`mmces service start/stop hdfs`** to fix the issue.

38. **`hdfs dfs -ls`** or another operation fails with a StandbyException.

    Running the **`hdfs dfs -ls`** command fails with a StandbyException exception:

    ```
    [root@scale12 transparency]# /usr/lpp/mmfs/hadoop/bin/hdfs dfs -ls /HDFS
    2020-04-06 16:26:25,891 INFO retry.RetryInvocationHandler:
    org.apache.hadoop.ipc.RemoteException(org.apache.hadoop.ipc.StandbyException):
    Operation category READ is not supported in state standby. Visit https://s.apache.org/sbnn-error
    at org.apache.hadoop.hdfs.server.namenode.ha.StandbyState.checkOperation(StandbyState.java:88)
    at org.apache.hadoop.hdfs.server.namenode.NameNode$NameNodeHAContext.checkOperation(NameNode.java:2010)
    at org.apache.hadoop.hdfs.server.namenode.FSNamesystem.checkOperation(FSNamesystem.java:1447)
    at org.apache.hadoop.hdfs.server.namenode.FSNamesystem.getFileInfo(FSNamesystem.java:3129)
    at org.apache.hadoop.hdfs.server.namenode.GPFSNamesystem.getFileInfo(GPFSNamesystem.java:494)
    at org.apache.hadoop.hdfs.server.namenode.NameNodeRpcServer.getFileInfo(NameNodeRpcServer.java:1143)
    at
    org.apache.hadoop.hdfs.protocolPB.ClientNamenodeProtocolServerSideTranslatorPB.getFileInfo(ClientNamenode
    ProtocolServerSideTranslatorPB.java:939)
    at
    org.apache.hadoop.hdfs.protocol.proto.ClientNamenodeProtocolProtos$ClientNamenodeProtocol$2.callBlockingM
    ethod(ClientNamenodeProtocolProtos.java)
    at org.apache.hadoop.ipc.ProtobufRpcEngine$Server$ProtoBufRpcInvoker.call(ProtobufRpcEngine.java:523)
    at org.apache.hadoop.ipc.RPC$Server.call(RPC.java:991)
    at org.apache.hadoop.ipc.Server$RpcCall.run(Server.java:872)
    at org.apache.hadoop.ipc.Server$RpcCall.run(Server.java:818)
    at java.security.AccessController.doPrivileged(Native Method)
    at javax.security.auth.Subject.doAs(Subject.java:422)
    at org.apache.hadoop.security.UserGroupInformation.doAs(UserGroupInformation.java:1729)
    at org.apache.hadoop.ipc.Server$Handler.run(Server.java:2678)
    , while invoking ClientNamenodeProtocolTranslatorPB.getFileInfo over scale12/192.0.2.21:8020 after 1
    failover attempts. Trying to failover after sleeping for 1157ms.
    ^C2020-04-06 16:26:27,097 INFO retry.RetryInvocationHandler: java.io.IOException:
    The client is stopped, while invoking ClientNamenodeProtocolTranslatorPB.getFileInfo
    over scale11/192.0.2.20:8020 after 2 failover attempts. Trying to failover after
    sleeping for 2591ms.
    ```

    Both the NameNodes are in standby and the CES has failed to select one as active. To verify, run the following command:

    ```
    /usr/lpp/mmfs/hadoop/bin/hdfs haadmin -getAllServiceState
    ```

    `scale01:8020 standby scale02:8020 standby`

Solution:

a. Check the NameNode that should be active by running the following command:

```
/usr/lpp/mmfs/bin/mmhealth node show -v HDFS_NAMENODE -N cesNodes
```

b. For one of the nodes, the output shows the `hdfs_namenode_wrong_state` event.

c. **`ssh`** to that node and set it manually to active by running the following command:

```
/usr/lpp/mmfs/hadoop/bin/hdfs haadmin -transitionToActive -scale
```

d. Wait for 30 seconds and verify if the NameNode is now active by running the following commands:

```
/usr/lpp/mmfs/hadoop/bin/hdfs haadmin -getAllServiceState
```

and

```
/usr/lpp/mmfs/bin/mmhealth node show -v HDFS_NAMENODE -N cesNodes
```

39. CES HDFS Transparency fails to start if the Java version is upgraded.

Solution

For information on troubleshooting this issue, see HDFS Transparency fails to start if the Java version is upgraded.

40. The **`mmhdfs`** command cannot recognize the FQDN hostnames if the NameNodes or DataNodes were added with short hostname.

If IBM Storage Scale and HDFS Transparency are set up with short hostname then there is no issue with using a short hostname.

If IBM Storage Scale is set up with FQDN and HDFS Transparency is set up with short hostname then **`mmhdfs`** does not recognize the node as a NameNode or DataNode.

For example, the **`mmhdfs hdfs status`** command will state that this is not a NameNode and will exit with a return code 1.

Solution:

Set up Transparency to use FQDN by updating the `hdfs-site.xml` to set the NameNodes to FQDN and the worker file hostnames to FQDN.

41. Multi-HDFS cluster deployment through IBM Storage Scale 5.1.1.0 installation toolkit is not supported.

Solution:

If you want to create multi-hdfs clusters on the same IBM Storage Scale, perform the following:

a. Clear the installation toolkit HDFS metadata, by running the following command:

```
/spectrumscale config hdfs clear
```

b. Follow "Adding a new HDFS cluster into existing HDFS cluster on the same GPFS cluster using install toolkit" on page 73.

Note: Ensure that the creation of the new HDFS fields are unique from already existing HDFS cluster. The installation toolkit will not be able to check if there are duplicate values. The installation toolkit HDFS metadata will be regenerated after the CES HDFS cluster is deployed but will only contain the new HDFS cluster information.

42. **`mmhealth node`** shows CES in Degraded state.

When you are creating a CES HDFS cluster, **`mmhealth node`** shows *CES -v* as degraded and with `hdfs_namenode_wrong_state` message.

```
[root@scale-31 ~]# mmhealth node show CES -v
Node name:      scale-31.openstacklocal
```

```
Component         Status        Status Change            Reasons
--------------------------------------------------------------------------------------------
-----------------
CES               DEGRADED      2021-05-05 09:52:29
hdfs_namenode_wrong_state(hdfscluster3)
  AUTH            DISABLED      2021-05-05 09:49:28      -
  AUTH_OBJ        DISABLED      2021-05-05 09:49:28      -
  BLOCK           DISABLED      2021-05-05 09:49:27      -
  CESNETWORK      HEALTHY       2021-05-05 09:49:58      -
    eth1          HEALTHY       2021-05-05 09:49:44      -
  HDFS_NAMENODE   DEGRADED      2021-05-05 09:52:29
hdfs_namenode_wrong_state(hdfscluster3)
  NFS             DISABLED      2021-05-05 09:49:25      -
  OBJECT          DISABLED      2021-05-05 09:49:28      -
  SMB             DISABLED      2021-05-05 09:49:26      -

[root@scale-31 ~]# mmhealth event show hdfs_namenode_wrong_state
Event Name:              hdfs_namenode_wrong_state
Event ID:                998178
Description:             The HDFS NameNode service state is not as expected (e.g. is in
STANDBY but is supposed to be ACTIVE or vice versa)
Cause:                   The command /usr/lpp/mmfs/hadoop/sbin/mmhdfs monitor checkHealth
-Y returned serviceState which does not match the expected state when looking at the
assigned ces IP attributes
User Action:             N/A
Severity:                WARNING
State:                   DEGRADED

[root@scale-31 ~]# hdfs haadmin -getAllServiceState
scale-31.openstacklocal:8020                       active
scale-32.openstacklocal:8020                       standby
[root@scale-31 ~]#

[root@scale-31 ~]# mmces address list
Address     Node                      Ces Group          Attributes
----------- ------------------------ ------------------ ------------------
192.0.2.0   scale-32.openstacklocal   hdfshdfscluster3   hdfshdfscluster3
192.0.2.1   scale-32.openstacklocal   none               none
192.0.2.2   scale-32.openstacklocal   none               none
192.0.2.3   scale-31.openstacklocal   none               none
192.0.2.4   scale-31.openstacklocal   none               none
192.0.2.5   scale-31.openstacklocal   none               none
[root@scale-31 ~]#
```

The issue here is that the CES IP is assigned to the Standby NameNode instead of the Active NameNode.

Solution:

The following are the three solutions for this problem:

- Manually set the active NameNode to standby on the node by running the `/usr/lpp/mmfs/hadoop/bin/hdfs haadmin -transitionToStandby -scale` command. Then on the other node, set the standby NameNode to active by running the `/usr/lpp/mmfs/hadoop/bin/hdfs haadmin -transitionToActive -scale` command.
- Move the CES IP to the active NameNode by running the `mmces address move --ces-ip <CES IP> --ces-node <node name>` command.
- Restart the CES HDFS NameNodes by running the following commands:

  ```
  mmces service stop HDFS -a
  mmces service start HDFS -a
  ```

43. Kerberos principal update not taking effect on changing KINIT_PRINCIPAL in `hadoop-env.sh`.

    Solution:

    The CES HDFS Kerberos information is cached at `/var/mmfs/tmp/krb5cc_ces`. Delete this file to force the update.

44. If Kerberos was configured on multiple HDFS Transparency clusters using a common KDC server and the supplied `gpfs_kerberos_configuration.py` script, `kinit` with the hdfs user principal fails for all the clusters except the most recent one.

The kerberos configuration script `gpfs_kerberos_configuration.py`, generates a keytab fie for the hdfs user under the `/etc/security/keytabs/hdfs.headless.keytab` default path. The kinit error occurs because the `gpfs_kerberos_configuration.py` script updated the keytab file and invalidated the copies of the keytab on the previous cluster.

Solution:

From the most recent HDFS Transparency cluster that the script was run, copy the keytab file to all the other HDFS Transparency cluster nodes where the script was run.

For example:

If Hadoop cluster A ran the `gpfs_kerberos_configuration.py` script which created the hdfs user principal and Hadoop cluster B ran the `gpfs_kerberos_configuration.py` script which then updated the original hdfs user keytab, copy the hdfs keytab from Hadoop cluster B to Hadoop cluster A to ensure that the Hadoop cluster A kinit works properly.

This limitation has been fixed in HDFS Transparency 3.1.1.6.

45. DataNodes are down after system reboot.

    Solution:

    HDFS Transparency DataNodes may not start automatically after a system reboot. As a workaround, you can manually start the DataNodes after the system reboot by using the following command from one of the CES nodes as root:

    ```
    #/usr/lpp/mmfs/hadoop/sbin/mmhdfs hdfs-dn start
    ```

46. HDFS administrative commands, such as **`hdfs haadmin`** and **`hdfs groups`** cannot be executed from HDFS clients where Kerberos is enabled. The HDFS client ensures that the CES-HDFS user principle has the CES-HOST name instead of the NameNode hostname. The administrative commands fail with the following error:

    ```
    Caused by: java.lang.IllegalArgumentException: Server has invalid Kerberos principal:
    nn/c88f2u33.pokprv.stglabs.ibm.com@HADOOP.COM, expecting:
    nn/c88f2u31b.pokprv.stglabs.ibm.com@HADOOP.COM
    at org.apache.hadoop.security.SaslRpcClient.getServerPrincipal(SaslRpcClient.java:337)
    at org.apache.hadoop.security.SaslRpcClient.createSaslClient(SaslRpcClient.java:234)
    at org.apache.hadoop.security.SaslRpcClient.selectSaslClient(SaslRpcClient.java:160)
    at org.apache.hadoop.security.SaslRpcClient.saslConnect(SaslRpcClient.java:390)
    at org.apache.hadoop.ipc.Client$Connection.setupSaslConnection(Client.java:622)
    at org.apache.hadoop.ipc.Client$Connection.access$2300(Client.java:413)
    at org.apache.hadoop.ipc.Client$Connection$2.run(Client.java:822)
    at org.apache.hadoop.ipc.Client$Connection$2.run(Client.java:818)
    at java.security.AccessController.doPrivileged(Native Method)
    at javax.security.auth.Subject.doAs(Subject.java:422)
    at org.apache.hadoop.security.UserGroupInformation.doAs(UserGroupInformation.java:1729)
    at org.apache.hadoop.ipc.Client$Connection.setupIOstreams(Client.java:818)
    ... 15 more
    ```

    To resolve this, we have to add the following key in the `core-site.xml` file on the client:

    ```
    hadoop.security.service.user.name.key.pattern=*
    ```

    While using Cloudera Manager:

    a. Go to **Clusters** > **IBM Spectrum Scale** > **Configuration** > **Cluster-wide Advanced Configuration Snippet (Safety Valve)** for the `core-site.xml` file.
    b. Add the **`hadoop.security.service.user.name.key.pattern=*`** parameter and restart related services.

Limitations and differences from native HDFS

This topic lists the limitations and the differences from the native HDFS.

IBM Storage Scale information lifecycle management (ILM) is supported in the Hadoop environment.

- If you are planning for stretch cluster with Hadoop, contact scale@us.ibm.com.

- If you are planning for Hadoop Disaster Recovery with AFM, contact scale@us.ibm.com.

Application interaction with HDFS transparency

The Hadoop application interacts with the HDFS transparency similar to their interactions with native HDFS. They can access data in the IBM Storage Scale file system using Hadoop file system APIs and Distributed File System APIs.

The application might have its own cluster that is larger than HDFS transparency cluster. However, all the nodes within the application cluster must be able to connect to all nodes in HDFS transparency cluster by RPC.

Yarn can define the nodes in cluster by using the worker files. However, HDFS transparency can use a set of configuration files that are different from yarn. In that case, the worker files in HDFS transparency can be different from the one in the yarn.

Application interface of HDFS transparency

In HDFS transparency, applications can use the APIs defined in `org.apache.hadoop.fs.FileSystem` class and `org.apache.hadoop.fs.AbstractFileSystem` class to access the file system.

Command line for HDFS Transparency

The HDFS shell command line can be used with HDFS Transparency.

Command Interface	**Sub-Commands**	**Comments**
hdfs dfs -xxx hadoop dfs -xxx	Supported	**`hdfs dfs -du`** does not report exact total value for a directory for all the HDFS Transparency versions before HDFS Transparency 3.1.0-1. **Note:** For HDFS Transparency, `hdfs dfs -du /path/to/target` needs to recursively go through all directories and files under `<gpfs.mnt.dir>/<gpfs.data.dir>/path/to/target`. If you have a lot of subdirectories and files under `<gpfs.mnt.dir>/<gpfs.data.dir>/path/to/target`, it is recommended to not run this command frequently. **Note:** • For HDFS Transparency, hdfs `dfs -du /path/to/target` needs to recursively go through all the directories and files under `<gpfs.mnt.dir>/<gpfs.data.dir>/path/to/target`. If there are many subdirectories and files under `<gpfs.mnt.dir>/<gpfs.data.dir>/path/to/target`, it is recommended to not run this command frequently. • If you are using the **Hadoop `dfs -du`** command to get the output size of a file, the size value might not correspond to the du command output from the same file on the IBM Storage Scale file system. The IBM Storage Scale file system will consider replication factor and snapshot value into the file count value. Therefore, use the POSIX ls -l output file size on the IBM Storage Scale file system to compare with the Hadoop du command output file size.

Command Interface	Sub-Commands	Comments
hdfs envvars	Supported	
hdfs getconf	Supported	
hdfs groups	Supported	
hdfs jmxget	Supported	
hdfs haadmin	Supported	
hdfs zkfc	Supported	
hdfs crypto	Supported since HDFS Transparency 3.0.0+	CES HDFS does not support **hdfs crypto -listZones**. For a workaround with the GPFS policy engine, see the step 31 in the Second generation HDFS Transparency Protocol troubleshooting topic.
hdfs httpfs	Not tested	Take HDFS WebHDFS for REST API
hdfs lsSnapshottableDir	Not supported	
hdfs oev	Not supported	
hdfs oiv	Not supported	
hdfs oiv_legacy	Not supported	
hdfs snapshotDiff	Not supported	
hdfs balancer	Not supported	
hdfs cacheadmin	Not supported	
hdfs diskbalancer	Not supported	
hdfs ec	Not supported	
hdfs journalnode	Not supported	
hdfs mover	Not supported	
hdfs namenode	Not supported	
hdfs nfs3	Not supported	
hdfs portmap	Not supported	
hdfs secondarynamenode	Not supported	
hdfs storagepolicies	Not supported	
hdfs dfsadmin	Not supported	

Note: HDFS administrative commands, such as **hdfs haadmin** and **hdfs groups** cannot be executed from HDFS clients where Kerberos is enabled. The HDFS client ensures that the CES-HDFS user principle has the CES-HOST name instead of the NameNode hostname. The administrative commands fail while doing the hostname matching.

To resolve this, we have to add the following key in the `core-site.xml` file on the client:

```
hadoop.security.service.user.name.key.pattern=*
```

Snapshot support

In native HDFS, it can create snapshot against one directory. IBM Storage Scale supports two kinds of snapshot: file system snapshot (global snapshot) and independent fileset snapshot.

Before HDFS Transparency 2.7.3-1, HDFS Transparency implemented the snapshot from the Hadoop interface as a global snapshot and creating snapshot from a remote mounted file system was not supported.

HDFS Transparency 2.7.3-2 and later supports creating snapshot from a remote mounted file system.

The snapshot interface from the Hadoop shell is as follows:

```
hadoop dfs -createSnapshot /path/to/directory <snapshotname>
```

For the `/path/to/directory`, HDFS Transparency checks the parent directories from right to left. If there is one directory linked with one IBM Storage Scale fileset (check the column “Path” from the output of **`mmlsfileset <fs-name>`**) and if the fileset is an independent fileset, the **`mmlsfileset`** command creates the snapshot against the independent fileset. For example, if `/path/to/directory` is linked with fileset1 and fileset1 is an independent fileset, the above command creates snapshot against fileset1. If not, Transparency checks `/path/to`, then checks `/path` followed by `/` (which is `/gpfs.mnt.dir/gpfs.data.dir` from IBM Storage Scale file system). If Transparency cannot find any independent fileset linked with the above path `/path/to/directory`, Transparency creates the `<snapshotname>` against the fileset root in IBM Storage Scale file system.

Limitation of snapshot capability for HDFS Transparency 2.7.3-2 and later:

- Do not create a snapshot frequently (For example, do not create more than one snapshot every hour) because creating a snapshot holds on all on-fly IO operations. One independent fileset on IBM Storage Scale file system supports only 256 snapshots. When you delete a snapshot, it is better to remove the snapshot from the oldest snapshot to the latest snapshot.
- On IBM Storage Scale level, only the root user and the owner of the linked directory of independent fileset can create snapshot for IBM Storage Scale fileset. On HDFS interface from HDFS Transparency, only super group users (all users belong to the groups defined by **`gpfs.supergroup`** in `/usr/lpp/mmfs/hadoop/etc/hadoop/gpfs-site.xml` and **`dfs.permissions.superusergroup`** in `/usr/lpp/mmfs/hadoop/etc/hadoop/hdfs-site.xml`) and the owner of directory can create snapshot against the `/path/to/directory`.

  For example, if the userA is the owner of `/path/to/directory` and `/path/to/directory` is the linked directory of one independent fileset or `/path/to/directory` is the child directory under the linked directory of one independent fileset, userA can create the snapshot against `/path/to/directory`.
- Currently, Transparency caches all fileset information when Transparency is started. After Transparency is started, newly created filesets will not be detected automatically. You need to run **`/usr/lpp/mmfs/hadoop/bin/gpfs dfsadmin -refresh hdfs://<namenode-hostname>:8020 refreshGPFSConfig`** to refresh the newly created filesets or you can restart the HDFS Transparency.
- Do not take nested fileset, such as `/gpfs/dependent_fileset1/independent_fileset1/dependent_fileset2/independent_fileset2`. Transparency creates the snapshot against the first independent fileset by checking the path from right to left. Also, the snapshots for independent filesets are independent. For example, the snapshot of independent fileset1 has no relationship with any other independent fileset.
- **`hadoop fs -renameSnapshot`** is not supported.
- Do not run **`hadoop dfs -createSnapshot`** or **`Hadoop dfs -deleteSnapshot`** under the `.snapshots` directory that is located in IBM Storage Scale file system. Otherwise, error such as `Could not determine current working directory` occurs.

  For example,

  ```
  [root@dn01 .snapshots]# hadoop fs -deleteSnapshot / snap1
  Error occurred during initialization of VM
  ```

```
java.lang.Error: Properties init: Could not determine current working directory.
    at java.lang.System.initProperties(Native Method)
    at java.lang.System.initializeSystemClass(System.java:1166)
```

- HDFS Transparency does not need to run **`hdfs dfsadmin --allowSnapshot`** or **`hdfs dfsadmin -disallowSnapshot`** commands.
- Snapshot is supported similarly for multiple IBM Storage Scale file systems.
- Snapshot for remote mounted file system is not supported if **`gpfs.remotecluster.autorefresh`** (`/usr/lpp/mmfs/hadoop/etc/hadoop/gpfs-site.xml`) is configured as *false*. By default, it is true.
- HDFS Transparency supports only the Hadoop snapshot create and delete functions. Hadoop snapshot **`list`** command will only list the path of the snapshot directory name and not the snapshot contents because the snapshots are created by the IBM Storage Scale snapshot commands and are stored in the Scale snapshot root directory where the Hadoop environment does not have access.
- To perform snapshot restores, see the following topics under the *Administering* section in IBM Storage Scale documentation:
  - *Restoring a file system from a snapshot* topic under the *Creating and maintaining snapshots of file systems* section
  - *Restoring a subset of files or directories from a local file system snapshot* topic under the *Managing file systems* section
  - *Restoring a subset of files or directories from local snapshots using the sample script* topic under the *Managing file systems* section

Hadoop ACL and IBM Storage Scale protocols

Hadoop supports only POSIX ACL. Therefore, HDFS Transparency supports only POSIX ACL. If your Hadoop applications involve ACL operations, you need to configure the type of authorization that is supported by the IBM Storage Scale **`-k`** option as *all* or *posix*. If not, the ACL operations from Hadoop will report exceptions.

If you want to run IBM Storage Scale protocol NFS, you need to configure the **`-k`** as *all*. When HDFS Transparency accesses the file configured with NFSv4 ACL, NFSv4 ACL does not usually take effect (NFSv4 ACL is configured to control the access from NFS clients and usually NFS clients are not Hadoop nodes).

If you want to run both HDFS Transparency and IBM Storage Scale protocol SMB, SMB requires IBM Storage Scale file system to be configured with **`-k nfs4`**. The workaround is to configure **`-k nfs4`** to enable CES/SMB and then change it into **`-k all`** after the SMB service is up (after enablement, SMB service can be started successfully with the file system configured with **`-k all`** when failover is triggered). This can make both the SMB and HDFS co-exist on the same IBM Storage Scale file system. However, even with this workaround, you cannot take the SMB client to control the ACL of the files/directories from IBM Storage Scale. It is verified that the SMB ACL does not work properly over directories with the file system configured as **`-k all`**. For more information on limitations, see CES HDFS Limitations and Recommendations.

Note:

- Ensure that the file system is set to *ACL ALL* in a multi-protocol environment. Also, do not set the ACL to NFSv4 format when ingesting data into it. If ACL is set to NFSv4 format, HDFS will not be able to access the data.
- In a multi-protocol environment, only one protocol should modify the ACLs and write to the directory/container while all the other protocols should only read in a staged manner.

  For example:

  NFS ingests the data and HDFS reads the data to run the analytics.

The difference between HDFS Transparency and native HDFS

The configuration that differ from HDFS in IBM Storage Scale.

Property name	Value	New definition or limitation
dfs.storage.policy.enabled	True/false	Not supported by HDFS Transparency. This means that storage policy commands like **hdfs storagepolicies** and configuration like **fs.setStoragePolicy** are not supported.
dfs.permissions.enabled	True/false	For HDFS protocol, permission check is always done.
dfs.namenode.acls.enabled	True/false	For native HDFS, the NameNode manages all meta data including the ACL information. HDFS can use this information to turn on or off the ACL checking. However, for IBM Storage Scale, HDFS protocol will not save the meta data. When ACL checking is on, the ACL will be set and stored in the IBM Storage Scale file system. If the admin turns ACL checking off, the ACL entries set before are still stored in IBM Storage Scale and remain effective. This will be improved in the next release.
dfs.blocksize	Long digital	Must be an integer multiple of the IBM Storage Scale file system blocksize (**mmlsfs -B**), the maximal value is 1024 * file-system-data-block-size (**mmlsfs -B**).
dfs.namenode.fs-limits.max-xattrs-per-inode	INT	Does not apply to HDFS Transparency.
dfs.namenode.fs-limits.max-xattr-size	INT	Does not apply to HDFS Transparency.
dfs.namenode.fs-limits.max-component-length	Not checked	Does not apply to HDFS Transparency; the file name length is controlled by IBM Storage Scale. Refer IBM Storage Scale FAQ for file name length limit (255 unicode-8 chars).
Native HDFS caching	Not supported	IBM Storage Scale has its own caching mechanism.

Property name	Value	New definition or limitation
NFS Gateway	Not supported	IBM Storage Scale provides POSIX interface and taking IBM Storage Scale protocol could give your better performance and scaling.

Functional limitations

- The maximum number of Extended Attributes (EA) is limited by IBM Storage Scale and the total size of the EA key. Also, the value must be less than a metadata block size in IBM Storage Scale.
- The EA operation on snapshots is not supported.
- Raw namespace is not implemented because it is not used internally.
- If **gpfs.replica.enforced** is configured as gpfs, the Hadoop shell command **hadoop dfs -setrep** does not take effect. Also, **hadoop dfs -setrep -w** stops functioning and does not exit. Also, if one file is smaller than inode size (by default, it is 4Kbytes per inode), IBM Storage Scale will store the file as data-in-inode. For these kinds of small files, the data replica of these data-in-inode file will be the replica of meta data instead of replica of data.
- HDFS Transparency NameNode does not provide *safemode* because it is stateless.
- HDFS Transparency NameNode does not need the second NameNode like native HDFS because it is stateless.
- Maximal replica for IBM Storage Scale is *3*.
- **hdfs fsck** does not work against HDFS Transparency. Instead, run **mmfsck**.
- IBM Storage Scale has no ACL entry number limit (maximal entry number is limited by Int32).
- **distcp --diff** is not supported over snapshot.
- + in file name is not supported if taking the schema `hftp://`. If not taking `hftp://`, + in file name works.
- In HDFS, files can only be appended. If a file is uploaded into the same location with the same file name to overwrite the existing file, then HDFS can detect this according to the inode change. However, IBM Storage Scale supports POSIX interface and other protocol interfaces (for example, NFS/SMB) and one file could be changed from non HDFS interface. Therefore, files loaded for Hadoop services to process cannot be modified until the process completes. Else the service or job fails.
- To view a list of HDFS 3.1.1 community fixes that are not yet supported by HDFS Transparency, see HDFS 3.1.1 community fixes.
- GPFS quota is not supported in HDFS Transparency.
- The **-px** option for the **hadoop fs -cp -px** command is not supported when SELinux is enabled. This is because HDFS cannot handle the system extended attribute (xattr) operation.
- **hadoop namenode -recover** is not supported.
- **hdfs haadmin -refreshNodes** is not supported.

hdfs haadmin

For CES HDFS integration, from IBM Storage Scale version 5.0.4.2, the **hdfs haadmin** command has the following new options:

- **-checkHealth -scale**
- **-transitionToActive/-transitionToStandby**

The option **scale** is used to retrieve the health and service state of the NameNodes on IBM Storage Scale.

The option **-transitionToActive/-transitionToStandby** is used to change the state of the local NameNode to active or standby.

Usage:

```
haadmin [-ns <nameserviceId>]
 [-checkHealth -scale [-all] [-Y]] # For Spectrum Scale usage only
 [-transitionToActive [--forceactive] -scale] # For Spectrum Scale usage only
 [-transitionToStandby -scale] # For Spectrum Scale usage only
 [-transitionToActive [--forceactive] <serviceId>]
```

Examples:

On the NameNode, check NameNode status by running the following command:

```
# /usr/lpp/mmfs/hadoop/bin/hdfs haadmin -checkHealth -scale -all
```

On the first NameNode, transition first NameNode to ACTIVE by running the following command:

```
 # /usr/lpp/mmfs/hadoop/bin/hdfs haadmin -transitionToActive --forceactive -scale
```

On the second NameNode, transition second NameNode to ACTIVE by running the following command:

```
# /usr/lpp/mmfs/hadoop/bin/hdfs haadmin -transitionToActive --forceactive -scale
```

On the secondary NameNode, transition second NameNode to STANDBY by running the following command:

```
# /usr/lpp/mmfs/hadoop/bin/hdfs haadmin -transitionToStandby -scale
```

Configuration parameters

This section lists and describes the configuration parameters for HDFS Transparency and `gpfs-site.xml`.

Configuration options for HDFS Transparency

The following configuration options are provided for HDFS Transparency:

1. `delete` option:

   Native HDFS deletes the metadata in the memory using the single threaded mechanism while HDFS Transparency deletes it under IBM Storage Scale distributed metadata using the same single threaded mechanism. From HDFS Transparency 3.1.0-5 and 3.1.1-2, HDFS Transparency will delete the metadata using the multi-threaded mechanism based on the sub-directories and files. The following parameters can be used to tune the delete operation threads under gpfs-site.xml:

Configuration options	Description
gpfs.parallel.deletion.max-thread-count	Specifies the number of threads used for parallel deletion. Default is 512.
gpfs.parallel.deletion.per-dir-threshold	Specifies the number of entries in a single directory that are handled by a single thread. If this threshold is reached a new thread is started. Default is 10000.
gpfs.parallel.deletion.sub-dir-threshold	Specifies the number of sub-directories (the number of all children, sub-children, sub-sub-children, and so on) that are handled by a single thread. If this threshold is reached a new thread is started. Default is 1000.

2. du option:

   Native HDFS collects the metadata statistics (for example, disk usage statistics, **`hdfs dfs -du`**, or count files and directories, **`hdfs dfs -count`**) in the memory using the single threaded mechanism while HDFS Transparency collects the metadata statistics under IBM Storage Scale distributed metadata using the same single threaded mechanism. From HDFS Transparency 3.1.0-6 and 3.1.1-2, HDFS Transparency will collect the metadata statistics using the multi-threaded mechanism based on the sub-directories and files. The following parameters can be used to tune the operation threads under gpfs-site.xml:

Configuration options	Description
gpfs.parallel.summary.max-thread-count	Specifies the number of threads used for parallel directory summary. Default is 512.
gpfs.parallel.summary.per-dir-threshold	Specifies the number of entries in a single directory that are handled by a single thread. If this threshold is reached a new thread is started. Default is 10000.
gpfs.parallel.summary.sub-dir-threshold	Specifies the number of sub-directories (the number of all children, sub-children, sub-sub-children, and so on) that are handled by a single thread. If this threshold is reached a new thread is started. Defaults is 1000.

3. `list` option:

   From HDFS Transparency 3.1.0-6 and 3.1.1-3, the following configuration options for using multiple threads to list a directory and load the metadata of its children are provided:

Configuration options	Description
gpfs.inode.update-thread-count	Specifies the total count of the threads that are used for running statistics on directory entries. The default value is 100. Therefore, by default, the NameNode will create a thread pool with 100 threads and use the thread pool to execute the statistics on directory entries.
gpfs.inode.max-update-thread-count-per-dir	Specifies the max count of the threads that are used to list a single directory. The default value is 8. Therefore, by default, no matter how big the directory is, at most 8 threads will be used to list the directory and load its children.
gpfs.inode.update-thread-count-factor-per-dir	Specifies the count of the children of a directory that are handled by a single directory-listing thread. The default value is 5000. Therefore, by default, if a directory has less than 5000 children, only 1 thread will be used to list the directory and load its children. If the directory has children that are more than or equal to 5000 but less than 10000, two threads will be used to list the directory and load its children, and so on. The total number of threads for a directory cannot exceed the **gpfs.inode.max-update-thread-count-per-dir** value.
gpfs.scandir-due-to-lookup-threshold	Specifies the threshold that is used to identify a large directory. If the number of children for a directory is greater than the **gpfs.inode.max-update-thread-count-per-dir** value, it is identified as a large directory. While listing this directory, the NameNode will try to prefetch the

Configuration options	Description
	metadata of its children to speed up the listing process. The default value is 10000.
gpfs.parallel.ls.max.invocation.max-thread-count (starting with 3.2.2-7)	Specifies the total count of the threads that are used for listing directory contents. The default value is 512. Therefore, by default, the NameNode will create a thread pool with 512 threads and use the thread pool to execute the listing directory contents.

4. DataNode option:

   From HDFS Transparency 3.1.0-9 and 3.1.1-6, the following configuration options for the DataNode locking mechanism are provided:

   - **dfs.datanode.lock.read.write.enabled**: If this parameter is set to *true*, the FsDataset lock will be a read/write lock. If it is set to *false*, all locks will be write locks. The default value is *true*.
   - **dfs.datanode.lock.fair**: If this parameter is set to *true*, the Datanode FsDataset lock will be used in the Fair mode. This helps to prevent the writer threads from being starved, but can lower the lock throughput. The default value is *true*.
   - **dfs.datanode.lock-reporting-threshold-ms**: When thread waits to obtain a lock, or if a thread holds a lock for more than the threshold, a log message will be written. The default value is *300*.

Configuration parameters for gpfs-site.xml

Table 15. Configuration parameters for gpfs-site.xml

Key	Default value	Description	Supported versions
gpfs.data.dir		Setting this to a subdirectory of the gpfs mount point will make this a subdirectory to the root directory from a hadoop client point of view. Leave it empty to make the whole gpfs file system visible to the hadoop client. When specifying the subdirectory, the gpfs mount point should not be included in the string.	3.1.0-x 3.1.1-x 3.2.2-x 3.3.0-0
gpfs.edit.log.retain.not-modified-for-seconds	2592000	The edit log files that have not been accessed by **gpfs.edit.log.retain.not-modified-for-seconds** will be automatically deleted by NameNode.	3.1.0-x 3.1.1-x 3.2.2-x 3.3.0-0

Table 15. Configuration parameters for `gpfs-site.xml` (continued)

Key	Default value	Description	Supported versions
`gpfs.encryption.enabled`	false	Enable or disable the encryption. It is important to understand the difference between HDFS-level encryption and in-built encryption with IBM Storage Scale. HDFS level encryption is per user based whereas in-built encryption is per node based. Therefore, if the use case demands more fine-grained control at the user level, use HDFS-level encryption. However, if you enable HDFS-level encryption, you will not be able to get in-place analytics benefits such as accessing the same data with HDFS and POSIX/NFS. This requires Ranger and Ranger KMS. If you plan to enable this, you should enable it on the native HDFS first and confirm it is working before you switch native HDFS into HDFS Transparency.	3.1.0-x 3.1.1-x 3.2.2-x 3.3.0-0
`gpfs.fileset.use-global-snapshot`	false	Use global snapshot or fileset snapshot. For more information, see *mmcrsnapshot command* in *IBM Storage Scale: Command and Programming Reference Guide*	3.1.0-x 3.1.1-x 3.2.2-x 3.3.0-0
`gpfs.inode.file-per-thread`	1000	This parameter tells how many files a thread should handle when using multiple threads to load metadata of the files under a directory.	3.1.0-x

Table 15. Configuration parameters for `gpfs-site.xml` (continued)

Key	Default value	Description	Supported versions
`gpfs.inode.getdirent-max-buffer-size`	64 * 1024 * 1024	This parameter indicates the size of the buffer used when reading the directory entries (**`readdir syscall`**).	3.1.0-x 3.1.1-x 3.2.2-x 3.3.0-0
`gpfs.inode.max-update-thread-count-per-dir`	8	Specifies the max count of the threads that are used to list a single directory. The default value is *8*. Therefore, by default, no matter how big the directory is, at most 8 threads will be used to list the directory and load its children.	3.1.0-x 3.1.1-x 3.2.2-x 3.3.0-0
`gpfs.inode.perf-log-duration-threshold`	10000	Specifies the threshold to identify if some log messages should be printed for some time-consuming operations.	3.1.0-x 3.1.1-x 3.2.2-x 3.3.0-0
`gpfs.inode.update-thread-count`	100	Specifies the total count of the threads that are used for directory listing. The default value is *100*. Therefore, by default, the NameNode will create a thread pool with 100 threads and use the thread pool to execute the directory-listing threads.	3.1.0-x 3.1.1-x 3.2.2-x 3.3.0-0

Table 15. Configuration parameters for gpfs-site.xml (continued)			
Key	**Default value**	**Description**	**Supported versions**
gpfs.inode.update-thread-count-factor-per-dir	5000	Specifies the count of the children of a directory that are handled by a single directory-listing thread. The default value is *5000*. Therefore, by default, if a directory has less than 5000 children, only 1 thread will be used to list the directory and load its children. If the directory has children that are more than or equal to 5000 but less than 10000, two threads will be used to list the directory and load its children, and so on. The total number of threads for a directory cannot exceed the **gpfs.inode.max-update-thread-count-per-dir** value.	3.1.0-x 3.1.1-x 3.2.2-x 3.3.0-0
gpfs.mnt.dir		Specifies the gpfs mount point.	3.1.0-x 3.1.1-x 3.2.2-x 3.3.0-0
gpfs.parallel.deletion.sub-dir-threshold	1000	Specifies the number of sub-directories (the number of all children, sub-children, sub-sub-children, and so on) that are handled by a single thread. If this threshold is reached a new thread is started.	3.1.0-x 3.1.1-x 3.2.2-x 3.3.0-0
gpfs.parallel.deletion.max-thread-count	512	Specifies the number of threads used for parallel deletion.	3.1.0-x 3.1.1-x 3.2.2-x 3.3.0-0
gpfs.parallel.deletion.per-dir-threshold	10000	Specifies the number of entries in a single directory that are handled by a single thread. If this threshold is reached a new thread is started.	3.1.0-x 3.1.1-x 3.2.2-x 3.3.0-0

Table 15. Configuration parameters for `gpfs-site.xml` (continued)

Key	Default value	Description	Supported versions
`gpfs.parallel.summary.max-thread-count`	512	Specifies the number of threads that are used for parallel directory summary.	3.1.0-x 3.1.1-x 3.2.2-x 3.3.0-0
`gpfs.parallel.summary.per-dir-threshold`	10000	Specifies the number of entries in a single directory that are handled by a single thread. If this threshold is reached a new thread is started.	3.1.0-x 3.1.1-x 3.2.2-x 3.3.0-0
`gpfs.parallel.summary.sub-dir-threshold`	1000	Specifies the number of sub-directories (the number of all children, sub-children, sub-sub-children, ...) that are handled by a single thread. If this threshold is reached a new thread is started.	3.1.0-x 3.1.1-x 3.2.2-x 3.3.0-0
`gpfs.ranger.enabled`	scale	Specifies the value to enable and disable the ranger support. Set it to true to enable and false to disable. From HDFS Transparency 3.1.0-6 and 3.1.1-3, the Ranger is always supported and this value should be set to *scale*. This parameter is removed in HDFS Transparency 3.3.x-x.	3.1.0-x 3.1.1-x
`gpfs.remotecluster.autorefresh`	true	Enables auto refresh of the GPFS configuration details in HDFS Transparency when the NameNode is started. When set to *false*, the user must manually run the `initmap.sh` script.	3.1.0-x 3.1.1-x 3.2.2-x 3.3.0-0

Table 15. Configuration parameters for `gpfs-site.xml` (continued)

Key	Default value	Description	Supported versions
gpfs.replica.enforced	gpfs	Set this parameter to *dfs*, if you want to use **dfs.replication**. Set this parameter to *gpfs*, if you want to use the default replication of gpfs. If you set this to *gpfs*, setReplication API will not take effect anymore and would break the **-setrep** command of the fs shell.	3.1.0-x 3.1.1-x 3.2.2-x 3.3.0-0
gpfs.scandir-due-to-lookup-threshold	10000	Specifies the threshold that is used to identify a large directory. If the number of children for a directory is greater than the **gpfs.inode.max-update-thread-count-per-dir** value, it is identified as a large directory. While listing this directory, the NameNode will try to prefetch the metadata of its children to speed up the listing process. The default value is 10000.	3.1.0-x 3.1.1-x 3.2.2-x 3.3.0-0
gpfs.slowop.count.threshold	50000	If an operation object count exceeds this threshold a log entry will be recorded to track the occurrence.	3.1.0-x 3.1.1-x 3.2.2-x 3.3.0-0
gpfs.slowop.duration.threshold	30	If an operation exceeds the threshold in seconds specified, a log entry will be recorded to track the slow operation.	3.1.0-x 3.1.1-x 3.2.2-x 3.3.0-0
gpfs.slowop.message.interval	30	Specifies the rate to log (in seconds) a long running operation.	3.1.0-x 3.1.1-x 3.2.2-x 3.3.0-0
gpfs.ssh.user	root	Specifies the user for the NameNode to connect to the nodes in the remote mount mode.	3.1.0-x 3.1.1-x 3.2.2-x 3.3.0-0

Table 15. Configuration parameters for `gpfs-site.xml` (continued)

Key	Default value	Description	Supported versions
gpfs.storage.type	shared	Specifies the type of storage. Set this to shared if you are using shared-storage cluster.	3.1.0-x 3.1.1-x 3.2.2-x 3.3.0-0
gpfs.limit.max.os-uid-gid.lookup	false	When enabled, limits the OS uid/gid lookups.	3.2.2-7+

Note: The *-x* under the *Supported versions* column indicates the latest version.

HDFS Transparency limitations and recommendations

This topic lists the limitations and recommendations for CES HDFS.

- In IBM Storage Scale 5.0.4.2, the installation toolkit does not support the ESS deployment for CES HDFS. For efix support, send an email to scale@us.ibm.com or upgrade to IBM Storage Scale 5.0.4.3 or later with the correct BDA integration toolkit "HDFS Transparency support matrix" on page 27.
- Red Hat Enterprise Linux is supported.
- CES HDFS supports installation of HDFS Transparency 3.1.1/3.2.2/3.3.0.
- Upgrade path from HDFS Transparency 3.1.0 and earlier to CES HDFS Transparency 3.1.1 is not supported.
- Upgrade path from HDFS Transparency 3.1.1 and earlier to CES HDFS Transparency 3.2.2 is not supported.
- Upgrade path from HDFS Transparency 3.2.2 and earlier to CES HDFS Transparency 3.3.0 is not supported.
- In IBM Storage Scale 5.0.4.2, remote mount setup is not supported through the installation toolkit. From IBM Storage Scale 5.0.4.3, remote mount setup is supported with the correct BDA integration toolkit "HDFS Transparency support matrix" on page 27.
- Ambari is not supported for CES HDFS integration.
- CES only monitors and does failover for HDFS Transparency NameNodes.
- ZKFailoverController is not used for CES HDFS HA.
- It is required to setup a CES shared root file system (**cesSharedRoot**) for storing CES shared configuration data, for protocol recovery and for other protocol specific purposes.
- In general, to install and configure CES, all protocols packages available on your platform must be installed. Installing a subset of available protocols is not supported.

  However, CES HDFS protocol is an exception. HDFS will only be installed if it is enabled.
- The **mmuserauth** is not used by CES HDFS. Manual Kerberos configuration is required to configure Hadoop services.
- The **mmhadoopctl** command is for HDFS Transparency 3.1.0 and earlier. Do not use **mmhadoopctl** in the CES HDFS environment.
- Do not use the **mmhdfs start/stop/restart** command on the NameNodes. Instead use the **mmces service start/stop hdfs** command to start and stop the NameNodes.
- For generic HDFS Transparency Hadoop distribution support information, see "Hadoop distribution support" on page 24.
- The **mmhdfs import/export/upload/config set** commands are based on the HDFS Transparency configuration directory: `/var/mmfs/hadoop/etc/hadoop`.

- The file system ACL setting for **-k nfs4** is not supported for the Hadoop environment. Set the file system ACL **-k** value to `all`. For more information, see "Hadoop ACL and IBM Storage Scale protocols" on page 239 and Issues that can be encountered if the IBM Storage Scale ACL is not set properly.
- CES HDFS does not support viewfs.
- CES HDFS does not support **hdfs crypto -listZones**. For a workaround with the GPFS policy engine, see the step 31 in the Second generation HDFS Transparency Protocol troubleshooting topic.
- CES HDFS does not support short-circuit mode.
- For remote mount limitation information, see *Limitations of protocols on remotely mounted file systems* in *IBM Storage Scale: Administration Guide*.
- For IBM Storage Scale software and scaling limitations, see *Software questions* and *Scaling questions* in IBM Storage Scale FAQ.
- For additional recommendations and restrictions, see CES HDFS Planning.

Chapter 3. IBM Storage Scale Hadoop performance tuning guide

Overview

Introduction

The Hadoop ecosystem consists of many open source projects. One of the central components is the Hadoop Distributed File System (HDFS).

HDFS is a distributed file system designed to run on the commodity hardware. Other related projects facilitate workflow and the coordination of jobs, support data movement between Hadoop and other systems, and implement scalable machine learning and data mining algorithms. HDFS lacks the enterprise class functions necessary for reliability, data management, and data governance. IBM's General Parallel File System (GPFS) is a POSIX compliant file system that offers an enterprise class alternative to HDFS.

There are a many similar tuning guides available for native HDFS. However, when you apply those tuning steps over IBM Storage Scale, usually you cannot get the best performance because of natural design difference between HDFS and IBM Storage Scale. With this section the customers can tune different Hadoop components when they run Hadoop over IBM Storage Scale and HDFS Transparency so that they get good performance.

All the tuning configurations mentioned in this section are for Hadoop 2.7.x and Hadoop 3.0.x.

Hadoop over IBM Storage Scale

By default, Hadoop uses HDFS schema (`hdfs://<namenode>:<portnumber>`) for all the components to read data from HDFS or write data into HDFS.

Hadoop also supports local file schema (`file:///`) and other HCFS schema for Hadoop components to read data from or write data into other distributed file systems.

IBM Storage Scale HDFS Transparency follows the Hadoop HCFS specification and provide the HDFS RPC level implementation for Hadoop components to read data from or write data into IBM Storage Scale. It also supports Kerberos, federation, and distcp.

Spark over IBM Storage Scale

Apache Spark is a fast and general engine for large scale data processing.

Spark supports multiple ways (such as `hdfs://`, `file:///`) for applications running over Spark to access the data in distributed file systems.

Performance overview

This section describes the type of configurations that impact job executions from MapReduce and Hive.

MapReduce

This section describes Yarn's job management and execution flow so that we know the impact on MapReduce job performance.

The topology of job management in Yarn is illustrated by Figure 21 on page 254. The picture is from Apache Hadoop YARN website:

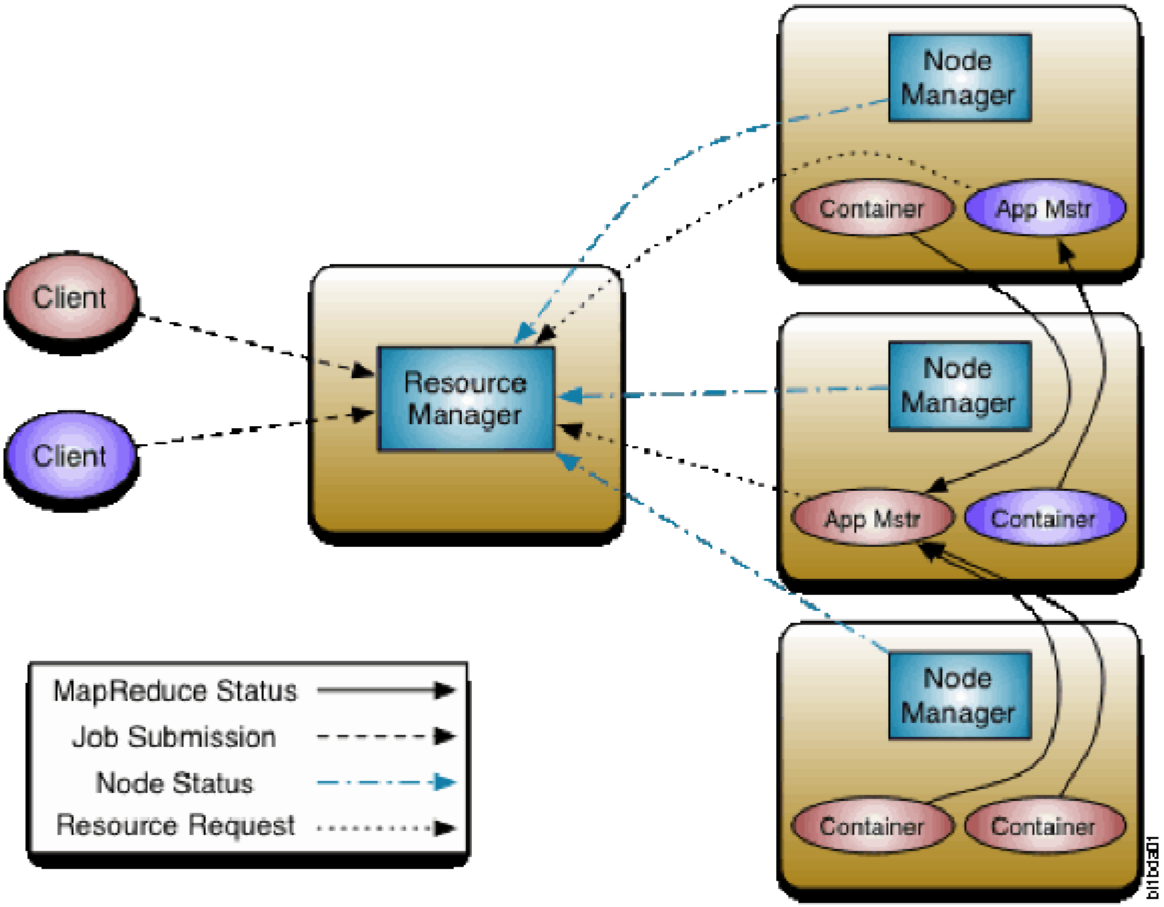


Figure 21. Topology of Job management in Yarn

After a client submits one job into Yarn, the Resource Manager receives the request and puts it into Resource Manager queue for scheduling. For each job, it has one App Master that coordinates with the Resource Manager and the Node Manager to request the resource for other tasks of the job and start these tasks after their resource is allocated.

From Figure 21 on page 254, all tasks are run over the nodes on which there is Node Manager service. Only these tasks will read data from distributed file system or write data into distributed file systems. For native HDFS, if HDFS DataNode services are not running over these Node Manager nodes, all the Map/Reduce jobs cannot leverage data locality for high read performance because all data must be transferred over network.

Also, App Master of one job takes configured CPU resources and memory resources.

To one specific map/reduce job, the execution flow is illustrated by Figure 22 on page 255:

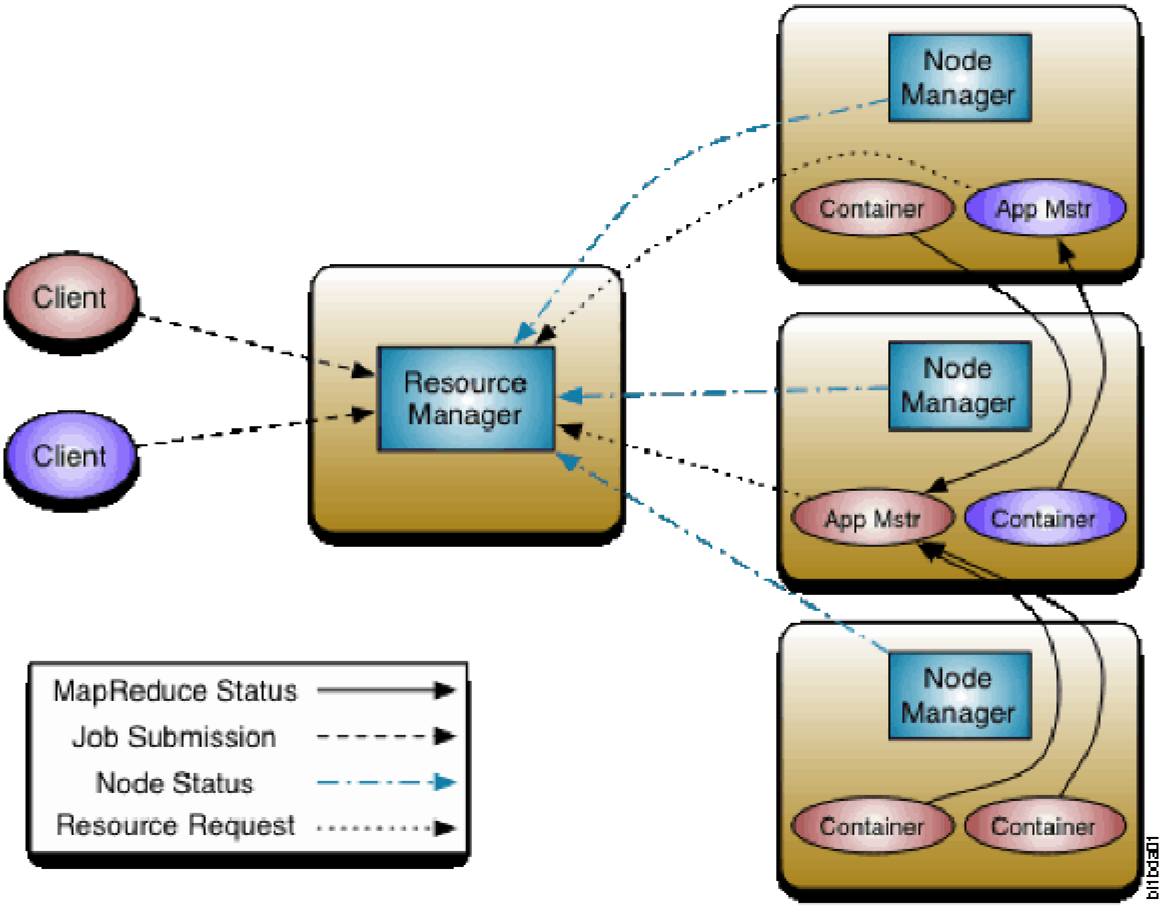


Figure 22. MapReduce Execution Flow

For each map task, it reads one split of input data from distributed file system. The split size is controlled by `dfs.blocksize`, `mapreduce.input.fileinputformat.split.minsize` and `mapreduce.input.fileinputformat.split.maxsize`. If data locality could be met, this improves the data read time. Map task spills the input data into Yarn's local directories when its buffer is filled up according to Yarn's configuration (controlled by `mapreduce.task.io.sort.mb` and `mapreduce.map.sort.spill.percent`). For spilled out data, map task needs to read them into memory, merge them, and write out again into Yarn's local directories as one file. This is controlled by `mapreduce.task.io.sort.factor` and `mapreduce.map.combine.minspills`. Therefore, if a spill happens, we need to write them, read them, and write them again. It is 3x times of IO time. Therefore, we need to tune Yarn's configuration to avoid spill.

After map task is done, its output is written onto the local directories. After that, if there are reduce tasks up, reduce task will take HTTP requests to fetch these data into local directories of the node on which the reduce task is running. This phase is called copy. If thread number of copy is too small, the performance is impacted. The thread number of copy is controlled by `mapreduce.reduce.shuffle.parallelcopies` and `mapreduce.tasktracker.http.threads`.

At the merge phase in reduce task, it keeps fetching data from different map task output into its local directories. If these data fill up reduce task's buffer, this data is written into local directories (this is controlled by `mapreduce.reduce.shuffle.input.buffer.percent` and `mapreduce.reduce.shuffle.merge.percent`). Also, if the spilled data file number on local disks exceed the `mapreduce.reduce.merge.inmem.threshold`, reduce task will also merge these files into larger ones. After all map data are fetched, reduce task will enter the sort phase which merges spilled files maintaining their sort order. This is done in rounds and the file number merged per round is controlled by `mapreduce.task.io.sort.factor`. Finally, one reduce task will generate one file in distributed file system.

Hive

Apache Hive is designed for traditional data warehousing tasks and it is not for OLTP workloads.

Apache Hive comprises of several components illustrated in Figure 23 on page 256:

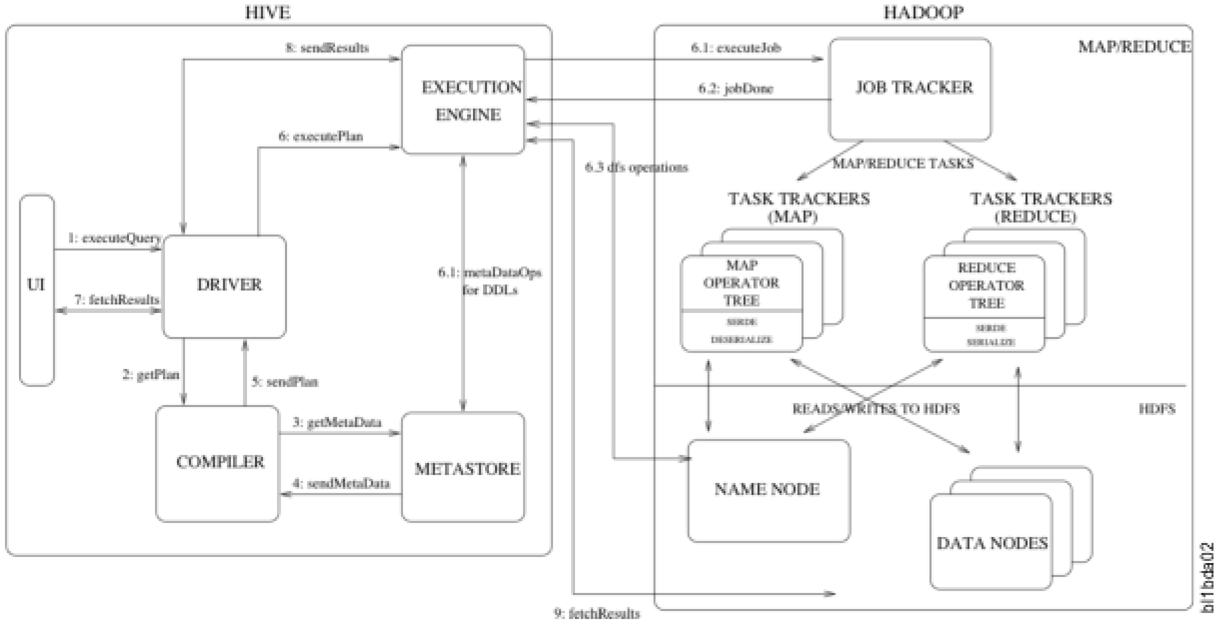


Figure 23. Framework of Hive (from Apache Hive website)

In Hadoop distro, such as IBM BigInsights, Hive Metastore server provides metadata related operations (such as database, table, partition). WebHCat server provides web interface for clients to execute queries against Hive. HiveServer2 server is a server interface that enables remote clients to execute queries against Hive and retrieve the results.

Hadoop performance planning over IBM Storage Scale

Storage model

Internal disks (FPO)

Similar to Hadoop HDFS, IBM Storage Scale FPO takes the sharing nothing framework. Every server in the cluster is a computing node and a storage node. IBM Storage Scale manages the local disks from each server and unifies them as one distributed file system for applications running over these nodes.

Figure 24 on page 257 illustrates a typical deployment:

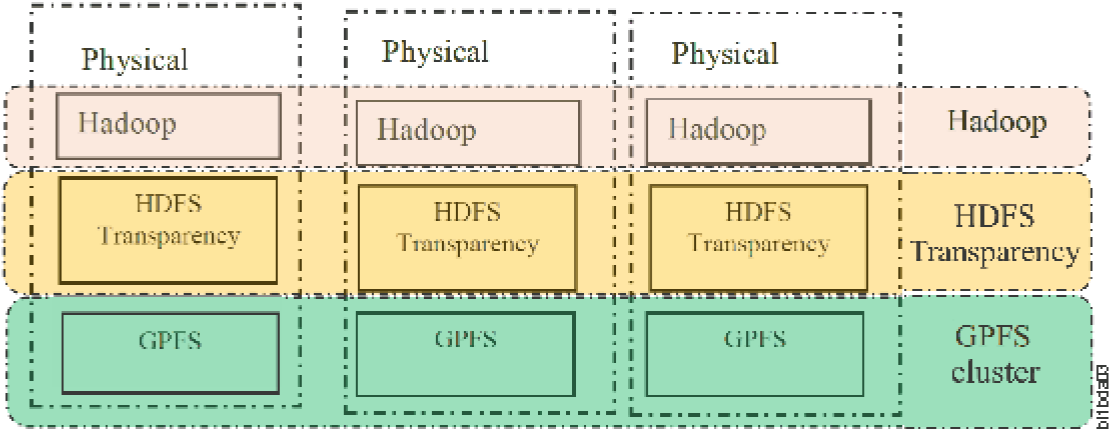


Figure 24. Hadoop over IBM Storage Scale FPO

If you are deploying Hadoop over IBM Storage Scale FPO, execute the following steps for better performance cluster:

1. IBM Storage Scale FPO.

   Refer to the *Configuring FPO* sub-section under the **Administering** > **File Placement Optimizer** section in IBM Storage Scale documentation.

   Some important configurations, such as, data replica, meta data replica, disk layout, failure group definition must be decided in this step. Also, you must tune the cluster configuration accordingly.

   You need to consider the shuffle directory for intermediate data.

   Note:

   - Because of limited network bandwidth, do not run IBM Storage Scale FPO over 1Gb ethernet adapter in the production environment.
   - 8~12 SAS or SATA disks are recommended disks per node with one 10Gb ethernet adapter. For meta data, SSD is recommended, especially for file/directory operation sensitive workloads.

2. HDFS Transparency nodes.

   All FPO nodes must be installed with HDFS Transparency. Also, you should avoid introducing Transparency nodes and other Hadoop services (especially for Yarn's NodeManager, Resource Manager, Hive's Hiveserver2, HBase Master and Region Servers) over IBM Storage Scale clients because this will make all the data for these servers go over network. Therefore, degrading the performance.

   Assign HDFS Transparency NameNode over IBM Storage Scale node with meta disks.

3. Server nodes for Yarn, HBase, Hive and Spark

   Assign Yarn ResourceManager over the node running with HDFS Transparency. Assign HBase Master and HDFS Transparency NameNode on different nodes because of memory allocation.

   HBase Master needs to be allocated with large memory for better performance. If you have many files, then assign the Transparency Namenode with more memory. If the node running Transparency Namenode does not have enough memory for HBase Master, assign them over different nodes.

4. Server nodes for other components.

   Try for each node to have even I/O workloads and CPU %.

ESS or shared storage

For Hadoop over shared storage (such as SAN storage), use the deployment in Figure 10 on page 10.

In this mode, Transparency reads the data from IBM Storage Scale NSD servers and sends them directly through RPC to Hadoop nodes. All data reading/writing on Transparency service nodes goes directly into file system through local disk paths (usually, it is Fiber Channel connections) and you get a better performance. In this mode, the data read flow is: **FC/local disks** > **NSD server** > **Transparency** > **network** > **Hadoop client**.

Note: In this mode, we do not deploy any Hadoop component over IBM Storage Scale NSD servers.

If you cannot install Transparency over IBM Storage Scale NSD servers (for example, ESS), you must install Transparency over IBM Storage Scale clients. If so, the deployment in Figure 5 on page 7 is recommended.

As compared to Figure 10 on page 10, the data read in Figure 5 on page 7 will be: **FC channel/disks** > **NSD server** > **network** > **Transparency** > **lo adapter** > **Hadoop client**.

Storage model consideration

This topic lists the advantages and disadvantages of different storage models.

Table 16. Sharing nothing -vs- Shared storage

Storage models	Advantages	Disadvantages
Sharing Nothing model	• Scaling computing and storage together by adding more nodes • High I/O bandwidth • Data locality for fast read	• Additional network bandwidth from data protection against disk failure or node failure • Low storage efficiency from 3 replica (~33% storage utility)
Shared storage model	• No additional network bandwidth from data protection against disk/node failure • 1 replica, high data efficiency • Scaling computing and storage separately/flexibly	• No data locality • Limited IO bandwidth

Consider the following factors while selecting a storage model:

- If your data is larger than 1 PB, select shared storage or ESS. If you take sharing nothing framework, scanning such a huge data will take up a long time when protecting data against disk failure or node failure.
- If the IBM Storage Scale cluster size is greater than 100 nodes, select shared storage or ESS. More nodes, the high possibility of node being down, and more data restripe for protection.
- If your data size is less than 500 TB but will be scaled to 1 PB+ in short term, select IBM Storage Scale sharing nothing model.
- If your data size is between 500 TB and 1 PB, refer to Table 16 on page 258 and your workloads for decision making.

Hardware configuration planning

This topic describes the hardware configuration to be used for FPO model and IBM Storage Scale client node in shared storage model.

Table 17. Hardware configuration for FPO model

Configuration	Recommended configuration
Network	At least one 10Gb+ ethernet adapter or IB adapters
Disk	SAS disks, 8~12 disks per node

Table 17. Hardware configuration for FPO model (continued)

Configuration	Recommended configuration
Memory	100+ GB per node
CPU	8+ logic processors

Note: For X86_64, with hyper thread on (by default), one physical core is mapped to two logic processors. For ppc64le, SMT 4 is recommended and one physical core is mapped to four logic processors.

Table 18. Hardware configuration for IBM Storage Scale client node in shared storage model

Configuration	Recommended configuration
Network	At least one 10 Gb+ ethernet adapter or IB adapters
Disk	SAS disks, ~ 6 disks for shuffle if possible
Memory	100+ GB per node
CPU	8+ logic processors

Note: It is better to take the same hardware configurations for all Hadoop nodes.

Performance guide

How to change the configuration for tuning

This topic lists the steps to change the configuration for tuning.

Refer to the following table to find the correct configuration directory and take the corresponding steps to update the configuration for tuning:

	Configuration Location	How to change the configuration
HortonWorks HDP	`/etc/hadoop/conf`	• Change the configuration from Ambari for HDFS, Yarn, MapReduce2, Hive, etc. • Restart the services to sync the configuration into `/etc/Hadoop/conf` on each Hadoop node.
Community Apache Hadoop	`$HADOOP_HOME/etc/hadoop`	• Modify the configuration xml files directly. • Take scp to sync the configurations into all other nodes.

	Configuration Location	**How to change the configuration**
Standalone Spark Distro	Refer the guide from your Spark distro	Usually, if taking HDFS schema as data access in Spark, the `hdfs-site.xml` and `core-site.xml` are defined by HADOOP_CONF_DIR in `$SPARK_HOME/spark-env.sh`. If so, you need to modify `hdfs-site.xml` and `core-site.xml` accordingly for tuning and sync the changes to all other Spark nodes.
HDFS Transparency	`/usr/lpp/mmfs/hadoop/etc/hadoop` (for HDFS Transparency 2.7.x) or `/var/mmfs/hadoop/etc/hadoop` (for HDFS Transparency 3.0.x)	• `hdfs-site.xml` and `core-site.xml` should be the same as the configuration for Hadoop or Spark. • If you take HortonWorks HDP, modify the configuration for `gpfs-site.xml` from Ambari/IBM Storage Scale and restart the HDFS service to sync the changes to all HDFS Transparency nodes. • If you take community Apache Hadoop, manually update `gpfs-site.xml` on one HDFS Transparency node and then run `mmhadoopctl connector syncconf /usr/lpp/mmfs/hadoop/etc/hadoop` (for HDFS Transparency 2.7.x) or `mmhadoopctl connector syncconf /var/mmfs/hadoop/etc/hadoop` (for HDFS Transparency 3.0.x) to sync the changes to all HDFS Transparency nodes.

System tuning

Tuning over IBM Storage Scale FPO

Refer to the following sub-sections under the *Configuring FPO* section in the *Administering* section in IBM Storage Scale documentation:

- *Basic Configuration Recommendations*
- *Configuration and tuning of Hadoop workloads*
- *Configuration and tuning of database workloads*
- *Configuration and tuning of SparkWorkloads*

Tuning over ESS/Shared Storage

If the customers deploy ESS, it is tuned after it is setup by the IBM team.

Memory tuning

This topic describes the memory tuning.

Refer the following table to plan your system memory per node:

Table 19. System Memory Allocation

Total Memory per Node	Recommended Reserved System Memory	Recommended Reserved HBase Memory	IBM Storage Scale Pagepool	Transparency NameNode Heap Size	Transparency DataNode Heap Size
16 GB	2 GB	2 GB	4GB	2GB+	2GB
24 GB	4 GB	4 GB	6GB	2GB+	2GB
48 GB	6 GB	8 GB	12GB	2GB+	2GB
64 GB	8 GB	8 GB	16GB	2GB+	2GB
72 GB	8 GB	8 GB	18GB	2GB+	2GB
96 GB	12 GB	16 GB	20GB	2GB+	2GB
128 GB	20 GB	24 GB	20GB	2GB+	2GB
256 GB	32 GB	32 GB	20GB	2GB+	2GB
512 GB	64 GB	64 GB	20GB	2GB+	2GB

Note: For detailed memory requirements, see "Recommended hardware resource configuration" on page 16.

HDFS Transparency DataNode service is a lightweight daemon and does not need a lot of memory.

For HDFS Transparency NameNode, when ranger support is enabled (by default, it is enabled and you could turn it off by setting **gpfs.ranger.enabled**=*false* in gpfs-site.xml), Transparency NameNode will cache inode information. Therefore, if your Transparency NameNode heap size is very small, JVM garbage collection is executed frequently. Usually, it is 2GB and you could increase this up to 4GB if you have a large set of files in your file system.

Note:

- From HDFS Transparency 3.1.0-6 and 3.1.1-3, ensure that the **gpfs.ranger.enabled** field is set to *scale*. The scale option replaces the original *true/false* values.
- Transparency NameNode does not manage FSImage as native HDFS does. It does not need large memory for large number of files as native HDFS.

The pagepool (memory cache for IBM Storage Scale) size for IBM Storage Scale is recommended for most cases in production. However, if you mainly run HBase and want HBase of the best performance, follow section 4.6 Tuning HBase/YCSB.

Table 20. How to change the memory size

Configuration	Guide
IBM Storage Scale pagepool size	mmchconfig pagepool=XG -N <node1,node2…> Need to restart IBM Storage Scale daemon to make the change effective.

Table 20. How to change the memory size (continued)

Configuration	Guide
Transparency NameNode Heap Size	**Ambari GUI** > **HDFS** > **Configs**, change the value of **`NameNode Java heap size`**. If you take community Hadoop, modify the variable **`HADOOP_NAMENODE_OPTS`** in `$HODOOP_HOME_DIR/etc/hadoop/hadoop-env.sh`. For example, `add -Xms2048m -Xmx2048m` to set the heap size as 2GB. If the option **`-Xms`** and **`-Xmx`** have been there, you can modify the number (For example, 2048 for the example) directly. Need to restart HDFS Transparency to make the change effective.
Transparency DataNode Heap size	**Ambari GUI** > **HDFS** > **Configs**, change the value of **`DataNode maximum Java heap size`**. If you take community Hadoop, modify the variable **`HADOOP_DATANODE_OPTS`** in `$HODOOP_HOME_DIR/etc/hadoop/hadoop-env.sh`. For example, `add -Xms2048m -Xmx2048m` to set the heap size as 2GB. If the option **`-Xms`** and **`-Xmx`** have been there, you could modify the number (For example, 2048 for the example) directly. Need to restart HDFS Transparency to make the change effective.

HDFS Transparency Tuning

Tuning for IBM Storage Scale FPO

If you deploy the Hadoop cluster through Ambari (for both HortonWorks and IBM BigInsights IOP), Ambari will do some default tuning according to your cluster and you do not need to do special tuning for HDFS Transparency.

The following table lists the most important configurations that we need to check for running HDFS Transparency over IBM Storage Scale FPO:

Table 21. Tuning configurations for Transparency over IBM Storage Scale FPO

Configurations	Default value	Recommended Value	Comments
`dfs.replication`	3	Keep consistent with your file system default data replica	Usually, it will be 3.

Table 21. Tuning configurations for Transparency over IBM Storage Scale FPO (continued)

Configurations	Default value	Recommended Value	Comments
dfs.blocksize	134217728	Chunk size	It should be blockGroupFactor * blocksize. Usually, it is recommended as 128 * 2MB or 256 * 2MB.
io.file.buffer.size	4096	Data block size of your file system or integral multiple of data block size of your file system but <= 1M	The max value should be <= 1M. If the value is too high, more JVM GC operations will occur.
dfs.datanode.handler.count	10	200	If you have more than 10 physical disks per node, you could increase this. For example, if you have 20 physical disks per node, increase it to 400.
dfs.namenode.handler.count	10	Refer the comments	This depends on the resource of NameNode and the Hadoop node number. If taking IBM Storage Scale Ambari Integration, 100 * loge (DataNode-Number) is used to calculate the value for **dfs.namenode.handler.count**. If not taking IBM Storage Scale Ambari integration, you could take 400 for ~10 Hadoop nodes; take 800 or higher for ~20 Hadoop nodes.
dfs.ls.limit	1000	100000	
dfs.client.read.shortcircuit.streams.cache.size	4096	Refer the comments	Change it as the IBM Storage Scale file system data blocksize.
dfs.datanode.transferTo.allowed	true	false	If this is true, the IO will be 4K mmap () for gpfs.

Note: If **dfs.datanode.handler.count** is very small, you might see socket time when HDFS Client connects to the DataNode.

If **nofile** and **noproc** from **ulimit** is less than 64K, you might see socket connection timeout. By default, **dfs.client.socket-timeout** is *60000 ms*. If your cluster is busy (for example, doing benchmark), you could configure this into *300000 ms* and configure **dfs.datanode.socket.write.timeout** into *600 seconds* (by default, it is *480000 ms*).

The above tuning should also be done for Hadoop HDFS client. If you take HortonWorks HDP, change the above configuration on Ambari GUI. After these changes, restart all services and ensure that these changes are synced into `/etc/hadoop/conf/hdfs-site.xml` and `/usr/lpp/mmfs/hadoop/etc/hadoop/hdfs-site.xml` (for HDFS Transparency 2.7.x) or `/var/mmfs/hadoop/etc/hadoop/hdfs-site.xml` (for HDFS Transparency 3.0.x). If you take open source Apache Hadoop, you need to update these configurations for Hadoop clients (`$HADOOP_HOME/etc/hadoop/hdfs-site.xml`) and take `/usr/lpp/mmfs/bin/mmhadoopctl` to sync your changes to HDFS Transparency configuration on all HDFS Transparency nodes.

Tuning for IBM Storage Scale over shared storage or ESS

If you deploy the Hadoop cluster through Ambari (for both HortonWorks and IBM BigInsights IOP), Ambari will do some default tuning according to your cluster.

The following table lists the most important configurations that we need to check for running HDFS Transparency over IBM Storage Scale ESS or shared storage:

Table 22. Tuning configurations for Transparency over IBM ESS or shared storage

Configurations	Default value	Recommended Value	Comments
dfs.replication	1	3, at least more than 1	Also set **gpfs.storage.type** to *shared*. For more information, see "Configure storage type data replication" on page 59.
dfs.blocksize	134217728	536870912	
io.file.buffer.size	4096(bytes)	Data block size of your file system or integral multiple of data block size of your file system but <= 1M	The max value should be <= 1M. If the value is too high, more JVM GC operations will occur.
dfs.datanode.handler.count	10	Refer the comments	Calculate this according to Hadoop node number and Transparency DataNode number: (40 * HadoopNodes)/TransparencyDataNodeNumber

Table 22. Tuning configurations for Transparency over IBM ESS or shared storage (continued)

Configurations	Default value	Recommended Value	Comments
dfs.namenode.handler.count	10	Refer the comments	This depends on the resource of NameNode and the Hadoop node number. If taking IBM Storage Scale Ambari Integration, 100 * loge (DataNodeNumber) is used to calculate the value for **dfs.namenode.handler.count**. If not taking IBM Storage Scale Ambari integration, you could take 400 for ~10 Hadoop nodes; take 800 or higher for ~20 Hadoop nodes.
dfs.ls.limit	1000	100000	
dfs.client.read.shortcircuit.streams.cache.size	4096	Refer the comments	Change it as the IBM Storage Scale file system data blocksize.
dfs.datanode.transferTo.allowed	true	false	If this is true, the IO will be 4K mmap() for gpfs.

The above tuning should also be done for Hadoop HDFS client. If you take HortonWorks HDP, change the above configuration on Ambari GUI. After these changes, you need to restart all services and ensure that these changes are synced into `/etc/hadoop/conf/hdfs-site.xml` and `/usr/lpp/mmfs/hadoop/etc/hadoop/hdfs-site.xml` (for HDFS Transparency 2.7.x) or `/var/mmfs/hadoop/etc/hadoop/hdfs-site.xml` (for HDFS Transparency 3.0.x). If you take open source Apache Hadoop, you need to update these configurations for Hadoop clients (`$HADOOP_HOME/etc/hadoop/hdfs-site.xml`) and take `/usr/lpp/mmfs/bin/mmhadoopctl` to sync your changes to HDFS Transparency configuration on all HDFS Transparency nodes.

Special tuning for IBM Storage Scale

For better performance, if all the Hadoop nodes are in IBM Storage Scale cluster (as IBM Storage Scale client or NSD server or FPO nodes), you should enable Hadoop short circuit read from HDFS Transparency 2.7.3 as this will bring data read performance. From HDFS Transparency 2.7.3-1, short circuit write is enabled to improve the data write performance when short circuit read is enabled.

If you take Hadoop distro, enable short circuit write from Ambari/HDFS. If you take open source Apache or Spark standalone distro, see Chapter 2, “IBM Storage Scale support for Hadoop,” on page 3 to enable the short circuit read.

If you do not run the Apache Ranger, disable the Ranger support and see Chapter 2, “IBM Storage Scale support for Hadoop,” on page 3.

If you enable HA, and the IO stress from Hadoop cluster is heavy, configure new service RPC port to avoid unnecessary active NameNode switch. Change the following configurations in `hdfs-site.xml`:

Table 23. Configurations in hdfs-site.xml

HA mode	Configuration	Recommendation	Comment
HA or non-HA	**dfs.namenode.service.handler.count**	80	
Non-HA	**dfs.namenode.servicerpc-address**	**<namenode>**: *8060*	
NameNode HA	**dfs.namenode.lifeline.rpc-address.<yourcluster>.nn1**	**<namenode1>** :*8052*	Change <yourcluster> into your HA cluster ID and **nn1** should be matched with **dfs.ha.namenodes.<yourcluster>**.
	dfs.namenode.lifeline.rpc-address.<yourcluster>.nn2	**<namenode2>** :*8052*	Change <yourcluster> into your HA cluster ID and **nn2** should be matched with **dfs.ha.namenodes.<yourcluster>**.
	dfs.namenode.servicerpc-address.<yourcluster>.nn1	**<namenode1>** :*8060*	Change <yourcluster> into your HA cluster ID and **nn1** should be matched with **dfs.ha.namenodes.<yourcluster>**.
	dfs.namenode.servicerpc-address.<yourcluster>.nn2	**<namenode2>** :*8060*	Change <yourcluster> into your HA cluster ID and **nn2** should be matched with **dfs.ha.namenodes.<yourcluster>**.

Setting up the NameNode lifeline and servicerpc

To set up the NameNode lifeline and servicerpc, perform the following:

1. Stop all the Hadoop services. If you are using Ambari UI, stop the services using the Ambari GUI.
2. Add the following parameters to hdfs-site. If you are using Ambari, under customer hdfs-site, set the **dfs.namenode.lifeline/dfs.namenode.servicerpc.address** port to a port that is available to be used on the host.

   For example, by default Yarn Resource Manager: yarn.resourcemanager.address=<resource_mgr_hostname>:8050 uses the port 8050 from the Ambari GUI. If the Yarn Resource Manager is located on the same node as the NameNode, then the **dfs.namenode.lifeline** and **servicerpc-address** port needs to be changed to a used port, like 8051.

   Set the appropriate values for your environment.

   In hdfs-site:

   ```
   dfs.namenode.lifeline.rpc-address.x86.nn1 = c902f10x13.gpfs.net:8052
   dfs.namenode.lifeline.rpc-address.x86.nn2 = c902f10x14.gpfs.net:8052
   dfs.namenode.lifeline.handler.count = 80
   dfs.namenode.lifeline.rpc-bind-host = 0.0.0.0
   dfs.namenode.servicerpc-address.x86.nn1 = c902f10x13.gpfs.net:8062
   dfs.namenode.servicerpc-address.x86.nn2 = c902f10x14.gpfs.net:8062
   dfs.namenode.service.handler.count = 80
   dfs.namenode.servicerpc-bind-host = 0.0.0.0
   ```
3. Start only the zookeeper servers.

4. As HDFS on a NameNode, run the following command:

   ```
   sudo su - hdfs
   ktname=/etc/security/keytabs/hdfs.headless.keytab;kinit `klist -k ${ktname} |tail -1|awk
   ‘{print $2}’` -kt ${ktname}
   /usr/lpp/mmfs/hadoop/bin/hdfs --config /var/mmfs/hadoop/etc/hadoop/ zkfc -formatZK
   ```

   By default, the HDFS Transparency hdfs shell (`/usr/lpp/mmfs/hadoop/bin/hdfs`) uses the HDFS Transparency conf (`/var/mmfs/hadoop/etc/hadoop`). If needed, you can set the **`--config`** parameter when you are executing the `/usr/lpp/mmfs/hadoop/bin/hdfs zkfc -formatZK` command. The `/usr/lpp/mmfs/hadoop/bin/hdfs --config /var/mmfs/hadoop/etc/hadoop/ zkfc -formatZK` command is used to format the Zookeeper instance **`zkfc`**(ZKFailoverController) component and not the HDFS Transparency NameNodes and DataNodes. The newly added parameters in `hdfs-site.xml` are effective only after the HDFS Transparency is started in the next step.

5. Start HDFS.

   ```
   [root@c902f10x13 ~]# mmhadoopctl connector status
   c902f10x14.gpfs.net: namenode pid is 27467
   c902f10x13.gpfs.net: namenode pid is 2215
   c902f10x14.gpfs.net: datanode pid is 24480
   c902f10x16.gpfs.net: datanode pid is 4891
   c902f10x13.gpfs.net: datanode pid is 31069
   c902f10x15.gpfs.net: datanode pid is 19421
   c902f10x14.gpfs.net: zkfc pid is 24841
   c902f10x13.gpfs.net: zkfc pid is 31646
   ```

6. Validate the Lifeline and Servicerpc setup.

   ```
   [root@c902f10x13 ~]# lsof -P -p2215 | grep LISTEN
   java    2215 root  204u     IPv4         1809998714       0t0        TCP
   c902f10x13.gpfs.net:50070 (LISTEN)
   java    2215 root  224u     IPv4         1810038823       0t0        TCP *:8062 (LISTEN)
   java    2215 root  234u     IPv4         1810038827       0t0        TCP *:8052 (LISTEN)
   java    2215 root  244u     IPv4         1810038831       0t0        TCP
   c902f10x13.gpfs.net:8020 (LISTEN)

   #From the active NameNode log:
   STARTUP_MSG:   build = https://github.com/apache/hadoop -r
   16b70619a24cdcf5d3b0fcf4b58ca77238ccbe6d; compiled by 'centos' on 2018-03-30T00:00Z
   STARTUP_MSG:   java = 1.8.0_112
   ************************************************************/
   2020-09-03 22:54:59,780 INFO  namenode.NameNode (LogAdapter.java:info(51)) - registered UNIX
   signal handlers for [TERM, HUP, INT]
   2020-09-03 22:54:59,784 INFO  namenode.NameNode (NameNode.java:createNameNode(1654)) -
   createNameNode []
   2020-09-03 22:55:00,016 INFO  impl.MetricsConfig (MetricsConfig.java:loadFirst(121)) -
   loaded properties from hadoop-metrics2.properties
   2020-09-03 22:55:00,247 INFO  timeline.HadoopTimelineMetricsSink
   (HadoopTimelineMetricsSink.java:init(85)) - Initializing Timeline metrics sink.
   2020-09-03 22:55:00,248 INFO  timeline.HadoopTimelineMetricsSink
   (HadoopTimelineMetricsSink.java:init(105)) - Identified hostname = c902f10x13.gpfs.net,
   serviceName = namenode
   2020-09-03 22:55:00,340 INFO  timeline.HadoopTimelineMetricsSink
   (HadoopTimelineMetricsSink.java:init(138)) - No suitable collector found.
   2020-09-03 22:55:00,342 INFO  timeline.HadoopTimelineMetricsSink
   (HadoopTimelineMetricsSink.java:init(190)) - RPC port properties configured: {8020=client,
   8062=datanode, 8052=healthcheck}
   ……
   2020-09-03 22:55:04,503 INFO  namenode.NameNode (NameNode.java:setServiceAddress(515)) -
   Setting ADDRESS c902f10x13.gpfs.net:8062
   2020-09-03 22:55:04,503 INFO  namenode.NameNode (NameNodeRpcServer.java:<init>(400)) -
   Lifeline RPC server is binding to 0.0.0.0:8052

   # If enabled related DEBUG log:
   2020-09-03 23:00:36,445 DEBUG ipc.Server (Server.java:processResponse(1461)) -
   IPC Server handler 7 on 8062: responding to Call#222 Retry#0
   org.apache.hadoop.hdfs.server.protocol.DatanodeProtocol.sendHeartbeat from 172.16.1.95:42340
   2020-09-03 23:00:36,445 DEBUG ipc.Server (Server.java:processResponse(1480)) -
   IPC Server handler 7 on 8062: responding to Call#222 Retry#0
   org.apache.hadoop.hdfs.server.protocol.DatanodeProtocol.sendHeartbeat from 172.16.1.95:42340
   Wrote 50 bytes.
   2020-09-03 23:00:36,494 DEBUG ipc.Server (Server.java:processOneRpc(2348)) -  got #680
   2020-09-03 23:00:36,494 DEBUG ipc.Server (Server.java:run(2663)) - IPC Server handler
   20 on 8052: Call#680 Retry#0 org.apache.hadoop.ha.HAServiceProtocol.getServiceStatus from
   ```

```
172.16.1.91:53314 for RpcKind RPC_PROTOCOL_BUFFER
2020-09-03 23:00:36,495 DEBUG namenode.GPFSPermissionChecker
(GPFSPermissionChecker.java:<init>(64)) - caller user: root isSuper: true fsowner: root
supergroups: [] + hdfs
2020-09-03 23:00:36,495 DEBUG ipc.Server (ProtobufRpcEngine.java:call(549)) - Served:
getServiceStatus, queueTime= 0 procesingTime= 1
2020-09-03 23:00:36,495 DEBUG ipc.Server (Server.java:processResponse(1461)) -
IPC Server handler 20 on 8052: responding to Call#680 Retry#0
org.apache.hadoop.ha.HAServiceProtocol.getServiceStatus from 172.16.1.91:53314
2020-09-03 23:00:36,495 DEBUG ipc.Server (Server.java:processResponse(1480)) -
IPC Server handler 20 on 8052: responding to Call#680 Retry#0
org.apache.hadoop.ha.HAServiceProtocol.getServiceStatus from 172.16.1.91:53314 Wrote 37
bytes.
```

Note:

1. The **`dfs.namenode.lifeline.rpc-address`** configuration requires you to restart the NameNodes, DataNodes and ZooKeeper Failover Controllers. For more information, see Scaling the HDFS NameNode (part 2).
2. The **`dfs.namenode.servicerpc-address`** configuration requires you to reset the ZooKeeper Failover Controllers as per the following Cloudera documentation: How do you enable NameNode service RPC port without HDFS service downtime?

   Format the zookeeper data structure by using the following command:

   ```
   /usr/lpp/mmfs/hadoop/bin/hdfs --config /var/mmfs/hadoop/etc/hadoop/zkfc -formatZK
   ```

Buffered logging and filtering

Learn to configure the Apache Log4j utility's buffering and filtering features, which have demonstrated reduction of load on the logging subsystem, indirectly increasing HDFS Transparency throughput.

By enabling HDFS Transparency log buffering, the I/O performance increases. This increased performance occurs because the HDFS workloads are not blocked by log messages being flushed to disk. If log messages are flushed to disk, it can adversely affect the performance of workloads in a busy cluster.

To mitigate said adverse effects, a buffered logging based on Apache Log4j was introduced since HDFS Transparency versions 3.2.2-6 and 3.1.1-15. Enabling buffered logging provides better HDFS Transparency throughput in comparison to unbuffered logging, if enabling logging is a requirement in order to debug any issue or otherwise.

Steps to configure buffered logging

1. In the `/var/mmfs/hadoop/etc/hadoop/log4j.properties` configuration file, add or update the following lines:

   ```
   log4j.appender.RFA=org.apache.hadoop.hdfs.server.namenode.GPFSRollingFileAppender
   log4j.appender.RFA.bufferedIO=true
   log4j.appender.RFA.bufferSize=8192
   ```

   Note: Comment out the earlier Apache Log4j appender.
2. Upload configurations to IBM Storage Scale CCR by issuing the following command:

   ```
   # mmhdfs config upload
   ```
3. If Cloudera is integrated, restart HDFS Transparency from **Cloudera Manager**.

   Otherwise, restart HDFS by using this command:

   ```
   # mmhdfs hdfs restart
   ```

Steps to configure filtering

You can selectively filter out log messages being logged into the log file. The filtering feature allows you to suppress messages that usually are not needed, for example:

```
log4j.appender.RFA.filter.1=org.apache.log4j.varia.StringMatchFilter
log4j.appender.RFA.filter.1.StringToMatch=Removing expired token
log4j.appender.RFA.filter.1.AcceptOnMatch=false
```

With this configuration, the log messages that include the text "Removing expired token" are not logged. For information about other filters, see the Apache Log4j template file (`/usr/lpp/mmfs/hadoop/template/log4j.properties.template`).

Hadoop/Yarn tuning

Common Tuning

Tuning for Yarn comprises of two parts: tuning MapReduce2 and tuning Yarn.

Follow the configurations in to tune MapReduce2 and Table 24 on page 269 to tune Yarn.

Table 24. Tuning MapReduce2

Configurations	Comments
mapreduce.map.memory.mb	Default: 1024 MB Recommended: 2048 MB
mapreduce.reduce.memory.mb	Default: 1024 MB Recommended: 4096 MB or larger The value could be considered according to the **yarn.nodemanager.resource.memory-mb** and the current task number on one node. For example, if you configure 100 GB for **yarn.nodemanager.resource.memory-mb** and you have
mapreduce.map.java.opts	Recommended: 75% * mapreduce.map.memory.mb or 80% * mapreduce.map.memory.mb
mapreduce.reduce.java.opts	Recommended: 75% * mapreduce.reduce.memory.mb or 80% * mapreduce.reduce.memory.mb.
mapreduce.job.reduce.slowstart.completedmaps	Default: 0.05 Different Yarn jobs could take different value for this configuration. You could specify this value when submitting Yarn job if your job wants to take different value for this.
mapreduce.map.cpu.vcores	1
mapreduce.reduce.cpu.vcores	1
mapreduce.reduce.shuffle.parallelcopies	Default: 5 Recommend: 30+

Table 24. Tuning MapReduce2 (continued)

Configurations	Comments
mapreduce.tasktracker.http.threads	Default: 40 If your cluster has more than 40 nodes, you could increase this to ensure that the reduce task on each host could have at least 1 thread for shuffle data copy.
yarn.app.mapreduce.am.job.task.listener.thread-count	Default: 30 If you have larger cluster for job (for example. your cluster is larger than 20 nodes and 16 logic processors per node) you could increase this to try.
mapreduce.task.io.sort.mb	Default: 100(MB) Recommended: 70% * mapreduce.map.java.opts
mapreduce.map.sort.spill.percent	Default: 80 Take default value and not change this.
mapreduce.client.submit.file.replication	Default: 10 Change it as the default replica of your IBM Storage Scale file system (check this by `mmlsfs <your-fs-name> -r`).
mapreduce.task.timeout	Default: 300000ms Change it into 600000s if you are running benchmark.

The following configurations are not used by Yarn and you do not need to change them:

```
mapreduce.jobtracker.handler.count
mapreduce.cluster.local.dir
mapreduce.cluster.temp.dir
```

Tuning Yarn

This section describes the configurations to be followed for tuning Yarn.

Table 25. Configurations for tuning Yarn

Configurations	Default	Recommended value	Comments
Resource Manager Heap Size (resourcemanager_heapsize)	1024	1024	
NodeManager Heap Size (nodemanager_heapsize)	1024	1024	
yarn.nodemanager. **resource.memory-mb**	8192	Refer comments	The total memory that could be allocated for Yarn jobs.

Table 25. Configurations for tuning Yarn (continued)

Configurations	Default	Recommended value	Comments
yarn.scheduler.minimum-allocation-mb	1024	Refer comments	This value should not be greater than **mapreduce.map.memory.mb** and **mapreduce.reduce.memory.mb**. And, **mapreduce.map.memory.mb** and **mapreduce.reduce.memory.mb** must be the multiple times of this value. For example, if this value is 1024MB, then, you cannot configure **mapreduce.map.memory.mb** as *1536 MB*.
yarn.scheduler.maximum-allocation-mb	8192	Refer comments	This value should not be smaller than **mapreduce.map.memory.mb** and **mapreduce.reduce.memory.mb**.
yarn.nodemanager.local-dirs	`${hadoop.tmp.dir}/nm-local-dir`	Refer comments	`hadoop.tmp.dir` is `/tmp/hadoop-${user.name}` by default. This will impact the shuffle performance. If you have multiple disks/partitions for shuffle, you could configure this as: `/hadoop/local/sd1`, `/hadoop/local/sd2`, `/hadoop/local/sd3`. If you have more disks for this configuration, you will have more IO bandwidth for Yarn's intermediate data.
yarn.nodemanager.log-dirs	`${yarn.log.dir}/userlogs`	Refer comments.	These directories are for Yarn job to write job task logs. It does not need a lot of bandwidth and therefore you can configure one directory for this configuration. For example, `/Hadoop/local/sd1/logs`.

Table 25. Configurations for tuning Yarn (continued)			
Configurations	**Default**	**Recommended value**	**Comments**
yarn.nodemanager. **resource.cpu-vcores**	8	Logic processor number	Configure this as the logic processor number. You could check the logic processor number according to /proc/cpuinfo. This is the common rule. However, if you run the job which takes all vcores of all nodes and the CPU% is not higher than 80%, you could increase this (for example, 1.5 X logic-processor-number). If you will run CPU sensitive workloads, keep the ratio of physical_cpu/vcores as 1:1. If you will run IO bound workloads, you could change this as 1:4. If you do not know your workloads, keep this as 1:1.
yarn.scheduler.minimum-allocation-vcores	1	1	
yarn.scheduler.maximum-allocation-vcores	32	1	
yarn.app.mapreduce.am. **resource.mb**	1536	Refer the comments	Configure this as the value for **yarn.scheduler.minimum-allocation-mb**. Usually, 1GB or 2GB is enough for this. **Note:** This is under MapReduce2.
yarn.app.mapreduce.am. **resource.cpu-vcores**	1	1	
Compression			
mapreduce.map.output. **compress**	false	true	Make the output of Map tasks compressed. Usually, this means to compress the Yarn's intermediate data.
mapreduce.map.output. **compress.codec**	**org.apache.hadoop.io.** **compress.DefaultCodec**	**org.apache.hadoop.io.** **compress.SnappyCodec**	

Configurations	Default	Recommended value	Comments
Table 25. Configurations for tuning Yarn (continued)			
mapreduce.output. **fileoutputformat.** **compress**	false	true	Make the job output compressed.
mapreduce.output. **fileoutputformat.compress.codec**	**org.apache.hadoop.io.** **compress.DefaultCodec**	**org.apache.hadoop.io.** **compress.GzipCodec**	
mapreduce.output. **fileoutputformat.compress.type**	RECORD	BLOCK	Specify the
Scheduling			
yarn.scheduler.capacity. **resource-calculator**	**org.apache.hadoop.yarn.util.** **resource.** **DefaultResourceCalculator**	Refer the comments	By default, it is **org.apache.hadoop.yarn.util.** **resource.DefaultResourceCalculator** which only schedules the jobs according to memory calculation. If you want to schedule the job according to memory and CPU, you could enable CPU scheduling from **Ambari** > **Yarn** > **Config**. After you enable CPU scheduling, the value will be **org.apache.hadoop.yarn.util.** **resource.DominantResourceCalculator**. If you find that your cluster node has 100% CPU% after taking default configuration, you could try to limit the concurrent tasks by enabling CPU scheduling. If not, no need to change this.

Specific tuning of Yarn for ESS/Shared Storage

This section describes the Yarn configurations to be tuned for ESS/Shared storage.

If you are running Hadoop over shared storage or ESS, the following must be tuned:

Configurations	default	Recommended value	Comments
yarn.scheduler.capacity.node-locality-delay	40	-1	Change this as *-1* to disable the locality-based scheduling. If not changing this, you will only have 40 concurrent tasks running over the cluster no matter the number of nodes in your cluster. Change this from **Ambari GUI** > **Yarn** > **Config** > **Advanced** > **Scheduler; Capacity Scheduler**.

Maximal Map and Reduce Task Calculation

This topic describes the calculations for Maximal Map and Reduce Task.

Value	Calculation
MaxMapTaskPerWave_mem	(yarn.nodemanager.resource.memory-mb * YarnNodeManagerNumber - yarn.app.mapreduce.am.resource.mb)/ mapreduce.map.memory.mb
MaxMapTaskPerWave_vcore	(yarn.nodemanager.resource.cpu-vcores * YarnNodeManagerNumber - yarn.app.mapreduce.am.resource.cpu)/ yarn.scheduler.minimum-allocation-vcores
TotalMapTaskPerWave	Equal to MaxMapTaskPerWave_mem by default
MaxReduceTaskPerNode_mem	(yarn.nodemanager.resource.memory * YarnNodeManagerNumber - yarn.app.mapreduce.am.resource.mb)/ mapreduce.reduce.memory.mb
MaxReduceTaskPerNode_vcore	(yarn.nodemanager.resource.cpu-vcores * YarnNodeManagerNumber - yarn.app.mapreduce.am.resource.cpu)/ yarn.scheduler.minimum-allocation-vcores
TotalReduceTaskPerWave	Equal to MaxReduceTaskPerNode_mem by default

If **yarn.scheduler.capacity.resource-calculator** is not changed, by default, **MaxMapTaskPerWave_mem** will take effective. Under this situation, if the **MaxMapTaskPerWave_vcore** is more than 2 times of **MaxMapTaskPerWave_mem**, you still have a lot of CPU resource and you could increase **MaxMapTaskPerWave_mem** by either increasing **yarn.nodemanager.resource.memory-mb** or decreasing **mapreduce.map.memory.mb**. However, if **MaxMapTaskPerWave_vcore** is smaller than **MaxMapTaskPerWave_mem**, means more than 1 task will run on the same logic processor and this might bring additional context switch cost.

It is similar for **MaxReduceTaskPerNode_mem** and **MaxReduceTaskPerNode_vcore**.

TotalMapTaskPerWave is the totally concurrent task number that you could run in one wave. **TotalReduceTaskPerWave** is the totally concurrent task number that you could run in one wave.

Performance sizing

A lot of factors, such as logic processor number, memory size, network bandwidth, storage bandwidth and the IBM Storage Scale deployment mode, can impact performance sizing. This section gives a brief throughput estimate sizing guide for teragen and terasort workloads as the query and transaction workload types (HBase, Hive) have too many factors to be able to give sizing rules.

Sizing the throughput from HDFS Transparency cluster could be done by two steps:

1. Sizing the throughput from IBM Storage Scale POSIX interface.
2. Calculating the throughput from HDFS Transparency.

To size the throughput of IBM Storage Scale POSIX interface, use the open source IOR benchmark to get the throughput of reads and writes from the POSIX interface. If you are not able to use IOR benchmark, estimate the throughput for IBM Storage Scale POSIX interface as follows:

- For IBM ESS, get the throughput number from the IBM product guide.
- For IBM Storage Scale FPO:
  - If the network bandwidth is greater than (Disk-number-per-node * disk-bandwidth), calculate using:

    ((Disk-number * Disk-bandwidth / Replica-number) * 0.7)
  - If network bandwidth is smaller than (Disk-number-per-node * disk-bandwidth), calculate using:

    ((Network-bandwidth-per-node * node-number) * 0.7)

Usually, it is recommended to take SSD for metadata so that the metadata operations in IBM Storage Scale FPO do not become the bottleneck. Under this condition, HDFS Transparency interface will yield approximately 70% to 80% throughput based on the POSIX interface throughput value. The benchmark throughput is impacted by the number of Hadoop nodes and the Hadoop-level configuration settings.

As for how to size Hadoop node number,

- For ESS:

  Calculate Hadoop node number by using the ESS official throughput value and the client network bandwidth:

  For example, if using ESS GL4s and the throughput is 36GB from IBM product guide, and the client has 10Gb network bandwidth, then will need 36GB/((10Gb/8) * 0.8) clients to drive the throughput.

  OR

  Calculate based only on the network bandwidth from the ESS configuration and the client network adapter throughput:

  For example, ESS configuration network bandwidth (e.g. 100GB) and the client network adapter throughput (e.g. 10GB), then will need (100GB/10GB) = 10 clients.
- For FPO:

  All Hadoop nodes should be IBM Storage Scale FPO nodes.

Teragen

When benchmarking Teragen, execute the following command:

```
hadoop jar /usr/hdp/current/hadoop-mapreduce-client/hadoop-mapreduce-examples.jar
teragen -Dmapreduce.job.maps=<JOB_MAP_NUMBER> -Ddfs.blocksize=<BLOCKSIZE>
<DATA_RECORDS>  /<OUTPUT_PATH>
```

In the above command, **<DATA_RECORDS>** is used to specify the number of records in your evaluation. One record is 100 bytes. So, 1000 000 000 0 records mean the data size 1000 000 000 0 * 100 bytes, which is closed to 1TB.

<OUTPUT_PATH> is the data output directory. Change this accordingly.

Teragen has no Reduce task and the value **<JOB_MAP_NUMBER>** is the one you need to plan with some efforts. Refer the **yarn.nodemanager.resource.memory-mb**, **yarn.scheduler.minimum-allocation-vcores**, **yarn.app.mapreduce.am.resource.cpu-vcores**, **yarn.app.mapreduce.am.resource.mb** in Table 25 on page 270.

The **<JOB_MAP_NUMBER>** should be equal to (MaxTaskPerNode_mem * YarnNodeManagerNumber - 1). Also, the value ((<DATA_RECORDS> * 100)/(1024*1024))/ <JOB_MAP_NUMBER> should not be very small and it should be close to **dfs.blocksize** or multiple times of **dfs.blocksize**. If the value ((<DATA_RECORDS> * 100)/(1024*1024))/ <JOB_MAP_NUMBER> is very small, your **<DATA_RECORDS>** is very small for your cluster.

If **yarn.scheduler.capacity.resource-calculator** is changed by enabling CPU scheduling from Ambari, the smaller value between **MaxTaskPerNode_mem** and **MaxTaskPerNode_vcore** will be effective. Under this situation, you could try to make **MaxTaskPerNode_vcore** and **MaxTaskPerNode_mem** close to each other. If not, that usually indicates you have free resources that are not yet utilized.

If you take the same block size (**dfs.blocksize** from core-site.xml) for your TeraGen job, you do not need to specify **-Ddfs.blocksize=<BLOCKSIZE>**. If you want to take different **dfs.blocksize** for the job, you could specify the **-Ddfs.blocksize=<BLOCKSIZE>** option.

TeraSort

TeraSort is a typical Map/Reduce job. It will take map tasks and reduce tasks.

Take the following command to run TeraSort:

```
hadoop jar /usr/hdp/current/hadoop-mapreduce-client/hadoop-mapreduce-examples.jar terasort \
-Dmapreduce.job.reduces=<REDUCE_TASKS> \
-Ddfs.blocksize=<DFS_BLOCKSIZE>   \
-Dmapreduce.input.fileinputformat.split.minsize=<DFS_BLOCKSIZE> \
-Dio.file.buffer.size=<IO_BUFFER_SIZE> \
-Dmapreduce.map.sort.spill.percent=0.8 \
-Dmapreduce.reduce.shuffle.merge.percent=0.96 \
-Dmapreduce.reduce.shuffle.input.buffer.percent=0.7 \
-Dmapreduce.reduce.input.buffer.percent=0.96 \
/<TERAGEN_DATA_INPUT> \
/<TERASORT_DATA_OUTPUT>
```

<IO_BUFFER_SIZE> must be equal to your IBM Storage Scale data pool block size (check this by **mmlspool <fs-name> all -L**).

<TERAGEN_DATA_INPUT> and **<TERASORT_DATA_OUTPUT>** must be specified according to your requirements.

<DFS_BLOCKSIZE> could be the default **dfs.blocksize** from hdfs-site.xml. If you want to take different block size, specify it here. For IBM Storage Scale ESS or shared storage, **<DFS_BLOCKSIZE>** cannot be equal to the data pool block size (usually, 1GB block size can give you a good performance). For IBM Storage Scale FPO, the **<DFS_BLOCKSIZE>** must be equal to your file system data blocksize * blockGroupFactor (check these two values from **mmlspool <fs-name> all -L**).

<DFS_BLOCKSIZE> impacts the map task number in your cluster. For the above command (**mapreduce.input.fileinputformat.split.minsize** is specified as **<DFS_BLOCKSIZE>**), the final map task number is calculated according to **<DFS_BLOCKSIZE>**. If you have only one file with size 512MB and you specify 500MB as **<DFS_BLOCKSIZE>**, you will have two splits or map tasks. If you have two 512MB files and you specify 500MB as **<DFS_BLOCKSIZE>**, you will get four splits or map tasks.

Note: The total split number is the final map task number and cannot be changed by options from TeraSort. If you do not follow the above guide, you might get incorrect split number and therefore impact the map phase in Terasort.

The ideal case for each map task, is that the to-be-processed data size is close but not larger than (70% * mapreduce.map.java.opts * 80%) and this could keep the intermediate data size as small as possible in the shuffle of job.

Very small **<DFS_BLOCKSIZE>** makes you have more map tasks. Map task number should be proper for the cluster to execute them in one wave or two waves. You should not execute map tasks in three or more waves because this will slow down the performance.

If the map task number is ((MaxTaskPerNode_mem * YarnNodeManagerNumber) - 1), all these map tasks can be handled in one wave. If the map task number is larger than ((MaxTaskPerNode_mem * YarnNodeManagerNumber) – 1), it should be between 1.75 * (MaxTaskPerNode_mem * YarnNodeManagerNumber) and 1.9 * (MaxTaskPerNode_mem * YarnNodeManagerNumber). You could try different map task number by changing file number from **<TERAGEN_DATA_INPUT>** and **<DFS_BLOCKSIZE>**.

Usually, **<REDUCE_TASKS>** should be executed in one wave. That means, **<REDUCE_TASKS>** should be equal to (TotalReduceTaskPerWave - 1).

DFSIO

DFSIO is shipped with Hadoop distro.

The following are the options for DFSIO:

```
$yarn jar /usr/hdp/current/hadoop-mapreduce-client/hadoop-mapreduce-client-jobclient.jar
TestDFSIO

Usage: TestDFSIO [genericOptions] -read [-random | -backward | -skip [-skipSize Size]] |
-write | -append | -truncate | -clean [-compression codecClassName] [-nrFiles N]
[-size Size[B|KB|MB|GB|TB]] [-resFile resultFileName] [-bufferSize Bytes] [-rootDir]
```

Usually, we care only about the `-read`, `-write`, `-nrFiles`, `-size` and `-bufferSize` options.

The option `-read` is used to evaluate the read performance. The option `-write` is used to evaluate the write performance. The option `-nrFiles` is used to specify the number of files you want to generate. The option `-size` is used to specify the file size for each file.

So, the total data that TestDFSIO will read/write is (nrFiles * size). DFSIO is simple in logic and it will start the nrFiles map tasks running over the whole Hadoop/Yarn cluster.

1st tuning guide: nrFiles and task number

While evaluating TestDFSIO, we need to consider the Yarn's configuration. If the maximum tasks per wave is **`TotalMapTaskPerWave`**, your nrFiles should be **`TotalMapTaskPerWave`**.

If it is IBM Storage Scale FPO, the file size `-size` should be at least 512MB or more (you could try 1GB, 2GB and 4GB). If it is shared storage or ESS, the file size `-size` should be 1GB or more (try 1GB, 2GB and 4GB).

Usually, according to experience, we take as many map tasks as possible for DFSIO read. For DFS write, we recommend to try the map tasks according to logic processors even if you have more free memory.

2nd tuning guide: nrFiles * size

The total data size (nrFiles * size) should be at least 4 times of the total physical memory size of all HDFS nodes. For example, if you want to compare the performance data of DFSIO between native HDFS and IBM Storage Scale, if you have 10 nodes for native HDFS DataNode (100GB physical memory size per DataNode), then your (nrFiles * size) over native HDFS should be 4 * (100GB per node * 10 DataNodes), ~ 4000GB. And then, the same (nrFiles * size) for IBM Storage Scale.

3rd tuning guide: -bufferSize

Try `-bufferSize` with the block size of IBM Storage Scale. This is the IO buffer size for task to write/read data.

TPC-H and TPC-DS for Hive

Tuning for Hive

Hive is Hadoop's SQL interface over HDFS. Therefore, the tuning is very similar for Hive as native HDFS.

For more information, refer Hive's Manual Guide.

The following tunings will be related with IO and therefore impacted by different distributed file system:

Table 26. Hive's Tuning

Configuration	Default Value	Recommended Value	Comments
hive.exec.compress.output	False	True	Compress the Hive's execution results before writing it out.
hive.exec.compress.intermediate	False	True	Compress the Hive's intermediate data before writing it out.
hive.intermediate.compression.codec	N/A	**org.apache.hadoop.io.compress.SnappyCodec**	
hive.intermediate.compression.type	N/A	BLOCK	
hive.exec.reducers.max	1009	Variable	This should be decided according to your cluster and Yarn's configuration.
hive.optimize.sort.dynamic.partition	False	True	When enabled, the dynamic partitioning column will be globally sorted. Therefore, we can keep only one record writer open for each partition value in the reducer, thereby reducing the memory pressure on reducers.

Table 26. Hive's Tuning (continued)

Configuration	**Default Value**	**Recommended Value**	**Comments**
`hive.llap.io.use.fileid.path`	True	False	This configuration is used if LLAP should use fileId (inode) based path to ensure better consistency for the cases of file overwrite. **Note:** HDFS Transparency does not support the use of LLAP inode-based look up because of the reuse of inode values in IBM Storage Scale. Therefore, you must set this configuration parameter to *False*.

If the Hive's job invokes a lot of Map/Reduce and generates a lot of intermediate data or output data, configuring the above will improve the Hive's job execution.

If you configure **`hive.exec.compress.output`** as *true*, you need to check the following configuration in Hadoop Yarn:

- **`mapreduce.output.fileoutputformat.compress`**=*true*
- **`mapreduce.output.fileoutputformat.compress.codec`**=*org.apache.hadoop.io.compress.SnappyCodec*
- **`mapreduce.output.fileoutputformat.compress.type`**=*BLOCK*

You can modify `hive-site.xml` and restart Hive service to make it effective for all the Hive jobs. Else, you can set them in Hive console and they will only be effective for the commands/jobs invoked from the Hive console:

```
#hive
hive> set hive.exec.compress.output=true;
hive> set mapreduce.output.fileoutputformat.compress=true;
hive> set
mapreduce.output.fileoutputformat.compress.codec=org.apache.hadoop.io.compress.SnappyCodec;
hive> set mapreduce.output.fileoutputformat.compress.type=BLOCK;
hive>
```

Running TPC-H/Hive

This section lists the steps for running TPC-H for Hive.

Execute the following steps to run TPC-H for Hive:

1. Download TPC-H from the official website and TPC-H_on_Hive from Running TPC-H queries on Hive.

Download TPC-H_on_Hive_2009-08-14.tar.gz.

2. Untar the above `TPC-H_on_Hive_2009-08-14.tar.gz` into `$TPC_H_HIVE_HOME`
3. Download DBGEN from the TPC-H website:

   Note: Register your information and download the `TPC-H_Tools_v<version>.zip`.
4. Untar `TPC-H_Tools_v<version>.zip` and build dbgen:

```
#unzip TPC-H_Tools_v<version>.zip
```

Assuming it is located under `$TPCH_DBGEN_HOME`

```
#cd $TPCH_DBGEN_HOME
#cd dbgen
#cp makefile.suite makefile
```

Update the following values in `$TPCH_DBGEN_HOME/dbgen/makefile` accordingly:

```
#vim makefile

CC = gcc
DATABASE = SQLSERVER
MACHINE=LINUX
WORKLOAD = TPCH
```

Modify the `$TPCH_DBGEN_HOME/dbgen/tpcd.h`:

```
vim $TPCH_DBGEN_HOME/dbgen/tpcd.h

#ifdef  SQLSERVER
#define GEN_QUERY_PLAN   "EXPLAIN;"
#define START_TRAN       "START TRANSACTION;\n"
#define END_TRAN         "COMMIT;\n"
#define SET_OUTPUT       ""
#define SET_ROWCOUNT     "limit %d;\n"
#define SET_DBASE        "use %s;\n"
#endif
```

Execute **make** to build dbgen:

```
#cd $TPCH_DBGEN_HOME/dbgen/
#make
…
gcc  -g -DDBNAME=\"dss\" -DLINUX -DSQLSERVER -DTPCH -DRNG_TEST
-D_FILE_OFFSET_BITS=64  -O -o qgen build.o bm_utils.o qgen.o
rnd.o varsub.o text.o bcd2.o permute.o speed_seed.o rng64.o -lm
```

5. Generate the transaction data:

```
#cd $TPCH_DBGEN_HOME/dbgen/
#./dbgen -s 500 □this is to generate 500GB data, change the value
accordingly for your benchmark
```

The generated data is stored under `$TPCH_DBGEN_HOME/dbgen/` and all files are named with the suffix `.tbl`:

```
# ls -la *.tbl
-rw-r--r-- 1 root root  24346144 Jul  5 08:41 customer.tbl
-rw-r--r-- 1 root root 759863287 Jul  5 08:41 lineitem.tbl
-rw-r--r-- 1 root root      2224 Jul  5 08:41 nation.tbl
-rw-r--r-- 1 root root 171952161 Jul  5 08:41 orders.tbl
-rw-r--r-- 1 root root 118984616 Jul  5 08:41 partsupp.tbl
-rw-r--r-- 1 root root  24135125 Jul  5 08:41 part.tbl
-rw-r--r-- 1 root root       389 Jul  5 08:41 region.tbl
-rw-r--r-- 1 root root   1409184 Jul  5 08:41 supplier.tbl
```

Note: You will need all the above files. If you find any file missing, you need to regenerate the data.

6. Prepare the data for TPC-H:

```
#cd $TPCH_DBGEN_HOME/dbgen/
#mv *.tbl $TPC_H_HIVE_HOME/data/

#cd $TPC_H_HIVE_HOME/data/
#./tpch_prepare_data.sh
```

Note: Before executing `tpch_prepare_data.sh`, you need to ensure that IBM Storage Scale is active (`/usr/lpp/mmfs/bin/mmgetstate -a`), file system is mounted (`/usr/lpp/mmfs/bin/mmlsmount <fs-name> -L`) and HDFS Transparency is active (`/usr/lpp/mmfs/bin/mmhadoopctl connector getstate`).

7. Check your prepared data:

```
#hadoop dfs -ls /tpch
```

Check that all the files listed in Step 5 are present.

8. Run TPC-H:

```
# cd $TPC_H_HIVE_HOME/
#export HADOOP_HOME= /usr/hdp/current/hadoop-client
#export HADOOP_CONF_DIR=/etc/Hadoop/conf
#export HIVE_HOME= /usr/hdp/current/hive-client

# vim benchmark.conf □ change the variables, such as LOG_FILE, if you want
#./tpch_prepare_data.sh
```

HBase/YCSB

HBase YCSB is a benchmark developed by Yahoo to test the performance of HBase.

When running YCSB to evaluate the HBase performance over IBM Storage Scale, take the tuning accordingly as explained in the following sections.

Tuning for YCSB/HBase

HBase configuration

1. Change the `hbase-site.xml` from Ambari if you take HortonWorks or IBM BigInsights. If you take open source HBase, you could modify `$HBASE_HOME/conf/hbase-site.xml` directly.

Table 27. HBase Configuration Tuning

Configuration	Default Value	Recommended Value	Comments
Java Heap	N/A	Refer the "Memory tuning" on page 261 section.	HBase Master server Heap Size; HBase Region Server Heap Size
hbase.regionserver.handler.count	30	60	
zookeeper.session.timeout	N/A	180000	
hbase.hregion.max.filesize	10737418240	10737418240	Check the default value. If it is not 10GB, change it into 10GB.
hbase.hstore.blockingStoreFiles	10	50	
hbase.hstore.compaction.max	10	10	

Table 27. HBase Configuration Tuning (continued)

Configuration	Default Value	Recommended Value	Comments
hbase.hstore.compaction.max.size	LONG.MAX_VALUE	Variable	If you see a lot of compaction, you could set this to 1GB to exclude those HFiles from compaction.
hbase.hregion.majorcompaction	604800000	0	Turn off the major compaction when running benchmark to ensure that the results are stable. In production, this should not be changed.
hbase.hstore.compactionThreshold	3	3	
hbase.hstore.compaction.max	10	3	

Table 28. IBM Storage Scale Tuning

Configuration	Default Value	Recommended Value	Comments
pagepool	1GB	30% of physical memory	30% of physical memory

Note: 30% of physical memory is only for running HBase/YCSB. In production, you need to consider the memory allocation for other workloads. If you run Map/Reduce jobs, Hive jobs over the same cluster, you need to trace off the performance for these different workloads. If you allocate more memory for pagepool because of HBase, you will have fewer memory for Map/Reduce jobs and therefore degrade the performance for Map/Reduce jobs.

Table 29. YCSB Configuration Tuning

Configuration	Default Value	Recommended Value	Comments
writebuffersize	12MB	12MB	
clientbuffering	False	True	For benchmark, keep this the same as what you use to run YCSB over native HDFS.
recordcount	1000	1000000	
operationcount	N/A	N/A	Depends on the number of operations you want to benchmark. For example, 20M operations
threads	N/A	Variable	Depends on the number of threads you want to benchmark.
requestdistribution	zipfian	Not changed	

Table 29. YCSB Configuration Tuning (continued)

Configuration	Default Value	Recommended Value	Comments
recordsize	100*10	Not changed	YCSB for HBase takes 100 bytes per field and 10 fields for one record.

Important: While creating the HBase table before running YCSB, you need to pre-split the table accordingly. For example, you need to pre-split the table into 100 partitions for ~10 HBase Region servers. If it is more than 10 HBase Region servers, you need to increase the pre-split partition number.

If you do not pre-split the table, all requests are handled by limited HBase Region servers and therefore the performance of YCSB is impacted.

Running YCSB/HBase

This section lists the steps to run YCSB/ HBASE.

1. Download YCSB from YCSB 0.14.0.
2. Untar the YCSB package into $YCSB_HOME.
3. Remove all the libraries shipped by YCSB:

   ```
   #rm -fr $YCSB_HOME/hbase10-binding/lib/*
   ```

4. Copy the libraries from your Hadoop distro:

   The following is for HortonWorks HDP 2.6:

   ```
   cp /usr/hdp/2.6.0.3-8/hbase/lib/* $YCSB_HOME/ hbase10-binding/lib/
   cp /usr/hdp/2.6.0.3-8/Hadoop/*.jar $YCSB_HOME/ hbase10-binding/lib/
   ```

5. Create the following script accordingly:

   The script for loading:

   ```
   vim ycsb_workload_load.sh
   #!/bin/bash
   set -x

   # <script> <thread-number> <YCSB_RESULT_HOME>

   ${YCSB_HOME}/bin/ycsb load hbase10 -P ${YCSB_HOME}/workloads/workloada
   -p columnfamily=family -p recordcount=${RECORD_COUNT} -s -threads $1
   -p measurementtype=timeseries -p timeseries.granularity=2000 2>&1 >
   ${YCSB_RESULT_HOME}/$2/workload_load.output.thread-$1-.`date "+%y%m%d_%H%M%S"`
   ```

   In the previous script, **RECORD_COUNT** is the record number you want to load into HBase. **RECORD_COUNT** should be 1M. **<thread-number>** depends on your running.

   If you want to change **writebuffersize** and **clientbuffering**, you could add -p <writebuffersize> -p <clientbuffering> for the previous YCSB command.

   The script for YCSB workload A/B/C/D/E/F:

   ```
   #vim ycsb_workload_a.sh
   #!/bin/bash
   # <script> <workloadA-recordcount> <thread-number> <result sub dir>

   ${YCSB_HOME}/bin/ycsb run hbase10 -P ${YCSB_HOME}/workloads/workloada
   -p columnfamily=family -p operationcount=${OPERATION_COUNT}
   -p recordcount=$1 -s -threads $2 -p measurementtype=timeseries
   -p timeseries.granularity=2000 2>&1 |
   tee ${YCSB_RESULT_HOME}/$3/workload_a.output.thread-$2.`date "+%y%m%d_%H%M%S"`
   ```

Similarly create the other scripts for workload B/C/D/E/F.

You need to first run the `ycsb_workload_load.sh` script. After it loads data into HBase, run `ycsb_workload_a.sh` or other scripts.

Spark

Spark tuning (HortonWorks or IBM Spectrum Conductor)

If you run Spark standalone distro (for example, community Spark, or IBM Spectrum Conductor™ with Spark), it is recommended to take IBM Storage Scale POSIX interface.

If you run Spark standalone distro on IBM Storage Scale FPO, refer the "Tuning for IBM Storage Scale FPO" on page 262 section.

If you run Spark standalone distro on IBM ESS, no need to tune further.

If you take Hadoop distro (for example, HortonWorks HDP), it is recommended to take IBM Storage Scale HDFS Transparency. Refer the "System tuning" on page 260 and "HDFS Transparency Tuning" on page 262 sections.

At Spark level, the following two should be tuned to make Spark work well on IBM Storage Scale:

Configuration	Default value	Recommended value

`spark.shuffle.file.buffer`
`($SPARK_HOME/conf/spark-defaults.conf)`
	32K	IBM Storage Scale data blocksize
		`spark_shuffle_file_buffer``=$(/usr/lpp/mmfs/bin/mmlsfs <filesystem_name> -B | tail -1 | awk ' { print $2} ')`
		If the blocksize of file system is larger than 2MB, configure 2MB for **`spark.shuffle.file.buffer`**.

`spark.local.dir` **Note:** This configuration will be overridden by either of the following environment variables set by the cluster manager: • SPARK_LOCAL_DIRS (Standalone) • MESOS_SANDBOX (Mesos) • LOCAL_DIRS (YARN)	`/tmp`	Configure the local directory for this (not configure this with IBM Storage Scale directory).
`spark.hadoop.mapreduce.fileoutputcommitter.algorithm.version`	1	Changing this into *2* can make Spark job commit fast.

As for other tuning for Spark, refer Spark configuration and Tuning Spark for Spark level tuning.

Benchmarking Spark

For benchmarking Spark, follow the guide from some popular benchmarks, such as spark-bench, BigDataBench-Spark.

Workloads/Benchmarks information

This topic describes the Workloads/Benchmarks information.

Teragen

Shipped with Hadoop release.

Terasort

Terasort is shipped with Hadoop package. It is located at `$BI_HOME/IHC/hadoop-example.jar`.

TPC-H

Usually, TPC-H for big data is done with Hive. Also, some users will do TPC-H for big data with Pig. The former one is more widely used.

TPC-H data generation (DBGEN) should be downloaded from TPC-H website:

TPC-H over hive:

Running TPC-H queries on Hive

TPC-H-Hive

Imperative and Declarative Hadoop: TPC-H in Pig and Hive

Hive is shipped with BigInsights. For more information, see APACHE HIVE TM.

YCSB

YCSB (Yahoo! Cloud System Benchmark) is widely used to benchmark some no SQL db, such as hbase. You can download it from YCSB.

Hibench

Hibench is a workload-mixed benchmark. It consists of nine different workloads (such as TeraGen, Terasort, DFSIO, Hive etc). It is used to evaluate the cluster performance under many different workloads.

HiBench

Hive

Shipped with BigInsights (v2.1). You can download the benchmark for Hive from Running TPC-H queries on Hive.

For more information, see APACHE HIVE TM.

Chapter 4. Cloudera Data Platform (CDP) Private Cloud Base

Overview

CDP Private Cloud Base is the on-premises version of Cloudera Data Platform. This new product combines the best of Cloudera Enterprise Data Hub and Hortonworks Data Platform Enterprise along with new features and enhancements across the stack. This unified distribution is a scalable and customizable platform where you can securely run many types of workloads.

CDP Private Cloud Base supports a variety of hybrid solutions where compute tasks are separated from data storage and where data can be accessed from remote clusters, including workloads created using CDP Private Cloud Experiences. This hybrid approach provides a foundation for containerized applications by managing storage, table schema, authentication, authorization, and governance. A high-level architecture of CDP Private Cloud base with IBM Storage Scale is shown in the following figure:

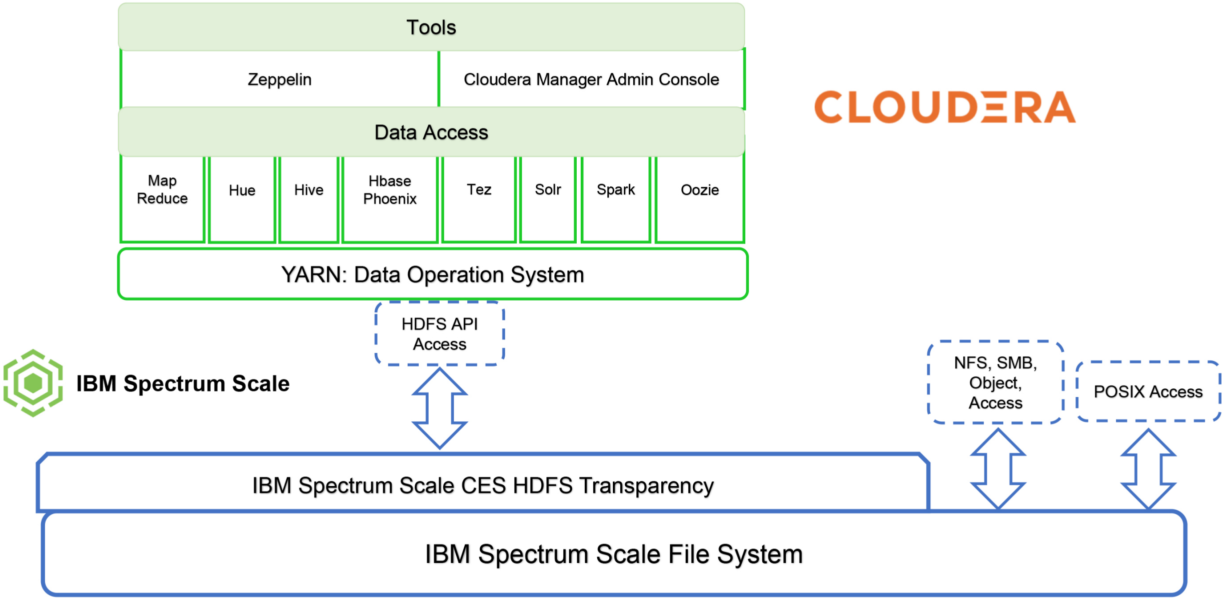


Figure 25. High-level architecture of Cloudera Data Platform Private Cloud Base with IBM Storage Scale

CDP Private Cloud Base is comprised of a variety of components such as Apache Spark, Apache Hive 3 and Apache HBase along with many other components for specialized workloads. You can select any combination of these services to create clusters that address your business requirements and workloads. Several pre-configured packages of services are also available for common workloads.

This Cloudera Data Platform (CDP) Private Cloud Base section describes the deployment of Cloudera Data Platform Private Cloud Base with IBM Storage Scale CES HDFS Transparency (CES HDFS) by using the Cloudera Manager (CM) Custom Service Descriptor (CSD) framework.

This document describes the deployment of Cloudera Data Platform Private Cloud Base with IBM Storage Scale CES HDFS Transparency (CES HDFS) by using the Cloudera Manager (CM) Custom Service Descriptor (CSD) framework. Follow this guide when you are using IBM Storage Scale as the storage for CDP Private Cloud Base because it contains deviation procedures for CDP Private Cloud Base.

Architecture

This topic describes the architecture of Cloudera Data Platform (CDP) Private Cloud Base with IBM Storage Scale.

As shown in Figure 26 on page 288 and Figure 27 on page 289, CDP Private Cloud Base can be deployed with IBM Storage Scale using Remote mount or single IBM Storage Scale cluster.

The benefits of separation of the Hadoop cluster hosts (master hosts, utility hosts, gateway hosts, or worker hosts) from the storage hosts (HDFS Transparency NameNodes and DataNodes) are as follows:

- The Hadoop layer and the storage layer can be managed separately and by different teams.
- As IBM Storage Scale is not installed on the Hadoop cluster hosts, you do not need specific Kernel levels on the Hadoop cluster hosts.
- Only the IBM Storage Scale hosts must have the same value for uid/gid.
- Only IBM Storage Scale requires password-less ssh for either a root or a non-root user with sudo privileges on all the nodes

Recommended configuration

- Storage hosts:
  - NameNode HA (2 NameNodes)
  - DataNode resiliency (3 DataNodes)

  Note: The performance also depends on the network and the number of DataNodes that can drive the bandwidth of the storage (ESS) and the number of Hadoop worker hosts.
- Hadoop cluster hosts:
  - For role assignments information about Hadoop cluster hosts, see Cloudera Runtime Cluster Hosts and Role Assignments.
- The NameNode cannot be colocated with the DataNode or with any other Hadoop services.

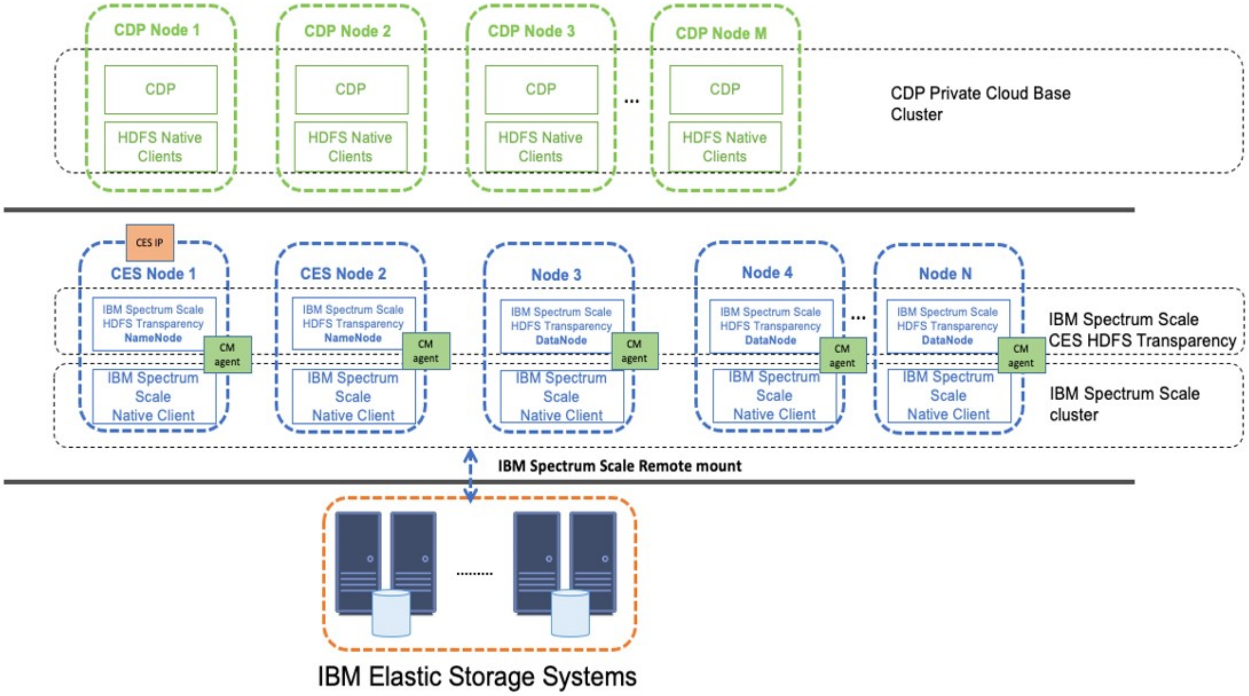

CDP Node 1
CDP Node 2
CDP Node 3
CDP Node M
CDP
HDFS Native Clients
...
CDP Private Cloud Base Cluster
CES IP
CES Node 1
CES Node 2
Node 3
Node 4
Node N
IBM Spectrum Scale HDFS Transparency NameNode
IBM Spectrum Scale HDFS Transparency DataNode
CM agent
IBM Spectrum Scale Native Client
IBM Spectrum Scale CES HDFS Transparency
IBM Spectrum Scale cluster
IBM Spectrum Scale Remote mount
IBM Elastic Storage Systems

Figure 26. Deploying Cloudera Data Platform (CDP) Private Cloud Base with IBM Storage Scale using Remote mount

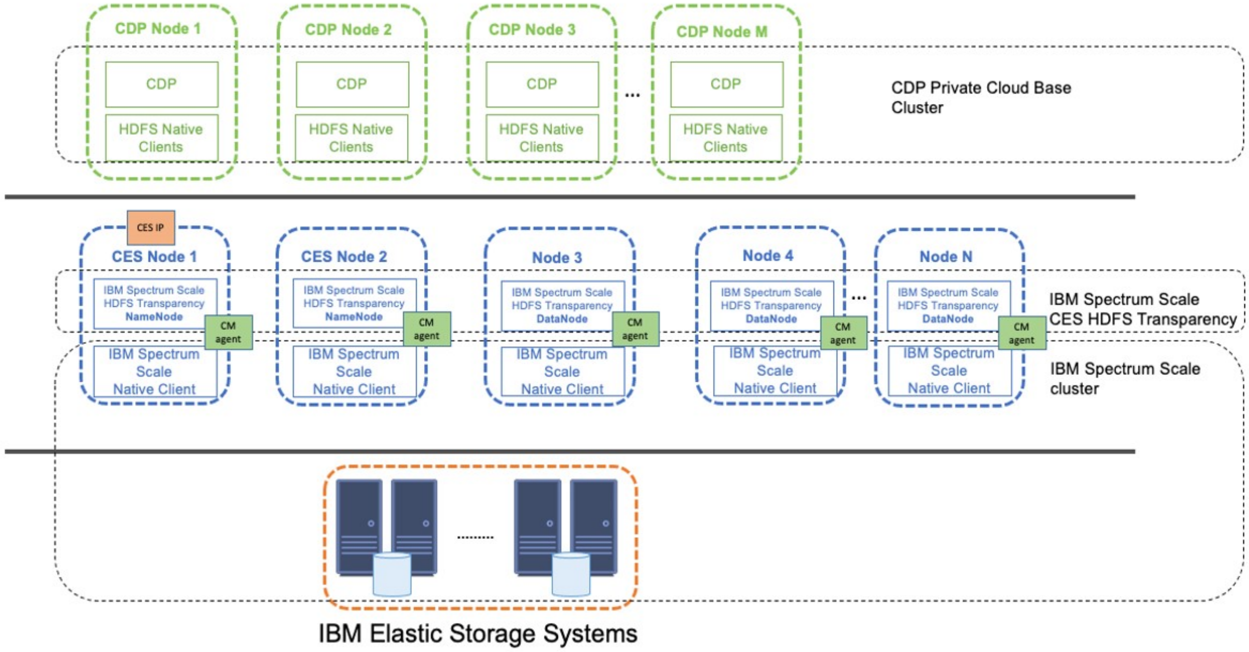

CDP Node 1
CDP Node 2
CDP Node 3
CDP Node M
CDP
HDFS Native Clients
...
CDP Private Cloud Base Cluster
CES IP
CES Node 1
CES Node 2
Node 3
Node 4
Node N
IBM Spectrum Scale HDFS Transparency NameNode
IBM Spectrum Scale HDFS Transparency DataNode
CM agent
IBM Spectrum Scale Native Client
IBM Spectrum Scale CES HDFS Transparency
IBM Spectrum Scale cluster
....
IBM Elastic Storage Systems

Figure 27. Deploying Cloudera Data Platform (CDP) Private Cloud Base with IBM Storage Scale using single IBM Storage Scale cluster

Cloudera Data Platform (CDP) consists of CDP Private Cloud Base cluster, IBM Storage Scale CES HDFS Transparency cluster and the shared storage layer.

CDP Private Cloud Base cluster

The CDP Private Cloud Base cluster consists of CDP nodes. One of these nodes hosts the Cloudera Manager where the IBM Storage Scale CSD will be placed into the CM directory for CSD jar files.

For CDP Private Cloud Base node roles recommendations, see Runtime Cluster Hosts and Role Assignments under the CDP Private Cloud Cloudera documentation.

IBM Storage Scale CES HDFS transparency cluster

The IBM Storage Scale CES HDFS Transparency cluster consists of NameNodes (CES protocol node and IBM Storage Scale client) and DataNodes (IBM Storage Scale client). The minimum requirement is to have two IBM Storage Scale HDFS Transparency NameNodes (HA) and three or more IBM Storage Scale HDFS Transparency DataNodes. The NameNodes are a part of the CES protocol nodes while the DataNodes are not a part of the CES protocol nodes. The CES HDFS Transparency nodes also consist of the IBM Storage Scale native clients. The Cloudera Manager Agent (CM agent) is also present in the IBM Storage Scale CES HDFS transparency cluster. The function of the CM agent is to facilitate the management of HDFS transparency NameNodes and HDFS transparency DataNodes from the Cloudera Manager in the CDP Private Cloud Base cluster.

The following figure shows the Cloudera and IBM Storage Scale/HDFS Transparency components on the CES HDFS nodes:

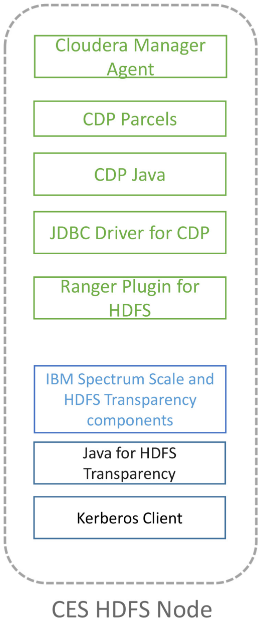


Figure 28. Cloudera and IBM Storage Scale/HDFS Transparency components on CES HDFS nodes

Cloudera and IBM Storage Scale/HDFS Transparency components on the CES HDFS nodes are described in the following list:

1. Cloudera Manager agent: The Cloudera Manager agent is a python-based agent. It consists of `cloudera-manager-agent` and `cloudera-manager-daemons` as its components. The Cloudera Manager agent can be installed through Cloudera Manager or you can also install it manually. If you are installing the Cloudera Manager agent directly on the hosts through Cloudera Manager, you need to provide the password or the ssh-private key of the managed host. You do not need the password or the ssh-private key of the managed host if you are installing manually.
2. CDP Private Cloud Base parcels: CDP parcels contains the installable for the CDP Private Cloud Base services. Hosts download the parcel using HTTP (wget) from Cloudera Manager.
3. CDP Private Cloud Base Java: Cloudera Manager requires to have the same version of Java on all the managed nodes. If the CM agent is installed using CM, CDP Private Cloud Base version of Java will also be installed using CM. For information on the Java level support, see "Hardware and software requirements" on page 292.
4. Ranger plug in for HDFS: Ranger plug in for HDFS is needed for the NameNode to cache the Ranger policies.
5. Java for HDFS Transparency: A version of Java must already be installed on HDFS Transparency prior to the node being managed by Cloudera Manager.
6. Kerberos client: For supported Kerberos distributions, see "Kerberos" on page 81.

IBM Storage Scale cluster

The IBM Storage Scale cluster as shown at the bottom of the Figure 26 on page 288 can either be IBM Elastic Storage system or any other shared storage system.

CES HDFS Transparency is remote mounted to ESS as shown in Figure 26 on page 288.

For information on Dual network deployment, see "Dual-network deployment" on page 311.

Note: If you plan to use object protocol, select the single IBM Storage Scale cluster architecture as shown in Figure 27 on page 289. For more information, see the *Limitations of protocols on remotely mounted file systems* topic in the *IBM Storage Scale: Administration Guide*.

Alternative architectures

This topic describes the alternative architectures that you can use when you do not require high availability (HA) for HDFS Transparency NameNode and when you want Hadoop services to be installed on the HDFS Transparency DataNode while knowing the limitations for these use cases.

- Non-HA HDFS Transparency NameNode architecture: CDP Private Cloud Base with IBM Storage Scale supports both NameNode HA and non-HA modes. You must use the non-HA NameNode option only for dev, test and non-production use cases.

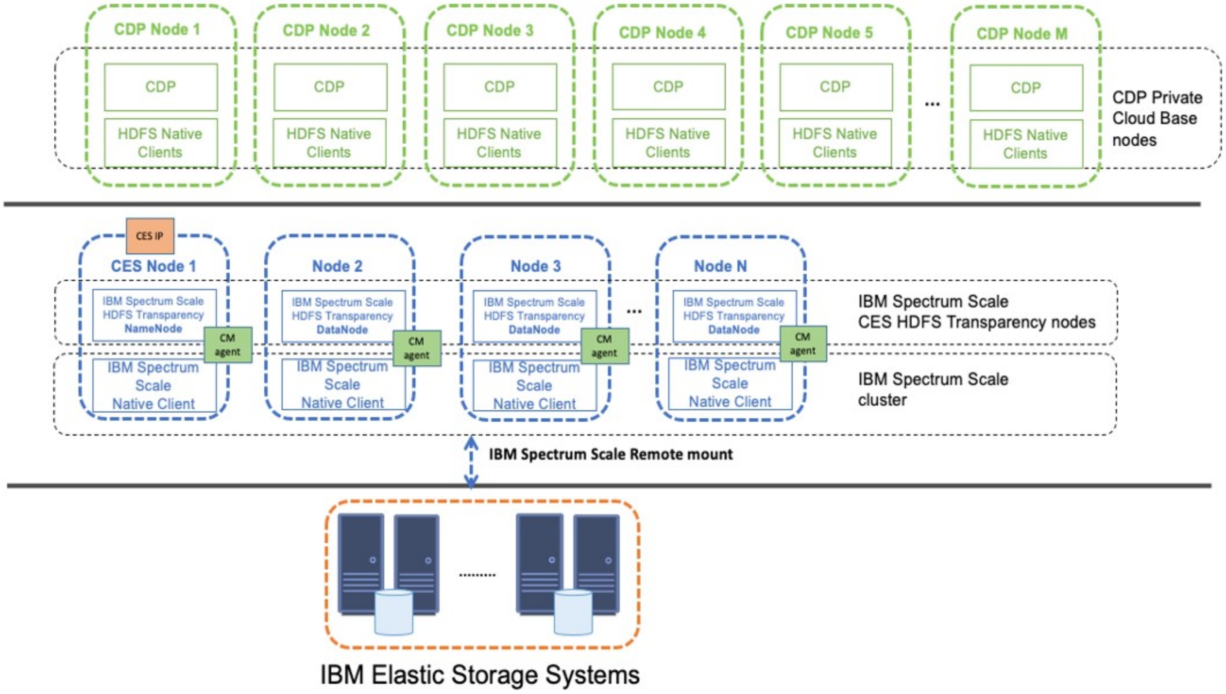

CDP Node 1
CDP Node 2
CDP Node 3
CDP Node 4
CDP Node 5
CDP Node M
CDP
HDFS Native Clients
...
CDP Private Cloud Base nodes
CES IP
CES Node 1
Node 2
Node 3
Node N
IBM Spectrum Scale HDFS Transparency NameNode
IBM Spectrum Scale HDFS Transparency DataNode
CM agent
IBM Spectrum Scale Native Client
IBM Spectrum Scale CES HDFS Transparency nodes
IBM Spectrum Scale cluster
IBM Spectrum Scale Remote mount
....
IBM Elastic Storage Systems

Figure 29. Non-HA HDFS Transparency NameNode architecture

- HDFS Transparency DataNode colocation architecture: A DataNode can have other Hadoop services colocated within the same node. Cloudera recommends that the DataNode (Worker) have specific services installed onto it. For a list of services that must be installed, see the *Worker Hosts* column in Cloudera Runtime Cluster Hosts and Role Assignments documentation.

  Note: The NameNode cannot be colocated with the DataNode.

  The HDFS Transparency DataNode colocation architecture has the following limitations:

  - Because IBM Storage Scale is installed on the Hadoop cluster hosts, it is not possible to manage the Hadoop cluster hosts and the storage hosts separately.
  - Requires specific Kernel levels on the Hadoop cluster hosts.
  - The IBM Storage Scale hosts must have the same value for all the uid/gid.
  - IBM Storage Scale requires password-less ssh for either a root or a non-root user with sudo privileges on all nodes.

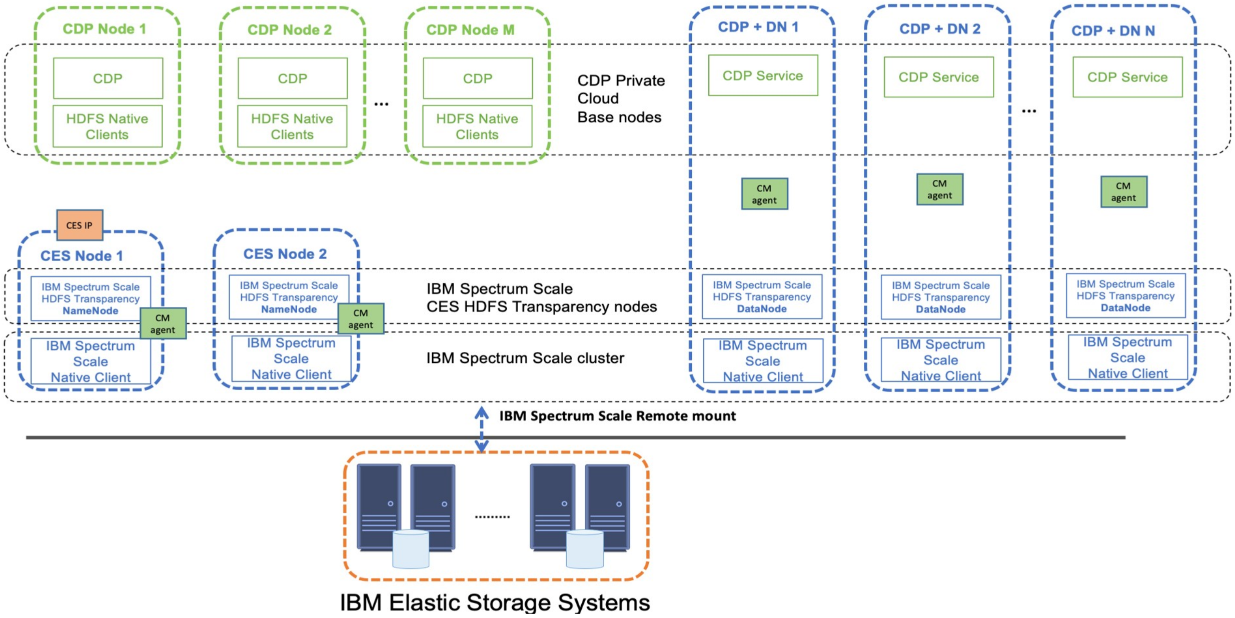

CDP Node 1
CDP
HDFS Native Clients
CDP Node 2
CDP
HDFS Native Clients
...
CDP Node M
CDP
HDFS Native Clients
CDP Private Cloud Base nodes
CDP + DN 1
CDP Service
CDP + DN 2
CDP Service
...
CDP + DN N
CDP Service
CM agent
CM agent
CM agent
CES IP
CES Node 1
IBM Spectrum Scale HDFS Transparency NameNode
CM agent
IBM Spectrum Scale Native Client
CES Node 2
IBM Spectrum Scale HDFS Transparency NameNode
CM agent
IBM Spectrum Scale Native Client
IBM Spectrum Scale CES HDFS Transparency nodes
IBM Spectrum Scale cluster
IBM Spectrum Scale HDFS Transparency DataNode
IBM Spectrum Scale Native Client
IBM Spectrum Scale HDFS Transparency DataNode
IBM Spectrum Scale Native Client
IBM Spectrum Scale HDFS Transparency DataNode
IBM Spectrum Scale Native Client
IBM Spectrum Scale Remote mount
....
IBM Elastic Storage Systems

Figure 30. HA and DataNode colocation architecture

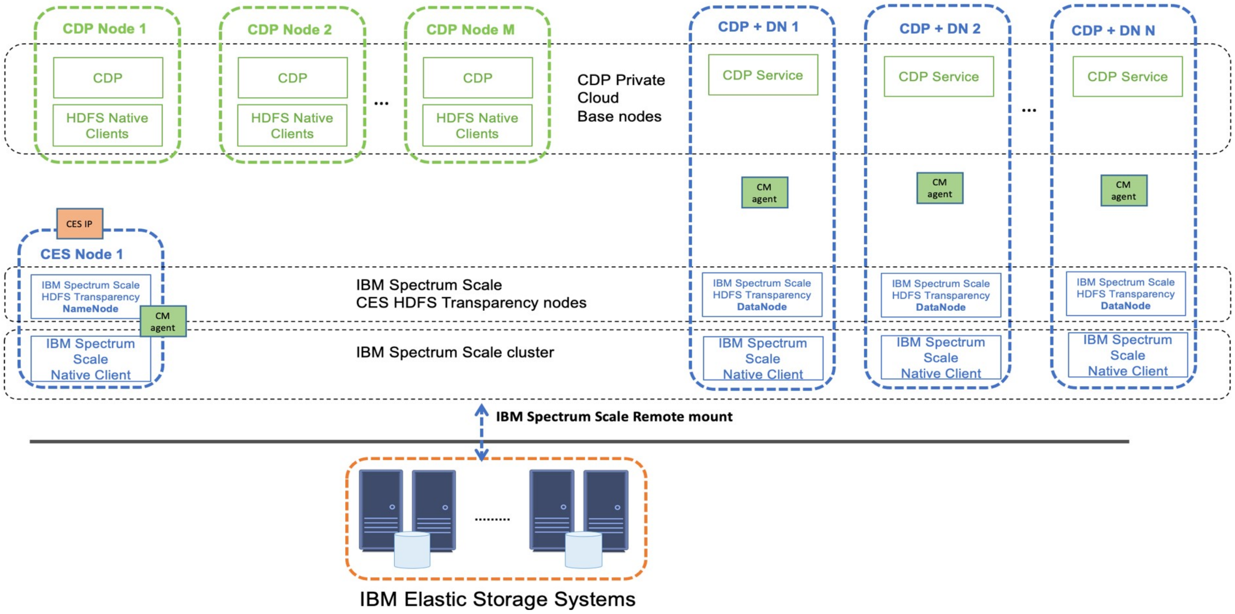

CDP Node 1
CDP
HDFS Native Clients
CDP Node 2
CDP
HDFS Native Clients
...
CDP Node M
CDP
HDFS Native Clients
CDP Private Cloud Base nodes
CDP + DN 1
CDP Service
CDP + DN 2
CDP Service
...
CDP + DN N
CDP Service
CM agent
CM agent
CM agent
CES IP
CES Node 1
IBM Spectrum Scale HDFS Transparency NameNode
CM agent
IBM Spectrum Scale Native Client
IBM Spectrum Scale CES HDFS Transparency nodes
IBM Spectrum Scale cluster
IBM Spectrum Scale HDFS Transparency DataNode
IBM Spectrum Scale Native Client
IBM Spectrum Scale HDFS Transparency DataNode
IBM Spectrum Scale Native Client
IBM Spectrum Scale HDFS Transparency DataNode
IBM Spectrum Scale Native Client
IBM Spectrum Scale Remote mount
....
IBM Elastic Storage Systems

Figure 31. Non-HA and DataNode colocation architecture

Planning

Review the "Hadoop IBM Storage Scale Architecture" on page 4 on which the configuration setup is to be used in your environment.

Hardware and software requirements

This section specifies the minimum hardware and software requirements for Cloudera Data Platform (CDP) Private Cloud Base, IBM Storage Scale CES HDFS transparency and IBM Storage Scale.

Hardware requirements

Following are the Hardware requirements for each product and component:

Cloudera Data Platform Private Cloud Base

For Cloudera Data Platform Private Cloud Base hardware requirements information for a specific release, see Hardware Requirements.

CES HDFS

For production, it is recommended to have two NameNodes and three or more DataNodes.

For production, it is recommended that each NameNode x86 server should have a minimum of two sockets with at least eight cores each with 128 GB memory.

For production, it is recommended that each DataNode x86 server should have a minimum of two sockets with at least eight cores each with 64 GB memory.

For better performance, reserve 20% of the system physical memory, subjected to a maximum of 20 GB per node, for IBM Storage Scale pagepool.

IBM Storage Scale

For IBM Storage Scale hardware requirements information, see the *Hardware requirements* topic in *IBM Storage Scale: Concepts, Planning, and Installation Guide*.

Shared storage

Shared storage can either be ESS, ECE or SAN based storage. Hadoop services including HDFS Transparency cannot be installed onto the storage nodes (SAN, ECE, ESS).

If you have ESS as a shared storage, refer the documentation for your model under IBM Elastic Storage System documentation. ESS is set up and tuned by IBM Lab Services.

If you have ECE as a shared storage, refer *IBM Storage Scale Erasure Code Edition Hardware requirements* in the IBM Storage Scale Erasure Code Edition Guide.

Note:

- Ensure that the CDP Private Cloud Base cluster and the CES HDFS cluster are on the same OS and architecture platform. Cloudera requires nodes to install its component software on the same OS and architecture platform. CES protocol does not support mixed architecture levels. For example, if CES HDFS protocol is on x86_64 platform, then all the CES protocol nodes and CDP Private Cloud Base must be on x86_64 platform. The shared storage can be on a different architecture platform. For example, if CES protocol nodes is on x86_64, ESS can be on power.
- FPO is not supported for CDP Private Cloud Base with IBM Storage Scale.

Software requirements

Following are the Software requirements for each product and component:

Cloudera Data Platform Private Cloud Base

OpenJDK 8

PostgreSQL 10 server

Python 2

CES HDFS

OpenJDK 8

Python 2 and Python 3.6 or later

IBM Storage Scale

For IBM Storage Scale software requirements information, see the *Software requirements* topic in *IBM Storage Scale: Concepts, Planning, and Installation Guide*.

Note: Ensure that the CDP Private Cloud Base cluster and the CES HDFS cluster are at the same Operating System level. The shared storage can be at a different Operating System level. For example, if CES protocol nodes are on RH 8.2, ESS can be on RH 7.7.

For more information on requirements for Hardware, Operating system, Database, and Java, see CDP Private Cloud Base Requirements and Supported Versions.

For information on limitations for CDP Private Cloud Base with IBM Storage Scale, see “Limitations” on page 343.

Cluster host and role assignments

Cloudera Data Platform Private Cloud Base

Refer CDP Private Cloud Base for the specific CDH release in the Runtime Cluster Hosts and Role Assignments section.

CES HDFS

NameNodes - 2 (HA)

DataNodes - 3 or more

Support Matrix

This section lists the support matrix for CDP Private Cloud Base.

Table 30. Support matrix

CDP Private Cloud Base	**Cloudera Manager (CM)**	**Cloudera Runtime (CDH)**	**IBM Storage Scale[1]**	**RHEL** **x86_64**	**RHEL** **ppc64le**
7.1.9	7.11.3-CHF1	7.1.9-CHF1	5.1.8.0+	7.9, ~~8.4~~, 8.6, 9.1,	7.9, ~~8.4~~, 8.6, 9.1,
7.1.8	7.7.1[2]	7.1.8	5.1.4.0+	7.9, ~~8.4~~, 8.6	7.9, ~~8.4~~, 8.6
7.1.7 SP 2	7.6.7	7.1.7.2	5.1.4.0 +	7.9, ~~8.4~~, 8.6	7.9, ~~8.4~~, 8.6
7.1.7 SP 1	7.6.1	7.1.7.1	5.1.2.2 +	7.9, ~~8.2~~, ~~8.4~~	7.9, ~~8.2~~, ~~8.4~~
7.1.7	7.4.4	7.1.7	5.1.1.2 +	7.9, ~~8.2~~	7.9, ~~8.2~~
7.1.6	7.3.1	7.1.6	5.1.1.0, 5.1.1.1	7.7, 7.9	7.7, 7.9
NA	7.2.3	7.1.4	5.1.0.1-5.1.0.3	7.7	7.7

[1] To ensure the version supports your environment and use case, see the Third-party filesystems documentation by Cloudera for IBM Storage Scale support information, before selecting a Cloudera version.

[2] Requires Python v3.8 for Hue.

Table 31. HDFS Transparency and CSD version for specific IBM Storage Scale version

IBM Storage Scale version	**HDFS Transparency version**	**CSD version**
5.1.9.0	3.1.1-15	1.2.1-0
5.1.8.1	3.1.1-14	1.2.0-0
5.1.7.1 - 5.1.8.0	3.1.1-13	1.2.0-0
5.1.7	3.1.1-12	1.2.0-0
5.1.6.1	3.1.1-12	1.2.0-0
5.1.6	3.1.1-11	1.2.0-0
5.1.5 - 5.1.5.1	3.1.1-10	1.2.0-0
5.1.4.1	3.1.1-10	1.2.0-0

Table 31. HDFS Transparency and CSD version for specific IBM Storage Scale version (continued)

IBM Storage Scale version	HDFS Transparency version	CSD version
5.1.4	3.1.1-9	1.2.0-0
5.1.3.0 - 5.1.3.2	3.1.1-8	1.2.0-0
5.1.2.9	3.1.1-12	1.2.0-0
5.1.2.6 - 5.1.2.8	3.1.1-10	1.2.0-0
5.1.2.2 - 5.1.2.5	3.1.1-8	1.2.0-0
5.1.2.1	3.1.1-7	1.2.0-0
5.1.2	3.1.1-6	1.2.0-0
5.1.1.2 - 5.1.1.4	3.1.1-5	1.2.0-0
5.1.1.1	3.1.1-5	1.1.0-0
5.1.1.0	3.1.1-4	1.1.0-0
5.1.0.1 - 5.1.0.3	3.1.1-3	1.0.0-0

From CDP Private Cloud Base 7.1.6, you can upgrade CDP Private Cloud Base to a newer CDP Private Cloud Base version on IBM Storage Scale.

Table 32. Upgrade support

Existing stack version			Stack version for upgrade		
CDP Private Cloud Base	**IBM Storage Scale CSD**	**IBM Storage Scale**	**CDP Private Cloud Base**	**IBM Storage Scale CSD**	**IBM Storage Scale**
7.1.6	1.1.0.0	5.1.1.0 5.1.1.1	7.1.7	1.2.0.0	5.1.1.2+

Note:

- For generic support information, see "Hadoop distribution support" on page 24.
- Click on the Cloudera Manager and Cloudera Runtime version to get to the Cloudera download information.
- The Cloudera download is behind a paywall and requires a username/password that you obtain with the license file. The download username/password is in the license file and can be extracted from CDP Private Cloud Base download if you are not able to locate the original license email.
- Unlike previous versions of HDFS Transparency, HDFS Transparency 3.1.1 is tightly coupled with IBM Storage Scale. You need to upgrade the IBM Storage Scale package to get the correct supported versions to IBM Storage Scale Big Data Analytics Integration Toolkit for HDFS Transparency (Toolkit for HDFS), IBM Storage Scale HDFS Transparency and IBM Storage Scale Cloudera Custom Service Descriptor (CDP CSD).
- For the location of the IBM Storage Scale packages, see "Downloads" on page 300.
- Support for CDP Private Cloud Base started with CDP Private Cloud Base CM 7.2.3 and CDH 7.1.4 certification with IBM Storage Scale 5.1.0.1.
- From CDP Private Cloud Base 7.1.6, TLS and HDFS encryption are supported with IBM Storage Scale.
- For CES HDFS limitations, see CES HDFS Limitations and Recommendations.
- The following components are not supported with IBM Storage Scale:

- Apache Kudu provides an emulated storage layer that is typically configured over Linux native storage, for example, ext4 filesystem. Instead of running Kudu, which is a storage layer, over IBM Storage Scale, a storage provider, users might consider using IBM BigSQL, Hbase, or Impala.
  - Apache Ozone is a separate filesystem from Cloudera so it is not relevant in the context of IBM Storage Scale filesystem.
- Starting from CDP 7.1.6, Impala is supported on the x86 platform. Impala is not supported on IBM Power.
- For best practices for deployment on Power, see the Cloudera Data Platform (CDP) Private Cloud Base on IBM Power and IBM Elastic Storage System (ESS) white paper.
- Java OpenJDK 11 is supported from CDP Private Cloud Base 7.1.7 SP1 with HDFS Transparency 3.1.1-8.

Preparing the environment

This section describes how to prepare the environment to install Cloudera Data Platform (CDP) Private Cloud Base, IBM Storage Scale CES HDFS Transparency and the shared storage system.

HDFS Transparency package

This topic helps in the preparation to install HDFS Transparency package.

IBM Storage Scale HDFS Transparency (HDFS Protocol) offers a set of interfaces that allow applications to use HDFS Client to access IBM Storage Scale through HDFS RPC requests.

All data transmission and metadata operations in HDFS are done through the RPC mechanism, and processed by the NameNode and the DataNode services within HDFS.

IBM Storage Scale HDFS Transparency is part of the IBM Storage Scale self-extracting archive package. For information on the supported version for your CDP Private Cloud Base and IBM Storage Scale environment, see "Support Matrix" on page 294.

For more information, see "HDFS Transparency download" on page 28 section.

The module name is `gpfs.hdfs-protocol-3.1.1-(version)`.

The IBM Storage Scale installation toolkit can only install HDFS Transparency from the IBM Storage Scale software self-extracting archive extracted directory.

If the HDFS Transparency package is not from the IBM Storage Scale software self-extracting archive directory, you need to manually install the HDFS Transparency package.

Note:

- For manual install, ensure that there is only one package of HDFS Transparency in the IBM Storage Scale repository. Rebuild the repository by executing the **"createrepo . "** command to update the repository metadata.
- Properly review and set "OS tuning for all nodes in HDFS Transparency" on page 55 and "Configure NTP to synchronize the clock in HDFS Transparency" on page 56 for all nodes.

IBM Storage Scale file system

This topic helps in the preparation to install IBM Storage Scale file system.

For IBM Storage Scale overview, see the *Product overview* section in the *IBM Storage Scale: Concepts, Planning, and Installation Guide*.

If you have purchased the IBM Storage Scale license, you can download the IBM Storage Scale base installation package files from IBM Passport Advantage.

For IBM Storage Scale version 5.1.0, full image is available at Fix Central.

For IBM Storage Scale trial and purchase licenses, see https://www.ibm.com/us-en/marketplace/scale-out-file-and-object-storage/purchase.

For ordering IBM Storage Scale, see Question 1.1 in IBM Storage Scale FAQ documentation.

Kernel, SELinux and NTP

This topic gives information about Kernel, SELinux and NTP.

Kernel

See Installation of Kernel packages under the IBM Storage Scale support for Hadoop Kernel section.

SELinux

For information on SE Linux, see “SELinux” on page 17.

NTP

For information on NTP, see “NTP” on page 18.

Note: Ensure that the date and time of the CDP Private Cloud Base cluster, CES HDFS cluster and ESS are in synchronization.

Network validation

While using a private network for Hadoop or IBM Storage Scale nodes, ensure that all the nodes, including the management nodes, have hostnames bound to the faster internal network or the data network.

On all the nodes, the **`hostname -f`** command must return the FQDN of the faster internal network. This network can be a bonded network. If the nodes do not return the FQDN, modify `/etc/sysconfig/network` and use the hostname command to change the FQDN of the node.

The `/etc/hosts` file host order listing must have the long hostname first before the short hostname.

If the nodes in your cluster have two network adapters, see “Dual-network deployment” on page 311.

Setting password-less ssh access for root

IBM Storage Scale Master is a role designated to the host on which the Master component of the IBM Storage Scale service is installed. It should be a part of the administrator nodes set. All the IBM Storage Scale cluster wide administrative commands including those for creation of the IBM Storage Scale cluster and the file-system are run from this host.

Password-less ssh access for root must be configured from the IBM Storage Scale Master node to all the other IBM Storage Scale nodes. This is needed for IBM Storage Scale to work. For non-adminMode central clusters, ensure that you have bi-directional password-less setup for the fully qualified and short names for all the GPFS™ nodes in the cluster. This must be done for the root user. For non-root Ambari environment, ensure that the non-root ID can perform bi-directional password-less SSH between all the GPFS nodes.

Note: BDA Ambari integration supports the admin mode central configuration of IBM Storage Scale. See *adminMode configuration attribute* in *IBM Storage Scale: Administration Guide*.

In this configuration, one or more hosts could be designated as IBM Storage Scale Administration (or Admin) nodes. By default, the GPFS Master is an Admin node. In Admin mode central configuration, it is sufficient to have only uni-directional password-less ssh for root from the Admin nodes to the non-admin nodes. This configuration ensures better security by limiting the password-less ssh access for root.

An example on setting up password-less access for root from one host to another:

1. Define Node1 as the IBM Storage Scale master.
2. Log on to Node1 as the root user.

   ```
   # cd /root/.ssh
   ```

3. Generate a pair of public authentication keys. Do not type a passphrase.

   ```
   # ssh-keygen -t rsa
   ```

   Generate the public-private rsa key pair.

Type the name of the file in which you want to save the key (/root/.ssh/id_rsa):

Type the passphrase.

Type the passphrase again.

The identification has been saved in /root/.ssh/id_rsa.

The public key has been saved in /root/.ssh/id_rsa.pub.

The key fingerprint is:

...

Note: During **`ssh-keygen -t rsa`**, accept the default for all.

4. Set the public key to the `authorized_keys` file.

   ```
   # cd /root/.ssh/; cat id_rsa.pub > authorized_keys
   ```

5. For clusters with adminMode as *allToAll*, copy the generated public key file to nodeX.

   ```
   # scp /root/.ssh/* root@nodeX:/root/.ssh
   ```

   where, nodeX is all the nodes.

   For clusters with adminMode as *central*, copy the generated public key file to nodeX.

   ```
   # scp /root/.ssh/* root@nodeX:/root/.ssh
   ```

   nodeX is all the nodes chosen for administration.

   Configure the password less ssh with non admin nodes (*nodeY*) in the clusters.

   ```
   # ssh-copy-id root@nodeY
   ```

   nodeY is rest of the cluster nodes.

6. Ensure that the public key file permission is correct.

   ```
   #ssh root@nodeX “chmod 700 .ssh; chmod 640 .ssh/authorized_keys"
   ```

7. Check password-less access

   ```
   # ssh node2

   [root@node1 ~]# ssh node2
   The authenticity of host 'gpfstest9 (192.0.2.0)' can't be established.
   RSA key fingerprint is 03:bc:35:34:8c:7f:bc:ed:90:33:1f:32:21:48:06:db.
   Are you sure you want to continue connecting (yes/no)?yes
   ```

   Note: You also need to run **`ssh node1`** to add the key into `/root/.ssh/known_hosts` for password-less access.

IBM Storage Scale local repository

IBM Storage Scale supports installation using the IBM Storage Scale installation toolkit or configuring a local IBM Storage Scale repository.

Note: If you have already setup an IBM Storage Scale file system, you can skip this section.

1. Ensure there is a Mirror repository server created before proceeding.
2. Set up the Local OS repository if needed.
3. Set up the Local IBM Storage Scale repository. This section helps you to set up the IBM Storage Scale and HDFS Transparency local repository.

ACL support

Ensure that the IBM Storage Scale file system ACL setting is set to *ALL*.

For more information, see HDFS and IBM Storage Scale filesystem ACL support.

Installing

This section lists the steps to install CES HDFS cluster and CDP Private Cloud Base cluster.

It is recommended to install CDP Private Cloud Base cluster and CES HDFS cluster on separate set of nodes to make these clusters inline with the isolation of data and compute.

Overview

This topic lists the high-level workflow for deploying an IBM Storage Scale CES HDFS based environment with CDP Private Cloud Base.

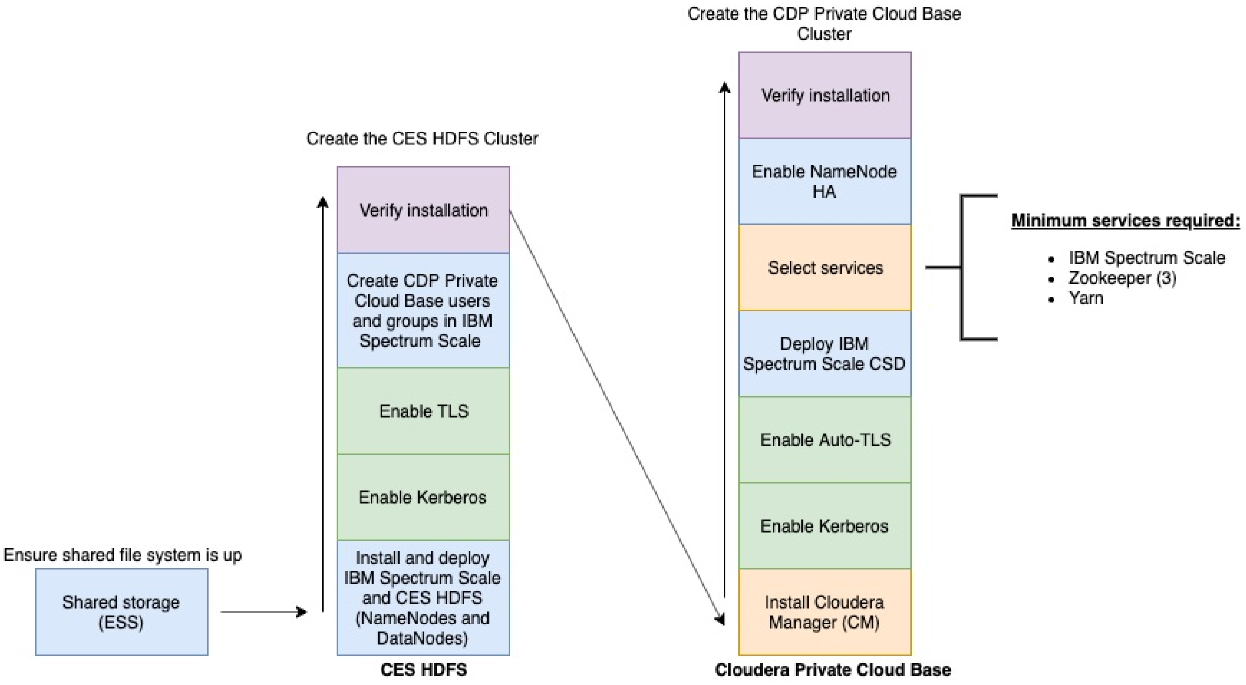

Ensure shared file system is up
Shared storage (ESS)
Create the CES HDFS Cluster
Verify installation
Create CDP Private Cloud Base users and groups in IBM Spectrum Scale
Enable TLS
Enable Kerberos
Install and deploy IBM Spectrum Scale and CES HDFS (NameNodes and DataNodes)
CES HDFS
Create the CDP Private Cloud Base Cluster
Verify installation
Enable NameNode HA
Select services
Deploy IBM Spectrum Scale CSD
Enable Auto-TLS
Enable Kerberos
Install Cloudera Manager (CM)
Cloudera Private Cloud Base
Minimum services required:
IBM Spectrum Scale
Zookeeper (3)
Yarn

Figure 32. Overview of CDP Private Cloud Base installation

- Designate separate nodes for IBM Storage Scale CES HDFS and CDP Private Cloud Base clusters.
- Set up the CES HDFS cluster:
  - Install the CES HDFS HA cluster.
  - If you need Kerberos, enable Kerberos on the CES HDFS cluster.
  - Verify the installation.
- Set up the CDP Private Cloud Base cluster:
  - Install the Cloudera Manager.
  - Optional: Enable Kerberos on the Cloudera Manager.
  - Optional: Enable Auto-TLS on the Cloudera Manager (TLS is supported from CDP Private Cloud Base 7.1.6).
  - Deploy IBM Storage Scale CSD and restart the Cloudera Manager.
  - Create a new CDP Private Cloud Base cluster. Select the IBM Storage Scale service along with Yarn, Zookeeper, and other services as needed.
  - Enable NameNode HA in the IBM Storage Scale service and restart all the services.

- Verify the installation.

Important: Adding the IBM Storage Scale service to an HDFS based CDP Private Cloud Base cluster is not supported. You must choose IBM Storage Scale service when you are creating a new CDP Private Cloud Base cluster.

Downloads

This section helps with download information for CDP Private Cloud Base and IBM Storage Scale.

Cloudera downloads

For Cloudera download information, see the "Support Matrix" on page 294.

IBM Storage Scale downloads

For IBM Storage Scale, BDA toolkit for HDFS, HDFS Transparency and CDP CSD download information, download the IBM Storage Scale self-extracting installation package. For more information, see "IBM Storage Scale file system" on page 296 and HDFS Transparency download.

This self-extracting installation package contains IBM Storage Scale, BDA Toolkit for HDFS, HDFS Transparency and CDP CSD package.

For example, for IBM Storage Scale 5.1.0.1 on RHEL7, the self-extracting installation package will place the packages into the following default directory:

```
/usr/lpp/mmfs/5.1.0.1/hdfs_rpms/rhel7/hdfs_3.1.1.x
```

Packages information

- IBM Storage Scale Big Data Analytics Integration Toolkit for HDFS Transparency (Toolkit for HDFS)
- IBM Storage Scale HDFS Transparency
- IBM Storage Scale Cloudera Custom Service Descriptor (CDP CSD)

Note:

- CDP Private Cloud Base with IBM Storage Scale is first certified with HDFS Transparency 3.1.1-3, BDA toolkit 1.0.2.1 and IBM Storage Scale CDP CSD 1.0.0-0 in the IBM Storage Scale 5.1.0.1 self-extracting installation package.
- The installation toolkit cannot be used if the HDFS Transparency package is downloaded as a patch (efix) package. Therefore, you must use the manual installation method to install the HDFS Transparency package.

Shared storage setup

Ensure that you have set up the shared storage that you are using.

If you have ESS as a shared storage, refer to the IBM Elastic Storage System documentation for your model. The ESS is setup and tuned by IBM Lab Services.

If you have ECE as a shared storage, see *IBM Storage Scale Erasure Code Edition Hardware requirements* in the IBM Storage Scale Erasure Code Edition Guide.

Ensure that the ACL is properly set up for the storage to be used for POSIX, HDFS and CES protocols. For information on the IBM Storage Scale ACL, see "ACL support" on page 299.

CES HDFS

The HDFS Transparency for CDP Private Cloud Base uses CES HDFS protocol cluster setup.

Setup the HDFS Transparency cluster

Setting up the HDFS Transparency for CDP Private Cloud Base using CES HDFS protocol cluster setup.

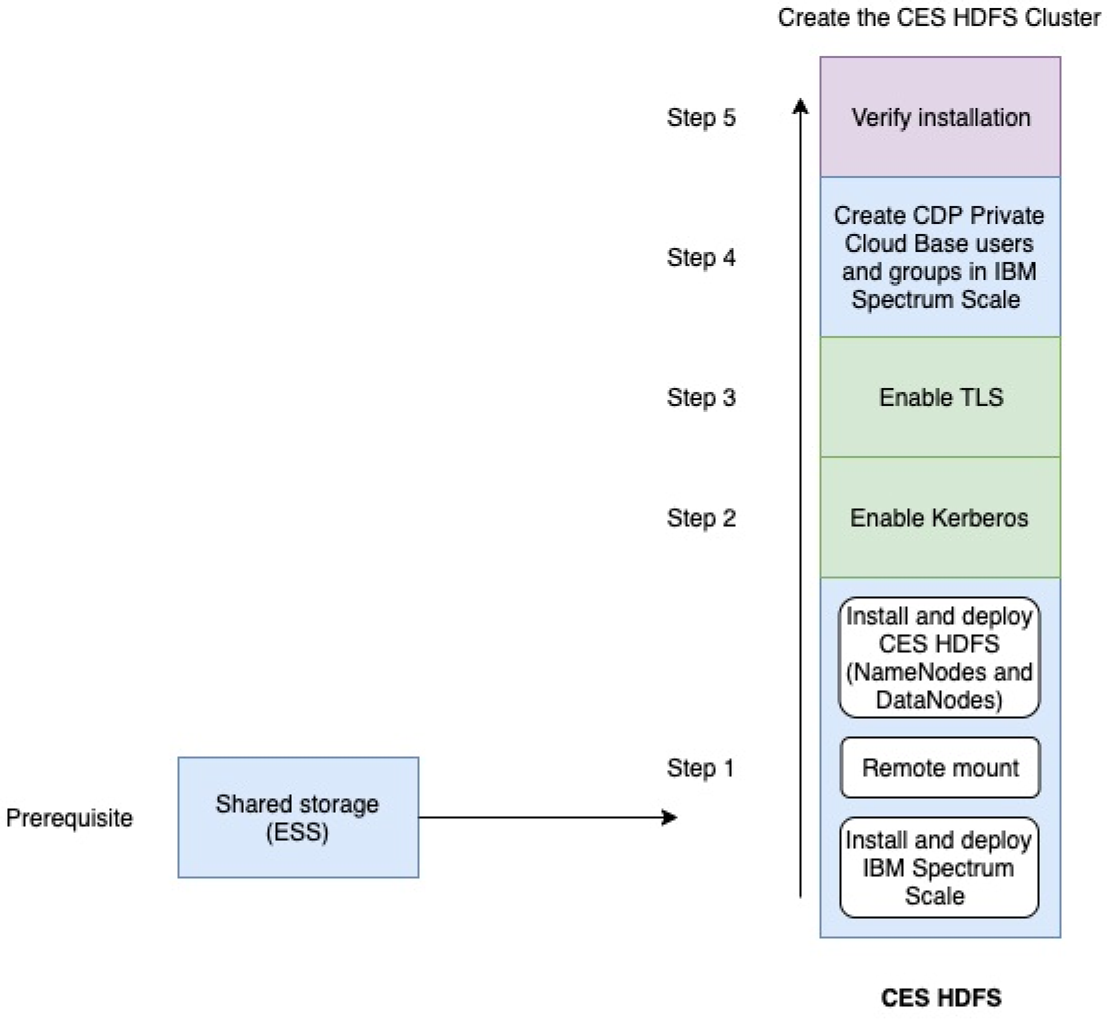

Create the CES HDFS Cluster
Step 5
Verify installation
Step 4
Create CDP Private Cloud Base users and groups in IBM Spectrum Scale
Step 3
Enable TLS
Step 2
Enable Kerberos
Install and deploy CES HDFS (NameNodes and DataNodes)
Step 1
Remote mount
Install and deploy IBM Spectrum Scale
Prerequisite
Shared storage (ESS)
CES HDFS

Figure 33. CES HDFS cluster installation overview

Prerequisite:

- Ensure that the shared storage is set up and available.

Steps:

1. Configure IBM Storage Scale on the CES HDFS cluster to access the shared storage. For example, the IBM Storage Scale cluster accesses ESS through the remote mount. After the storage is accessible, create the CES HDFS NameNodes and DataNodes onto the CES HDFS cluster. For information about setting up and installing CES HDFS, see "Installing" on page 29.
2. Optional: Enable Kerberos. To enable Kerberos, see "Enabling Kerberos for CES HDFS and CDP Private Cloud Base" on page 315.
3. Optional: Enable TLS. To enable TLS, see Enable TLS.
4. Set up the users and groups in the CES HDFS cluster. All the users and groups id must be same in an IBM Storage Scale cluster to work properly. To set up the users and groups id onto the IBM Storage Scale CES HDFS cluster, see "Configuring users, groups and file system access for IBM Storage Scale" on page 302.

5. Verify the CES HDFS cluster to ensure you can read and write to the storage file system.

After the CES HDFS cluster is configured, you can set up the CDP Private Cloud cluster.

Note:

- When IBM Storage Scale is installed, the IBM Storage Scale CSD (`gpfs.hdfs.cloudera.cdp.csd-<version-number>.noarch.rpm`) package resides in `/usr/lpp/mmfs/<IBM Storage Scale version>/hdfs_rpms/rhel/hdfs_3.1.1.x`. You need to move the package to the CDP Private Cloud cluster.
- If on the CES HDFS cluster NameNode HA, Kerberos and/or TLS are enabled, set up the CDP Private Base cluster with the same configuration.

Configuring users, groups and file system access for IBM Storage Scale

This section shows how to create CDP Private Cloud Base users and groups on the HDFS Transparency nodes, and also how to configure the IBM Storage Scale file system access.

Overview

1. Check if you have a Windows AD or LDAP-based network
2. Create CDP Private Cloud Base users and groups
3. (Optional): Create CDP Private Cloud Base users and groups on a new node being added (only for Add Node)
4. Verify the users and groups
5. Create any custom user or group (Optional)
6. Configure Hadoop supergroup for HDFS Transparency
7. Set ownership for the IBM Storage Scale file system Hadoop root directory

When you register hosts to Cloudera Manager, Cloudera Manager creates Hadoop users and groups corresponding to the services on all the managed hosts. These users and groups must be manually created on the IBM Storage Scale HDFS Transparency hosts before registering these hosts to the Cloudera Manager. Because IBM Storage Scale is a POSIX file system, it is required that any common system user and group must have the same UID and GID across all the IBM Storage Scale nodes.

1. If you are using Windows AD or LDAP-based network, users and groups for Hadoop users on your HDFS Transparency nodes should have consistent UID and GID across all the HDFS Transparency nodes. In that case, skip the next step and go to step 3.
2. If you are using local users, run the following command on one of the HDFS Transparency nodes to create these users and groups. The command can dynamically fetch the list of the NameNode and DataNode hosts from the cluster configuration and add the Hadoop users and groups to those hosts.

   A password-less SSH channel should exist for root from that host to all the other HDFS Transparency nodes for the command to run.

   ```
   /usr/lpp/mmfs/hadoop/scripts/gpfs_create_hadoop_users_dirs.py --create-users-and-groups
   ```

   This command performs the following actions:

   - Creates the CDP Private Cloud Base users and groups as defined in the `user_grp_dir_metadata.json` file.
   - Ensures that all such users and groups have consistent UID and GID across the hosts.
   - Creates a system group called supergroup to be used as Hadoop supergroup. hdfs, mapred and yarn users are added as members of this supergroup.
   - The output of the command is logged to `/var/log/user_group_configuration.log` file.

Note: If you are using HDFS Transparency 3.1.1-5 or earlier, the command should be used with the **`--hadoop-hosts`** option as follows:

```
/usr/lpp/mmfs/hadoop/scripts/gpfs_create_hadoop_users_dirs.py --create-users-and-groups --
hadoop-hosts <comma separated list of HDFS Transparency NameNodes and DataNodes>
```

For example:

```
/usr/lpp/mmfs/hadoop/scripts/gpfs_create_hadoop_users_dirs.py --create-users-and-groups --
hadoop-hosts nn01.gpfs.net,nn02.gpfs.net, dn01.gpfs.net,dn02.gpfs.net
```

3. (Optional): If you want to add only one NameNode or DataNode to an existing HDFS Transparency cluster, you need to input all the existing hostnames for the HDFS Transparency cluster and the new hostnames to the `gpfs_create_hadoop_users_dirs.py` script. This script ensures that all the values of UID/GID for the users and groups in the IBM Storage Scale cluster are consistent.

   For example:

   Existing HDFS Transparency hosts:

   ```
   nn01.gpfs.net,nn02.gpfs.net, dn01.gpfs.net,dn02.gpfs.net
   ```

   New DataNode being added:

   ```
   dn03.gpfs.net
   ```

   Run the following command to create the Hadoop users/groups on `dn03.gpfs.net`:

   ```
   /usr/lpp/mmfs/hadoop/scripts/gpfs_create_hadoop_users_dirs.py --create-users-and-groups --
   hadoop-hosts nn01.gpfs.net,nn02.gpfs.net,dn01.gpfs.net,dn02.gpfs.net,dn03.gpfs.net
   ```

4. Verify the users and groups by running the following command:

   ```
   /usr/lpp/mmfs/hadoop/scripts/gpfs_create_hadoop_users_dirs.py --verify-users-and-groups
   ```

5. (Optional): You can also create any custom user/group on the IBM Storage Scale nodes using the **`--create-custom-hadoop-user-group user-name[:group1[,group2..]]`** command. The command ensures that such a user/group is created with consistent UID/GID across all the nodes.

   - In the following example we create a user called *testuser* across the HDFS Transparency nodes. The user is created as a part of the hadoop group:

     ```
     # /usr/lpp/mmfs/hadoop/scripts/gpfs_create_hadoop_users_dirs.py --create-custom-hadoop-
     user-group testuser:hadoop
     Checking current state of the system..
     Group: hadoop already present on host nn01.gpfs.net
     Group: hadoop already present on host dn01.gpfs.net
     Group: testuser(10030) added successfully on host nn01.gpfs.net
     Group: testuser(10030) added successfully on host dn01.gpfs.net
     User: testuser(10028) added successfully on host nn01.gpfs.net
     User: testuser(10028) added successfully on host dn01.gpfs.net
     ```

   - On every CDP Private Cloud Base nodes which is not an IBM Storage Scale node, run the following command:

     ```
     # /usr/bin/useradd testuser
     ```

6. Configure Hadoop supergroup for HDFS Transparency.

   Ensure that HDFS Transparency is stopped by running the following command:

   ```
   /usr/lpp/mmfs/hadoop/sbin/mmhdfs hdfs status
   ```

   If HDFS Transparency is still running, stop it by using the following command:

   ```
   /usr/lpp/mmfs/hadoop/sbin/mmhdfs hdfs stop
   ```

Set the **dfs.permissions.superusergroup** parameter to *supergroup* by running the following command:

```
/usr/lpp/mmfs/hadoop/sbin/mmhdfs config set hdfs-site.xml -k
dfs.permissions.superusergroup=supergroup
```

Upload the configuration by running the following command:

```
/usr/lpp/mmfs/hadoop/sbin/mmhdfs config upload
```

7. Set ownership for the IBM Storage Scale Hadoop root directory.

   It is recommended to set the ownership for the IBM Storage Scale Hadoop root directory as `hdfs:supergroup` with 755 (`rwxr-xr-x`) permissions. By default, it is set to `root:root`.

   ```
   # /usr/bin/chown hdfs:supergroup <IBM Storage Scale mount directory>/< IBM Storage Scale
   Hadoop data directory>
   ```

   For example:

   ```
   # /usr/bin/chown hdfs:supergroup /ibm/gpfs/datadir1
   ```

   You can retrieve the IBM Storage Scale mount directory (**gpfs.mnt.dir**) and the IBM Storage Scale Hadoop data directory (**gpfs.data.dir**) using the following commands on a CES HDFS cluster node:

   ```
   /usr/lpp/mmfs/hadoop/sbin/mmhdfs config get gpfs-site.xml -k gpfs.mnt.dir -k gpfs.data.dir
   ```

   where, **gpfs.data.dir**=*datadir1* and **gpfs.mnt.dir**=*/ibm/gpfs*.

Installing Cloudera Data Platform Private Cloud Base with IBM Storage Scale

This section describes the steps to create a new CDP Private Cloud Base cluster with the IBM Storage Scale file system specific configuration.

For Cloudera documentation and download information, see "Support Matrix" on page 294.

Note: Before implementation ensure that you first read the entire section because there are deviations from CDP Private Cloud Base installation documentation when IBM Storage Scale is integrated.

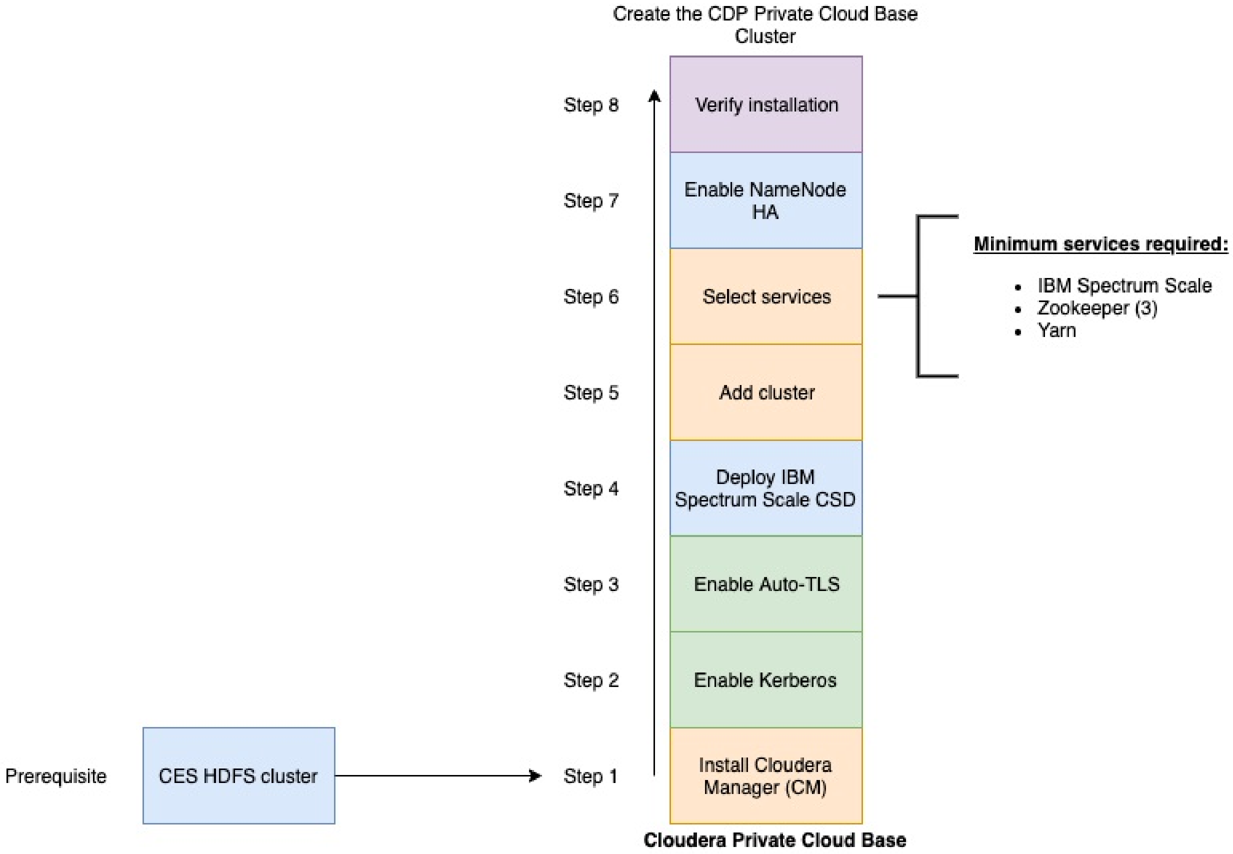


Figure 34. Overview of CDP Private Cloud Base cluster installation with IBM Storage Scale integration

Note: Ensure that CES HDFS cluster configuration is completed and up. For more information, see "CES HDFS" on page 301. If the NameNode HA, Kerberos and/or TLS are enabled on the CES HDFS cluster, the CDP Private Base cluster must be setup with the same configuration.

1. Install the Cloudera Manager (CM). For more information on the CDP Private Base version, see the CDP Private Cloud Base Installation Guide.

   Cloudera has the following two types of installation methods:

   - Trial Installation: The trial installation is to install the trial version of CDP Private Cloud Base in a non-production environment for demonstration and proof-of-concept use cases. This installation method is recommended for trial deployments but is not supported for production deployments because it is not designed to scale.
   - Production Installation: This topic describes the information for installing CDP Private Cloud Base using the Production Installation method.

   a. Stop the HDFS Transparency services.

      On the CES HDFS cluster, stop the HDFS Transparency NameNodes and DataNodes by running the following commands:

      ```
      # /usr/lpp/mmfs/hadoop/sbin/mmhdfs hdfs-dn stop
      # /usr/lpp/mmfs/bin/mmces service stop HDFS -a
      ```

      Note: You need to stop the HDFS Transparency nodes because the Cloudera Manager can only manage HDFS Transparency NameNodes and DataNodes if they are started using the Cloudera Manager.

   b. On the Cloudera Manager node, install Cloudera Manager and ensure that you can log in to the Cloudera Manager GUI.

      Perform the following steps to create a new CDP Private Cloud Base cluster:

- Log in to the Cloudera Manager GUI using the following credentials:

  ```
  username: admin
  password: admin
  ```

- Upload the CDP Private Cloud Base license.

Note: If you are planning to enable Kerberos and TLS in Cloudera Manager in step 2 and step 3 respectively, then Kerberos and TLS should already be enabled on the CES HDFS cluster.

2. Optional: Enable Kerberos in Cloudera Manager.

   It is recommended to enable Kerberos before you proceed to the next step.

   To enable Kerberos, see "Kerberos" on page 315.

3. Optional: Enable Auto-TLS in Cloudera Manager.

   Ensure that you enable Kerberos in Cloudera Manager before you enable auto-TLS.

   To enable TLS, see "Enabling TLS" on page 322.

   In **Cloudera manager** > **Enable Auto-TLS** > **Generate CA** > **Trusted CA Certificates Location**, you could leave the *Trusted CA Certificates Location* field blank so that Cloudera Manager can auto-generate a new certificate.

   If you already have a certificate, enter its path.

4. Deploy the IBM Storage Scale CSD.

   - From the IBM Storage Scale cluster, get the `gpfs.hdfs.cloudera.cdp.csd-<version-number>.noarch.rpm` package and copy it to the Cloudera Manager node. To get the package from the self-extracting installed path, see "Downloads" on page 300.

     For example,

     ```
     /usr/lpp/mmfs/5.1.1.0/hdfs_rpms/rhel7/hdfs_3.1.1.x
     ```

   - As root, log in to the Cloudera Manager node and install the IBM Storage Scale Cloudera Custom Service Descriptor (CDP CSD) package by running the following command:

     ```
     # rpm -ivh /root/gpfs.hdfs.cloudera.cdp.csd-<version-number>.noarch.rpm
     ```

   - Restart the Cloudera Manager server by running the following command:

     ```
     # systemctl restart cloudera-scm-server.service
     ```

   - Check for any errors in the `/var/log/cloudera-scm-server/cloudera-scm-server.log` file.

5. Add a cluster in Cloudera Manager.

   a. Ensure the CES HDFS NameNodes and DataNodes are not running.

   b. Click **Add cluster**.

   c. Enter the CDP Private Cloud Base cluster name and click **Continue**.

   d. Register Hosts, click **Search** and then click **Continue**.

      Note:

      - Before registering the hosts, ensure that the DNS names across the cluster are resolvable. All the hostnames from CES HDFS and CDP Private Cloud Base nodes must return FQDN values.
      - In addition to registering CDP Private Cloud Base hosts, you must register the HDFS Transparency NameNode and DataNode hosts from your CES HDFS cluster. To find the HDFS Transparency nodes, run the following command on the CES HDFS cluster:

        ```
        # /usr/lpp/mmfs/hadoop/sbin/mmhdfs hdfs status
        ```

- For registering the HDFS Transparency hosts to Cloudera Manager, the ssh private key for root or the root password of the CES HDFS nodes must be provided in the Cloudera Manager wizard. After the hosts are registered, you can change the password or remove the private key.

e. Select Repository location. It can be a public Cloudera repository or your own local repository. For local repository, select **Custom Repository** and enter the required details.

f. Under **CDH and other software**, select **Use Parcels** (Recommended) and click **Parcel Repositories and Network Settings**.

g. Add the relevant parcel information in the **Remote Parcel Repository URLs** tab and remove all the URLs that are not relevant. Click **Close** and then click **Continue**.

h. Under **Select JDK**, select **Install a Cloudera-provided version of OpenJDK** and click **Continue**.

Note: Cloudera needs to have the same version of all the common software, including Java™, on all the managed hosts. Otherwise, the hosts might report bad health.

i. Under **Enter Login Credentials**, enter a common userid/password or the ssh private key for the managed hosts and click **Continue**.

j. Under **Install Agents**, wait for all the installations to complete successfully and for the **Install Parcels** window to show up.

k. Under **Install Parcels**, wait for the packages to be downloaded, distributed, unpacked and activated. Click **Continue**.

l. Under **Inspect Cluster**, click **Inspect Network Performance and Inspect Hosts**. Click **Continue**.

6. Install services for CDP Private Cloud Base cluster.

   During the installation of a new CDP Private Cloud Base cluster, you can specify the services that you want to install.

   a. Select the services that you want to install on your CDP Private Cloud Base cluster. IBM Storage Scale service must be included as part of this initial cluster creation.

   Note:

   - CDP Private Cloud Base with IBM Storage Scale is supported only with a new installation setup with IBM Storage Scale service as the file system. The minimum services required to be selected are Zookeeper, Yarn and IBM Storage Scale service. If you are planning to use the Ranger, Solr and Atlas services, it is recommended to include them at the time of initial cluster creation.
   - Creating a CDP Private Cloud Base cluster without the IBM Storage Scale service and then adding the IBM Storage Scale service later is not supported.
   - While creating the new CDP Private Cloud Base cluster, do not include the HDFS service. If you include the HDFS service, unrecoverable errors might occur.
   - Do not place any Hadoop services, other than the IBM Storage Scale service, on the CES HDFS cluster hosts. However, it is permitted and recommended to add Gateway roles for other Hadoop services on the CES HDFS cluster hosts.
   - If you need Ranger, add Solr and Ranger services together with the IBM Storage Scale service at the time of initial cluster creation. Otherwise, if you want to add Ranger later, a workaround is needed for Solr as mentioned in Solr does not start after adding Ranger.
   - If you need Hive, Livy or Oozie, enable the proxyuser settings for HDFS Transparency for these CDP Private Cloud Base services by following Enable proxyuser settings for HDFS Transparency.

     This step is not needed if the CES HDFS cluster was created using the IBM Storage Scale Install Toolkit.
   - Hive on Tez should be installed for HiveServer2 (HS2) for all the Hive tables (managed and external tables). For more information, see Hive on Tez introduction in the Cloudera documentation.
   - If you need Oozie, configure **`dfs.namenode.fs-limits.min-block-size`** = *<dfs.blocksize>* on the client side through the Cloudera Manager. For more information, see item “15” on page 348 on CDP troubleshooting.

b. In the IBM Storage Scale service installation wizard, perform the following:

   i) Assign the NameNode and DataNode roles based on the actual CES HDFS NameNode and DataNode hosts.

   ii) Assign the Gateway roles to one or more CDP Private Cloud Base nodes. Assigning these roles help in creating the HDFS client config xmls under `/etc/hadoop`. These xmls are required to run the HDFS client commands.

   iii) Set the following IBM Storage Scale parameters:

   - **`default_fs_name`** to *hdfs://<myceshost>:8020*
   - **`webhdfs_url`** to *http://<myceshost>:50070/webhdfs/v1*
   - **`transparency.namenode.http.port`** to *50070*. This is default NameNode JMX metrics port.
   - **`transparency.datanode.http.port`** to *1006*. This is DataNode JMX metrics port. The default value is *9864* if Kerberos is disabled and *1006* if Kerberos is enabled.

     Note:

     - In this example, the hostname corresponding to CES IP configured on HDFS Transparency is *<myceshost>*.
     - 8020/50070 are the default RPC and HTTP ports for NameNodes. If you are not using these default ports, update the parameters accordingly.

c. If you are adding Ranger, additional configurations are needed for HDFS Transparency. For information on configuring HDFS Transparency and the required configuration parameters needed for the Ranger service, see "Enabling Ranger" on page 316.

d. Save the changes and then proceed to configure other services, followed by starting all the services. Ensure that all the services have started successfully and that there are no errors.

e. After Cloudera Private Cloud Base is created, additional Kerberos specific inputs may be required for the IBM Storage Scale service. Set the following IBM Storage Scale parameters:

   - Go to **IBM Spectrum Scale service** > **Configuration** > **HDFS Client Advanced Configuration Snippet (Safety Valve) for hdfs-site.xml** and add the following custom parameters:
     - Add the **`dfs.namenode.kerberos.principal.pattern`** parameter and set its value as NameNode principal regular expression. This could be as open as `*`.
     - Add the **`hadoop.security.service.user.name.key.pattern`** parameter and set its value as `*`.
   - **`spectrumscale_keytab`** is the actual path of the keytab file configured for HDFS Transparency NameNode. The default value is `/etc/security/keytabs/nn.service.keytab`. Update this parameter if the default path is not used.
   - **`scale_hdfs_principal_name`** is the actual Kerberos principal configured for HDFS Transparency NameNode. The default value is *nn*. Update this parameter if the default path is not used.

7. Enable NameNode HA.

   Enable HA if the CES HDFS cluster NameNode HA is enabled.

   To enable NameNode HA, see "Enabling NameNode HA" on page 310.

8. Verify the CDP Private Cloud Base with IBM Storage Scale environment.

   To verify the cluster, see "Verifying installation" on page 309.

Verifying installation

After you have deployed CDP Private Cloud Base with the IBM Storage Scale service, verify the installation setup.

1. If the CES HDFS cluster is Kerberos enabled, ensure that Kerberos is properly configured for HDFS Transparency. For more information, see “Verifying Kerberos” on page 127.
2. On the CES HDFS cluster, ensure that there is an active NameNode by running the following commands. If Kerberos is enabled, a Kerberos token for the HDFS user is needed.

   ```
   # kinit -kt /etc/security/keytabs/ces-<clustername>.headless.keytab ces-<clustername>@<Realm
   name> -c /var/mmfs/tmp/krb5cc_ces
   ```

   where, *<clustername>* is the cluster name of your CES HDFS and *<Realm>* is the Realm name of your Kerberos. For example, IBM.COM.

   ```
   # export KRB5CCNAME=/var/mmfs/tmp/krb5cc_ces
   # hdfs haadmin -getAllServiceState
   Namenode1.gpfs.net:8020                           active
   Namenode2.gpfs.net:8020                           standby
   ```

   You must see one active NameNode.
3. To verify the regular HDFS client, run the following command:

   ```
   # hdfs dfs -ls /
   ```

4. Log in to one of the CDP Private Cloud Base cluster node setup with a gateway role for the IBM Storage Scale service. These gateway nodes have the configuration files required by the HDFS client.
5. If Kerberos is enabled, the HDFS client needs a valid Kerberos token before it can access the IBM Storage Scale file system. You can create a regular OS user across all the nodes and then define a user principal and optionally a keytab file corresponding to that user on your KDC server.

   In the following example, a user named *testuser* is created with the principal name as `testuser@IBM.COM` and keytab filename as **`testuser.headless.keytab`**.

   - Create a regular OS user on all the nodes by following step 5 in “Configuring users, groups and file system access for IBM Storage Scale” on page 302.
   - Log into the KDC server and create a Kerberos principal and keytab for the *testuser*.

     ```
     # kadmin.local addprinc -randkey -maxrenewlife 7d +allow_renewable testuser
     # kadmin.local ktadd -k /etc/security/keytabs/testuser.headless.keytab testuser@<Realm
     Name>
     ```

   - Obtain a token for the *testuser*.

     ```
     # kinit -kt /etc/security/keytabs/testuser.headless.keytab testuser@<Realm Name>
      # klist
     ```

6. To verify the regular HDFS client, run the following command:

   ```
   # echo “hello world” > /tmp/hello
   # hdfs dfs -ls /
   # hdfs dfs -put /tmp/hello /tmp
   # hdfs dfs -cat /tmp/hello
   ```

7. To verify WebHDFS client, run the following command:

   ```
   # echo “hello world” > /tmp/hello
   # hdfs dfs -ls webhdfs://<CES HDFS Cluster-name>/
   # hdfs dfs -put /tmp/hello webhdfs://<CES HDFS Cluster-name>/tmp/
   # hdfs dfs -cat webhdfs://<CES HDFS Cluster-name>/tmp/hello
   ```

   where, *<CES HDFS Cluster-name>* is the name of your HDFS namespace.

Configuring

Enabling NameNode HA

This topic lists the steps to enable NameNode HA within CDP Private Cloud Base with IBM Storage Scale service in Cloudera Manager.

Note: Because the compute and storage architecture are decoupled, the server-side administration of NameNode HA is managed by the IBM Storage Scale CES protocol. Unlike native HDFS, Zookeeper/zkfc is not used for IBM Storage Scale NameNode HA.

The following are the two steps to the HA enablement process:

- Server side: NameNode HA can be enabled in the CES HDFS cluster during the installation and deployment using the IBM Storage Scale installation toolkit. However, if NameNode HA is not enabled on your CES HDFS cluster, follow "Change CES HDFS NON-HA cluster into CES HDFS HA cluster" on page 67 to enable it.
- Client side: Now enable the NameNode HA for the IBM Storage Scale service in the Cloudera Manager by enabling the NameNode HA for the CDP Private Cloud Base cluster. This means that when a NameNode failover event occurs in the IBM Storage Scale CES HDFS cluster, HDFS clients and Hadoop workloads running on the CDP Private Cloud Base cluster retry to connect similar native HDFS HA environments.

In the following procedure, the HDFS Transparency cluster name is *<cluster-name>* and the hostname corresponding to CES IP configured on HDFS Transparency is *<myceshost>*:

1. From the Cloudera Manager GUI, stop all services.
2. In the Cloudera Manager GUI, modify **`default_fs_name`** (Default File System URL) from *hdfs://<myceshost>:8020* to *hdfs://<cluster-name>*.
3. In the Cloudera Manager GUI, add the following configurations:
   - Click **SpectrumScale** > **Actions Configuration** > **Cluster-wide Advanced Configuration Snippet (Safety Valve) for core-site.xml** and set the custom parameter to the following value:

     `fs.defaultFS` to *hdfs://<cluster-name>*
   - Click **Spectrum Scale** > **Configuration** > **Transparency NameNode Advanced Configuration Snippet (Safety Valve) for core-site.xml** and set the same custom fs.defaultFS parameter to the following value:

     `fs.defaultFS` to *hdfs://<cluster-name>*
   - Click **SpectrumScale** > **Configuration** > **HDFS Client Advanced Configuration Snippet (Safety Valve) for hdfs-site.xml** and set the custom parameters to the following values:
     - **`dfs.nameservices`** to *<cluster-name>*
     - **`dfs.ha.namenodes.<cluster-name>`** to *nn1*
     - **`dfs.namenode.rpc-address.<cluster-name>.nn1`** to *<myceshost>:8020*
     - **`dfs.namenode.http-address.<cluster-name>.nn1`** to *<myceshost>:50070*
     - **`dfs.client.failover.proxy.provider.<cluster-name>`** to *org.apache.hadoop.hdfs.server.namenode.ha.ConfiguredFailoverProxyProvider*
     - If the CES HDFS cluster is NameNode HA enabled, set **`dfs.namenode.https-address.<cluster-name>.nn1`** to *<myceshost>:50470*.
     - Under **SpectrumScale** > **Configuration**, search for **`webhdfs_url`** parameter and set the value to *blank*.

   Note: 8020, 50070 and 50470 are the default RPC, HTTP and HTTPS ports for NameNode. If you are not using these default ports, you must update the parameters accordingly.
4. Save the changes.

5. Restart the services with stale client configuration.
6. To view the NameNode Active/Standby states using IBM Storage Scale CLI, see “Monitoring NameNodes” on page 340 and to view the NameNode Active/Standby states using Cloudera Manager, see “Monitoring” on page 340.

Dual-network deployment

Installation toolkit does not support the dual-network. Therefore, you need to configure the IBM Storage Scale dual network manually.

This section describes the recommended network configuration for CDP Private Cloud Base, HDFS Transparency and IBM Storage Scale cluster if more than one network is configured in the environment.

For example:

```
[root@c902f09x05 ~]# mmlscluster
GPFS cluster information
========================
  GPFS cluster name:         SS5022.gpfs.net
…………….
…………….
 Node  Daemon node name          IP address   Admin node name      Designation
--------------------------------------------------------------------------------
   1   c902f09x05-eth4.gpfs.net  128.20.1.26  c902f09x05.gpfs.net  quorum
   2   c902f09x07-eth4.gpfs.net  128.20.1.28  c902f09x07.gpfs.net  quorum
   3   c902f09x08-eth4.gpfs.net  128.20.1.29  c902f09x08.gpfs.net  quorum
   4   c902f09x06-eth4.gpfs.net  128.20.1.27  c902f09x06.gpfs.net
```

In the above example, the `Daemon node name` and `IP address` fields correspond to the Daemon network used for data traffic in IBM Storage Scale and the `Admin node name` corresponds to the network used for running IBM Storage Scale administration commands (such as **`mmlscluster`**, **`mmgetstate`** etc).

The `Admin node name` and `Daemon node name` can be changed by using the **mmchnode** command. For more information, see *GPFS node adapter interface names* in the *IBM Storage Scale: Concepts, Planning, and Installation Guide*.

In a dual network environment, there are two networks: Network 1 and Network 2. The following are the recommended network setup configuration options for the IBM Storage Scale cluster:

1. Deploy Cloudera components, HDFS Transparency, CES IP/Hostname and IBM Storage Scale Admin network in a common network. For example, Network 1.

   The CDP Private Cloud Base service daemons (for example, Yarn ResourceManager) and HDFS Transparency daemons (for example, NameNode) should be in the same network to be able to communicate with each other over RPC.
2. Deploy IBM Storage Scale daemon network on the other network. For example, Network 2. Usually this is the high-speed network for IBM Storage Scale data traffic.

The separation of IBM Storage Scale admin and daemon networks offers the following benefits:

- If the cluster is busy with heavy I/O, using the same network for admin and daemon can cause administrative commands to run slower.
- IBM Storage Scale Admin network requires passwordless-ssh from one central node to the other nodes. The daemon network does not require ssh. This can help to restrict ssh access to only one selected network to address security concerns.

Updating configuration

This section provides a high-level overview on making the configuration changes applicable to HDFS Transparency in a CDP integrated environment.

Manually updating the configuration

HDFS has both the server and client configurations. Unlike Cloudera Hortonworks Data Platform (HDP) Ambari that manages the HDFS server and client configurations with a common set of .xml files, Cloudera Manager (CM) configures the HDFS server and client configuration separately.

For more information, see Cloudera Manager server and client configuration section in the CDP Private Cloud Base documentation.

After IBM Storage Scale is integrated, Cloudera Manager no longer manages the HDFS server-side configurations. Cloudera Manager manages only the HDFS client-side configurations. This aligns with the separation of compute and storage architecture between Cloudera CDP Private Cloud Base and IBM Storage Scale.

The following table lists an example of how the `hdfs-site.xml` parameters specific to the server and the client roles are managed:

Table 33. Example showing `hdfs-site.xml` parameters management

Configuration files	**Configuration management**	**Comments**
HDFS server configuration	IBM Storage Scale CCR repository will sync the .xml files into the `/var/mmfs/hadoop/etc/hadoop` directory	
	The values in **Cloudera Manager GUI** > **IBM Spectrum Scale service** > **Transparency NameNode Advanced Configuration Snippet (Safety Valve) for hdfs-site.xml** are synced to a private per-process directory, under `/var/run/cloudera-scm-agent/process/process-name`. For more information, see Cloudera Manager server and client configuration.	These Configurations are presented in Cloudera Manager GUI but are not used by IBM Storage Scale HDFS Transparency.
HDFS client configurations	The values in **Cloudera Manager GUI** > **IBM Spectrum Scale service** > **Configuration** > **HDFS Client Advanced Configuration Snippet (Safety Valve) for hdfs-site.xml** are synced to the `/etc/hadoop/conf` directory on the nodes that have the IBM Storage Scale Gateway roles so that it can be consumed by other Cloudera services and HDFS clients.	This is HDFS client shipped with CDP Private Cloud Base that is leveraged by other CDP services.
	The HDFS .xml files are located in the `/var/mmfs/hadoop/etc/hadoop` directory.	This is the HDFS client shipped with HDFS Transparency. This HDFS client is not commonly used in a CDP integrated environment.

Important: The procedure to update the configuration varies depending on which HDFS component is being changed.

- HDFS server components:
  - HDFS Transparency NameNodes
  - HDFS Transparency DataNodes
- HDFS client components:
  - The hdfs, webhdfs commands and Java APIs from the HDFS client shipped with CDP Private Cloud Base .
  - The hdfs, webhdfs commands and Java APIs from IBM Storage Scale under `/usr/lpp/mmfs/hadoop/bin/` directory.

Follow the specific process to update the configuration based on the server or client configurations from CDP Private Cloud Base or IBM Storage Scale HDFS Transparency:

- “Updating only the HDFS server configurations” on page 313
- “Updating only the HDFS client (CDP Private Cloud Base) configurations” on page 313
- “Updating Ranger configurations” on page 313
- “Updating Kerberos configurations” on page 314

Updating only the HDFS server configurations

To update the server-side configuration (for example, **dfs.namenode.handler.count** value), run the following steps:

1. Stop the HDFS Transparency services from the Cloudera Manager GUI by clicking **Cloudera Manager** > **IBM Spectrum Scale service** > **Stop**.
2. Log in to one of the CES HDFS NameNodes and update the server-side configuration by running the **mmhdfs config set** command and then uploading the changed configuration to the IBM Storage Scale CCR repository using the **mmhdfs config upload** command.

   ```
   # mmhdfs config set hdfs-site.xml -k dfs.namenode.handler.count=800
   # /usr/lpp/mmfs/hadoop/bin/mmhdfs config upload
   ```

   For more information on the **mmhdfs** command, see *IBM Storage Scale: Command and Programming Reference Guide*.
3. Start the HDFS Transparency services from the Cloudera Manager GUI by clicking **Cloudera Manager** > **IBM Spectrum Scale service** > **Start**.

   Note: You must start the HDFS Transparency services from Cloudera Manager so that Cloudera can display the states of the NameNodes and DataNodes properly.

Updating only the HDFS client (CDP Private Cloud Base) configurations

To change the client-only configuration (for example, adding or updating the *dfs.client.** values in hdfs-site.xml), run the following steps:

- On the Cloudera Manager GUI, click **Cloudera Manager** > **IBM Spectrum Scale service** > **Configuration** > **HDFS Client Advanced Configuration Snippet (Safety Valve) for hdfs-site.xml** > **Update the configuration** > **Save**.
- On the Cloudera Manager GUI, click **Deploy the client configuration** to propagate the updated client configuration from the Cloudera Manager database to /etc/hadoop/conf.

  Note: You do not need to restart the HDFS Transparency service for the changes to take effect.

Updating Ranger configurations

A ranger is closely integrated with the HDFS server and client. The Ranger plug-in runs within the NameNode process space. When the Ranger plug-in is enabled for HDFS, ranger-specific .xml files (ranger-hdfs-security.xml, ranger-hdfs-policymgr-ssl.xml and ranger-hdfs-audit.xml) are generated within a private directory specific to the HDFS Transparency NameNode process, under the /var/run/cloudera-scm-agent/process/process-name directory. For more information, see Cloudera Manager server and client configuration.

When you restart the HDFS Transparency NameNodes from the Cloudera Manager GUI, the Ranger configuration files are synced to the /var/mmfs/hadoop/etc/hadoop HDFS Transparency configuration directory. But these updates are not uploaded to the IBM Storage Scale CCR repository. Therefore, any Ranger specific configuration changes require a workaround to get into the CCR.

To start the HDFS Transparency NameNodes correctly, update the IBM Storage Scale CCR by following NameNodes do not start after updating the Ranger configuration.

Updating Kerberos configurations

Cloudera Manager does not manage Kerberos for the IBM Storage Scale service because the CDP Private Cloud Base cluster and the CES HDFS cluster are different clusters and are loosely integrated. Therefore, Kerberos setup needs to be manually enabled first on the CES HDFS cluster.

In the Cloudera Manager GUI, the **Enable Kerberos** action under **Cluster name** > **Action** has no effect on the HDFS server-side configuration. This enablement only enables Kerberos for the HDFS client-side. The HDFS client-side configuration files under `/etc/hadoop/conf` are updated to reflect the updates from the Cloudera Manager Kerberos enablement.

To make Kerberos-specific changes to HDFS, see "Updating only the HDFS server configurations" on page 313.

Note:

- Cloudera Manager requires the following Kerberos-specific information from HDFS Transparency during the initial deployment to create the configuration files and directories properly:
  - NameNode keytab location (parameters: **`spectrumscale_keytab`**, **`service-wide`**)
  - HDFS Principal Name (parameters: **`scale_hdfs_principal_name`**, **`service-wide`**)
- If the default NameNode principal name (nn) or NameNode keytab path (`/etc/security/keytab/nn.service.keytab`) on the HDFS server side is changed, the corresponding parameters must also be changed in Cloudera Manager.

Administering

Adding nodes

Perform the following steps before adding the new NameNodes and DataNodes to the CDP Private Cloud Base cluster:

1. Before adding the new nodes to the CDP Private Cloud Base cluster, you must add them to the IBM Storage Scale HDFS Transparency cluster. For information on adding the nodes to the HDFS Transparency cluster, see Administration guide.
2. If Kerberos is enabled, see additional steps under "Prerequisites for Kerberos" on page 64.
3. If TLS is enabled, see step 2.d under Enabling TLS for HDFS Transparency to copy the IBM Storage Scale certificates files (`spectrum_scale_ces_hdfs_truststore.jks`, `spectrum_scale_ces_hdfs_keystore.jks` and `spectrum_scale_ces_hdfs_cacerts.pem`) from the existing HDFS Transparency cluster to the new node.
4. As IBM Storage Scale is a POSIX file system, Hadoop users and groups needed by CDP Private Cloud Base cluster must have the same UID and GID across all the IBM Storage Scale nodes. To create the users and groups on the new node, run step 1 through step 3 under "Configuring users, groups and file system access for IBM Storage Scale" on page 302.

Adding a new NameNode

This topic lists the steps to add a new NameNode.

On the CES HDFS cluster, perform the following:

1. Add the new NameNode to your CES HDFS cluster by following the steps in CES HDFS Administration section.
2. Stop the new HDFS Transparency NameNode by running the following command on the new node:

   ```
   # /usr/lpp/mmfs/bin/mmces service stop HDFS
   ```

   You need to stop the HDFS Transparency NameNode because the Cloudera Manager can only manage the HDFS Transparency nodes when they are started using the Cloudera Manager.

On the Cloudera Manager GUI, perform the following:

1. Click **IBM Spectrum Scale** > **Instances** > **Add Role Instances to add the new NameNode**.
2. Restart all the services in the Cloudera Manager.

Adding a new DataNode

This topic lists the steps to add a new DataNode.

On the CES HDFS cluster, perform the following:

1. Add the new DataNode to your CES HDFS cluster by following the steps in CES HDFS Administration section.
2. Stop the new HDFS Transparency DataNode by running the following command on the new node:

   ```
   #/usr/lpp/mmfs/hadoop/sbin/mmhdfs datanode stop
   ```

   You need to stop the HDFS Transparency DataNode because the Cloudera Manager can only manage the HDFS Transparency nodes when they are started using the Cloudera Manager.

On the Cloudera Manager GUI, perform the following:

1. Click **IBM Spectrum Scale** > **Instances** > **Add Role Instances to add the new DataNode**.
2. Start the new DataNode from the Cloudera Manager.

Kerberos

This section lists the information on enabling and disabling Kerberos.

Enabling Kerberos for CES HDFS and CDP Private Cloud Base

This topic lists the steps to enable Kerberos on the CES HDFS and CDP Private Cloud Base clusters.

Cloudera Manager does not manage Kerberos for the IBM Storage Scale service because CDP Private Cloud Base cluster and CES HDFS cluster are different clusters and are loosely integrated. Therefore, Kerberos setup needs to be manually enabled first on the CES HDFS cluster. To enable Kerberos on the CES HDFS cluster, you need to create the service principals and keytabs for HDFS Transparency NameNodes and DataNodes and also ensure that the HDFS client can access the IBM Storage Scale file system. Then enable Kerberos for the CDP Private Cloud Base cluster which kerberizes the rest of the CDP Private Cloud Base services. For example, YARN, Zookeeper, Hive etc.

Overview flow for Kerberos is as follows:

1. Enable Kerberos for HDFS Transparency
2. Enable Kerberos for Cloudera Manager
3. Enable Kerberos for IBM Storage Scale service

Enabling Kerberos for HDFS Transparency

1. From Cloudera Manager stop all the services including the IBM Storage Scale service.
2. You can use the KDC server that is set up for your CDP Private Cloud Base cluster. If the KDC server has not been set up, create a new KDC server by following the "Setting up the Kerberos server" on page 108 section for manual setup or follow the "Configuring Kerberos using the Kerberos script provided with IBM Storage Scale" on page 117 section.
3. Enable Kerberos configurations for HDFS Transparency by following the "Setting up Kerberos for HDFS Transparency nodes" on page 109 section for manual setup or follow the "Configuring Kerberos using the Kerberos script provided with IBM Storage Scale" on page 117 section.
4. Follow the "Verifying Kerberos" on page 127 section to ensure that Kerberos is properly configured for HDFS Transparency.

Enabling Kerberos for Cloudera Manager

1. On the Cloudera Manager GUI, click **Cluster name** > **Action** > **Enable Kerberos** and enter the following details:
    a. Select aes256-cts-hmac-sha1-96 as the encryption type.
    b. Enter the KDC Admin principal (for example, root/admin) and password.
    c. Enter the KDC Realm.
    d. Enter the KDC hostname FQDN.
    e. Do not manage the kerberos.conf file through Cloudera. Therefore, uncheck the box.
    f. Copy the /etc/krb5.conf file from your KDC server host to all the CDP Private Cloud Base cluster hosts.
    g. For Cloudera manager to create principals for the Hadoop services, enter account credentials for the KDC account that has the permissions to create other principal.

Enabling Kerberos for IBM Storage Scale service

1. Add the IBM Storage Scale parameters needed for Kerberos by following the steps under "Installing Cloudera Data Platform Private Cloud Base with IBM Storage Scale" on page 304.
2. Verify that the HDFS client from the CDP Private Cloud Base nodes can access the storage through HDFS Transparency.
    a. Verify that you can list the directories/files from the HDFS client. Run the following command from any CDP Private Cloud Base host:

    ```
    hadoop fs -ls /
    ```

    b. Verify that the WebHDFS client can also list the directories/files. Run the following command from any CDP Private Cloud Base host:

    ```
    hadoop fs -ls webhdfs://<cluster-name>
    ```

Disabling Kerberos

You cannot disable Kerberos for the CDP Private Base cluster after enabling it. Cloudera does not have the option to disable Kerberos.

Ranger

Ranger provides a centralized authorization by extensible plug-in-based model. Ranger plug-ins run in the host components. For example, NameNode, HiveServer2, Kafka Broker etc.

Ranger provides the following functionalities:

- Provides role-based access control to HDFS resources. For example, it provides rwx policies to HDFS files and directories. Also, these policies can be defined on the Ranger Admin GUI.
- Leverages Solr to store temporary audit logs.
- Leverages HDFS Transparency to store persistent audit logs.

The Ranger plug-in is loaded by the NameNode at startup. The plug-in downloads Ranger policies from the Ranger Admin server periodically and caches them locally. This ensures better performance as the NameNode can look up the policies locally rather than having to connect to the Ranger server every time an authorization check is needed.

The parameter **ranger.plugin.hdfs.policy.pollIntervalMs** in ranger-hdfs-security.xml determines how aggressively the plug-in would perform caching of policies. It defaults to 30 seconds.

Enabling Ranger

This topic lists the steps to enable Ranger.

Prerequisites

- Ensure that the CES HDFS cluster is functional. If the IBM Storage Scale service is already added to Cloudera Manager, ensure that the CDP Private Cloud Base cluster that is integrated with the IBM Storage Scale service is functional as well. For verifying, follow the steps listed in "Verifying installation" on page 309.
- Before you enable Ranger, Kerberos must be enabled on CES HDFS and the CDP Private Cloud Base clusters.
- All the CDP Private Cloud Base services should have Kerberos enabled. Click **Administration** > **Security** > **Status tab** > **Kerberos** > **Enabled check-box**. You must see all the services enabled for Kerberos. For more information, see "Problem determination" on page 344.
- Before creating the database for Ranger, ensure that you perform a workaround for MySQL/MariaDB by following Installing Ranger service may fail with the following SQL error from MySQL/MariaDB.

It is recommended to add the Solr and Ranger services together with the IBM Storage Scale service at the time of initial CDP Private Cloud Base cluster creation. However, you can add these services later as well.

Overall flow required to enable Ranger is as follows:

1. If the CDP Private Cloud Base cluster with IBM Storage Scale already exists, then stop all the cluster services.
2. Add the Solr and Ranger services in Cloudera Manager.
3. Configure the IBM Storage Scale service for Ranger.
4. Configure the CES HDFS cluster for Ranger.
5. Start all CDP Private Cloud Base cluster services.

Procedure

1. Stop all the CDP Private Cloud Base cluster services from Cloudera Manager.

   If the HDFS Transparency services were started manually using the `mmhdfs`/`mmces` commands, stop them as well.
2. Add the Solr and Ranger services.

   If Solr is being added to an existing CDP Private Cloud Base cluster with IBM Storage Scale, then the following workaround is needed for Solr to start properly:

   - In Cloudera Manager console, click **Solr** > **Configuration**, search for ZNode and set the value of the Solr configuration parameter **`ZooKeeper ZNode`** to */solr-infra*.
   - Ensure that the **Kerberos configuration** checkbox is enabled for the Solr and Ranger services.
   - After adding Ranger, the Solr service changes its name to *CDP-INFRA-SOLR*.
3. Configure the IBM Storage Scale service for Ranger.

   a. Click **IBM Spectrum Scale** > **Configuration**. Search for **`hadoop.security.authorization`** and enable this option by clicking on the checkbox.

   b. Go to Cloudera Manager, click **IBM Spectrum Scale** > **Configuration** > **Transparency NameNode Advanced Configuration Snippet (Safety Valve) for ranger-hdfs-security.xml**, and add the following custom configuration:

   ```
   Name: ranger.plugin.hdfs.policy.rest.ssl.config.file
   Value: ranger-hdfs-policymgr-ssl.xml
   ```

   c. Click **IBM Spectrum Scale** > **Configuration** > **Transparency NameNode Advanced Configuration Snippet (Safety Valve) for ranger-hdfs-policymgr-ssl.xml** and add the following custom configurations:

   ```
           <property>
               <name>xasecure.policymgr.clientssl.truststore</name>
               <value>/var/lib/cloudera-scm-agent/agent-cert/
   spectrum_scale_ces_hdfs_truststore.jks</value>
           </property>
           <property>
               <name>xasecure.policymgr.clientssl.truststore.type</name>
   ```

```
            <value>jks</value>
        </property>
        <property>
            <name>xasecure.policymgr.clientssl.truststore.credential.file</name>
            <value>jceks://file/var/lib/cloudera-scm-agent/agent-cert/
rangerpluginssl.jceks</value>
        </property>
        <property>
            <name>hadoop.security.credential.provider.path</name>
            <value>jceks://file/var/lib/cloudera-scm-agent/agent-cert/
rangerpluginssl.jceks</value>
        </property>
```

d. If you are using CDP Private Cloud Base 7.1.8 or later, in Cloudera Manager navigate to **Core Configuration Service** > **Configuration** and search for `Additional Rules to Map Kerberos Principals to Short Names`.

If you are using the earlier versions of CDP Private Cloud Base, in Cloudera Manager navigate to **IBM Spectrum Scale** > **Configuration** and search for `Additional Rules to Map Kerberos Principals to Short Names`.

Then, change the **`{DEFAULT_RULES}`** value to the following set of rules as needed for Ranger and HDFS Transparency:

```
RULE:[2:$1@$0](rangeradmin@IBM.COM)s/(.*)@IBM.COM/ranger/
RULE:[2:$1@$0](rangertagsync@IBM.COM)s/(.*)@IBM.COM/rangertagsync/
RULE:[2:$1@$0](rangerusersync@IBM.COM)s/(.*)@IBM.COM/rangerusersync/
RULE:[2:$1@$0](rangerkms@IBM.COM)s/(.*)@IBM.COM/keyadmin/
RULE:[2:$1/$2@$0](nn/.*@.*IBM.COM)s/.*/hdfs/
RULE:[2:$1/$2@$0](dn/.*@.*IBM.COM)s/.*/hdfs/
RULE:[1:$1@$0](hdfs@IBM.COM)s/@.*//
RULE:[1:$1@$0](.*@IBM.COM)s/@.*//
```

Note: `Additional Rules to Map Kerberos Principals to Short Names` is translated to **`auth_to_local`** rules in `core-site.xml`. Cloudera Manager appends some more rules followed by 'DEFAULT' as the last line. Do not add 'DEFAULT' as the last line in this text field because it prevents the translation of the rules appended by Cloudera Manager.

Replace IBM.COM with your Kerberos Realm name in the above example rules.

e. Save and deploy the client configuration. Do not start any services yet.

4. Configure HDFS Transparency for Ranger.

a. Enabling Ranger requires setting the server-side configuration on the HDFS Transparency NameNodes.

i) Update the HDFS Transparency configuration files and upload the changes.

- Log in to one of the CES HDFS NameNode.
- Get the config files by running the following commands:

```
# mkdir /tmp/hdfsconf
# mmhdfs config export /tmp/hdfsconf core-site.xml,hdfs-site.xml,hadoop-env.sh,gpfs-
site.xml
# cd /tmp/hdfsconf/
```

- Update the config files in `/tmp/hdfsconf` with the following changes based on your environment:

File: core-site.xml

```
<property>
<name>hadoop.security.auth_to_local</name>
<value>
RULE:[2:$1@$0](rangeradmin@IBM.COM)s/(.*)@IBM.COM/ranger/
RULE:[2:$1@$0](rangertagsync@IBM.COM)s/(.*)@IBM.COM/rangertagsync/
RULE:[2:$1@$0](rangerusersync@IBM.COM)s/(.*)@IBM.COM/rangerusersync/
RULE:[2:$1@$0](rangerkms@IBM.COM)s/(.*)@IBM.COM/keyadmin/
….. <other existing rules>
DEFAULT
```

```
</value>
</property>
```

File: hdfs-site.xml

```
  <property>
    <name>dfs.permissions</name>
    <value>true</value>
  </property>
  <property>
    <name>dfs.permissions.enabled</name>
    <value>true</value>
  </property>
  <property>
    <name>dfs.permissions.ContentSummary.subAccess</name>
    <value>true</value>
  </property>
 <property>
    <name>dfs.namenode.inode.attributes.provider.class</name>
    <value>org.apache.ranger.authorization.hadoop.RangerHdfsAuthorizer</value>
  </property>
```

File: gpfs-site.xml

```
<property>
    <name>gpfs.ranger.enabled</name>
    <value>scale</value>
  </property>
```

File: hadoop-env.sh

Note: Based on your environment, substitute the right path to the CDH ranger-hdfs-plugin library.

```
for f in /opt/cloudera/parcels/CDH/lib/ranger-hdfs-plugin/lib/*.jar;
do
export HADOOP_CLASSPATH=$HADOOP_CLASSPATH:$f
done

for f in /opt/cloudera/parcels/CDH/lib/hadoop/client/jersey-client.jar;
do
export HADOOP_CLASSPATH=$HADOOP_CLASSPATH:$f
done
export HADOOP_CLASSPATH=$HADOOP_CLASSPATH: /opt/cloudera/parcels/
CDH-7.1.7-1.cdh7.1.7.p0.15945976/lib/hadoop/client/jackson-core-asl.jar
```

When Auto-TLS is enabled in Cloudera Manager and either of the following conditions are true, additional configuration is needed:

- Ranger service is enabled after the creation of the initial IBM Storage Scale integrated CDP cluster
- auto-TLS is enabled in Cloudera manager after the creation of the initial IBM Storage Scale integrated CDP cluster
- Ranger High Availability (HA) is enabled

File: hadoop-env.sh

```
export HADOOP_CREDSTORE_PASSWORD=none
```

ii) Import the files into CES HDFS cluster by running the following command:

```
# mmhdfs config import /tmp/hdfsconf  core-site.xml
# mmhdfs config import /tmp/hdfsconf  hdfs-site.xml
# mmhdfs config import /tmp/hdfsconf  hadoop-env.sh
```

iii) Upload the changes to CES HDFS cluster by running the following command.

```
# mmhdfs config upload
```

iv) Additional configurations when TLS is enabled:

- Ranger needs the IBM Storage Scale CES HDFS Truststore password to be encrypted in a `jceks` file. On a CES NameNode, create the `jceks` file for Ranger with the following command:

  ```
  # java -cp "/opt/cloudera/parcels/CDH/lib/ranger-hdfs-plugin/install/lib/*"
  org.apache.ranger.credentialapi.buildks create "sslTrustStore" -value
  <truststore_password> -provider "jceks://file/var/lib/cloudera-scm-agent/agent-cert/
  rangerpluginssl.jceks" -storetype "jceks"
  ```

  Replace **`<truststore_password>`** with the corresponding actual passwords from under "Manually enabling TLS for HDFS Transparency" on page 131, or if the automation script was used then the same can be retrieved from the `/var/mmfs/hadoop/etc/hadoop/ssl-server.xml` file.

- Validate the above `jceks` file (optional):

  ```
  # HADOOP_CREDSTORE_PASSWORD=none java -cp /opt/cloudera/cm/lib/security-
  *.jar com.cloudera.enterprise.crypto.GenericKeyStoreTypePasswordExtractor "jceks"
  "/var/lib/cloudera-scm-agent/agent-cert/rangerpluginssl.jceks" "sslTrustStore"
  ```

- Distribute the above `jceks` file to all the CES HDFS NameNodes.

  For each CES HDFS, run the following command:

  ```
  scp /var/lib/cloudera-scm-agent/agent-cert/rangerpluginssl.jceks root@<CES HDFS
  Host>:/var/lib/cloudera-scm-agent/agent-cert/
  ```

v) Ensure that you create ranger, rangertagsync, rangerusersync and keyadmin user using the `gpfs_create_hadoop_users_dirs.py` script. Log in to a CES HDFS NameNode and run the following commands:

```
/usr/lpp/mmfs/hadoop/scripts/gpfs_create_hadoop_users_dirs.py --create-custom-hadoop-
user-group ranger
/usr/lpp/mmfs/hadoop/scripts/gpfs_create_hadoop_users_dirs.py --create-custom-hadoop-
user-group rangertagsync
/usr/lpp/mmfs/hadoop/scripts/gpfs_create_hadoop_users_dirs.py --create-custom-hadoop-
user-group rangerusersync
/usr/lpp/mmfs/hadoop/scripts/gpfs_create_hadoop_users_dirs.py --create-custom-hadoop-
user-group keyadmin
```

5. Start services:

   - If TLS is enabled, start the NameNodes using the workaround mentioned in issue and then start all the services from Cloudera Manager.
   - If TLS is not enabled, start all the services from Cloudera Manager as usual.

Verifying Ranger policy

After Ranger is configured, you should verify that the Ranger-based resource control is working properly.

Go to **Cloudera Manager** > **Ranger Admin GUI** > **HDFS service** > **Plugin Status**.

If the Ranger HDFS plug-in running with the HDFS Transparency NameNode and the Ranger Admin service are able to communicate with each other successfully, you should see an active HDFS plug-in under this tab.

This section shows an example (Kerberos enabled) of how to verify a HDFS resource policy using the *testuser* OS user id which is created in step 5 of "Verifying installation" on page 309.

1. Log in to any CDP node and obtain a Kerberos token for the *testuser* principal.

   ```
   # kinit -kt /etc/security/keytabs/testuser.headless.keytab testuser@<Realm Name>
   ```

2. Create a HDFS resource and ensure that *testuser* can append write to the HDFS resource:

   ```
   # echo “hello test ranger policy” > /tmp/rangertest
   # hdfs dfs -put /tmp/rangertest /tmp
   # hadoop fs -appendToFile /tmp/rangertest /tmp/rangertest
   ```

3. Create a Ranger policy for the `/tmp/rangertest` HDFS resource:

   a. Log into the Ranger Admin GUI

      - No SSL: http://<Ranger Admin host>:6080
      - SSL: https://<Ranger Admin host>:6182

   b. Create a policy for HDFS resource `/tmp/rangertest` to allow only read access for the testuser.

   c. Set **`Deny All Other Accesses`** to *TRUE*.

4. Save the policy and wait for 20 seconds (configurable using **`ranger.plugin.hdfs.policy.pollIntervalMs`**) for the policy to be downloaded to the NameNode.

5. Ensure that as *testuser* you can READ the file. As the Ranger policy is in place, `WRITE` must be denied.

```
# hdfs dfs -cat /tmp/rangertest
# hdfs dfs -appendToFile /tmp/rangertest /tmp/rangertest
appendToFile: Permission denied: user=testuser, access=WRITE, inode=/tmp/rangertest
```

   Note: Ranger policies are not enforced for users that are a part of the HDFS supergroup **`dfs.permissions.superusergroup`**, as they are considered as superusers of the HDFS file system. Therefore, use a regular user ID to validate the Ranger policies.

Disabling Ranger

This topic lists the steps to disable Ranger.

Perform the following steps on the CDP Private Cloud Base cluster and the CES HDFS cluster to disable Ranger:

1. Configure CES HDFS cluster.

   a. Stop all the services from the Cloudera Manager by clicking **Cluster-name** > **Actions** > **Stop**.

   b. Update the HDFS Transparency configuration files and upload the changes.

      - Get the config files by running the following commands:

```
# mkdir /tmp/hdfsconf
# mmhdfs config export /tmp/hdfsconf   core-site.xml,hdfs-site.xml,hadoop-env.sh
```

      - Remove the following Ranger specific configurations from the following config files in `/tmp/hdfsconf`:

        File: core-site.xml

```
<property>
<name>hadoop.security.auth_to_local</name>
<value>
RULE:[2:$1@$0](rangeradmin@IBM.COM)s/(.*)@IBM.COM/ranger/
RULE:[2:$1@$0](rangertagsync@IBM.COM)s/(.*)@IBM.COM/rangertagsync/
RULE:[2:$1@$0](rangerusersync@IBM.COM)s/(.*)@IBM.COM/rangerusersync/
….. <other existing rules>
DEFAULT
</value>
</property>
```

        File: hdfs-site.xml

```
  <property>
    <name>dfs.permissions</name>
    <value>true</value>
  </property>
  <property>
    <name>dfs.permissions.enabled</name>
    <value>true</value>
  </property>
  <property>
    <name>dfs.permissions.ContentSummary.subAccess</name>
    <value>true</value>
  </property>
 <property>
    <name>dfs.namenode.inode.attributes.provider.class</name>
    <value>org.apache.ranger.authorization.hadoop.RangerHdfsAuthorizer</value>
  </property>
```

File: hadoop-env.sh

Note: Based on your environment, substitute the right path to the CDH ranger-hdfs-plugin library.

```
for f in /opt/cloudera/parcels/CDH/lib/ranger-hdfs-plugin/lib/*.jar;
do
export HADOOP_CLASSPATH=$HADOOP_CLASSPATH:$f
done

export HADOOP_CLASSPATH=$HADOOP_CLASSPATH:/root/postgres-jar/postgresql-42.1.4.jre7.jar

for f in /opt/cloudera/parcels/CDH/lib/hadoop/client/jersey-client.jar;
do
export HADOOP_CLASSPATH=$HADOOP_CLASSPATH:$f
done
```

- Import the files into CES HDFS cluster by running the following command:

```
# mmhdfs config import /tmp/hdfsconf  core-site.xml
# mmhdfs config import /tmp/hdfsconf  hdfs-site.xml
# mmhdfs config import /tmp/hdfsconf  hadoop-env.sh
```

- Upload the changes to CES HDFS cluster by running the following command.

```
# mmhdfs config upload
```

2. Configure the IBM Storage Scale service to disable Ranger.
   - Click **IBM Spectrum Scale** > **Configuration**. Then search for **hadoop.security.authorization** and disable this option by unchecking the check box.
   - Save and deploy the client configuration.
   - Start all the services from the Cloudera Manager.

Transport Layer Security (TLS)

This section describes how to configure TLS on CDP Private Cloud Base clusters integrated with CES HDFS.

Transport Layer Security (TLS)/Secure Sockets Layer (SSL) provides privacy and data integrity between applications communicating over a network by encrypting the packets transmitted between the endpoints. Configuring TLS/SSL for any system typically involves creating a private key and a public key for use by the server and client processes to negotiate an encrypted connection at runtime.

Enabling TLS

This topic lists the steps to enable Transport Layer Security (TLS) on CDP Private Cloud Base clusters integrated with CES HDFS.

Following are the steps to enable TLS on CDP Private Cloud Base clusters. Each of these steps is explained below in detail.

1. Enable TLS for HDFS Transparency
2. Enable Auto-TLS for Cloudera Manager
3. Create Cloudera Data Platform Private Cloud Base with IBM Storage Scale
4. Stop all services from IBM Cloudera Manager
5. Update Cloudera and IBM Storage Scale Trust Stores
6. Update TLS configurations for IBM Storage Scale service
7. Update Metrics configurations for IBM Storage Scale service.
8. Start all services from IBM Cloudera Manager

Note:

- The CDP Private Cloud Base cluster and the CES HDFS Transparency cluster must have Kerberos enabled before you enable auto-TLS on Cloudera Manager.
- The CES HDFS Transparency cluster must be TLS enabled before you enable enabling auto-TLS on Cloudera Manager.

1. Enable TLS for HDFS Transparency.

   Cloudera Manager does not manage TLS for the IBM Storage Scale service because CDP Private Cloud Base cluster and CES HDFS cluster are loosely integrated, and are considered different clusters even if the DataNodes are colocated with the CDP Private Cloud Base nodes. Therefore, TLS needs to be manually enabled on the CES HDFS cluster. Also, verify that TLS is working on CES HDFS before proceeding to enable TLS on the Cloudera Manager.

   For more information, see “Enabling TLS for HDFS Transparency using the automation script” on page 129 and “Manually enabling TLS for HDFS Transparency” on page 131.

2. Enable Auto-TLS for Cloudera Manager.

   Enable auto-TLS on Cloudera Manager before creating CDP Private Cloud Base clusters. This enables TLS for the rest of the CDP Private Cloud Base services (for example, YARN, Zookeeper, Hive, etc). However, enabling auto-TLS does not automatically enable TLS for IBM Storage Scale. Therefore, TLS for IBM Storage Scale must first be enabled independently of Cloudera.

3. Create Cloudera Data Platform Private Cloud Base with IBM Storage Scale.

   If the CDP Private Cloud Base cluster with IBM Storage Scale is already created, proceed to the next step. Otherwise, create it now, by following the instructions in “Installing Cloudera Data Platform Private Cloud Base with IBM Storage Scale” on page 304.

4. Stop all services from IBM Cloudera Manager.

   In order to update the certificates, all the services in Cloudera and in IBM Storage Scale CES HDFS Transparency need to be stopped.

5. Update Cloudera and IBM Storage Scale truststores.

   Click on **Action** > **Deploy Client configuration** from the main cluster view in Cloudera Manager. This will propagate `ssl-client.xml` and the other configuration files to under `/etc/hadoop/conf`. The `ssl-client.xml` file will be referenced under this step below.

   Then exchange Cloudera and IBM Storage Scale certificates to each other's truststore. There are two options to do so, either by using the provided automation or by using the manual procedure.

   a. **Using Automation** -

      From IBM Storage Scale 5.1.1.1 and HDFS Transparency 3.1.1-5, you can use the `gpfs_tls_configuration.py` script to update Cloudera Manager and the CES HDFS certificates in each other's trust store so that Cloudera services can work properly with HDFS Transparency in a TLS-enabled environment. This can be done using the following automation script, or you can import the certificates manually as mentioned in the following step:

      The `gpfs_tls_configuration.py` script can be used to exchange the certificates.

      This script performs following steps:

      i) Imports the Cloudera Manager public certificates to IBM Storage Scale trust store.

      ii) Imports the IBM Storage Scale certificates to the Cloudera Manager trust store.

      Run the following command to exchange the certificates from one of the CES NameNodes:

      ```
      /usr/lpp/mmfs/hadoop/scripts/gpfs_tls_configuration.py integrate-with-cdp CDP-TRUSTSTORE-
      PASSWORD-FILE
      ```

      where, CDP-TRUSTSTORE-PASSWORD-FILE is the JSON file containing the cdp-truststore-password.

```
{
  "cdp-truststore-password": "PASSWORD"
}
```

The CDP trust store password is located in `/etc/hadoop/conf/ssl-client.xml` under the **`ssl.client.truststore.password`** parameter.

Note: After the script has run successfully, the password file will be automatically deleted for security reason.

For example:

```
  /usr/lpp/mmfs/hadoop/scripts/gpfs_tls_configuration.py integrate-with-cdp /tmp/
cdppassword.json
  [ INFO  ] Importing Cloudera Manager public certificate (cm-auto-global_cacerts.pem)
to /etc/security/serverKeys/spectrum_scale_ces_hdfs_truststore.jks
  [ INFO  ] Copying updated /etc/security/serverKeys/
spectrum_scale_ces_hdfs_truststore.jks to /var/lib/cloudera-scm-agent/agent-cert/
  [ INFO  ] Distributing /etc/security/serverKeys/spectrum_scale_ces_hdfs_truststore.jks
to all the Transparency nodes
  [ INFO  ] Adding all scale   nodes certificates to cm-auto-global_cacerts.pem
  [ INFO ] NOTE: You will need to manually distribute the following files to all the CDP
nodes :
            1. /var/lib/cloudera-scm-agent/agent-cert/cm-auto-global_cacerts.pem
            2. /var/lib/cloudera-scm-agent/agent-cert/cm-auto-global_truststore.jks
```

Distribute the following files to all the CDP nodes:

```
  1. /var/lib/cloudera-scm-agent/agent-cert/cm-auto-global_cacerts.pem
  2. /var/lib/cloudera-scm-agent/agent-cert/cm-auto-global_truststore.jks
```

b. **Using Manual procedure** -

For IBM Storage Scale 5.1.1.0 and HDFS Transparency 3.1.1-4:

Update Cloudera Manager and CES HDFS certificates in each other's trust store so that Cloudera services can work properly with HDFS Transparency in a TLS enabled environment.

Log into the CES HDFS NameNode containing the HDFS Transparency certificate (.pem) files. To find that NameNode, see step 2 in “Manually enabling TLS for HDFS Transparency” on page 131.

Log into that NameNode and run the following commands:

- Add Cloudera Manager global CA certificate to CES HDFS Truststore `spectrum_scale_ces_hdfs_truststore.jks`.

  ```
  # keytool -noprompt -importcert -alias cloudera-agents -file /var/lib/cloudera-
  scm-agent/agent-cert/cm-auto-global_cacerts.pem -keystore /etc/security/serverKeys/
  spectrum_scale_ces_hdfs_truststore.jks
  ```

  This command prompts the user to enter the CES HDFS Master Truststore password. To find the Truststore password, see step 5.c of “Manually enabling TLS for HDFS Transparency” on page 131.
- Append the HDFS Transparency certificates to Cloudera global CA certificate `agent-cert/cm-auto-global_cacerts.pem` by running the following command:

  ```
  # cat /etc/security/serverKeys/trust_stores/*.pem >> /var/lib/cloudera-scm-agent/agent-
  cert/cm-auto-global_cacerts.pem
  ```
- Import the HDFS Transparency certificates to Cloudera global Truststore `cm-auto-global_truststore.jks`.

For each `<.pem file>` in the `/etc/security/serverKeys/trust_store/` directory, run the following command:

```
keytool -noprompt -importcert -alias <FQDN Hostname Corresponding the .pem
file> -file /etc/security/serverKeys/trust_store/<name of .pem file> -keystore /var/lib/
cloudera-scm-agent/agent-cert/cm-auto-global_truststore.jks
```

For example,

If *namenode1.gpfs.net.pem* is the certificate file corresponding to the host *namenode1.gpfs.net*, then run the following command:

```
keytool -noprompt -importcert -alias namenode1.gpfs.net -file /etc/security/serverKeys/
trust_store/namenode1.gpfs.net.pem -keystore /var/lib/cloudera-scm-agent/agent-cert/cm-
auto-global_truststore.jks
```

This command prompts the user to enter the Cloudera Manager global Truststore password. To find the Truststore password, see `/etc/hadoop/conf/ssl-client.xml` on any Cloudera node.

- Run the **`keytool -importcert`** command for all the `.`pem files, until all the certificates have been imported to Cloudera global Truststore.
- Distribute the CES HDFS Truststore to all the HDFS Transparency nodes (NameNodes and DataNodes). For ease of use, you may use the following bash shell code snippet:

```
cd /etc/security/serverKeys
export HDFS_TRANSPARENCY_NODES="<space separated list of all HDFS Transparency
hostnames FQDN"
## example: export HDFS_TRANSPARENCY_NODES=”nn01.gpfs.net dn01.gpfs.net dn03.gpfs.net"
for hosts in $HDFS_TRANSPARENCY_NODES
do
  scp spectrum_scale_ces_hdfs_truststore.jks ${hosts}:/etc/security/serverKeys
done
```

- Distribute all the modified IBM Storage Scale and Cloudera Truststore files and certificates (.pem files) to all the CDP Private Cloud Base cluster nodes, including all the IBM Storage Scale nodes registered to Cloudera Manager. For ease of use, you may use the following bash shell code snippet:

```
export CMSTORES_DIR="/var/lib/cloudera-scm-agent/agent-cert/"
cd /etc/security/serverKeys
cp spectrum_scale_ces_hdfs_truststore.jks ${CMSTORES_DIR}/
cd ${CMSTORES_DIR}/
export ALL_NODES="<space separated list of all Cloudera hostnames FQDN"
## example: export ALL_NODES=”cldr1.gpfs.net cldr2.gpfs.net"
for hosts in $ALL_NODES
do

  scp cm-auto-global_truststore.jks ${hosts}:${CMSTORES_DIR}
  scp cm-auto-global_cacerts.pem  ${hosts}:${CMSTORES_DIR}
done
```

6. Update TLS configurations for IBM Storage Scale service.

   a. Go to Cloudera Manager.

   b. Click **IBM Spectrum Scale** > **Configuration** > **HDFS Client Advanced Configuration Snippet (Safety Valve) for hdfs-site.xml** and add the following custom configuration:

```
<property>
  <name>dfs.namenode.https-address.<cluster name>.nn1</name>
  <value><CES_HOSTNAME>:50470</value>
</property>
<property>
  <name>dfs.http.policy</name>
  <value>HTTPS_ONLY</value>
</property>
<property>
  <name>dfs.client.https.need-auth</name>
  <value>false</value>
</property>
```

where, *<CES_HOSTNAME>* is the FQDN Hostname corresponding to the CES IP configured for your CES HDFS cluster.

<cluster name> is the name of your CES HDFS cluster that is also your HDFS namespace.

If you want both the secure and unsecure http connections, set **dfs.http.policy** to *HTTP_AND_HTTPS*.

c. Save and deploy the client configuration. Do not start any services yet.

7. Update Metrics configurations for IBM Storage Scale service.

   a. Click **IBM Spectrum Scale** > **Configuration** and search for the following configurations and update them as follows:

```
Spectrum Scale TLS Enabled = true
HDFS Transparency DataNode HTTP Port = 1006
HDFS Transparency NameNode HTTP Port = 50470
```

   Note: These configurations are needed for HDFS metrics to appear in Cloudera Manager when TLS is enabled.

   b. Save and deploy the client configuration.

8. Start services:

   - If Ranger is enabled, start the NameNodes using the workaround mentioned in issue. Then start all the services from Cloudera Manager.
   - If Ranger is not enabled, start all the services from Cloudera Manager as usual.

Tip: Consider the following useful commands:

- To view the contents of a particular keystore, use the **keytool -list** command. For example, use the next command to view the certificates of the IBM Storage Scale Trust Store on the NameNodes:

```
# keytool -list -keystore /etc/security/serverKeys/spectrum_scale_ces_hdfs_keystore.jks
```

  You should see two entries in this keystore.

- To ensure that the `https` services are active, use the **openssl s_client -connect** command. For example:

```
openssl s_client -connect <Active NameNode hostname>:50470
openssl s_client -connect <Active NameNode hostname>:50470 -servername <Active NameNode
hostname>
openssl s_client -connect <DataNode hostname>:9869
```

  Replace the port numbers used here with the actual configured values as applicable to your environment.

Verifying TLS

This section describes the steps to verify TLS security for CDP Private Cloud Base clusters with IBM Storage Scale.

Run kinit with a valid keytab to obtain a Kerberos ticket first. For more information, see "Verifying installation" on page 309.

1. Verify the secure HDFS Java™ (swebhdfs) client provided by Cloudera to perform simple I/O operations with HDFS Transparency by running the following commands:

```
     # echo "hello world" > /tmp/hello
     # /usr/bin/hdfs dfs -ls swebhdfs://<HDFS HA Namespace>/
     # /usr/bin/hdfs dfs -put /tmp/hello swebhdfs://<HDFS HA Namespace >/tmp/
     # /usr/bin/hdfs dfs -cat swebhdfs://<HDFS HA Namespace>/tmp/hello
```

   where, *<HDFS HA Namespace>* is defined by the **fs.defaultFS** parameter in your `/etc/hadoop/conf/core-site.xml`.

2. Verify the https client by running the following command:

   ```
   # curl -ku: --negotiate https://<CES_HOSTNAME>:50470/webhdfs/v1/?op=LISTSTATUS
   ```

   where, *<CES_HOSTNAME>* is the FQDN hostname corresponding to the CES IP configured for your CES HDFS cluster.

   Note:

   - For Non-HA CES HDFS clusters, use the *<CES_HOSTNAME>:<port>* format instead of Namespace for the **`hdfs`** commands.
   - For **`curl`** commands, always use the *<CES_HOSTNAME>:<port>* format. For Kerberos enabled clusters, substituting *<CES_HOSTNAME>* with *<CES-IP>* will fail with HTTP 401 (Auth) error, as the Kerberos principal is created only for the CES hostname.

Rotating Auto-TLS certificates

Security policies of an organization might require the security administrator to rotate the Auto-TLS certificates from Cloudera Manager.

You need to complete the following steps to rotate the Auto-TLS certificates from Cloudera Manager when IBM Storage Scale is integrated:

1. Stop all the services from IBM Cloudera Manager.

   In order to update the certificates, all the services in Cloudera and IBM Storage Scale CES HDFS Transparency must be stopped.
2. Recycle the Auto-TLS certificates.

   Click **Cloudera Manager** > **Security** > **Rotate Auto-TLS certificates**.

   This action deletes the `/var/lib/cloudera-scm-agent/agent-cert` directory from all the CDP nodes and regenerates them. As a result, the new CDP truststore will no longer contain the IBM Storage Scale entries.
3. Remove the stale Cloudera certificates entries from the IBM Storage Scale truststore. Run the following commands as root on one of the CES HDFS NameNodes:

   ```
   cd /etc/security/serverKeys/
   keytool -delete -alias cm-auto-global_cacerts -keystore
   spectrum_scale_ces_hdfs_truststore.jks
   ```
4. From the same NameNode, follow the Update Cloudera and IBM Storage Scale truststores step to exchange certificates between them. As a result, following actions are completed.

   - Cloudera Manager public certificates are imported to IBM Storage Scale truststore.
   - The older IBM Storage Scale certificates are removed from Cloudera Manager truststore.
   - The new IBM Storage Scale certificates are imported to Cloudera Manager truststore.
5. If Ranger is enabled, the Ranger TLS plugin file `rangerpluginssl.jceks` needs to be re-created as well. Re-create the same by following Additional configurations when TLS is enabled.
6. Start the services:

   - If Ranger is enabled, start the NameNodes by using the workaround that is mentioned in Ranger issues with TLS enabled and then start all the services from Cloudera Manager.
   - If Ranger is not enabled, start all the services from Cloudera Manager as usual.

Configuring Apache Knox

This section describes how to configure Apache Knox on CDP Private Cloud Base clusters integrated with CES HDFS.

The Apache Knox Gateway (or simply, Apache Knox) extends the perimeter security for Hadoop. By encapsulating Kerberos, Apache Knox eliminates the need for client software or client configuration for

Kerberos; and thus simplifies the access model. For more information, see Apache Knox Gateway in Cloudera documentation.

To be able to use Apache Knox with HDFS Transparency, more configurations are needed, as described in the following steps.

1. Configure HDFS Transparency:

   Set the **hadoop.proxyuser.knox.groups** parameter to * by using the following command:

   ```
   /usr/lpp/mmfs/hadoop/sbin/mmhdfs config set core-site.xml -k hadoop.proxyuser.knox.groups=*
   ```

   To upload the configuration, issue the next command:

   ```
   /usr/lpp/mmfs/hadoop/sbin/mmhdfs
   ```

   Then, restart HDFS Transparency services.
2. Go to **Cloudera Manager** > **Knox service**, and set the **WEBHDFS:url** parameter to *https://<myceshost>:50470/webhdfs*.

   Where *myceshost* is the hostname corresponding to the CES IP configured for HDFS Transparency and *50470* is the default HTTPS port for **NameNodes**.
3. Apache Knox uses Linux PAM authentication. The Apache Knox user should have permission for the /etc/shadow file for PAM to be able to authenticate local users to Apache Knox. For more information, see An introduction to Pluggable Authentication Modules (PAM) in Linux in Red Hat documentation.

To configure the Apache Knox user, perform the following steps.

1. Make the following configurations:

   ```
   groupadd shadow
   chgrp shadow /etc/shadow
   usermod -a -G shadow knox
   chmod 600 /etc/shadow
   setfacl -m "u:knox:r--" /etc/shadow
   rm -f /var/run/nologin
   ```
2. Validate that users can access HDFS Transparency through Apache Knox by using a file listing command:

   ```
   curl -ikv -u <user> https://<knox-host>:<knox port>/gateway/cdp-proxy-api/webhdfs/v1/?
   op=LISTSTATUS
   ```

HDFS encryption

HDFS implements transparent, end-to-end encryption. After configuring HDFS, data read from and written to special HDFS directories is transparently encrypted and decrypted without requiring changes to the user application code. This encryption is also end-to-end, which means that the data can only be encrypted and decrypted by the client.

For more information, see "HDFS encryption" on page 192.

This section describes how to enable HDFS encryption for CDP Private Cloud Base clusters integrated with IBM Storage Scale.

Enabling HDFS encryption

This topic lists the steps to enable HDFS encryption for CES HDFS and CDP Private Cloud Base clusters.

Note:

- Enabling HDFS encryption for CES HDFS and CDP Private Cloud Base clusters requires that the Ranger, TLS and Kerberos are enabled. Before you proceed to enable HDFS encryption, ensure that Ranger, TLS and Kerberos are fully functional.

- Ensure Ranger policies are working properly by following the steps in "Verifying Ranger policy" on page 320. Otherwise Ranger KMS server might fail to start.
- HDFS encryption requires the `Ranger-KMS with Key Trustee Server` Cloudera service and the Key Trustee Server (KTS) service. These two services can be added only after the CDP Private Cloud Base cluster is deployed and not during the first time the cluster is created.

1. Configure Key Trustee Server (KTS) repository in Cloudera Manager.
   a. Go to **Cloudera Manager** > **Parcels** > **Parcel Repositories & Network Settings**.
   b. Add Repository URL where *KEYTRUSTEE_SERVER-X.X.X.X-X.keytrusteeX.X.X.X.pX.XXXXXXX-XXX.parcel* is present in **Remote Parcel Repository URLs**.
   c. Click **Save & Verify Configuration**.
   d. Click **Close**.
   e. On KEYTRUSTEE_SERVER, click **Download** > **Distribute** > **Activate**.
2. Add Key Trustee Server (KTS) service to CDP cluster.
   a. Go to Cloudera Manager, click **Cluster** > **Actions** > **Add Service** and install the Key Trustee Server (KTS) service. Follow the wizard and install the service.

      Note: While on the **Setup Entropy** screen, if sufficient entropy is available, you can skip the installation of rng-tools. Otherwise, if entropy drops while generating the secrets, the Cloudera Manager wizard might become unresponsive. In that case, go back to the **Setup Entropy** screen to install and configure the rng-tools package.
   b. Start Key Trustee Server from the Cloudera Manager UI.
3. Add Ranger KMS with Key Trustee Server to CDP cluster.
   a. Go to Cloudera Manager, click **Cluster** > **Actions** > **Add Service** and install the Ranger KMS with Key Trustee Server service.
   b. Follow the wizard and complete the installation.
4. Update CES HDFS configuration to enable native HDFS encryption.
   a. Stop the IBM Storage Scale service from Cloudera Manager. The NameNodes and DataNodes should be stopped.
   b. Log into a CES HDFS node and run the following commands to update the CES HDFS configuration to enable encryption:

      ```
      # mmhdfs config set gpfs-site.xml -k gpfs.encryption.enabled=true -k
      gpfs.ranger.enabled=scale
      # mmhdfs config set core-site.xml -k hadoop.security.key.provider.path=kms://
      https@<RANGER_KMS_HOST>:9494/kms
      ```

      Replace *<RANGER_KMS_HOST>* with the FQDN hostname of your Ranger KMS server.

      Note: If you had moved your Ranger KMS service from one host to another, ensure that you update the **`hadoop.security.key.provider.path`** parameter to the correct host.
   c. Upload the configuration to CCR

      ```
      # mmhdfs config upload
      ```
   d. Go to Cloudera Manager, click **Cluster** > **Actions menu**.
   e. Stop all the services.
   f. Deploy client configuration.
   g. Start all the services.

   Note:

   - If the NameNodes or Ranger KMS service fails to start because of a known issue, see Ranger workaround.

- In you see an authorization exception in Ranger KMS GUI, see While creating the encryption key you see an authorization exception in the Ranger KMS GUI.

Verifying HDFS encryption

This section describes the steps to verify HDFS encryption on CDP Private Cloud Base with IBM Storage Scale.

1. Log in to Ranger GUI as **keyadmin**.
2. In order to create keys, select **cm_kms policy** > **Edit the policy** > **add role for a regular user**.

   You may add more roles as needed. In this example, we use a *testuser* user as created in "Verifying installation" on page 309.
3. Get a Kerberos token for *testuser* and create a new encryption key.

   ```
   # kinit -kt /etc/security/keytabs/testuser.headless.keytab testuser@<Your Realm name>
   # hadoop key create mykey
   ```
4. Create an empty directory to be created as an encryption zone. Then, designate the `/tmp/myzone` directory as an encryption zone.

   For this purpose, this example uses the *hdfs* user that is a part of the Hadoop supergroup.

   ```
   # kinit -kt /etc/security/keytabs/hdfs.headless.keytab hdfs@<Your Realm Name>
   # hadoop fs -mkdir /tmp/myzone
   # hadoop fs -chown testuser:testuser /tmp/myzone
   # hdfs crypto -createZone -keyName mykey -path /tmp/myzone
   ```
5. Log in as *testuser* and verify the zone.

   For this test, use an input file (for example, `/tmp/helloWorld`). Run the following commands:

   ```
   #  kinit -kt /etc/security/keytabs/testuser.headless.keytab testuser@<Your Realm name>
   # hadoop fs -put /tmp/helloWorld /tmp/myzone/
   # hdfs crypto -getFileEncryptionInfo -path /tmp/myzone/helloWorld
   console output: {cipherSuite: {name: AES/CTR/NoPadding, algorithmBlockSize:
   16}, cryptoProtocolVersion: CryptoProtocolVersion{description='Encryption zones',
   version=1, unknownValue=null}, edek: 2010d301afbd43b58f10737ce4e93b39, iv:
   ade2293db2bab1a2e337f91361304cb3, keyName: mykey, ezKeyVersionName: mykey@0}
   ```

Rolling restart

This topic describes how to perform a rolling restart of the HDFS Transparency components in a Cloudera CDP managed environment.

Configuration updates to HDFS Transparency may be applied at runtime, without causing downtime to HDFS Transparency. This may be achieved by performing rolling restart of NameNodes and DataNodes as per the following logic:

1. Log in to an HDFS Transparency node as root and apply the configuration updates using **`mmhdfs config set`** command. However, if you are updating `hadoop-env.sh`, edit the `/var/mmfs/hadoop/etc/hadoop/hadoop-env.sh` file directly rather than using the **`mmhdfs config set`** command.
2. Upload the configuration to IBM Storage Scale CCR by running the **`mmhdfs config upload`** command.
3. Start with the rolling restarts of NameNodes. For rolling restart of NameNodes, follow the next procedure to make the NameNode state transitions quicker:

   - Log in to one of the CES HDFS NameNodes as root.
   - Find out which hosts are running the current active and standby NameNode, by running the following command:

     ```
     /usr/lpp/mmfs/hadoop/bin/hdfs haadmin -getAllServiceState
     ```

- From **Cloudera Manager** > **IBM Spectrum Scale service** > **instances**, restart the currently Standby NameNode.
- Verify that the standby NameNode is running and is in good health. This should ensure there was no error with the applied configurations.
- On the active CES HDFS NameNode host as root, manually failover the active NameNode to the standby node (which was just restarted), by running following command:

  ```
  /usr/lpp/mmfs/bin/mmces address move --ces-ip x.x.x.x --ces-node <standby_namenode_host>
  ```

- Verify and ensure that the active/standby nodes are now swapped, by running the following command as before:

  ```
  /usr/lpp/mmfs/hadoop/bin/hdfs haadmin -getAllServiceState
  ```

- From **Cloudera Manager** > **IBM Spectrum Scale service** > **instances**, restart the other NameNode (which would now be in standby).

4. Continue with the rolling restart of DataNodes by following Rolling Restart in the Cloudera Manager documentation.

Multiple IBM Storage Scale file system support

Set up a CES HDFS cluster with storage type set to `shared,shared`. The storage setup must consist of two file system.

To add support for multiple IBM Storage Scale file system, follow the steps listed below:

1. Set up the two file systems.
2. Set up CES HDFS cluster using the first mounted file system for HDFS and cesSharedRoot.
3. Verify the NameNode state and the health of CES HDFS nodes using the first mounted file system.
4. Add the second mounted file system using the following steps:

   a. Stop all CES HDFS services in the CES HDFS cluster by running the following command:

   ```
   mmhdfs hdfs stop
   ```

   b. Edit the values of the **`gpfs.storage.type`**, **`gpfs.mnt.dir`** and **`gpfs.replica.enforced`** parameters in `/var/mmfs/hadoop/etc/hadoop/gpfs-site.xml` as follows:

   ```
     <property>
       <name>gpfs.mnt.dir</name>
       <value>/fs1,/fs2</value>
     </property>
     <property>
       <name>gpfs.storage.type</name>
       <value>shared,shared</value>
     </property>
     <property>
       <name>gpfs.replica.enforced</name>
       <value>gpfs,gpfs</value>
     </property>
   ```

5. Upload the config to update the CCR by running the following command

   ```
   mmhdfs config upload
   ```

6. Start the CES HDFS services and HDFS Transparency by running the following command:

   ```
   mmhdfs hdfs start
   ```

7. Verify the state of the NameNode and DataNode by running the following commands:

   ```
   mmhdfs hdfs-nn status
   mmhdfs hdfs-dn status
   ```

Note: To list all the files/directories, run the following command:

```
hadoop dfs -ls /
```

IBM Storage Scale service management

Cloudera Manager offers a basic IBM Storage Scale service management.

You can use the IBM Storage Scale service for the following:

- Start and stop the HDFS Transparency NameNode and DataNode.
- Monitor metrics from HDFS Transparency.
- Put the HDFS Transparency NameNode and DataNodes into the maintenance mode.

You cannot use the IBM Storage Scale service for the following:

- Start and stop the IBM Storage Scale file system and daemons.
- Manage Kerberos for HDFS Transparency.

Note: These must be done independent of the Cloudera Manager.

Verify HDFS Transparency version

To check the version of HDFS Transparency, you need to run manual commands on the CES HDFS cluster node from the command line interface.

Run the following commands to check the HDFS Transparency version:

```
# rpm -qa | grep gpfs.hdfs-protocol
```

or

```
# mmdsh -N c902f10x05,c902f10x06,c902f10x07,c902f10x08 "rpm -qa | grep gpfs.hdfs-protocol"
```

Note:

- If the password-less ssh is not set up, you need to go to each node and run the manual commands.
- If the password-less ssh is set up, you can run a password-less ssh query to check the rpm that is installed.

Verify IBM Storage Scale service CSD version

To check the version of CSD, you need to run the rpm command on the Cloudera Manager server from the command line interface.

Run the following rpm command to check the CSD version:

```
# rpm -qa | grep gpfs.hdfs.cloudera.cdp.csd
```

Verifying the CDP upgrade

This topic lists the steps to verify the CDP upgrade.

1. Check that the updated CSD is loaded in Cloudera Manager by viewing the Cloudera Manager server log file. In `/var/log/cloudera-scm-server/cloudera-scm-server.log`, check for the following:

   `main:com.cloudera.csd.components.CsdRegistryImpl: Installed 1 handlers for CSD [SPECTRUMSCALE_C<CM_VERSION>-<CSD_VERSION>]`

   where, *<CM_VERSION>-<CSD_VERSION>* are the updated Cloudera Manager and CSD versions.

   For example:

For CDP stack 7.1.7 and CSD version 1.2.0 ensure that `SPECTRUMSCALE_C717-1.2.0` is loaded.

```
# grep -ir 'Installed 1 handlers for CSD \[SPECTRUM' /var/log/cloudera-scm-server/cloudera-
scm-server.log
2021-08-13 06:05:21,012 INFO main:com.cloudera.csd.components.CsdRegistryImpl: Installed 1
handlers for CSD [SPECTRUMSCALE_C716-1.2.0]
2021-08-13 06:05:21,034 INFO main:com.cloudera.csd.components.CsdRegistryImpl: Installed 1
handlers for CSD [SPECTRUMSCALE_C717-1.2.0]
```

2. Verify the HDFS Transparency version on all CES HDFS nodes by running the following command:

```
# rpm -q gpfs.hdfs-protocol
gpfs.hdfs-protocol-3.1.1-5.x86_64
```

3. Verify that the IBM Storage Scale file system (for example, “gpfs1”) is mounted correctly from one HDFS node:

```
# mmlsmount gpfs1 -L
File system gpfs1 is mounted on the following four nodes:
172.16.1.67 c902f10x01
172.16.1.69 c902f10x02
172.16.1.73 c902f10x04
172.16.1.71 c902f10x03
```

4. Verify the HDFS status from one HDFS node:

```
# /usr/lpp/mmfs/hadoop/sbin/mmhdfs hdfs status
c902f10x02.gpfs.net: scaleces: namenode pid is 2267
c902f10x01.gpfs.net: scaleces: namenode pid is 7792
c902f10x04.gpfs.net: scaleces: datanode pid is 6231
c902f10x03.gpfs.net: scaleces: datanode pid is 20365
```

5. Run mapreduce teragen / terasort job as shown in the following example:

```
# yarn jar /opt/cloudera/parcels/CDH/lib/hadoop-mapreduce/hadoop-mapreduce-examples.jar
teragen 500 test_run
# yarn jar /opt/cloudera/parcels/CDH/lib/hadoop-mapreduce/hadoop-mapreduce-examples.jar
terasort test_run test_run_sorted
```

Monitoring

For information on monitoring the IBM Storage Scale cluster, see the *Monitoring* section in the *IBM Storage Scale: Problem Determination Guide*.

Monitoring IBM Storage Scale service using Cloudera Manager

This section lists the steps to monitor the IBM Storage Scale service using Cloudera Manager.

Prerequisites:

Ensure that the **`transparency.namenode.http.port`** and **`transparency.datanode.http.port`** parameters are correctly set within the IBM Storage Scale service as described in “Installing Cloudera Data Platform Private Cloud Base with IBM Storage Scale” on page 304.

Steps

1. Go to the **Cloudera Manager GUI** > **Clusters** > **Your cluster view**.
2. Click the drop-down on the right side of page and select **Add from Chart Builder**.
3. To list all the graphs for DataNode, in the query box enter the following:

```
select * where roleType=TRANSPARENCY_DATANODE
```

4. Click **Build Chart**.
5. In Facets, select **All Separate** to see all the attributes in individual graphs.
6. You can write the same query for NameNode as follows:

```
select * where roleType=TRANSPARENCY_NAMENODE
```

For more information on the TSQuery format, see tsquery Syntax.

Following are the NameNode and DataNode graph lists with their meanings:

Attribute Name	Meaning	Regular expression matching to the JMX bean
spectrumscale_hdfs_block_checksum_op_avg_time	Block Checksum Average Time	Hadoop:service=DataNode,name=DataNodeActivity-*::BlockChecksumOpAvgTime
spectrumscale_hdfs_block_checksum_op_num_ops	Block Checksum Operations	Hadoop:service=DataNode,name=DataNodeActivity-*::BlockChecksumOpNumOps
spectrumscale_hdfs_block_reports_avg_time	Block Reports Average Time	Hadoop:service=DataNode,name=DataNodeActivity-*::BlockReportsAvgTime
spectrumscale_hdfs_block_reports_num_ops	Block Reports Operations	Hadoop:service=DataNode,name=DataNodeActivity-*::BlockReportsNumOps
spectrumscale_hdfs_block_verification_failures	Block Verification Failures	Hadoop:service=DataNode,name=DataNodeActivity-*::BlockVerificationFailures
spectrumscale_hdfs_blocks_cached	The total number of HDFS blocks cached over the lifetime of the process.	Hadoop:service=DataNode,name=DataNodeActivity-*::BlocksCached
spectrumscale_hdfs_blocks_get_local_path_info	Blocks Get Local Path Info	Hadoop:service=DataNode,name=DataNodeActivity-*::BlocksGetLocalPathInfo
spectrumscale_hdfs_blocks_read	Blocks Read	Hadoop:service=DataNode,name=DataNodeActivity-*::BlocksRead
spectrumscale_hdfs_blocks_removed	Blocks Removed	Hadoop:service=DataNode,name=DataNodeActivity-*::BlocksRemoved
spectrumscale_hdfs_blocks_replicated	Blocks Replicated	Hadoop:service=DataNode,name=DataNodeActivity-*::BlocksReplicated
spectrumscale_hdfs_blocks_uncached	The total number of HDFS blocks uncached over the lifetime of the process.	Hadoop:service=DataNode,name=DataNodeActivity-*::BlocksUncached
spectrumscale_hdfs_blocks_verified	Blocks Verified	Hadoop:service=DataNode,name=DataNodeActivity-*::BlocksVerified
spectrumscale_hdfs_blocks_written	Blocks Written	Hadoop:service=DataNode,name=DataNodeActivity-*::BlocksWritten
spectrumscale_hdfs_bytes_read	Number of bytes read	Hadoop:service=DataNode,name=DataNodeActivity-*::BytesRead
spectrumscale_hdfs_bytes_written	Bytes Written	Hadoop:service=DataNode,name=DataNodeActivity-*::BytesWritten
spectrumscale_hdfs_cache_reports_avg_time	The average time to generate cache reports on the DataNode.	Hadoop:service=DataNode,name=DataNodeActivity-*::CacheReportsAvgTime

Attribute Name	Meaning	Regular expression matching to the JMX bean
spectrumscale_hdfs_cache_reports_num_ops	The total number of generate cache reports operations on the DataNode.	Hadoop:service=DataNode,name=DataNodeActivity-*::CacheReportsNumOps
spectrumscale_hdfs_copy_block_op_avg_time	Copy Block Average Time	Hadoop:service=DataNode,name=DataNodeActivity-*::CopyBlockOpAvgTime
spectrumscale_hdfs_copy_block_op_num_ops	Copy Block Operations	Hadoop:service=DataNode,name=DataNodeActivity-*::CopyBlockOpNumOps
spectrumscale_hdfs_flush_nanos_avg_time	Average Disk Flush Time	Hadoop:service=DataNode,name=DataNodeActivity-*::FlushNanosAvgTime
spectrumscale_hdfs_flush_nanos_num_ops	Disk Flushes	Hadoop:service=DataNode,name=DataNodeActivity-*::FlushNanosNumOps
spectrumscale_hdfs_fsync_nanos_avg_time	Average Disk Fsync Time	Hadoop:service=DataNode,name=DataNodeActivity-*::FsyncNanosAvgTime
spectrumscale_hdfs_fsync_nanos_num_ops	Disk Fsyncs	Hadoop:service=DataNode,name=DataNodeActivity-*::FsyncNanosNumOps
spectrumscale_hdfs_fsync_num_ops	Fsync Operations	Hadoop:service=DataNode,name=DataNodeActivity-*::FsyncCount
spectrumscale_hdfs_heartbeats_avg_time	Heartbeat Average Time	Hadoop:service=DataNode,name=DataNodeActivity-*::HeartbeatsAvgTime
spectrumscale_hdfs_heartbeats_num_ops	Heartbeats	Hadoop:service=DataNode,name=DataNodeActivity-*::HeartbeatsNumOps
spectrumscale_hdfs_send_data_packet_blocked_on_network_nanos_avg_time	Send Data Packet Blocked On Network Average Time	Hadoop:service=DataNode,name=DataNodeActivity-*::SendDataPacketBlockedOnNetworkNanosAvgTime
spectrumscale_hdfs_send_data_packet_blocked_on_network_nanos_num_ops	Send Data Packet Blocked On Network Operations	Hadoop:service=DataNode,name=DataNodeActivity-*::SendDataPacketBlockedOnNetworkNanosNumOps
spectrumscale_hdfs_send_data_packet_transfer_nanos_avg_time	Send Data Packet Transfer Average Time	Hadoop:service=DataNode,name=DataNodeActivity-*::SendDataPacketTransferNanosAvgTime
spectrumscale_hdfs_send_data_packet_transfer_nanos_num_ops	Send Data Packet Transfer Operations	Hadoop:service=DataNode,name=DataNodeActivity-*::SendDataPacketTransferNanosNumOps

Attribute Name	Meaning	Regular expression matching to the JMX bean
spectrumscale_hdfs_write_block_op_avg_time	Write Block Average Time	Hadoop:service=DataNode,name=DataNodeActivity-*::WriteBlockOpAvgTime
spectrumscale_hdfs_write_block_op_num_ops	Write Block Operations	Hadoop:service=DataNode,name=DataNodeActivity-*::WriteBlockOpNumOps
spectrumscale_hdfs_writes_from_local_client	Writes From Local Clients	Hadoop:service=DataNode,name=DataNodeActivity-*::WritesFromLocalClient
spectrumscale_hdfs_writes_from_remote_client	Writes From Remote Clients	Hadoop:service=DataNode,name=DataNodeActivity-*::WritesFromRemoteClient
spectrumscale_hdfs_packet_ack_round_trip_time_nanos_avg_time	Packet Ack Round Trip Average Time	Hadoop:service=DataNode,name=DataNodeActivity-*::PacketAckRoundTripTimeNanosAvgTime
spectrumscale_hdfs_packet_ack_round_trip_time_nanos_num_ops	Packet Ack Round Trip Operations	Hadoop:service=DataNode,name=DataNodeActivity-*::PacketAckRoundTripTimeNanosNumOps
spectrumscale_hdfs_read_block_op_avg_time	Read Block Average Time	Hadoop:service=DataNode,name=DataNodeActivity-*::ReadBlockOpAvgTime
spectrumscale_hdfs_read_block_op_num_ops	Read Block Operations	Hadoop:service=DataNode,name=DataNodeActivity-*::ReadBlockOpNumOps
spectrumscale_hdfs_reads_from_local_client	Reads From Local Clients	Hadoop:service=DataNode,name=DataNodeActivity-*::ReadsFromLocalClient
spectrumscale_hdfs_reads_from_remote_client	Reads From Remote Clients	Hadoop:service=DataNode,name=DataNodeActivity-*::ReadsFromRemoteClient
spectrumscale_hdfs_replace_block_op_avg_time	Replace Block Operation Average Time	Hadoop:service=DataNode,name=DataNodeActivity-*::ReplaceBlockOpAvgTime
spectrumscale_hdfs_replace_block_op_num_ops	Replace Block Operations	Hadoop:service=DataNode,name=DataNodeActivity-*::ReplaceBlockOpNumOps
spectrumscale_hdfs_jvm_blocked_threads	Blocked threads	Hadoop:service=DataNode,name=JvmMetrics::ThreadsBlocked
spectrumscale_hdfs_jvm_gc_count	Number of garbage collections	Hadoop:service=DataNode,name=JvmMetrics::GcCount
spectrumscale_hdfs_jvm_gc_time_ms	Total time spent garbage collecting.	Hadoop:service=DataNode,name=JvmMetrics::GcTimeMillis
spectrumscale_hdfs_jvm_heap_committed_mb	Total amount of committed heap memory.	Hadoop:service=DataNode,name=JvmMetrics::MemHeapCommittedM

Attribute Name	Meaning	Regular expression matching to the JMX bean
spectrumscale_hdfs_jvm_heap_used_mb	Total amount of used heap memory.	Hadoop:service=DataNode,name=JvmMetrics::MemHeapUsedM
spectrumscale_hdfs_jvm_max_memory_mb	Maximum allowed memory.	Hadoop:service=DataNode,name=JvmMetrics::MemMaxM
spectrumscale_hdfs_jvm_new_threads	New threads	Hadoop:service=DataNode,name=JvmMetrics::ThreadsNew
spectrumscale_hdfs_jvm_non_heap_committed_mb	Total amount of committed non-heap memory.	Hadoop:service=DataNode,name=JvmMetrics::MemNonHeapCommittedM
spectrumscale_hdfs_jvm_non_heap_used_mb	Total amount of used non-heap memory.	Hadoop:service=DataNode,name=JvmMetrics::MemNonHeapUsedM
spectrumscale_hdfs_jvm_pause_time	The amount of extra time the jvm was paused above the requested sleep time. The JVM pause monitor sleeps for 500 milliseconds and any extra time it waited above this is counted in the pause time.	Hadoop:service=DataNode,name=JvmMetrics::GcTotalExtraSleepTime
spectrumscale_hdfs_jvm_pauses_info_threshold_count	Number of JVM pauses longer than the info threshold but shorter than the warning threshold. By default the info threshold is set to 1 second. To change use this configuration key JvmPauseMonitorService.info-threshold.ms	Hadoop:service=DataNode,name=JvmMetrics::GcNumInfoThresholdExceeded
spectrumscale_hdfs_jvm_pauses_warn_threshold_count	Number of JVM pauses longer than the warning threshold. By default the warning threshold is set to 10 second. To change use this configuration key JvmPauseMonitorService.warn-threshold.ms	Hadoop:service=DataNode,name=JvmMetrics::GcNumWarnThresholdExceeded
spectrumscale_hdfs_jvm_runnable_threads	Runnable threads	Hadoop:service=DataNode,name=JvmMetrics::ThreadsRunnable
spectrumscale_hdfs_jvm_terminated_threads	Terminated threads	Hadoop:service=DataNode,name=JvmMetrics::ThreadsTerminated
spectrumscale_hdfs_jvm_timed_waiting_threads	Timed waiting threads	Hadoop:service=DataNode,name=JvmMetrics::ThreadsTimedWaiting
spectrumscale_hdfs_jvm_waiting_threads	Waiting threads	Hadoop:service=DataNode,name=JvmMetrics::ThreadsWaiting
spectrumscale_hdfs_log_error	Logged Errors	Hadoop:service=DataNode,name=JvmMetrics::LogError
spectrumscale_hdfs_log_fatal	Logged Fatals	Hadoop:service=DataNode,name=JvmMetrics::LogFatal
spectrumscale_hdfs_log_info	Logged Infos	Hadoop:service=DataNode,name=JvmMetrics::LogInfo

Attribute Name	Meaning	Regular expression matching to the JMX bean
spectrumscale_hdfs_log_warn	Logged Warnings	Hadoop:service=DataNode,name=JvmMetrics::LogWarn
spectrumscale_hdfs_login_failure_avg_time	Average Failed Login Time	Hadoop:service=DataNode,name=UgiMetrics::LoginFailureAvgTime
spectrumscale_hdfs_login_failure_num_ops	Login Failures	Hadoop:service=DataNode,name=UgiMetrics::LoginFailureNumOps
spectrumscale_hdfs_login_success_avg_time	Average Successful Login Time	Hadoop:service=DataNode,name=UgiMetrics::LoginSuccessAvgTime
spectrumscale_hdfs_login_success_num_ops	Login Successes	Hadoop:service=DataNode,name=UgiMetrics::LoginSuccessNumOps
spectrumscale_hdfs_metrics_dropped_pub_all	Dropped Metrics Updates By All Sinks	Hadoop:service=DataNode,name=MetricsSystem,sub=Stats::DroppedPubAll
spectrumscale_hdfs_metrics_num_active_sinks	Active Metrics Sinks Count	Hadoop:service=DataNode,name=MetricsSystem,sub=Stats::NumActiveSinks
spectrumscale_hdfs_metrics_num_active_sources	Active Metrics Sources Count	Hadoop:service=DataNode,name=MetricsSystem,sub=Stats::NumActiveSources
spectrumscale_hdfs_metrics_num_all_sinks	All Metrics Sinks Count	Hadoop:service=DataNode,name=MetricsSystem,sub=Stats::NumAllSinks
spectrumscale_hdfs_metrics_num_all_sources	All Metrics Sources Count	Hadoop:service=DataNode,name=MetricsSystem,sub=Stats::NumAllSources
spectrumscale_hdfs_metrics_publish_avg_time	Metrics Publish Average Time	Hadoop:service=DataNode,name=MetricsSystem,sub=Stats::PublishAvgTime
spectrumscale_hdfs_metrics_publish_num_ops	Metrics Publish Operations	Hadoop:service=DataNode,name=MetricsSystem,sub=Stats::PublishNumOps
spectrumscale_hdfs_metrics_snapshot_avg_time	Metrics Snapshot Average Time	Hadoop:service=DataNode,name=MetricsSystem,sub=Stats::SnapshotAvgTime
spectrumscale_hdfs_metrics_snapshot_num_ops	Metrics Snapshot Average Operations	Hadoop:service=DataNode,name=MetricsSystem,sub=Stats::SnapshotNumOps
spectrumscale_hdfs_rpc_authentication_failures	RPC Authentication Failures	Hadoop:service=DataNode,name=RpcActivityForPort\\d+::RpcAuthenticationFailures
spectrumscale_hdfs_rpc_authentication_successes	RPC Authentication Successes	Hadoop:service=DataNode,name=RpcActivityForPort\\d+::RpcAuthenticationSuccesses

Attribute Name	Meaning	Regular expression matching to the JMX bean
spectrumscale_hdfs_rpc_authorization_failures	RPC Authorization Failures	Hadoop:service=DataNode,name=RpcActivityForPort\\d+::RpcAuthorizationFailures
spectrumscale_hdfs_rpc_authorization_successes	RPC Authorization Successes	Hadoop:service=DataNode,name=RpcActivityForPort\\d+::RpcAuthorizationSuccesses
spectrumscale_hdfs_rpc_call_queue_length	RPC Call Queue Length	Hadoop:service=DataNode,name=RpcActivityForPort\\d+::CallQueueLength
spectrumscale_hdfs_rpc_num_open_connections	Open RPC Connections	Hadoop:service=DataNode,name=RpcActivityForPort\\d+::NumOpenConnections
spectrumscale_hdfs_rpc_processing_time_avg_time	Average RPC Processing Time	Hadoop:service=DataNode,name=RpcActivityForPort\\d+::RpcProcessingTimeAvgTime
spectrumscale_hdfs_rpc_processing_time_num_ops	RPCs Processed	Hadoop:service=DataNode,name=RpcActivityForPort\\d+::RpcProcessingTimeNumOps
spectrumscale_hdfs_rpc_queue_time_avg_time	Average RPC Queue Time	Hadoop:service=DataNode,name=RpcActivityForPort\\d+::RpcQueueTimeAvgTime
spectrumscale_hdfs_rpc_queue_time_num_ops	RPCs Queued	Hadoop:service=DataNode,name=RpcActivityForPort\\d+::RpcQueueTimeNumOps
spectrumscale_hdfs_rpc_received_bytes	RPC Received Bytes	Hadoop:service=DataNode,name=RpcActivityForPort\\d+::ReceivedBytes
spectrumscale_hdfs_rpc_sent_bytes	RPC Sent Bytes	Hadoop:service=DataNode,name=RpcActivityForPort\\d+::SentBytes
spectrumscale_hdfs_xceivers	Transceivers	Hadoop:service=DataNode,name=DataNodeInfo::XceiverCount
spectrumscale_hdfs_connections	Current number of connections to NameNode	Hadoop:service=NameNode,name=FSNamesystem::TotalLoad
spectrumscale_hdfs_fsnamesystem_lockqueuelength	Number of threads waiting to acquire FSNameSystem lock	Hadoop:service=NameNode,name=FSNamesystem::LockQueueLength
spectrumscale_hdfs_active_connection_holdinglease	Number of active clients holding lease	Hadoop:service=NameNode,name=FSNamesystem::NumActiveClients
spectrumscale_hdfs_state	Current state of the file system: Safemode or Operational	Hadoop:service=NameNode,name=FSNamesystem::FSState
spectrumscale_hdfs_ha_state	Current state of the NameNode: initializing or active or standby or stopping state	Hadoop:service=NameNode,name=FSNamesystem::tag.HAState
spectrumscale_hdfs_rpc_queue_time_num_ops	RPCs Queued	Hadoop:service=NameNode,name=RpcActivityForPort\\d+::RpcQueueTimeNumOps

Attribute Name	Meaning	Regular expression matching to the JMX bean
spectrumscale_hdfs_rpc_queue_time_avg_time	Average RPC Queue Time	Hadoop:service=NameNode,name=RpcActivityForPort\d+::RpcQueueTimeAvgTime
spectrumscale_hdfs_rpc_processing_time_num_ops	RPCs Processed	Hadoop:service=NameNode,name=RpcActivityForPort\d+::RpcProcessingTimeNumOps
spectrumscale_hdfs_rpc_processing_time_avg_time	Average RPC Processing Time	Hadoop:service=NameNode,name=RpcActivityForPort\d+::RpcProcessingTimeAvgTime
spectrumscale_hdfs_rpc_call_queue_length	RPC Call Queue Length	Hadoop:service=NameNode,name=RpcActivityForPort\d+::CallQueueLength
spectrumscale_hdfs_rpc_num_open_connections	Open RPC Connections	Hadoop:service=NameNode,name=RpcActivityForPort\d+::NumOpenConnections

IBM Storage Scale management GUI

The IBM Storage Scale management GUI provides an easy way to configure and manage various features that are available with the IBM Storage Scale system.

For information on the IBM Storage Scale management GUI, see *Introduction to IBM Storage Scale GUI* in *IBM Storage Scale: Concepts, Planning, and Installation Guide*.

Monitoring CES HDFS cluster

The **mmhealth** command displays the results of the background monitoring for the health of a node and the services that are hosted on the node. You can use the **mmhealth** command to view the health status of a whole cluster in a single view.

For information on monitoring the CES HDFS cluster, see *Monitoring system health by using the mmhealth command* in *IBM Storage Scale: Problem Determination Guide*.

Monitoring NameNodes

Presently, Cloudera Manager does not have the ability to display Active/Standby status of the HDFS Transparency NameNodes.

To check the status of NameNodes, see *step 2* in "Verifying installation" on page 309.

Monitoring

This topic describes how to monitor HDFS Transparency in a Cloudera managed environment.

The IBM Storage Scale service subscribes to the standard Cloudera monitoring infrastructure. For details on Cloudera monitoring, seeAccessing the Cloudera Manager Admin Console.

The time series data from HDFS Transparency on the Cloudera Manager GUI may be leveraged to monitor usage, performance, and other aspects. To view time series data, click **Cloudera Manager home page** > **Charts** > **Chart Builder**. In the Chart Builder query box, type the following to list all the graphs for HDFS Transparency NameNode.

```
select * where roleType=TRANSPARENCY_NAMENODE
```

To view all the attributes in individual graphs, click **All Separate** under **Facets**.

The query can be written for DataNode by replacing the **`roleType`** to *TRANSPARENCY_DATANODE*.

For more details on Cloudera TSQuery format and syntax, see tsquery Language.

You might query for individual parameters as well. For example, to find out the Active and Standby NameNodes, run the following query:

```
select spectrumscale_hdfs_ha_state where roleType=TRANSPARENCY_NAMENODE
```

In the resulting graph, Value of *2* indicates `Standby` and Value *3* indicates `Active State`.

Similarly, run the following tsquery to see the status of the filesystem·

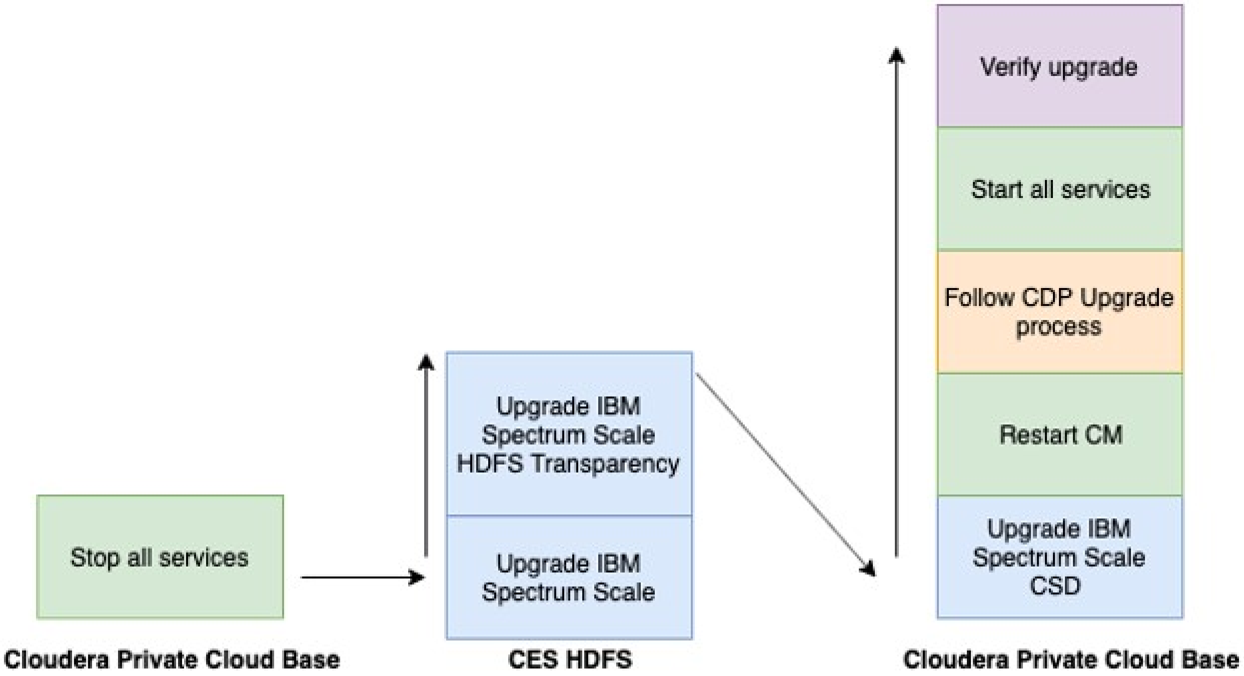
```
select spectrumscale_hdfs_state
```

Upgrading

Upgrading CDP

This section lists the steps to upgrade CDP Private Cloud Base cluster integrated with IBM Storage Scale from existing CDP Private Cloud Base version to a newer CDP Private Cloud Base version.

Note: To ensure that the correct software stack compatibility versions are adhered to, see the CDP Private Cloud Base support matrix.


Stop all services
Cloudera Private Cloud Base
Upgrade IBM Spectrum Scale HDFS Transparency
Upgrade IBM Spectrum Scale
CES HDFS
Verify upgrade
Start all services
Follow CDP Upgrade process
Restart CM
Upgrade IBM Spectrum Scale CSD
Cloudera Private Cloud Base

Offline upgrade procedure

1. Stop all the CDP cluster services from Cloudera Manager. This will stop CDP Hadoop services and CES HDFS Transparency services.
2. Upgrade the IBM Storage Scale software.

   You can upgrade either manually or by using the installation toolkit.

   - For installation toolkit upgrade process, see Performing online upgrade by using the installation toolkit.

Note: The installation toolkit automatically updates only the HDFS package when the HDFS protocol is enabled. To check if HDFS protocol is enabled, run the **./spectrumscale node list** command.

- For manual upgrade process, see the following sections:
  - Upgrading IBM Storage Scale non-protocol Linux nodes
  - Upgrading IBM Storage Scale protocol nodes.
  - Upgrade the HDFS Transparency package on each node by running the following command:

    ```
    rpm -Uvh gpfs.hdfs-protocol-3.1.1-<version>.<os>.rpm
    ```

    Note: The IBM Storage Scale extract package will contain the new IBM Storage Scale CSD package. If you are using the installation toolkit, it will be on the installer node.

3. Upgrade the IBM Storage Scale CSD by running the following steps:

   a. Log in to the Cloudera Manager node.

   b. Copy the latest IBM Storage Scale CSD package from the IBM Storage Scale Installer node to a local directory. For example, `/root/latest_csd/`.

   ```
   # scp <Spectrum Scale Installer node>:/usr/lpp/mmfs/<Spectrum Scale version>/
   hdfs_rpms/rhel/hdfs_3.1.1.x/gpfs.hdfs.cloudera.cdp.csd-<Latest CSD version>.noarch.rpm /
   root/latest_csd/
   ```

   For example:

   ```
   # scp <Spectrum Scale Installer node>:/usr/lpp/mmfs/5.1.2.0/hdfs_rpms/rhel/hdfs_3.1.1.x/
   gpfs.hdfs.cloudera.cdp.csd-1.2.0-0.noarch.rpm /root/latest_csd/
   ```

   c. Upgrade the IBM Storage Scale CSD package by running the following command:

   ```
   # cd /root/latest_csd
   # rpm -Uvh gpfs.hdfs.cloudera.cdp.csd-<Latest CSD version>.noarch.rpm For example:# rpm
   -Uvh gpfs.hdfs.cloudera.cdp.csd-1.2.0-0.noarch.rpm
   ```

   d. Restart the Cloudera Manager. The restart picks up the new IBM Storage Scale CSD for Cloudera Manager.

4. Upgrade the CDP stack by following the steps mentioned in Upgrading CDP Private Cloud Base to a higher version.
5. From Cloudera Manager, start all the services. This will start the CDP Hadoop services and the CES HDFS Transparency NameNodes and DataNodes.
6. Verify the upgrade. To verify, see "Verifying the CDP upgrade" on page 332.

Upgrading IBM Storage Scale

This section lists the steps for upgrading IBM Storage Scale.

The upgrade process is based on the CDP Private Cloud Base support matrix to ensure that the correct software stack compatibility versions are adhered to.

For example, CDP 7.1.7 supports IBM Storage Scale 5.1.1.2 and 5.1.1.3. Both these IBM Storage Scale releases contain the same IBM Storage Scale CSD and HDFS Transparency versions. If you want to upgrade from IBM Storage Scale 5.1.1.2 to 5.1.1.3 while staying at CDP 7.1.7, you only need to upgrade the IBM Storage Scale version on the CES HDFS Transparency IBM Storage Scale client nodes.

IBM Storage Scale upgrade process

You can upgrade IBM Storage Scale either manually or by using the installation toolkit.

Note: Starting with HDFS Transparency 3.1.1-15 and HDFS Transparency 3.2.2-6, dependent JAR files need to be provided. This is also required as a prerequisite for an upgrade. For more information, see the instructions to provide dependent JAR files.

Upgrading IBM Storage Scale by using the Installation toolkit

For upgrading IBM Storage Scale by using the installation toolkit, see "Installation toolkit upgrade process for HDFS Transparency" on page 47. This will upgrade all the IBM Storage Scale nodes in the IBM Storage Scale cluster.

Note: The installation toolkit automatically updates only the HDFS package when the HDFS protocol is enabled. To check if HDFS protocol is enabled, run the **`./spectrumscale node list`** command.

Upgrading IBM Storage Scale manually

1. For upgrading IBM Storage Scale manually, see the following:
   - Upgrading IBM Storage Scale non-protocol Linux nodes
   - Upgrading IBM Storage Scale protocol nodes
2. Upgrade the HDFS Transparency package on each node by running the **`rpm -Uvh gpfs.hdfs-protocol-3.1.1-<version>.<os>.rpm`** command or by following "Manual rolling upgrade for HDFS Transparency" on page 51.

Limitations

This topic lists the limitations for CDP Private Cloud Base with IBM Storage Scale.

- The TLS certificates that are created by using the automation script `/usr/lpp/mmfs/hadoop/scripts/gpfs_tls_configuration.py` have only 90 days validity and would expire thereafter. This will be fixed in a future release of HDFS Transparency. For an interim fix, contact IBM support.
- Open JDK is only supported. Oracle JDK is not supported.
- TLS/SSL is supported from CDP Private Cloud Base 7.1.6.
- IPV6 is not supported.
- Short-circuit read and Short-circuit write are not supported.
- FPO is not supported.
- HDFS encryption is supported from CDP Private Cloud Base 7.1.6.
- Upgrading from HDP to CDP Private Cloud Base is not supported.
- Kudu and Ozone are not supported.
- For production, it is recommended to have a minimum of 2 NameNodes (HA) and 3 DataNodes for the CES HDFS cluster setup.
- For Hive Warehouse Connector to work in Spark client mode, the **`spark.driver.log.persistToDfs.enabled`** parameter must be set to *false*. Therefore, the logs are written to the local storage and not IBM Storage Scale.
- The installation toolkit cannot be used if the CES HDFS cluster is kerberized. Instead, use manual installation.
- Ubuntu and SLES are not supported.
- Do not use Java 1.8.0_242 and later when Kerberos ticket_lifetime and/or renew_lifetime is set. Using a higher version results in a failure for HDFS Transparency to start.
- Hadoop services and CES HDFS cannot be colocated on the same node as the ECE node.
- The NameNode cannot be colocated with the DataNode or with any other Hadoop services.
- Starting from CDP 7.1.8, Impala is supported only on the x86 platform. Impala is not supported on IBM Power.

For more information, see "Limitations and differences from native HDFS" on page 234.

Problem determination

This topic contains information on troubleshooting the CES HDFS and CDP Private Cloud Base issues.

For CES HDFS known problem determination, see "Troubleshooting" on page 212.

For information on troubleshooting the CDP Private Cloud Base issues, see the following workarounds:

1. If Kerberos is enabled, sometimes Zookeeper or Yarn might not start successfully due to issues with the keytab generation.

   Solution:

   Zookeeper

   a. In the Cloudera Manager, click **Zookeeper** > **Configuration** and search for `kerberos`.

   b. Enable the check boxes for `Enable Kerberos Authentication` and `Enable Server to Server SASL Authentication`.

   Yarn

   a. In the Cloudera Manager, click **Yarn** > **Configuration** and search for `kerberos`.

   b. Enable the check boxes for `Enable Kerberos Authentication for HTTP Web-Consoles`.

2. Impersonation error with Hive/Oozie/Livy even when the proxyuser settings are set already within the IBM Storage Scale service in the Cloudera Manager.

   For example, the following error is seen:

   ```
   org.apache.hadoop.ipc.RemoteException(org.apache.hadoop.security.authorize.AuthorizationExce
   ption):
   User: <component>/<host>@<realm> is not allowed to impersonate <user>
   ```

   Solution:

   a. Stop the IBM Storage Scale service in Cloudera Manager.

   b. Enable proxyuser settings for HDFS Transparency as mentioned in "Enable and Configure CES HDFS" on page 42.

   c. Restart the IBM Storage Scale service from Cloudera Manager.

   d. In the Cloudera Manager Hive service, for Hiveserver2 to start and function normally, set **`hive.metastore.event.db.notification.api.auth`** to *false* on `hive-site.xml`.

3. Solr does not start after adding Ranger.

   Solution:

   It is recommended to add Solr and Ranger services together with the IBM Storage Scale service at the time of initial CDP Private Cloud Base cluster creation. However, if Ranger and Solr were added later, following workaround is needed for Solr to start properly.

   a. Log in to the Cloudera Manager console.

   b. While adding the Solr service, set the **`ZooKeeper ZNode`** parameter to *solr-infra* in the configuration wizard or if you had already added Solr but Solr does not come up properly, click **Solr** > **Configuration** and search for ZNode and set the value of the Solr configuration **`ZooKeeper ZNode`** parameter to *solr-infra*.

   c. Ensure that Kerberos checkbox is enabled in the Solr configuration.

   d. Continue to add the Solr and Ranger services. Skip if you have already added the services.

   e. After adding Ranger, the Solr service changes its name to CDP-INFRA-SOLR.

   f. Restart Solr and Ranger.

   g. Ensure that Solr and Ranger have started successfully. Do not proceed unless Solr and Ranger appear healthy.

4. Ranger issues with TLS enabled.

You may encounter one of the following issues if Ranger is enabled together with TLS security:

a. NameNodes do not start after updating the configurations and throw the following error:

```
org.apache.hadoop.hdfs.server.namenode.NameNode: Failed to start namenode.
java.lang.IllegalArgumentException: bound must be positive
```

b. NameNodes can start but Ranger policies do not work. NameNode log shows the following error message:

```
org.apache.ranger.admin.client.RangerAdminRESTClient: Failed to get response,
Error is : TrustManager is not specified2022-08-23 13:16:37,809 ERROR
org.apache.ranger.admin.client.RangerAdminRESTClient: Error getting Roles; Received NULL
….
Error getting policies; Received NULL response!!. secureMode=true, ....
```

Root cause of the issue:

At the time of starting the IBM Storage Scale service from Cloudera Manager, there are three additional configuration files (ranger-hdfs-security.xml, ranger-hdfs-policymgr-ssl.xml and ranger-hdfs-audit.xml) generated for HDFS Transparency. In certain scenarios, these three files do not get propagated to IBM Storage Scale CCR, causing the above problems.

Solution:

Start the NameNodes as following:

a. To obtain the latest `spectrumscale-TRANSPARENCY_NAMENODE` directory under `/run/cloudera-scm-agent/process/` on the NameNode, start the IBM Storage Scale service from Cloudera Manager.

b. Stop the IBM Storage Scale service from Cloudera Manager.

c. Log into any one of the HDFS Transparency NameNode hosts.

d. Run the following commands to find the current *<NameNode configuration directory>*:

```
# cd /run/cloudera-scm-agent/process/
# ls -lrt| grep spectrumscale-TRANSPARENCY_NAMENODE | tail -n 1
```

e. Run the following commands to upload the Ranger configuration files from the *<NameNode configuration directory>*, as found in the above step, to IBM Storage Scale CCR:

```
# cd <NameNode configuration directory>
# mmhdfs config import --nocheck . ranger-hdfs-security.xml,ranger-hdfs-policymgr-
ssl.xml,ranger-hdfs-audit.xml
# mmhdfs config upload
```

f. Start the IBM Storage Scale service from Cloudera Manager.

5. In the IBM Storage Scale service, metrics show NO DATA after enabling Kerberos.

   After enabling Kerberos, the HTTP port value for DataNode changes to less than *1024*. Therefore, metrics starts showing NO DATA.

   Solution:

   a. Go to Cloudera Manager GUI, click **IBM Spectrum Scale** > **Configuration** and type HTTP `Port` in filter.

   b. Set **`transparency.datanode.http.port`** to *1006*.

6. If solr is pre-installed, solr znode changes to /solr-infra when you add Ranger or Atlas on a cluster with Ranger.

   Solution:

   a. Rename znode back to /solr.

   b. Renaming the znode causes an Atlas initialization issue. To address this issue, restart Atlas on correct znode as follows:

      i) Stop Atlas.
      ii) Go to Atlas Service Actions, click **Initialize Atlas** > **Start Atlas**.

7. Installing Ranger service may fail with the following SQL error from MySQL/MariaDB:

   ```
   SQLException : SQL state: HY000 java.sql.SQLException: This function has none of
   DETERMINISTIC,
   NO SQL, or READS SQL DATA in its declaration and binary logging is enabled (you *might*
   want
   to use the less safe log_bin_trust_function_creators variable) ErrorCode: 1418
   ```

   Solution:

   a. Before creating the database for Ranger, run the following command on the SQL prompt. The user account running the command must have MySQL or MariaDB administrator privilege:

      ```
      SET GLOBAL log_bin_trust_function_creators = 1;
      ```

   b. After the Ranger installation completes, run the following command to roll back the above value to its default setting of *0*:

      ```
      SET GLOBAL log_bin_trust_function_creators = 0;
      ```

      Note: The above operations might make the MySQL or MariaDB database less secure and less robust during this duration. Other options that you can try are as follows:

      - Use a commercial database for Ranger backend storage rather than an open source option such as MySQL or MariaDB.
      - Use a separate MySQL or MariaDB instance exclusively for Ranger.

8. Hive service shows a health check issue in Hive Metastore Canary.

   Hive metastore canary fails with following error:

   ```
   2020-11-02 17:44:53,054 WARN com.cloudera.cmon.firehose.polling.CreateDirectoryTask:
   Exception while creating directory '/user/hue', for 'hue:hue', with permission: 775
   org.apache.hadoop.security.AccessControlException: Permission denied: user=hue,
   access=WRITE, inode="/user":hdfs:supergroup:drwxr-xr-x
   ```

   Solution:

   a. Add hue to the Hadoop supergroup.
   b. On all the HDFS Transparency nodes, run the following command:

      ```
      usermod -G supergroup hue
      ```

      Setting hue with the supergroup permission is now able to create the `/user/hue` directory which had failed during the Hive service Canary health check in the Hive Metastore.

   c. After the health check is resolved, remove the hue user from the Hadoop supergroup.

9. Webhdfs does not work with the CES HDFS IP/hostname. For example, the **hdfs dfs -ls webhdfs://<hdfs namespace>/** command throws an authentication error.

   Solution:

   a. Stop the IBM Storage Scale service using Cloudera Manager.
   b. Create an additional NameNode HTTP principal with the CES HDFS IP/hostname.

      For example:

      ```
      kadmin.local -q "addprinc -randkey HTTP/<myceshost>@IBM.COM
      ```

   c. To update the `spnego.service.keytab` keytab files, see "Setting up Kerberos for HDFS Transparency nodes" on page 109. Update the `spnego.service.keytab` files for each NameNode.

      i) Backup `/etc/security/keytabs/spnego.service.keytab`.

ii) Update the `spnego.service.keytab` keytab file by importing the above HTTP principal.

For example:

```
kadmin.local ktadd -k /etc/security/keytabs/spnego.service.keytab HTTP/
myceshdfs.gpfs.net@IBM.COM
```

iii) If you have used the supplied Kerberos script with HDFS Transparency v3.1.1-3, the NameNode host principal might be missing in the `spnego.service.keytab` file. Therefore, import the `spnego.service.keytab` file.

For example:

```
kadmin.local ktadd -k /etc/security/keytabs/spnego.service.keytab host/
nn01.gpfs.net@IBM.COM
```

iv) Move the `spnego.service.keytab` file to the corresponding NameNode host.

d. Set the **`dfs.web.authentication.kerberos.principal`** parameter to *:

```
<property>
  <name>dfs.web.authentication.kerberos.principal</name>
  <value>*</value>
</property>
```

e. Ensure that the CES HDFS hostname resolves from DNS and not just from an entry in the `/etc/hosts` file.

f. Start the IBM Storage Scale service using Cloudera Manager

10. Creating a Livy interactive session using REST API as follows might fail or hang:

```
curl -u : --negotiate -X POST --data '{"kind" : "spark"}' -H "Content-Type: application/
json" <livy host>:8998/sessions
```

This is particularly observed in the IBM Power platform.

Solution:

a. Disable the Livy recovery mode within Livy service by setting **`livy.server.recovery.mode`** to *off*.

b. Recreate the session.

11. DataNode colocation

The Zeppelin service fails to start when the Zeppelin server is colocated with the HDFS Transparency DataNode. The CM agent creates the `/var/lib/zeppelin` directory with *root*:*root* permissions.

Solution:

a. Change the permission of the directory to *zeppelin*:*zeppelin*.

b. Restart Zeppelin.

12. An error is seen when you try to create a Livy interactive session using REST API.

When you are trying to execute Livy REST API (for example: `curl -u : --negotiate -X POST --data '{"kind" : "spark"}' -H "Content-Type: application/json" <livy host>:8998/sessions`), the following error occurs:

```
org.apache.hadoop.ipc.RemoteException(java.lang.ArithmeticException): / by zero
org.apache.hadoop.hdfs.server.namenode.GPFSNamesystemV0.getAdditionalBlock(GPFSNamesystemV0.
java:711)
org.apache.hadoop.hdfs.server.namenode.NameNodeRpcServer.addBlock(NameNodeRpcServer.java:864
)
org.apache.hadoop.hdfs.protocolPB.ClientNamenodeProtocolServerSideTranslatorPB.addBlock(Clie
ntNamenodeProtocolServerSideTranslatorPB.java:549)
org.apache.hadoop.hdfs.protocol.proto.ClientNamenodeProtocolProtos$ClientNamenodeProtocol$2.
callBlockingMethod(ClientNamenodeProtocolProtos.java)
org.apache.hadoop.ipc.ProtobufRpcEngine$Server$ProtoBufRpcInvoker.call(ProtobufRpcEngine.jav
a:523) org.apache.hadoop.ipc.RPC$Server.call(RPC.java:991)
org.apache.hadoop.ipc.Server$RpcCall.run(Server.java:872)
org.apache.hadoop.ipc.Server$RpcCall.run(Server.java:818)
java.security.AccessController.doPrivileged(Native Method)
javax.security.auth.Subject.doAs(Subject.java:422)
org.apache.hadoop.security.UserGroupInformation.doAs(UserGroupInformation.java:1729)
org.apache.hadoop.ipc.Server$Handler.run(Server.java:2678)
```

Solution:

a. Ensure that the **dfs.blocksize** parameter in the Cloudera Manager GUI in the IBM Storage Scale service matches the **dfs.blocksize** parameter in the CES HDFS configuration in `/var/mmfs/hadoop/etc/hadoop`.

b. Restart the Livy service after the **dfs.blocksize** parameter in the Cloudera Manager GUI matches the **dfs.blocksize** parameter in the CES HDFS configuration.

13. Unable to create the managed Hive tables.

    All the tables that are created are external tables even when you explicitly requested to create managed tables.

    Solution:

    For creating the managed Hive tables, install the Hive on Tez service in CDP Private Cloud Base. For information on adding the Hive on Tez service, see Installing Hive on Tez and adding a HiveServer role.

14. While creating the encryption key you see an authorization exception in the Ranger KMS GUI.

    Solution:

    Add the following parameters in `kms-site.xml` from Cloudera Manager GUI and retry:

    - **hadoop.kms.proxyuser.rangeradmin.hosts**=*
    - **hadoop.kms.proxyuser.rangeradmin.groups**=*
    - **hadoop.kms.proxyuser.rangeradmin.users**=*

15. When uploading Oozie shareLib in IBM Storage Scale, you face the error `blocksize(xxxx) should be an integral mutiple of dataBlockSize(yyyy)`. This error occurs because Oozie always tries different blocksize values when uploading shareLib.

    Solution:

    a. Go to **Cloudera Manager GUI** > **IBM Storage Scale** > **Configuration**.

    b. Search the **HDFS Client Advanced Configuration Snippet (Safety Valve)** for `hdfs-site.xml` and add or update **dfs.namenode.fs-limits.min-block-size** = *<dfs.blocksize>*

    c. Save and deploy the client configuration.

    d. Restart IBM Storage Scale and Oozie services.

16. The **hadoop_secure_web_ui** configuration parameter is not effective for the Yarn service when IBM Storage Scale is integrated.

    In the Yarn service, even if the **hadoop_secure_web_ui** configuration parameter is set to *Enabled*, the **Resource Manager** and the **History Server** web user interfaces still use simple authentication.

    Solution:

    This issue is fixed in Cloudera Manager 7.6.1+ for IBM Storage Scale as HDFS provider. Upgrade Cloudera Manager to 7.6.1.

Chapter 5. Cloudera HDP 3.X

This section describes the deployment of Cloudera Hortonworks Data Platform (HDP®) 3.X on the IBM Spectrum Scale™ file system by using the Apache© Ambari framework.

Note: Starting with IBM Spectrum Scale 5.1.1.2, Cloudera Hortonworks Data Platform (HDP) is no longer supported. This content is kept in the documentation set merely for reference.

This section specifies the HDP deviations requirements for IBM Spectrum Scale integration and IBM Spectrum Scale specifics information and should be read beforehand to understand any deviation required when following the Hortonworks installation documentation for your specific platform.

Note: HDP 3.x is certified with IBM Spectrum Scale. This certification is for IBM Spectrum Scale software. Therefore, it applies to all the deployment models of IBM Spectrum Scale, including IBM Elastic Storage Server. The certification applies to HDP with HDF running on x86 or Power servers.

Planning

Review the "Hadoop IBM Storage Scale Architecture" on page 4 on which the configuration setup is to be used in your environment.

Hardware requirements

This section specifies the hardware requirements to install Hortonworks Data Platform (HDP®), IBM Spectrum Scale Ambari management pack and HDFS Transparency on IBM Spectrum Scale.

In addition to the normal operating system, IBM Spectrum Scale and Hadoop requirements, the Transparency connector has minimum hardware requirements of one CPU (processor core) and 4GB to 8GB physical memory on each node where it is running. This is a general guideline and might vary. For more planning information, see Chapter 2, "IBM Storage Scale support for Hadoop," on page 3.

Preparing the environment

This section describes how to prepare the environment to install Hortonworks Data Platform (HDP®), IBM Spectrum Scale Ambari management pack, and HDFS Transparency on IBM Spectrum Scale.

The IBM Spectrum Scale Ambari management pack can be used for Ambari 2.7 for Hortonworks HDP to setup the IBM Spectrum Scale Service.

HDP requires specific version combinations for the Ambari, Mpack and HDFS Transparency. The IBM Spectrum Scale file system is independent of the versioning for HDP, Mpack and HDFS Transparency. Hortonworks has been certified to work with IBM Spectrum Scale 4.2.3 and later.

Support matrix

This section describes the Hadoop distribution support matrix.

Cloudera Hortonworks Data Platform (HDP) release is end of support. For more information, see Cloudera lifecycle policy.

Table 34. Hadoop distribution support matrix

IBM Spectrum Scale Ambari management pack (Mpack)	HDFS Transparency	Hadoop distribution	Ambari version	RHEL x86_64	RHEL ppc64le	SLES x86
Mpack 2.7.0.10	HDFS Transparency 3.1.0-1 to latest 3.1.0-x stream	HDP 3.1.5 HDP 3.1.4	Ambari 2.7.5.17-6	7.9	NA	NA
Mpack 2.7.0.9	HDFS Transparency 3.1.0-1 to latest 3.1.0-x stream	HDP 3.1.5 HDP 3.1.4	Ambari 2.7.5.0 Ambari 2.7.4.0	7.6/7.7/7.8/7.9	7.6/7.7	SLES 12 SP3/SLES 12 SP4[1]
Mpack 2.7.0.8	HDFS Transparency 3.1.0-1 to latest 3.1.0-x stream	HDP 3.1.5 HDP 3.1.4	Ambari 2.7.5.0 Ambari 2.7.4.0	7.2/7.3/7.4/7.5/7.6/7.7/7.8/7.9	7.4/7.5/7.6/7.7	SLES 12 SP3/SLES 12 SP4[1]
Mpack 2.7.0.7 Superseded by Mpack 2.7.0.8	HDFS Transparency 3.1.0-1 to latest 3.1.0-x stream	HDP 3.1.5 HDP 3.1.4	Ambari 2.7.5.0 Ambari 2.7.4.0	7.2/7.3/7.4/7.5/7.6/7.7	7.4/7.5/7.6/7.7	SLES 12 SP3/SLES 12 SP4[1]
Mpack 2.7.0.6 Superseded by Mpack 2.7.0.8	HDFS Transparency 3.1.0-1 to latest 3.1.0-x stream	HDP 3.1.5	Ambari 2.7.5.0	7.2/7.3/7.4/7.5/7.6/7.7	7.4/7.5/7.6/7.7	SLES 12 SP3/SLES 12 SP4[1]
Mpack 2.7.0.5 Superseded by Mpack 2.7.0.8	HDFS Transparency 3.1.0-1 to latest 3.1.0-x stream	HDP 3.1.4	Ambari 2.7.4.0	7.2/7.3/7.4/7.5/7.6	7.4/7.5/7.6	SLES 12 SP3[1]
Mpack 2.7.0.4 Superseded by Mpack 2.7.0.8	HDFS Transparency 3.1.0-1 to latest 3.1.0-x stream	HDP 3.1.4	Ambari 2.7.4.0	7.2/7.3/7.4/7.5/7.6	7.4/7.5/7.6	SLES 12 SP3[1]
Mpack 2.7.0.3	HDFS Transparency 3.1.0-1 to latest 3.1.0-x stream	HDP 3.1.0	Ambari 2.7.3.0	7.2/7.3/7.4/7.5/7.6	7.4/7.5/7.6	SLES 12 SP3[1]

Table 34. Hadoop distribution support matrix (continued)						
IBM Spectrum Scale Ambari management pack (Mpack)	**HDFS Transparency**	**Hadoop distribution**	**Ambari version**	**RHEL** **x86_64**	**RHEL** **ppc64le**	**SLES x86**
Mpack 2.7.0.2	HDFS Transparency 3.1.0-0	HDP 3.1.0	Ambari 2.7.3.0	7.2/7.3/7.4/ 7.5/7.6	7.4/7.5/7.6	SLES 12 SP3[1]
Mpack 2.7.0.1	HDFS Transparency 3.1.0-0	HDP 3.0.1	Ambari 2.7.0.1	7.2/7.3/7.4/ 7.5	7.4/7.5	
Mpack 2.7.0.0	HDFS Transparency 3.0.0-0	HDP 3.0.0	Ambari 2.7.0.0	7.2/7.3/7.4/ 7.5	7.4/7.5	

[1] For SLES12 environment, Mpack 2.7.0.2 only supports new HDP installations.

Note:

- HDP and Mpack requires Python 2.7.
- HDP Java Development Kits (JDKs): OpenJDK on `ppc64le` and Oracle JDK on x86_64.
- IBM Spectrum Scale file system release supported: 4.1.1.3+, 4.2.2.3+, 4.2.3.1+,5.0.X+ on pc64le and x86_64.

  RH 7.5 is supported from V4.2.3.9 and V5.0.1.1.

  RH 7.6 is supported from V4.2.3.13 and V5.0.2.2.
- IBM Spectrum Scale Management GUI function requires RHEL 7.2+ at IBM Spectrum Scale version 4.2.X+.
- IBM Spectrum Scale snap data for Hadoop function requires RHEL 7.2+ at IBM Spectrum Scale version 4.2.2.X+.
- Preserving Kerberos token delegation during NameNode failover with HDP is supported when Mpack 2.7.0.1 and HDFS Transparency 3.1.0-0 are applied together.
- HDP 3.1 with Mpack 2.7.0.3 and HDFS Transparency 2.7.0.3 supports ViewFS in Ambari. For more information, see "Configure ViewFs on HDFS clusters without HA" on page 171 and "Configuring ViewFs on HDFS cluster with HA" on page 173.
- HDP supports HDFS Transparency 3.1.0-x stream and earlier.
- CES HDFS does not support HDP.
- With Mpack 2.7.0.7, you can upgrade HDP 3.1.x HA without unintegrating HDFS Transparency. For more information, see "Upgrading HDP overview" on page 373.
- From Mack 2.7.0.8, to change the IBM Spectrum Scale tunables, run the corresponding **mm*** command for that specific tunable on the command line. Therefore, the tunables seen in the IBM Spectrum Scale service panel and the IBM Spectrum Scale cluster/file system will not be the same. Refer to the IBM Spectrum Scale cluster/file system information from command line for current values of the tunables for the node/cluster.

  When you are deploying Mpack, upgrading Mpack, reintegrating HDFS Transparency, and restarting the IBM Spectrum Scale service, if the **gpfs.storage.type** parameter is set to *shared*, the IBM Spectrum Scale configurations will not be overwritten.

Setup

This section gives the setup information for Hortonworks Data Platform (HDP) and IBM Spectrum Scale Hadoop integration support.

Local Repository server

Set up a local repository server to be used for Ambari, IBM Spectrum Scale, and for the OS repository.

1. Set up the Mirror repository server.
2. Set up the Local OS repository if needed.

Base packages

The following packages must be installed on all IBM Spectrum Scale nodes:

```
$ yum -y install kernel-devel cpp gcc gcc-c++ binutils
```

For more information, see IBM Spectrum Scale *Software requirements* topic in the *IBM Storage Scale: Concepts, Planning, and Installation Guide*.

HDP packages:

- Install libtirpc-devel package (From RH optional packages) on all nodes.
- MySQL community edition

  For new database install option through HDP for Hive Metastore, the MySQL community would require internet access or have a local repository setup to deploy on the Hive Metastore host. For more information, see MySQL Community Edition repository.

The following recommended packages can be downloaded to all nodes:

`acl`, `libacl` – to enable Hadoop ACL support

`libattr` – to enable Hadoop extended attributes

`java-<version>-openjdk-devel` – Development tool-kit required for short circuit

Some of these packages are installed by default while installing the operating system.

Create the anonymous user id

If the anonymous user id does not exist already, create it for all the nodes in the IBM Spectrum Scale cluster on the Hadoop cluster.

In non-kerberos clusters, the anonymous user is mandatory while starting the IBM Spectrum Scale Service.

Ensure that the UID and GID for the anonymous user have the same value for all the nodes in the IBM Spectrum Scale cluster.

The anonymous user id is used for Hive.

For more information, see "User and group ids" on page 355.

Hortonworks Data Platform (HDP)

This topic helps in the preparation to install Hortonworks Data Platform (HDP).

If you are installing the HDP package, follow the *Getting Ready*, *Meet Minimum System Requirements*, *Prepare the Environment*, *Using a Local Repository* and *Obtaining Public Repositories* sections from the installation guide of your platform. For the *Apache Ambari Installation for IBM Power Systems* guide and *Apache Ambari Installation* guide for the x86 platform, see Hortonworks Documentation website for the HDP version you are using.

Scale integration deviation requirement:

- Maximum Open Files Requirements

Set the maximum number of open file descriptors to *65535*.

- When HDP is integrated with IBM Spectrum Scale, there is no Secondary NameNode.
- For specific changes related to Kernel, SELinux and NTP, password-less ssh and User/group id, see "IBM Spectrum Scale file system" on page 353.

HDFS Transparency package

This topic helps in the preparation to install HDFS Transparency package.

IBM Spectrum Scale HDFS Transparency (HDFS Protocol) offers a set of interfaces that allows applications to use HDFS Client to access IBM Spectrum Scale through HDFS RPC requests.

All data transmission and metadata operations in HDFS are done through the RPC mechanism, and processed by the NameNode and the DataNode services within HDFS.

From IBM Spectrum Scale 5.0.3, the IBM Spectrum Scale `gpfs_rpms` directory will no longer host a version of the HDFS Transparency. Multiple versions of HDFS Transparency will reside in the `hdfs_rpms` directory. Choose the HDFS Transparency version corresponding to the support matrix and copy it to location where the gpfs rpms resides.

IBM Spectrum Scale HDFS Transparency is independently installed from IBM Spectrum Scale and provided as an rpm package. HDFS Transparency supports both local and shared storage modes.

You can download the IBM Spectrum Scale HDFS Transparency from the "HDFS Transparency download" on page 28 section.

The module name is `gpfs.hdfs-protocol-3.<release>.0-(version)`.

Save this module in the IBM Spectrum Scale repository.

Note:

- Ensure that there is only one package of the transparency in the IBM Spectrum Scale repository. Rebuild the repository by executing the **`"createrepo . "`** command to update the repository metadata.
- Properly review and set "OS tuning for all nodes in HDFS Transparency" on page 55 and "Configure NTP to synchronize the clock in HDFS Transparency" on page 56 for all nodes.

IBM Spectrum Scale file system

This topic helps in the preparation to install IBM Spectrum Scale file system.

For IBM Spectrum Scale overview, see the *Product overview* section in the *IBM Storage Scale: Concepts, Planning, and Installation Guide*.

If you have purchased the IBM Spectrum Scale license, you can download the IBM Spectrum Scale base installation package files from the IBM Passport Advantage website.

For IBM Spectrum Scale version 4.1.1.7 and later or version 4.2.0.1 and later, full images are available through Fix Central.

For IBM Spectrum Scale trial and purchase licenses: https://www.ibm.com/us-en/marketplace/scale-out-file-and-object-storage/purchase

To order IBM Spectrum Scale, see Question 1.1 in IBM Spectrum Scale FAQ documentation.

The latest IBM Spectrum Scale update package (PTF) files can be obtained from Fix Central.

Note: Starting with release 4.2.3, IBM Spectrum Scale Express Edition is no longer available.

Kernel, SELinux and NTP

This topic gives information about Kernel, SELinux and NTP.

Kernel

See Installation of Kernel packages under the IBM Spectrum Scale support for Hadoop Kernel section.

SELinux

SELinux must be in disabled mode.

NTP

For Hadoop section, see "NTP" on page 18 under the IBM Spectrum Scale software requirement section.

Network validation

While using a private network for Hadoop DataNodes, ensure that all nodes, including the management nodes, have hostnames bound to the faster internal network or the data network.

On all nodes, the hostname -f must return the FQDN of the faster internal network. This network can be a bonded network. If the nodes do not return the FQDN, modify `/etc/sysconfig/network` and use the hostname command to change the FQDN of the node.

The `/etc/hosts` file host order listing must have the long hostname first before the short hostname. Otherwise, the HBase service check in Ambari can fail.

If the nodes in your cluster have two network adapters, see Dual Network Deployment.

Setting password-less ssh access for root

IBM Spectrum Scale Master is a role designated to the host on which the Master component of the IBM Spectrum Scale service is installed. It should be a part of the administrator nodes set. All the IBM Spectrum Scale cluster wide administrative commands including those for creation of the IBM Spectrum Scale cluster and the file-system are run from this host.

Password-less ssh access for root must be configured from the IBM Spectrum Scale Master node to all the other IBM Spectrum Scale nodes. This is needed for IBM Spectrum Scale to work. For non-adminMode central clusters, ensure that you have bi-directional password-less setup for the fully qualified and short names for all the GPFS™ nodes in the cluster. This must be done for the root user. For non-root Ambari environment, ensure that the non-root ID can perform bi-directional password-less SSH between all the GPFS nodes.

Note: BDA Ambari integration supports the admin mode central configuration of IBM Spectrum Scale (*adminMode configuration attribute* topic in the *IBM Storage Scale: Administration Guide*).

In this configuration, one or more hosts could be designated as IBM Spectrum Scale Administration (or Admin) nodes. By default, the GPFS Master is an Admin node. In Admin mode central configuration, it is sufficient to have only uni-directional password-less ssh for root from the Admin nodes to the non-admin nodes. This configuration ensures better security by limiting the password-less ssh access for root.

An example on setting up password-less access for root from one host to another:

1. Define Node1 as the IBM Spectrum Scale master.
2. Log on to Node1 as the root user.

   ```
   # cd /root/.ssh
   ```

3. Generate a pair of public authentication keys. Do not type a passphrase.

   ```
   # ssh-keygen -t rsa
   ```

   Generate the public-private rsa key pair.

   Type the name of the file in which you want to save the key (/root/.ssh/id_rsa):

   Type the passphrase.

   Type the passphrase again.

   The identification has been saved in /root/.ssh/id_rsa.

   The public key has been saved in /root/.ssh/id_rsa.pub.

   The key fingerprint is:

   ...

Note: During **`ssh-keygen -t rsa`**, accept the default for all.

4. Set the public key to the `authorized_keys` file.

   ```
   # cd /root/.ssh/; cat id_rsa.pub > authorized_keys
   ```

5. For clusters with adminMode as *allToAll*, copy the generated public key file to nodeX.

   ```
   # scp /root/.ssh/* root@nodeX:/root/.ssh
   ```

   where, nodeX is all the nodes.

   For clusters with adminMode as *central*, copy the generated public key file to nodeX.

   ```
   # scp /root/.ssh/* root@nodeX:/root/.ssh
   ```

   nodeX is all the nodes chosen for administration.

   Configure the password less ssh with non admin nodes (*nodeY*) in the clusters.

   ```
   # ssh-copy-id root@nodeY
   ```

   nodeY is rest of the cluster nodes.

6. Ensure that the public key file permission is correct.

   ```
   #ssh root@nodeX “chmod 700 .ssh; chmod 640 .ssh/authorized_keys"
   ```

7. Check password-less access

   ```
   # ssh node2

   [root@node1 ~]# ssh node2
   The authenticity of host 'gpfstest9 (192.0.2.0)' can't be established.
   RSA key fingerprint is 03:bc:35:34:8c:7f:bc:ed:90:33:1f:32:21:48:06:db.
   Are you sure you want to continue connecting (yes/no)?yes
   ```

   Note: You also need to run **`ssh node1`** to add the key into `/root/.ssh/known_hosts` for password-less access.

User and group ids

Ensure that all user IDs and group IDs used in the cluster for running jobs, accessing the IBM Spectrum Scale file system or for the Hadoop services must be created and have the same values across all the IBM Spectrum Scale nodes. This is required for IBM Spectrum Scale.

The recommendation is to create the uid/gid manually on all the nodes for the service before deploying the service. This is because Ambari creates inconsistent uid/gid across the cluster after the initial deployment on a fresh cluster. See Jira AMBARI-12616. For a list of the service user and group account uid/gid see the *Default user accounts* and *Default group accounts* sections under Understanding service users and groups.

Note: Ensure that these user and group accounts uid and gid have the same values across all the IBM Spectrum Scale nodes.

- If you are using LDAP, create the IDs and groups on the LDAP server and ensure that all nodes can authenticate the users.
- If you are using local IDs, the IDs must be the same on all nodes with the same ID and group values across the nodes.
- If you setup remote mount access for IBM Spectrum Scale, the owning cluster does not require to have the Hadoop uid and gid configured because there are no applications running on those nodes. However, if the owning cluster have other applications from non Hadoop clients, they need to ensure that the uid and gid used by the Hadoop cluster are not the same as the one used by the non Hadoop clients.
- The anonymous user is not used by Hive if the **`hive.server2.authentication`** is configured as LDAP or Kerberos enabled. However, the default setting for **`hive.server2.authentication`** is set

to *NONE*. Therefore, no authentication is done for Hive's requests to the Hiveserver2 (meta data). This means that all the requests are completed as anonymous user. For more information, see "Create the anonymous user id" on page 352 section.

Note: UID or GID is the common way for a Linux system to control access from users and groups. For example, if the user Yarn UID=100 on node1 generates data and the user Yarn UID=200 on node2 wants to read this data, the read operation fails because of permission issues.

Keeping a consistent UID and GID for all users on all nodes is important to avoid unexpected issues.

For the initial installation through Ambari, the UID or GID of users are consistent across all nodes. However, if you deploy the cluster for the second time, the UID or GID of these users might be inconsistent over all nodes (as per the AMBARI-10186 issue that was reported to the Ambari community).

After deployment, check whether the UID is consistent across all nodes. If it is not, you must fix it by running the following commands on each node, for each user or group that must be fixed:

##### Change UID of one account:

```
usermod -u <NEWUID><USER>
```

##### Change GID of one group:

```
groupmod -g <NEWGID><GROUP>
```

##### Update all files with old UID to new UID:

```
find / -user <OLDUID> -exec chown -h <NEWUID> {} \;
```

##### Update all files with old GID to new GID:

```
find / -group <OLDGID> -exec chgrp -h <NEWGID> {} \;
```

##### Update GID of one account:

```
usermod -g <NEWGID><USER>
```

IBM Spectrum Scale local repository

IBM Spectrum Scale only supports installation through a local repository.

Note: If you have already setup an IBM Spectrum Scale file system, you can skip this section.

1. Ensure there is a Mirror repository server created before proceeding.
2. Setup the Local OS repository if needed.
3. Setup the Local IBM Spectrum Scale repository. This section helps you to set up the IBM Spectrum Scale and HDFS Transparency local repository.

IBM Spectrum Scale service (Mpack)

The IBM Spectrum Scale Ambari management pack is an Ambari service for IBM Spectrum Scale.

For traditional Hadoop clusters that use HDFS, an HDFS service is displayed in the Ambari console to provide a graphical management interface for the HDFS configuration (hdfs-site.xml) and the Hadoop cluster (core-site.xml). Through the Ambari HDFS service, you can start and stop the HDFS service, make configuration changes, and implement the changes across the cluster.

The management pack creates an Ambari IBM Spectrum Scale service to start, stop, and make configuration changes to IBM Spectrum Scale and HDFS Transparency. Once the IBM Spectrum Scale HDFS Transparency is integrated, the HDFS service manages the HDFS Transparency NameNodes and DataNodes instead of the native HDFS components.

To download the management pack from the IBM Spectrum Scale, go to the "Mpack download" on page 357 topic.

The management pack version 2.7.x.x contains the following files:

- `SpectrumScaleIntegrationPackageInstaller-2.7.x.x.bin`
- `SpectrumScaleMPackInstaller.py`
- `SpectrumScaleMPackUninstaller.py`
- `SpectrumScale_UpgradeIntegrationPackage` [Upgrade IBM Spectrum Scale Mpack]
- `mpack_utils.py`
- `sum.txt`

Note: Ensure that all the packages reside in the same directory before executing the executables.

Mpack download

Visit IBM Fix Central to download the IBM Spectrum Scale Management Pack for Ambari (Mpack) and search for *Spectrum_Scale_Management_Pack_MPACK_for_Ambari-<version>-noarch-Linux* to find the correct package.

Untar the download package:

```
tar zxvf Spectrum_Scale_Management_Pack_MPACK_for_Ambari-<version>-noarch-Linux.tgz
```

Ensure that you review the "Support matrix" on page 349 section to get the correct Mpack for your environment.

Ambari user id for IBM Spectrum Scale deployment

The user id to deploy and manage the IBM Spectrum Scale Mpack should be able to invoke the Ambari REST API. It is also recommended that the user id be a local Ambari user id and not the Domain user id.

The default is usually the Ambari user id.

If the user id is not able to invoke the Ambari REST API, then the GPFS slave fails to get a response from Ambari with the following error:

```
File "/usr/lib64/python2.7/urllib2.py", line 409, in _call_chain
    result = func(*args)
  File "/usr/lib64/python2.7/urllib2.py", line 558, in http_error_default
    raise HTTPError(req.get_full_url(), code, msg, hdrs, fp)
urllib2.HTTPError: HTTP Error 500: Server Error
```

In the ambari-server.log, there must be messages stating that the domain user is being added to various groups and performing permission checks before the URL request is being processed. This will later cause an internal server error in the Ambari server.

Preparing for FPO environment

This section provides the information for preparing for a FPO deployment. To create a FPO cluster, you need to create a NSD file beforehand.

Preparing a stanza file

The Ambari install process can install and configure a new IBM Spectrum Scale cluster file system and configure it for Hadoop workloads. To support this task, the installer must know the disks available in the cluster and how you want to use them. If you do not indicate preferences, intelligent defaults are used. The stanza file is used for new FPO deployment through Ambari by setting the GPFS NSD file field under the IBM Spectrum Scale Standard Configs panel.

Simple NSD File

This section describes a simple NSD file with an example.

Simple NSD file can only be used for full disk that have not already been partitioned as input to Ambari.

All disks of GPFS NSD server nodes requires to be listed in the NSD stanza file.

The following is an example of a preferred simple IBM Spectrum Scale NSD file:

There are 7 nodes, each with 6 disk drives to be defined as NSDs. All information must be continuous with no extra spaces.

```
$ cp /var/lib/ambari-server/resources/gpfs_nsd.sample /var/lib/ambari-server/resources/gpfs_nsd
$ cat /var/lib/ambari-server/resources/gpfs_nsd

DISK|compute001.private.dns.zone:/dev/sdb,/dev/sdc,/dev/sdd,/dev/sde,/dev/sdf,/dev/sdg
DISK|compute002.private.dns.zone:/dev/sdb,/dev/sdc,/dev/sdd,/dev/sde,/dev/sdf,/dev/sdg
DISK|compute003.private.dns.zone:/dev/sdb,/dev/sdc,/dev/sdd,/dev/sde,/dev/sdf,/dev/sdg
DISK|compute005.private.dns.zone:/dev/sdb,/dev/sdc,/dev/sdd,/dev/sde,/dev/sdf,/dev/sdg
DISK|compute006.private.dns.zone:/dev/sdb,/dev/sdc,/dev/sdd,/dev/sde,/dev/sdf,/dev/sdg
DISK|compute007.private.dns.zone:/dev/sdb,/dev/sdc,/dev/sdd,/dev/sde,/dev/sdf,/dev/sdg
```

If you want to select disks such as SSD drives for metadata , add the label -meta to those disks.

In a simple NSD file, add the label meta for the disks that you want to use as metadata disks, as shown in the following example. If -meta is used, the Partition algorithm is ignored.

```
$ cat /var/lib/ambari-server/resources/gpfs_nsd

DISK|compute001.private.dns.zone:/dev/sdb-meta,/dev/sdc,/dev/sdd
DISK|compute002.private.dns.zone:/dev/sdb-meta,/dev/sdc,/dev/sdd
DISK|compute003.private.dns.zone:/dev/sdb-meta,/dev/sdc,/dev/sdd
DISK|compute005.private.dns.zone:/dev/sdb-meta,/dev/sdc,/dev/sdd
DISK|compute006.private.dns.zone:/dev/sdb,/dev/sdc,/dev/sdd
DISK|compute007.private.dns.zone:/dev/sdb,/dev/sdc,/dev/sdd
```

In the simple NSD file, `/dev/sdb` from compute001, compute002, compute003, and compute005 are specified as meta disks in the IBM Spectrum Scale file system.

The partition algorithm is ignored if the nodes listed in the simple NSD file do not match the set of nodes that will be used for the NodeManager service. If nodes that are not NodeManagers are in the NSD file or nodes that will be NodeManagers are not in the NSD file, no partitioning will be done.

Standard NSD file

This section describes a standard NSD file with an example.

The following is an example of a Standard IBM Spectrum Scale NSD File.

```
%pool: pool=system  blockSize=256K layoutMap=cluster allowWriteAffinity=no
%pool: pool=datapool blockSize=2M layoutMap=cluster allowWriteAffinity=yes writeAffinityDepth=1
blockGroupFactor=256

# gpfstest9
%nsd: nsd=node9_meta_sdb  device=/dev/sdb servers=gpfstest9 usage=metadataOnly failureGroup=101
pool=system
%nsd: nsd=node9_meta_sdc  device=/dev/sdc servers=gpfstest9 usage=metadataOnly
failureGroup=101  pool=system

%nsd: nsd=node9_data_sde2  device=/dev/sde2 servers=gpfstest9 usage=dataOnly failureGroup=1,0,1
pool=datapool
%nsd: nsd=node9_data_sdf2  device=/dev/sdf2 servers=gpfstest9 usage=dataOnly failureGroup=1,0,1
pool=datapool

# gpfstest10
%nsd: nsd=node10_meta_sdb  device=/dev/sdb servers=gpfstest10 usage=metadataOnly
failureGroup=201 pool=system
%nsd: nsd=node10_meta_sdc  device=/dev/sdc servers=gpfstest10 usage=metadataOnly
failureGroup=201 pool=system

%nsd: nsd=node10_data_sde2  device=/dev/sde2 servers=gpfstest10 usage=dataOnly
failureGroup=2,0,1 pool=datapool
%nsd: nsd=node10_data_sdf2  device=/dev/sdf2 servers=gpfstest10 usage=dataOnly
failureGroup=2,0,1 pool=datapool

# gpfstest11
%nsd: nsd=node11_meta_sdb  device=/dev/sdb servers=gpfstest11 usage=metadataOnly
failureGroup=301 pool=system
%nsd: nsd=node11_meta_sdc  device=/dev/sdc servers=gpfstest11 usage=metadataOnly
failureGroup=301 pool=system

%nsd: nsd=node11_data_sde2  device=/dev/sde2 servers=gpfstest11 usage=dataOnly
failureGroup=3,0,1 pool=datapool
```

```
%nsd: nsd=node11_data_sdf2  device=/dev/sdf2 servers=gpfstest11 usage=dataOnly
failureGroup=3,0,1 pool=datapool
```

Note: Starting with IBM Spectrum Scale version 5.0.0, the default block size is 4M.

Type the `/var/lib/ambari-server/resources/gpfs_nsd` filename in the NSD stanza field.

Because of the limitations of the Ambari framework, the NSD file must be copied to the Ambari server in the `/var/lib/ambari-server/resources/` directory. Ensure that the correct file name is specified on the IBM Spectrum Scale Customize Services page.

If you are using a standard NSD stanza file and only one datapool is defined, you can either specify the policy file or let it be done by IBM Spectrum Scale. However, if you have more than one data pool, you should specify a policy to define the location of the data in the data pool. If there is no policy specified, by default the data will be stored to the first data pool only.

Policy File

This section describes a policy file with an example.

The `bigpfs.pol` is an example of a policy file.

```
RULE 'default' SET POOL 'datapool'
```

Preparing for the ECE environment

The ECE storage is required to be configured as shared storage to be used in the Hadoop environment.

Ensure that the "Recommended hardware resource configuration" on page 16 for ECE is met and follow the *IBM Spectrum Scale Erasure Code Edition* documentation for installation.

Note: Ensure that the IBM Spectrum Scale file system ACL setting is set to **ALL**. For more information, see HDFS and IBM Spectrum Scale file system ACL support section.

After the ECE storage is set up and configured, add the ECE storage to the Hadoop environment by using the same method as the shared storage mode. See "Deploy HDP or IBM Spectrum Scale service on pre-existing IBM Spectrum Scale file system" on page 404 section.

Note: FPO is not supported on ECE.

Installation

This section describes the installation and deployment of HDP and IBM Spectrum Scale™ Hadoop integration that consists of the management pack and HDFS Transparency connector.

This installation section will describe how to install ESS Remote mount with all Hadoop nodes as IBM Spectrum Scale nodes as the 1st deployment model as described in the "Hadoop IBM Storage Scale Architecture" on page 4.

Before starting the software deployment, follow the Planning section to setup the environment and download the software.

Note:

- This chapter describes how to add IBM Spectrum Scale service as a root user. If you plan to restrict root access, review the "Restricting root access" on page 450.
- To install the IBM Spectrum Scale service, an existing HDFS cluster is required. This can be created by installing the HDP stack with native HDFS.

For other installation scenarios, see "Configuration" on page 394.

ESS setup

The ESS is setup and tuned by IBM Lab Services.

Note: Ensure that the IBM Spectrum Scale file system ACL setting is set to *ALL*.

For more information, see the HDFS and IBM Spectrum Scale file system ACL support section.

Adding Services

HDFS and IBM Spectrum Scale are different file systems. If services are added only to one of the file systems, the other file system does not have the data for that service. Therefore, on switching from native HDFS to IBM Spectrum Scale or vice versa, the service cannot provide the data that you entered before switching the file system.

Ensure that you follow the "User and group ids" on page 355 section for uid/gid consistency setup and verification.

The following are the minimum services required to be installed before you install the IBM Spectrum Scale service:

- HDFS
- Yarn
- Mapreduce2
- Zookeeper
- SmartSense (HDP)

When adding new services to HDP, ensure that you review all configurations in the Customize Services wizard tabs. Especially, fields that set directories. This is because IBM Spectrum Scale is a shared file system. Ensure that any services do not use a shared file system mount point when it requires a local directory. See **Create HDP cluster>Installing, Configuring, and Deploying a Cluster >Customize Services>Directories** section.

If the service is already added with the remote mount point set as one of the directories, and you want to modify or remove the remote mount point directory from the service, ensure that you restart the service to pick up the new configuration changes.

If you are planning to install High Availability (HA), ensure that you setup the HA in native HDFS mode. Do not install HA when IBM Spectrum Scale Service is integrated.

If IBM Spectrum Scale is integrated, see "HDFS NameNode High Availability [HA]" on page 394.

Create HDP cluster

Follow the Installing Ambari section in Hortonworks HDP installation documentation (https://docs.hortonworks.com) for your specific platform.

This section will mention the deviations required in the setup while integrating with IBM Spectrum Scale.

Note: Ambari uses the mount points it finds on the Ambari server to set up the services during deployment. If you do not want to use the mount points, ensure that you unmount the mount points on the Ambari server node before starting Ambari or find and change the mount point in the configuration setting for each service that uses it. See **Create HDP cluster>Installing, Configuring, and Deploying a Cluster >Customize Services>Directories** for a list of services that use the mount point value.

Ensure that you follow the "User and group ids" on page 355 section for uid/gid consistency setup and verification.

Install Ambari bits

Ensure that the `ambari.repo` is setup on `/etc/yum.repos.d` on the Ambari server host.

On the Ambari server host, run:

```
yum install ambari-server
```

- Update the `/etc/ambari-server/conf/ambari.properties` file to point to the correct Open JDK and JCE files.

  ```
  $ vi /etc/ambari-server/conf/ambari.properties
  jdk1.8.jcpol-url to point to the correct jce_policy-8.zip file.
  jdk1.8.url to point to the correct jdk-8u112-linux-x64.tar.gz file.
  ```

- Update the number of threads in `/etc/ambari-server/conf/ambari.properties` file.

  The size of the threadpool must be set to the number of logical cpus on the node on which the Ambari server is running. When the number of threads is not enough in Ambari, the system might suffer a heartbeat loss and the DataNodes might go down. The Ambari GUI might not be able to start if enough threads are not available. This is especially true for Power system.

  Threadpool values requiring to be modified in `/etc/ambari-server/conf/ambari.properties`:

  ```
  $ vi /etc/ambari-server/conf/ambari.properties
  server.execution.scheduler.maxThreads=<number of logical cpu's>
  client.threadpool.size.max=<number of logical cpu's>
  agent.threadpool.size.max=<number of logical cpu's>
  ```

  To calculate the number of logical cpus:

  ```
  $ lscpu
  Thread(s) per core:    8
  Core(s) per socket:    1
  Socket(s):             20

  Number of logical cpu's = Thread(s) per core x Core(s) per socket x Socket(s) = 8 x 1 x 20 =
  160
  ```

Set up the Ambari server

This topic describes how to set up the Ambari server.

On the Ambari server host, run:

```
ambari-server setup
```

Note: If you are using Hive MySQL, you also need to set up the MySQL connector and run `ambari-server setup --jdbc-db=mysql --jdbc-driver=/path/to/mysql/mysql-connector-java.jar`.

See Hortonworks Using Hive with MySQL.

If you plan to use a non-root user id, see "Restricting root access" on page 450.

Installing, configuring, and deploying a cluster

This section states the deviations from the Hortonworks documentation, Installing, Configuring, and Deploying a Cluster section when integrating IBM Spectrum Scale service.

Start the Ambari server

This section describes the procedure to start the Ambari server.

On the Ambari server host, run: `ambari-server start`.

Log in to Apache Ambari

Open `http://<your.ambari.server>:8080` in your web browser.

Select version

Select the software version and method of delivery for your cluster.

Using a Local Repository requires you to configure the software in a repository available in your network.

For this deployment:

- Select Use Local Repository.
- Set HDP and HDP UTILS repository for redhat7/redhat-ppc7.

Install options

The cluster wizard prompts for general information on how to setup the cluster.

In the Target hosts section:

- The ESS I/O servers must not be a part of the Ambari cluster.
- Ambari requires a list of fully qualified domain names (FQDNs) of the nodes in the cluster.
- Verify that the list of the host names used in the Ambari Target Hosts section are the data network addresses that IBM Spectrum Scale uses for the cluster setup. Otherwise, during the installation of the IBM Spectrum Scale service, the installation fails and gives an `Incorrect hostname` error.
- Ensure that the hostname does not have mixed case. Otherwise, failures might occur when starting services. It is recommended to use all lower case.

For this deployment, the ssh user is set to *root*.

If you plan to use a non-root user id, see "Restricting root access" on page 450.

Choose services

Choose the services to install into the cluster.

Note: For this configuration, Ranger is unchecked. You can add the Ranger at a later point.

For Scale integration specific information regarding adding services, see "Adding Services" on page 360 section.

Assign masters

The Cluster Install wizard assigns the master components for selected services.

Note:

- GPFS Master node should be colocated on the Ambari server host.
- The native HDFS NameNode will also become the IBM Spectrum Scale HDFS Transparency NameNode.
- It is recommended to assign the Yarn Resource Manager onto the native HDFS NameNode.

Assign slaves and clients

The Cluster Install wizard assigns the slave components to appropriate hosts in the cluster.

Note: Select all nodes to be DataNodes. This installs HDFS Transparency on all the IBM Spectrum Scale nodes on the Hadoop cluster.

Customize services

Customize services set of tabs to review and modify your cluster setup. The wizard sets defaults for each options.

Ensure that you review these settings carefully.

Note: Accumulo and HiveServer2 if installed on the same host will have port binding conflicts.

To work around the port conflict when starting the services:

- Put Accumulo and HiveServer2 on different hosts or
- Use non-default port for either of the service

Hive Database

In Hive panel, if you are using MySQL you need to set up the MySQL connector.

To use MySQL with Hive, you must download the MySQL Connector/J. On downloading to the Ambari Server host, run:

```
ambari-server setup --jdbc-db=mysql --jdbc-driver=/path/to/mysql/com.mysql.jdbc.Driver
```

Ambari SSL configuration

The Knox gateway server uses the 8443 port by default. The Knox gateway fails to start if the Ambari HTTPS uses the same port. Ensure that the Ambari HTTPS port and the Knox gateway port are unique, and are not used by other processes.

Directories

If the cluster has a mounted file system, Ambari can select mounted paths other than / as the default directory value for some of its services, when the local file system must be used. This might include a GPFS mounted directory. Either unmount the mount points on the Ambari server before starting Ambari, or you must manually find all the places in the Ambari installation configuration and set it to a local directory. Otherwise HDP services will not start or run correctly as the nodes in the cluster are accessing the same directories.

Following is the list of service configurations that need the local directory to be configured:

Service	Field Name	GUI Field Name	Local directory setting
HDFS	**dfs.namenode.name.dir**	NameNode directories	/hadoop/hdfs/namenode
HDFS	**dfs.datanode.data.dir**	DataNode directories	/hadoop/hdfs/data
Yarn	**yarn.nodemanager.local-dirs**	YARN NodeManager Local directories	/hadoop/yarn/local
Yarn	**yarn.nodemanager.log-dirs**	YARN NodeManager Log directories	/hadoop/yarn/log
Ambari Metrics	**hbase.rootdir**	HBase root directory	file:///var/lib/ambari-metrics-collector/hbase
Kafka	**log.dirs**	Log directories	/kafka-logs
Oozie	**oozie_data_dir**	Oozie Data Dir	/hadoop/oozie/data
Zookeeper	**dataDir**	ZooKeeper directory	/hadoop/zookeeper

Install, start and test

Ambari installs, starts, and runs a simple test on each component.

In the event of any failure during the initial cluster deployment, it is a good practice to go through each service one by one by running its service check command. Ambari runs all the service checks as part of the installation wizard, but if anything were to fail, Ambari might not have run all the service checks. On the dashboard page for each service in the Ambari GUI, go to each service's panel and click **Actions** > **Run Service Check**.

Summary - Complete

After the summary page shows a list of accomplished tasks, choose **Complete** to open the Ambari dashboard.

Note:

- Ensure that all services are up.
- Ensure that all service checks passed.

- Ensure that all the UID and GID have the same value across all the IBM Spectrum Scale cluster hosts after HDP is deployed.

Establish an IBM Spectrum Scale cluster on the Hadoop cluster

Establish a local IBM Spectrum Scale cluster on the Hadoop cluster. This local IBM Spectrum Scale cluster accesses the ESS via remote mount. This creates a multi-cluster Scale environment and one IBM ESS storage can be shared with different groups where the remote mount mode can isolate the storage management from the IBM Spectrum Scale local cluster.

Note:

- Ensure that the version of IBM Spectrum Scale on the local cluster is higher than or same as the version on the file system owning the cluster (ESS).
- The **maxblocksize** value requires to be the same on the local IBM Spectrum Scale cluster and the ESS cluster. The **maxblocksize** value can be set up during the installation of the local IBM Spectrum Scale cluster to be the same value as the ESS cluster. If the **maxblocksize** is not set, it defaults to *1 MB* for releases prior to IBM Spectrum Scale version 5.0.0, however from 5.0.0 release onwards it is set to *4 MB*.

Steps

1. Ensure that the `gpfs.repo` is in the `/etc/yum.repos.d` directory on all the nodes in the Hadoop cluster. On each node, run: `yum clean all; yum makecache`.

   For example, the `/etc/yum.repos.d/gpfs.repo` file contains:

   ```
   [GPFS-5.0.1]
   name=gpfs-5.0.1
   baseurl=http://60.2.0.229/repos/rhel/5.0.1/GPFS_5.0.1
   enabled=1
   gpgcheck=0
   ```

   Note: Ensure that the **gpfscheck** value is set to zero.

2. Install the IBM Spectrum Scale on your cluster. See the *Manually installing the IBM Spectrum Scale software packages on Linux nodes* topic in the *IBM Storage Scale: Concepts, Planning, and Installation Guide*.

   For example, on each of the Hadoop nodes, run:

   ```
   yum -y install gpfs.adv* gpfs.base* gpfs.crypto* gpfs.ext* gpfs.gpl* gpfs.gskit* gpfs.lice*
   gpfs.msg*
   ```

3. Build the kernel portability layer on each node by issuing the following command:

   ```
   /usr/lpp/mmfs/bin/mmbuildgpl
   ```

4. Follow the *Steps for establishing and starting your IBM Spectrum Scale cluster* topic in the *IBM Storage Scale: Concepts, Planning, and Installation Guide* for your specific Scale version.

   Note:

   - Do not create NSD (**mmcrnsd**) or a file system (**mmcrfs**) because this is a remote mount environment.
   - Ensure that the Ambari server is set as the IBM Spectrum Scale quorum node. The IBM Spectrum Scale Master node resides on the Ambari server node and requires to be set as a quorum node.

5. Check the **maxblocksize** value on the local cluster and ESS cluster by running the following command:

   ```
   /usr/lpp/mmfs/bin/mmlsconfig | grep maxblocksize
   ```

If **maxblocksize** value on the local cluster is not set or not the same as the ESS cluster, then on the local cluster, run the following command:

```
/usr/lpp/mmfs/bin/mmchconfigmaxblocksize=<ESSmaxblocksizevalue>
```

6. Start IBM Spectrum Scale by issuing the **mmstartup** command.

   ```
   /usr/lpp/mmfs/bin/mmstartup -a
   ```

7. Ensure that all the IBM Spectrum Scale nodes are in active state.

   ```
   /usr/lpp/mmfs/bin/mmgetstate -a
   ```

8. Tune the local cluster as an ESS client:

   For remote mount mode for Hadoop cluster (1st model: Remote mount with all Hadoop nodes as IBM Spectrum Scale nodes), run the following commands:

   On ESS run:

   ```
   scp /usr/lpp/mmfs/samples/gss/gssClientConfig.sh root@<Hadoop_local_scale_cluster_host>:</path-to-
   gssclient>
   ```

   On Hadoop local scale cluster host, run:

   ```
   <path-to-gssclient>/gssClientConfig.sh all
   ```

   However, if the IBM Spectrum Scale clients nodes and the ESS nodes are in the same cluster (3rd model: Single cluster with all Hadoop nodes as IBM Spectrum Scale nodes), then run the `gssClientConfig.sh` script from the ESS node with `<path-to-gssclient>/gssClientConfig.sh <gpfs-client-node1,gpfs-client-node2,gpfs-client-node3,...>`. For additional information, see the *Adding IBM Spectrum Scale nodes to the ESS cluster* topic in the *Elastic Storage Server: Quick Deployment Guide*.

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

Configure remote mount access

To configure remote mount access, an existing local IBM Spectrum Scale cluster is required. This cluster must be a different cluster from the ESS-based cluster.

See "Establish an IBM Spectrum Scale cluster on the Hadoop cluster" on page 364 on how to create the local IBM Spectrum Scale cluster.

Note:

- The Hadoop local IBM Spectrum Scale cluster is the accessingCluster.
- The ESS IBM Spectrum Scale cluster is the owningCluster.
- Ensure that the local clusters have password-less ssh to the first node or all the nodes listed in the contact node list. To see the contact node list, run the **mmremotecluster show all** command.

Follow the *Mounting a remote GPFS file system* topic in the *IBM Storage Scale: Administration Guide* to configure remote mount access between the multiple IBM Spectrum Scale clusters. After the remote mount is configured, ensure that the accessing cluster can read/write to the mount point using POSIX.

Note: Remote GPFS file system mount is the preferred deployment model for Cloudera HDP based deployments. For other deployment options such as single scale cluster configuration (**gpfs.storage.type**=*shared*), see "Deploy HDP or IBM Spectrum Scale service on pre-existing IBM Spectrum Scale file system" on page 404.

Install Mpack package

This topic lists the steps to install the management pack.

Note: Before you proceed, ensure that you review the "Support matrix" on page 349 section to download the correct package for your environment.

1. Ensure that the management pack, "IBM Spectrum Scale service (Mpack)" on page 356, is downloaded and unzipped into a local directory on the Ambari server node. This example uses the /root/GPFS_Ambari directory.

   ```
   $ cd /root/GPFS_Ambari
   $ tar -xvzf SpectrumScaleMPack-2.7.X.X.noarch.tar.gz
   ```

2. Stop all services:

   Log in to Ambari. Click **Actions** > **Stop All**.

3. On the Ambari server node, as root Install the Management Pack for IBM Spectrum Scale by running the SpectrumScaleIntegrationPackageInstaller-2.7.X.X.bin executable:

   Note: The Management Pack can only be executed as root.

   - On the Ambari server node, run cd /root/GPFS_Ambari to enter the directory.
   - Run the installer bin to accept the license. The Mpack will be automatically generated and installed on the Ambari server, and the Ambari server will be restarted after the executable completes.

     ```
     $ cd /root/GPFS_Ambari
     $ ./SpectrumScaleIntegrationPackageInstaller-2.7.0.0.bin
     ```

     If you want the installer to automatically accept the license, run the installer bin as follows:

     ```
     $ cd /root/GPFS_Ambari
     $ ./SpectrumScaleIntegrationPackageInstaller-2.7.0.0.bin --accept-licence
     ```

     This will run the Installer non-interactively. This may be useful to users who might want to automate the Mpack installation.

   Ensure that you know the input values before running the installer script.

   Input fields:

   - Ambari server port number: The port that was set up during the Ambari installation.
   - Ambari server IP address: The Ambari server IP address used during Ambari installation. If a node has multiple networks, specifying the IP address guarantees that the address is used.
   - Ambari server username: The Ambari server admin user name.
   - Ambari server password: The Ambari server admin user password.
   - Kerberos settings:
     - Enter kdc principal: The kdc server principal if Kerberos is enabled
     - Enter kdc password: The kdc password if Kerberos is enabled

     The Mpack does not save the KDC principal and password. The information is used only for validation checking to ensure that one can connect and authenticate with the Kerberos server. If the validation fails, the user is notified using a warning message and the Mpack continues to setup the add Scale service option. The failure does not affect the adding of the Scale service option.

   Note: This script automatically restarts the Ambari server.

   To complete this installation, "Deploy the IBM Spectrum Scale service" on page 367 in Ambari GUI.

Deploy the IBM Spectrum Scale service

This section lists the steps to add and deploy the IBM Spectrum Scale™ service through Ambari.

These steps are based on the architecture stated in the "Deployment model" on page 6 section.

- Before you proceed, ensure that you review the Preparing the environment section.
- Review the "Limitations" on page 457 section.
- The IBM Spectrum Scale cluster running the application requires to have consistent uid/gid configuration. Ensure all the user ID and group ID and Hadoop service user ID and group ID are the same across all the IBM Spectrum Scale nodes on the local Hadoop cluster. The owning cluster of the remote mount does not require uid/gid setup. For more information, see "User and group ids" on page 355.

  The user id anonymous is required. See "Create the anonymous user id" on page 352.
- If configured as non-root, see "Restricting root access" on page 450 section.
- For cluster with IBM Spectrum Scale installed:
  - Ensure that IBM Spectrum Scale is active and mounted.

    ```
    /usr/lpp/mmfs/bin/mmgetstate -a
    /usr/lpp/mmfs/bin/mmlsmount all or
    /usr/lpp/mmfs/bin/mmmount <fs-name> -a
    ```
  - Ensure the local Hadoop IBM Spectrum Scale cluster set the IBM Spectrum Scale Master node (which is the Ambari server node) as a quorum node.
- Secondary NameNode:
  - The Secondary NameNode in native HDFS is not needed for HDFS Transparency because the HDFS Transparency NameNode is stateless and does not maintain FSImage-like or EditLog information.
  - The Secondary NameNode should not be shown in the HDFS service GUI when the IBM Spectrum Scale service is integrated.
- Ambari uses the mount points it finds on the Ambari server to set up the services during deployment. If you do not want to use the mount points, ensure that you unmount the mount points on the Ambari server node before starting Ambari or find and change the mount point in the configuration setting for each service that uses it. See **Create HDP cluster** > **Installing, Configuring, and Deploying a Cluster** > **Customize Services** > **Directories** section.
- For HDP 3.X using Ambari 2.7.X, Ambari will add directories in addition to the default `/hadoop/hdfs` directory path.

  Ensure that you review the HDFS NameNode and DataNode directories and Yarn local directories and other directories listed in the Customize Services Directories to ensure that only the required directories are listed.

Log in to Apache Ambari

Log back into Ambari to add the IBM Spectrum Scale service.

Note:

- All services should be down. If not, ensure to stop the services.
- For cluster with IBM Spectrum Scale file system installed, ensure IBM Spectrum Scale is active and mounted. Ensure that you review the Ambari mount point usage bullet under the "Deploy the IBM Spectrum Scale service" on page 367 section before proceeding.

Click **Services** > **Add service**.

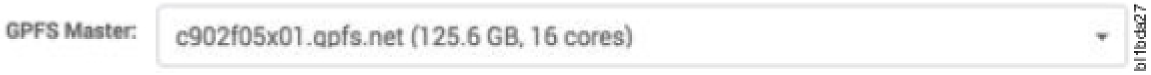
Choose services

On the Add Service Wizard, choose services panel, select the IBM Spectrum Scale package and click **Next**.

Note: The actual version for IBM Spectrum Scale deployment is based on the IBM Spectrum Scale repository. Hence the value shown in the Choose Services panel will not correspond to your actual IBM Spectrum Scale version.

Assign masters

Colocate the GPFS™ Master component to the same host as the Ambari-server. Click **Next**.


GPFS Master: c902f05x01.gpfs.net (125.6 GB, 16 cores)

Assign slaves and clients

Select the GPFS Node components check box on ALL hosts on the **Assign Slaves and Agents** page. Click **Next**.

Note:

- For client-only nodes where you do not want IBM Spectrum Scale, do not select the GPFS Node option.
- Review the GPFS Node column for the NameNode and DataNodes hosts that are part of the HDFS cluster.
- Selecting the GPFS Node column for those nodes run the IBM Spectrum Scale and IBM Spectrum Scale HDFS Transparency.
- The GPFS Master node is a GPFS Node which is the Ambari Server node.
- HDFS Transparency DataNode is required to be a Hadoop DataNode, a NodeManager, and a GPFS Node.

Customize services

Configuration fields on both standard and advanced tabs are populated with values that are taken from the IBM Spectrum Scale Hadoop performance tuning guide.

For all setups, the parameters with a lock icon must not be changed after deployment. These include parameters like the cluster name, remote shell, file system name, and max data replicas. Review the IBM Spectrum Scale configuration parameter checklist.

Review whether all the services configuration are correct, especially the ones listed in the "Create HDP cluster" on page 360 > "Installing, configuring, and deploying a cluster" on page 361 > "Customize services" on page 362 > Directories section. When adding the IBM Spectrum Scale services, the service configuration directories might change to include the mount points that are now mounted.

For HDP 3.X using Ambari 2.7.X, Ambari will add directories in addition to the default `/hadoop/hdfs` directory path.

Ensure that you review the HDFS NameNode and DataNode directories and Yarn local directories and other directories listed in the Customize Services Directories to ensure that only the required directories are listed.

Note:

- Assign the metadata disks to the HDFS transparency NameNode running over a GPFS node.
- Assign Yarn ResourceManager on the node running HDFS Transparency NameNode.

Ensure that you check the values are correct for your environment before clicking **Next**.

Standard Tab

Review GPFS Environment Definition

Field	**Value**
GPFS Cluster Name	Name of IBM Spectrum Scale on local Hadoop cluster See the *mmlscluster command* topic in the *IBM Storage Scale: Command and Programming Reference Guide*.
GPFS quorum nodes	Nodes that are IBM Spectrum Scale quorum nodes. See the *mmlscluster command* topic in the *IBM Storage Scale: Command and Programming Reference Guide*.
`gpfs.storage.type`	remote (for ESS remote mount deployment)
`gpfs.remotecluster.autorefresh`	True (for ESS remote mount deployment)
`gpfs.ssh.user`	User id IBM Spectrum Scale configured as. This example uses the root user id.

Review GPFS file system Definition

Field	**Value**
GPFS file system Name	Local name of the remote mounted file system. Only one local name can be configured.
`gpfs.mnt.dir`	Local mount point name. Only one mount point name can be configured.
Verify if the disks are already formatted as NSDs	A value of Yes specifies that the NSDs are to be created only if each disk has not been formatted as a NSD by a previous invocation of the **`mmcrnsd`** command. The default value is Yes. A value of No specifies that the disks are to be created irrespective of their previous state. **Important:** Setting this field to *No* would result in data stored over those NSDs being erased during the service deployment process. Therefore, set this field to *No* only when intended.

For example, the fields would then be set as follows based on the "Configure remote mount access" on page 365 configuration settings:

gpfs.storage.type:	**Remote**
GPFS file system Name:	`essfs`
`gpfs.mnt.dir:`	`/essfs`

Note:

- In Ambari, the IBM Spectrum Scale configuration value for **`gpfs.storage.type`** is set as remote. However, the **`gpfs.storage.type`** value that is seen in the `/var/mmfs/hadoop/etc/hadoop/gpfs-site.xml` is set as shared. Ensure that you update only through Ambari.
- Do not set the GPFS NSD stanza file for ESS or Share mode.

Create Hadoop local cache disks

[Optional] For ESS/shared storage, create the local cache disk for Hadoop usage.

1. Create the Hadoop local cache disk stanza file, hadoop_disk, in the /var/lib/ambari-server/resources directory.

   Hadoop local cache disk stanza file example:

   ```
   # cat /var/lib/ambari-server/resources/hadoop_disk
       DISK|compute001.private.dns.zone:/dev/sdb,/dev/sdc,/dev/sdd,/dev/sde,/dev/sdf,
       /dev/sdg,/dev/sdi,/dev/sdj,/dev/sdk,/dev/sdl,/dev/sdm,/dev/sdn,/dev/sdo,/dev/sdp
       DISK|compute002.private.dns.zone:/dev/sdb,/dev/sdc,/dev/sdd,/dev/sde,/dev/sdf,
       /dev/sdg,/dev/sdi,/dev/sdj,/dev/sdk,/dev/sdl,/dev/sdm,/dev/sdn,/dev/sdo,/dev/sdp
       DISK|compute003.private.dns.zone:/dev/sdb,/dev/sdc,/dev/sdd,/dev/sde,/dev/sdf,
       /dev/sdg,/dev/sdi,/dev/sdj,/dev/sdk,/dev/sdl,/dev/sdm,/dev/sdn,/dev/sdo,/dev/sdp
       DISK|compute005.private.dns.zone:/dev/sdb,/dev/sdc,/dev/sdd,/dev/sde,/dev/sdf,
       /dev/sdg,/dev/sdi,/dev/sdj,/dev/sdk,/dev/sdl,/dev/sdm,/dev/sdn,/dev/sdo,/dev/sdp
       DISK|compute006.private.dns.zone:/dev/sdb,/dev/sdc,/dev/sdd,/dev/sde,/dev/sdf,
       /dev/sdg,/dev/sdi,/dev/sdj,/dev/sdk,/dev/sdl,/dev/sdm,/dev/sdn,/dev/sdo,/dev/sdp
   ```

2. Add the filename, hadoop_disk, to the **Hadoop local cache disk stanza file** field in the **Standard config** tab.

Advanced Tab

Review Advanced gpfs-ambari-server-env Definition

Field	Value
AMBARI_USER_PASSWORD	Ambari user password
GPFS_REPO_URL	There must be no leading or trailing spaces in the repo name. This directory should contain the IBM Spectrum Scale file system rpms and the HDFS Transparency rpm. Only one HDFS Transparency rpm should reside in this directory. For example: http://60.2.0.229/repos/GPFS/x86_64/rhel/5.0.1

Kerberos setting

The Kerberos principal and password can be set through the IBM Spectrum Scale service in Ambari and saved into the Ambari database. After the IBM Spectrum Scale service is deployed, the Kerberos credentials are required only for the Unintegrate Transparency Scale service UI action. Therefore, arbitrary values can be used when setting up the Kerberos principal and password during deployment. However, valid credentials are required to be saved before executing the Unintegrate Transparency Scale action.

KDC_Principal	Kerberos Principal
KDC_PRINCIPAL_PASSWORD	Kerberos Principal password

If Kerberos is disabled, ignore the **KDC_PRINCIPAL** and **KDC_PASSWORD** fields under the **Customize Services** panel.

If Kerberos is already enabled, then enter the **KDC_PRINCIPAL** and **KDC_PASSWORD** fields under the **Customize Services** panel.

In a Kerberos environment, verify all the configuration information in the Customize Services panel before clicking **NEXT** to go to configure the Configure Identities panel.

Note: Under Advanced gpfs-env, the IBM Spectrum Scale version and HDFS Transparency version will be set automatically once deployment is completed. These fields are set by Ambari whenever HDFS service is restarted. All the IBM Spectrum Scale nodes must have the same version of HDFS Transparency and IBM Spectrum Scale file system installed.

Review

Review the configuration before installation. Click **Deploy**.

Install, start and test

Selected services will be installed and started. Click **Next**.

Note:

- If deployment fails, fix any issues and click on the **Retry** button.
- If some services failed to start, you can Click **Next** to complete the deployment and start the service up manually on the Ambari dashboard.

Summary > Complete

After the summary page shows a list of accomplished tasks. Choose **Complete** to open up the Ambari dashboard.

If the IBM Spectrum Scale mount point was unmounted on the Ambari server, after the ADD Service deployment completes, the IBM Spectrum Scale mount point is automatically remounted.

Note: After the IBM Spectrum Scale service is integrated, the HDFS scripts are modified so that the HDFS service in Ambari will operate on the IBM Spectrum Scale HDFS Transparency components instead of native HDFS.

Post setup

This section lists the steps to be performed post setup.

1. Log in to Ambari GUI.
2. On a separate console, check the `/usr/lpp/mmfs/bin/mmlsfs -r` replication value and update the HDFS config panel **`dfs.replication`** field in Ambari if the value does not match the **`mmlsfs`** command output. For more information, see the *mmlsfs command* topic in the *IBM Storage Scale: Command and Programming Reference Guide*.

   Note: For IBM® ESS file system, if the file system replication is set to 1, the HDFS **`dfs.replication`** field should be set as *1* in the HDFS config panel in Ambari to ensure that the services do not get misleading error message for not having enough space. For example: spark2 thrift server down due to do not have enough space error.
3. IBM Spectrum Scale service is up with Restart Required icon.

   Restart IBM Spectrum Scale service.

   Note: From Mpack 2.7.0.3, the Restart Required icon will not appear after the IBM Spectrum Scale service is deployed.

   Run service check for IBM Spectrum Scale service.
4. HDFS Transparency is now integrated into HDFS service.
5. Start all other services from Ambari. Click **Ambari GUI** > **Services** > **Start All**. If some services did not start properly, start them by going to the host dashboard, and restarting each service individually.

   Note: HDFS will get alerts on the NameNodes.

   a. HDFS Transparency does not perform the checkpointing because IBM Spectrum Scale is stateless. Disable the alert as NameNode checkpoint is not relevant. From **HDFS panel** > **Alert** > **NameNode Last Checkpoint** > **State:Enabled** > **Confirmation panel** > **Confirm Disable**.

   b. Disable the NameNode Blocks Health as this value is not relevant when IBM Spectrum Scale is integrated. From **HDFS panel** > **Alert** > **NameNode Blocks Health** > **State:Enabled** > **Confirmation panel** > **Confirm Disable**.

c. Disable the NameNode HDFS Pending Deletion Blocks as this value is not relevant when IBM Spectrum Scale is integrated. From **HDFS panel** > **Alert** > **NameNode** > **HDFS Pending Deletion Blocks** > **State:Enabled** > **Confirmation panel** > **Confirm Disable**.

Important:

- Restart all affected components with Stale Configs when the dashboard displays the request.
- If any configuration in the `gpfs-site` is changed in the IBM Spectrum Scale dashboard in Ambari, a restart required alert is displayed for the IBM Spectrum Scale service and the HDFS service. Check your environment to ensure that the changes made are in effect.
- IBM Spectrum Scale service must be restarted and only then can the HDFS service be restarted.

Verifying installation

After HDP with IBM Spectrum Scale service is deployed, verify the installation setup.

Note: To run the IBM Spectrum Scale commands, add the `/usr/lpp/mmfs/bin` directory to the environment PATH.

1. Verify all uid/gid values for user to ensure that they are consistent across all IBM Spectrum Scale nodes. Check by using **`mmdsh -N all id<user-name>`** to see whether the UID is consistent across all nodes.
2. Check the IBM Spectrum Scale installed packages on all nodes by using **`rpm -qa | grep gpfs`** to verify that all base IBM Spectrum Scale packages have been installed.
3. As user, ambari-qa, check the user id and access to the file system.

```
HDFS commands:
$hadoop fs -ls /user

POSIX commands:
$pwd; ls -ltr

#Create a file with entry
$ echo "My test" > mytest
$ cat mytest

HDFS commands:
$ hadoop fs -cat mytest

POSIX commands:
$ rm mytest
$ls -ltr
```

4. Run wordcount as user.

   a. Create a mywordcountfile.

   b. Use the mywordcountfile file to be used as input to the wordcount program.

```
$ yarn jar
/usr/hdp/3.0.0.0-1634/hadoop-mapreduce/ hadoop-mapreduce-examples-3.1.0.3.0.0.0-1634.jar
wordcount mywordcountfile wc_output
```

   c. Check the output in directory

```
$ hadoop fs -ls wc_output
```

5. Run teragen/terasort. See Hortonworks Smoke Test Mapreduce.

Upgrading and uninstallation

Upgrading HDP overview

When IBM Spectrum Scale service is integrated with HDP, there is a specific set of steps that you need to perform to finish the update.

There are four steps for upgrading HDP and IBM Spectrum Scale service stack:

1. Ambari upgrade
2. IBM Spectrum Scale Mpack upgrade
3. HDFS Transparency upgrade
4. HDP upgrade

The upgrading process is different depending on the HDP and IBM Spectrum Scale stack versions and the NameNode HA enablement. Ensure that you check the "Support matrix" on page 349 and follow the correct upgrade procedure.

Note:

- The HDP stack comprises Ambari and HDP.
- The Mpack stack comprises the IBM Spectrum Scale service and HDFS Transparency.

Upgrade procedures

From Mpack 2.7.0.7, a new set of procedure is introduced. Therefore, there are two upgrade paths depending on your environment:

Upgrade path	Environment
Upgrade path 1	HDP 3.1.x HA enabled environment
Upgrade path 2	HDP 3.1.x Non-HA environment HDP 3.0.x environment

Upgrading HDP 3.1.x HA environment

Starting from Mpack 2.7.0.7 a new procedure is introduced for the NameNode HA environment so that the Unintegrate Transparency action is not executed. The Hadoop cluster always uses HDFS Transparency protocol and IBM Spectrum Scale file system, and no longer requires to revert to native HDFS. For reverting to native HDFS, you need to set the Kerberos admin principal information in the IBM Spectrum Scale config panel (if Kerberos is enabled) and also need to bring back the same journal nodes before the IBM Spectrum Scale service was integrated.

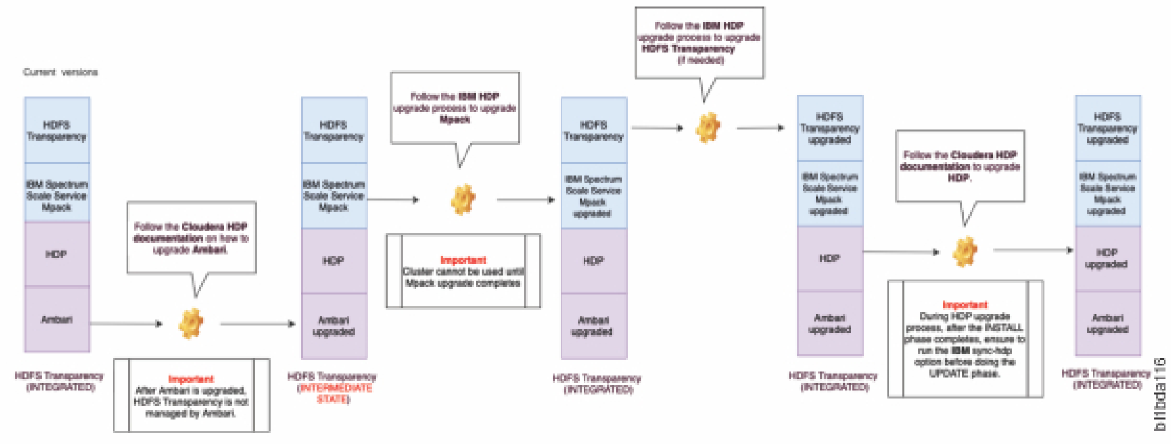

Current versions
HDFS Transparency
IBM Spectrum Scale Service Mpack
HDP
Ambari
HDFS Transparency (INTEGRATED)
Follow the Cloudera HDP documentation on how to upgrade Ambari.
Important
After Ambari is upgraded, HDFS Transparency is not managed by Ambari.
HDFS Transparency
IBM Spectrum Scale Service Mpack
HDP
Ambari upgraded
HDFS Transparency (INTERMEDIATE STATE)
Follow the IBM HDP upgrade process to upgrade Mpack
Important
Cluster cannot be used until Mpack upgrade completes
HDFS Transparency
IBM Spectrum Scale Service Mpack upgraded
HDP
Ambari upgraded
HDFS Transparency (INTEGRATED)
Follow the IBM HDP upgrade process to upgrade HDFS Transparency (if needed)
HDFS Transparency upgraded
IBM Spectrum Scale Service Mpack upgraded
HDP
Ambari upgraded
HDFS Transparency (INTEGRATED)
Follow the Cloudera HDP documentation to upgrade HDP.
Important
During HDP upgrade process, after the INSTALL phase completes, ensure to run the IBM sync-hdp option before doing the UPDATE phase.
HDFS Transparency upgraded
IBM Spectrum Scale Service Mpack upgraded
HDP upgraded
Ambari upgraded
HDFS Transparency (INTEGRATED)
bl1bda116

Figure 35. High-level HDP 3.1.x HA and Mpack upgrade process

Prerequisites:

1. Current HDP environment is configured with NameNode HA.

2. Current HDP version is 3.1.0.0 or later.
3. Current Ambari version is 2.7.3.0 or later.
4. Current Mpack version is 2.7.0.2 or later.
5. The upgrade from these lower versions are required to go to the supported HDP 3.1.4 or HDP 3.1.5 with Mpack 2.7.0.7 that contains this new capability.

If your environment fulfills all the requirements, then you need to follow the new upgrade procedure introduced with Mpack 2.7.0.7 under the "Upgrading HDP 3.1.x HA and Mpack stack" on page 375 section.

Upgrade paths to Mpack 2.7.0.7

Based on the "Support matrix" on page 349, the following are the upgrade support paths from Mpack level 2.7.0.6 and earlier to go to Mpack 2.7.0.7:

HDP level	Ambari level	Mpack level	Upgrade component levels to	Comments
3.1.5	2.7.5	2.7.0.6	Mpack 2.7.0.7	Only upgrading Mpack
3.1.4	2.7.4	2.7.0.5, 2.7.0.4	Mpack 2.7.0.7	Only upgrading Mpack
3.1.4	2.7.4	2.7.0.5, 2.7.0.4	Ambari 2.7.5, Mpack 2.7.0.7, HDP 3.1.5	Upgrading entire stack
3.1.0	2.7.3	2.7.0.3, 2.7.0.2	Ambari 2.7.4, Mpack 2.7.0.7, HDP 3.1.4	Support only upgrading entire stack
3.1.0	2.7.3	2.7.0.3, 2.7.0.2	Ambari 2.7.5, Mpack 2.7.0.7, HDP 3.1.5	Support only upgrading entire stack

Upgrading HDP 3.1.x non-HA environment

For all non NameNode HA, you cannot directly upgrade the HDP version without first performing the Unintegrate Transparency action. This action must be followed by saving the IBM Spectrum Scale configuration and removing the IBM Spectrum Scale service.

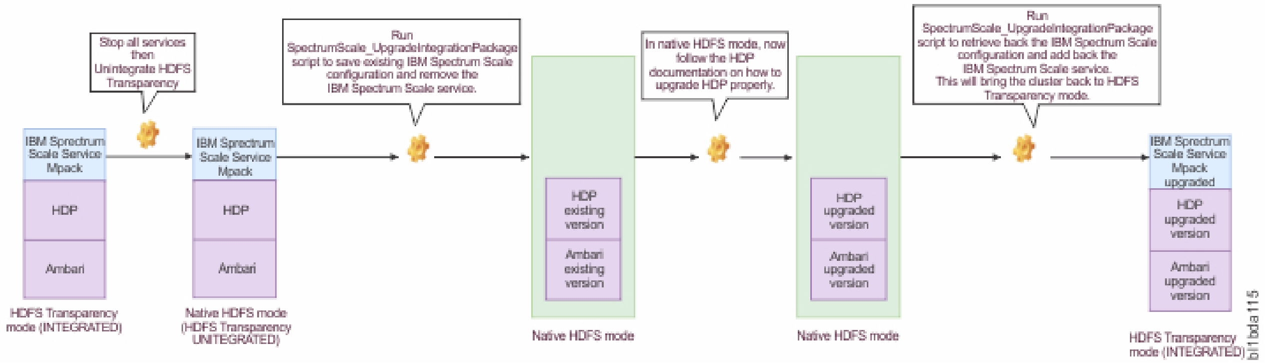

Stop all services then Unintegrate HDFS Transparency
Run SpectrumScale_UpgradeIntegrationPackage script to save existing IBM Spectrum Scale configuration and remove the IBM Spectrum Scale service.
In native HDFS mode, now follow the HDP documentation on how to upgrade HDP properly.
Run SpectrumScale_UpgradeIntegrationPackage script to retrieve back the IBM Spectrum Scale configuration and add back the IBM Spectrum Scale service. This will bring the cluster back to HDFS Transparency mode.
IBM Sprectrum Scale Service Mpack
HDP
Ambari
HDFS Transparency mode (INTEGRATED)
IBM Sprectrum Scale Service Mpack
HDP
Ambari
Native HDFS mode (HDFS Transparency UNITEGRATED)
HDP existing version
Ambari existing version
Native HDFS mode
HDP upgraded version
Ambari upgraded version
Native HDFS mode
IBM Sprectrum Scale Service Mpack upgraded
HDP upgraded version
Ambari upgraded version
HDFS Transparency mode (INTEGRATED)
bl1bda115

Figure 36. High-level HDP 3.1.x non HA and Mpack upgrade process

You need to follow the "Upgrading HDP 3.1.x non-HA" on page 381 to upgrade HDP and IBM Spectrum Scale service stack.

Upgrading HDP 3.0.x environment

For HDP 3.0 in HA or non-HA mode, follow the "Upgrading HDP 3.1.x non-HA environment" on page 374.

Mpack package directories for HDP 3.x and Mpack stack

Ensure that you review the "Support matrix" on page 349 before you proceed.

As the root user, download the target Mpack in a directory on the Ambari server node. For information on downloading the management packs, see the "IBM Spectrum Scale service (Mpack)" on page 356.

Note: The downloaded management pack should be stored and unzipped in a directory that is different from the directory where the currently installed Mpack resides. Ensure that the Mpack directory is preserved for future upgrade use.

For example, the currently installed Mpack is at 2.7.0.3 version and the plan is to upgrade to Mpack 2.7.0.7 version.

In this example, the new (target) Mpack has been downloaded in the `/root/GPFS_Ambari/upgrade_Mpack` directory. The Mpack contains the upgrade script (`SpectrumScale_UpgradeIntegrationPackage`) to upgrade the MPack. The upgrade script must be run from the directory that contains the new Mpack.

Ensure that the current Mpack installable package resides on a separate directory on the Ambari server node. This example uses the `/root/GPFS_Ambari/currently_installed_Mpack` directory. The `SpectrumScaleMPackUninstaller.py` script used as part of this procedure must be run from the directory that contains the existing Mpack.

Upgrading HDP 3.1.x HA and Mpack stack

This section describes the HDP, Ambari, IBM Spectrum Scale MPack and HDFS Transparency upgrade process for HDP 3.1.x HA environment going to Mpack 2.7.0.7.

In the "Upgrading HDP overview" on page 373, see Figure 35 on page 373 for the upgrade flow.

You must plan a cluster maintenance window and prepare for cluster downtime when you perform this upgrade.

Note:

- Ensure that you check the "Support matrix" on page 349 and also ensure that the target HDP, Ambari and Mpack versions are compatible.
- To see the most recently updated default configuration modifications for IBM Spectrum Scale Mpacks, refer to the "Summary of changes" on page xxxi.
- For HDP upgrade, only the express upgrade is supported.
- Upgrading MPack does not affect the IBM Spectrum Scale file system.

Upgrading Ambari for HDP 3.1.x HA and Mpack stack

Ensure that the instructions in the following sections are followed before proceeding:

- Review "Upgrading HDP 3.1.x HA and Mpack stack" on page 375.
- Review "Support matrix" on page 349.
- Follow "Mpack package directories for HDP 3.x and Mpack stack" on page 375.

Ambari can be upgraded with HDFS Transparency in integrated state when the services are up. This new procedure requires the Mpack to be upgraded to Mpack 2.7.0.7 and later.

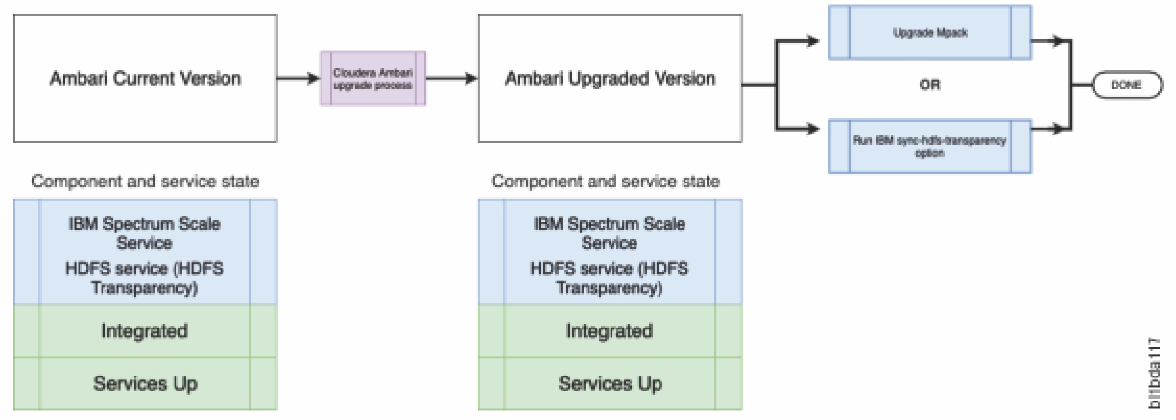

Ambari Current Version
Cloudera Ambari upgrade process
Ambari Upgraded Version
Upgrade Mpack
OR
Run IBM sync-hdfs-transparency option
DONE
Component and service state
IBM Spectrum Scale Service
HDFS service (HDFS Transparency)
Integrated
Services Up
Component and service state
IBM Spectrum Scale Service
HDFS service (HDFS Transparency)
Integrated
Services Up
bl1bda117

Figure 37. Ambari upgrade flow for HDP 3.x HA and Mpack stack

1. Log in to Ambari.
2. Upgrade Ambari by following the Hortonworks documentation process to upgrade section.
3. After Ambari is upgraded, either of the following cases is possible:
   - Case 1: Mpack is already at version 2.7.0.7 or later.

     If the Mpack does not require an upgrade, then after Ambari is upgraded, sync the HDFS Transparency files by running the following command:

     ```
     $ cd <current_Mpack>
     $ ./SpectrumScale_UpgradeIntegrationPackage --sync-hdfs-transparency
     ```
   - Case 2: Mpack needs to be upgraded.

     If Mpack needs to be upgraded, automatically sync the HDFS Transparency files by executing the following steps:

     a. If Mpack needs to be upgraded, follow the "Upgrading IBM Spectrum Scale service (Mpack) for HDP 3.1.x HA and Mpack stack" on page 376.

     b. If HDFS Transparency needs to be upgraded, follow the "Upgrading HDFS Transparency for HDP 3.1.x HA and Mpack stack" on page 379.

     c. Ensure that the required HDFS Transparency is updated before proceeding to HDP upgrade.
4. If upgrading HDP, follow the "Upgrading HDP for HDP 3.1.x HA and Mpack stack" on page 379.

Upgrading IBM Spectrum Scale service (Mpack) for HDP 3.1.x HA and Mpack stack

Ensure that the instructions in the following sections are followed before proceeding:

- Review "Upgrading HDP 3.1.x HA and Mpack stack" on page 375.
- Review "Support matrix" on page 349.
- Follow "Mpack package directories for HDP 3.x and Mpack stack" on page 375.
- Follow "Upgrading Ambari for HDP 3.1.x HA and Mpack stack" on page 375, if Ambari upgrade is required.

Mpack can be upgraded with HDFS Transparency in integrated state but all the services need to be stopped, including the IBM Spectrum Scale (file system) service.

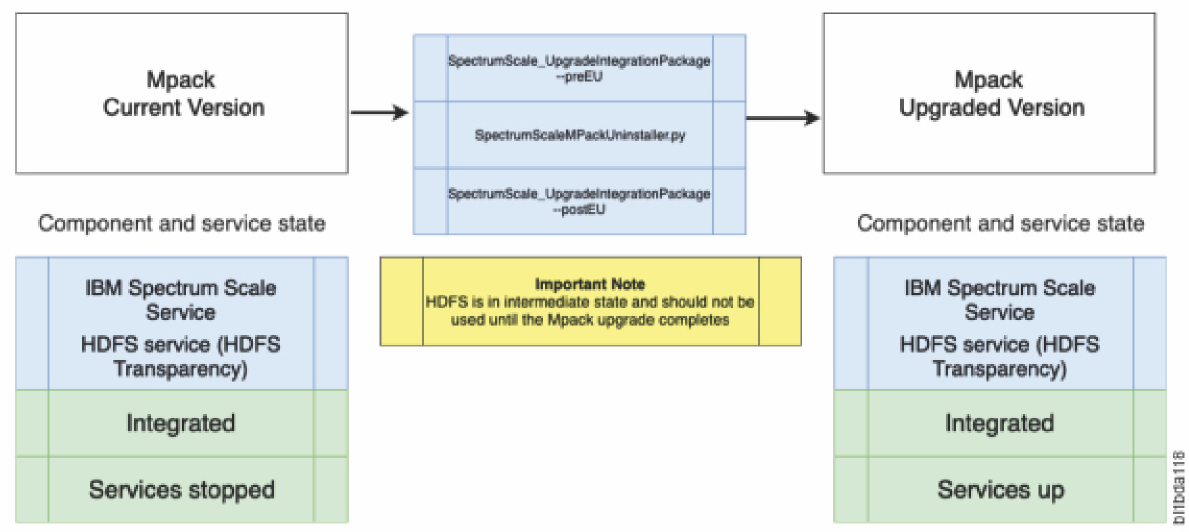

Mpack
Current Version
Component and service state
IBM Spectrum Scale Service
HDFS service (HDFS Transparency)
Integrated
Services stopped
SpectrumScale_UpgradeIntegrationPackage --preEU
SpectrumScaleMPackUninstaller.py
SpectrumScale_UpgradeIntegrationPackage --postEU
Important Note
HDFS is in intermediate state and should not be used until the Mpack upgrade completes
Mpack
Upgraded Version
Component and service state
IBM Spectrum Scale Service
HDFS service (HDFS Transparency)
Integrated
Services up
bl1bda118

Figure 38. IBM Storage Scale Mpack upgrade flow for HDP 3.x HA and Mpack stack

Note: During Mpack upgrade, the cluster cannot be used as it will neither be in native HDFS state nor in HDFS Transparency state until the Mpack upgrade is completed.

1. Stop all the services by clicking **Ambari** > **Actions** > **Stop All**.
2. As the root user on the Ambari server node, from the `/root/GPFS_Ambari/upgrade_Mpack` directory, run the `SpectrumScale_UpgradeIntegrationPackage` script with the **`--preEU`** option.

   The **`--preEU`** option helps with the following:

   a. Saves the existing IBM Spectrum Scale service configuration information into the JSON files in the local directory where the script was executed.

   b. For Mpack 2.7.0.8 and earlier, the **`--preEU`** option removes the IBM Spectrum Scale service from the Ambari server so that you can upgrade to the newer IBM Spectrum Scale Mpack. For Mpack 2.7.0.9 and later, after the **`--preEU`** step, the user needs to log in to the Ambari Server UI and manually delete the IBM Spectrum Scale service.

   Before you proceed, review the following for the upgrade script:

```
$ cd /root/GPFS_Ambari/upgrade_Mpack
$ ./SpectrumScale_UpgradeIntegrationPackage --preEU
************************************************************
***STARTING WITH PRE EXPRESS UPGRADE STEPS***
************************************************************
Enter the Ambari server host name or IP address. If SSL is configured, enter host name,
to verify the SSL certificate. Default=192.0.2.22 :
Enter Ambari server port number. If it is not entered, the installer will take default port
8080 :
Enter the Ambari server username, default=admin :
Enter the Ambari server password :
SSL Enabled (True/False) (Default False):
…
# Note: If Kerberos is enabled, then the KDC principal and password information are
required.
Enter kdc principal:
Enter kdc password:
```

3. On the Ambari server node, run the following command:

```
$ rm /var/lib/ambari-server/resources/mpacks/SpectrumScaleExtension-MPack-
<MPACK-VERSION>/extensions/SpectrumScaleExtension/<MPACK-VERSION>/services/GPFS/package/
scripts/.integration_completed
```

   where, **`<MPACK-VERSION>`** is your currently installed Mpack version (for example, 2.7.0.5).

4. As a root user, on the Ambari server, run the MPack uninstaller script (**SpectrumScaleMPackUninstaller.py**), from the currently installed Mpack directory, to remove the existing MPack from Ambari.

```
$ cd /root/GPFS_Ambari/currently_installed_Mpack
$./SpectrumScaleMPackUninstaller.py
INFO: ***Starting the MPack Uninstaller***Enter Ambari Server Port Number.
If it is not entered, the uninstaller will take default port 8080:
INFO: Taking default port 8080 as Ambari Server Port Number.
Enter Ambari Server IP Address : 192.0.2.22
Enter Ambari Server Username, default=admin :
INFO: Taking default username "admin" as Ambari Server Username.--preEU
Enter Ambari Server Password :
INFO: Verifying Ambari Server Address, Username and Password.
INFO: Verification Successful.
INFO: Spectrum Scale Service is not added to Ambari.
INFO: Spectrum Scale MPack Exists. Removing the MPack.
INFO: Reverting back Spectrum Scale Changes performed while MPack installation.
INFO: Deleted the Spectrum Scale Link Successfully.
INFO: Removing Spectrum Scale MPack.
INFO: Performing Ambari Server Restart.
INFO: Ambari Server Restart Completed Successfully.
INFO: Spectrum Scale MPack Removal Successfully Completed.
```

5. At this point, the HDP cluster is neither in the Native HDFS nor the HDFS Transparency state. Therefore, the HDP cluster should not be used.

   Note: Removing the Mpack does not restore the native HDFS Journal Nodes.

6. As a root user, on the Ambari server node, from the new Mpack directory (`/root/GPFS_Ambari/upgrade_Mpack`), run the `SpectrumScale_UpgradeIntegrationPackage` script with the **`--postEU`** option.

   Note: If Kerberos is enabled, more inputs are required.

   Before you proceed, for the **`--postEU`** option, review the following:

```
$ cd /root/GPFS_Ambari/upgrade_Mpack
$ ./SpectrumScale_UpgradeIntegrationPackage --postEU
Are you sure you want to upgrade the GPFS Ambari integration package (Y/N)? (Default Y):
*************************************************************
***STARTING POST EXPRESS UPGRADE STEPS***
*************************************************************
INFO: Found ambari version '2.7.5.7', proceeding to install Mpack version '2.7.0.7'.
Starting post Express Upgrade steps.
Enter the Ambari server host name or IP address. If SSL is configured,
enter host name, to verify the SSL certificate. Default=192.0.2.22  :
Enter Ambari server port number. If it is not entered, the installer will take default port
8080  :
Enter the Ambari server username, default=admin  :
Enter the Ambari server password  :
SSL Enabled (True/False) (Default False):
Enter kdc principal:
Enter kdc password:
…….
INFO: ***Starting the Spectrum Scale Mpack Installer v2.7.0.7***
…
INFO: Adding Spectrum Scale MPack : ambari-server install-mpack
--mpack=./SpectrumScaleExtension-MPack-2.7.0.7.tar.gz -v
….
Starting to deploy the Spectrum Scale service in Ambari via REST call.
……
Upgrade of the Spectrum Scale Service completed. From the Ambari GUI,
check the IBM Spectrum Scale installation progress through the background operations panel.
….
IMPORTANT:  You need to ensure that the HDFS Transparency package, gpfs.hdfs-protocol-3.X,
is updated in the Spectrum Scale repository.
Then follow the "Upgrade Transparency" service action in the Spectrum Scale service
UI panel to propagate the package to all the GPFS Nodes.
After that is completed, invoke the "Start All" services in Ambari.
$
```

7. The Mpack is now upgraded and the IBM Spectrum Scale service reappears in the Ambari GUI.

   The following steps are for the HDFS Transparency and HDP upgrades.

8. If HDFS Transparency needs to be upgraded, follow the "Upgrading HDFS Transparency for HDP 3.1.x HA and Mpack stack" on page 379.

   Note:

   - Ensure that the IBM Spectrum Scale rpms in the IBM Spectrum Scale service configuration panel, under **Advanced gpfs-ambari-server-env** > **GPFS_REPO_URL**, are of the same version as the IBM Spectrum Scale version currently installed on the system.
   - Do not change the **GPFS_REPO_URL** to point to a new URL during the upgrade phase.
9. Ensure that the required HDFS Transparency is updated before you proceed to HDP upgrade.
10. If HDP needs to be upgraded, follow the instructions in the section "Upgrading HDP for HDP 3.1.x HA and Mpack stack" on page 379.

Upgrading HDFS Transparency for HDP 3.1.x HA and Mpack stack

Ensure that the instructions in the following sections are followed before proceeding:

- Review "Upgrading HDP 3.1.x HA and Mpack stack" on page 375.
- Review "Support matrix" on page 349.
- Follow "Upgrading Ambari for HDP 3.1.x HA and Mpack stack" on page 375, if Ambari upgrade is required.
- Follow "Upgrading IBM Spectrum Scale service (Mpack) for HDP 3.1.x HA and Mpack stack" on page 376, if Mpack upgrade is required.

HDFS Transparency can be upgraded with HDFS Transparency in the integrated state. But you need to stop HDFS Transparency to perform the upgrade.

1. If HDFS Transparency needs to be upgraded, there are a few upgrade options:
   a. To upgrade using the Ambari GUI, follow the "Upgrading HDFS Transparency" on page 385 section.

      Note:

      - Ensure that the IBM Spectrum Scale rpms in the IBM Spectrum Scale service configuration panel under **Advanced gpfs-ambari-server-env** > **GPFS_REPO_URL**, are of the same version as the IBM Spectrum Scale version currently installed on the system.
      - Do not change the **GPFS_REPO_URL** to point to a new URL during the upgrade phase.

   b. Manually perform rpm or yum update on each node.
2. If you are performing the entire HDP and Mpack upgrade process, upgrade HDFS Transparency through the Ambari GUI, as HDFS Transparency is to be upgraded all at once and the services are required to be down. See Note.
3. If you are just upgrading HDFS Transparency and do not want to have down time and are not upgrading the other services and components in the upgrade flow, upgrade using the Rolling upgrade process.

Upgrading HDP for HDP 3.1.x HA and Mpack stack

Ensure that the instructions in the following sections are followed before proceeding:

- Review "Upgrading HDP 3.1.x HA and Mpack stack" on page 375.
- Review "Support matrix" on page 349.
- Follow "Mpack package directories for HDP 3.x and Mpack stack" on page 375.
- Follow "Upgrading Ambari for HDP 3.1.x HA and Mpack stack" on page 375, if Ambari upgrade is required.
- Follow "Upgrading IBM Spectrum Scale service (Mpack) for HDP 3.1.x HA and Mpack stack" on page 376, if required.
- Follow "Upgrading HDFS Transparency for HDP 3.1.x HA and Mpack stack" on page 379, if required.

HDP can be upgraded when the services are up and HDFS Transparency is in the integrated state.

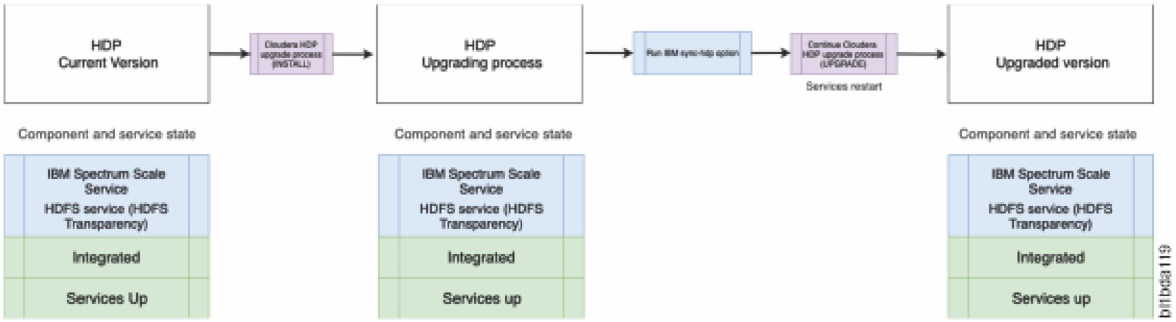

HDP
Current Version
Cloudera HDP upgrade process (INSTALL)
HDP
Upgrading process
Run IBM sync-hdp option
Continue Cloudera HDP upgrade process (UPGRADE)
Services restart
HDP
Upgraded version
Component and service state
IBM Spectrum Scale Service
HDFS service (HDFS Transparency)
Integrated
Services Up
Component and service state
IBM Spectrum Scale Service
HDFS service (HDFS Transparency)
Integrated
Services up
Component and service state
IBM Spectrum Scale Service
HDFS service (HDFS Transparency)
Integrated
Services up
bl1bda119

Figure 39. HDP upgrade flow for HDP 3.x HA and Mpack stack

1. Start All services and perform the service checks. This is needed by HDP Upgrade as a prerequisite.
2. Upgrade HDP by following the Hortonworks HDP upgrade process documentation.

   The HDP Upgrade process happens in two phases:

   a. In the Install phase, the newer version of the binaries corresponding to the HDP services gets installed.

   b. In the Upgrade phase, the services are restarted one by one.
3. After the Install phase is finished and before the Upgrade phase begins, the latest mapreduce framework file needs to be copied to the IBM Spectrum Scale file system. Run the following commands on the Ambari server as a root user:

   a. Create the following IBM Spectrum Scale directory to store the latest mapreduce framework file (`mapreduce.tar.gz`):

   ```
   # mkdir -p <Spectrum Scale filesystem mount directory>/<HDFS Transparency data
   directory>/hdp/apps/<Target HDP version>/mapreduce
   ```

   b. Copy the latest mapreduce framework file from HDP Hadoop path to the corresponding IBM Spectrum Scale path:

   ```
   # cp -p /usr/hdp/<Target HDP version>/hadoop/mapreduce.tar.gz     <Spectrum Scale
   filesystem mount directory>/<HDFS Transparency data directory>/hdp/apps/<Target HDP
   version>/mapreduce/mapreduce.tar.gz
   ```

   where,

   - **`<Target HDP version>`** is the target HDP version you are upgrading to. This can be found by running the **`hdp-select versions`** command. For example:

     ```
     # hdp-select versions
     3.1.4.72-2
     3.1.4.95-3
     ```

     The higher version from the command output is your target HDP version. In the above example, it is *3.1.4.95-3*.
   - **`<Spectrum Scale filesystem mount directory>`** is same as the value of the **`gpfs.mnt.dir`** parameter from `/var/mmfs/hadoop/etc/hadoop/gpfs-site.xml`.
   - **`< HDFS Transparency data directory>`** is same as the value of the **`gpfs.data.dir`** parameter from `/var/mmfs/hadoop/etc/hadoop/gpfs-site.xml`.

   For example:

   ```
   # mkdir -p /ibm/gpfs1/datadir1/hdp/apps/3.1.4.95-3/mapreduce
   # cp -p /usr/hdp/3.1.4.95-3/hadoop/mapreduce.tar.gz /ibm/gpfs1/datadir1/hdp/apps/
   3.1.4.95-3/mapreduce/mapreduce.tar.gz
   ```

Note: If you have multiple IBM Spectrum Scale file systems configured, repeat steps “3.a” on page 380 and “3.b” on page 380 for each file system mount point defined in `/var/mmfs/hadoop/etc/hadoop/gpfs-site.xml`.

4. Perform the Upgrade phase for the HDP upgrade process as specified in the Cloudera HDP upgrade documentation.

Post update process for HDP 3.x and Mpack stack

1. Restart all the components displaying the restart icon.
2. If the **`Start All`** fails, try to individually start each service. Ensure that you manually first start the services in the Ambari order. For more information, see “Manually starting services in Ambari” on page 397.
3. The `NameNode Last Checkpoint` alert can be ignored and can be disabled.
4. If the HBase master failed to start with the `FileAlreadyExistsException` error, restart HDFS and then restart the HBase master.

Upgrading HDP 3.1.x non-HA

This section describes the HDP and IBM Spectrum Scale MPack upgrade process for HDP 3.1.x non-HA and HDP 3.0.x and earlier.

In the “Upgrading HDP overview” on page 373 section, see Figure 36 on page 374 for the flow.

You must plan a cluster maintenance window and prepare for cluster downtime when you upgrade the IBM Spectrum Scale MPack.

Note:

- You must perform the Mpack upgrade only if the target Mpack version is supported on your HDP level. Ensure that you check the support matrix and verify whether the Mpack version is supported with your HDP level.
- To see the default configuration modifications under IBM Spectrum Scale Mpacks, refer to the Big data and analytics section under the Big Data and Analytics - summary of changes.
- For HDP upgrade, only express upgrade is supported.
- The cluster must be at management pack version 2.7.0.0 or later.
- Upgrading MPack does not affect the IBM Spectrum Scale file system.
- Ensure that the anonymous user id is created and have the same uid/gid in your cluster before upgrading. From Mpack 2.4.2.6, having an anonymous user id is mandatory. For more information, see “Create the anonymous user id” on page 352.
- Before you proceed with the upgrade process in a Kerberized environment, you need to set the KDC_PRINCIPAL and KDC_PRINCIPAL_PASSWORD values in the **IBM Spectrum Scale services** > **Configs** > **Advanced section** and save the configuration. If the environment is Kerberized, the unintegrate HDFS Transparency service action requires the KDC_PRINCIPAL and KDC_PRINCIPAL_PASSWORD values to be configured in advance.
- If you are planning to migrate from a Mpack version 2.7.0.3 or earlier to Mpack version 2.7.0.4 or later, a workaround solution is required. For information, see Upgrade failures from Mpack 2.7.0.3 or earlier to Mpack 2.7.0.4 - 2.7.0.6.

Procedure

1. As the root user, download a management pack at a higher PTF version than the version of IBM Spectrum Scale service installed on your system, onto a directory on the Ambari server node. For information on downloading the management packs, see “IBM Spectrum Scale service (Mpack)” on page 356.

   Note: The downloaded management pack should be stored and unzipped in a directory different than the currently installed version of the Mpack.

In this example, the downloaded management pack has been downloaded in the `/root/GPFS_Ambari/upgrade_Mpack` directory. The management pack contains the upgrade script to upgrade the MPack.

For example, if the currently installed Mpack is at 2.7.0.0 version then plan to upgrade to Mpack 2.7.0.1 version.

The `SpectrumScale_UpgradeIntegrationPackage` script used for upgrade and migration is run from the `/root/GPFS_Ambari/upgrade_Mpack` directory.

Ensure that the current Mpack installable package resides on a separate directory on the Ambari server node. This example uses the `/root/GPFS_Ambari/currently_installed_Mpack` directory.

The `SpectrumScaleMPackUninstaller.py` script used as part of this procedure would have to be run from the `/root/GPFS_Ambari/currently_installed_Mpack` directory.

2. Log in to Ambari.
3. Stop all the services. Click **Ambari** > **Actions** > **Stop All**[1].

   [1] For FPO cluster, do not run STOP ALL from the Ambari GUI. Refer to the Limitations > General sections on how to properly stop IBM Spectrum Scale.

   Note: Ensure that the IBM Spectrum Scale file system is not being accessed using either HDFS or POSIX so that it can be unmounted and stopped properly. For more information, see Why did the IBM Spectrum Scale service did not stop or restart properly? in the General Problem determination section.
4. After all the services have stopped, unintegrate the transparency.

   Note: If you run the unintegrate HDFS Transparency more than once consecutively, unpredictable errors will occur and would cause the cluster to be in an unusable state. In such cases, contact scale@us.ibm.com.

   To unintegrate the transparency, run the following steps:

   a. Click **Spectrum Scale** > **Service Actions** > **Unintegrate Transparency**.
   b. On the Ambari server node, run the **`ambari-server restart`** command to restart the Ambari server.

      Note: Do not start any services.
5. If the IBM Spectrum Scale service is not already stopped, stop the IBM Spectrum Scale service by clicking **Ambari** > **Spectrum Scale** > **Service Actions** > **Stop**.
6. As the root user on the Ambari server node, from the `/root/GPFS_Ambari/upgrade_Mpack` directory, run the **`SpectrumScale_UpgradeIntegrationPackage`** script with the `preEU` option.

   The **`--preEU`** option saves the existing IBM Spectrum Scale service information into JSON files in the local directory where the script was run. It also removes the IBM Spectrum Scale service from the Ambari cluster so that the BI cluster can be properly migrated. This does not affect the IBM Spectrum Scale file system.

   Note: If you are migrating from Mpack version 2.7.0.3 or earlier to Mpack version 2.7.0.4 or later, run the **`SpectrumScale_UpgradeIntegrationPackage`** script with the **`preEU`** option command from the `currently_installed_Mpack` instead. Then copy the generated files specified in Upgrade failures from Mpack 2.7.0.3 or earlier to Mpack 2.7.0.4 - 2.7.0.6. to the `upgrade_Mpack` directory.

   Before you proceed, review the following questions for the upgrade script and have the information for your environment handy. If Kerberos is enabled, more inputs are required.

   ```
   $ cd /root/GPFS_Ambari/upgrade_Mpack
   $ ./SpectrumScale_UpgradeIntegrationPackage --preEU
   Are you sure you want to upgrade the GPFS Ambari integration package (Y/N)? (Default Y):
   ***********************************************************
   ***STARTING WITH SPECTRUM SCALE EXPRESS UPGRADE PRE STEPS***
   ***********************************************************
   Enter the Ambari server User:(Default admin ):
   ```

```
Enter the password for the Ambari server.
Password:
Retype password:
SSL Enabled (True/False) (Default False):
Enter the Ambari server Port. (Default 8080):
...
# Note: If Kerberos is enabled, then the KDC principal and password information are
required.
Kerberos is Enabled. Proceeding with Configuration
Enter kdc principal:
Enter kdc password:
...
```

7. As a root user on the Ambari server, run the MPack uninstaller script, `SpectrumScaleMPackUninstaller.py`, from the currently installed Mpack directory, to remove the existing MPack link in Ambari.

   The removal of the IBM Spectrum Scale service during the `SpectrumScale_UpgradeIntegrationPackage --preEU` does not remove the Mpack link in the Ambari database. After the service is removed, remove the link.

   ```

   $ cd /root/GPFS_Ambari/currently_installed_Mpack
   $./SpectrumScaleMPackUninstaller.py
   INFO: ***Starting the MPack Uninstaller***

   Enter Ambari Server Port Number. If it is not entered, the uninstaller will take
   default port 8080:
   INFO: Taking default port 8080 as Ambari Server Port Number.
   Enter Ambari Server IP Address : 192.0.2.22
   Enter Ambari Server Username, default=admin :
   INFO: Taking default username "admin" as Ambari Server Username.
   Enter Ambari Server Password :
   INFO: Verifying Ambari Server Address, Username and Password.
   INFO: Verification Successful.
   INFO: Spectrum Scale Service is not added to Ambari.
   INFO: Spectrum Scale MPack Exists. Removing the MPack.
   INFO: Reverting back Spectrum Scale Changes performed while MPack installation.
   INFO: Deleted the Spectrum Scale Link Successfully.
   INFO: Removing Spectrum Scale MPack.
   INFO: Performing Ambari Server Restart.
   INFO: Ambari Server Restart Completed Successfully.
   INFO: Spectrum Scale MPack Removal Successfully Completed.
   ```

8. After you are in native HDFS, log in to the Ambari server and perform the following checks:

   a. Check the *Directories* section under "Customize services" on page 362 to ensure that the service field names values do not contain any IBM Spectrum Scale directory paths. If there are any, remove those paths and save the configuration. For example, check values for the following fields:

      ```
      dfs.datanode.data.dir
      dfs.namenode.name.dir
      yarn.nodemanager.log-dirs
      yarn.nodemanager.local-dirs
      ```

   b. IBM Spectrum Scale service does not have journal nodes. However, after it is back in native HDFS, the journal nodes are restored. If you are using Kerberos, the journal nodes are required to have the proper principals configured. If not, you need to create the principals for them after unintegrating HDFS transparency.

9. HDP is now in the native HDFS mode.

   - If you plan to upgrade HDP to a newer level, follow the process defined in the Hortonworks documentation to upgrade the HDP and the Ambari versions that the Mpack level supports.
   - After HDP and Ambari are upgraded, ensure that you stop all the services before you proceed to re-deploy the IBM Spectrum Scale service.

10. Ensure all services have stopped.

11. On the Ambari server node as root, from the `/root/GPFS_Ambari/upgrade_Mpack` directory, run the **`SpectrumScale_UpgradeIntegrationPackage`** script with the **`--postEU`** option in the directory where the **`--preEU`** step was run and where the JSON configurations were stored.

    Note: If you are migrating from Mpack version 2.7.0.3 or earlier to Mpack version 2.7.0.4 or later, ensure that the generated files specified in Upgrade failures from Mpack 2.7.0.3 or earlier to Mpack 2.7.0.4 - 2.7.0.6. are copied to the `upgrade_Mpack` directory before running the **`SpectrumScale_UpgradeIntegrationPackage script --postEU`** command.

    Before you proceed, for the **`--postEU`** option, review the following questions and have the information for your environment handy. If Kerberos is enabled, more inputs are required.

```
$ cd /root/GPFS_Ambari/upgrade_Mpack
$ ./SpectrumScale_UpgradeIntegrationPackage --postEU
Are you sure you want to upgrade the GPFS Ambari integration package (Y/N)?
(Default Y):
*************************************************************
***STARTING WITH SPECTRUM SCALE EXPRESS UPGRADE POST STEPS***
*************************************************************
Starting Post Express Upgrade Steps. Enter Credentials
Enter the Ambari server User:(Default admin ):
Enter the password for the Ambari server.
Password:
Retype password:
SSL Enabled (True/False) (Default False):
Enter the Ambari server Port. (Default 8080):
....
# Accept License
Do you agree to the above license terms? [yes or no]
yes
Installing...
Enter Ambari Server Port Number. If it is not entered, the installer will take
default port 8080 :
INFO: Taking default port 8080 as Ambari Server Port Number.
Enter Ambari Server IP Address :
192.0.2.22
Enter Ambari Server Username, default=admin :
INFO: Taking default username "admin" as Ambari Server Username.
Enter Ambari Server Password :
...
Enter kdc principal:
Enter kdc password:
...
From the Ambari GUI, check the IBM Spectrum Scale installation progress through
the background
operations panel.
Enter Y only when installation of the Spectrum Scale service using REST call
process is completed.
(Default N)Y ** SEE NOTE BELOW **
Waiting for the Spectrum Scale service to be completely installed.
...
Waiting for server start....................
Ambari Server 'start' completed successfully.
*************************************************************
Upgrade of the Spectrum Scale Service completed successfully.
*************************************************************
*********************************************************************************
**************
IMPORTANT: You need to ensure that the HDFS Transparency package, gpfs.hdfs-
protocol-2.7.3.X,
is updated in the Spectrum Scale repository. Then follow the "Upgrade
Transparency" service
action in the Spectrum Scale service UI panel to propagate the package to all
the GPFS Nodes.
After that is completed, invoke the "Start All" services in Ambari.
*********************************************************************************
**************
```

Note: If the Mpack requires a corresponding HDFS Transparency update version, ensure that the process in the "Upgrading HDFS Transparency" on page 385 is done before doing a **Start All** in the next step.

12. Start all the services.

    Click **Ambari** > **Actions** > **Start All**.

    Restart all the components by using the restart icon.

    Note:

    - If the **Start All** fails, try starting each of the services individually. Ensure that the manual starting services in the Ambari order is executed first. For more information, see "Manually starting services in Ambari" on page 397.
    - If the IBM Spectrum Scale service is restarted by using the restart icon, the HDFS service also needs to be restarted.
    - The `NameNode Last Checkpoint` alert can be ignored and can be disabled.
    - If the HBase master failed to start with `FileAlreadyExistsException` error, restart HDFS and then restart the HBase master.

Upgrading HDFS Transparency

You must plan a cluster maintenance window, and prepare for the cluster downtime while upgrading the HDFS Transparency.

You can update HDFS Transparency as follows:

- Using Ambari GUI under IBM Spectrum Scale service, Upgrade Transparency action
- Using yum update on the nodes
- Using rpm update on the nodes

The IBM Spectrum Scale update package and the HDFS Transparency is upgraded separately.

This section describes how to update HDFS Transparency through Ambari.

1. Save the new IBM Spectrum Scale HDFS Transparency package into the existing IBM Spectrum Scale yum repository. Ensure that this IBM Spectrum Scale GPFS yum repository is the same repository as the one specified in the GPFS_REPO_URL. Click **Ambari IBM Spectrum Scale** > **Configs** > **Advanced gpfs-ambari-server-env** > **GPFS_REPO_URL**.

   If the yum repository is different, see GPFS yum repo directory to update the yum repository.

   Remove the old version of the IBM Spectrum Scale HDFS Transparency or save it at another location.

   Note:

   - Ensure that the IBM Spectrum Scale rpms in the IBM Spectrum Scale service configuration panel under **`Advanced gpfs-ambari-server-env`**, **`GPFS_REPO_URL`**, are the same as the IBM Spectrum Scale version currently installed on the system.
   - Do not change the **`GPFS_REPO_URL`** to point to a new URL during the upgrade phase.
   - Ensure that only one HDFS Transparency version is in the **`GPFS_REPO_URL`**.
   - Update only the HDFS Transparency version in the **`GPFS_REPO_URL`** if needed and run **`"createrepo . "`** to update the repolist.

2. Go to the IBM Spectrum Scale yum directory, and rebuild the yum database by running the **`createrepo`** command.

   ```
   $ cd /var/www/html/repos/GPFS/<Scale_version>/gpfs_rpms
   $ createrepo .
   Spawning worker 0 with 2 pkgs
   Spawning worker 1 with 2 pkgs
   Spawning worker 2 with 2 pkgs
   Spawning worker 3 with 2 pkgs
   Workers Finished
   ```

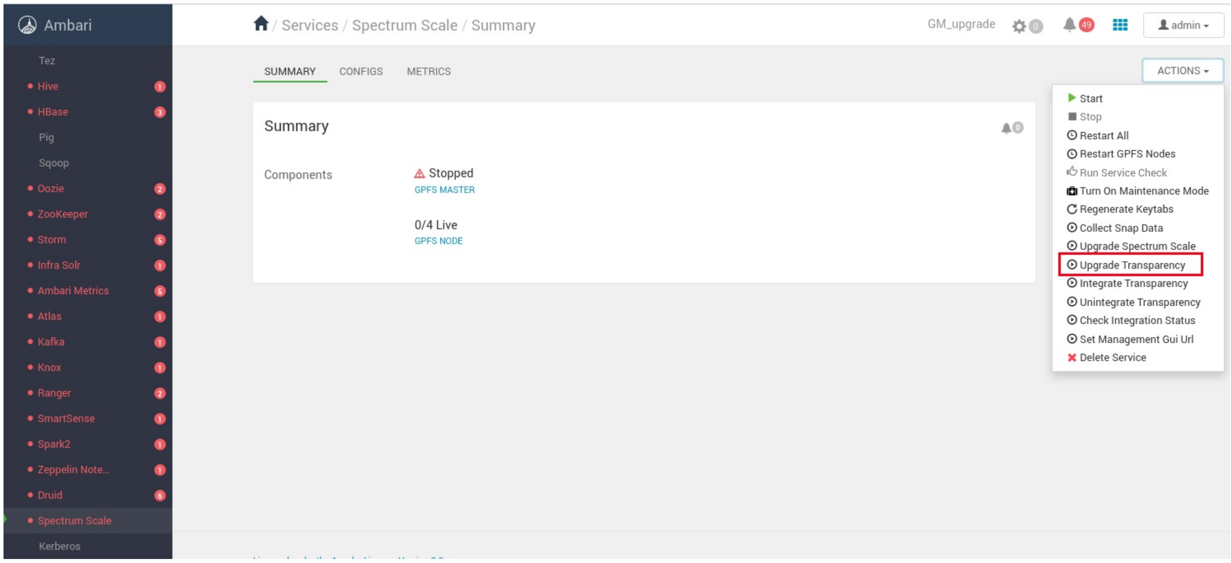
```
Saving Primary metadata
Saving file lists metadata
Saving other metadata
Generating sqlite DBs
Sqlite DBs complete
```

3. From the dashboard, select **Actions** > **Stop All**[1] to stop all the services.

   [1] For FPO cluster, do not run STOP ALL from the Ambari GUI. Refer to the Limitations > General section on how to properly stop IBM Spectrum Scale.

   Note: To upgrade the HDFS Transparency, the IBM Spectrum Scale file system does not need to be stopped. If you do not want to stop the IBM Spectrum Scale file system, do not select **Actions** > **Stop All**. Instead, stop all the services individually by going into each service panel, and clicking **Actions** > **Stop** for all, except the IBM Spectrum Scale service.

4. From the dashboard, click **Spectrum Scale** > **Actions** > **Upgrade Transparency**.



5. Check to see if the correct version of the HDFS Transparency is installed on all the GPFS nodes.

```
$ rpm -qa | grep hdfs-protocol
gpfs.hdfs-protocol-3.0.0-0.x86_64
```

   If the HDFS Transparency version is not correct on a specific node, then manually install the correct version onto that node.

   To manually install the HDFS Transparency on a specific node:

   a. Remove the existing HDFS Transparency package by running the following command:

```
$ yum erase gpfs.hdfs-protocol<Old version>
$ yum clean all
```

   b. `yum install` the new package.

```
$ yum install gpfs.hdfs-protocol-3.0.0.0.<OS>.rpm
```

6. On the dashboard, click **Actions** > **Start All**.

   Note: If you are performing a BI/HDP upgrade or migrate procedure, do not start any services.

7. Check that the HDFS Transparency connector NameNode and DataNode are functioning.

```
$ /usr/lpp/mmfs/hadoop/sbin/mmhadoopctl connector getstate
c902f05x01.gpfs.net: namenode running as process 18150.
c902f05x01.gpfs.net: datanode running as process 22958.
c902f05x02.gpfs.net: datanode running as process 26416.
c902f05x03.gpfs.net: datanode running as process 17275.
c902f05x04.gpfs.net: datanode running as process 15560
```

Upgrading IBM Spectrum Scale file system

You must plan a cluster maintenance window, and prepare for cluster downtime while upgrading the IBM Spectrum Scale file system. Ensure that all the services are stopped, and that no processes are accessing the IBM Spectrum Scale file system.

Follow the *Upgrading* section in the *IBM Storage Scale: Concepts, Planning, and Installation Guide* to upgrade to a newer release version of IBM Spectrum Scale. For additional information on upgrading IBM Spectrum Scale, see *Upgrade on Linux nodes* topic under the Quick reference section in the IBM Spectrum Scale documentation.

To upgrade to a PTF version of IBM Spectrum Scale™, consider the version that you are upgrading from. Consider co-existence and compatibility issues and then you can use the Upgrade IBM Spectrum Scale action in Ambari to update to the IBM Spectrum Scale PTF level.

You can update the IBM Spectrum Scale packages through the Ambari GUI. This function will upgrade the IBM Spectrum Scale packages and build the GPFS portability layer to all the Ambari GPFS nodes. The packages will follow the same rules as specified in Local IBM Spectrum Scale repository.

Note: Only offline upgrade of IBM Spectrum Scale is supported through this Ambari interface.

Upgrading the IBM Spectrum Scale file system package and Upgrading HDFS Transparency are done separately.

You can get the PTF packages from IBM Fix Central and extract the packages as stated in the README file.

You must put all the update packages (PTF) into a yum repository. If the Yum repository is not the existing IBM Spectrum Scale yum repository path specified in Ambari, add the yum repository URL to Ambari IBM Spectrum Scale configuration. For information on updating the yum repository, see GPFS yum repo directory.

Note: Only root Ambari installation can upgrade the IBM Spectrum Scale function through Ambari. Under non-root Ambari installation, IBM Spectrum Scale file system must be upgraded manually as stated in the README file for the specific PTF package in Fix Central. Ensure that all services are stopped, and that no processes are accessing the file system before proceeding to do the upgrade.

1. Go to the IBM Spectrum Scale yum directory and rebuild the yum database by using the **createrepo** command.

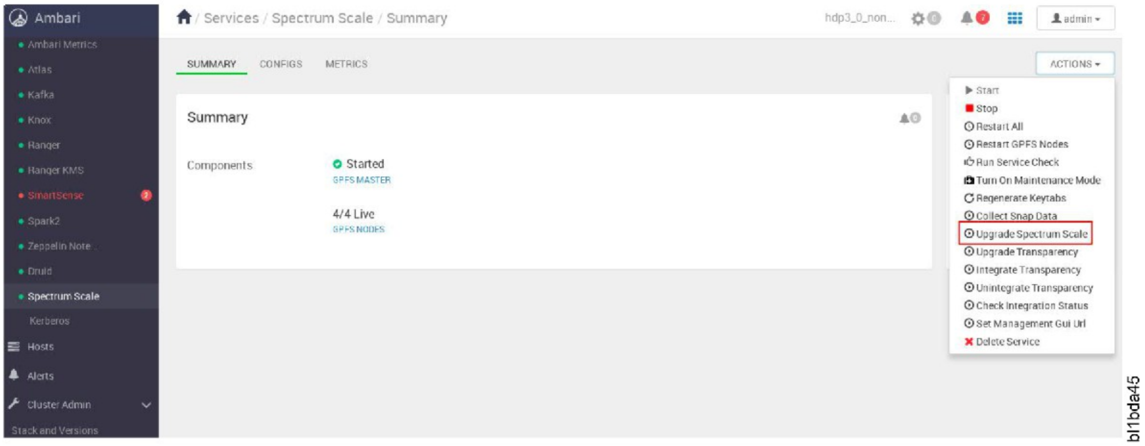
   ```
   $ createrepo .
   Spawning worker 0 with 4 pkgs
   Spawning worker 1 with 4 pkgs
   Spawning worker 2 with 4 pkgs
   Spawning worker 3 with 4 pkgs
   Workers Finished
   Saving Primary metadata
   Saving file lists metadata
   Saving other metadata
   Generating sqlite DBs
   Sqlite DBs complete
   ```

2. From the dashboard, select **Actions** > **Stop All**[1] to stop all services.

   [1] For FPO cluster, do not run STOP ALL from the Ambari GUI. Refer to the Limitations > General section on how to properly stop IBM Spectrum Scale.

3. If you are doing release upgrade, follow the *Upgrading* topic in the *IBM Storage Scale: Concepts, Planning, and Installation Guide*.

   If you are doing PTF upgrade, from the dashboard, select **Spectrum Scale** > **Actions** > **Upgrade Spectrum Scale**.



4. Verify that the selected IBM Spectrum Scale PTF packages are installed on the nodes.

```
$ xdsh c902f05[x01-x04] "rpm -qa | grep gpfs"
```

Note: Check that the gpfs version installed on the nodes are the updated ones.

5. From the dashboard, select **Actions** > **Start All**.

Note: IBM Spectrum Scale starts with the latest PTF packages. Verify that HDFS Transparency NameNode and DataNodes are functioning.

```
$ /usr/lpp/mmfs/hadoop/sbin/mmhadoopctl connector getstate
c902f05x01.gpfs.net: namenode running as process 18150.
c902f05x01.gpfs.net: datanode running as process 22958.
c902f05x04.gpfs.net: datanode running as process 15560.
c902f05x03.gpfs.net: datanode running as process 17275.
c902f05x02.gpfs.net: datanode running as process 26416.
```

Note: After all nodes are upgraded to the new level of IBM Spectrum Scale, use the cluster for a while with the new level installed. Then follow the instructions defined under the topic *Completing the upgrade to a new level of IBM Spectrum Scale to finalize the upgrade* in the *IBM Storage Scale: Concepts, Planning, and Installation Guide*.

HDP 2.6.4 to HDP 3.1.0.0

For migrating from HDP 2.6.4 to HDP 3.1, ensure that you first map out and download all the software prerequisites. You must plan a cluster maintenance window and prepare for cluster downtime during the upgrade.

Table 35. Packages required for upgrading to HDP 3.1

Package	**Version**
HDP	3.1.0.0
Ambari	2.7.3.0
Management Pack (Mpack)	2.7.0.3
HDFS Transparency	3.1.0-1 to latest 3.1.0-x stream

Note:

- Supports upgrading only from HDP 2.6.4 to HDP 3.1.0.0 version and not to a higher HDP 3.1.x version.
- Check the OS and IBM Spectrum Scale version in your current environment to ensure that those versions are compatible with the HDP, Mpack and HDFS Transparency version. See Table 34 on page 350 and FAQ Q2.2 Which Linux distributions are supported by IBM Spectrum Scale.

- Ensure that the other packages in your environment are compatible with the support matrix.
- If you do not have Kerberos enabled before upgrade, then do not enable Kerberos until the entire migration process is completed and IBM Spectrum Scale service is added back. For more information, see Enabling Kerberos.
- Migrating to HDP with IBM Spectrum Scale service does not affect the IBM Spectrum Scale file system.
- Ensure that an anonymous user id is created and has the same uid/gid in your cluster before upgrading.

1. As the root user, download the management pack (as stated in Table 35 on page 388) onto a directory on the Ambari server node. Ensure that the management pack is at a higher PTF version than the version of IBM Spectrum Scale service installed on your system. For information on downloading the management packs, see the topic "IBM Spectrum Scale service (Mpack)" on page 356.

   Note: The downloaded management pack should be stored and unzipped in a different directory than the currently installed version of the Mpack.

   In this example, the downloaded management pack has been downloaded in the `/root/GPFS_Ambari/upgrade_Mpack` directory. The management pack contains the upgrade script to upgrade the Mpack.

   For example, if the currently installed Mpack is at 2.4.2.7 version, plan to upgrade to Mpack 2.7.0.3 version.

   The `SpectrumScale_UpgradeIntegrationPackage` script used for upgrade and migration is run from the `/root/GPFS_Ambari/upgrade_Mpack` directory.

   Ensure that the current Mpack installable package resides on a separate directory on the Ambari server node. This example uses the `/root/GPFS_Ambari/currently_installed_Mpack` directory.

   The `SpectrumScaleMPackUninstaller.py` script used as part of this procedure would have to be run from the `/root/GPFS_Ambari/currently_installed_Mpack` directory.
2. Log in to Ambari.
3. Disable short circuit if enabled.

   For more information, see "Short-circuit read (SSR)" on page 427.
4. Generate an IBM Spectrum Scale snapshot.

   To create a snapshot, ensure that all POSIX and HDFS application and directory/file accesses are stopped.

   Ensure that IBM Spectrum Scale is active.

   If you are using shared file system via remote mount, execute the snapshot command on the Owning cluster.

   Check if `/gpfs.mnt.dir/gpfs.data.dir` is an independent fileset.

   Run **`mmlsfileset <filesystem> -L`** to check the InodeSpace value. If the InodeSpace is *0*, then this is the root fileset. If the InodeSpace is a unique number, then this is an independent fileset.

   If this is an independent fileset, create the snapshot using the following command:

   ```
   mmcrsnapshot fsname snapshotname -j filesetname
   ```

   If this is not an independent fileset or if the **`gpfs.data.dir`** value is blank, then create a global file system snapshot using the following command:

   ```
   mmcrsnapshot fsname snapshotname
   ```
5. Stop all the services. Click **Ambari** > **Actions** > **Stop All**[1].

   [1] - For FPO cluster, do not run STOP ALL from the Ambari GUI. Refer to the General section on how to properly stop IBM Spectrum Scale.

6. After all the services have stopped, unintegrate the transparency.

   Follow the steps in Unintegrating Transparency and ensure that the **ambari-server restart** command is run.

   Note: Do not start the services.

7. If the IBM Spectrum Scale service is not already stopped, click **Ambari** > **Spectrum Scale** > **Service Actions** > **Stop**.

8. On the Ambari server node as root, from the /root/GPFS_Ambari/upgrade_Mpack directory, run the **SpectrumScale_UpgradeIntegrationPackage** script with the **--preEU** option.

   The **--preEU** option saves the existing IBM Spectrum Scale service information into JSON files in the local directory where the script was run. It also removes the IBM Spectrum Scale service from the Ambari cluster so that the cluster can be properly migrated. This does not affect the IBM Spectrum Scale file system.

   Before you proceed, review the following questions for the upgrade script and have the information for your environment handy. If Kerberos is enabled, more inputs are required:

   ```
   Where the upgradeMpack=mpack2703

   [root@c902f10x09 mpack2703]# ./SpectrumScale_UpgradeIntegrationPackage --preEU
   Are you sure you want to upgrade the GPFS Ambari integration package (Y/N)? (Default Y):
   ************************************************************
   ***STARTING WITH PRE EXPRESS UPGRADE STEPS***
   ************************************************************
   Enter the Ambari server username:(Default admin ):
   Enter the password for the Ambari server.
   Password:
   Retype password:
   SSL Enabled (True/False) (Default False):
   Enter the Ambari server Port. (Default 8080):
   http://c902f10x09.gpfs.net:8080
   {
     "href" : "http://c902f10x09.gpfs.net:8080/api/v1/clusters",
   …
   Service STATEtrue
   Successfully completed DELETE call to remove the Spectrum Scale service.
   …
   Starting ambari-server
   Ambari Server running with administrator privileges.
   Organizing resource files at /var/lib/ambari-server/resources...
   Ambari database consistency check started...
   No errors were found.
   Ambari database consistency check finished
   Server PID at: /var/run/ambari-server/ambari-server.pid
   Server out at: /var/log/ambari-server/ambari-server.out
   Server log at: /var/log/ambari-server/ambari-server.log
   Waiting for server start....................
   Ambari Server 'start' completed successfully.
   [root@c902f10x09 upgradeMpack]#
   ```

9. As a root user on the Ambari server, run the Mpack uninstaller script, SpectrumScaleMPackUninstaller.py, from the currently installed Mpack directory, to remove the existing Mpack link in Ambari.

   ```
   Where the currently_installed_Mpack=mpack2427

   [root@c902f10x09 mpack2427]# ./SpectrumScaleMPackUninstaller.py
   INFO: ***Starting the Mpack Uninstaller***

   Enter Ambari Server Port Number. If it is not entered, the uninstaller will take default
   port 8080  :
   INFO: Taking default port 8080 as Ambari Server Port Number.
   Enter Ambari Server IP Address  :   192.0.2.22
   Enter Ambari Server Username, default=admin :
   INFO: Taking default username "admin" as Ambari Server Username.
   Enter Ambari Server Password  :
   INFO: Verifying Ambari Server Address, Username and Password.
   INFO: Verification Successful.
   INFO: Spectrum Scale Service is not added to Ambari.
   INFO: Spectrum Scale MPack Exists. Removing the MPack.
   INFO: Reverting back Spectrum Scale Changes performed while mpack installation.
   INFO: Deleted the Spectrum Scale Link Successfully.
   ```

```
INFO: Removing Spectrum Scale MPack.
INFO: Performing Ambari Server Restart.
INFO: Ambari Server Restart Completed Successfully.
INFO: Spectrum Scale Mpack Removal Successfully Completed.
[root@c902f10x09 mpack2420]#
```

10. Start all services. Click **Ambari** > **Actions** > **Start All**.

    Wait for all the services to start. At this stage, native HDFS is used.

    Check to ensure that the HDFS Transparency is not active, by executing the following commands:

    - On NameNodes: **`ps -eaf | grep namenode | grep -v mmfs`**
    - On DataNodes: **`ps -eaf | grep datanode | grep -v mmfs`**

    Now, HDP is in the native HDFS mode.

11. To upgrade from HDP 2.6.4 to HDP3.1, refer to the Hortonworks migration guide for the following procedures depending on the specific architecture:

    Upgrading to HDP 3.1 on Power

    Upgrading to HDP 3.1 for x86_64

    Note:

    - When migrating to HDP in an x86 environment, ensure that the procedure given in the Switch from IBM Open JDK to Oracle JDK section is completed.
    - Ensure that you properly follow the Hortonworks HDP migration guide. Some steps to take extra notices on:
      - In “Preparing to Upgrade Ambari”:
        - Put the ambari-metrics into maintenance mode.
        - Make a safe copy of the Ambari server configuration file (`/etc/ambari-server/conf/ambari.properties.3`)
      - Ensure that all the services are up and active, all the critical alerts are resolved, and all the service check passed before performing the express upgrade to HDP 2.6.4.
      - In “Upgrade Ambari":

        If you are using the default Postgres database for Ambari server, you need to upgrade Postgres to a supported version. For more information, see Hortonworks documentation.

        Back-up your existing Ambari database before upgrading the Ambari server database. For example, HDP 3.1.0 requires Postgres 9.6 or 10.2.

        Postgres must be upgraded before the Ambari server is upgraded.

        On Power systems, Postgres has dependencies on the `advance-toolchain-at*-runtime`, `advance-toolchain-at*-devel`, and `advance-toolchain-at*-perf` packages. Install the advance-toolchain before upgrading Postgres.

        Note: Remove the current Postgres version and re-install with a new Postgres version on the Power systems to avoid the following error:

        ```
        "Checking cluster versions /usr/bin/pg_ctl-orig: relocation error: /opt/
        at10.0/lib64/power8/libpthread.so.0: symbol __libc_vfork, version
        GLIBC_PRIVATE not defined in file libc.so.6 with link time reference
        could not get pg_ctl version data using “/usr/bin/pg_ctl” --version: No
        such file or directory."
        ```

        This is due to dependencies issues with the advance-toolchain on Power systems.
      - In “Post-upgrade Tasks”:

        Hive Post-upgrade Tasks: If the data directory resides in IBM Spectrum Scale, then the Hive directory changes would need to be run when the IBM Spectrum Scale service is re-integrated.

12. After Ambari and HDP are upgraded, ensure that you stop all the services before you proceed to re-deploy the IBM Spectrum Scale service.

    Click **Ambari** > **Actions** > **Stop All**.

    Wait until all services have stopped. Ensure that the native HDFS has stopped running.

13. HDP 3.1.x supports HDFS Transparency version 3.1.0-x and later. Only HDFS Transparency 3.1.0 stream is supported by HDP.

    Add the HDFS Transparency version as stated in Table 35 on page 388 into the GPFS repo directory.

    Ensure that the older HDFS Transparency version is removed from the repo directory because only one HDFS Transparency rpm can reside in the GPFS repo directory.

    Run **"createrepo . "** to update the repo metadata.

14. Add the IBM Spectrum Scale service.

    On the Ambari server node as root, from the `/root/GPFS_Ambari/upgrade_Mpack` directory, run the **SpectrumScale_UpgradeIntegrationPackage** script with the **--postEU** option in the directory where the **--preEU** step was run and where the JSON configurations were stored.

    Before you proceed, for the **--postEU** option, review the following questions, and have the information for your environment handy. If Kerberos is enabled, more inputs are required.

```
Where the upgradeMpack=mpack2703

[root@c902f10x09 mpack2703]# ./SpectrumScale_UpgradeIntegrationPackage --postEU
Are you sure you want to upgrade the GPFS Ambari integration package (Y/N)? (Default Y):
*************************************************************
***STARTING WITH SPECTRUM SCALE EXPRESS UPGRADE POST STEPS***
*************************************************************
Starting Post Express Upgrade Steps. Enter Credentials
Enter the Ambari server User:(Default admin ):
Enter the password for the Ambari server.
Password:
Retype password:
SSL Enabled (True/False) (Default False):
Enter the Ambari server Port. (Default 8080):
....
# Accept License
Do you agree to the above license terms? [yes or no]
yes
Installing...
Enter Ambari Server Port Number. If it is not entered, the installer will take default port
8080:
INFO: Taking default port 8080 as Ambari Server Port Number.
Enter Ambari Server IP Address : 192.0.2.22
Enter Ambari Server Username, default=admin :
INFO: Taking default username "admin" as Ambari Server Username.
Enter Ambari Server Password :
...
Enter kdc principal:
Enter kdc password:
...
From the Ambari GUI, check the IBM Spectrum Scale installation progress through the
background
operations panel.
Enter Y only when installation of the Spectrum Scale service using REST call process is
completed.
(Default N)Y ** SEE NOTE BELOW **
Waiting for the Spectrum Scale service to be completely installed.
...
Waiting for server start...................
Ambari Server 'start' completed successfully.
*************************************************************
Upgrade of the Spectrum Scale Service completed successfully.
*************************************************************
********************************************************************************************
***
IMPORTANT: You need to ensure that the HDFS Transparency package, gpfs.hdfs-protocol-3.0.x,
is updated in the Spectrum Scale repository. Then follow the "Upgrade Transparency" service
action in the Spectrum Scale service UI panel to propagate the package to all the GPFS
Nodes.
After that is completed, invoke the "Start All" services in Ambari.
********************************************************************************************
```

```
***
```

15. Update the HDFS Transparency package to all the GPFS nodes.

    HDP 3.1.x requires HDFS Transparency version 3.1.0-x. Update the HDFS Transparency package before you start any services.

    Ensure that the HDFS Transparency package, `gpfs.hdfs-protocol-3.1.0.X`, is updated in the IBM Spectrum Scale repository as stated in Step 11.

    From Ambari GUI, go to **Upgrade Transparency service action** in the **Spectrum Scale service** UI window to propagate the new package to all the GPFS Nodes. For more information, see Upgrading Transparency.

    Ensure to check that all the GPFS nodes have the HDFS Transparency upgraded to the correct version by running the following command:

    ```
    mmdsh -N all “rpm -qa | grep gpfs.hdfs-protocol”
    ```

16. Start all services.

    Click **Ambari** > **Actions** > **Start All**.

    Restart all components by using the **Restart** icon.

17. If short circuit was disabled earlier and needs to be enabled, enable it now.

    See “Short-circuit read (SSR)” on page 427.

    Note:

    - If the IBM Spectrum Scale service is restarted by using the restart icon, the HDFS service also needs to be restarted.
    - The `NameNode Last Checkpoint` alert, `NameNode Blocks Health` alert and `NameNode HDFS Pending Deletion Blocks` alert can be ignored and disabled.
    - If the HBase master failed to start with `FileAlreadyExistsException` error, restart HDFS and then restart the HBase master.

HDP to CDP migration

For enterprises that want to migrate from their Hadoop cluster to CDP Private Cloud Base with IBM Spectrum Scale, contact IBM account team/Lab Based Services (LBS) or Cloudera professional services (PS) to help determine the correct approach for your environment.

For Hadoop clusters with IBM Spectrum Scale FPO environment, you must use the side-by-side migration to migrate to CDP Private Cloud Base with IBM Spectrum Scale.

Uninstalling IBM Spectrum Scale Mpack and service

Before you upgrade Mpack on an HDP cluster, see “Preparing the environment” on page 349 to check whether the new Mpack upgrade supports the HDP version installed on the cluster.

This topic lists the steps to uninstall IBM Spectrum Scale Mpack and service.

1. Follow the steps in “Unintegrating HDFS Transparency” on page 433 and ensure that the `ambari-server restart` is run.
2. Uninstalling the management pack and the IBM Spectrum Scale service

   On the Ambari server host, as root, execute the following script:

   ```
   $ python SpectrumScaleMPackUninstaller.py
   ```

   Follow the script and enter the required details.

The SpectrumScaleMPackUninstaller.py script is in the download package along with the Mpack and license bin executables.

The SpectrumScaleMPackUninstaller.py uninstalls the IBM Spectrum Scale Mpack and the IBM Spectrum Scale service from Ambari. It verifies the settings before it uninstalls the Mpack and IBM Spectrum Scale service. If the services are still running, and if the HDFS Transparency is not unintegrated, the SpectrumScaleMPackUninstaller.py script will exit and request user action.

If you face a Kerberos credential error when you are uninstalling the Mpack, see When you are uninstalling the Mpack, the Mpack uninstaller script might throw a Kerberos credential error even when the correct credentials were provided.

Note: Running this script does not remove the IBM Spectrum Scale packages and disks. The IBM Spectrum Scale file system is preserved as is. For the FPO cluster created through Ambari, the mounted local disks /opt/mapred/local* and entries in /etc/fstab are preserved as is.

Configuration

Setting up High Availability [HA]

This section contains information on how to setup high availability to protect against planned and unplanned events.

HDFS NameNode High Availability [HA]

This process sets up a standby NameNode configuration so that failover can happen automatically.

In order to setup HA, IBM Spectrum Scale HDFS Transparency has to be unintegrated and revert back to native HDFS mode.

Follow these steps to configure High Availability option when IBM Spectrum Scale™ service is integrated:

1. Log in to the Ambari GUI.
2. If the Ambari GUI has IBM Spectrum Scale service deployed, and HDFS Transparency is integrated, follow the steps to "Unintegrating HDFS Transparency" on page 433.

   Ensure that you run the **ambari-server restart**.
3. Verify that the IBM Spectrum Scale HDFS Transparency integration state is in unintegrated state. For more information, see Verifying Transparency integration state.
4. From the Ambari dashboard, click the HDFS service.

   Select **Actions** > **Enable NameNode HA** and follow the steps.

   Note: NameNode needs to be deployed onto a GPFS™ Node.
5. If the Ambari GUI has IBM Spectrum Scale service deployed, follow the steps under "Integrating HDFS Transparency" on page 431 to integrate HDFS Transparency.

   Ensure that you run the **ambari-server restart**.

For more information on setting up the NameNode HA, see Hortonworks Configuring NameNode HA.

Yarn Resource Manager HA

This process sets up a Resource Manager configuration so that failover can happen automatically.

This topic describes how to enable ResourceManager High Availability if the IBM Spectrum Scale service is already integrated with HDP.

1. Turn on Maintenance Mode on IBM Spectrum Scale.
2. Click **Ambari Spectrum Scale service** > **Actions** > **Turn On Maintenance Mode**. This will ensure that IBM Spectrum Scale is not stopped while Resource Manager HA is enabled in the next step.

3. Enable Resource Manager HA as per Configuring ResourceManager High Availability.
4. Turn off Maintenance Mode on IBM Spectrum Scale service.
5. Restart HDFS service.

IBM Spectrum Scale configuration parameter checklist

The IBM Spectrum Scale checklist shows the parameters that affect the system, the Standard, and Advanced tabs in the Ambari wizard.

These are the important IBM Spectrum Scale parameters checklists:

Standard tab	**Rule**	**Advanced tab**	**Rule**
Cluster Name		Advanced core-site: `fs.defaultFS`	Ensure that `hdfs://localhost:8020` is used
File system Name		Advanced gpfs-advance: `gpfs.quorum.nodes`	The node number must be odd. **Note:** In SAN storage deployment model with tiebreaker disk, quorum nodes can be even.
File system Mount Point			
NSD stanza file	For more information, see "Preparing for FPO environment" on page 357.		
Policy file	For more information, see Policy File.		
Hadoop local cache disk stanza file	For more information, see "Deploy HDP or IBM Spectrum Scale service on pre-existing IBM Spectrum Scale file system" on page 404.		
Default Metadata Replicas	<= Max Metadata Replicas		
Default Data Replicas	<= Max Data Replicas		
Max Metadata Replicas			
Max Data Replicas			

HDFS and IBM Spectrum Scale file system ACL support

HDFS and IBM Spectrum Scale HDFS Transparency support only POSIX ACL.

Ensure that you check the ACL value on the IBM Spectrum Scale file system and set it to all.

Otherwise, the Hadoop command will fail with `Operation not supported:` but POSIX command can be executed successfully.

To check the ACL value, run:

```
# /usr/lpp/mmfs/bin/mmlsfs <filesystem> -k
```

```
flag       value          description
-----   -----------   ----------------------------
-k         nfs4           ACL semantics in effect
```

To change the ACL value to all, run:

```
# /usr/lpp/mmfs/bin/mmchfs <filesystem> -k all
```

To check the ACL value after modification, run:

```
# /usr/lpp/mmfs/bin/mmlsfs <filesystem> -k
```

```
flag      value             description
-----   ----------     --------------------------
-k         all            ACL semantics in effect
```

Dual-network deployment

This section describes the recommended network configuration for Ambari to manage the Hadoop and IBM Spectrum Scale cluster if more than one network is configured in the environment. It is recommended to map out how Ambari and IBM Spectrum Scale network configurations are able to communicate before deploying HDP or IBM Spectrum Scale in your environment.

For a Hadoop distribution like HDP®, all Hadoop components are managed by Ambari™. IBM Spectrum Scale has GPFS Daemon node name (daemon network) and GPFS Admin node name (admin network) adapter interface names. For more information, see the *GPFS node adapter interface names* topic in the *IBM Storage Scale: Concepts, Planning, and Installation Guide*.

For example:

```
[root@c902f09x05 ~]# mmlscluster
GPFS cluster information
========================
  GPFS cluster name:         SS5022.gpfs.net
…………….
…………….
 Node  Daemon node name          IP address   Admin node name      Designation
-------------------------------------------------------------------------------
   1   c902f09x05-eth4.gpfs.net  128.20.1.26  c902f09x05.gpfs.net  quorum
   2   c902f09x07-eth4.gpfs.net  128.20.1.28  c902f09x07.gpfs.net  quorum
   3   c902f09x08-eth4.gpfs.net  128.20.1.29  c902f09x08.gpfs.net  quorum
   4   c902f09x06-eth4.gpfs.net  128.20.1.27  c902f09x06.gpfs.net
```

In the above example, Daemon node name and IP address are the daemon network used for data traffic in IBM Spectrum Scale. The Admin node name is the admin network used for IBM Spectrum Scale commands (such as **mmlscluster**, **mmgetstate**, etc). The IBM Spectrum Scale Admin node name and Daemon node name could be changed by the **mmchnode** command.

In a dual network environment, there are two networks: Network 1 and Network 2.

The following are the recommended network setup configuration options for the local IBM Spectrum Scale cluster:

1. IBM Spectrum Scale cluster using the same hostname for both Admin node name and Daemon node name.

   Configure Admin node name and Daemon node name on Network 1 so that IBM Spectrum Scale service can be managed from the Ambari GUI. It is also possible to configure Network 2 as the subnets for IBM Spectrum Scale so that all the IBM Spectrum Scale data traffic will go over Network 2. For setting up the cluster, see IBM Spectrum Scale support for Hadoop > Dual network interfaces.
2. IBM Spectrum Scale cluster using different hostnames for both Admin node name and Daemon node name.

a. Define the IBM Spectrum Scale Admin node name (admin network) on Network 1 and the Daemon node name (daemon network) on Network 2 for IBM Spectrum Scale cluster that uses different admin and daemon hosts.

or

b. Define the IBM Spectrum Scale Admin node name (admin network) on Network 2 and the Daemon node name (daemon network) on Network 1 for IBM Spectrum Scale cluster that uses different admin and daemon hosts.

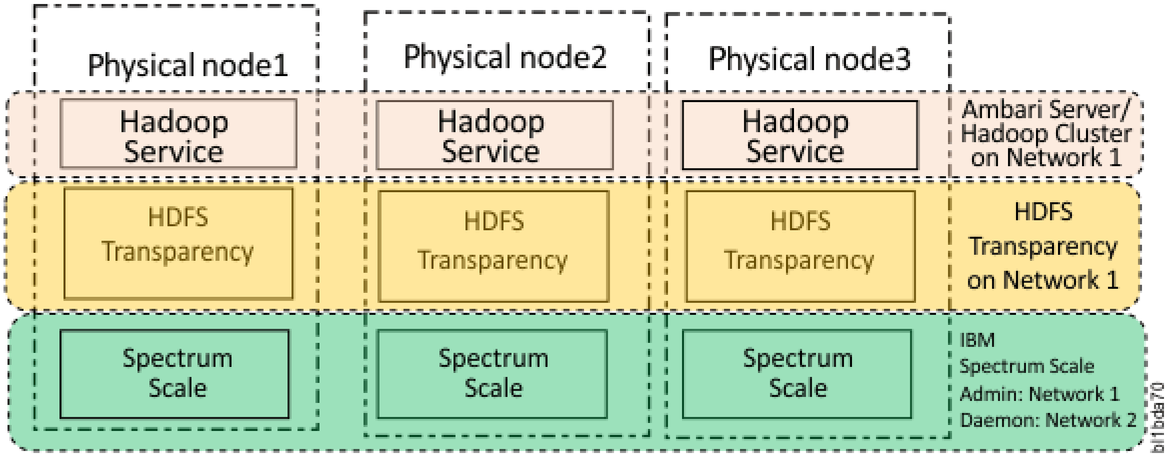


Figure 40. Example of network setup configuration

The above figure shows the 2(a) network setup configuration option where, Ambari and the IBM Spectrum Scale Admin is on Network 1 and the IBM Spectrum Scale daemon is on Network 2.

On setting up the network this way, when the IBM Spectrum Scale service is deployed, it will be able to recognize the network connection between the Ambari and the admin or daemon network so that it can communicate with the IBM Spectrum Scale cluster to manage it properly.

The separation of IBM Spectrum Scale admin and daemon networks offers the following benefits:

- IBM Spectrum Scale uses the admin network for running cluster wide administrative commands, (for example, **`mmgetstate -a`**). When the cluster is busy with heavy I/O, using the same network for admin and daemon can cause administrative commands to run slower.
- IBM Spectrum Scale Admin network requires ssh to be open. The daemon network does not require ssh. This can restrict ssh access to only one selected network for security concerns.

Note:

- IBM Spectrum Scale service configures HDFS Transparency to use the same network interface as Ambari and native HDFS.
- For FPO configuration, the FPO file system will be created, if not pre-existing, during the IBM Spectrum Scale service deployment.
- IBM Spectrum Scale cluster wide administrative commands are run via SSH within the Admin interface. Hence, the GPFS master (which is also the Ambari server node) requires to have root passwordless SSH to all the GPFS Nodes in the cluster. If the IBM Spectrum Scale cluster is non-root (i.e. using sudo-wrapper), passwordless ssh is required for the non-root user. For more information, see “Restricting root access” on page 450. The Daemon interface is used for GPFS data traffic over RPC. For more information, see “Password-less ssh access” on page 53.

Manually starting services in Ambari

If you do not do a **Start All** and plan to start each service individually, the following sequence must be followed:

1. IBM Spectrum Scale service

2. If have HA, then zookeeper
3. HDFS
4. Yarn
5. Mapreduce2

Then other services can be started.

Setting up local repository

Mirror repository server

IBM Storage Scale requires a local repository. Therefore, select a server to act as the mirror repository server. This server requires the installation of the Apache HTTP server or a similar HTTP server.

Every node in the Hadoop cluster must be able to access this repository server. This mirror server can be defined in the DNS, or you can add an entry for the mirror server in `/etc/hosts` on each node of the cluster.

- Create an HTTP server on the mirror repository server, such as Apache httpd. If the Apache httpd is not already installed, install it with the **`yum install httpd`** command. You can start the Apache httpd by running one of the following commands:
  - **`apachectl start`**
  - **`service httpd start`**
- [Optional]: Ensure that the http server starts automatically on reboot by running the following command:
  - **`chkconfig httpd on`**
- Ensure that the firewall settings allow inbound HTTP access from the cluster nodes to the mirror web server.
- On the mirror repository server, create a directory for your repositories, such as `<document root>/repos`. For Apache httpd with document root `/var/www/html`, type the following command:
  - **`mkdir -p /var/www/html/repos`**
- Test your local repository by browsing the web directory:
  - **`http://<yum-server>/repos`**

For example:

```
# rpm -qa | grep httpd
# service httpd start
# service httpd status
Active: active (running)  Check to ensure is active
# systemctl enable httpd
```

Local OS repository

You must create the operating system repository because some of the IBM Storage Scale files, such as rpms have dependencies on all nodes.

1. Create the repository path:

   ```
   mkdir /var/www/html/repos/<rhel_OSlevel>
   ```

2. Synchronize the local directory with the current yum repository:

   ```
   cd /var/www/html/repos/<rhel_OSlevel>
   ```

   Note: Before going to the next step, ensure that you have registered your system. For instructions to register a system, refer to Get Started with Red Hat Subscription Manager. Once the server is subscribed, run the following command: **`subscription-manager repos --enable=<repo_id>`**

3. Run the following command:

   ```
   reposync --gpgcheck -l --repoid=rhel-7-server-rpms --download_path=/var/www/html/repos/
   <rhel_OSlevel>
   ```

4. Create a repository for this node:

   ```
   createrepo -v /var/www/html/repos/<rhel_OSlevel>
   ```

5. Ensure that all the firewalls are disabled or that you have the httpd service port open, because yum uses http to get the packages from the repository.
6. On all nodes in the cluster that require the repositories, create a file in `/etc/yum.repos.d called local_<rhel_OSlevel>.repo`.
7. Copy this file to all nodes. The contents of this file must look like the following:

   ```
   [local_rhel_version]
   name=local_rhel_version
   enabled=1
   baseurl=http://<internal IP that all nodes can reach>/repos/<rhel_OSlevel>
   gpgcheck=0
   ```

8. Run the **`yum repolist`** and **`yum install rpms`** without external connections.

Local IBM Spectrum Scale repository

The following list of rpm packages for IBM Spectrum Scale 4.1.1 and later can help verify the edition of IBM Spectrum Scale:

IBM Spectrum Scale Edition	rpm package list
Data Management available from version 4.2.3 and later.	This edition provides identical functionality as IBM Spectrum Scale Advanced Edition under capacity-based licensing. For more information, see the *Capacity-based licensing* topic in the *IBM Storage Scale: Concepts, Planning, and Installation Guide*.
Standard Edition	<Express Edition rpm list> + gpfs.ext
Advanced Edition	<Standard Edition rpm list> + gpfs.crypto For IBM Spectrum Scale 4.2 release and later, add gpfs.adv to the list above.

Note: From IBM Spectrum Scale 5.0.2, the `gpfs.ext` package is not available.

The following example uses IBM Spectrum Scale 5.0.1.1 version.

1. On the repository web server, create a directory for your IBM Spectrum Scale repos, such as `<document root>/repos/GPFS`. For Apache httpd with document root `/var/www/html`, type the following command:

   ```
   mkdir -p /var/www/html/repos/GPFS/5.0.1.1
   ```

2. Obtain the IBM Spectrum Scale software. If you have already installed IBM Spectrum Scale manually, skip this step. Download the IBM Spectrum Scale package. In this example, IBM Spectrum Scale 5.0.1.1 is downloaded from Fix Central, the package is unzipped, and the installer is extracted.

   For example, As root or a user with sudo privileges, run the installer to get the IBM Spectrum Scale packages into a user-specified directory via the `--dir` option:

   ```
   chmod +x Spectrum_Scale_Advanced-5.0.1.1-x86_64-Linux-install

   ./Spectrum_Scale_Advanced-5.0.1.1-x86_64-Linux-install --silent --dir /var/www/html/repos/
   GPFS/5.0.1.1
   ```

Note: The `--silent` option is used to accept the software license agreement, and the `--dir` option places the IBM Spectrum Scale rpms into the directory `/var/www/html/repos/GPFS/5.0.1.1/gpfs_rpms`. Without specifying the `--dir` option, the default location is `/usr/lpp/mmfs/gpfs_rpms/5.0.1.1/gpfs_rpms`.

3. If the packages are extracted into the IBM Spectrum Scale default directory, `/usr/lpp/mmfs/5.0.1.1/gpfs_rpms`, copy all the IBM Spectrum Scale files that are required for your installation environment into the IBM Spectrum Scale repository path:

```
cd /usr/lpp/mmfs/5.0.1.1/gpfs_rpms

cp -R * /var/www/html/repos/GPFS/5.0.1.1/gpfs_rpms
```

4. The following packages will not be installed by Ambari:

   - gpfs.crypto
   - gpfs.gui
   - gpfs.scalemgmt
   - gpfs.tct

   Ambari requires only the following packages:

   - gpfs.base
   - gpfs.gpl
   - gpfs.docs
   - gpfs.gskit
   - gpfs.msg.en_US
   - gpfs.ext (Not available from IBM Spectrum Scale 5.0.2 version)
   - gpfs.crypto (if Advanced edition is used)
   - gpfs.adv (if IBM Spectrum Scale 4.2 Advanced edition is used)

   The IBM Spectrum Scale repo will not install the protocol and transparent cloud tier (gpfs.tct) packages when installing through Ambari.

5. Copy the HDFS Transparency package to the IBM Spectrum Scale repo path.

   Note: The repo must contain only one HDFS Transparency package. Remove all old transparency packages.

```
cp gpfs.hdfs-protocol-3.0.0-(version)  /var/www/html/repos/GPFS/5.0.1.1/gpfs_rpms
```

6. Check for the IBM Spectrum Scale packages in the `/root/` directory. If the package exists, relocate them to a subdirectory. There are known issues with IBM Spectrum Scale package in the /root that cause the Ambari installation to fail.
7. Create the yum repository:

```
# cd /var/www/html/repos/GPFS/5.0.1.1/gpfs_rpms
# createrepo .
```

8. Access the repository at http://<yum-server>/repos/GPFS/5.0.1.1/gpfs_rpms.

MySQL community edition repository

If you are using the new database option for Hive MetaStore through HDP, HDP will create MySQL community edition repositories on the Hive Metastore host which will require internet access to download.

On a host with internet access, use the repo information to obtain a local copy of the packages to create a local repository.

If you have a local MySQL repo, create the `mysql-community.repo` file to point to the local repo on the Hive Metastore host.

For example:

```
[mysql56-community]
name=MySQL 5.6 Community Server
baseurl=http://<REPO_HOST>/repos/MySQL_community
enabled=1
gpgcheck=0
```

Only the following MySQL packages are required for HDP:

```
mysql-community-libs
mysql-community-common
mysql-community-client
mysql-community-server
```

HDP creates the following MySQL community repos:

HDP Power: Creates 1 repo file

`mysql-community.repo:`

```

# Enable to use MySQL 5.6
[mysql56-community]
name=MySQL 5.6 Community Server
baseurl=http://s3.amazonaws.com/dev.hortonworks.com/
HDP-UTILS-1.1.0.21/repos/mysql-ppc64le/
enabled=1
gpgcheck=0
gpgkey=file:/etc/pki/rpm-gpg/RPM-GPG-KEY-mysql
```

HDP x86: Creates 2 repo files

`mysql-community.repo:`

```

[mysql-connectors-community]
name=MySQL Connectors Community
baseurl=http://repo.mysql.com/yum/mysql-connectors-community/el/7/$basearch/
enabled=1
gpgcheck=1
gpgkey=file:/etc/pki/rpm-gpg/RPM-GPG-KEY-mysql

[mysql-tools-community]
name=MySQL Tools Community
baseurl=http://repo.mysql.com/yum/mysql-tools-community/el/7/$basearch/
enabled=1
gpgcheck=1
gpgkey=file:/etc/pki/rpm-gpg/RPM-GPG-KEY-mysql

# Enable to use MySQL 5.5
[mysql55-community]
name=MySQL 5.5 Community Server
baseurl=http://repo.mysql.com/yum/mysql-5.5-community/el/7/$basearch/
enabled=0
gpgcheck=1
gpgkey=file:/etc/pki/rpm-gpg/RPM-GPG-KEY-mysql

# Enable to use MySQL 5.6
[mysql56-community]
name=MySQL 5.6 Community Server
baseurl=http://repo.mysql.com/yum/mysql-5.6-community/el/7/$basearch/
enabled=1
gpgcheck=1
gpgkey=file:/etc/pki/rpm-gpg/RPM-GPG-KEY-mysql

# Note: MySQL 5.7 is currently in development. For use at your own risk.
# Please read with sub pages: https://dev.mysql.com/doc/relnotes/mysql/5.7/en/
[mysql57-community-dmr]
name=MySQL 5.7 Community Server Development Milestone Release
baseurl=http://repo.mysql.com/yum/mysql-5.7-community/el/7/$basearch/
enabled=0
```

```
gpgcheck=1
gpgkey=file:/etc/pki/rpm-gpg/RPM-GPG-KEY-mysql
```

`mysql-community-source.repo:`

```

[mysql-connectors-community-source]
name=MySQL Connectors Community - Source
baseurl=http://repo.mysql.com/yum/mysql-connectors-community/el/7/SRPMS
enabled=0
gpgcheck=1
gpgkey=file:/etc/pki/rpm-gpg/RPM-GPG-KEY-mysql

[mysql-tools-community-source]
name=MySQL Tools Community - Source
baseurl=http://repo.mysql.com/yum/mysql-tools-community/el/7/SRPMS
enabled=0
gpgcheck=1
gpgkey=file:/etc/pki/rpm-gpg/RPM-GPG-KEY-mysql

[mysql55-community-source]
name=MySQL 5.5 Community Server - Source
baseurl=http://repo.mysql.com/yum/mysql-5.5-community/el/7/SRPMS
enabled=0
gpgcheck=1
gpgkey=file:/etc/pki/rpm-gpg/RPM-GPG-KEY-mysql

[mysql56-community-source]
name=MySQL 5.6 Community Server - Source
baseurl=http://repo.mysql.com/yum/mysql-5.6-community/el/7/SRPMS
enabled=0
gpgcheck=1
gpgkey=file:/etc/pki/rpm-gpg/RPM-GPG-KEY-mysql

[mysql57-community-dmr-source]
name=MySQL 5.7 Community Server Development Milestone Release - Source
baseurl=http://repo.mysql.com/yum/mysql-5.7-community/el/7/SRPMS
enabled=0
gpgcheck=1
gpgkey=file:/etc/pki/rpm-gpg/RPM-GPG-KEY-mysql
```

Configuring LogSearch

To setup LogSearch when IBM Spectrum Scale is integrated in an HDP environment, configuration changes are required to point to the correct NameNode, DataNode and ZKFC logs associated with IBM Spectrum Scale.

1. Click **Ambari GUI** > **Log Search service** > **Quick Links** > **Log Search UI**.
2. Login to **Log Search UI** and click on the top right corner button. Choose the **Configuration Editor** option.
3. In the **Configuration Editor** page, change the log path for the following components:
   a. `hdfs_datanode`
   b. `hdfs_namenode`
   c. `hdfs_zkfc`

   Example (existing) entry for NameNode:

   ```
   "type": "hdfs_namenode",
   "rowtype": "service",
   "path": "/var/log/hadoop/hdfs/hadoop-hdfs-namenode-*.log"


   change to:

   "type": "hdfs_namenode",
   "rowtype": "service",
   "path": "/var/log/hadoop/root/hadoop-root-namenode-*.log"
   ```

   Make similar changes for DataNode and zkfc.
4. Restart HDFS service.

Hadoop Kafka/Zookeeper and IBM Spectrum Scale Kafka/Zookeeper

IBM Spectrum Scale has new audit logging capability, File Audit Logging (FAL) that uses its own libraries for Kafka and zookeeper. In IBM Spectrum Scale FAL, it will install the Kafka and Zookeeper that were shipped from IBM Spectrum Scale.

To use the IBM Spectrum Scale Kafka and Zookeeper, and Hadoop Kafka and Zookeeper from HDP, you must install the Kafka and Zookeeper on different nodes for IBM Spectrum Scale and Hadoop.

In a new environment, IBM Spectrum Scale FPO is installed by the IBM Spectrum Scale Ambari Mpack. The packages for Kafka and zookeeper can be installed when HDP is installed or after the Mpack is installed. If it is required to install IBM Spectrum Scale audit logging, follow the IBM Spectrum Scale documentation to install the Kafka and Zookeeper on different nodes other than the ones installed for the HDP Hadoop cluster.

When you are installing HDP, if IBM Spectrum Scale is already installed with audit logging, check where the zookeeper and Kafka are to be installed. Ensure that the HDP zookeeper and Kafka are on nodes that are not where the IBM Spectrum Scale Kafka and zookeeper reside. If needed, change to different hosts in Ambari.

For example, if node1/node2/node3 are configured as IBM Spectrum Scale FAL message queue nodes, do not install HDP or community versions of Kafka/Zookeeper onto node1/node2/node3.

Note: When you stop IBM Spectrum Scale FAL message queue service, all Kafka and Zookeeper daemons on the nodes are stopped. This includes any HDP or community Kafka/Zookeeper daemons running on the same nodes.

Create Hadoop local directories in IBM Spectrum Scale

If you did not create local disks to host the Yarn and Mapreduce local temporary files, and want to use the existing IBM Spectrum Scale file system, then you need to create directories under IBM Spectrum Scale file system for each node.

Note: Performance might be impacted if the local disks are not used.

Recommendation: Create two partitions, one for local directories and one for IBM Spectrum Scale.

If you need to use the IBM Spectrum Scale file system, then create a fileset within IBM Spectrum Scale to host the local directories.

This section details the steps for creating a fileset within IBM Spectrum Scale to host the local directories:

1. Ensure that IBM Spectrum Scale is active and mounted.

   Start the IBM Spectrum Scale cluster on the console of any one node in the IBM Spectrum Scale cluster, by running the following command:

   ```
   /usr/lpp/mmfs/bin/mmstartup -a
   ```

   Mount the file system over all nodes by running the following command:

   ```
   /usr/lpp/mmfs/bin/mmmount <fs-name> -a
   ```

2. Create a fileset in IBM Spectrum Scale and set a policy to use only one replica:

   ```
   # Create a GPFS fileset
   $ mkdir /bigpfs/hadoop
   $ export PATH=$PATH:/usr/lpp/mmfs/bin
   $ mmcrfileset bigpfs local
   $ mmlinkfileset bigpfs local -J /bigpfs/hadoop/local

   # Create policy file
   $ vi hadoop.policy
   rule 'fset_local' set pool 'datapool' REPLICATE (1,1) FOR FILESET (local)
   rule 'default' set pool 'datapool'
   $ mmchpolicy bigpfs hadoop.policy

   # Verify fileset
   $ cd /bigpfs/hadoop/local
   ```

```
$ dd if=/dev/zero of=log bs=1M count=100
$ mmlsattr -d -L log
# Verify the output for data replication is 1
```

3. Create local directories for each host using the host name as the directory name for simplicity, and then change the permission. Run the following command to create the local directory from one of the IBM Spectrum Scale node.

```
for host in <your host name list>
do
 echo "$host"
 mkdir -p /bigpfs/hadoop/local/$host
done
chown -R yarn:hadoop /bigpfs/hadoop/local
chmod -R 755 /bigpfs/hadoop/local/*
```

4. For each node, link a local directory to its corresponding node directory named with its host name:

```
for host in <your host name list>
do
echo "$host"
mmdsh -N $host "ln -s /bigpfs/hadoop/local/$host /hadoop/yarn/local"
mmdsh -N $host "chown -R yarn:hadoop /hadoop/yarn/local"

// If additional user created directories are configured under Yarn in the shared storage,
// then ensure to create the corresponding user created directories in the local host and
// link them to the share storage directories.
// For example: yarn.nodemanager.log-dirs is set to /hadoop/yarn/log
// mmdsh -N $host "mkdir -p /hadoop/yarn/local/log"
// mmdsh -N $host "chown -R yarn:hadoop /hadoop/yarn/log"
done
```

5. After installation, click **Ambari GUI** > **Yarn** > **CONFIGS** to set the Yarn's configuration **`yarn.nodemanager.local-dirs`** as `/hadoop/yarn/local`.

Deploy HDP or IBM Spectrum Scale service on pre-existing IBM Spectrum Scale file system

If you have one of the following configurations, you can deploy HDP with the IBM Spectrum Scale service:

1. Pre-existing IBM Spectrum Scale cluster in FPO configuration.
2. Pre-existing IBM Spectrum Scale cluster with remote mount file system configuration.
3. Pre-existing IBM Spectrum Scale cluster in which the GPFS client nodes belongs to an ESS-based IBM Spectrum Scale cluster.

The steps for deployment are as follows:

1. A quorum node on the pre-existing must be selected as the IBM Spectrum Scale Master node.
2. Ensure that IBM Spectrum Scale is active and mounted.

```
[root@c902f09x13 ~]# mmgetstate -a

Node number Node name GPFS state
-------------------------------------------
1 c902f09x13 active
2 c902f09x16 active
3 c902f09x14 active
4 c902f09x15 active
[root@c902f09x13 ~]# mmlsmount bigpfs -L

File system bigpfs is mounted on 3 nodes:
192.0.2.0 c902f09x13
192.0.2.1 c902f09x14
192.0.2.2 c902f09x15
192.0.2.3 c902f09x16
[root@c902f09x13 ~]#
```

Note: Ensure that the FPO or the local Hadoop IBM Spectrum Scale cluster is set to automount on reboot by running the following command:

```
/usr/lpp/mmfs/bin/mmchfs <filesystem name> -A yes
```

3. Follow “Create HDP cluster” on page 360.
4. Follow “Install Mpack package” on page 366.
5. Follow the “Deploy the IBM Spectrum Scale service” on page 367, with the following deviations:
   - If you have not started the IBM Spectrum Scale cluster on the Ambari Assign Slaves and Clients page, click the **Previous** button to go back to **Assign Master** page in Ambari. Then start the IBM Spectrum Scale cluster, and mount the file system onto all the nodes. Go back to the Ambari GUI to continue to the **Assign Slaves and Client** page.
   - Verify that the **`gpfs.storage.type`** is set to
     - *local* for FPO
     - *shared* for Single cluster with all Hadoop nodes as IBM Spectrum Scale nodes
     - *remote* for Remote mount with all Hadoop nodes as IBM Spectrum Scale nodes
   - Verify the **`yarn.nodemanager.local-dirs`** and **`yarn.nodemanager.log-dirs`** values are set to an available mounted local partitioned directories that already exist in your file system.

     For example:

     Mounted local partitioned directories - `/opt/mapred/local<NUM>`

     `yarn.nodemanager.local-dirs``=/opt/mapred/local1/yarn`, `/opt/mapred/local2/yarn`, `/opt/mapred/local3/yarn`

     `yarn.nodemanager.log-dirs``=/opt/mapred/local1/yarn/logs`, `/opt/mapred/local2/yarn/logs`, `/opt/mapred/local3/yarn/logs`
   - Do not set the GPFS NSD stanza file field.

     For FPO, the IBM Spectrum Scale NSD stanza file is not required because the file system already exists. Because Ambari does not allow a blank value, leave the default value of IBM Spectrum Scale NSD stanza file.

     Note: If you accidentally place a value in the GPFS NSD stanza file field which was originally blank, and then try to remove it, you must leave in a “blank” character for Ambari to proceed.
   - **Single Scale cluster configuration**

     For **`gpfs.storage.type=shared`** the local cluster hosts with GPFS components (GPFS_Master or GPFS_Node) selected in the UI, are added on to the ESS/Shared IBM Spectrum Scale cluster.
     - Setting **`gpfs.storage.type`**=shared for Shared storage means this will create a single scale cluster configuration.
     - Setting **`gpfs.storage.type`**=shared for ESS and creating the `/var/lib/ambari-server/resources/shared_gpfs_node.cfg` file on the Ambari server will create a single scale cluster configuration. The file must contain only one FQDN of a node in the shared management host cluster, and password-less SSH must be configured from the Ambari server to this node. Ambari uses this one node to join the GNR/ESS cluster. Ensure that the file has at least 444 permission.
     - [Optional] To create local cache disks, see **Deploy the IBM Spectrum Scale service>Customize Services>Create Hadoop local cache disks** section.

       Note: If you are not using shared storage, you do not need this configuration, and you can leave this local cache disk parameter unchanged in the Ambari GUI.
       - Verify the following fields have the correct information that match your preinstalled IBM Spectrum Scale file system (GPFS) cluster.
         - GPFS cluster name
         - GPFS quorum nodes

- GPFS File System Name
- **gpfs.mnt.dir**
- **gpfs.storage.type**

Deploy FPO

In File Placement Optimizer (FPO) mode, data blocks are stored in chunks in IBM Spectrum Scale, and replicated to protect against disk and node failure. DFS clients run on the storage node so that they can leverage the data locality for executing the tasks quickly.

For the local storage mode configuration, "Short-circuit read (SSR)" on page 427 is recommended to improve the access efficiency.

Note: Ambari only supports creating an IBM Spectrum Scale FPO file system.

Follow the Installation but to create a new FPO cluster, the following deviation is to be followed:

- Skip "ESS setup" on page 359.
- Follow "Create HDP cluster" on page 360.
- Skip "Establish an IBM Spectrum Scale cluster on the Hadoop cluster" on page 364.
- Skip "Configure remote mount access" on page 365.
- Follow "Install Mpack package" on page 366.
- Follow the "Deploy the IBM Spectrum Scale service" on page 367 with the following deviations:

Under Assign Masters:

- All the Yarn's NodeManager nodes should be FPO nodes with the same number of disks for each node specified in the NSD stanza.

Under Customize Services:

- Configuration fields on both standard and advanced tabs are populated with values taken from the Hadoop performance tuning guide.
- Verify that the **gpfs.storage.type** is set to *local*.
- If you do not plan to have a sub-directory under the IBM Spectrum Scale mount point, do not click on the **gpfs.data.dir** field to preserve the field to not have any values set.
- Ensure the **yarn.nodemanager.local-dirs** and **yarn.nodemanager.local-logs** are set to a dummy local directory initially. When a new FPO is deployed, partitioned local directories dynamically replace the ones in **yarn.nodemanager.local-dirs** after the FPO system is created. Manually check to ensure that the **yarn.nodemanager.local-logs** value is set correctly. For more information, see "Disk-partitioning algorithm" on page 414.
- Create an NSD file, gpfs_nsd, and place it into the /var/lib/ambari-server/resources directory. Ensure that the permission on the file is at least 444. Add the NSD filename, gpfs_nsd, to the GPFS File system > GPFS NSD stanza file field in the Standard Config tab.

Two types of NSD files are supported for file system auto creation. One is the preferred simple format and another is the standard IBM Spectrum Scale NSD file format for IBM Spectrum Scale experts.

Simple NSD

If a simple NSD file is used, Ambari selects the proper metadata and data ratio for you. If possible, Ambari creates partitions on some disks for the Hadoop intermediate data, which improves the Hadoop performance. Simple NSD does not support existing partitioned disks in the cluster.

- Disk partitioning under Ambari would happen only if the following conditions are met:
  1. The NSD stanza requires all GPFS nodes to be specified in the NSD stanza file.
  2. Each of those nodes should have the same number of disks specified in the stanza file.
  3. Number of host entries in the stanza file/NSD servers should be equal to the number of Node managers. This requires all hosts running GPFS node to be set up as a Node manager too. If you do

not want Hadoop jobs to run on a specific GPFS node host (For example, on the Ambari server host), you could remove the Node manager component from that host after deploying the IBM Spectrum™ scale service.

For more details on disk partitioning, see the following:

- “Disk-partitioning algorithm” on page 414.
- “Partitioning function matrix in automatic deployment” on page 415.

Standard NSD

If the cluster has a partitioned file system, only a Standard NSD file can be used.

For standard IBM Spectrum Scale NSD file is used, administrators are responsible for the storage space arrangement.

- Apply the partition algorithm.

  Apply the algorithm for system pool and usage.
- Apply the failure group selection rule.

  Failure groups are created based on the rack location of the node.
- Define the Rack mapping file.

  Nodes can be defined to belong to racks.
- Partition the function matrix.

  The reason why one disk is divided into two partitions is so that one partition is used for the ext3 or ext4 to store the map or reduce intermediate data, while the other partition is used as a data disk in the IBM Spectrum Scale file system. Also, only data disks can be partitioned. Metadata disks cannot be partitioned.
- A policy file is required when a standard IBM Spectrum Scale NSD file is used.

  A policy file, gpfs_fs.pol, must be created and placed into the /var/lib/ambari-server/resources directory. Add the policy filename, gpfs_fs.pol, into the **GPFS policy file** field in the Standard Config tab.

  For more information on creating policy files, see “Policy File” on page 359.

For more information on each of the set-up points for standard NSD file, see “IBM Spectrum Scale-FPO deployment” on page 414.

Note: Deploying HDP over an existing IBM Spectrum Scale FPO cluster through Ambari, requires to either store the Yarn’s intermediate data into the IBM Spectrum Scale file system, or use idle disks formatted as a local file system. It is recommended to use the idle disks formatted as a local file system. For more information, see “Deploy HDP or IBM Spectrum Scale service on pre-existing IBM Spectrum Scale file system” on page 404.

Hadoop Storage Tiering

When using Hadoop Storage Tiering configuration, the jobs running on the Hadoop cluster with native HDFS can read and write the data from IBM Spectrum Scale in real time.

There would be only one copy of the data. For more information, see Hadoop Storage Tiering.

For information on viewfs, see “Apache Hadoop ViewFs support” on page 178.

Limited Hadoop nodes as IBM Spectrum Scale nodes

For any deployment model, you do not have to put all the Hadoop cluster nodes as GPFS nodes.

1. On all the Hadoop hosts that are either a NameNode or a DataNode in the native HDP cluster, assign a GPFS node so that the Transparency NameNodes and DataNodes are able to do a RPC in IBM Spectrum Scale.

2. If you need to configure additional Transparency DataNodes other than the native DataNodes, assign a GPFS Node on them as well.
3. You could also have GPFS Nodes that do not use Transparency service. For example, if you want to use a host GPFS protocol node, assign GPFS Node on that host.

Configuring multiple file system mount point access

Currently, the IBM Spectrum Scale service GUI can support only up to two file systems.

Note: The **gpfs.storage.type** has to be configured during the initial deployment of the IBM Spectrum Scale service and it cannot be changed later.

During the IBM Spectrum Scale Ambari deployment, the following fields are required for setting up the multiple file system access:

Fields	Description
gpfs.storage.type	Type of Storage. Comma-delimited string. The first value will be treated as the primary file system and the values after that will be treated as the secondary file systems. Only the following combination of file system values is supported: **gpfs.storage.type**=*local,remote* **gpfs.storage.type**=*remote,remote* **gpfs.storage.type**=*shared,shared*
gpfs.mnt.dir	Mount point directories for the file systems. Comma-delimited string. The first entry is for the primary file system. The second entry is the secondary file system.
gpfs.replica.enforced	Replication type for each file system (dfs or gpfs). Comma-delimited string. The first entry is for the primary file system. The second entry is the secondary file system.
gpfs.data.dir	Only one value must be specified. Null is a valid value. The data directory is created only for the primary file system.
GPFS file system name	Names of the file systems. Comma-delimited string. The first entry is for the primary file system. The second entry is the secondary file system.

Note:

1. If **gpfs.storage.type** has a local value, a pre-existing IBM Spectrum Scale cluster is required. If an FPO file system is not created, it can be created if the NSD stanza files are specified. If an FPO file system is created, the information is propagated in Ambari.
2. If **gpfs.storage.type** has remote value, the pre-existing IBM Spectrum Scale remote mounted file system is required. For information on how to configure remote mount file system, see "Configure remote mount access" on page 365.

Follow the instructions based on the type of deployment model that you have:

1. Add remote mount file systems access to existing HDP and an FPO file system that was deployed by IBM Spectrum Scale Ambari service.

Prerequisites:

- Deployed HDP.
- Deployed FPO file system via IBM Spectrum Scale service through Ambari. The Ambari server requires to be on the GPFS master node.
- Pre-existing remote mount file system.

Use the **`gpfs.storage.type=local,remote`** configuration setting.

On the Ambari server node on the local FPO file system:

- Stop All services.

  On the Ambari UI, click **Actions** > **Stop All**[1] to stop all the services.
- On the owning IBM Spectrum Scale cluster, run the **`/usr/lpp/mmfs/bin/mmlsmlount all`** command to ensure that the file system is mounted.

  This step is needed for the IBM Spectrum Scale deploy wizard to automatically detect the existing file systems.
- Update the IBM Spectrum Scale configuration:

  Click **Ambari GUI** > **Spectrum Scale** > **Configs tab** and update the following fields:

  - **`gpfs.storage.type`**
  - **`gpfs.mnt.dir`**
  - **`gpfs.replica.enforced`**
  - **`gpfs.data.dir`**
  - **`GPFS FileSystem Name`**

  In this example, the primary file system mount point is `/localfs` and the secondary file system mount point is `/remotefs`.

  Setting of the fields would be as follows:

  ```
  gpfs.storage.type=local,remote
  gpfs.mnt.dir=/localfs,/remotefs
  gpfs.replica.enforced=dfs,dfs
  gpfs.data.dir=myDataDir OR gpfs.data.dir=
  GPFS FileSystem Name=localfs,remotefs
  ```
- Restart the IBM Spectrum Scale service.
- Restart any service with **Restart Required** icon.
- Click **Ambari** > **Actions** > **Start All** to start all the services.

2. Add remote mount file systems access to existing HDP and an IBM Spectrum Scale FPO file system that was deployed manually.

   Prerequisites:

   - An FPO file system that is manually created.
   - Deployed HDP on the manually created FPO file system. The Ambari server requires to be on the GPFS master node.
   - Pre-existing remote mount file system.

   Use the **`gpfs.storage.type=local,remote`** configuration setting.

   On the Ambari server node on the local FPO file system, perform the following:

   - Stop All services.

     On the Ambari UI, click **Actions** > **Stop All**[1] to stop all the services.
   - Start the IBM Spectrum Scale service cluster.

     On the local IBM Spectrum Scale cluster, run the `/usr/lpp/mmfs/bin/mmstartup -a` command.

- Ensure all the remote mount file system is active and mounted.
- On each IBM Spectrum Scale cluster, run the `/usr/lpp/mmfs/bin/mmgetstate -a` command to ensure it is started.

  This step is needed for the IBM Spectrum Scale deploy wizard to automatically detect the existing file systems.
- Deploy the IBM Spectrum Scale service on the pre-existing file system.

  During deployment, the wizard would detect both the file systems and would populate the IBM Spectrum Scale config UI with recommended values for **`gpfs.storage.type`**, **`gpfs.mnt.dir`** **`gpfs.replica.enforced`**, **`gpfs.data.dir`** and **`GPFS FileSystem Name`** fields. Review the recommendations and correct them as needed before you continue to deploy the service.

  In this example, the primary file system mount point is `/localfs` and the secondary file system mount point is `/remotefs`.

  Setting of the fields would be as follows:

  ```
  gpfs.storage.type=local,remote
  gpfs.mnt.dir=/localfs,/remotefs
  gpfs.replica.enforced=dfs,dfs
  gpfs.data.dir=myDataDir OR gpfs.data.dir=
  GPFS FileSystem Name=localgpfs,remotegpfs
  ```
- Click **Ambari** > **Actions** > **Start All** to start all the services.

3. Add remote mount file systems access to existing HDP and a manually created IBM Spectrum Scale cluster.

   Create the FPO file system onto the local IBM Spectrum Scale cluster.

   Prerequisites:

   - A manual IBM Spectrum Scale cluster is created.
   - No FPO file system was created.
   - Deployed HDP onto the manual IBM Spectrum Scale cluster. The Ambari server requires to be on the GPFS master node.
   - Pre-existing remote mount file system.

   Use **`gpfs.storage.type=local,remote`** configuration setting.

   On the Ambari server node on the local cluster:

   - Stop All services.

     On the Ambari UI, click **Actions** > **Stop All**[1] to stop all the services.
   - Start IBM Spectrum Scale service cluster.

     On the local IBM Spectrum Scale cluster, run the `/usr/lpp/mmfs/bin/mmstartup -a` command.
   - Ensure all the remote mount file system is active and mounted.
   - On each IBM Spectrum Scale cluster, run the `/usr/lpp/mmfs/bin/mmgetstate -a` command to ensure it is started.

     This step is needed for the IBM Spectrum Scale deploy wizard to automatically detect the existing file systems.
   - Deploy the IBM Spectrum Scale service.

     During deployment, the wizard would detect both the file systems and would populate the IBM Spectrum Scale config UI with recommended values for **`gpfs.storage.type`**, **`gpfs.mnt.dir`**, **`gpfs.replica.enforced`**, **`gpfs.data.dir`** and **`GPFS FileSystem Name`** fields. Review the recommendations and correct them as needed before you continue to deploy the service.

     In this example, the primary file system mount point is `/localfs` and the secondary file system mount point is `/remotefs`.

    – Configure fields for FPO cluster:

        - Update the NSD stanza file.

          If this is a standard stanza file, update the policy file field.

        - Review the replication fields. Default is set to 3.

    In this example, the primary file system mount point is `/localfs` and the secondary file system mount point is `/remotefs`.

    Setting of the fields would be as follows:

    ```
    gpfs.storage.type=local,remote
    gpfs.mnt.dir=/localfs,/remotefs
    gpfs.replica.enforced=dfs,dfs
    gpfs.data.dir=myDataDir OR gpfs.data.dir=
    GPFS FileSystem Name=localfs,remotefs
    ```

    Note: The newly created FPO cluster is set as the primary file system. The remote mounted file system is set as the secondary file system.

- Restart IBM Spectrum Scale service.
- Restart any service with the **Restart Required** icon.
- On the Ambari UI, click **Actions** > **Start All** to start all the services.

4. Add only the remote mount file systems access to existing HDP and a manually created IBM Spectrum Scale cluster.

   Prerequisites:

   - A manual IBM Spectrum Scale cluster is created.
   - Deployed HDP onto the manual IBM Spectrum Scale cluster. The Ambari server node requires to be on the GPFS master node.
   - Pre-existing remote mount file systems.

   Use **`gpfs.storage.type=remote,remote`** configuration setting.

   On the Ambari server node, on the local cluster:

   - Stop All services.

     On the Ambari UI, click **Actions** > **Stop All**[1] to stop all the services.

   - Start the IBM Spectrum Scale cluster.

     On the local IBM Spectrum Scale cluster, run the `/usr/lpp/mmfs/bin/mmstartup -a` command.

   - Ensure all the remote mount file system is active and mounted.
   - On each IBM Spectrum Scale cluster, run the `/usr/lpp/mmfs/bin/mmgetstate -a` command to ensure it is started.

     This step is needed for the IBM Spectrum Scale deploy wizard to automatically detect the existing file systems.

   - Deploy the IBM Spectrum Scale service.

     During deployment, the wizard would detect both the file systems and would populate the IBM Spectrum Scale config UI with recommended values for **`gpfs.storage.type`**, **`gpfs.mnt.dir`** **`gpfs.replica.enforced`**, **`gpfs.data.dir`** and **`GPFS FileSystem Name`** fields. Review the recommendations and correct them as needed before you continue to deploy the service.

     In this example, the primary file system mount point is `/remotefs1` and the secondary file system mount point is `/remotefs2`.

     Setting of the fields would be as follows:

     ```
     gpfs.storage.type=remote,remote
     gpfs.mnt.dir=/remotefs1,/remotefs2
     gpfs.replica.enforced=dfs,dfs
     ```

```
gpfs.data.dir=myDataDir OR gpfs.data.dir=
GPFS FileSystem Name=remotefs1,remotefs2
```

- On the Ambari UI, click **Actions** > **Start All** to start all the services.

5. Add the shared file system access to an existing HDP Scale cluster from an ESS or IBM Spectrum Scale cluster. Shared file system mode is a single GPFS cluster where the Hadoop Scale cluster is part of the existing ESS or IBM Spectrum Scale cluster.

   Prerequisites:

   - Deployed HDP cluster.
   - Two pre-existing GPFS file system.

   Use the `gpfs.storage.type=shared,shared` configuration setting.

   On the Ambari server node, on the local cluster:

   - Stop All services.

     On the Ambari UI, click **Actions** > **Stop All** to stop all the services.
   - Ensure that each of the file system is active and mounted on the ESS or the IBM Spectrum Scale cluster.
   - Deploy the IBM Spectrum Scale service. During deployment, the wizard would detect both the file systems and would populate the IBM Spectrum Scale config UI with recommended values for **`gpfs.storage.type`**, **`gpfs.mnt.dir`**, **`gpfs.replica.enforced`**, **`gpfs.data.dir`** and GPFS file system Name fields. Review the recommendations and correct them as needed before you continue to deploy the service.

     In this example, the primary file system mount point is `/essfs1` and the secondary file system mount point is `/essfs2`.

     Setting of the fields would be as follows:

     ```
     gpfs.storage.type=/shared,/shared
     gpfs.mnt.dir=/essfs1,/essfs2
     gpfs.replica.enforced=dfs,dfs
     gpfs.data.dir=myDataDir OR gpfs.data.dir=
     GPFS FileSystem Name=essfs1,essfs2
     ```
   - After the IBM Spectrum Scale service is deployed successfully, on the Ambari UI, click **Actions** > **Start All** to start all the services.

[1] For FPO cluster, do not run STOP ALL from the Ambari GUI. Refer to the Limitations > General section on how to properly stop IBM Spectrum Scale.

Support for Big SQL

Big SQL V 6.0.0 is supported from HDP 3.1 with Mpack 2.7.0.3 and HDFS Transparency 3.1.0-1.

The **IBM Db2 Big SQL service** > **Configs** > **Advanced bigsql-env** > **Database** path cannot be changed after Big SQL is installed. Therefore, Big SQL must be installed after IBM Spectrum Scale service is installed so that the Database path is set correctly to use the IBM Spectrum Scale storage.

For more information about IBM Db2® Big SQL V 6.0.0, see IBM Db2 Big SQL V6.0 documentation.

Issue:

All the HDP services are up but the Big SQL job might fail with the following error if the hdp_version (after Big SQL integration) is not setup properly:

```
[04:16:25 ERROR     ]    <Stream 1-Main>    Caused by:
[04:16:25 ERROR     ]    <Stream 1-Main>    com.ibm.utm.core.sql.UtmSqlException: Error
executing statement:
LOAD HADOOP USING FILE URL '/user/user1/biga_load//data/small_types_pos_data.unl' WITH SOURCE
PROPERTIES
('field.delimiter'='|', 'date.time.format'='yyyy-MM-dd-HH.mm.ss.SSSSSSSSS',
'date.time.format'='yyyy-MM-dd-HH.mm.ss.SSSSSS', 'date.time.format'='MM-dd-yyyy',
'escape.char'='\\') INTO TABLE ALL_TYPES_TAB
```

```
Sql Code:  -5111
IsamCode:  0
Sql State: 58005
[04:16:25 ERROR     ]    <Stream 1-Main>    Error code: -5111
[04:16:25 ERROR     ]    <Stream 1-Main>    SQLSTATE   : 58005
[04:16:25 ERROR     ]    <Stream 1-Main>    Message    : The LOAD HADOOP statement failed
because of an
error with a component. Component name: "". Reason code: "4:URISyntaxExce". Log entry
identifier:
"[BSL-0-1d2d2361d]". Job identifier: "".. SQLCODE=-5111, SQLSTATE=58005, DRIVER=3.71.22
[04:16:25 ERROR     ]    <Stream 1-Main>      Caused by:
[04:16:25 ERROR     ]    <Stream 1-Main>      com.ibm.db2.jcc.am.SqlException: The LOAD HADOOP
statement
failed because of an error with a component. Component name: "". Reason code:
"4:URISyntaxExce". Log entry
identifier: "[BSL-0-1d2d2361d]". Job identifier: "".. SQLCODE=-5111, SQLSTATE=58005,
DRIVER=3.71.22
```

The JobHistory will have the following error:

```
Log Type: prelaunch.err
Log Upload Time: Wed Apr 03 04:22:00 -0400 2019
Log Length: 1027
/hadoop/yarn/local/usercache/bigsql/appcache/application_1554277915663_0004
/container_e28_1554277915663_0004_02_000001/launch_container.sh: line 38: $PWD:$PWD
/mr-framework/hadoop/share/hadoop/mapreduce/*:$PWD/mr-framework/hadoop/share/hadoop/
mapreduce/lib/*:
$PWD/mr-framework/hadoop/share/hadoop/common/*:$PWD/mr-framework/hadoop/share/hadoop/
common/lib/*:
$PWD/mr-framework/hadoop/share/hadoop/yarn/*:$PWD/mr-framework/hadoop/share/hadoop/yarn/lib/*:
$PWD/mr-framework/hadoop/share/hadoop/hdfs/*:$PWD/mr-framework/hadoop/share/hadoop/hdfs/lib/*:
$PWD/mr-framework/hadoop/share/hadoop/tools/lib/*:
/usr/hdp/${hdp.version}/hadoop/lib/hadoop-lzo-0.6.0.${hdp.version}.jar:
/etc/hadoop/conf/secure:/usr/ibmpacks/bigsql/6.0.0.0/bigsql/hive-client/lib/*:
/usr/ibmpacks/bigsql/6.0.0.0/bigsql/hive-client/lib/*:/usr/hdp/3.1.0.0-78/atlas/hook/hive/*:
/usr/ibmpacks/bigsql/6.0.0.0/bigsql/hive-client/lib/hive-hcatalog-
core-3.1.0.3.1.0.0-78.jar:job.jar/*:
job.jar/classes/:job.jar/lib/*:$PWD/*:$PWD/atlas-application.properties: bad substitution

Log Type: prelaunch.out
Log Upload Time: Wed Apr 03 04:22:00 -0400 2019
Log Length: 25
Setting up env variables
```

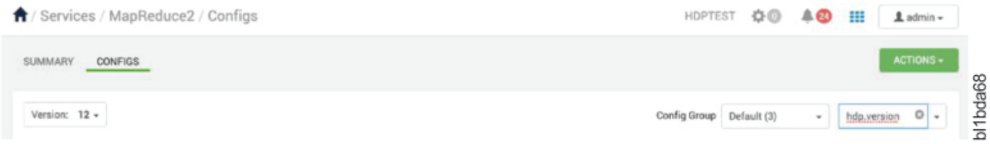
To fix the issue:

1. On an HDP node, run the **`hdp-select versions`** command to get the HDP version.

   ```
   [root@c902f14x01 ~]# hdp-select versions
   3.1.0.0-78
   [root@c902f14x01 ~]#
   ```

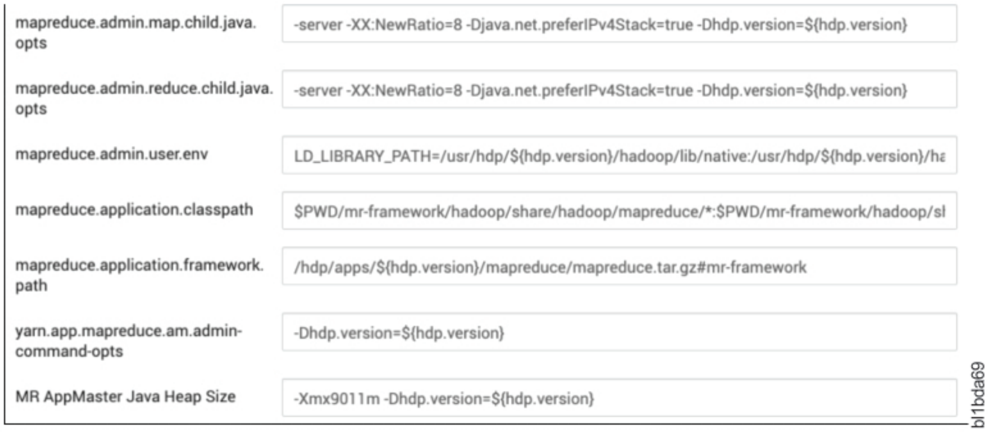
   This example found 3.1.0.0-78 as the HDP version value.

2. Go to the Ambari GUI, select **MapReduce2 service** > **Configs** and search for hdp.version.

3. Replace all the ${hdp.version} values with the actual HDP version number.

   For example, change all the fields below with ${hdp.version} tag to 3.1.0.0-78 value.



mapreduce.admin.map.child.java.opts	-server -XX:NewRatio=8 -Djava.net.preferIPv4Stack=true -Dhdp.version=${hdp.version}
mapreduce.admin.reduce.child.java.opts	-server -XX:NewRatio=8 -Djava.net.preferIPv4Stack=true -Dhdp.version=${hdp.version}
mapreduce.admin.user.env	LD_LIBRARY_PATH=/usr/hdp/${hdp.version}/hadoop/lib/native:/usr/hdp/${hdp.version}/ha
mapreduce.application.classpath	$PWD/mr-framework/hadoop/share/hadoop/mapreduce/*:$PWD/mr-framework/hadoop/sl
mapreduce.application.framework.path	/hdp/apps/${hdp.version}/mapreduce/mapreduce.tar.gz#mr-framework
yarn.app.mapreduce.am.admin-command-opts	-Dhdp.version=${hdp.version}
MR AppMaster Java Heap Size	-Xmx9011m -Dhdp.version=${hdp.version}

4. Restart all the required services.

Administration

IBM Spectrum Scale-FPO deployment

This section provides the information for FPO deployment.

Disk-partitioning algorithm

If a simple NSD file is used without the -meta label, Ambari assigns metadata and data disks and partitions the disk according to the following rules:

1. If node number is less than or equal to four:
   - If the disk number of each node is less than or equal to three, put all disks to system pool, and set usage = metadataanddata. Partitioning is not done.
   - If the disk number of each node is greater than or equal to four, assign metaonly and dataonly disks based on a 1:3 ratio on each node. The MAX metadisk number per node is four. Partitioning is done if all NodeManager nodes are also NSD nodes, and have the same number of NSD disks.
2. If the node number is equal to or greater than five:
   - If the disk number of each node is less than or equal to two, put all disks to the system pool, and usage is metadataanddata. Partitioning is not done.
   - Set four nodes to metanodes where meta disks are located. Others are DataNodes.
   - Failure groups are created based on the failure group selection rule.
   - Assign meta disk and data disks to the meta node. Assign only data disk to the DataNode. The ratio follows best practice, and falls between 1:3 and 1:10.
   - If all GPFS nodes have the same number of NSD disks, create a local partition on data disks for Hadoop intermediate data.

Failure Group selection rules

Failure groups are created based on rack allocation of the nodes. One rack mapping file is supported (Rack Mapping File).

Ambari reads this rack mapping file, and assigns one failure group per rack. The rack number must be three or greater than three. If rack mapping file is not provided, virtual racks are created for data fault toleration.

1. If the node number is less than five, each node is on a different rack.

2. If the node number is greater than or equal to five, and node number is less than 10, every two nodes are put in one virtual rack.
3. If the node number is greater than or equal to ten and node number is less than 21, every three nodes are put in one virtual rack.
4. If the node number is greater than or equal to 21, every 10 nodes are put in one virtual rack.

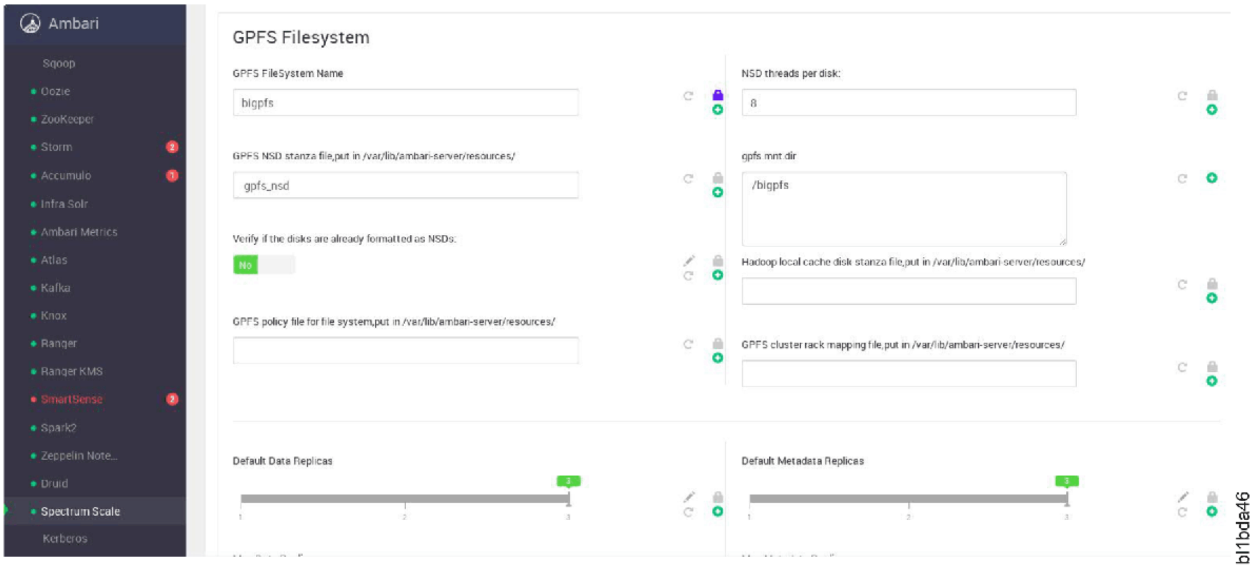
Rack mapping file

Nodes can be defined to belong to racks. For three or more racks, the failure groups of the NSD will correspond to the rack the node is in.

A sample file is available on the Ambari server at `/var/lib/ambari-server/resources/mpacks/SpectrumScaleExtension-MPack-<version>/extensions/SpectrumScaleExtension/<version>/services/GPFS/package/templates/racks.sample`. To use, copy the `racks.sample` file to the `/var/lib/ambari-server/resources` directory.

```
$ cat /var/lib/ambari-server/resources/racks.sample

#Host/Rack map configuration file
#Format:
#[hostname]:/[rackname]
#Example:
#mn01:/rack1
#NOTE:
#The first character in rack name must be "/"
mn03:/rack1
mn04:/rack2
dn02:/rack3
```



GPFS Filesystem

GPFS FileSystem Name: bigpfs

GPFS NSD stanza file,put in /var/lib/ambari-server/resources/: gpfs_nsd

Verify if the disks are already formatted as NSDs: No

GPFS policy file for file system,put in /var/lib/ambari-server/resources/:

NSD threads per disk: 8

gpfs.mnt.dir: /bigpfs

Hadoop local cache disk stanza file,put in /var/lib/ambari-server/resources/:

GPFS cluster rack mapping file,put in /var/lib/ambari-server/resources/:

Default Data Replicas: 3

Default Metadata Replicas: 3

Figure 41. AMBARI RACK MAPPING

Partitioning function matrix in automatic deployment

Each data disk is divided into two parts. One part is used for an ext4 file system to store the map, or reduce intermediate data, while the other part is used as a data disk in the IBM Spectrum Scale file system. Only the data disks can be partitioned. Meta disks cannot be partitioned.

If a node is not selected as NodeManager for Yarn there will not be a map or reduce tasks running on that node. In this case, partitioning the disks of the node is not favorable because the local partition will not be used.

The following table describes the partitioning function matrix:

Table 36. IBM Spectrum Scale partitioning function matrix

Node manager host list	**Specify the standard NSD file**	**Specify the simple NSD file without the -meta label**	**Specify the simple NSD file with the -meta label**
#1: <node manager host list> == <IBM Spectrum Scale NSD server nodes> The node manager hostlist is equal to IBM Spectrum Scale NSD server nodes.	No partitioning. Create an NSD directly with the NSD file.	Partition and select the meta disks for the customer according to Disk-partitioning algorithm and Failure Group selection rules.	No partitioning. All disks marked with the -meta label are used for metadata NSD disks. All others are marked as data NSDs.
#2: <node manager host list>><IBM Spectrum Scale NSD server nodes> Some node manager hosts are not in the IBM Spectrum Scale NSD server nodes but all IBM Spectrum Scale NSD server nodes are in the node manager host list.	No partitioning. Create the NSD directly with the specified NSD file.	No partitioning, but select the meta disks for the customer according to Disk-partitioning algorithm and Failure Group selection rules.	No partitioning. All disks marked with the -meta label are used for metadata NSD disks. All others are marked as data NSDs.
<node manager host list><<IBM Spectrum Scale NSD server nodes> Some IBM Spectrum Scale NSD server nodes are not in the node manager host list but all node manager host lists are in the IBM Spectrum Scale NSD server nodes.	No partitioning. Create the NSD directly with the specified NSD file.	No partitioning, but select the meta disks for customer according to Disk-partitioning algorithm and Failure Group selection rules.	No partitioning. All disks marked with the -meta label are used for metadata NSD disks. All others are marked as data NSDs.

For standard NSD files, or simple NSD files with the -meta label, the IBM Spectrum Scale NSD and file system are created directly.

To specify the disks that must be used for metadata, and have data disks partitioned, use the `partition_disks_general.sh` script to partition the disks first, and specify the partition that is used for GPFS NSD in a simple NSD file.

Send an email to scale@us.ibm.com to request the `partition_disks_general.sh` script.

For example:

```
$ cat /var/lib/ambari-server/resources/gpfs_nsd

DISK|compute001.private.dns.zone:/dev/sdb-meta,/dev/sdc2,/dev/sdd2
DISK|compute002.private.dns.zone:/dev/sdb-meta,/dev/sdc2,/dev/sdd2
DISK|compute003.private.dns.zone:/dev/sdb-meta,/dev/sdc2,/dev/sdd2
DISK|compute005.private.dns.zone:/dev/sdb-meta,/dev/sdc2,/dev/sdd2
DISK|compute006.private.dns.zone:/dev/sdb,/dev/sdc2,/dev/sdd2
DISK|compute007.private.dns.zone:/dev/sdb,/dev/sdc2,/dev/sdd2
```

After deployment is done by this mode, manually update the `yarn.nodemanager.local-dirs` and `yarn.nodemanager.log-dirs` files to contain the directory list from the disk partitions that are used to map or reduce intermediate data.

Ranger

Enabling Ranger

This section provides instructions to enable Ranger.

Ranger can be configured before or after IBM Spectrum Scale service is deployed. The HDFS Transparency does not need to be in an unintegrated state.

Ranger procedure

This topic lists the steps to install Ranger

Follow these steps to enable Ranger:

- Configuring MySQL for Ranger
- Installing Ranger through Ambari
- Enabling Ranger HDFS plugin
- Logging into Ranger UI

Configuring MySQL for Ranger

Prepare the environment by configuring MySQL to be used for Ranger.

1. Create a non-root user to create the Ranger databases.

   In this example, the username *rangerdba* with password *rangerdba* is used.

   a. Log in as the root user to the DB host node. Ensure that the DB is running. This is the node that has MySQL installed, which is usually the Hive server node. Use the following commands to create the *rangerdba* user, and grant the user adequate privileges:

   ```
   CREATE USER 'rangerdba'@'localhost' IDENTIFIED BY 'rangerdba';
   GRANT ALL PRIVILEGES ON *.* TO 'rangerdba'@'localhost';
   CREATE USER 'rangerdba'@'%' IDENTIFIED BY 'rangerdba';
   GRANT ALL PRIVILEGES ON *.* TO 'rangerdba'@'%';
   GRANT ALL PRIVILEGES ON *.* TO 'rangerdba'@'localhost' WITH GRANT OPTION;
   GRANT ALL PRIVILEGES ON *.* TO 'rangerdba'@'%' WITH GRANT OPTION;
   FLUSH PRIVILEGES;
   ```

   After setting the privileges, use the **`exit`** command to exit MySQL.

   b. Reconnect to the database as user *rangerdba* by using the following command:

   ```
   mysql -u rangerdba -prangerdba
   ```

   After testing the *rangerdba* login, use the **`exit`** command to exit MySQL.

2. Check MySQL Java connector.

   a. Run the following command to confirm that the `mysql-connector-java.jar` file is in the Java share directory. This command must be run on the Ambari server node.

   ```
   ls /usr/share/java/mysql-connector-java.jar
   ```

Note: If the `/usr/share/java/mysql-connector-java.jar` is not found, install the `mysql-connector-java` package on the `ambari-server node`

```
$ yum install mysql-connector-java
```

b. Use the following command to set the `jdbc/driver/path` based on the location of the MySQL JDBC driver .jar file. This command must be run on the Ambari server node.

```
ambari-server setup --jdbc-db={database-type} --jdbc-driver={/jdbc/driver/path}
```

For example:

```
ambari-server setup --jdbc-db=mysql --jdbc-driver=/usr/share/java/mysql-connector-java.jar
```

Installing Ranger through Ambari
This topic lists the steps to install Ranger through Ambari.

1. Log in to Ambari UI.
2. Add the Ranger service. Click **Ambari dashboard** > **Actions** > **Add Service**.

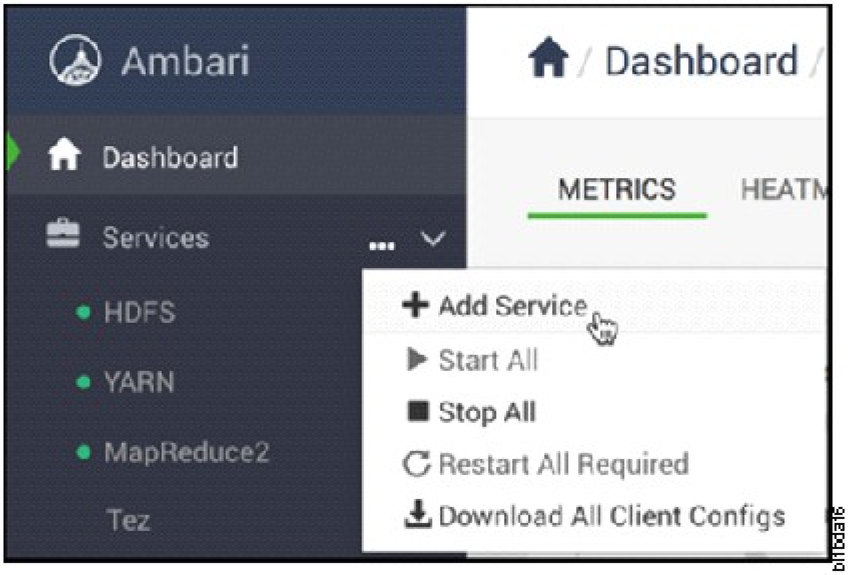
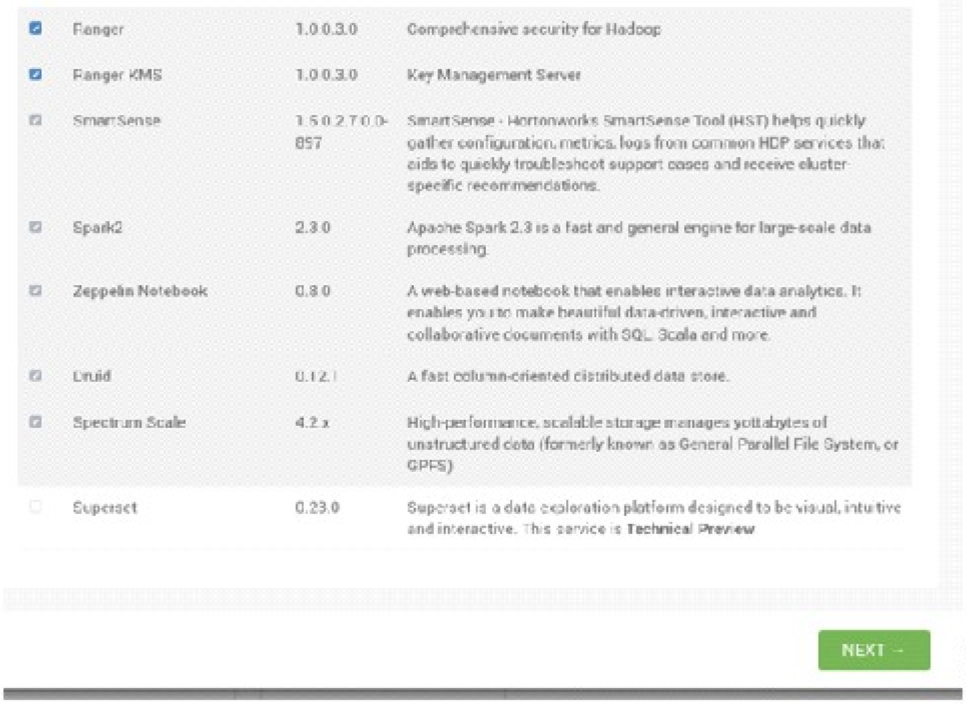


Figure 42. Ambari dashboard

3. On the **Choose Services** page, select **Ranger** and **Ranger KMS**.

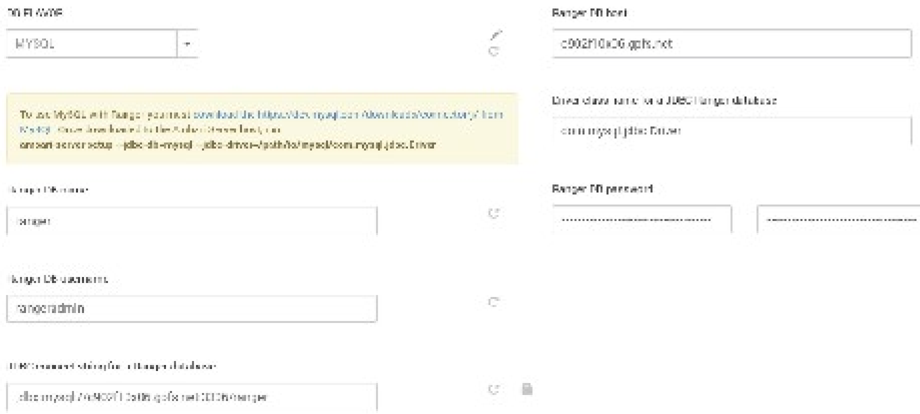


☑	Ranger	1.0.0.3.0	Comprehensive security for Hadoop
☑	Ranger KMS	1.0.0.3.0	Key Management Server
☑	SmartSense	1.5.0.2.7.0.0-897	SmartSense - Hortonworks SmartSense Tool (HST) helps quickly gather configuration, metrics, logs from common HDP services that aids to quickly troubleshoot support cases and receive cluster-specific recommendations.
☑	Spark2	2.3.0	Apache Spark 2.3 is a fast and general engine for large-scale data processing.
☑	Zeppelin Notebook	0.8.0	A web-based notebook that enables interactive data analytics. It enables you to make beautiful data-driven, interactive and collaborative documents with SQL, Scala and more.
☑	Druid	0.12.1	A fast column-oriented distributed data store.
☑	Spectrum Scale	4.2.x	High-performance, scalable storage manages yottabytes of unstructured data (formerly known as General Parallel File System, or GPFS)
☐	Superset	0.23.0	Superset is a data exploration platform designed to be visual, intuitive and interactive. This service is **Technical Preview**

bl1bda17

4. Customize the services. In the Ranger Admin dashboard, configure the following:

- Under “DB Flavor”, select MYSQL.
- For the Ranger DB host, the host name must be the location of MYSQL.
- For Ranger DB username, set the value to *rangeradmin*.
- For Ranger DB password, set the value to *rangeradmin*.



DB FLAVOR: MYSQL

Ranger DB host: c902f10x06.gpfs.net

To use MySQL with Ranger, you must download the https://dev.mysql.com/downloads/connector/j/ from MySQL. Once downloaded to the Ambari Server host, run:
ambari-server setup --jdbc-db=mysql --jdbc-driver=/path/to/mysql/com.mysql.jdbc.Driver

Driver class name for a JDBC Ranger database: com.mysql.jdbc.Driver

Ranger DB name: ranger

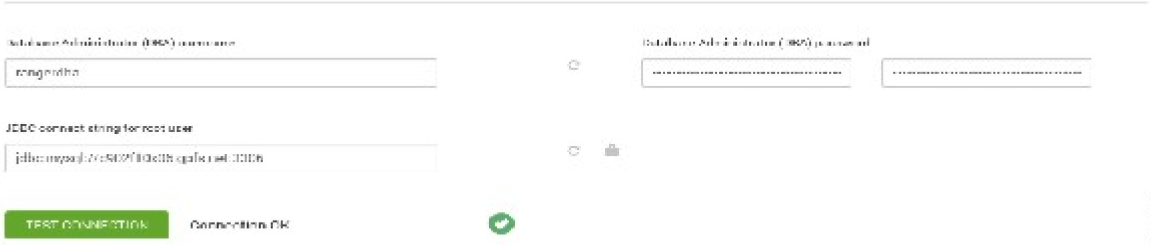
Ranger DB password: ································

Ranger DB username: rangeradmin

JDBC connect string for a Ranger database: jdbc:mysql://c902f10x06.gpfs.net:3306/ranger

bl1bda18

- For the Database Administrator (DBA) username, set the value to *root*.
- For the Database Administrator (DBA) password, set the value to *password for root user*.
- Click on the Test Connection button and ensure that the connection result is OK.



Database Administrator (DBA) username: rangerdba

Database Administrator (DBA) password: ································

JDBC connect string for root user: jdbc:mysql://c902f10x06.gpfs.net:3306

Connection OK ✓

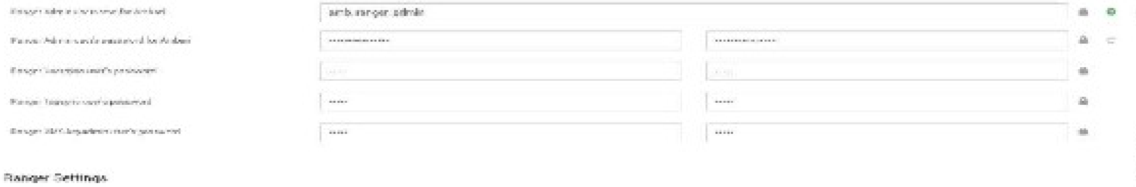
bl1bda19

- In ADVANCED tab set password for:

a. Ranger Usersync user's password

b. Ranger Tagsync user's password

c. Ranger KMS keyadmin user's password



amb_ranger_admin

Ranger Settings

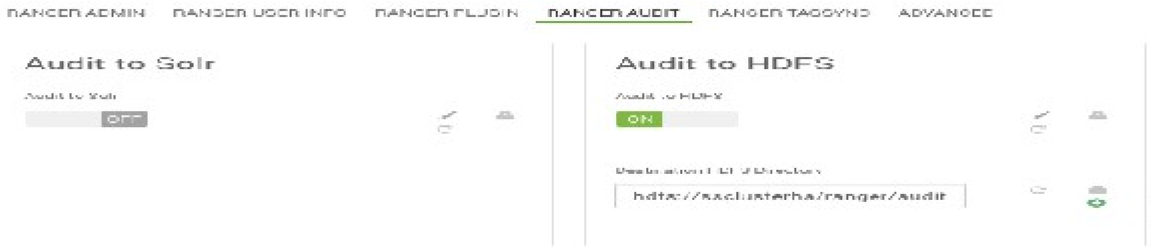
bl1bda20

- Provide password for 'Advanced ranger-env' tab for 'Ranger Admin user's password'

Advanced ranger-env

bl1bda21

- In the Ranger Audit tab, ensure that the Audit to Solr option is disabled.



Audit to Solr

OFF

Audit to HDFS

ON

hdfs://saclusterha/ranger/audit

bl1bda22

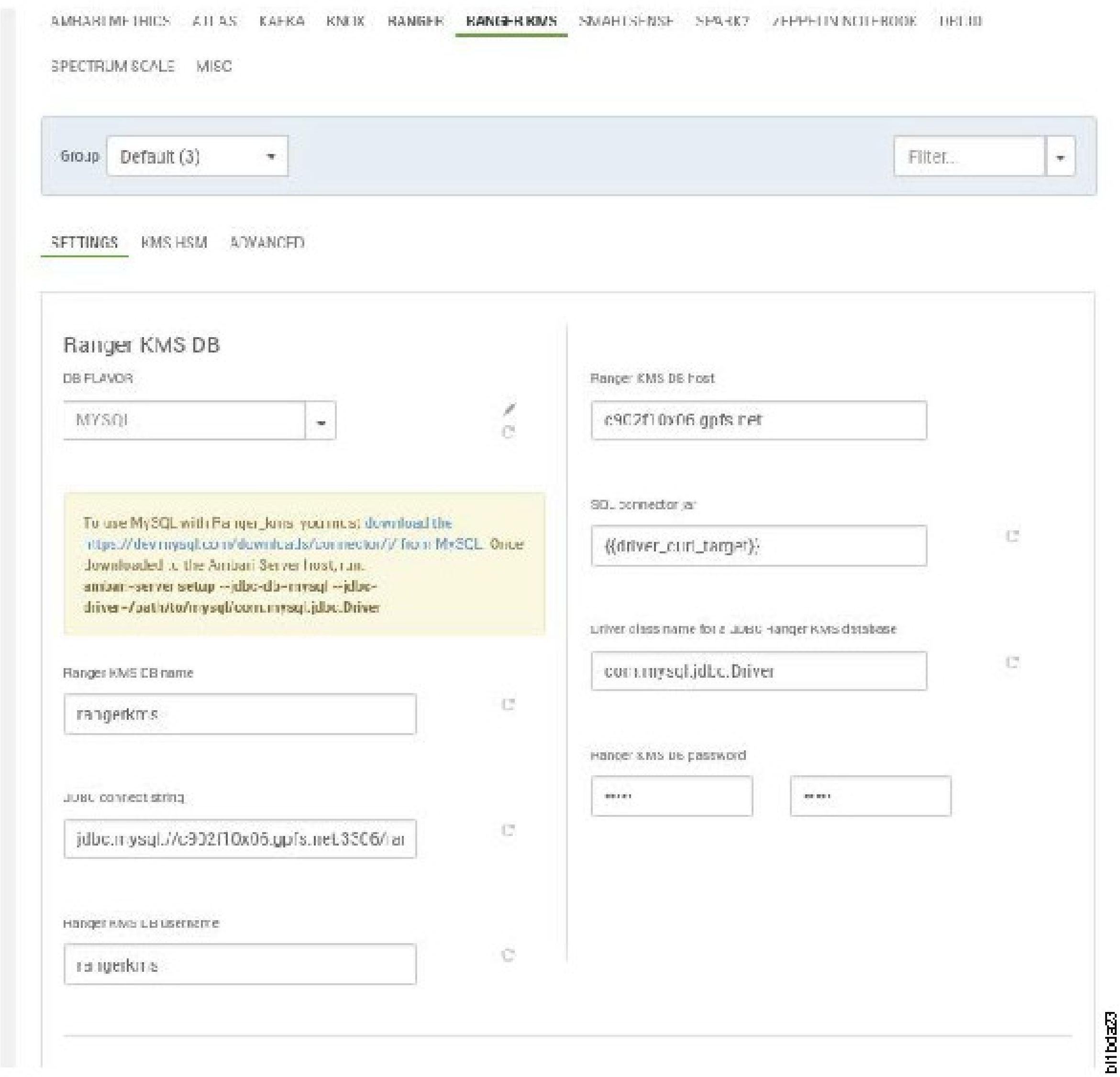
5. Customize the services, In the 'RANGER KMS'

   a. Under “DB Flavor”, select MYSQL.

   b. For the 'Ranger KMS DB host', the hostname must be the location of MYSQL.

   c. For Ranger KMS DB username, set the value to rangerkms.

   d. For Ranger KMS DB password, set the value to rangerkms.



Group: Default (3)

Ranger KMS DB

DB FLAVOR: MYSQL

To use MySQL with Ranger_kms you must download the https://dev.mysql.com/downloads/connector/j/ from MySQL. Once downloaded to the Ambari Server host, run:
ambari-server setup --jdbc-db=mysql --jdbc-driver=/path/to/mysql/com.mysql.jdbc.Driver

Ranger KMS DB name: rangerkms

JDBC connect string: jdbc:mysql://c902f10x06.gpfs.net:3306/ar

Ranger KMS DB username: rangerkms

Ranger KMS DB host: c902f10x06.gpfs.net

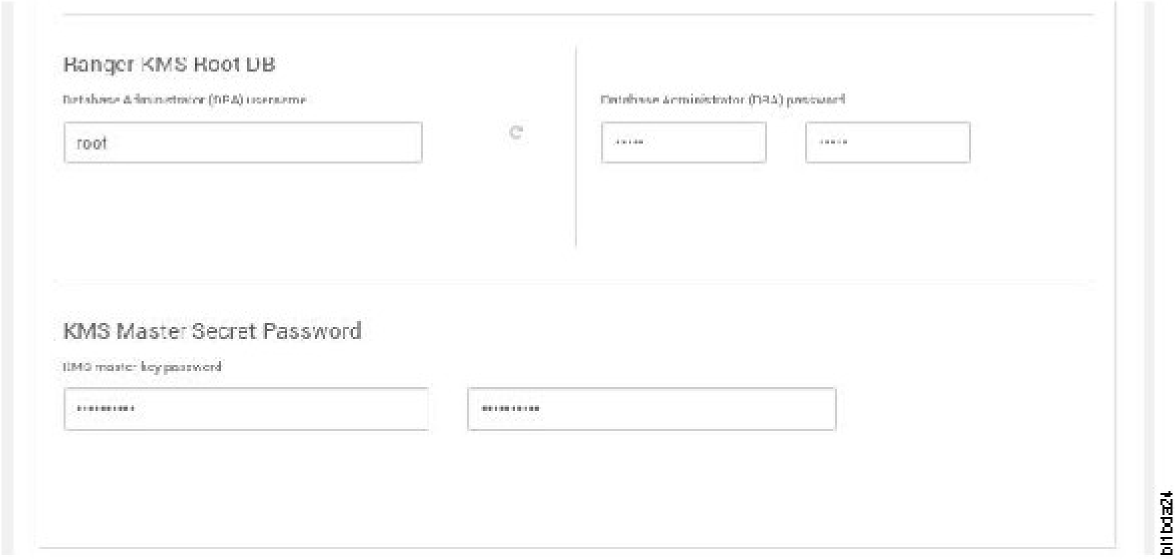
SQL connector jar: {{driver_curl_target}}

Driver class name for a JDBC Ranger KMS database: com.mysql.jdbc.Driver

Ranger KMS DB password: ••••• •••••

e. For 'Database Administrator (DBA) password', set the root user password.

f. For 'KMS master key password' set new password.



Ranger KMS Root DB

Database Administrator (DBA) username: root

Database Administrator (DBA) password: ••••• •••••

KMS Master Secret Password

KMS master key password: •••••••••• ••••••••••

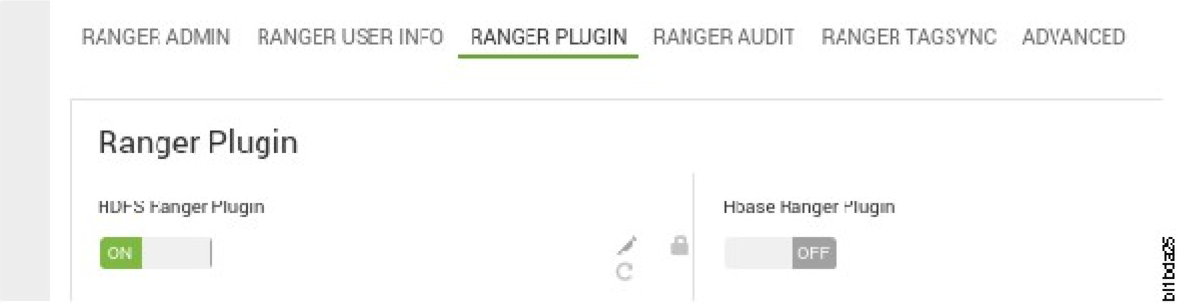
g. Deploy and complete the installation.

   Assign the Ranger server to be on the same node as the HDFS Transparency NameNode for better performance.

   Select **Next** > **Next** > **Deploy**.

Enabling Ranger HDFS plug-in
This topic lists the steps to enable Ranger HDFS plug-in

1. From the dashboard, click **Configs tab** > **Advanced tab** > **Advanced ranger-hdfs-plugin-properties**, check the box for Enable Ranger for HDFS.



Ranger Plugin

HDFS Ranger Plugin: ON

Hbase Ranger Plugin: OFF

2. Save the configuration. The `Restart required` message is displayed at the top of the page. Click **Restart**, and select **Restart All Affected** to restart the HDFS service, and load the new configuration.

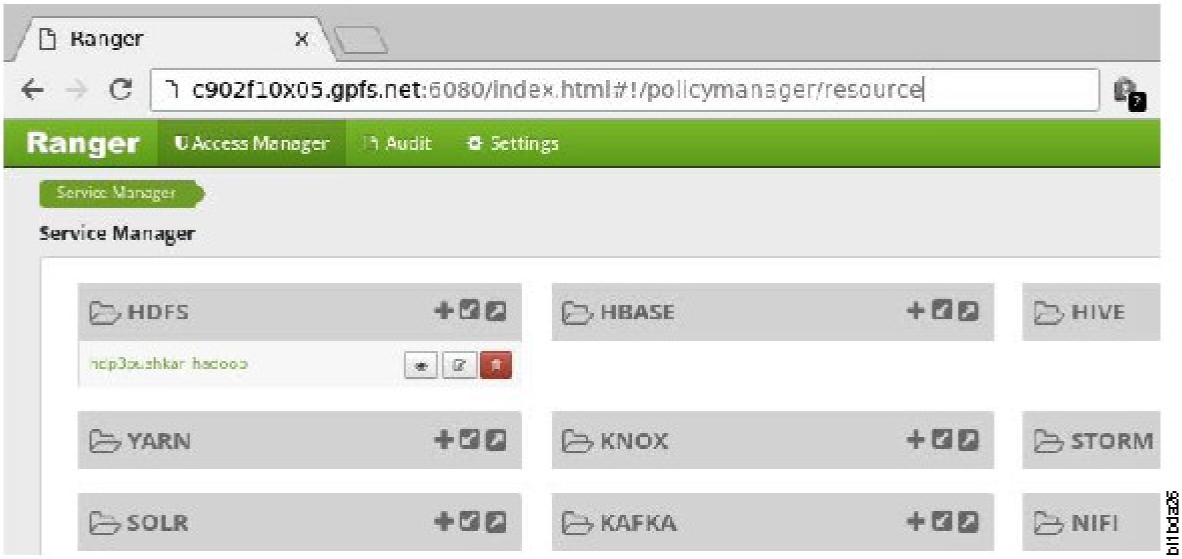
   Note: After each step we need to save the config and **Restart All Affected** for the services requesting for it and load the new configuration.

   After the HDFS restarts, the Ranger plug-in for HDFS is enabled.

Logging into Ranger UI
This topic provides instructions to log in to the Ranger UI.

To log into the Ranger UI, log onto: `http://<gateway>:6080` using the following username and password:

User ID/Password: admin/admin



c902f10x05.gpfs.net:6080/index.html#!/policymanager/resource

Service Manager

HDFS: hdp3pushkar_hadoop

Disabling Ranger

If you do not want to use Ranger any more, do the following:

From the dashboard, click **Configs tab** > **Advanced tab** > **Advanced ranger-hdfs-plugin-properties** and uncheck the Disable Ranger box for HDFS.

To increase performance, you could also disable Ranger in HDFS Transparency by executing the following steps:

1. Log in to the Ambari GUI.
2. Select the **IBM Spectrum Scale service** > **Configs** and set the value of **`gpfs.ranger.enabled`** to *false*.
3. Save the configuration.
4. Restart IBM Spectrum Scale service, then restart HDFS to sync this value to all the nodes.

Kerberos

Enabling Kerberos

Only MIT KDC is supported for IBM Spectrum Scale service through Ambari.

If you are using Kerberos that is not MIT KDC:

1. Disable the Kerberos.
2. Install IBM Spectrum Scale service.
3. Enable Kerberos.

   Note:

   - If Kerberos is not disabled, then the IBM Spectrum Scale service can hang.
   - For Kerberos issues, see "Service fails to start" on page 480.

Enabling Kerberos when the IBM Spectrum Scale service is not integrated

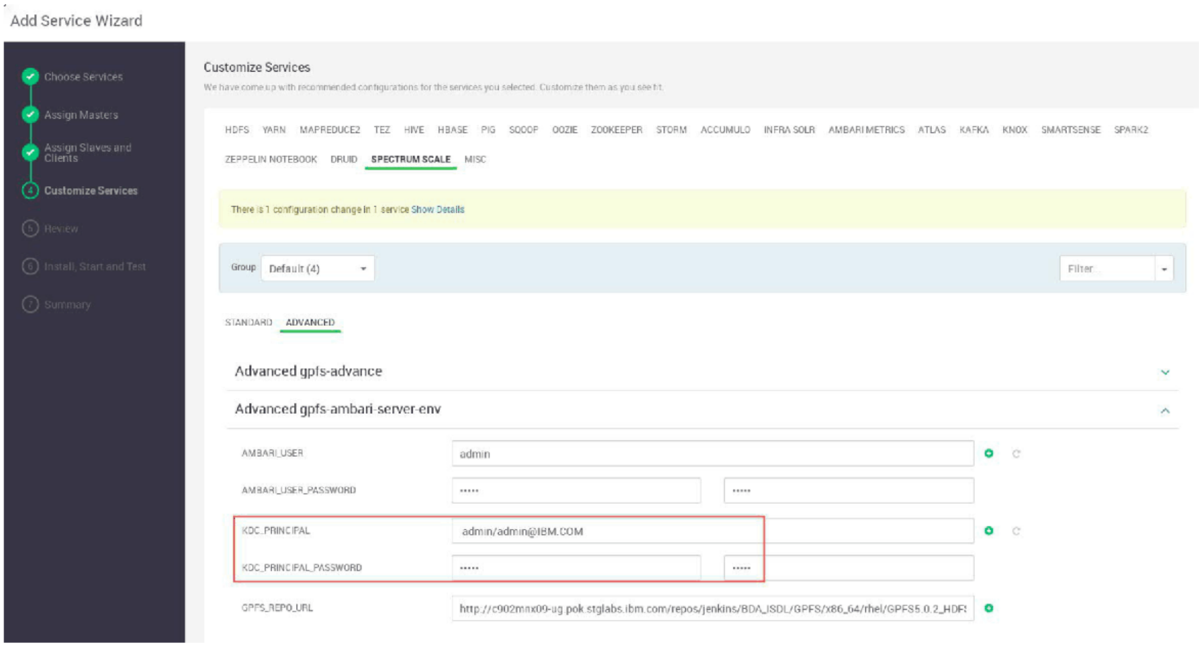
IBM Spectrum Scale service is not integrated into Ambari.

1. Follow "Setting up KDC server and enabling Kerberos" on page 425 to enable Kerberos. This is before deploying IBM Spectrum Scale service in "Install Mpack package" on page 366 and "Deploy the IBM Spectrum Scale service" on page 367.

   Once the Kerberos is enabled, the KDC information must be set during the deployment of the IBM Spectrum Scale Customizing Services panel.
2. During the IBM Spectrum Scale service deployment phase of Customizing Services:

   When adding the IBM Spectrum Scale service to a Kerberos-enabled system into Ambari, the KDC_PRINCIPAL and the KDC_PRINCIPAL_PASSWORD fields seen in the Customize Services screen must be updated with the actual values.

   Input the correct KDC admin principal and KDC admin principal password into the fields:



Add Service Wizard

Customize Services

We have come up with recommended configurations for the services you selected. Customize them as you see fit.

There is 1 configuration change in 1 service Show Details

Group: Default (4)

Advanced gpfs-advance

Advanced gpfs-ambari-server-env

AMBARI_USER	admin	
AMBARI_USER_PASSWORD	•••••	•••••
KDC_PRINCIPAL	admin/admin@IBM.COM	
KDC_PRINCIPAL_PASSWORD	•••••	•••••
GPFS_REPO_URL	http://c902mnx09-ug.pok.stglabs.ibm.com/repos/jenkins/BDA_ISDL/GPFS/x86_64/rhel/GPFS5.0.2_HDF!	

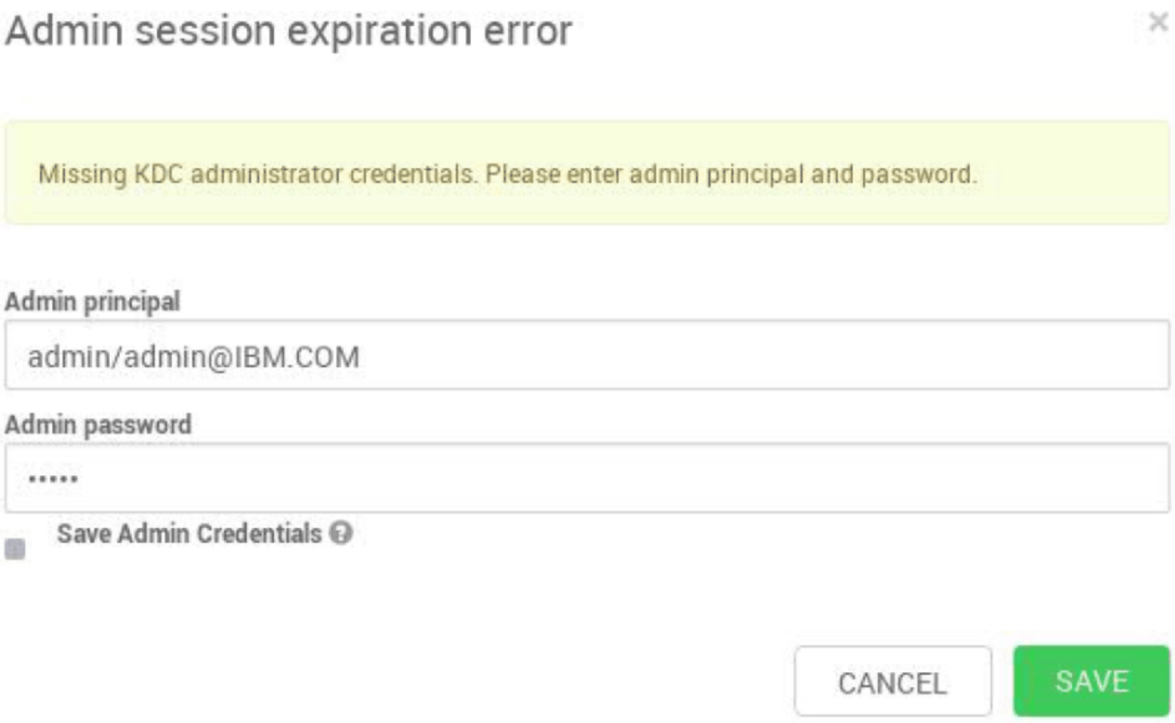
bl1bda47

After all the required fields are set for the **customized services** panel, review all the fields in **Customizing Services** before clicking **NEXT**.

Note: The Admin principal and Admin password are the same as the corresponding KDC_PRINCIPAL and KDC_PRINCIPAL_PASSWORD values.

```
KDC_PRINCIPAL=Admin principal
KDC_PRINCIPAL_PASSWORD=Admin password
```

The KDC admin principal and KDC admin principal password are generated when the KDC server is set up.



Admin session expiration error

Missing KDC administrator credentials. Please enter admin principal and password.

Admin principal: admin/admin@IBM.COM

Admin password: •••••

□ Save Admin Credentials

bl1bda48

3. Continue in **Customizing Services** tab to continue installation of the IBM Spectrum Scale service.

Enabling Kerberos when the IBM Spectrum Scale service is integrated

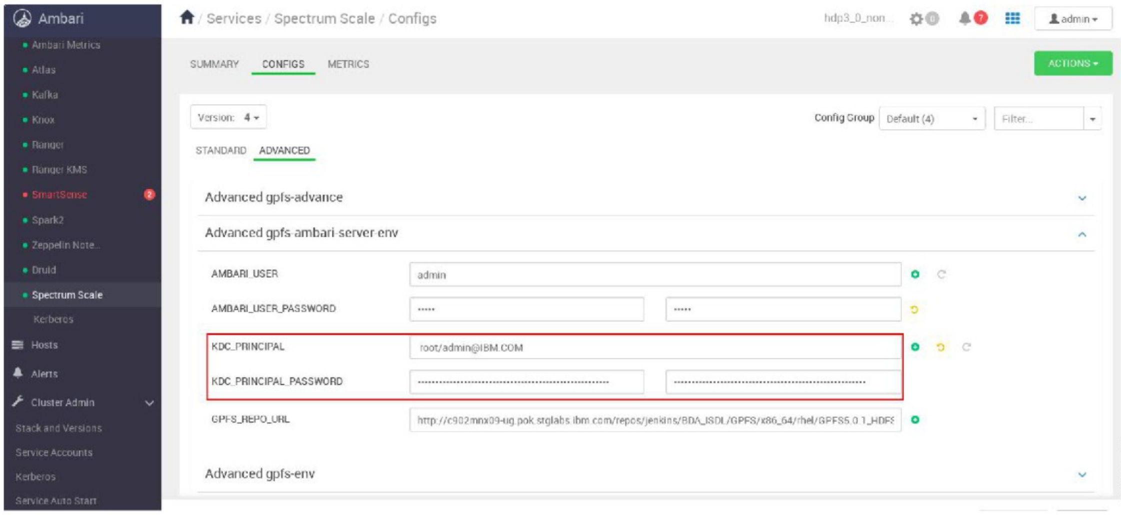
If Kerberos is to be enabled after IBM Spectrum Scale service is already integrated, the KDC_PRINCIPAL and KDC_PRINCIPAL_PASSWORD is required to be set in the IBM Spectrum Scale Configuration panel. Add the principal and password before enabling Kerberos. While enabling or disabling Kerberos, you do not need to stop the IBM Spectrum Scale service.

1. Follow the steps in "Setting up KDC server and enabling Kerberos" on page 425 to enable Kerberos.

   Note: During the enable Kerberos process, the Start and Test service is done. If the check services fail, you must exit, and go to the next step to add the KDC principal and password into IBM Spectrum Scale.

2. From Ambari, click **Spectrum Scale service** > **Configs tab** > **Advanced** > **Advanced gpfs-ambari-server-env**.

   Type the KDC principal values and the KDC principal password values. Save the configuration.



Version: 4

Config Group: Default (4)

Advanced gpfs-ambari-server-env

AMBARI_USER	admin	
AMBARI_USER_PASSWORD	•••••	•••••
KDC_PRINCIPAL	root/admin@IBM.COM	
KDC_PRINCIPAL_PASSWORD	••	••
GPFS_REPO_URL	http://c902mnx09-ug.pok.stglabs.ibm.com/repos/jenkins/BDA_ISDL/GPFS/x86_64/rhel/GPFS5.0.1_HDFS	

bl1bda49

3. Restart all the services. Click **Ambari panel** > **Service Actions** > **Stop All and Start All**[1].

   [1] For FPO cluster, do not run STOP ALL from the Ambari GUI. Refer to the Limitations > General section on how to properly stop IBM Spectrum Scale.

Setting up KDC server and enabling Kerberos

This topic provides steps to set up KDC server and enable Kerberos.

1. To set up the Key Distribution Center (KDC) server:

   - For HDP, follow the Install a new MIT KDC documentation.

   Note: If the KDC server is already implemented, skip this step.

2. On the Ambari GUI, click **Admin** > **Kerberos**, and follow the GUI panel guide to enable the Kerberos service.

Kinit on the NameNodes

This topic describes kinit on the NameNodes. These commands are run internally during HDFS service start up when the IBM Spectrum Scale service is integrated.

On the NameNodes, run: `# kinit -kt /etc/security/keytabs/nn.service.keytab nn/NN_HOSTNAME@REALM_NAME`

where,

- NN_HOSTNAME is the NameNode host name (FQDN).
- REALM_NAME is the KDC Realm.
- nn is the Kerberos NameNode naming convention created during Kerberos setup.

For example: `kinit -kt /etc/security/keytabs/nn.service.keytab nn/c902f05x01.gpfs.net@IBM.COM`.

Note: If in a non-root environment, this command is internally run with sudo privilege.

If HA, run the command on both the NameNodes.

Kinit on the DataNodes

This topic describes kinit on the DataNodes.

On the DataNodes, run: `# kinit -kt /etc/security/keytabs/dn.service.keytab dn/DN_HOSTNAME@REALM_NAME`.

where,

- DN_HOSTNAME is the DataNode host name (FQDN).
- REALM_NAME is the KDC Realm.
- dn is the Kerberos DataNode naming convention created during Kerberos setup.

For example: `kinit -kt /etc/security/keytabs/dn.service.keytab dn/c902f05x01.gpfs.net@IBM.COM`.

If in a non-root environment, ensure that you run this command with sudo privilege.

Issues in Kerberos enabled environment

This section lists the issues in the kerberos enabled environment and their workarounds.

Bad local directories in Yarn

If Yarn shows an alert for bad local directories when IBM Spectrum Scale is integrated, and if the Yarn service check failed then Yarn does not have the correct permission to access the local mounted directories created by IBM Spectrum Scale. Click **Ambari** > **Yarn** > **Configs** > **Advanced** > **Node Manager**, and review the `yarn.nodemanager.local-dirs` for the local directory values.

Workaround

Fix the local directory permissions on all nodes to have yarn:hadoop user ID and group ID permissions. Restart all services, or go back to the previous step and continue with the process.

For example,

```
# Local directories under /opt/mapred
/dev/sdf1 on /opt/mapred/local1 type ext4 (rw,relatime,data=ordered)
/dev/sdg1 on /opt/mapred/local2 type ext4 (rw,relatime,data=ordered)
/dev/sdh1 on /opt/mapred/local3 type ext4 (rw,relatime,data=ordered)

# Check the directories under /opt/mapred
In /opt/mapred directory:
drwxrwxrwx 6 root root 4096 Mar  8 23:19 local3
drwxrwxrwx 6 root root 4096 Mar  8 23:19 local2
              drwxrwxrwx 6 root root 4096 Mar  8 23:19 local1

# Workaround:
# Change permission from root:root to yarn:hadoop for all the local* directories under /opt/
mapred
# for all the nodes.

Under /opt/mapred directory:
chown yarn.hadoop local*

drwxrwxrwx 6 yarn hadoop 4096 Mar  8 23:19 local3
drwxrwxrwx 6 yarn hadoop 4096 Mar  8 23:19 local2
              drwxrwxrwx 6 yarn hadoop 4096 Mar  8 23:19 local1

# Restart all services (Or go back to your previous step and continue with the process).
```

Nodemanager failure due to device busy

Nodemanager fails to start due to local directory error:

```
OSError: [Errno 16] Device or resource busy: '/opt/mapred/local1'
```

To fix this issue:

1. Go to **Yarn** > **Configs** > **Search** for **`yarn.nodemanager.local-dirs`**.
2. Check the values for the Yarn local directories.
3. The correct local directory values must contain the Yarn directory in the local directory path. For example:

```
yarn.nodemanager.local-dirs="/opt/mapred/local1/yarn,/opt/mapred/local2/yarn,/opt/mapred/
local3/yarn"
```

4. If the `<local-dir>/yarn` is not specified in `yarn.nodemanager.local-dirs`, add the path, and save the configuration.

Journal nodes not installed in the native HDFS HA

If you are unintegrating from a Kerberos-enabled NameNode HA mode environment to native HDFS, you may sometimes find JournalNodes components missing in HDFS. In such cases, manually install the journal nodes.

Disabling Kerberos

This topic lists the steps to disable kerberos.

Note: While enabling or disabling Kerberos, you do not need to stop the IBM Spectrum Scale service.

To disable Kerberos from Ambari:

1. Go to **Ambari GUI** > **Admin** > **Kerberos** > **Disable Kerberos**.

Short-circuit read (SSR)

In HDFS, read requests go through the DataNode. When the client requests the DataNode to read a file, the DataNode reads that file off the disk, and sends the data to the client over a TCP socket. The short-circuit read (SSR) obtains the file descriptor from the DataNode, allowing the client to read the file directly.

This is possible only in cases where the client is colocated with the data, and is used in the FPO mode. The short-circuit reads provide a substantial performance boost to many applications.

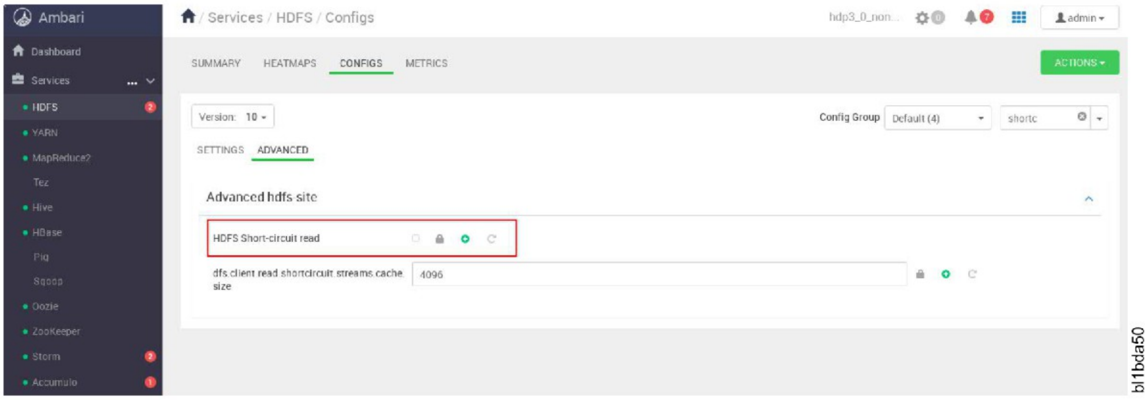
Prerequisite: Install the Java OpenJDK development tool-kit package, `java-<version>-openjdk-devel`, on all nodes.

The short-circuit read is disabled by default in IBM Spectrum Scale Ambari management pack.

To disable or enable the short-circuit read in Ambari with IBM Spectrum Scale:

You must plan a cluster maintenance window, and prepare for cluster downtime when disabling or enabling short circuit.

- Check (enable) or uncheck (disable) the HDFS Short-circuit read box from the **Ambari HDFS dashboard** > **Configs tab** > **Advanced tab** > **Advanced hdfs-site panel**. Save the configuration.
- Stop all services. Click **Ambari** > **Actions** > **Stop All**.
- Start all services. Click **Ambari** > **Actions** > **Start All**.



Disabling short circuit write

This section describes how to disable short circuit write.

Note: By default, the short circuit write is enabled only if the short circuit read is enabled.

1. Go to **Ambari GUI** > **Spectrum Scale** > **Custom gpfs-site**, add the **`gpfs.short-circuit-write.enabled=false`** property, and save the configuration.
2. Restart IBM Spectrum Scale service.
3. Restart HDFS service.
4. Restart any services that are down.

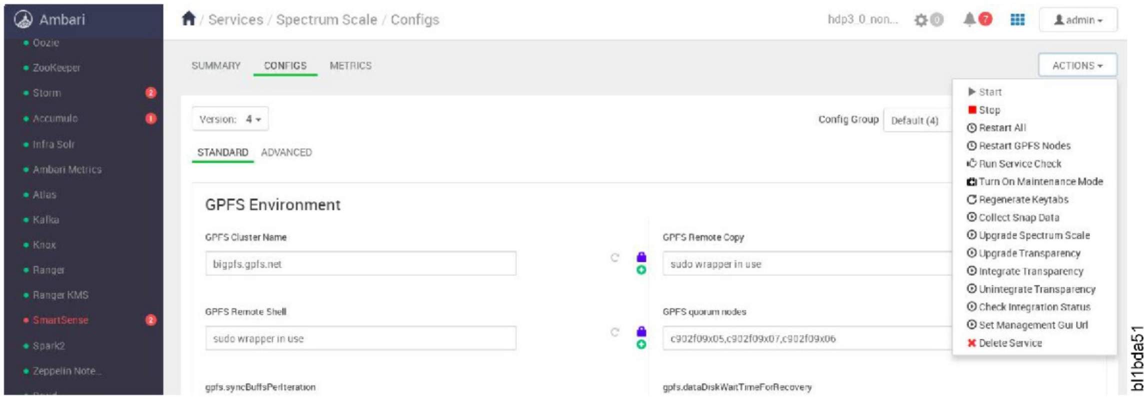
   Note: If **`gpfs.short-circuit-write.enabled`** is *disabled*, there will be a lot of traffic over the local network lo adapter when you run a `teragen` job.

IBM Spectrum Scale service management IBM Spectrum Scale

Manage the IBM Spectrum Scale through the IBM Spectrum Scale dashboard. The status and utilization information of IBM Spectrum Scale and HDFS Transparency can be viewed on this panel.

Actions dropdown list

To go to the Service Actions dropdown list, click **Spectrum Scale** > **Actions**.

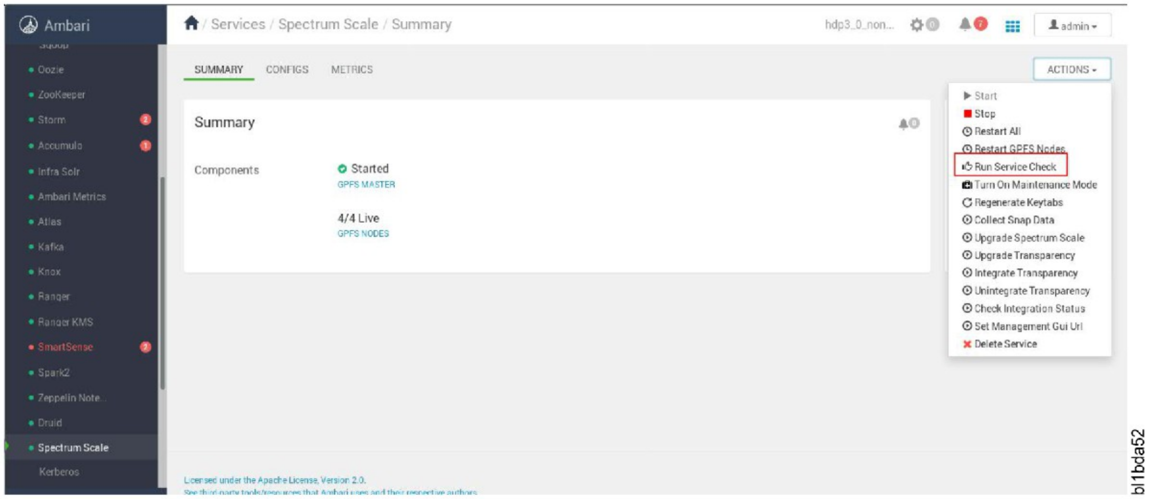


Note: Do not use the **Delete Service** action from the dropdown **Actions** menu. If you wish to get rid of the IBM Spectrum Scale service, follow the procedure in "Uninstalling IBM Spectrum Scale Mpack and service" on page 393.

When the IBM Spectrum Scale Mpack and service is deleted, the IBM Spectrum Scale file system and packages are preserved as is. For an FPO cluster created through Ambari, the mounted local disks `/opt/mapred/local*` and entries in `/etc/fstab` are preserved as is.

Running the service check

To check the status and stability of the service, run a service check on the IBM Spectrum Scale dashboard by clicking **Run Service Check** in the **Actions** dropdown menu.



- Review the service check output logs for any issues.
- To manually check the HDFS Transparency NameNodes and DataNodes state, run the following command:

  ```
  /usr/lpp/mmfs/bin/mmhadoopctl connector getstate
  ```

  ```
  $ /usr/lpp/mmfs/bin/mmhadoopctl connector getstate
  c902f05x01.gpfs.net: namenode running as process 4749.
  c902f05x01.gpfs.net: datanode running as process 10214.
  c902f05x02.gpfs.net: datanode running as process 4767.
  c902f05x03.gpfs.net: datanode running as process 8204.
  ```

Stop all without stopping IBM Spectrum Scale service

To prevent IBM Spectrum Scale service from being stopped when you click **ACTION** > **STOP ALL**, place the IBM Spectrum Scale service into maintenance mode.

Click **Ambari GUI** > **Spectrum Scale service** > **Actions** > **Turn on Maintenance Mode**.

This prevents any Ambari actions from occurring on the service that is in maintenance mode.

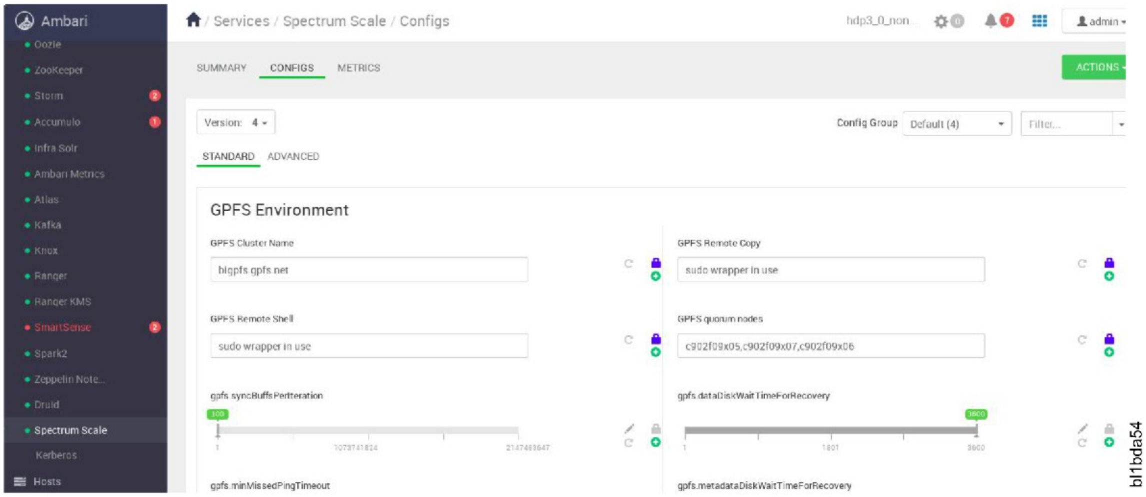
To get out of Maintenance Mode, click **Ambari GUI** > **Spectrum Scale service** > **Actions** > **Turn off Maintenance Mode**.

Note: For FPO cluster, do not run STOP ALL from the Ambari GUI. Refer to the Limitations > General section on how to properly stop IBM Spectrum Scale.

Modifying IBM Spectrum Scale service configurations

The IBM Spectrum Scale service has standard and advanced configuration panels.

Click **Ambari GUI** > **Spectrum Scale** > **Configs tab**.



Version: 4

Config Group Default (4)

GPFS Environment

GPFS Cluster Name: bigpfs.gpfs.net

GPFS Remote Copy: sudo wrapper in use

GPFS Remote Shell: sudo wrapper in use

GPFS quorum nodes: c902f09x05,c902f09x07,c902f09x06

gpfs.syncBuffsPerIteration: 100

gpfs.dataDiskWaitTimeForRecovery: 3600

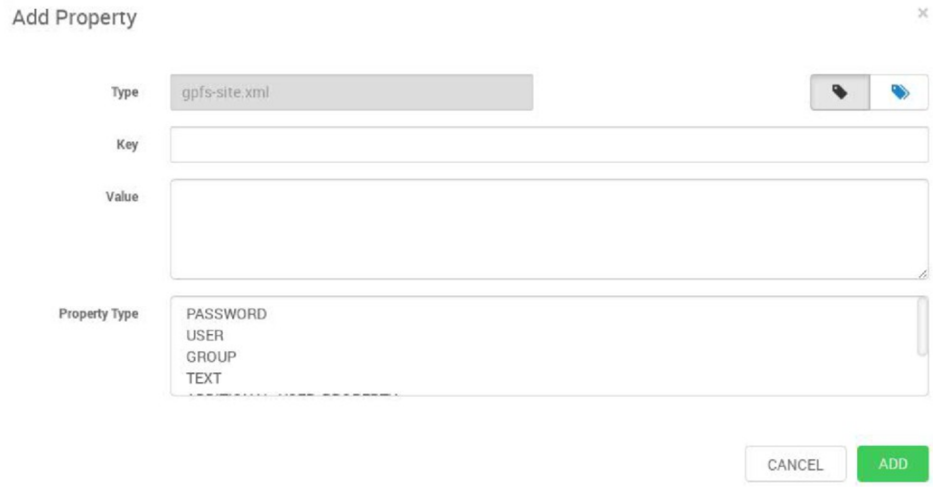
gpfs.minMissedPingTimeout

gpfs.metadataDiskWaitTimeForRecovery

bl1bda54

Limitation

Key value pairs that are newly added into the IBM Spectrum Scale management pack GUI Advanced configuration **Custom Add Property** panel do not become effective in the IBM Spectrum Scale file system. Therefore, any values not seen in the Standard or Advanced configuration panel need to be set manually on the command line using the IBM Spectrum Scale **`/usr/lpp/mmfs/bin/mmchconfig`** command.



Add Property

Type: gpfs-site.xml

Key:

Value:

Property Type:
PASSWORD
USER
GROUP
TEXT

bl1bda53

In Ambari, if any configuration in the `gpfs-site` is changed in the IBM Spectrum Scale dashboard, a Stop All[1] service followed by a Start All service is required. Check your environment to ensure that the changes made are in effect.

[1] For FPO cluster, do not run STOP ALL from the Ambari GUI. Refer to the Limitations > General sections on how to properly stop IBM Spectrum Scale.

Note: You must plan a cluster maintenance window and prepare for cluster downtime when restarting the IBM Spectrum Scale service and the HDFS service. Ensure that no I/O activities are active on the IBM Spectrum Scale file system before shutting down IBM Spectrum Scale. If the I/O activities are active, IBM Spectrum Scale fails to shut down as the kernel extension cannot be unloaded.

GPFS yum repo directory

If the IBM Spectrum Scale yum repo directory is changed, you need to update the GPFS_REPO_URL in Ambari for the upgrade process to know where the packages are located.

To update the GPFS_RPO_URL in Ambari:

1. Log in to Ambari.

2. Click **IBM Spectrum Scale service** > **Configs** > **Advanced** > **Advanced gpfs-ambari-server-env** > **GPFS_REPO_URL**, update the **GPFS_REPO_URL** value.

   Syntax: `http://<yum-server>/<REPO_DIR_LOCATION_OF_PACKAGES>`

   For example, http://c902mnx09.gpfs.net/repos/GPFS/5.0.1/gpfs_rpms
3. Save the GPFS_REPO_URL configuration.
4. The GPFS_REPO_URL becomes effective during the Upgrading IBM Spectrum Scale and Upgrading HDFS Transparency process.

HDFS and IBM Spectrum Scale restart order

When the IBM Spectrum Scale service is integrated, HDFS Transparency NameNodes and DataNodes are managed as HDFS NameNode and DataNode components respectively in the Ambari HDFS service.

When configuration is changed in IBM Spectrum Scale™, the following restart order should be followed:

1. Stop the HDFS service.
2. Restart the IBM Spectrum Scale service.
3. Restart the HDFS service.

Integrating HDFS Transparency

You must plan a cluster maintenance window, and prepare for the cluster down time when integrating the HDFS Transparency with the native HDFS. After each integration, you must run the **`ambari-server restart`** on the Ambari server node. Ensure that all the services are stopped.

To integrate the HDFS Transparency with the native HDFS:

1. On the dashboard, click **Services** > **Stop All**[1] to stop all services. Verify that all services are stopped. If not, stop the services.

   [1]For FPO cluster, do not run STOP ALL from the Ambari GUI. Refer to the Limitations > General section on how to properly stop IBM Spectrum Scale.
2. Click **Spectrum Scale** > **Actions** > **Integrate Transparency**.

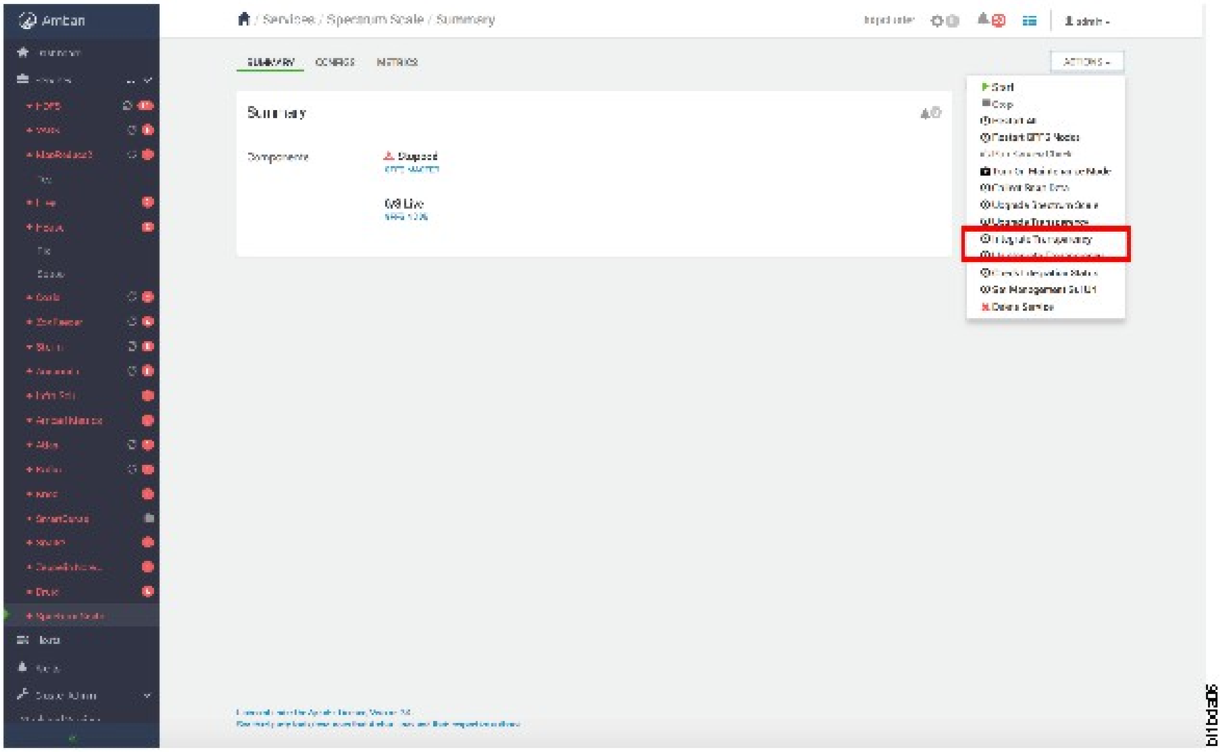
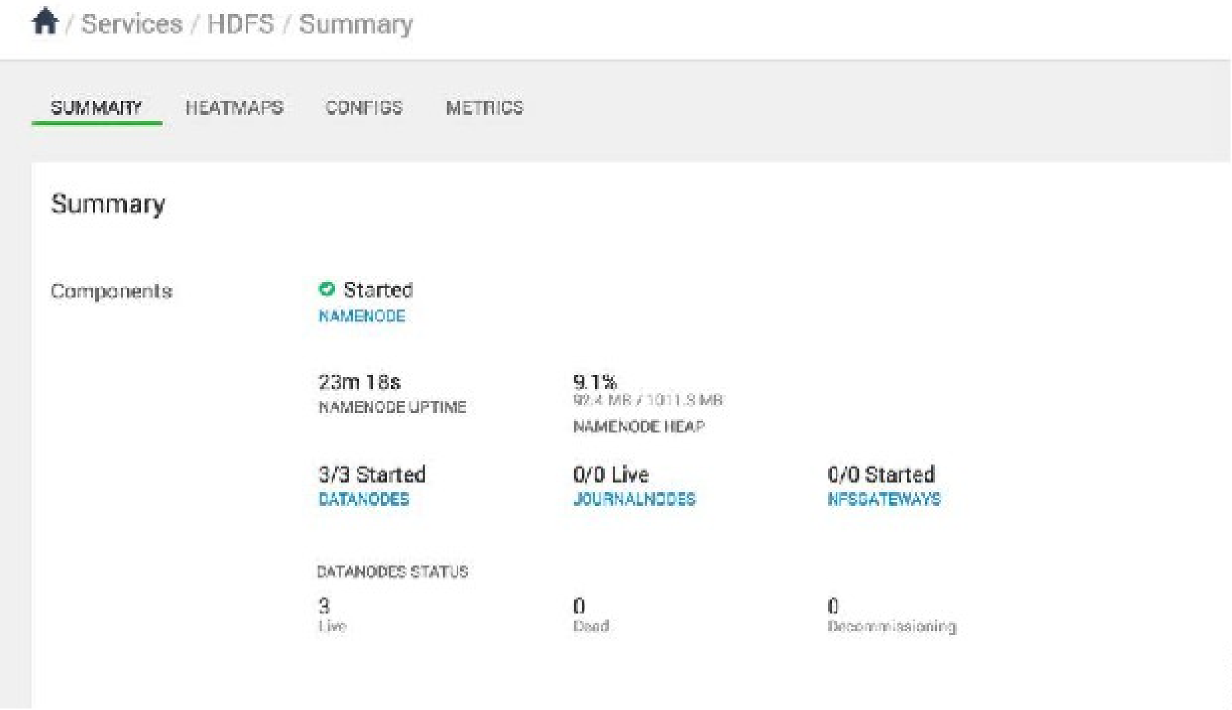


Figure 43. IBM SPECTRUM SCALE INTEGRATE TRANSPARENCY

3. On the Ambari server node, run the **ambari-server restart** command to restart the Ambari server.
4. Log back in to the Ambari GUI.
5. Start all the services from Ambari GUI. The Hadoop cluster starts using IBM Spectrum Scale and the HDFS Transparency. The HDFS dashboard displays the NameNode and DataNode status of the HDFS Transparency.

On the HDFS dashboard, check the NameNode and DataNodes status.

Note: JournalNodes are not used when IBM Spectrum Scale service is integrated.



Summary

Components

Started
NAMENODE

23m 18s
NAMENODE UPTIME

9.1%
92.4 MB / 1011.3 MB
NAMENODE HEAP

3/3 Started
DATANODES

0/0 Live
JOURNALNODES

0/0 Started
NFSGATEWAYS

DATANODES STATUS

3
Live

0
Dead

0
Decommissioning

Command verification

To verify that the HDFS Transparency is available, use the following command to check the connector state:

```
# Ensure all node GPFS state are active
/usr/lpp/mmfs/bin/mmgetstate -a
# Ensure all the NameNode and DataNodes are running.
/usr/lpp/mmfs/hadoop/sbin/mmhadoopctl connector getstate
```

For more information on how to verify the HDFS transparency integration state, see "Verifying Transparency integration state" on page 434.

Cluster environment

After the IBM Spectrum Scale service is deployed, IBM Spectrum Scale HDFS Transparency is used instead of HDFS. HDFS Transparency inherits the native HDFS configuration and adds the additional changes for the HDFS Transparency to function correctly.

After IBM Spectrum Scale is deployed, a new HDFS configuration set V2 is created, and is visible in the **HDFS Service Dashboard** > **CONFIG HISTORY**.

Unintegrating HDFS Transparency

You must plan a cluster maintenance window, and prepare for the cluster downtime while unintegrating the HDFS Transparency back to native HDFS. After each unintegration, you need to run the **`ambari-server restart`** on the Ambari server node. Ensure that all the services are stopped.

1. Log in to the Ambari GUI with the same Ambari user id that was used during the deployment of the IBM Spectrum Scale service. Usually the "admin" user id, has the administrative privileges. If a different Ambari user id is used, the Unintegrate Transparency action fails.
2. On the dashboard, click **Actions** > **Stop All**[1] to stop all services.

   [1] For FPO cluster, do not run STOP ALL from the Ambari GUI. Refer to the Limitations > General section on how to properly stop IBM Spectrum Scale.
3. If Kerberos is enabled, set the **KDC_PRINCIPAL** and **KDC_PRINCIPAL_PASSWORD** values in the **IBM Spectrum Scale services** > **Configs** > **Advanced**. Save the configuration changes.
4. Click **Spectrum Scale** > **Actions** > **Unintegrate Transparency**.

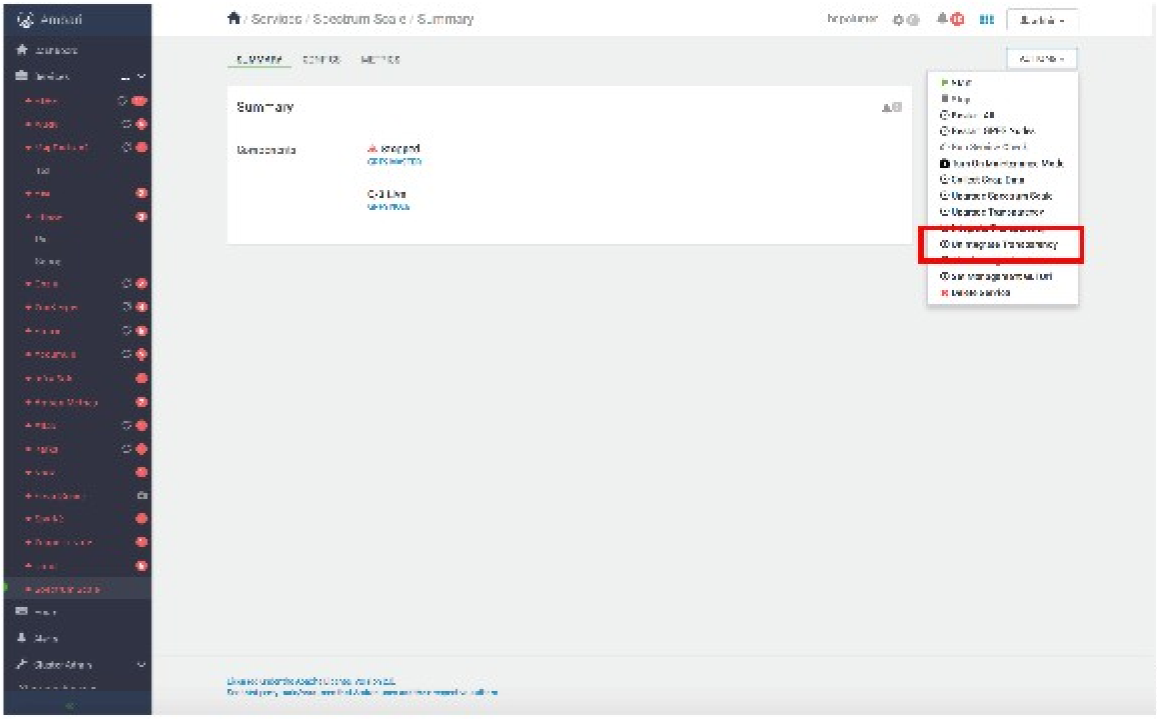


Figure 44. IBM SPECTRUM SCALE UNINTEGRATE TRANSPARENCY

5. On the Ambari server node, run the **`ambari-server restart`** command to restart the Ambari server.
6. Log back in to the Ambari GUI.
7. Start all services from the Ambari GUI. The Hadoop cluster starts using native HDFS. The IBM Spectrum Scale service is not removed from the Ambari panel, and will be displayed in GREEN. IBM Spectrum Scale will function, but the HDFS Transparency will not function.

   Note: When unintegrated back to native HDFS, the HDFS configuration used remains the same as the HDFS configuration used by the IBM Spectrum Scale prior to unintegration. If you must revert to the original HDFS configuration, go to the HDFS dashboard, and make the configuration changes in the **Configs** tab.

Command verification

To verify that the HDFS Transparency is not available, use the following command to check the connector state:

```
# Ensure all node GPFS state are active
/usr/lpp/mmfs/bin/mmgetstate -a
# Ensure no Transparency NameNode and DataNodes are running.
/usr/lpp/mmfs/hadoop/sbin/mmhadoopctl connector getstate
```

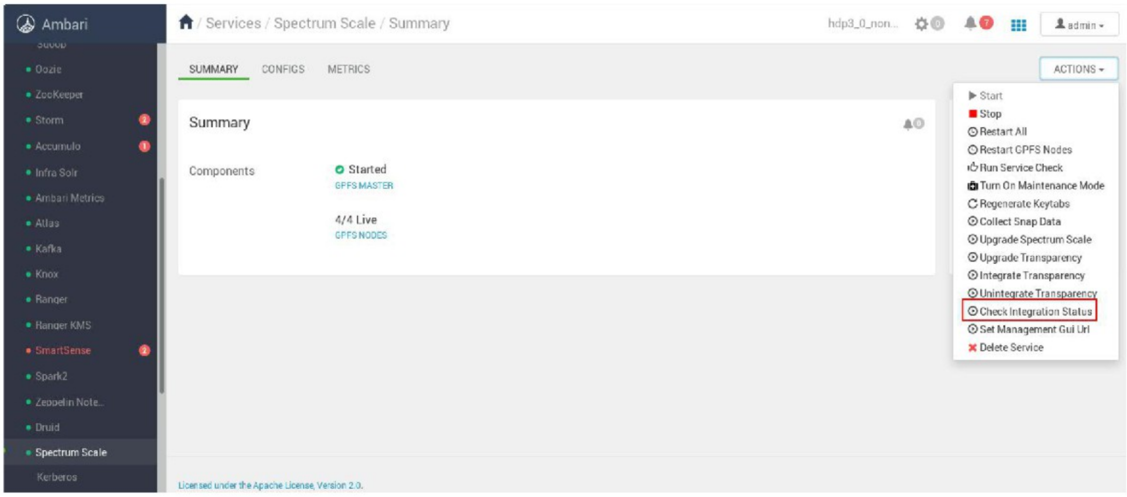
Cluster environment

After using the IBM Spectrum Scale Unintegrate Transparency function, the native HDFS will be in effect. The configuration from HDFS service before the unintegrate phase will still be in effect. The IBM Spectrum Scale configuration will not affect the native HDFS functionality. If you must revert back to the original native HDFS configuration, go to the HDFS dashboard, and select the V1 configuration version under the **Configs** tab.

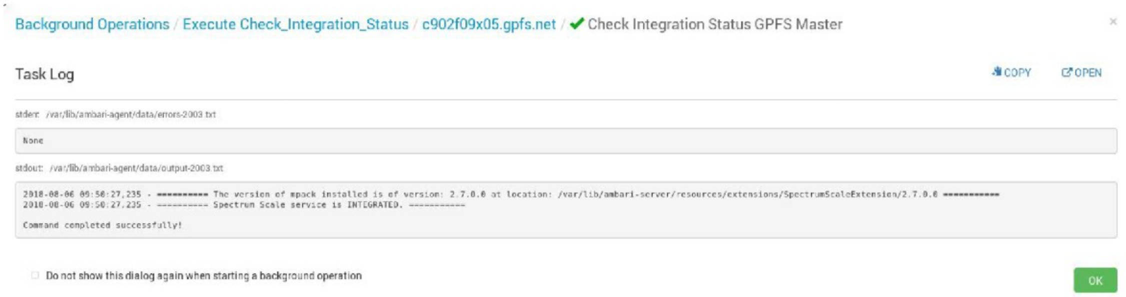
For information on verifying the Transparency integration state, see "Verifying Transparency integration state" on page 434.

Verifying Transparency integration state

To verify the HDFS Transparency integration state, click **Ambari GUI** > **Spectrum Scale** > **Actions** > **Check Integration Status**.



After the process completes, check the output log for the state information.


Background Operations / Execute Check_Integration_Status / c902f09x05.gpfs.net / ✓ Check Integration Status GPFS Master
Task Log
COPY
OPEN
OK
bl1bda55

stderr: /var/lib/ambari-agent/data/errors-2003.txt

```
None
```

stdout: /var/lib/ambari-agent/data/output-2003.txt

```
2018-08-06 09:50:27,235 - ========== The version of mpack installed is of version: 2.7.0.0 at location: /var/lib/ambari-server/resources/extensions/SpectrumScaleExtension/2.7.0.0 ==========
2018-08-06 09:50:27,235 - ========== Spectrum Scale service is INTEGRATED. ==========

Command completed successfully!
```

□ Do not show this dialog again when starting a background operation

Verify IBM Spectrum Scale Mpack version

This topic describes how to verify the IBM Spectrum Scale Mpack version.

To verify the IBM Spectrum Scale Mpack version, click **Ambari GUI** > **Spectrum Scale** > **Actions** > **Check Integration Status**.

Ambari node management

This section provides information to add, delete, move, and set up a node in Ambari.

Adding a host

This topic provides information to add a new IBM Spectrum Scale node. The IBM Spectrum Scale node can be an IBM Spectrum Scale client, HDFS Transparency NameNode or DataNode.

See Preparing the environment section to prepare the new nodes.

Note:

- Ensure that the IBM Spectrum Scale service is in integrated state before adding the node.
- On the new host being added, create a userid and groupid called *anonymous* with the same value as all the other GPFS nodes. For more information, see "Create the anonymous user id" on page 352.
- If you are adding new nodes to an existing cluster, and if the nodes being added already have IBM Spectrum Scale installed on them, ensure that the new nodes are at the same version of IBM Spectrum Scale as the existing cluster. Do not mix GPFS Nodes with different versions of IBM Spectrum Scale software in a GPFS cluster.

  If you are adding a new node to an existing cluster with inconsistent IBM Spectrum Scale versions, the new node will not install even if the failed installed node might still be displayed in the cluster list in Ambari. To delete the failed node from the cluster in Ambari, see "Deleting a host" on page 442.

  The new nodes can then be added to the Ambari cluster by using the Ambari web interface.

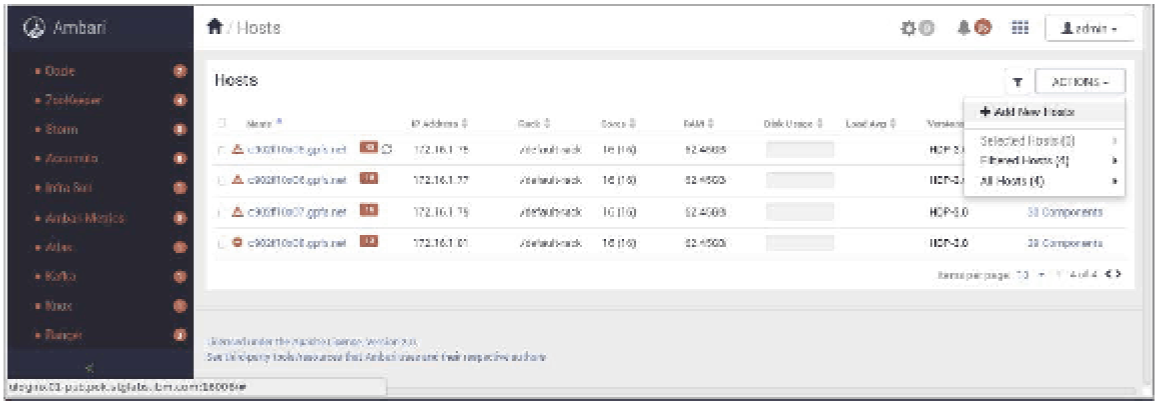
  For more information, see "Adding GPFS node component" on page 441.
- If the IBM Spectrum Scale cluster is configured in admin mode central mode, following steps need to be performed as prerequisite:
  1. On the node to be added, execute:
     a. Install all the GPFS packages.
     b. Build the GPL layer using: `/usr/lpp/mmfs/bin/mmbuildgpl`.
  2. On the admin node (usually the ambari server node) execute:
     a. `/usr/lpp/mmfs/bin/mmaddnode -N <FQDN-of-new-node>`
     b. `/usr/lpp/mmfs/bin/mmchlicense server --accept -N <FQDN-of-new-node>`
     c. `/usr/lpp/mmfs/bin/mmmount all`
     d. `/usr/lpp/mmfs/bin/mmlsmount all`
- If the host that is to be added already has HDFS Transparency installed and configured, and you want to add this host to an existing HDFS Transparency cluster through Ambari, you need to erase the HDFS Transparency packages and configuration files on the host that is to be added. This is to ensure that the GPFS_Node component install step does not fail because of the stale configuration information.

  Perform the following steps for cleaning up stale configuration on the host to be added:
  1. Uninstall the existing HDFS Transparency package by running the following command:

     ```
     # yum erase gpfs.hdfs-protocol
     ```
  2. Remove all the HDFS Transparency configuration XML files under the `/var/mmfs/hadoop/etc/hadoop/` directory.

The new nodes can then be added to the Ambari cluster by using the Ambari web interface.

1. On the Ambari dashboard, click **Hosts** > **Actions** > **Add New Hosts**.



Name	IP Address	Rack	Cores	RAM	Disk Usage	Load Avg	Versions	
c902f10x05.gpfs.net	172.16.1.75	/default-rack	16 (16)	62.46GB			HDP-3.0	
c902f10x06.gpfs.net	172.16.1.77	/default-rack	16 (16)	62.46GB			HDP-3.0	
c902f10x07.gpfs.net	172.16.1.79	/default-rack	16 (16)	62.46GB			HDP-3.0	33 Components
c902f10x08.gpfs.net	172.16.1.81	/default-rack	16 (16)	62.46GB			HDP-3.0	33 Components

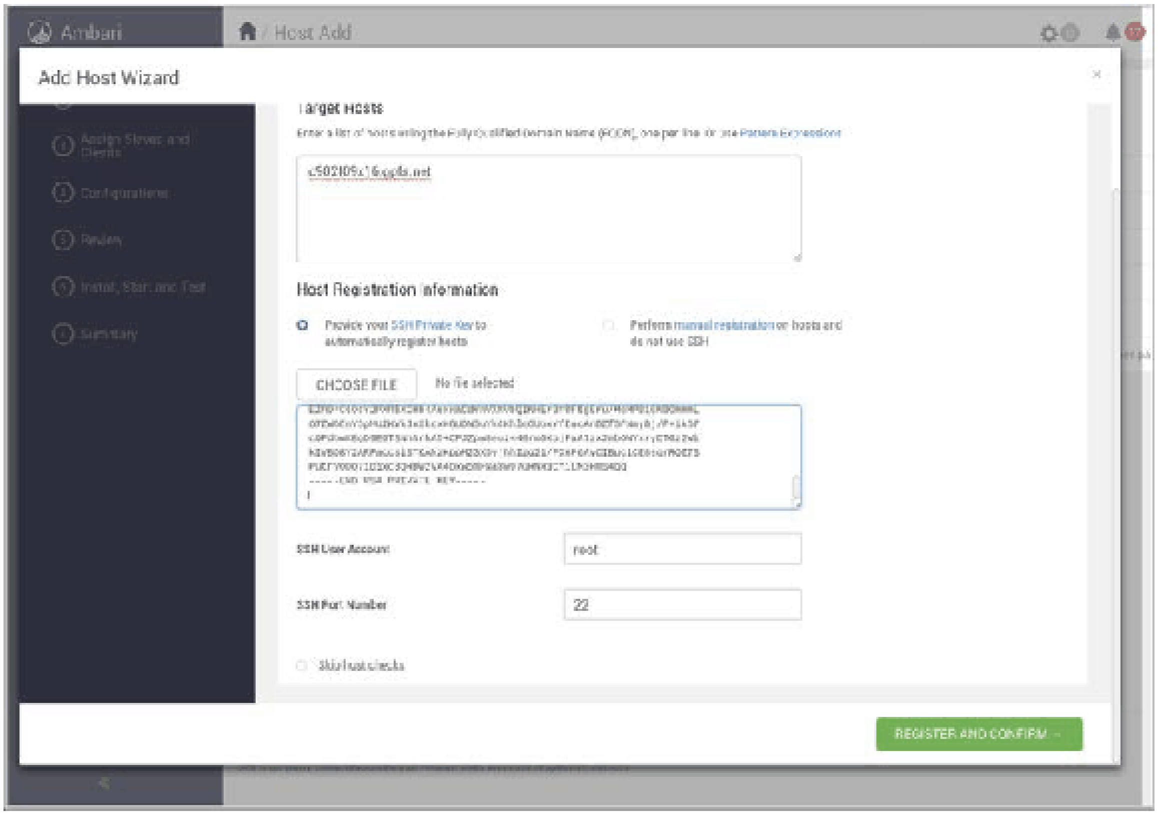
Items per page: 10 1 - 4 of 4

Licensed under the Apache License, Version 2.0.
See third-party tools/resources that Ambari uses and their respective authors

2. Specify the new node information, and click **Registration and Confirm**.

   Note:

   - The SSH Private Key is the key of the user on the Ambari Server.
   - If the warning is due to user id already existing and these are the user ids that were predefined for the cluster, then the warning can be ignored. Otherwise, if there are other host check failures, then check for the failure by clicking on the link and follow the directions in the pop up window.



Target Hosts

Enter a list of hosts using the Fully Qualified Domain Name (FQDN), one per line. Or use Pattern Expressions

c502f09x16.gpfs.net

Host Registration Information

● Provide your SSH Private Key to automatically register hosts

○ Perform manual registration on hosts and do not use SSH

No file selected

-----END RSA PRIVATE KEY-----

SSH User Account: root

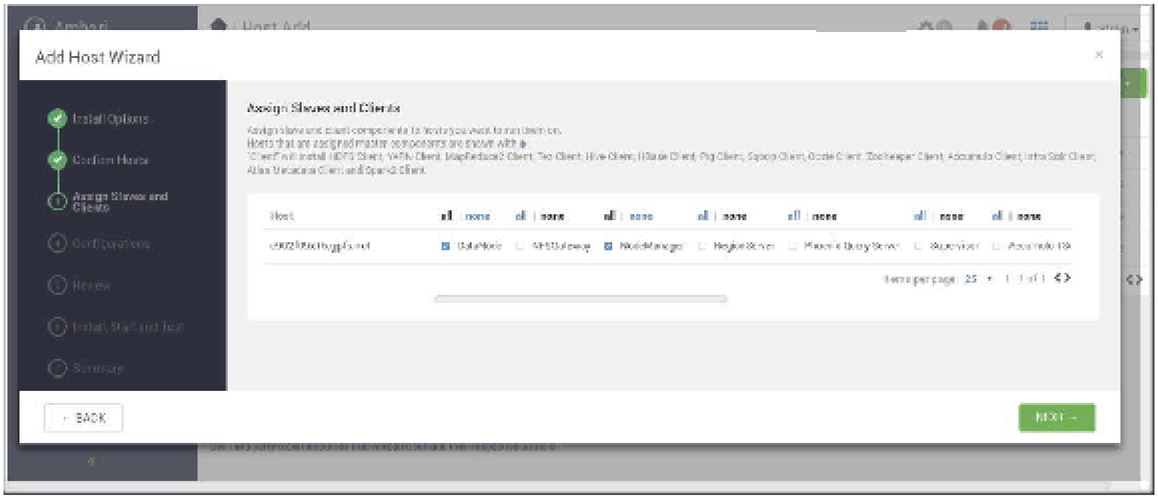
SSH Port Number: 22

□ Skip host checks

3. Select the services that you want to install on the new node.

   Note:

   - If HDFS Transparency DataNode is needed on a host, select DataNode, NodeManager, and GPFS Node components for that host.
   - If you want only the IBM Spectrum Scale client and not the HDFS Transparency components on a host, select only the GPFS Node component.

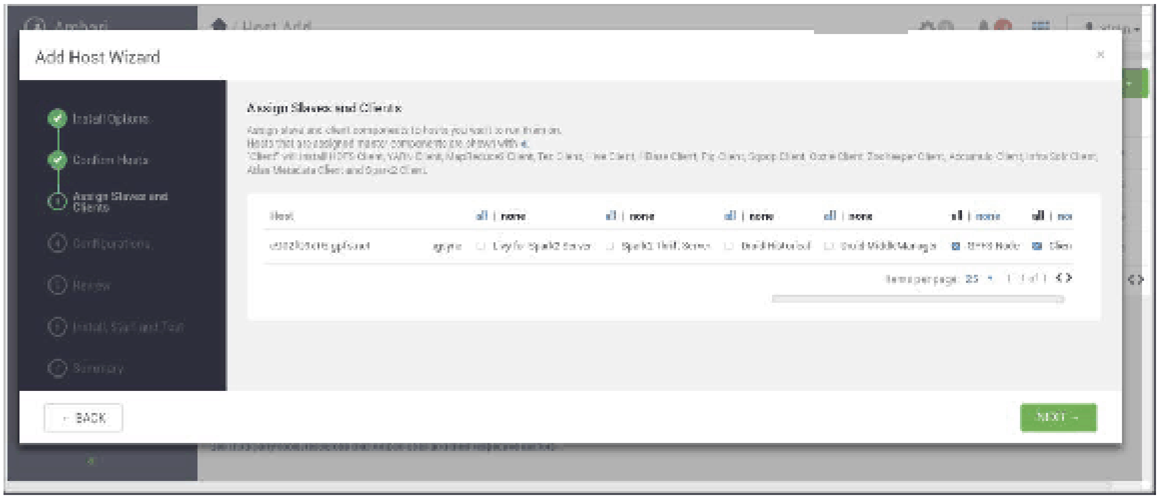


Assign Slaves and Clients

Assign slave and client components to hosts you want to run them on.
Hosts that are assigned master components are shown with ✱.
"Client" will install HDFS Client, YARN Client, MapReduce2 Client, Tez Client, Hive Client, HBase Client, Pig Client, Sqoop Client, Oozie Client, ZooKeeper Client, Accumulo Client, Infra Solr Client, Atlas Metadata Client and Spark2 Client.

Host	all | none	all | none	all | none	all | none	all | none	all | none	all | none
[illegible]	☒ DataNode	□ NFSGateway	☒ NodeManager	□ RegionServer	□ Phoenix Query Server	□ Supervisor	□ Accumulo TS

Items per page: 25 ▾ 1 - 1 of 1 ‹ ›



Assign Slaves and Clients

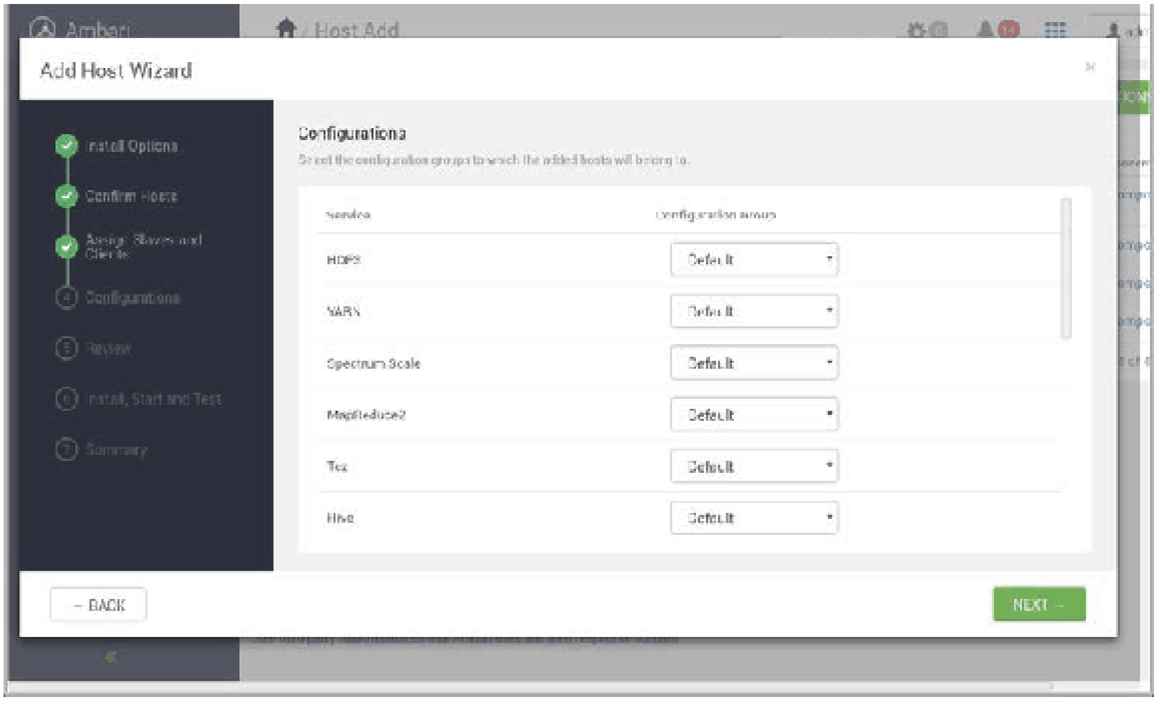
Assign slave and client components to hosts you want to run them on.
Hosts that are assigned master components are shown with ✱.
"Client" will install HDFS Client, YARN Client, MapReduce2 Client, Tez Client, Hive Client, HBase Client, Pig Client, Sqoop Client, Oozie Client, ZooKeeper Client, Accumulo Client, Infra Solr Client, Atlas Metadata Client and Spark2 Client.

Host		all | none	all | none	all | none	all | none	all | none	all | no
[illegible]	agyne	□ Livy for Spark2 Server	□ Spark2 Thrift Server	□ Druid Historical	□ Druid MiddleManager	☒ GPFS Node	☒ Clien

Items per page: 25 ▾ 1 - 1 of 1 ‹ ›

For more information, see "Adding GPFS node component" on page 441.

4. If several configuration groups are created, select one of them for the new node.

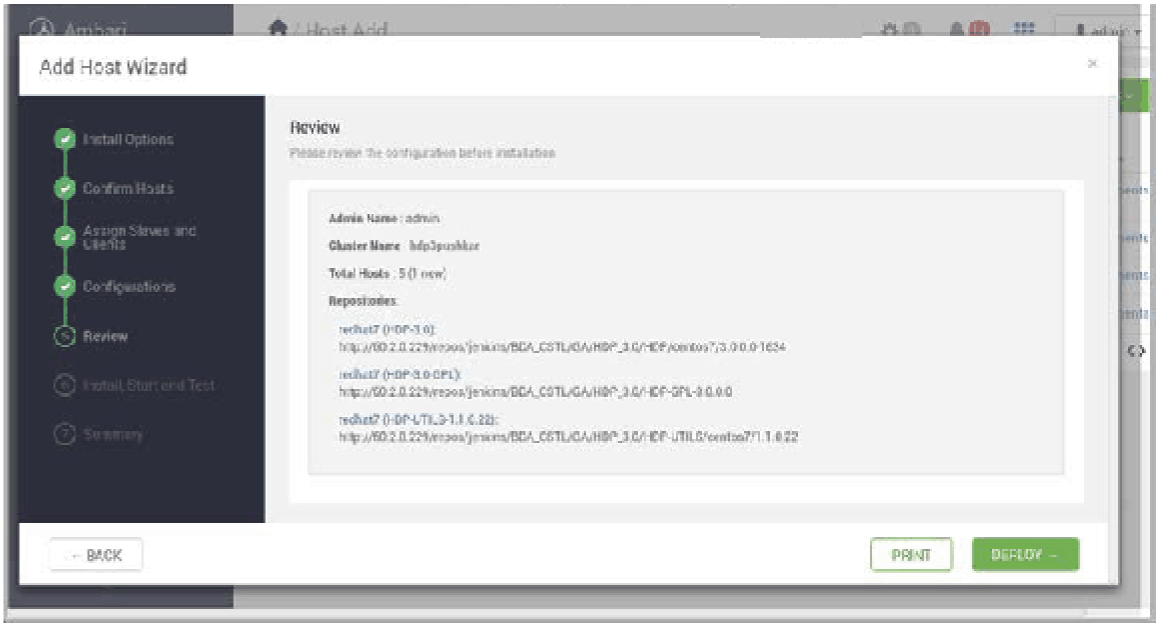


Configurations

Select the configuration groups to which the added hosts will belong to.

Service	Configuration groups
HDFS	Default
YARN	Default
Spectrum Scale	Default
MapReduce2	Default
Tez	Default
Hive	Default

5. Review the information and start the deployment by clicking **Deploy**.



Review

Please review the configuration before installation

Admin Name : admin

Cluster Name : hdp3pushkar

Total Hosts : 5 (1 new)

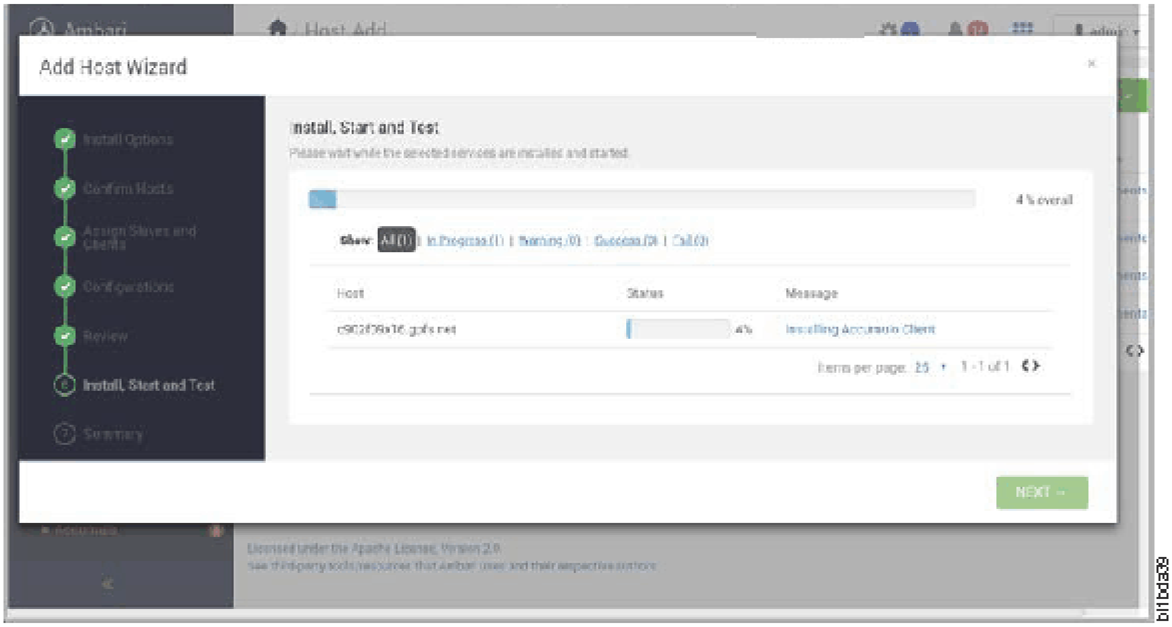
Repositories:

redhat7 (HDP-3.0):
http://60.2.0.229/repos/jenkins/BCA_CSTL/GA/HDP_3.0/HDP/centos7/3.0.0.0-1634

redhat7 (HDP-3.0-GPL):
http://60.2.0.229/repos/jenkins/BCA_CSTL/GA/HDP_3.0/HDP-GPL-3.0.0.0

redhat7 (HDP-UTILS-1.1.0.22):
http://60.2.0.229/repos/jenkins/BCA_CSTL/GA/HDP_3.0/HDP-UTILS/centos7/1.1.0.22

6. Install, Start and Test panel.



Install, Start and Test

Please wait while the selected services are installed and started.

4 % overall

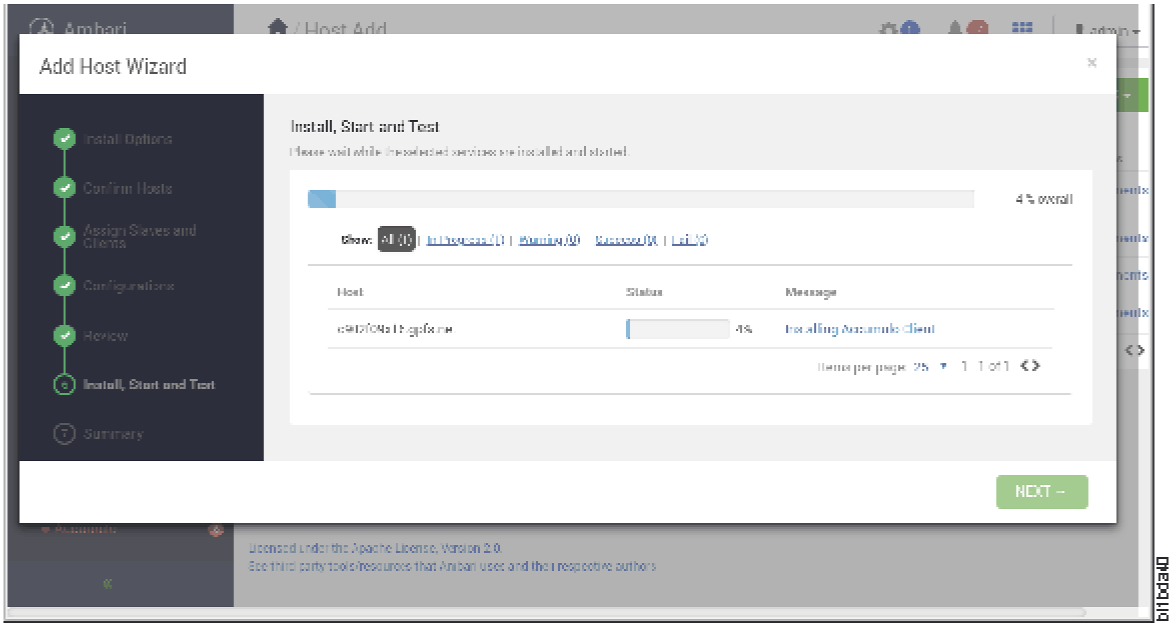
Show: All (1) | In Progress (1) | Warning (0) | Success (0) | Fail (0)

Host	Status	Message
c902f09x16.gpfs.net	4%	Installing Accumulo Client

Items per page: 25 1 - 1 of 1

Licensed under the Apache License, Version 2.0.
See third-party tools/resources that Ambari uses and their respective authors

7. After the Install, Start and Test wizard finishes, click **Complete**.



Install, Start and Test

Please wait while the selected services are installed and started.

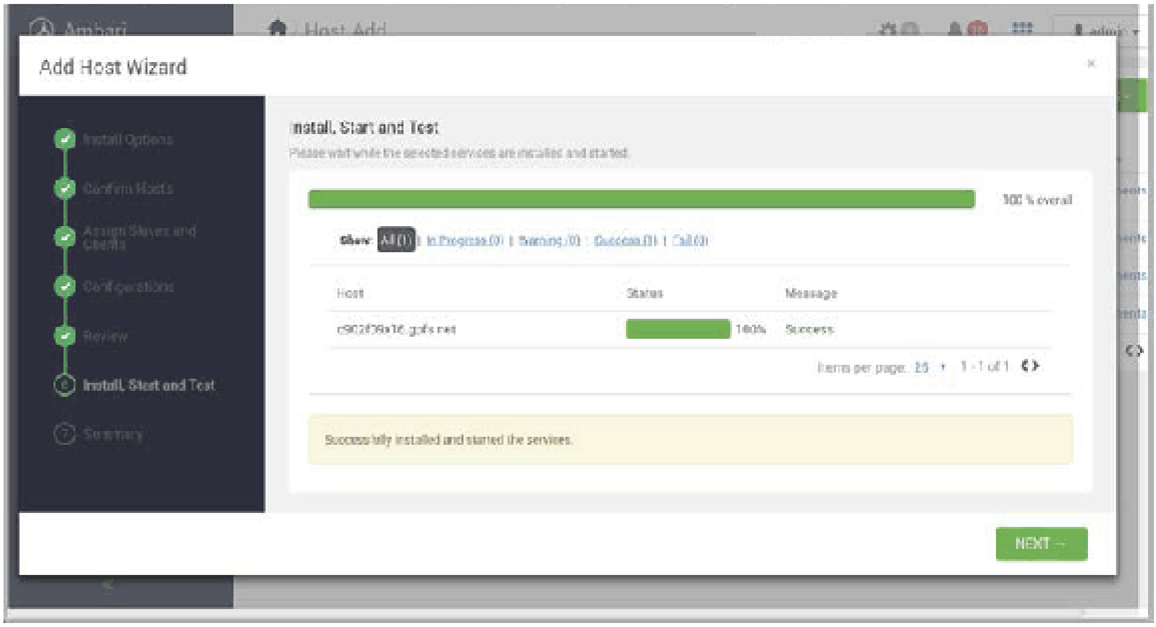
4 % overall

Show: All (1) | In Progress (1) | Warning (0) | Success (0) | Fail (0)

Host	Status	Message
c902f09x16.gpfs.net	4%	Installing Accumulo Client

Items per page: 25 1 - 1 of 1

Licensed under the Apache License, Version 2.0.
See third-party tools/resources that Ambari uses and their respective authors



Add Host Wizard

Install, Start and Test

Please wait while the selected services are installed and started.

100 % overall

Show: All (1) | In Progress (0) | Warning (0) | Success (1) | Fail (0)

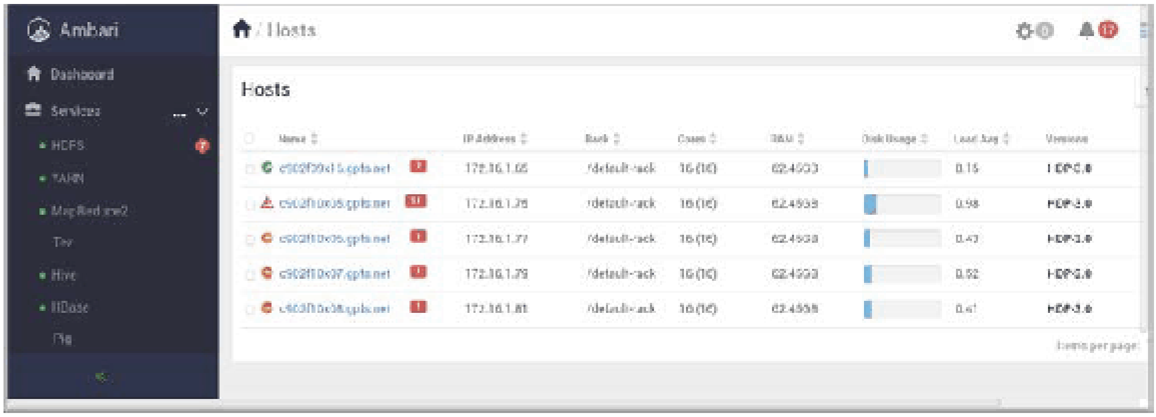
Host	Status	Message
c902f09x16.gpfs.net	100%	Success

Items per page: 25 1 - 1 of 1

Successfully installed and started the services.

8. A new node is added to the Ambari cluster.

   From Hosts dashboard, the new node is added to the host list.



Hosts

Name	IP Address	Rack	Cores	RAM	Disk Usage	Load Avg	Versions
c902f09x15.gpfs.net	172.16.1.65	/default-rack	16 (16)	62.45GB		0.15	HDP-3.0
c902f10x06.gpfs.net	172.16.1.76	/default-rack	16 (16)	62.45GB		0.98	HDP-3.0
c902f10x05.gpfs.net	172.16.1.77	/default-rack	16 (16)	62.45GB		0.43	HDP-3.0
c902f10x07.gpfs.net	172.16.1.79	/default-rack	16 (16)	62.45GB		0.92	HDP-3.0
c902f10x08.gpfs.net	172.16.1.81	/default-rack	16 (16)	62.45GB		0.41	HDP-3.0

9. For any service with the restart required icon, go to the service dashboard, select **Restart** > **Restart All Affected**.

   Note: Ambari does not create NSDs on the new nodes. To create IBM Spectrum Scale NSDs and add NSDs to the file system, follow the steps under the *Adding disks to a file system* topic in the *IBM Storage Scale: Administration Guide*.

10. Restart HDFS service in Ambari.

    Check the cluster information.

    Note: In case of FPO, Ambari does not create NSDs on the new nodes. To create IBM Spectrum Scale NSDs and add NSDs to the file system, follow the steps under the *Adding disks to a file system* topic in the *IBM Storage Scale: Administration Guide*.

    Check the cluster information.

```
[root@c902f05x01 ~]# /usr/lpp/mmfs/bin/mmlscluster

GPFS cluster information
========================
  GPFS cluster name:         bigpfs.gpfs.net
  GPFS cluster id:           8678991139790049774
  GPFS UID domain:           bigpfs.gpfs.net
  Remote shell command:      /usr/bin/ssh
  Remote file copy command:  /usr/bin/scp
  Repository type:           CCR

Node  Daemon node name     IP address   Admin node name      Designation
--------------------------------------------------------------------------
   1   c902f05x01.gpfs.net  192.0.2.11  c902f05x01.gpfs.net  quorum
   2   c902f05x04.gpfs.net  192.0.2.17  c902f05x04.gpfs.net  quorum
   3   c902f05x03.gpfs.net  192.0.2.15  c902f05x03.gpfs.net  quorum
   4   c902f05x02.gpfs.net  192.0.2.13  c902f05x02.gpfs.net
   5   c902f05x05.gpfs.net  192.0.2.19  c902f05x05.gpfs.net

[root@c902f05x01 ~]#

[root@c902f05x01 ~]# /usr/lpp/mmfs/bin/mmgetstate -a

Node number  Node name        GPFS state
------------------------------------------
       1      c902f05x01       active
       2      c902f05x04       active
       3      c902f05x03       active
       4      c902f05x02       active
       5      c902f05x05       active

[root@c902f05x01 ~]#

[root@c902f05x01 ~]# /usr/lpp/mmfs/bin/mmlsnsd

File system   Disk name    NSD servers
---------------------------------------------------------------------------
bigpfs        gpfs1nsd     c902f05x01.gpfs.net
bigpfs        gpfs2nsd     c902f05x02.gpfs.net
bigpfs        gpfs3nsd     c902f05x03.gpfs.net
bigpfs        gpfs4nsd     c902f05x04.gpfs.net
bigpfs        gpfs5nsd     c902f05x03.gpfs.net
bigpfs        gpfs6nsd     c902f05x02.gpfs.net
bigpfs        gpfs7nsd     c902f05x01.gpfs.net
bigpfs        gpfs8nsd     c902f05x04.gpfs.net
bigpfs        gpfs9nsd     c902f05x02.gpfs.net
bigpfs        gpfs10nsd    c902f05x03.gpfs.net
bigpfs        gpfs11nsd    c902f05x04.gpfs.net
bigpfs        gpfs12nsd    c902f05x01.gpfs.net
bigpfs        gpfs13nsd    c902f05x02.gpfs.net
bigpfs        gpfs14nsd    c902f05x03.gpfs.net
bigpfs        gpfs15nsd    c902f05x04.gpfs.net
bigpfs        gpfs16nsd    c902f05x01.gpfs.net

[root@c902f05x01 ~]#

[root@c902f05x05 ~]# mount | grep bigpfs
bigpfs on /bigpfs type gpfs (rw,relatime)
[root@c902f05x05 ~]#

[root@c902f05x01 ~]# /usr/lpp/mmfs/hadoop/sbin/mmhadoopctl connector getstate

c902f05x01.gpfs.net: namenode running as process 17599.
c902f05x01.gpfs.net: datanode running as process 21978.
c902f05x05.gpfs.net: datanode running as process 5869.
c902f05x04.gpfs.net: datanode running as process 25002.
c902f05x03.gpfs.net: datanode running as process 10908.
c902f05x02.gpfs.net: datanode running as process 6264.
[root@c902f05x01 ~]#
```

Adding GPFS node component

The GPFS node component setting in the **Ambari Assign Slaves and Clients** panel is to set the host to install the IBM Spectrum Scale packages.

This setting should be enabled in the following scenarios:

- While adding new IBM Spectrum Scale clients, NameNodes or Data Nodes.
- The GPFS Node component was not selected next to the host during the deployment and the host requires the IBM Spectrum Scale packages.

For information on adding hosts through Ambari, see "Adding a host" on page 435.

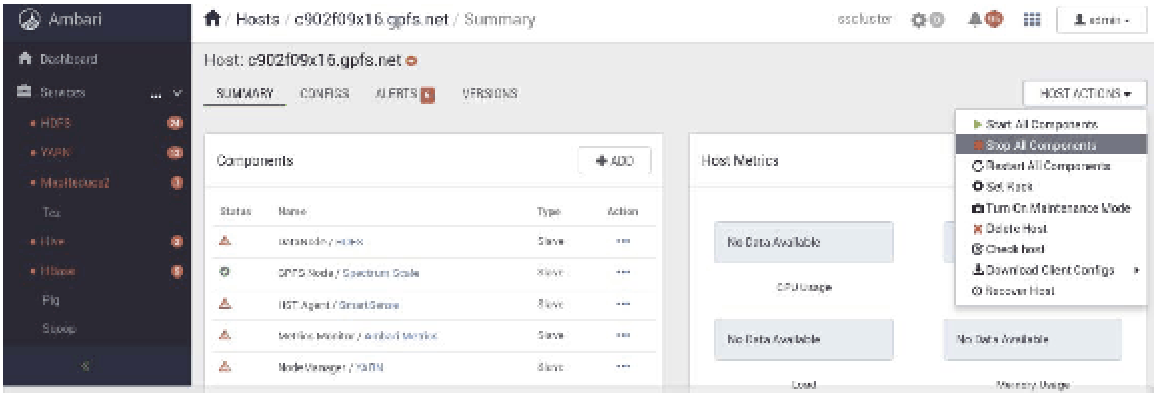
1. On Ambari dashboard, select **Hosts** > **Choose host** > **Components** > **Add** (Choose GPFS Node component).
2. Log back into Ambari.
3. From the dashboard, select **HDFS** > **Actions** > **Restart All**.

Deleting a host

This topic provides information on how to delete a node.

1. Stop all the components on the node to be deleted.

   For example: `c902f09x16.gpfs.net`
2. From Ambari dashboard, click **Hosts tab**, and select *the host that must be removed* and then click **Host Actions** > **Stop All Components**.

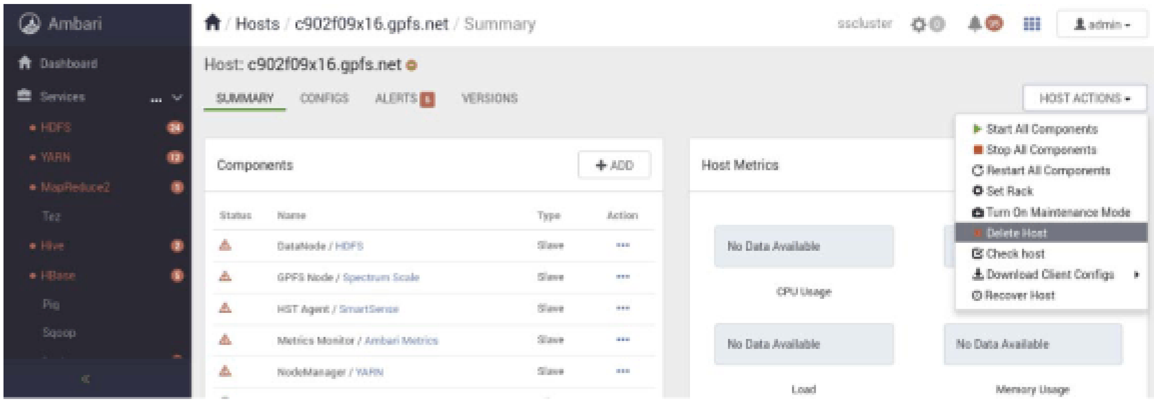


bl1bda28

3. Stop the Ambari agent on the host to be deleted.

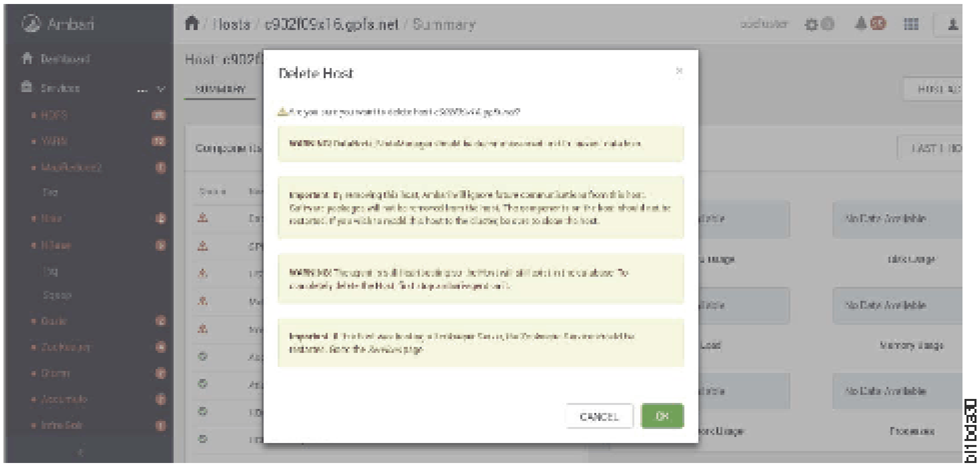
```
[root@c902f09x16 ~]
# ambari-agent stop
Verifying Python version compatibility...
Using python  /usr/bin/python2
Found ambari-agent PID: 22182
Stopping ambari-agent
Removing PID file at
/var/run/ambari-agent/ambari-agent.pid
ambari-agent successfully stopped
[root@c902f09x16 ~]#
```

4. To delete the host, click **Host Actions** > **Delete Hosts**.



bl1bda29

The system displays a Warning message.



Delete Host

Are you sure you want to delete host c902f09x16.gpfs.net?

WARNING! DataNode, NodeManager should be decommissioned [illegible]

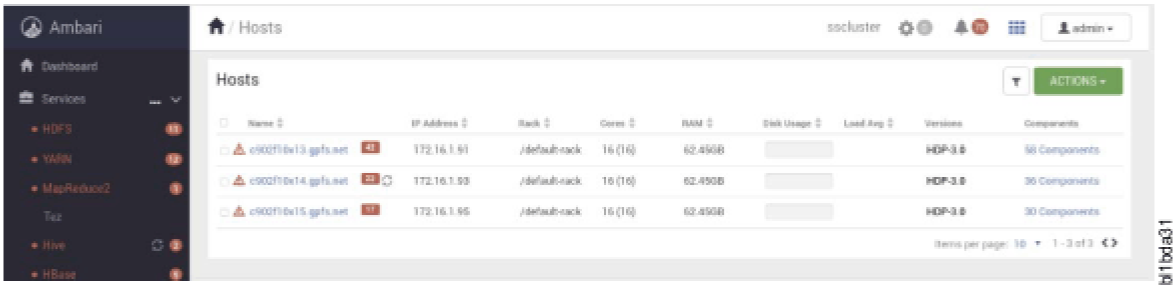
Important: By removing this host, Ambari will ignore future communications from this host. Software packages will not be removed from the host. The components on the host should not be restarted. If you wish to readd the host to the cluster, be sure to clean the host.

WARNING! The agent on all host(s) [illegible]. To completely delete the Host, first stop ambari-agent on it.

Important: If the host was hosting a Zookeeper Server, the Zookeeper Service should be restarted. Go to the Services page.

5. Click **OK**.

   The node is deleted from the Hosts list.



Hosts

Name	IP Address	Rack	Cores	RAM	Disk Usage	Load Avg	Versions	Components
c902f10x13.gpfs.net	172.16.1.91	/default-rack	16 (16)	62.45GB			HDP-3.0	58 Components
c902f10x14.gpfs.net	172.16.1.93	/default-rack	16 (16)	62.45GB			HDP-3.0	36 Components
c902f10x15.gpfs.net	172.16.1.95	/default-rack	16 (16)	62.45GB			HDP-3.0	30 Components

Items per page: 10 1 - 3 of 3

6. Restart the HDFS service.

   You must plan a cluster maintenance window, and prepare for the cluster downtime when restarting the HDFS service.

   To restart the HDFS service, follow the steps listed below:

   a. From the dashboard, select **HDFS** > **Actions** > **Restart All**.
   b. After the HDFS service restarts, the deleted host is removed from the HDFS Transparency. The DataNodes status is 4/4 started, and the DataNodes Status is 4 live.

   Note: This does not remove the Ambari packages and the IBM Spectrum Scale™ packages and NSD disks. It does not remove the node from the GPFS cluster either. Follow the IBM Spectrum Scale documentation on removing disks and packages from the environment.

```
[root@c902f10x13 ~]# /usr/lpp/mmfs/bin/mmgetstate -a

Node number  Node name        GPFS state
-------------------------------------------
      1      c902f10x13       active
      2      c902f10x14       active
      3      c902f10x15       active
      4      c902f09x16       down
[root@c902f10x13 ~]#
[root@c902f10x13 ~]# /usr/lpp/mmfs/hadoop/sbin/mmhadoopctl connector getstate
c902f10x13.gpfs.net: namenode running as process 3413.
c902f10x14.gpfs.net: namenode running as process 5237.
c902f10x15.gpfs.net: datanode running as process 24456.
c902f10x13.gpfs.net: datanode running as process 15439.
c902f10x14.gpfs.net: datanode running as process 17884.
[root@c902f10x13 ~]#
```

Moving a NameNode

IBM Spectrum Scale HDFS Transparency NameNode is stateless, and does not maintain the FSimage-like information. The **`move NameNode`** option is not supported by the Ambari HDFS GUI when HDFS Transparency is integrated with the installed management pack version 4.1-X and later.

Note:

- The move NameNode script can only be run as a root.
- The move NameNode script can be executed in a Kerberized environment when the IBM Spectrum Scale service is integrated.

Note: When the HDFS Transparency is integrated, the Move NameNode option sets the new NameNode to be the same value for both the HDFS NameNode and the HDFS Transparency NameNode.

For example,

Environment

HDFS Transparency = Integrated

HDFS NameNode = c902f09x02

HDFS Transparency NameNode = c902f09x02

- Execute Move NameNode:

  Current NameNode (c902f09x02) will be moved to a new NameNode (c902f09x03)

Environment

HDFS Transparency = Integrated

HDFS NameNode = c902f09x03

HDFS Transparency NameNode = c902f09x03

Note: If the HDFS Transparency is unintegrated, the native HDFS NameNode must still have the same Move NameNode host value as when it was integrated. Therefore, do not run the Move NameNode service after the HDFS Transparency is unintegrated for the same Move NameNode host. For instructions on how to properly use native HDFS after unintegration, see section "Revert to native HDFS after move NameNode" on page 445.

Move NameNode in integrated state

This section provides the steps to move NameNode when the IBM Spectrum Scale service is integrated.

Instructions for HA cluster

This topic describes the steps to manually move a NameNode when HDFS Transparency is in integrate state.

1. From the dashboard, select **Actions** > **Stop All**[1].

   [1] For FPO cluster, do not run STOP ALL from the Ambari GUI. Refer to the Limitations > General section on how to properly stop IBM Spectrum Scale.
2. On the Ambari server host, run the following command:

   ```
   python /var/lib/ambari-server/resources/mpacks/SpectrumScaleExtension-MPack-
   <version>/extensions/SpectrumScaleExtension/<version>/
   services/GPFS/package/files/COMMON/MoveNameNodeTransparency.py
   ```

   Follow the command prompts and type the required input.

   ```
   $ python /var/lib/ambari-server/resources/mpacks/SpectrumScaleExtension-MPack-2.7.0.0/
   extensions/
   SpectrumScaleExtension/2.7.0.0/services/GPFS/package/files/COMMON/MoveNameNodeTransparency.py
   Enter the Ambari Server User:(Default User admin ):
   Enter the Password for Ambari Server.
   Password:
   Retype password:
   ```

```
SSL Enabled (True/False) (Default False):
Enter the Ambari Server Port.(Default 8080)
Enter the Fully Qualified HostName of the Source NameNode which has to be Removed:-
c902f09x02.gpfs.net
Enter the Fully Qualified HostName of the Destination NameNode has to be Added:
c902f09x03.gpfs.net
```

Note:

- SSL Enabled means Ambari HTTPS.
- The source NameNode must be one of the NameNodes when HA is enabled, and the destination must be one of HDFS Transparency node.

3. From the dashboard, select **Actions** > **Start All**.
4. The process of moving the NameNode is now completed. Verify that the Active NameNode and the Standby NameNode are correct.

Instructions for a non-HA cluster
This topic provides the steps to manually move the NameNode when HDFS Transparency is in integrate state.

1. On the dashboard, click **Actions** > **Stop All**[1].

   [1] For FPO cluster, do not run STOP ALL from the Ambari GUI. Refer to the Limitations > General sections on how to properly stop IBM Spectrum Scale.

2. On the Ambari server host, run the following command:

   ```
   python /var/lib/ambari-server/resources/mpacks/SpectrumScaleExtension-MPack-
   <version>/extensions/SpectrumScaleExtension/<version>/services/GPFS/package/files/COMMON/
   MoveNameNodeTransparency.py
   ```

   Follow the command prompts and type the required input.

   ```
   $ python /var/lib/ambari-server/resources/mpacks/SpectrumScaleExtension-MPack-2.7.0.0/
   extensions/SpectrumScaleExtension/2.7.0.0/services/GPFS/package/files/COMMON/
   MoveNameNodeTransparency.py
   Enter the Ambari Server User:(Default User admin ):
   Enter the Password for Ambari Server.
   Password:
   Retype password:
   SSL Enabled (True/False) (Default False):
   Enter the Ambari Server Port.(Default 8080)
   Enter the Fully Qualified HostName of the Source
   NameNode which has to be Removed:c902f09x02.gpfs.net
   Enter the Fully Qualified HostName of the Destination
   NameNode has to be Added:c902f09x03.gpfs.net
   ```

   Note:

   - SSL Enabled means Ambari HTTPS.
   - The destination node must be one of the HDFS Transparency node.

3. From the dashboard, select **Actions** > **Start All**.
4. Moving the NameNode process is now completed. Verify that the Active NameNode and the Standby NameNode are correct.

Revert to native HDFS after move NameNode

The **move Namenode** is executed when IBM Spectrum Scale is integrated using HDFS Transparency. However, you can later choose to use the native HDFS instead by unintegrating HDFS Transparency.

In that case, you must follow these steps, to ensure that the native HDFS have the correct NameNode setting.

1. Follow the steps 1-3 of "Unintegrating HDFS Transparency" on page 433 section to revert to native HDFS mode.
2. Do not start all the services after unintegrating.

3. Ensure **ambari-server restart** was run on the Ambari server.
4. If you have a NameNode HA enabled environment, then follow the HA steps listed below. Else, follow the non-HA environment steps.

If HA is enabled, perform the following steps, for example:

NameNode being moved: c902f09x02

Execute the Move NameNode service during the HDFS Transparency Integration to the new NameNode c902f09x04.

NameNode not moved: c902f09x03

1. Start the Zookeeper Server from the Ambari GUI.
2. Start the NameNode that was not moved (c902f09x03) from the Hosts dashboard, clicking the **NameNode that was not moved** > **Summary tab** > **Components** > **NameNode / HDFS (Active or Standby)** > **Start**. This will start only the NameNode. Do not start any other services or hosts.
3. Format the ZKFC on the NameNode that was not moved (c902f09x03) by running the following command:

   ```
   sudo su hdfs -l -c 'hdfs zkfc -formatZK'
   ```

4. On the new NameNode (c902f09x04), run the following command:

   ```
   sudo su hdfs -l -c 'hdfs namenode -bootstrapStandby'
   ```

5. Start All services.

   From the dashboard, select **Actions** > **Start All**. The Hadoop cluster will now use native HDFS.

If HA is not enabled, perform the following steps, for example:

NameNode being moved: c902f09x02

Execute the **Move Namenode** service during the HDFS Transparency integration to a new NameNode (c902f09x03).

1. Copy the contents of /hadoop/hdfs/namenode from the NameNode being moved (c902f09x02) to /hadoop/hdfs/namenode on the new NameNode (c902f09x03).
2. On the new NameNode (c902f09x03), run the following commands:
   a. **chown -R hdfs:hadoop /hadoop/hdfs/namenode**
   b. **mkdir -p /var/lib/hdfs/namenode/formatted**
3. Start All services.

   From the dashboard, select **Actions** > **Start All**. The Hadoop cluster will now use native HDFS.

Moving the Ambari server

This section describes how to move the Ambari server onto a new host.

Moving the Ambari server is supported only if the current Ambari server host and the IBM Spectrum Scale service are active and functional.

The IBM Spectrum Scale master component is tightly integrated with the Ambari server, therefore the Moving the Ambari Server cannot be run when the IBM Spectrum Scale is in integrate state.

Plan a cluster maintenance window and prepare for cluster downtime.

Note:

- The new host has to be a GPFS node.

- Requires the `SpectrumScale_UpgradeIntegrationPackage` script which is packaged with the Mpack package. This script is used to remove the Scale Mpack and service, and reinstall it to a different location. The software stack is not upgraded. Ignore the `STARTING WITH SPECTRUM SCALE EXPRESS UPGRADE POST STEPS` and `STARTING WITH SPECTRUM SCALE EXPRESS UPGRADE PRE STEPS` outputs from the script.

1. Log in to Ambari.
2. Stop all the services by clicking **Ambari** > **Actions** > **Stop All**.
3. After all the services have stopped, unintegrate the transparency.

   Follow the steps in Unintegrating Transparency, and ensure that the **`ambari-server restart`** is run.

   Note: Do not start the services.
4. Check if the IBM Spectrum Scale has stopped by running **`/usr/lpp/mmfs/bin/mmgetstate -a`**.

   If the IBM Spectrum Scale service has not stopped, stop it by clicking **Ambari** > **Spectrum Scale** > **Actions** > **Stop**.
5. On the Ambari server node as root, run the `SpectrumScale_UpgradeIntegrationPackage` script with the **`--preEU`** option.

   The **`--preEU`** option saves the existing IBM Spectrum Scale service information into JSON files in the local directory where the script was run. It also removes the IBM Spectrum Scale service from the Ambari cluster. This does not affect the IBM Spectrum Scale file system.

   Before proceeding, review the following questions and have the information ready for your environment. If Kerberos is enabled, more inputs are required.

   ```
   $ cd /root/GPFS_Ambari
   $ ./SpectrumScale_UpgradeIntegrationPackage --preEU
   Are you sure you want to upgrade the GPFS Ambari integration package (Y/N)? (Default Y):
   ***STARTING WITH PRE EXPRESS UPGRADE STEPS***
   ************************************************************
   Enter the Ambari server username:
   Enter the password for the Ambari server.
   SSL Enabled (True/False) (Default False):
   Enter the Ambari server Port. (Default 8080):
   ...
   # Note: If Kerberos is enabled, then the KDC principal and password information are
   required.
   Kerberos is Enabled. Proceeding with Configuration
   Enter kdc principal:
   Enter kdc password:
   ```
6. Run the Mpack uninstaller script to remove the existing Mpack.

   `$./SpectrumScaleMPackUninstaller.py`
7. Move the Ambari server to the new host. For more information, see Moving the Ambari Server.
8. Move the directory that contains the Mpack and the JSON configurations files to the new host where the **`SpectrumScale_UpgradeIntegrationPackage --preEU setp`** was run.
9. Modify the existing Ambari Server name with the new host name in the `gpfs-master-node.txt` file.
10. Modify the key value **`gpfs.webui.address`** with the new host name in the `gpfs-advance.json` file. Replace the **`gpfs.webui.address`**:`https://<existing ambari server>` with **`gpfs.webui.address`**:`https://<new host name>`.
11. On the Ambari server node as root, run the `SpectrumScale_UpgradeIntegrationPackage` script with the **`--postEU`** option in the directory where the **`--preEU`** step was run and where the JSON configurations were stored.

    Before proceeding, review the following questions and have the information ready for your environment. If Kerberos is enabled, more inputs are required.

    ```
    $ ./SpectrumScale_UpgradeIntegrationPackage --postEU
    Are you sure you want to upgrade the GPFS Ambari integration package (Y/N)? (Default Y):
    *************************************************************
    ***STARTING WITH SPECTRUM SCALE EXPRESS UPGRADE POST STEPS***
    ```

```
*************************************************************
Starting Post Express Upgrade Steps. Enter Credentials
Enter the Ambari server User:(Default admin ):
Enter the password for the Ambari server.
Password:
Retype password:
SSL Enabled (True/False) (Default False):
Enter the Ambari server Port. (Default 8080):
....
# Accept License
Do you agree to the above license terms? [yes or no]
yes
Installing...
Enter Ambari Server Port Number. If it is not entered, the installer will take default port
8080 :
INFO: Taking default port 8080 as Ambari Server Port Number.
Enter Ambari Server IP Address :
192.0.2.17
Enter Ambari Server Username, default=admin :
INFO: Taking default username "admin" as Ambari Server Username.
Enter Ambari Server Password :
...
Enter kdc principal:
Enter kdc password:
...
```

```
*********************************************************************************************************
*****************
Upgrade of the Spectrum Scale Service completed successfully.
From the Ambari GUI, check the IBM Spectrum Scale installation progress through the background
operations panel.
*********************************************************************************************************
*****************
*********************************************************************************************************
******************
IMPORTANT:  You need to ensure that the HDFS Transparency package, gpfs.hdfs-protocol-3.0.X, is updated
in the Spectrum
Scale repository. Then follow the "Upgrade Transparency" service action in the Spectrum Scale service UI
panel to propagate
the package to all the GPFS Nodes.
After that is completed, invoke the "Start All" services in Ambari.
*********************************************************************************************************
******************
```

12. Start all services by clicking **Ambari** > **Actions** > **Start All**.

    Restart all components by using the restart icon.

    Note:

    - If the IBM Spectrum Scale service is restarted by using the restart icon, HDFS service also needs to be restarted.
    - The NameNode last checkpoint alert can be ignored and can be disabled.
    - If the HBase master failed to start with `FileAlreadyExistsException` error, restart HDFS and then HBase master.

Ambari maintenance mode support for IBM Spectrum Scale service

Managing and monitoring a cluster means you will want to perform hardware, firmware, or OS maintenance on a host or want to test a service configuration change. Using the Ambari Maintenance Mode can help to suppress alerts and omit bulk operations for specific services when performing hardware or software maintenance.

There are different scopes to the Ambari maintenance mode:

- Service level - All the components of a Ambari service are excluded from a cluster-wide Start All/Stop All operation.
- Component level - A specific component of the service can be put in the maintenance mode. In such a case, that specific component is excluded from a service-level start/stop operation or a cluster-wide Start All/Stop All operation.
- Host level - When a particular Ambari host is put in the maintenance mode, all the components located on that host will be put in maintenance mode. In such a case, these components are excluded from a service-level start/stop operation or a cluster-wide Start All/Stop All operations.

For more information on Ambari Maintenance mode, see Cloudera Managing and Monitoring a Cluster.

Prior to Mpack 2.7.0.9, service-level maintenance mode was honored for the IBM Spectrum Scale service. However, component-level and host-level maintenance modes were not supported for the IBM Spectrum Scale service components. From Mpack 2.7.0.9, this support for component and host-level maintenance modes is added but is subject to the following constraints:

1. The IBM Spectrum Scale service is the IBM Spectrum Scale filesystem. The IBM Spectrum Scale service contains the GPFS_MASTER and GPFS_NODE components. Every node that contains a NameNode or DataNode will have a GPFS_NODE component. The GPFS_NODE is the component of the IBM Spectrum Scale filesystem. Therefore, if the GPFS_NODE is stopped or set to maintenance mode on a node, the NameNode or DataNode on that node requires to be stopped or set to maintenance mode as well. The NameNode or DataNode component cannot access the filesystem if the GPFS_NODE component is stopped.
2. The GPFS_MASTER and GPFS_NODE in the Ambari server cannot be in the maintenance mode. This is because the GPFS_MASTER is responsible for executing the IBM Spectrum Scale commands in the cluster and it requires the GPFS_NODE component to be up and running in order to run those commands.
3. In case of a START ALL operation, if less than three IBM Spectrum Scale quorum nodes are in non-maintenance mode, then the Ambari Maintenance mode is disabled for the IBM Spectrum Scale components and host levels and the START ALL will start all the IBM Spectrum Scale nodes. This is because IBM Spectrum Scale cannot function properly if enough quorum nodes are not available.
4. For FPO and local clusters, if **`gpfs.storage.type`** is set to *local*, special processing is required for filesystem mounting and mounting operations that Ambari Maintenance mode cannot address. Therefore, the components and host-level Ambari Maintenance modes for FPO and local clusters is not supported.
5. With Ambari maintenance mode, you must use the Ambari GUI and not the IBM Spectrum Scale CLI commands to stop and start the HDFS Transparency nodes.

Maintenance procedure for IBM Spectrum Scale node using Ambari maintenance mode

If you need to do maintenance (servicing) of the IBM Spectrum Scale nodes, you can set the Ambari maintenance mode for the GPFS_NODE and the corresponding NameNode and DataNode components residing on the same node.

Note: The Ambari maintenance mode is not supported in the FPO (local) filesystems.

General procedure for maintenance

The following procedure is for GPFS_NODE that is not colocated with the Ambari server:

1. In an HA environment, if the active NameNode 1 requires to be serviced, ensure that the failover of NameNode 1 has completed and that NameNode 1 is now on standby. NameNode 2 should now become the active NameNode.
2. To service only a subset of the IBM Spectrum Scale nodes, ensure that there are enough IBM Spectrum Scale quorum nodes for IBM Spectrum Scale to stay healthy.

   Run the following command and check the **Designation** field on the quorum node:

   ```
   # mmlscluster
   ```

   In order to avoid losing quorum, if the quorum nodes are not enough, move the quorum designation to the other IBM Spectrum Scale nodes that are not being serviced.

   For information on checking and setting the quorum nodes, see the *Which nodes in my cluster are quorum nodes?* topic in *IBM Storage Scale: Problem Determination Guide*.
3. In Ambari, to set the component-level maintenance mode for each node, perform the following:

- Stop the GPFS_NODE and the NameNode or DataNode components on the node that is to be serviced.
- Set the GPFS_NODE and the NameNode and DataNode components to the Ambari maintenance mode on the node to be serviced.

4. Service the IBM Spectrum Scale nodes that were set to the Ambari maintenance mode.
5. From Ambari GUI, disable the Ambari maintenance mode for the `GPFS_NODE` and the NameNode or DataNode components on the node that was serviced. Perform this for each node.
6. Start the GPFS_NODE and the NameNode or DataNode components on each node or perform an Ambari START ALL.
7. If the quorum designation was moved in step 2, you can move it back to the original designated IBM Spectrum Scale node.

Maintenance procedure for the Ambari server node with colocated GPFS_NODE and GPFS_MASTER

The GPFS_MASTER and GPFS_NODE on the Ambari server cannot honor the Ambari maintenance mode. The GPFS_MASTER is also responsible for executing the IBM Spectrum Scale commands in the cluster. Therefore, this node requires to be serviced on its own.

1. In order to avoid quorum loss, if the GPFS_MASTER and GPFS_NODE on the Ambari server have quorum designation, move the quorum designation to other IBM Spectrum Scale nodes that are not being serviced.

   For information on checking and setting the quorum nodes, see the *Which nodes in my cluster are quorum nodes?* topic in *IBM Storage Scale: Problem Determination Guide*.

   If there is an Active NameNode present on the GPFS_MASTER node, initiate a failover and ensure that the current NameNode is on standby.
2. To set the component-level maintenance mode on the Ambari server, perform the following:
   - Stop the GPFS_MASTER, GPFS_NODE and NameNode or DataNode components.
   - Set the GPFS_MASTER, GPFS_NODE and NameNode or DataNode components to the maintenance mode.

     Note: After you stop the IBM Spectrum Scale service components, you cannot manage the IBM Spectrum Scale service from the Ambari server.
3. On the Ambari server, service the IBM Spectrum Scale node.
4. From the Ambari GUI, disable the maintenance mode for GPFS_MASTER, GPFS_NODE and NameNode or DataNode components on the Ambari server that was serviced.
5. On the Ambari server, start the GPFS_MASTER, GPFS_NODE and the NameNode or DataNode components.
6. If the quorum designation was moved in step 1, you can move it back to the original designated Ambari server node.

Restricting root access

For many secure environments that requires restricted access and limits the services that run as the root user, the Ambari must be configured to operate without direct root access.

First follow the "Planning" on page 349 section, and ensure that the kernel* packages are installed beforehand as root.

Perform the following steps to set up Ambari and IBM Spectrum Scale for a non-root user:

1. Create a user ID that can perform passwordless ssh between all the nodes in the cluster. This non-root user ID is required to configure the Ambari server and agents when setting up the Ambari cluster in step 3.
2. Verify that the root ID and the Ambari server non-root ID can perform passwordless SSH.

Bi-directional passwordless SSH must work for the non-root ID from the GPFS Master node (Ambari server) to all the GPFS nodes and to itself (Ambari server node).

Root ID must be able to perform passwordless SSH from the GPFS Master node (Ambari server) to all the GPFS nodes and to itself (Ambari server node), uni-directional only.

The BI example uses am_agent as the non-root id for the Ambari server, the Ambari agents, and the IBM Spectrum Scale cluster user.

The HDP example uses ambari-server as the non-root id for the Ambari server, and am_agent as the non-root id for the Ambari agents and the IBM Spectrum Scale cluster.

The user ID and group ID of this user must be same. The user ID and group ID of the non-root ID must be same.

For example,

As root: **`ssh am_agent@<ambari-agent-host>`** must work without a password.

As am_agent: **`ssh am_agent@<ambari-agent-host>`** must work without a password.

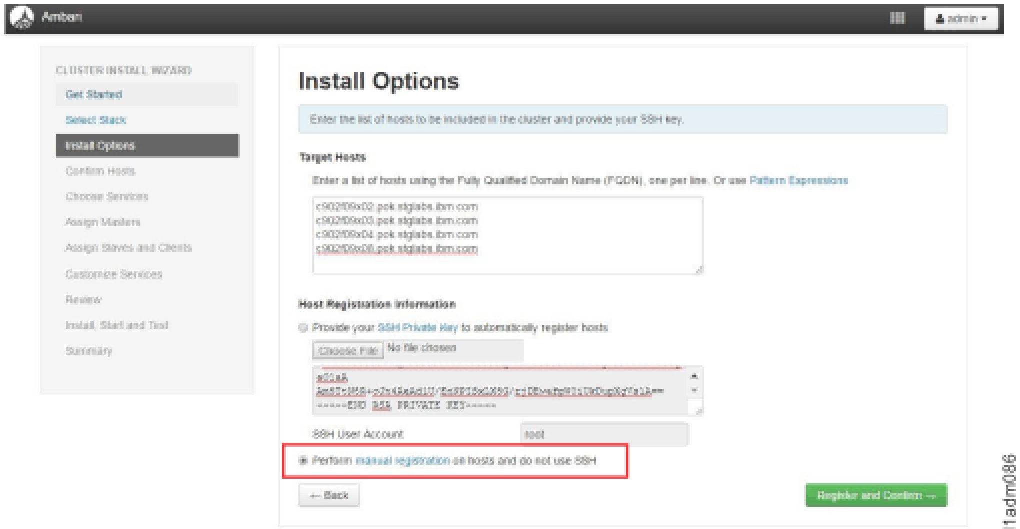
3. Set up an Ambari cluster as the non-root user.

   For BI, follow the steps in the IBM BigInsights Installation documentation under Configuring Ambari for non-root access.

   For HDP, follow the steps in the Hortonworks Installation documentation under Configuring Ambari for non-root.

   Note: Once you are at the Host Registration wizard, ensure the following:

   - The SSH User Account specifies the non-root user ID.
   - The manual host registration radio button in the Ambari UI is set. This will ensure that the Ambari agent processes will run as the non-root user, and execute the IBM Spectrum Scale service integration code.



Install Options

Enter the list of hosts to be included in the cluster and provide your SSH key.

Target Hosts

Enter a list of hosts using the Fully Qualified Domain Name (FQDN), one per line. Or use Pattern Expressions

```
c902f09x02.pok.stglabs.ibm.com
c902f09x03.pok.stglabs.ibm.com
c902f09x04.pok.stglabs.ibm.com
c902f09x05.pok.stglabs.ibm.com
```

Host Registration Information

○ Provide your SSH Private Key to automatically register hosts

Choose File No file chosen

```
[illegible]
[illegible]
-----END RSA PRIVATE KEY-----
```

SSH User Account: root

● Perform manual registration on hosts and do not use SSH

bl1adm086

4. Configure IBM Spectrum Scale without remote root by following the steps in the *Configuring sudo* topic in the *IBM Storage Scale: Administration Guide*.

   The non-root user/group id used in the Configuring sudo section of the IBM Spectrum Scale document is the Ambari agent non-root user/group id.

5. Additionally, on each host, modify the `/etc/sudoers` file to include the following changes:

   - Add the list of allowed commands for the non-root user:

   `/usr/bin/cd /usr/lpp/mmfs/src`, `/usr/bin/curl`, `/usr/bin/make Autoconfig`, `/usr/bin/make World`, `/usr/bin/make`

`InstallImages`, `/usr/lpp/mmfs/hadoop/sbin/mmhadoopctl`, `/usr/lpp/mmfs/hadoop/sbin/hadoop-daemon.sh`, `/usr/lpp/mmfs/hadoop/sbin/gpfs_hdfs_pkg.sh, /usr/sbin/parted, /usr/sbin/partprobe,/sbin/mkfs.ext4`

BI sudoers

The Ambari Server user and group is am_agent: am_agent.

The Ambari Agent user and group is am_agent:am_agent.

The IBM Spectrum Scale cluster user and group is am_agent:am_agent.

Example of `/etc/sudoers` file added entries in BI environment:

```
# Ambari IOP Customizable Users
am_agent ALL=(ALL) NOPASSWD:SETENV:  /bin/su hdfs *, /bin/su ambari-qa *, /bin/su zookeeper
*,
/bin/su knox  *, /bin/su ams *, /bin/su flume *, /bin/su hbase *, /bin/su spark *,
/bin/su hive *, /bin/su hcat *, /bin/su kafka *, /bin/su mapred *, /bin/su oozie *,
/bin/su sqoop *, /bin/su storm *, /bin/su yarn *, /bin/su solr *, /bin/su titan *,
/bin/su ranger *, /bin/su kms *

# Ambari value-adds Customizable Users
am_agent ALL=(ALL) NOPASSWD:SETENV: /bin/su - bigsheets *, /bin/su uiuser *,
/bin/su tauser *, /bin/su - bigr *

#Ambari Non-Customizable Users
am_agent ALL=(ALL) NOPASSWD:SETENV: /bin/su mysql *

# Ambari IOP Commands
am_agent ALL=(ALL) NOPASSWD:SETENV:  /usr/bin/yum,/usr/bin/zypper, /usr/bin/apt-get,
/bin/mkdir,  /usr/bin/test, /bin/ln, /bin/chown, /bin/chmod, /bin/chgrp, /usr/sbin/groupadd,
/usr/sbin/groupmod, /usr/sbin/useradd,  /usr/sbin/usermod, /bin/cp, /usr/sbin/setenforce,
/usr/bin/stat,  /bin/mv, /bin/sed,/bin/rm, /bin/kill, /bin/readlink, /usr/bin/pgrep,  /bin/
cat,
/usr/bin/unzip, /bin/tar, /usr/bin/tee, /bin/touch, /usr/bin/iop-select, /usr/bin/conf-
select,
/usr/iop/current/hadoop-client/sbin/hadoop-daemon.sh, /usr/lib/hadoop/bin/hadoop-daemon.sh,
/usr/lib/hadoop/sbin/hadoop-daemon.sh,  /sbin/chkconfig gmond off,  /sbin/chkconfig gmetad
off,
/etc/init.d/httpd *, /sbin/service  iop-gmetad start, /sbin/service iop-gmond start,
/usr/sbin/gmond,  /usr/sbin/update-rc.d ganglia-monitor *, /usr/sbin/update-rc.d gmetad *,
/etc/init.d/apache2 *, /usr/sbin/service iop-gmond *, /usr/sbin/service  iopgmetad *,
/sbin/service mysqld *, /sbin/service mysql *,
/usr/bin/python2.6/var/lib/ambari-agent/data/tmp/validateKnoxStatus.py *,
/usr/iop/current/knox-server/bin/knoxcli.sh *, /usr/bin/dpkg *, /bin/rpm  *, /usr/sbin/hst *,
/usr/sbin/service mysql *, /usr/sbin/service mariadb *, /usr/bin/ambari-python-wrap,
/usr/bin/cd /usr/lpp/mmfs/src, /usr/bin/curl, /usr/bin/make Autoconfig, /usr/bin/make World,
/usr/bin/make InstallImages, /usr/lpp/mmfs/hadoop/sbin/mmhadoopctl,
/usr/lpp/mmfs/hadoop/sbin/hadoop-daemon.sh, /usr/lpp/mmfs/hadoop/bin/gpfs,
/usr/lpp/mmfs/hadoop/sbin/gpfs_hdfs_pkg.sh, /usr/sbin/parted, /usr/sbin/partprobe,
/sbin/mkfs.ext4

# Ambari value-adds Commands
am_agent ALL=(ALL) NOPASSWD:SETENV:  /usr/bin/updatedb  *, /usr/bin/sh *, /usr/bin/scp *,
/usr/bin/pkill *, /bin/unlink *, /usr/bin/mysqld_safe, /usr/bin/mysql_install_db, /usr/bin/R,
/usr/bin/Rscript,  /bin/bash, /usr/bin/kinit, /usr/bin/hadoop, /usr/bin/mysqladmin,
/usr/sbin/userdel, /usr/sbin/groupdel,  /usr/sbin/ambari-server, /usr/bin/klist
Cmnd_Alias BIGSQL_SERVICE_AGNT=/var/lib/ambari-agent/cache/stacks/BigInsights/*/services/
BIGSQL/package/scripts/*
Cmnd_Alias BIGSQL_SERVICE_SRVR=/var/lib/ambari-server/resources/stacks/BigInsights/*
/services/BIGSQL/package/scripts/*
Cmnd_Alias BIGSQL_DIST_EXEC=/usr/ibmpacks/current/bigsql/bigsql/bin/*,
/usr/ibmpacks/current/bigsql/bigsql/libexec/*,
/usr/ibmpacks/current/bigsql/bigsql/install/*, /usr/ibmpacks/current/IBM-DSM/ibm-
datasrvrmgr/bin/*,
/usr/ibmpacks/bin/*/*
Cmnd_Alias BIGSQL_OS_CALLS=/bin/su, /usr/bin/getent, /usr/bin/id, /usr/bin/ssh, /bin/echo,
/usr/bin/scp, /bin/find, /usr/bin/du, /sbin/mkhomedir_helper, /bin/curl

am_agent ALL=(ALL) NOPASSWD:SETENV:/bin/*, /usr/bin/*, /usr/sbin/*, /usr/bin/R, /usr/bin/
Rscript,
BIGSQL_SERVICE_AGNT, BIGSQL_SERVICE_SRVR, BIGSQL_DIST_EXEC, BIGSQL_OS_CALLS

Defaults exempt_group = am_agent
Defaults !env_reset,env_delete-=PATH
Defaults: am_agent !requiretty
```

```
#GPFS cluster non-root added
# Preserve GPFS environment variables:
Defaults env_keep += "MMMODE environmentType GPFS_rshPath GPFS_rcpPath mmScriptTrace
GPFSCMDPOR-TRANGE GPFS_CIM_MSG_FORMAT"

# Allow members of the gpfs group to run all
commands but only selected commands without a password:
%am_agent ALL=(ALL) PASSWD: ALL, NOPASSWD: /usr/lpp/mmfs/bin/mmremote, /usr/bin/scp,
/bin/echo, /usr/lpp/mmfs/bin/mmsdrrestore

# Disable requiretty for group gpfs:
Defaults:%am_agent !requiretty
```

HDP sudoers

The Ambari Server user and group is ambari-server:hadoop.

The Ambari Agent user and group is am_agent:am_agent.

The IBM Spectrum Scale cluster user and group is am_agent:am_agent.

Example of `/etc/sudoers` file added entries in HDP environment:

```
# Ambari Commands
ambari-server ALL=(ALL) NOPASSWD:SETENV: /bin/mkdir -p /etc/security/keytabs, /bin/chmod *
/etc/security/keytabs/*.keytab, /bin/chown * /etc/security/keytabs/*.keytab, /bin/chgrp *
/etc/security/keytabs/*.keytab, /bin/rm -f /etc/security/keytabs/*.keytab, /bin/cp -p -f
/var/lib/ambari-server/data/tmp/* /etc/security/keytabs/*.keytab

#Sudo Defaults - Ambari Server(In order for the agent to run its commands non-interactively,
some defaults need to be overridden)
Defaults exempt_group = ambari-server
Defaults !env_reset,env_delete-=PATH
Defaults: ambari-server !requiretty

# Ambari Agent non root configuration
# Ambari Customizable Users
am_agent ALL=(ALL) NOPASSWD:SETENV: /bin/su hdfs *,/bin/su ambari-qa *,/bin/su ranger *,
/bin/su zookeeper *,/bin/su knox *,/bin/su falcon *,/bin/su ams *, /bin/su flume *,/bin/su
hbase *,
/bin/su spark *,/bin/su accumulo *,/bin/su hive *,/bin/su hcat *,/bin/su kafka *,/bin/su
mapred *,
/bin/su oozie *,/bin/su sqoop *,/bin/su storm *,/bin/su tez *,/bin/su atlas *,/bin/su yarn *,
/bin/su kms *,/bin/su activity_analyzer *,/bin/su livy *,/bin/su zeppe-lin *,/bin/su infra-
solr *,
/bin/su logsearch *

# Ambari: Core System Commands

am_agent ALL=(ALL) NOPASSWD:SETENV: /usr/bin/yum,/usr/bin/zypper,/usr/bin/apt-get, /bin/
mkdir,
/usr/bin/test, /bin/ln, /bin/ls, /bin/chown, /bin/chmod, /bin/chgrp, /bin/cp, /usr/sbin/
setenforce,
/usr/bin/test, /usr/bin/stat, /bin/mv, /bin/sed, /bin/rm, /bin/kill, /bin/readlink, /usr/bin/
pgrep,
/bin/cat, /usr/bin/unzip, /bin/tar, /usr/bin/tee, /bin/touch, /usr/bin/mysql, /sbin/service
mysqld *,
/usr/bin/dpkg *, /bin/rpm *, /usr/sbin/hst *, /sbin/service rpcbind *, /sbin/service
portmap *,
/usr/bin/cd, /usr/lpp/mmfs/src, /usr/bin/curl, /usr/bin/make Au-toconfig, /usr/bin/make
World,
/usr/bin/make InstallImages, /usr/lpp/mmfs/hadoop/sbin/mmhadoopctl,
/usr/lpp/mmfs/hadoop/sbin/hadoop-daemon.sh, /usr/lpp/mmfs/hadoop/bin/gpfs,
/usr/lpp/mmfs/hadoop/sbin/gpfs_hdfs_pkg.sh, /usr/sbin/parted, /usr/sbin/partprobe, /sbin/
mkfs.ext4

# Ambari: Hadoop and Configuration Commands
am_agent ALL=(ALL) NOPASSWD:SETENV: /usr/bin/hdp-select, /usr/bin/conf-select,
/usr/hdp/current/hadoop-client/sbin/hadoop-daemon.sh, /usr/lib/hadoop/bin/hadoop-daemon.sh,
/usr/lib/hadoop/sbin/hadoop-daemon.sh, /usr/bin/ambari-python-wrap *

# Ambari: System User and Group Commands
am_agent ALL=(ALL) NOPASSWD:SETENV: /usr/sbin/groupadd, /usr/sbin/groupmod,
/usr/sbin/useradd, /usr/sbin/usermod

# Ambari: Knox Commands
am_agent ALL=(ALL) NOPASSWD:SETENV: /usr/bin/python2.6
/var/lib/ambari-agent/data/tmp/validateKnoxStatus.py *, /usr/hdp/current/knox-server/bin/
knoxcli.sh
```

```
# Ambari: Ranger Commands
am_agent ALL=(ALL) NOPASSWD:SETENV: /usr/hdp/*/ranger-usersync/setup.sh, /usr/bin/ranger-
usersync-stop,
/usr/bin/ranger-usersync-start, /usr/hdp/*/ranger-admin/setup.sh *,
/usr/hdp/*/ranger-knox-plugin/disable-knox-plugin.sh *,
/usr/hdp/*/ranger-storm-plugin/disable-storm-plugin.sh *,
/usr/hdp/*/ranger-hbase-plugin/disable-hbase-plugin.sh *,
/usr/hdp/*/ranger-hdfs-plugin/disable-hdfs-plugin.sh *,
/usr/hdp/current/ranger-admin/ranger_credential_helper.py,
/usr/hdp/current/ranger-kms/ranger_credential_helper.py,
/usr/hdp/*/ranger-*/ranger_credential_helper.py

# Ambari Infra and LogSearch Commands
am_agent ALL=(ALL) NOPASSWD:SETENV: /usr/lib/ambari-infra-solr/bin/solr *,
/usr/lib/ambari-logsearch-logfeeder/run.sh *, /usr/sbin/ambari-metrics-grafana *,
/usr/lib/ambari-infra-solr-client/solrCloudCli.sh *

# Sudo Defaults - Ambari Agent (In order for the agent to run its commands non-interactively,
some defaults need to be overridden)
Defaults exempt_group = am_agent
Defaults !env_reset,env_delete-=PATH
Defaults: am_agent !requiretty

#GPFS cluster non-root added
# Preserve GPFS environment variables:
Defaults env_keep += "MMMODE environmentType GPFS_rshPath GPFS_rcpPath mmScriptTrace
GPFSCMDPOR-TRANGE GPFS_CIM_MSG_FORMAT"

# Allow members of the gpfs group to run all commands but only selected
commands without a password:
%am_agent ALL=(ALL) PASSWD: ALL, NOPASSWD: /usr/lpp/mmfs/bin/mmremote, /usr/bin/scp,
/bin/echo, /usr/lpp/mmfs/bin/mmsdrrestore

# Disable requiretty for group gpfs:
Defaults:%am_agent !requiretty
```

6. Perform the steps from "Deploy the IBM Spectrum Scale service" on page 367 to add the module as the root user.

   Note: You must restart Ambari as root. Exceptions occurs as non-root user. However, this issue is not shown on Ambari 2.5.0.3 when an Ambari-server restarts with non-root user.

7. Perform the steps from "Deploy the IBM Spectrum Scale service" on page 367.

   This requires restarting Ambari as root. Exceptions occur as non-root user. However, this issue is not shown on Ambari 2.5.0.3 when ambari-server restart with a non-root user.

   Note:

   - There might be an issue with HBase stopping in a non-root environment. For more information, see the "Troubleshooting Ambari" on page 472 section.
   - In non-root Ambari environment, the Hive service check might fail. For resolution, see the "Troubleshooting Ambari" on page 472 section.

IBM Spectrum Scale management GUI

The IBM Spectrum Scale management GUI must be manually installed and accessed.

Installation instructions for IBM Spectrum Scale management GUI are available in the *Manually installing IBM Spectrum Scale management GUI* topic in the *IBM Storage Scale: Concepts, Planning, and Installation Guide*.

The IBM Spectrum Scale management GUI can be accessed as a quick link URL within the IBM Spectrum Scale Ambari service.

If you are running IBM Spectrum Scale 4.2.0 or later, the rpms required to install the GUI are included in Standard and Advanced Editions for Linux on x86 and Power (Big Endian or Little Endian). The GUI requires RHEL 7.

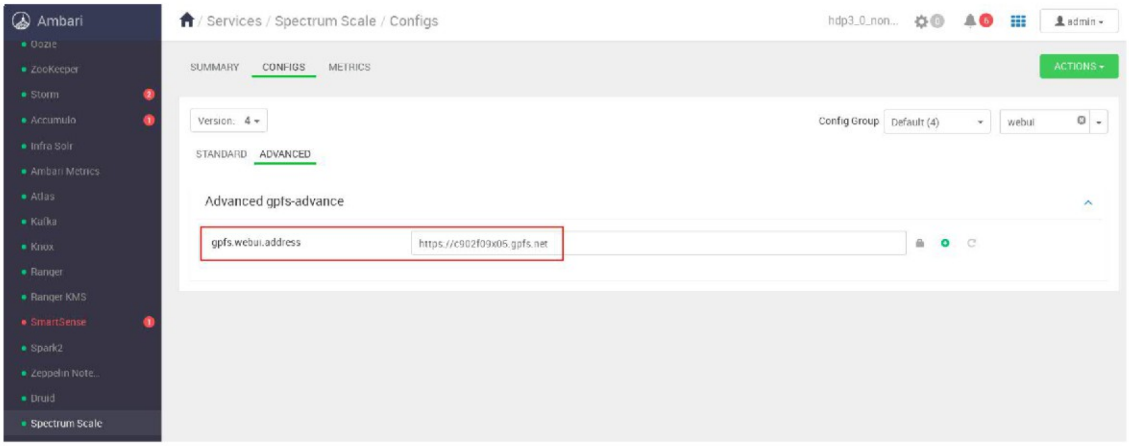
Procedure

1. Deploy IBM Spectrum Scale Management GUI.
2. Set the **`gpfs.webui.address`** field in the IBM Spectrum Scale Service configuration advanced panel.

For example, `https://<ambari_server_fully_qualified_hostname>/gui`

From **Ambari GUI** > **IBM Spectrum Scale service**, select **Configs** > **Advanced** > **Advanced gpfs-advance**.

Add the URL to the **`gpfs.webui.address`** field.



Advanced gpfs-advance

gpfs.webui.address https://c902f09x05.gpfs.net

bl1bda57

3. Restart the IBM Spectrum Scale service.
4. Sync the configuration.

   Click **Ambari GUI** > **IBM Spectrum Scale service** > **Actions** > **Set Management GUI URL**.
5. Restart Ambari server.

IBM Spectrum Scale versus Native HDFS

When IBM Spectrum Scale service is added, the native HDFS is no longer used. The Hadoop application interacts with HDFS transparency similar to their interactions with the native HDFS.

The application can access HDFS by using Hadoop file system APIs and Distributed File System APIs. The application can have its own cluster that is larger than the HDFS protocol cluster. However, all the nodes within the application cluster must be able to connect to all the nodes in the HDFS protocol cluster by RPC.

Note: The Secondary NameNode and Journal nodes in native HDFS are not needed for HDFS Transparency because of the following reasons:

- The HDFS Transparency NameNode is stateless.
- Metadata are distributed.
- The NameNode does not maintain the FSImage-like or EditLog information.

Functional limitations

This topic lists the functional limitations.

General

- The maximum number of Extended Attributes (EA) is limited by IBM Spectrum Scale. The total size of the EA key and value must be less than a metadata block size in IBM Spectrum Scale.
- The EA operation on snapshots is not supported.
- Raw namespace is not implemented because it is not used internally.
- If **`gpfs.replica.enforced`** is configured as gpfs, the Hadoop shell command **`hadoop dfs -setrep`** does not take effect. Also, **`hadoop dfs -setrep -w`** stops functioning and does not exit.
- HDFS Transparency NameNode does not provide *safemode* because it is stateless.

- HDFS Transparency NameNode does not need the second NameNode like native HDFS because it is stateless.
- Maximal replica for IBM Spectrum Scale is *3* from code. However, the maximal replica for your file system might be less than 3. You can check this by running `/usr/lpp/mmfs/bin/mmlsfs <fs-name> -R`.
- IBM Spectrum Scale has no ACL entry number limit. The maximal entry number is limited by Int32.
- **`SendPacketDownStreamAvgInfo`** and **`SlowPeersReport`** from `http://<namenode/datanode:port>/jmx` are not supported.
- GPFS file data replication factor on ESS requires to be set to 1, and dfs.replica should be set to 1.
- HDFS supported interface for `hdfs xxx` is `hdfs dfs xxx`. Other interface from `hdfs xxx` is considered native HDFS specific, that is not used by the HDFS Transparency.

  These are some examples of what is not supported:

  – fsck
  – dfsadmin

    - - safemode

  – Native HDFS caching (cacheadmin)
  – NameNode format not needed to run (**`namenode -format`**)
- Distcp over snapshot is not supported.
- For HDFS Transparency version 3.0.x, the environment variables above can be exported, except for HADOOP_COMMON_LIB_NATIVE_DIR.

  This is because HDFS Transparency uses its own native `.so` library.

  For HDFS Transparency version 3.0.x:

  – If you did not export HADOOP_CONF_DIR, then HDFS Transparency will read all the configuration files under `/var/mmfs/hadoop/etc/hadoop` such as the gpfs-site.xml file and the `hadoop-env.sh` file.
  – If you export `HADOOP_CONF_DIR`, then HDFS Transparency will read all the configuration files under `$HADOOP_CONF_DIR`. Since `gpfs-site.xml` is required for HDFS Transparency, it will only read the `gpfs-site.xml` file from the `/var/mmfs/hadoop/etc/hadoop` directory.

For HDP

- The "+" is not supported when using `hftp://namenode:50070`.

Functional differences

This topic lists the functional differences.

- ACLs is limited to 32 in native HDFS but not in IBM Spectrum Scale.
- File name length is limited in native HDFS while IBM Spectrum Scale uses maximal 255 utf-8 chars.
- The **`hdfs fsck`** is not supported in HDFS Transparency. Instead, use the IBM Spectrum Scale **`mmfsck`** command. If your file system is mounted, run `/usr/lpp/mmfs/bin/mmfsck -o -y`. If your file system is not mounted, run `/usr/lpp/mmfs/bin/mmfsck -y`.

Configuration that differs from native HDFS in IBM Spectrum Scale

This topic lists the differences between native HDFS and IBM Spectrum Scale.

Table 37. NATIVE HDFS AND IBM SPECTRUM SCALE DIFFERENCES

Property name	Value	New definition or limitation
`dfs.permissions.enabled`	True/false	For HDFS protocol, the permission check is always done.

Table 37. NATIVE HDFS AND IBM SPECTRUM SCALE DIFFERENCES (continued)

Property name	Value	New definition or limitation
`dfs.namenode.acls.enabled`	True/false	For native HDFS, the NameNode manages all metadata, including the ACL information. HDFS can use this to turn the ACL checking on or off. However, for IBM Spectrum Scale, the HDFS protocol does not hold the metadata. When on, the ACL is set and stored in the IBM Spectrum Scale file system. If the administrator turns it off later, the ACL entries that are set and stored in IBM Spectrum Scale take effect. This will be improved in the next release.
`dfs.blocksize`	Long digital	Must be a multiple of the IBM Spectrum Scale file system block size (`mmlsfs -B`). The maximal value is `1024 * file-system-data-block-size` (`mmlsfs -B`).
`gpfs.data.dir`	String	A user in Hadoop must have full access to this directory. If this configuration is omitted, a user in Hadoop must have full access to `gpfs.mount.dir`.
`dfs.namenode.fs-limits.max-xattrs-per-inode`	INT	Does not apply to the HDFS protocol.
`dfs.namenode.fs-limits.max-xattr-size`	INT	Does not apply to the HDFS protocol.
`dfs.namenode.fs-limits.max-component-length`	INT	Does not apply to HDFS Transparency. The file name length is controlled by IBM Spectrum Scale. Refer to IBM Spectrum Scale FAQ for file name length limit.
Native HDFS encryption	Supported	For more information, see "HDFS encryption" on page 192.
Native HDFS caching	Not supported	IBM Spectrum Scale.
NFS Gateway	Not supported	IBM Spectrum Scale provides POSIX interface and taking IBM Spectrum Scale protocol could give you better performance and scaling.

Limitations

Limitations and information

Known information, limitations and workarounds for IBM Spectrum Scale and HDFS Transparency integration are stated in this section.

General

- The IBM Spectrum Scale service does not support the rolling upgrade of IBM Spectrum Scale and Transparency from the Ambari GUI.
- The rolling upgrade of Hortonworks HDP cluster is not supported if the IBM Spectrum Scale service is still integrated.
- The minimum recommended version for IBM Spectrum Scale is 4.1 and above. HDFS Transparency is not dependent on the version of IBM Spectrum Scale.
- Manual Kerberos setup requires Kerberos setting in Ambari to be disabled before deploying IBM Spectrum Scale mpack. If IBM Spectrum Scale service is already installed, the HDFS Transparency requires to be unintegrated before enabling Kerberos in Ambari.
- Federation Support

  Federation is supported for open source Apache Hadoop stack. The HDFS Transparency connector supports two or more IBM Spectrum Scale file systems to act as one uniform file system for Hadoop applications. For more information, see "Overview" on page 139.
- The latest JDK supported version for Ambari is 1.8.0.77.
- Ambari is required to be restarted as root in a non-root environment, to avoid exceptions.
- All configuration changes must be made through the Ambari GUI, and not manually set into the HDFS configuration files or into the HDFS Transparency configuration files. This is to ensure that the configuration changes are propagated properly.
- In your existing cluster, if the HDFS settings in the HDFS Transparency configuration files were manually changed (For example: settings in core-site, hdfs-site, or log4j.properties in `/var/mmfs/hadoop/etc/hadoop`) and these changes were not implemented in the existing native HDFS configuration files, during the deployment of Ambari IOP or HDP and IBM Spectrum Scale service, the HDFS Transparency configuration is replaced by the Ambari UI HDFS configurations. Therefore, save changes that are set for the HDFS Transparency configuration files so that these values can later be applied through the Ambari GUI.
- For FPO systems, ensure that you follow the proper steps to stop/start IBM Spectrum Scale. Otherwise, restarting the IBM Spectrum Scale NSD might not be possible after NSDs go down and auto recovery fails. This can occur when doing **STOP ALL/START ALL** from Ambari which stops IBM Spectrum Scale without properly handling the NSDs in **FPO** mode for Mpack 2.4.2.6 and earlier and for Mpack 2.7.0.0. For more information, see IBM Spectrum Scale NSD are not able to be recovered in FPO clusters (Stop/Start of Scale service via Ambari GUI).

Installation

- Ambari only supports the creation of IBM Spectrum Scale FPO file system.
- While creating an Ambari IOP or HDP cluster, you do not need to create a local partition file system to be used for HDFS if you plan to install IBM Spectrum Scale FPO through Ambari. IBM Spectrum Scale Ambari management pack will create the recommended partitions for the local temp disks and IBM Spectrum Scale disks. The local temp disks are mounted and used for the Yarn local directories.
- If disks are partitioned before creating the IBM Spectrum Scale FPO through Ambari, the standard NSD is required to be used.
- Ensure that the GPFS Master and the Ambari server are colocated. The Ambari server must be part of the Ambari and GPFS cluster. This implies that the Ambari server host is defined as an Ambari agent host in the **Add Hosts UI** panel while setting up the Hadoop cluster. Otherwise, IBM Spectrum Scale service fails to install if the nodes are not colocated.
- If you need to deploy the IOP or HDP over an existing IBM Spectrum Scale FPO cluster, either store the Yarn's intermediate data into the IBM Spectrum Scale file system, or use idle disks formatted as a local file system. It is recommended to use the latter method. If a new IBM Spectrum Scale cluster is created

through the Ambari deployment, all the Yarn's NodeManager nodes should be FPO nodes with the same number of disks for each node specified in the NSD stanza.

- If you are deploying Ambari HDP on top of an existing IBM Spectrum Scale and HDFS Transparency cluster:
  - Perform a backup of the existing HDFS and HDFS Transparency configuration before proceeding to deploy Ambari IOP or HDP, or deploy the IBM Spectrum Scale service with Ambari on a system that has HDFS Transparency installed on it.
  - Ensure that the HDFS configuration provided through the Ambari UI is consistent with the existing HDFS configuration.
    - The existing HDFS NameNode and DataNode values must match the Ambari HDFS UI NameNode and DataNode values. Otherwise, the existing HDFS configuration will be overwritten by the default Ambari UI HDFS parameters after the Add Service Wizard completes.
    - The HDFS DataNodes being assigned in the **Assign Slaves and Clients** page in Ambari must contain the existing HDFS Transparency DataNodes. If the host did not have HDFS DataNode and GPFS Node set in Ambari, data on that host is not accessible, and cluster might be under replicated. If the node was not configured as an HDFS DataNode and GPFS node during the **Assign Slaves and Clients**, the host can add those components through the HOSTS component panel to resolve those issues. For more information, see "Adding GPFS node component" on page 441.
  - The HDFS NameNodes specified in the Ambari GUI during configuration must match the existing HDFS Transparency NameNodes.
  - Verify that the host names that are used are the data network addresses that IBM Spectrum Scale uses for its cluster setup. Otherwise in an existing or shared file system, the IBM Spectrum Scale service fails during installation because of a wrong host name.
  - While deploying HDP over an existing IBM Spectrum Scale file system, the IBM Spectrum Scale cluster must be started, and the file system must be mounted on all the nodes before starting the Ambari deployment.
- When deploying the Ambari IOP or HDP cluster, ensure there are no mount points in the cluster. Otherwise, the Ambari will take the shared mount point directory as the directory for the open source services. This will cause the different nodes to write to the same directory.
- Ensure that all the hosts for the IBM Spectrum Scale cluster contain the same domain name while creating the cluster through Ambari.
- IBM Spectrum Scale service requires that all the NameNodes and DataNodes are GPFS nodes.
- The IBM Spectrum Scale Ambari management pack uses the manual installation method and not the IBM Spectrum Scale installation toolkit.
- If installing a new FPO cluster through Ambari, Ambari creates the IBM Spectrum Scale with the recommended settings for FPO, and builds the GPFS portability layer on each node.
- It is recommended to assign HDFS Transparency NameNode running over GPFS node with metadata disks.
- It is recommended to assign Yarn ResourceManager node to be running HDFS Transparency NameNode.
- When you are deploying the IBM Spectrum Scale service, the **`gpfs.replica.enforced`** parameter might appear as **`dfs`** in the Ambari Scale service GUI, even though HDFS Transparency (3.1.0.5 and later) sets it to **`gpfs`** by default. Therefore, it is important to set the **`gpfs.replica.enforced`** parameter value to **`gpfs`** in Ambari. Otherwise, HDFS Transparency will use **`dfs`** as the value for the **`gpfs.replica.enforced`** parameter instead of **`gpfs`**.

  Update the **`gpfs.replica.enforced`** parameter to **`gpfs`** in the service wizard and proceed with the deployment.

Configuration

- After adding and removing nodes from Ambari, some aspects of the IBM Spectrum Scale configuration, such as page pool, as seen by running the **mmlsconfig** command, are not refreshed until after the next restart of the IBM Spectrum Scale Ambari service. However, this does not impact the functionality.
- Short circuit is disabled when IBM Spectrum Scale service is installed. For information on how to enable or disable Short Circuit, see Short Circuit Read Configuration.

Ambari GUI

- If any GPFS node other than the GPFS Master is stopped, the IBM Spectrum Scale panel does not display any alert.
- The NFS gateway is displayed on the HDFS dashboard but is not used by HDFS Transparency. NFS gateway is not supported. Use IBM Spectrum Scale protocol for better scaling if your application requires NFS interface.
- The **IBM Spectrum Scale Service UI Panel** > **Actions** > **Collect_Snap_Data** does not work if you configure an optional argument file (`/var/lib/ambari-server/resources/gpfs.snap.args`).
- For IBM Spectrum Scale GUI quick link, it is required to initialize the IBM Spectrum Scale management GUI before accessing through Ambari quick links. See IBM Spectrum Scale management GUI.

Node management

- Ambari adds nodes and installs the IBM Spectrum Scale software on the existing IBM Spectrum Scale cluster, but does not create or add NSDs to the existing file system.
- Adding a node in Ambari fails if the node to be added does not have the same IBM Spectrum Scale version or the same HDFS Transparency version as the version currently installed on the Ambari IBM Spectrum Scale HDFS Transparency cluster. Ensure that the node to be added is at the same IBM Spectrum Scale level as the existing cluster.
- Decommissioning a DataNode is not supported when IBM Spectrum Scale is integrated.
- Moving a NameNode from the Ambari HDFS UI when HDFS Transparency is integrated is not supported. To manually move the NameNode, see Moving a NameNode.
- New key value pairs added to the IBM Spectrum Scale Ambari management pack GUI Advance configuration **Custom Add Property** panel are not effective in the IBM Spectrum Scale file system. Therefore, any values not seen in the Standard or Advanced configuration panel will need to be set manually on the command line using the IBM Spectrum Scale `/usr/lpp/mmfs/bin/mmchconfig` command.

IBM Spectrum Scale

- Ensure that bi-directional password-less SSH is set up between all GPFS Nodes.
- The Hadoop services IDs and groups are required to have the same values across the cluster. Any user name needs a user ID in the OS or active directory service when writing to the file system. This is required for IBM Spectrum Scale.
  - **If you are using LDAP/AD**: Create the IDs and groups on the LDAP server, and ensure that all nodes can authenticate the users.
  - **If you are using local IDs**: The IDs must be the same on all nodes with the same ID and group values across the nodes.
- IBM Spectrum Scale only supports installation through a local repository.
- The management pack does not support IBM Spectrum Scale protocol and Transparent Cloud Tiering (TCT) packages.
- Ensure that in an HDFS Transparency environment, the IBM Spectrum Scale file system is set to permit any supported POSIX ACL types. Issue **mmlsfs <Device> -k** to ensure the **-k** value is set to *all*.

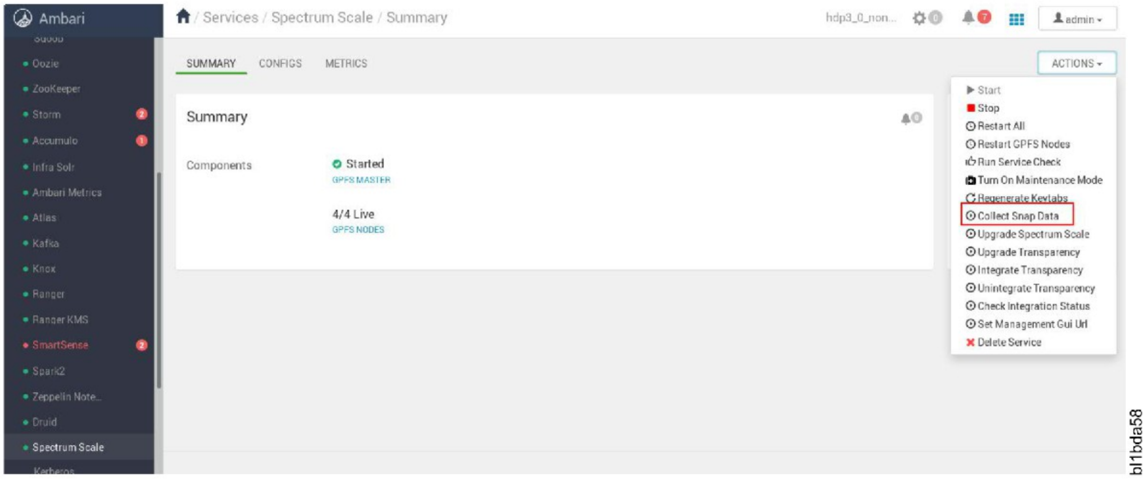
HDP

- The Manage JournalNodes is shown in **HDFS** > **Actions** submenu. This function should not be used when IBM Spectrum Scale service is deployed.
- The + is not supported when using `hftp://namenode:50070`.

Problem determination

Snap data collection

You can collect the IBM Spectrum Scale snap data from the Ambari GUI. The command is run by the IBM Spectrum Scale Master, and the snap data is saved to `/var/log/ambari.gpfs.snap.<timestamp>` on the IBM Spectrum Scale Master node.



By default, the IBM Spectrum Scale Master runs the following command:

`/usr/lpp/mmfs/bin/gpfs.snap -d /var/log/ambari.gpfs.snap.<timestamp> -N <all nodes> --check-space --timeout 600`

Where **`<all nodes>`** is the list of nodes in the IBM Spectrum Scale cluster and in the Ambari cluster. The external nodes in a shared cluster, such as ESS servers, are not included.

Note:

- If your cluster has IBM Spectrum Scale file system version 4.2.2.0 and later, **`gpfs.snap`** will include the `--hadoop` option.
- If you run the Collect Snap Data through Ambari GUI, the Ambari logs will be captured into a tar package under the `/var/log` directory. The base **`gpfs.snap --hadoop option`** command does not capture the Ambari logs. The Ambari logs are only captured by clicking, **IBM Spectrum Scale Service** > **Actions** > **Collect Snap Data** in the Ambari GUI.

You can also override the default behavior of this snap command by providing the arguments to the **`gpfs.snap`** command in the file `/var/lib/ambari-server/resources/gpfs.snap.args`. This works only if you are running the **`gpfs.snap`** on the command line. For example, if you wanted to write the snap data to a different location, collect the snap data from all nodes in the cluster, and increase the timeout. You can provide a **`gpfs.snap.args file`** option similar to that in the following example:

```
# cat /var/lib/ambari-server/resources/gpfs.snap.args
-d /root/gpfs.snap.out -a --timeout 1200
```

The Ambari snap data tar package is stored as `/var/log/ambari.mpack.snap.<TIMESTAMP>.tar.gz` on the IBM Spectrum Scale Master node.

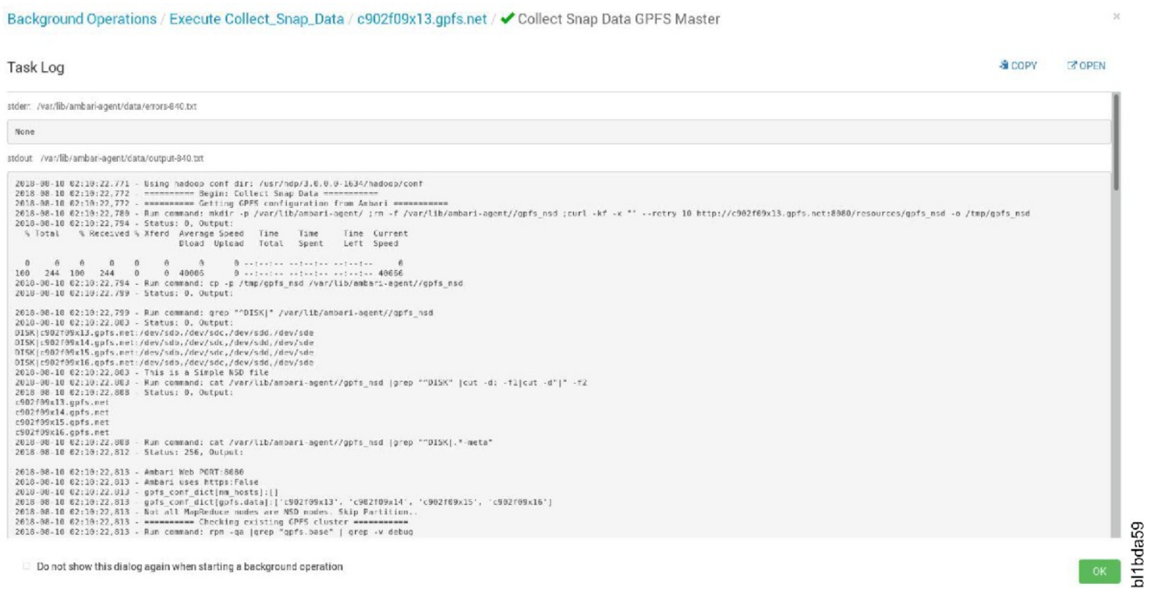
The Ambari snap captures the following information:

1. From all Ambari client:

```
- /var/log/hadoop/root/*
- /var/lib/ambari-agent/data/
- /var/log/ambari-agent/ambari-agent.log
```

2. From Ambari-server:

```
- /var/log/ambari-server/ambari-server.log
- /var/run/ambari-server/stack-recommendations/
```



Background Operations / Execute Collect_Snap_Data / c902f09x13.gpfs.net / ✓ Collect Snap Data GPFS Master
×
Task Log
COPY
OPEN
OK
bl1bda59

stderr: /var/lib/ambari-agent/data/errors-840.txt

```
None
```

stdout: /var/lib/ambari-agent/data/output-840.txt

```
2018-08-10 02:10:22,771 - Using hadoop conf dir: /usr/hdp/3.0.0.0-1634/hadoop/conf
2018-08-10 02:10:22,772 - ========== Begin: Collect Snap Data ===========
2018-08-10 02:10:22,772 - ========== Getting GPFS configuration from Ambari ===========
2018-08-10 02:10:22,780 - Run command: mkdir -p /var/lib/ambari-agent/ ;rm -f /var/lib/ambari-agent//gpfs_nsd ;curl -kf -x "" --retry 10 http://c902f09x13.gpfs.net:8080/resources/gpfs_nsd -o /tmp/gpfs_nsd
2018-08-10 02:10:22,794 - Status: 0, Output:
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed

  0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0
100   244  100   244    0     0  40086      0 --:--:-- --:--:-- --:--:-- 40666
2018-08-10 02:10:22,794 - Run command: cp -p /tmp/gpfs_nsd /var/lib/ambari-agent//gpfs_nsd
2018-08-10 02:10:22,799 - Status: 0, Output:

2018-08-10 02:10:22,799 - Run command: grep "^DISK|" /var/lib/ambari-agent//gpfs_nsd
2018-08-10 02:10:22,803 - Status: 0, Output:
DISK|c902f09x13.gpfs.net:/dev/sdb,/dev/sdc,/dev/sdd,/dev/sde
DISK|c902f09x14.gpfs.net:/dev/sdb,/dev/sdc,/dev/sdd,/dev/sde
DISK|c902f09x15.gpfs.net:/dev/sdb,/dev/sdc,/dev/sdd,/dev/sde
DISK|c902f09x16.gpfs.net:/dev/sdb,/dev/sdc,/dev/sdd,/dev/sde
2018-08-10 02:10:22,803 - This is a Simple NSD file
2018-08-10 02:10:22,803 - Run command: cat /var/lib/ambari-agent//gpfs_nsd |grep "^DISK" |cut -d: -f1|cut -d"|" -f2
2018-08-10 02:10:22,808 - Status: 0, Output:
c902f09x13.gpfs.net
c902f09x14.gpfs.net
c902f09x15.gpfs.net
c902f09x16.gpfs.net
2018-08-10 02:10:22,808 - Run command: cat /var/lib/ambari-agent//gpfs_nsd |grep "^DISK|.*-meta"
2018-08-10 02:10:22,812 - Status: 256, Output:

2018-08-10 02:10:22,813 - Ambari Web PORT:8080
2018-08-10 02:10:22,813 - Ambari uses https:False
2018-08-10 02:10:22,813 - gpfs_conf_dict[nm_hosts]:[]
2018-08-10 02:10:22,813 - gpfs_conf_dict[gpfs.data]:['c902f09x13', 'c902f09x14', 'c902f09x15', 'c902f09x16']
2018-08-10 02:10:22,813 - Not all MapReduce nodes are NSD nodes. Skip Partition..
2018-08-10 02:10:22,813 - ========== Checking existing GPFS cluster ===========
2018-08-10 02:10:22,813 - Run command: rpm -qa |grep "gpfs.base" | grep -v debug
```

□ Do not show this dialog again when starting a background operation

Figure 45. AMBARI COLLECT SNAP DATA

General

Note: For known HDFS Transparency issues, see "HDFS Transparency protocol troubleshooting" on page 212.

1. Data capturing for problem determination

   Solution:

   Capture the following data for problem determination:

   - Failed services and HDFS service logs from the Ambari log outputs from the Ambari UI. The log outputs are seen in the operations logs in the Ambari UI from the `output*.txt` and `error*.txt` outputs.
   - Ambari server and agent log `/var/log/ambari-server/ambari-server.log` and `/var/log/ambari-agent/ambari-agent.log`.
   - Transparency NameNode and DataNode logs.
   - ZKFC log from NameNode host - `/var/log/hadoop/root/hadoop-root-zkfc*.log`.
   - The following software versions:
     - Management pack installed on the Ambari server node, go to the `/var/lib/ambari-server/resources/mpacks` directory and get the directory package installed.
     - HDFS Transparency version: `rpm -qa | grep gpfs.hdfs-protocol`.
     - IBM Spectrum Scale version.
   - `SpectrumScaleMPackInstaller.py/SpectrumScaleMPackUninstaller.py/SpectrumScale_UpgradeIntegrationPackage` scripts failure: Capture the SpectrumScale* log from the directory where the script is located. Any produced *.json files will also reside in this directory.

   Find Ambari Mpack version

   From Mpack 2.7.0.0, get the Mpack version through Ambari GUI service action.

   Otherwise, get the Mpack version through the Ambari directory under `/var/lib/ambari-server`.

For example:

```
/var/lib/ambari-server/resources/extensions/SpectrumScaleExtension/2.4.2.0/services/GPFS
```

This example is using Mpack 2.4.2.0.

2. What IBM Spectrum Scale edition is required for the Ambari deployment?

   Solution: If you want to perform a new installation, including cluster creation and file system creation, use the Standard or Advanced edition because the IBM Spectrum Scale file system policy is used by default. If you only have the Express Edition, select **Deploy HDP** over existing IBM Spectrum Scale file system.

3. Why do I fail in registering the Ambari agent?

   Solution: Run **`ps -elf | grep ambari`** on the failing agent node to see what it is running. Usually, while registering in the agent node, there must be nothing under `/etc/yum.repos.d/`. If there is an additional repository that does not work because of an incorrect path or yum server address, the Ambari agent register operation will fail.

4. Which yum repository must be under `/etc/yum.repos.d`?

   Solution: Before registering, on the Ambari server node, under `/etc/yum.repos.d`, there is only one Ambari repository file that you create in Installing the Ambari server rpm. On the Ambari agent, there must be no repository files related with Ambari. After the Ambari agent has been registered successfully, the Ambari server copies the Ambari repository to all Ambari agents. After that, the Ambari server creates the HDP and HDP-UTILS repository over the Ambari server and agents, according to your specification in the Ambari GUI in **Select Stack** section.

   If you interrupt the Ambari deployment, clean the files before starting up Ambari the next time, especially when you specify a different IBM Spectrum Scale, HDP, or HDP-UTILS yum URL.

5. Must all nodes have the same root password?

   Solution: No, this is unnecessary. You only need to specify the ssh key file for root on the Ambari server.

6. How to check the superuser and the supergroup?

   Solution:

   For HortonWorks HDP 3.0, HDFS Transparency 3.0 has removed the configuration **`gpfs.supergroup`** defined in `/var/mmfs/hadoop/etc/hadoop/gpfs-site.xml`.

   By default, the groups from the configuration **`dfs.permissions.superusergroup`** in `/var/mmfs/hadoop/etc/hadoop/hdfs-site.xml` and the group root are super groups.

7. Why am I unable to connect to the Ambari Server through the web browser?

   Solution: If you cannot connect to the Ambari Server through the web browser, check to see if the following message is displayed in the Ambari Server log which is in `/var/log/ambari-server`:

   ```
   WARN [main] AbstractConnector:335 - insufficient threads configured for
   SelectChannelConnector@0.0.0.0:8080
   ```

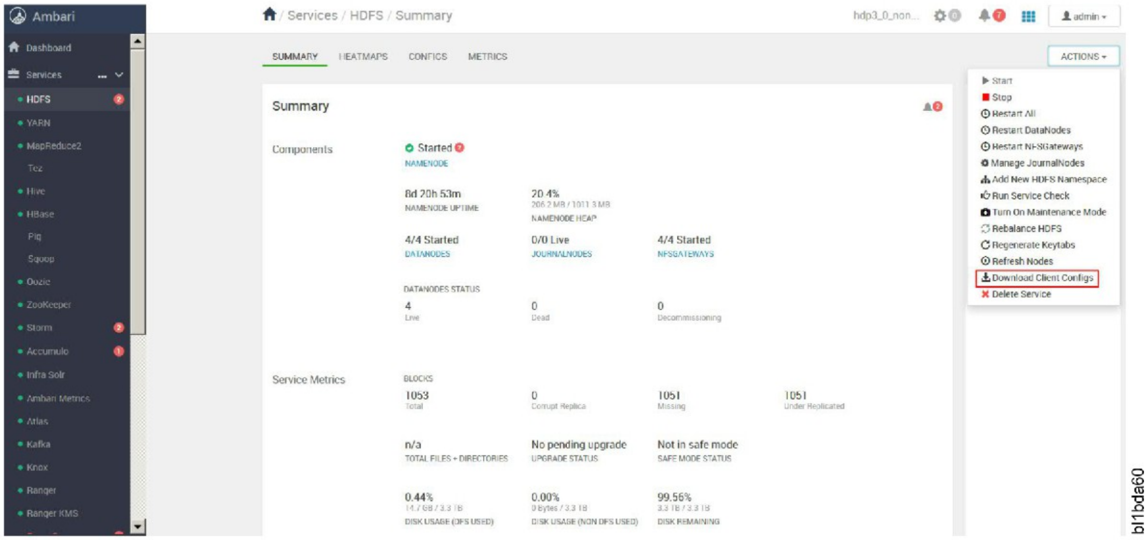
   The size of the thread pool can be increased to match the number of CPUs on the node where the Ambari Server is running.

   For example, if you have 160 CPUs, add the following properties to `/etc/ambari-server/conf/ambari.properties`:

   ```
   server.execution.scheduler.maxThreads=160
   agent.threadpool.size.max=160
   client.threadpool.size.max=160
   ```

8. HDFS Download Client Configs does not contain HDFS Transparency configuration.

   Solution: In the HDFS dashboard, go to **Service Actions** > **Download Client Configs**, the tar configuration downloaded does not contain the HDFS Transparency information.



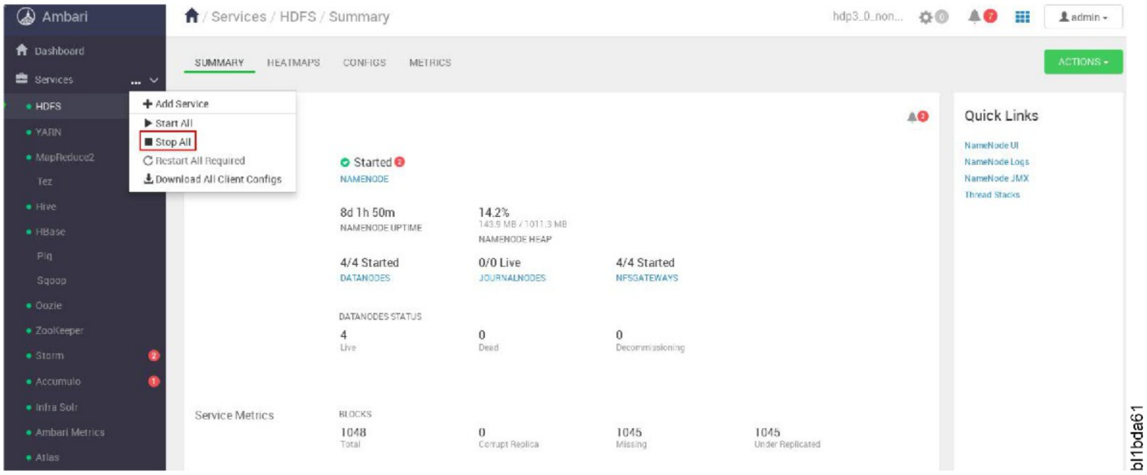
The workaround is to tar up the HDFS Transparency directory.

Run the following command on a HDFS Transparency host to tar up the HDFS Transparency directory into /tmp:

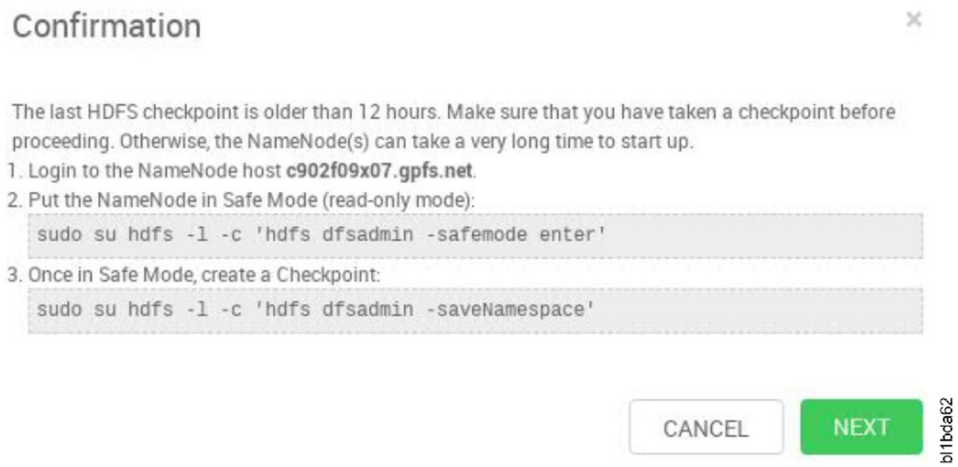
```
# cd /var/mmfs/hadoop/etc/
# tar -cvf /tmp/hdfs.transparency.hadoop.etc.tar hadoop
```

9. HDFS checkpoint confirmation warning message from **Actions** > **Stop All**[1] when integrated with IBM Spectrum Scale.

   Solution: When IBM Spectrum Scale is integrated, the NameNode is stateless. The HDFS Transparency does not support the HDFS **dfsadmin** command.



Therefore, when doing **Ambari dashboard** > **Actions** > **Stop All**[1], Ambari will generate a confirmation box to ask user to do an HDFS checkpoint using the **hdfs dfsadmin -safemode** commands. This is not needed when HDFS Transparency is integrated, and this step can be skipped. Click on next to skip this step.



Confirmation

The last HDFS checkpoint is older than 12 hours. Make sure that you have taken a checkpoint before proceeding. Otherwise, the NameNode(s) can take a very long time to start up.

1. Login to the NameNode host **c902f09x07.gpfs.net**.
2. Put the NameNode in Safe Mode (read-only mode):

```
sudo su hdfs -l -c 'hdfs dfsadmin -safemode enter'
```

3. Once in Safe Mode, create a Checkpoint:

```
sudo su hdfs -l -c 'hdfs dfsadmin -saveNamespace'
```

10. What happens if the Ambari admin password is modified after installation?

    Solution: When the Ambari admin password was modified, the new password is required to be set in the IBM Spectrum Scale service.

    To change the Ambari admin password in IBM Spectrum Scale, follow these steps:

    - Log in to the Ambari GUI.
    - Click **Spectrum Scale** > **Configs tab** > **Advanced tab** > **Advanced gpfs-ambari-server-env** > **AMBARI_USER_PASSWORD** to update the Ambari admin password.

    If the Ambari admin password is not modified in the IBM Spectrum Scale Advanced configuration panel, starting Ambari services might fail. For example, Hive starting fails with exception errors.

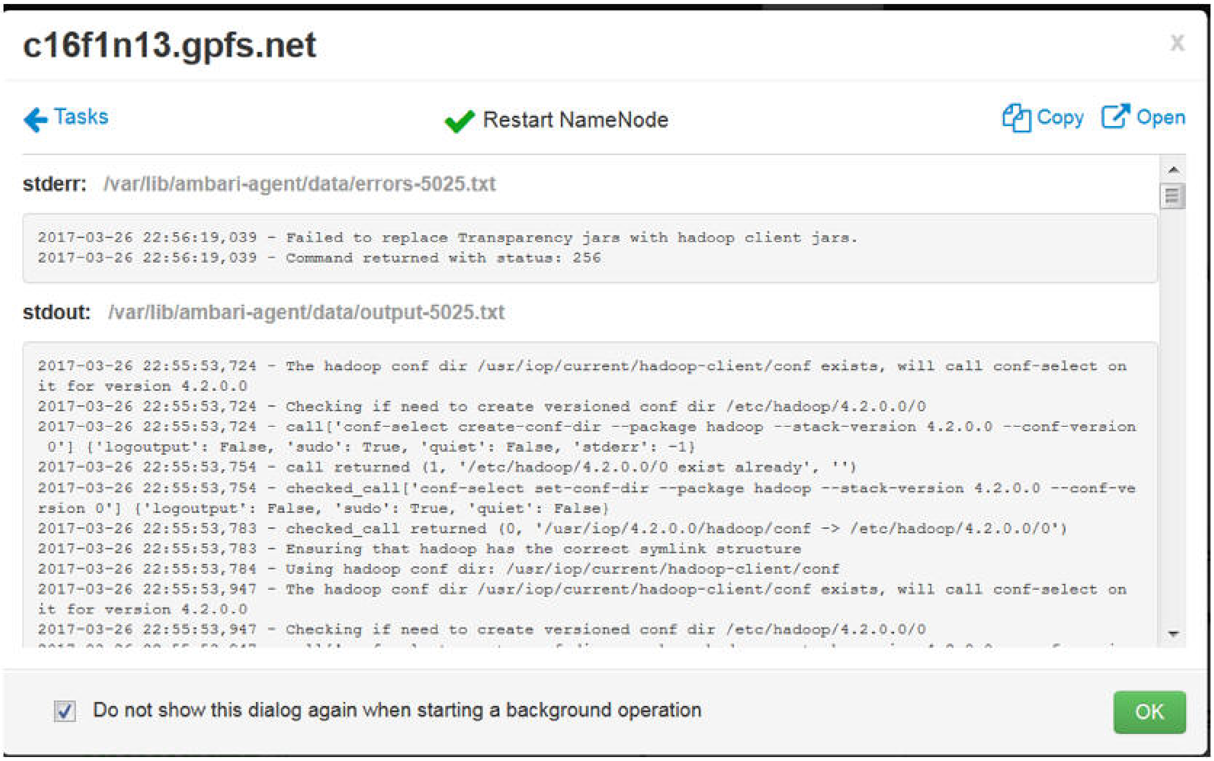
11. Kerberos authentication error during Unintegrate Transparency action

    ```
    ERROR: Kerberos Authentication Not done Successfully. Exiting Unintegration.
    Enter Correct Credentials of Kerberos KDC Server in Spectrum Scale Configuration.
    ```

    Solution:

    If error occurs in a Kerberos environment, check to ensure that the KDC_PRINCIPAL and KDC_PRINCIPAL_PASSWORD values in **Spectrum Scale services** > **Configs** > **Advanced tab** have the correct values. Save the configuration changes.

12. NameNodes and DataNodes failed with the error `Fail to replace Transparency jars with hadoop client jars` when short-circuit is enabled.



stderr: /var/lib/ambari-agent/data/errors-5025.txt

```
2017-03-26 22:56:19,039 - Failed to replace Transparency jars with hadoop client jars.
2017-03-26 22:56:19,039 - Command returned with status: 256
```

stdout: /var/lib/ambari-agent/data/output-5025.txt

```
2017-03-26 22:55:53,724 - The hadoop conf dir /usr/iop/current/hadoop-client/conf exists, will call conf-select on
it for version 4.2.0.0
2017-03-26 22:55:53,724 - Checking if need to create versioned conf dir /etc/hadoop/4.2.0.0/0
2017-03-26 22:55:53,724 - call['conf-select create-conf-dir --package hadoop --stack-version 4.2.0.0 --conf-version
 0'] {'logoutput': False, 'sudo': True, 'quiet': False, 'stderr': -1}
2017-03-26 22:55:53,754 - call returned (1, '/etc/hadoop/4.2.0.0/0 exist already', '')
2017-03-26 22:55:53,754 - checked_call['conf-select set-conf-dir --package hadoop --stack-version 4.2.0.0 --conf-ve
rsion 0'] {'logoutput': False, 'sudo': True, 'quiet': False}
2017-03-26 22:55:53,783 - checked_call returned (0, '/usr/iop/4.2.0.0/hadoop/conf -> /etc/hadoop/4.2.0.0/0')
2017-03-26 22:55:53,783 - Ensuring that hadoop has the correct symlink structure
2017-03-26 22:55:53,784 - Using hadoop conf dir: /usr/iop/current/hadoop-client/conf
2017-03-26 22:55:53,947 - The hadoop conf dir /usr/iop/current/hadoop-client/conf exists, will call conf-select on
it for version 4.2.0.0
2017-03-26 22:55:53,947 - Checking if need to create versioned conf dir /etc/hadoop/4.2.0.0/0
```

☑ Do not show this dialog again when starting a background operation

Solution: Install the Java OpenJDK development tool-kit package, `java-<version>-openjdk-devel`, on all the Transparency nodes. Ensure that the version is compatible with your existing JDK version. See HDFS Transparency package.

13. ssh rejects additional ssh connections which causes the HDFS Transparency syncconf connection to be rejected.

    Solution: If the **`ssh maxstartup`** value is too low, then the ssh connections can be rejected.

    Review the ssh configuration values, and increase the maxstartup value.

    For example:

    Review ssh configuration:

    ```
    # sshd -T | grep -i max
    maxauthtries 6
    maxsessions 10
    clientalivecountmax 3
    maxstartups 10:30:100
    ```

    Modify the ssh configuration: Modify the `/etc/ssh/sshd_config` file to set the **`maxstartup`** value.

    ```
    maxstartups 1024:30:1024
    ```

    Restart the ssh daemon:

    ```
    # service sshd restart
    ```

14. Not able to view Solr audits in Ranger.

    Solution: To resolve this issue:

    a. Remove the solr ranger audit write lock file if it exists as root or as the owner of the file.

    ```
    $ ls /bigpfs/apps/solr/data/ranger_audits/core_node1/data/index/write.lock
    $ rm /bigpfs/apps/solr/data/ranger_audits/core_node1/data/index/write.lock
    ```

    b. Restart HDFS and Solr.

    Click **Ambari GUI** > **HDFS** > **Actions** > **Restart All**

Click **Ambari GUI** > **Solr** > **Actions** > **Restart All**

15. On restarting the service that failed due to network port being in use, the NameNode is still up after doing a STOP ALL[1] from Ambari GUI or HDFS service > STOP.

    Solution: As a root user, ssh to the NameNode to check if the NameNode is up:

    ```
    # ps -ef | grep namenode
    ```

    If it exists, then kill the NameNode pid

    ```
    # kill -9 namenode_pid
    ```

    Restart the service.
16. UID/GID failed with illegal value `Illegal value: USER = xxxxx > MAX = 8388607`

    Solution: If you have installed Ranger, and you need to leverage Ranger capabilities, then you need to make the UID/GID less than 8388607.

    If you do not need Ranger, follow these steps to disable Ranger from HDFS Transparency:

    a. On the Ambari GUI, click **IBM Spectrum Scale** > **Configs** and set the **Add `gpfs.ranger.enabled`** to *false*.
    b. Save the configuration.
    c. Restart IBM Spectrum Scale.
    d. Restart HDFS.
17. What to do when I see performance degradation when using HDFS Transparency version 2.7.3-0 and earlier?

    Solution:

    For HDFS Transparency version 2.7.3-0 and below, if you see performance degradation and you are not using Ranger, set the **`gpfs.ranger.enabled`** to *false*.

    a. On the Ambari GUI, click **Spectrum Scale** > **Configs** > **Advanced** > **Custom gpfs-site** and set the **Add `gpfs.ranger.enabled`** to *false*.
    b. Save the configuration.
    c. Restart IBM Spectrum Scale.
    d. Restart HDFS.
18. Why did the IBM Spectrum Scale service not stop or restart properly?

    This can be a result of a failure to unmount the IBM Spectrum Scale file system which may be busy. See the IBM Spectrum Scale operation task output in Ambari to verify the actual error messages.

    Solution:

    Stop all services. Ensure the IBM Spectrum Scale file system is not being accessed either via HDFS or POSIX by running the **`lsof`** or **`fuser`** command. Stop or restart the IBM Spectrum Scale service again.

    For FPO cluster, do not run STOP ALL from the Ambari GUI. Refer to the Limitations > General section on how to properly stop IBM Spectrum Scale.
19. IBM Spectrum Scale service cannot be deployed in a non-root environment.

    Solution:

    If the deployment of IBM Spectrum Scale service in a non-root environment fails with the `Error message: Error occurred during stack advisor command invocation: Cannot create /var/run/ambari-server/stack-recommendations`, go to I cant add new services into ambari.
20. User permission denied when Ranger is disabled.

If Kerberos is enabled and Ranger is disabled, the user gets the permission denied errors when accessing the file system for HDFS Transparency 3.0.0 and earlier.

Solution:

Check the Kerberos principal mapping **`hadoop.security.auth_to_local`** field in the `/var/mmfs/hadoop/etc/hadoop/core-site.xml` or in Ambari under HDFS Config to ensure that the NameNode and DataNode are mapped to root instead of HDFS. For example, change

```
FROM:
RULE:[2:$1@$0](dn@COMPANY.DIV.COM)s/.*/hdfs/
RULE:[2:$1@$0](nn@COMPANY.DIV.COM)s/.*/hdfs/

TO:
RULE:[2:$1@$0](dn@COMPANY.DIV.COM)s/.*/root/
RULE:[2:$1@$0](nn@COMPANY.DIV.COM)s/.*/root/
```

Restart the HDFS service in Ambari or HDFS Transparency by using the following command:

```
/usr/lpp/mmfs/bin/mmhadoopctl connector stop; /usr/lpp/mmfs/bin/mmhadoopctl connector start
```

21. Updating ulimit settings for HDFS Transparency.

    After updating the ulimit values on your nodes, perform the following procedure for HDFS Transparency to pick up the ulimit values properly.

    Solution:

    a. Restart each node’s Ambari agent by issuing the following command:

    ```
    ambari-agent restart
    ```

    b. Restart HDFS service from Ambari.

22. In Kerberized environment, getting Ambari error due to user fail to authenticate.

    If Kerberos is enabled and the uid got changed, the Kerberos ticket cache will be invalid for that user.

    Solution:

    If the user fails to authenticate, run the **`kinit list`** command to find the path to the ticket cache and remove the `krb5*` files.

    For example:

    As a user, run the **`kinit list`**.

    Check the **`Ticket cache`** value (For example, Ticket cache: FILE: `/tmp/krb5cc_0`).

    Remove the `/tmp/krb5cc_0` file from all nodes.

    Note: Kerberos regenerates the file on the node.

23. Quicklinks NameNode GUI are not accessible from HDFS service in multihomed network environment.

    In multihomed networks, the cluster nodes are connected to more than one network interface.

    The Quicklinks from HDFS service are not accessible with the following errors:

    ```
    This site can't be reached.
    <Host> refused to connect.
    ERR_CONNECTION_REFUSED
    ```

    Solution:

    For fixing the NameNode binding so that HDFS service NameNode UI can be accessed properly, see the following Hortonworks documentation:

    - Fixing Hadoop issues In Multihomed Environments
    - Ensuring HDFS Daemons Bind All Interfaces.

Ensure that you do a HDFS service restart after changing the values in HDFS configuration in Ambari.

24. **`Enable Kerberos`** action fails.

    Solution:

    If the IBM Spectrum Scale service is integrated, **`Enable Kerberos`** action might fail due to an issue with GPFS Service Check underneath. In such cases, retry the operation.

25. Enable the autostart of services when IBM Spectrum Scale is integrated.

    Solution:

    a. In Ambari GUI, go to **Admin** > **Service Auto Start Configuration** and enable autostart.

    b. Enable autoload and automount on the IBM Spectrum Scale cluster (on the HDP cluster side).

    c. If ESS is being used, enable autoload on the ESS cluster.

    For more information, see IBM Spectrum Scale **`mmchfs fsname -A yes`** for automount and **`mmchconfig autoload=yes`** commands.

26. GPFS Master fails with the error message: `The UID and GID of the user "anonymous" is not uniform across all the IBM Spectrum Scale hosts`.

    Solution:

    a. Ensure that the userid/groupid for the user *anonymous* are uniform across all the GPFS hosts in the cluster. Correct the inconsistent values on any GPFS host.

    b. If there is no *anonymous* userid/groupid existing on a GPFS host, ensure that you create the same *anonymous* userid/groupid value as all the other GPFS hosts' *anonymous* userid/groupid value in the same IBM Spectrum Scale cluster.

       Example on how to create the *anonymous* user as a regular OS user across all the GPFS hosts. If you are using LDAP or other network authentication service, refer to their respective documentation.

       Create the GID first by running the following command:

       ```
       mmdsh -N all groupadd -g <common group ID> anonymous
       ```

       where, `<common group ID>` can be set to a value like *11888*.

       Create the UID by running the following command:

       ```
       mmdsh -N all useradd -u <common group ID> anonymous -g anonymous
       ```

       where, `<common group ID>` can be set to a value like *11889*.

27. IBM Spectrum Scale installation fails during deployment in Ambari due to scripts not found error.

    `stdout: /var/lib/ambari-agent/data/output-402.txt` Caught an exception while executing custom service command:

    ```
    <class 'ambari_agent.AgentException.AgentException'>:
    'Script /var/lib/ambari-agent/cache/extensions/SpectrumScaleExtension
    /2.7.0.1/services/GPFS/package/scripts/slave.py does not exist';
    ```

    Solution:

    See Ambari Release Notes SPEC-57 for resolution.

28. IBM Spectrum Scale service installation in Ambari fails in the stack-advisor because the default login shell used does not propagate the error return code of zero for the shell command properly.

    The log file: `/var/run/ambari-server/stack-recommendations/<number>/stackadvisor.out` shows errors:

    `mmlsfs`: No file systems were found.

    `mmlsfs`: Command failed. Examine previous error messages to determine cause.

Error occured in the stack advisor.

Error details: local variable 'mount' referenced before assignment.

Solution:

Check to see if the default login shell returns an error return code of zero '0' for the failed command. If successful, the command should return a value > 0.

Run the following command on the Ambari server:

```
ssh -q -o BatchMode=yes -o StrictHostKeyChecking=no <USER>@<AMBARI-HOST-NAME> "sudo cat /
notpresentfile"
```

The <USER> is either the root or non-root Ambari user depending on how Ambari was configured. If the command returns a zero '0' return code, then you need to update the default login shell to use the 'bash' shell for the Ambari user.

29. Unable to stop IBM Spectrum Scale service in Mpack 2.7.0.4.

    In Mpack 2.7.0.4, if **gpfs.storage.type** is set to shared, stopping the IBM Spectrum Scale service from Ambari reports a failure in the UI even if the operation had succeeded internally.

    Solution:

    To workaround this issue:

    a. Before you stop IBM Spectrum Scale or do a STOP All, set the IBM Spectrum Scale service to maintenance mode.
    b. On the command line, stop IBM Spectrum Scale using the **mmshutdown** command.

       ```
       # /usr/lpp/mmfs/bin/mmshutdown -a
       ```

    c. Put the IBM Spectrum Scale service out of maintenance mode.
    d. Start the IBM Spectrum Scale service or do a Start All using Ambari.

30. If SSL is enabled in Ambari, running the `SpectrumScaleMPackUninstaller.py` script to uninstall the IBM Spectrum Scale Mpack with an IP address might fail with a certificate error during the validation of the Ambari server's credentials.

    Solution:

    Depending on the SSL certificate that the Ambari server is registered with (hostname or IP address), using the IP address of the Ambari server while running the `SpectrumScaleMPackUninstaller.py` script can give a certificate error because the certificate is registered with the hostname. Therefore, provide the Ambari server's hostname instead of the IP address when the Mpack Uninstaller scripts prompts for the Ambari server IP address.

31. Ambari 2.7.X adding additional directories during deployment.

    Solution:

    For HDP 3.X using Ambari 2.7.X, Ambari will add directories in addition to the default `/hadoop/hdfs` directory path. Ensure that you review the HDFS NameNode and DataNode directories and Yarn local directories and other directories listed in the "Customize services" on page 362 directories to ensure that only the required directories are listed.

    For example, when integrating/unintegrating Scale service:

    ```
    DFS NameNode : /hadoop/hdfs/namenode,/.snapshots/hadoop/hdfs/namenode,
    /opt/hadoop/hdfs/namenode,/srv/hadoop/hdfs/namenode,/usr/local/hadoop/hdfs/namenode,
    /var/cache/hadoop/hdfs/namenode,/var/crash/hadoop/hdfs/namenode,
    /var/lib/libvirt/images/hadoop/hdfs/namenode,/var/lib/machines/hadoop/hdfs/namenode,
    /var/lib/mailman/hadoop/hdfs/namenode,/var/lib/mariadb/hadoop/hdfs/namenode,
    /var/lib/mysql/hadoop/hdfs/namenode,/var/lib/named/hadoop/hdfs/namenode,
    /var/lib/pgsql/hadoop/hdfs/namenode,/var/log/hadoop/hdfs/namenode,
    /var/opt/hadoop/hdfs/namenode,/var/spool/hadoop/hdfs/namenode,/var/tmp/hadoop/hdfs/namenode
    ```

```
DFS DataNode /hadoop/hdfs/data,/.snapshots/hadoop/hdfs/data,/opt/hadoop/hdfs/data,
/srv/hadoop/hdfs/data,/usr/local/hadoop/hdfs/data,/var/cache/hadoop/hdfs/data,
/var/crash/hadoop/hdfs/data,/var/lib/libvirt/images/hadoop/hdfs/data,
/var/lib/machines/hadoop/hdfs/data,/var/lib/mailman/hadoop/hdfs/data,
/var/lib/mariadb/hadoop/hdfs/data,/var/lib/mysql/hadoop/hdfs/data,
/var/lib/named/hadoop/hdfs/data,/var/lib/pgsql/hadoop/hdfs/data,/var/log/hadoop/hdfs/data,
/var/opt/hadoop/hdfs/data,/var/spool/hadoop/hdfs/data,/var/tmp/hadoop/hdfs/data
```

Even though HDFS Transparency does not use the NameNode and DataNode listed above, the native HDFS will need to use them.

The default directory path is `/hadoop/hdfs/namenode` and `/hadoop/hdfs/data`. All other directories are not needed.

32. Ambari 2.7.x - Cannot find a valid baseurl for repo.

    For Ambari 2.7.x, Ambari writes empty baseurl values to the repo files when using a local repository causing stack installation failures.

    Solution:

    See AMBARI-25069/SPEC-58/BUG-116328 workaround:

    For Ambari 2.7.0.0: Ambari 2.7.0 Known Issues.

    For Ambari 2.7.1.0: Ambari 2.7.1 Known Issues.

    For Ambari 2.7.3.0: Ambari 2.7.3 Known Issues.

33. The IBM Spectrum Scale Mpack installer fails with `No JSON object could be decoded` error.

    If the Ambari certificate is expired or self-signed or is invalid, the Mpack installation fails while executing the REST API calls.

    Error seen:

```

    INFO: ***Starting the Spectrum Scale Mpack Installer v2.7.0.7***
    Enter the Ambari server host name or IP address. If SSL is configured, enter host name,
to verify the SSL certificate. Default=192.0.2.22  :   c902f09x05.gpfs.net
    Enter Ambari server port number. If it is not entered, the installer will take default
port 8080  :   9443
    Enter the Ambari server username, default=admin  :   admin
    Enter the Ambari server password  :
    INFO: Verifying Ambari server address, username and password.
    Traceback (most recent call last):
    File "./SpectrumScaleMPackInstaller.py", line 312, in <module>
        InstallMpack(**darg)
    File "./SpectrumScaleMPackInstaller.py", line 162, in InstallMpack
        cluster_details = verify(ambari_hostname.strip(), ambari_username.strip(),
ambari_password, ambari_port)
    File "/root/mpack2707/mpack_utils.py", line 417, in verify
        clusters_json=json.loads(result)
    File "/usr/lib64/python2.7/json/__init__.py", line 338, in loads
        return _default_decoder.decode(s)
    File "/usr/lib64/python2.7/json/decoder.py", line 366, in decode
        obj, end = self.raw_decode(s, idx=_w(s, 0).end())
    File "/usr/lib64/python2.7/json/decoder.py", line 384, in raw_decode
        raise ValueError("No JSON object could be decoded")
    ValueError: No JSON object could be decoded
    SpectrumScaleMPackInstaller failed.
```

    Solution:

    The following are the two possible solutions:

    a. Enable `urllib2` to work with the self-signed certificate by setting **`verify`** to *disable* in the `/etc/python/cert-verification.cfg` file. For more information, see Certificate verification in Python standard library HTTP clients.
    b. Configure Ambari with the correct SSL certificate.

34. Mpack Installation / Uninstallation fails while restarting Ambari due to `Server not yet listening on http port` timeout error.

Error seen:

```
    ERROR: Failed to run Ambari server restart command, with error: Using python  /usr/bin/
python
    Restarting ambari-server
    Waiting for server stop...
    Ambari Server stopped
    Ambari Server running with administrator privileges.
    Organizing resource files at /var/lib/ambari-server/resources...
    Ambari database consistency check started...
    Server PID at: /var/run/ambari-server/ambari-server.pid
    Server out at: /var/log/ambari-server/ambari-server.out
    Server log at: /var/log/ambari-server/ambari-server.log
    Waiting for server
start.............................................................................
.............
    DB configs consistency check found warnings. See /var/log/ambari-server/ambari-server-
check-database.log for more details.
    ERROR: Exiting with exit code 1.
    REASON: Server not yet listening on http port 8080 after 90 seconds. Exiting..
```

Solution:

a. Increase the timeout by adding or updating the **server.startup.web.timeout** property on the Ambari server to 180 seconds in the /etc/ambari-server/conf/ambari.properties file. For more information, see change the port for ambari server.

b. Retry the Mpack install / uninstall procedure.

[1] For FPO cluster, do not run STOP ALL from the Ambari GUI. Refer to the Limitations > General section on how to properly stop IBM Spectrum Scale.

Troubleshooting Ambari

This section contains troubleshooting information for Ambari issues.

1. Problem determination information

   Solution:

   Capture the following data for problem determination:

   - Failed services and HDFS service logs from the Ambari log outputs from the Ambari UI. The log outputs are seen in the operations logs in Ambari UI from the output*.txt and error*.txt outputs
   - Ambari server and agent log /var/log/ambari-server/ambari-server.log and /var/log/ambari-agent/ambari-agent.log
   - Transparency NameNode and DataNode logs (See HDFS Transparency Debugging in previous slide)
   - ZKFC log from NameNode host - /var/log/hadoop/root/hadoop-root-zkfc*.log
   - Software versions
     - Management pack installed: On the Ambari server node, go to directory /var/lib/ambari-server/resources/mpacks and get the directory package installed.
     - HDFS Transparency version: rpm -qa | grep gpfs.hdfs-protocol.
     - IBM Spectrum Scale version.
   - SpectrumScaleMPackInstaller.py/SpectrumScaleMPackUninstaller.py/SpectrumScale_UpgradeIntegrationPackage scripts failure: Capture the SpectrumScale* log from the directory where the script is located. Any produced *.json files will also reside in this directory.

   Find Ambari Mpack version

   From Mpack 2.7.0.0, get the Mpack version through "Verify IBM Spectrum Scale Mpack version" on page 434. Otherwise, get the Mpack version through the Ambari directory under /var/lib/ambari-server.

For example: `/var/lib/ambari-server/resources/extensions/SpectrumScaleExtension/2.4.2.0/services/GPFS`

This example is using Mpack 2.4.2.0.

2. Disable Ranger in Ambari

   Solution:

   In HDFS Transparency version 2.7.2-1, a new property field can be used to disable Ranger. To disable Ranger through Ambari, go to **Ambari GUI** > **IBM Spectrum Scale** > **Configs** > **Advanced** > **Custom gpfs-site** > **Add property** and set the key field to **`gpfs.ranger.enabled`** and the value field to *false*. Save the config, restart IBM Spectrum Scale and then restart HDFS to sync this value to all the nodes.

3. IBM Spectrum Scale service missing from Ambari when installed after HDP and BigSQL version 5.0

   Solution:

   In GPFS Ambari integration module version 2.4.2.0, if the IBM Spectrum Scale service is missing from Ambari when installed after HDP and BigSQL version 5.0, then the following manual steps are required to enable it.

   Note: This example uses http. If your cluster uses https, replace the http with https.

   Where,

   ```
   <Ambari_server_ip> = Ambari server ip address

   <Ambari_server_port> = Ambari server port

   <Ambari admin id> = The Ambari admin id

   <Ambari admin passwd> = The Ambari admin password value
   ```

   a. Check the IBM Spectrum Scale service in Ambari database.

      Note: The GPFS Ambari extension link is missing even though GPFS Ambari `/var/lib/ambari-server/resources/mpacks/SpectrumScaleExtension-MPack-2.4.2.0/extensions/SpectrumScaleExtension` directory exists.

      Run the command:

      ```
      # curl -u <Ambari admin id>:<Ambari admin passwd> -H 'X-Requested-By:ambari' -X
      GET 'http://<Ambari_server_ip>:<Ambari_server_port>/api/v1/links'
      ```

      Note: The GPFS Ambari extension is missing. Only the IBM-Big_SQL extension is seen.

   b. Create the GPFS Ambari extension link.

      Run the command:

      ```
      # curl -u a<Ambari admin id>:<Ambari admin passwd> -H 'X-Requested-By: ambari' -X
      POST -d '{"ExtensionLink": {"stack_name":"HDP", "stack_version": "2.6",
      "extension_name":
      "SpectrumScaleExtension", "extension_version": "2.4.2.0"}}' http://
      <Ambari_server_ip>/api/v1/links/
      ```

   c. Check to ensure that the extension link for GPFS Ambari is created.

      Run the command:

      ```
      # curl -u <Ambari admin id>:<Ambari admin passwd> -H 'X-Requested-By:ambari' -X
      GET 'http://<Ambari_server_ip>:<Ambari_server_port>/api/v1/links'
      ```

      For example, you should see similar information:

      ```
        {

            "href" : "http://9.30.95.166:8081/api/v1/links/51",

            "ExtensionLink" : {
      ```

```
        "extension_name" : "SpectrumScaleExtension",

        "extension_version" : "2.4.2.0",

        "link_id" : 51,

        "stack_name" : "HDP",

        "stack_version" : "2.6"

      }

    }
```

d. Restart Ambari server.

   Run the command:

```
# ambari-server restart
```

e. Log into Ambari GUI, ensure that the IBM Spectrum Scale service is visible in the GUI.

4. IBM Spectrum Scale service failed to stop or restart.

   This can be a result of a failure to unmount the IBM Spectrum Scale file system which may be busy. See the IBM Spectrum Scale operation task output in Ambari to verify the actual error messages.

   Solution:

   Stop all the services. Ensure that the IBM Spectrum Scale file system is not being accessed either by using HDFS or POSIX by running the lsof or fuser command. Stop or restart the IBM Spectrum Scale service again.

5. Missing IBM Spectrum Scale Quick link in IBM Spectrum Scale Ambari Mpack 2.4.2.1.

   In IBM Spectrum Scale Ambari Mpack version 2.4.2.1, the IBM Spectrum Scale Quick link is missing.

   Solution:

   a. On the Ambari server, edit the `/var/lib/ambari-server/resources/mpacks/SpectrumScaleExtension-MPack-2.4.2.1/extensions/SpectrumScaleExtension/2.4.2.1/services/GPFS/quicklinks/quicklinks.json` file to add in the field "component_name": "GPFS_MASTER" under the **links** > **url** with the "IBM Spectrum Scale Management UI" label section.

```
        "links": [
            {
                "requires_user_name": "false",
                "name": "spectrum_scale_management_ui",
                "url": "https://c902f09x05.gpfs.net",
                "label": "Spectrum Scale Management UI",
                "port": {
                    "regex": "\\w*:(\\d+)"
                },
                "component_name": "GPFS_MASTER"
            }
        ]
```

   b. Restart Ambari server using the following command:

```
/usr/sbin/ambari-server restart
```

6. In Ambari, when Ranger is enabled NameNode failed to start.

   In Mpack version 2.4.2.3 and below, if Ranger is enabled through Ambari, the HDFS Transparency NameNode might fail to start and no logging is seen.

   Solution 1:

   Restart the HDFS service again though the HDFS Ambari UI. If the NameNodes are still not able to come up, then follow Solution 2.

Solution 2:

Depending on the system installed packages, the CLASSPATH set from `/usr/share/java/*.jar` might contain jars that are not valid for HDFS Transparency.

HDFS Transparency only requires the `/usr/share/java/mysql-connector-java.jar` to be set in the CLASSPATH and not all the jars from `/usr/share/java`.

There are 2 connector.py files that need to be patched to ensure that the CLASSPATH is set properly through Ambari.

a. From Scale path: `/var/lib/ambari-server/resources/mpacks/SpectrumScaleExtension-MPack-<version>/extensions/SpectrumScaleExtension/<version>/services/GPFS/package/scripts/connector.py`
b. From HDFS path: `/var/lib/ambari-server/resources/common-services/HDFS/<version>/package/scripts/connector.py`

Steps:

a. Save copy of the connector.py from both the Scale path and HDFS path.
b. Edit the Scale path connector.py to change the following:

   FROM

```
            line4="for f in /usr/share/java/*.jar; do"

            line5="  export HADOOP_CLASSPATH=$HADOOP_CLASSPATH:$f"
            line6="done"

            f.write("\n")
            f.write("\n")
            f.write(line1)
            f.write("\n")
            f.write(line2)
            f.write("\n")
            f.write(line3)
            f.write("\n")
            f.write(line4)
            f.write("\n")
            f.write(line5)
            f.write("\n")
            f.write(line6)
            f.write("\n")
```

   TO

```

            # Change line4 to explicitly set only the mysql-connector-java.jar

             line4="  export HADOOP_CLASSPATH=$HADOOP_CLASSPATH:/usr/share/java/mysql-
connector-java.jar"

            # Remove line5 and line6

            f.write("\n")

            f.write("\n")
            f.write(line1)
            f.write("\n")
            f.write(line2)
            f.write("\n")
            f.write(line3)
            f.write("\n")
            f.write(line4)
            f.write("\n")
            # Remove line5 and line6 and extra newlines
```

c. Copy the Scale path connector.py to the HDFS path.
d. Restart Ambari

```
# /usr/sbin/ambari-server restart
```

7. Mapreduce service check/jobs failed due to permission failure.

   Mapreduce service check time out or job failed due to permission failure for yarn.

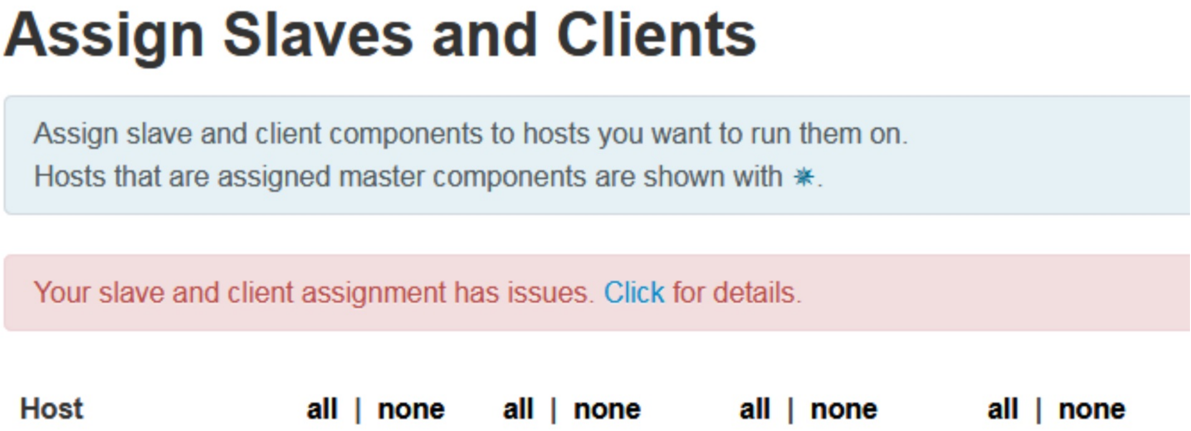
   Solution:

   For Yarn, ensure that the **`yarn yarn.nodemanager.local-dirs`** and **`yarn.nodemanager.log-dirs`** are writable for the user yarn. If not, add write permission to the **`yarn.nodemanager.local-dirs`** and **`yarn.nodemanager.log-dirs`** directories for the user yarn.

8. Getting `Spectrum Scale is stopped` issue in **Assign Slaves and Clients** panel

   For IBM Spectrum Scale Mpack 2.4.2.2 and below:

   When deploying the IBM Spectrum Scale service onto a pre-existing IBM Spectrum Scale cluster, the **Assign Slaves and Clients** panel will show there is an issue:



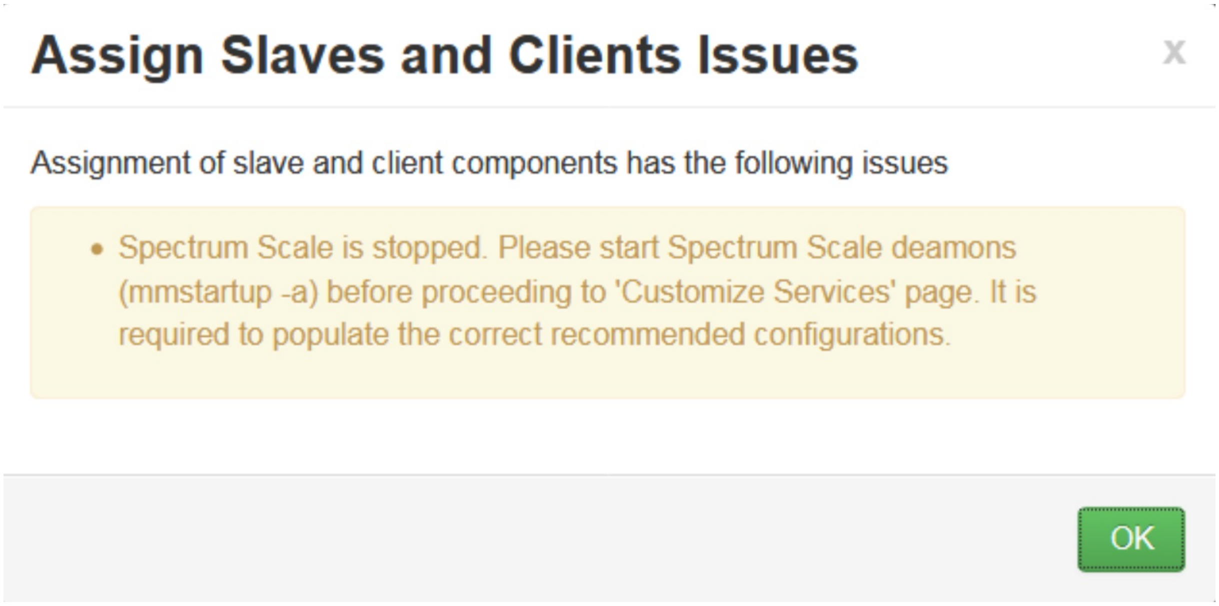
Assign Slaves and Clients

Assign slave and client components to hosts you want to run them on.
Hosts that are assigned master components are shown with ✱.

Your slave and client assignment has issues. Click for details.

The "Click for details" will show a pop up panel that states that the IBM Spectrum Scale daemons are stopped.

For pre-existing cluster, the IBM Spectrum Scale daemons are required to be active and the mount points are available.



Assign Slaves and Clients Issues

Assignment of slave and client components has the following issues

- Spectrum Scale is stopped. Please start Spectrum Scale deamons (mmstartup -a) before proceeding to 'Customize Services' page. It is required to populate the correct recommended configurations.

However, when you are executing the **`/usr/lpp/mmfs/bin/mmgetstate -a`** command, the IBM Spectrum Scale cluster is shown as already being active. The **`/usr/lpp/mmfs/bin/mmlsmount {Device | all, }`** command also shows that all the mounts points are available on all the nodes.

Solution:

For pre-existing IBM Spectrum Scale cluster, if the **mmgetstate -a** shows that the nodes in the cluster are not active, then ensure that you start IBM Spectrum Scale up by executing the **/usr/lpp/mmfs/bin/mmstartup -a** command. Ensure that IBM Spectrum Scale is active and mount points are available on all the nodes before deploying the IBM Spectrum Scale service.

For pre-existing IBM Spectrum Scale cluster, if the **mmgetstate -a** shows that the nodes in the cluster are active and all the mount points are available on all the nodes, then ignore this issue pop up panel and continue with the deployment of the IBM Spectrum Scale service by clicking OK. This is a known bug and is being tracked internally.

9. Yarn Timeline Service 2.0 fails to start

   HDP 3.0:

   The Timeline Service 2.0 in Yarn fails to start.

   Solution:

   There is a new implementation of the Timeline service in HDP 3.0 named as Timeline Service 2.0. It can run in 2 modes (Embedded mode or System service mode) depending on the cluster capacity. To check the mode that is being set, filter the search for **is_hbase_system_service_launch** under the YARN configuration. If this value is checked, it is running in the system service mode. If it is running in the system service mode, follow the set of best practices from the Hortonworks website.

   The following important steps should be performed after Integrating/UnIntegrating the IBM Spectrum Scale service and Enabling/Disabling the Kerberos: Managing Data Operating System

   If the ERROR `client.ApiServiceClient: Failed to destroy service ats-hbase, because it is still running` error is seen in the step 1 mentioned above, perform the following steps:

   a. Check the status of the ats-hbase service by executing the following command:

   ```
   yarn app -status ats-hbase
   ```

   b. If the state is STOPPED, perform the following:

   Get the application_ID from the ResourceManager UI from the Ambari GUI and then run:

   ```
   yarn -kill -appId <application_ID>
   yarn app -destroy ats-hbase
   ```

   You might need to remove the following directory:

   `/<gpfs.mnt.dir value>/<gpfs.data.dir value>/user/yarn-ats/{stack-version}`

   For example,

   ```
   rm -rf /gpfs/datadir_1/user/yarn-ats/{stack-version}
   ```

   c. Run all the service checks to ensure that all the services are successful.

10. IBM Spectrum Scale NSD are not able to be recovered in FPO clusters (Stop/Start of Scale service via Ambari GUI).

    Solution:

    When you are performing STOP ALL/START ALL in a FPO environment in Ambari, the IBM Spectrum Scale NSD is not able to be recovered. To prevent the IBM Spectrum Scale file system from stopping in the Ambari STOP ALL, place the IBM Spectrum Scale service in the Maintenance mode first before you execute the Ambari STOP ALL. For more information, see "Stop all without stopping IBM Spectrum Scale service" on page 429.

    For how to properly stop/start IBM Spectrum Scale for FPO, see IBM Spectrum Scale NSD are not able to be recovered in FPO.

    Then you can put the IBM Spectrum Scale service off the Maintenance mode.

11. Ambari 2.7.X adding additional directories during deployment.

    Solution:

    For HDP 3.X using Ambari 2.7.X, Ambari will add in the additional directories besides the default /hadoop/hdfs directory path. Ensure that you review the HDFS NameNode and DataNode directories and Yarn local directories and other directories listed in the "Customize services" on page 362 directories to ensure that only the required directories are listed. For example, when you are integrating/unintegrating Scale service:

    DFS NameNode :

    ```
    /hadoop/hdfs/namenode,/.snapshots/hadoop/hdfs/namenode,/opt/hadoop/hdfs/
    namenode,/srv/hadoop/hdfs/namenode,/usr/local/hadoop/hdfs/namenode,/var/
    cache/hadoop/hdfs/namenode,/var/crash/hadoop/hdfs/namenode,/var/lib/
    libvirt/images/hadoop/hdfs/namenode,/var/lib/machines/hadoop/hdfs/
    namenode,/var/lib/mailman/hadoop/hdfs/namenode,/var/lib/mariadb/hadoop/
    hdfs/namenode,/var/lib/mysql/hadoop/hdfs/namenode,/var/lib/named/hadoop/
    hdfs/namenode,/var/lib/pgsql/hadoop/hdfs/namenode,/var/log/hadoop/
    hdfs/namenode,/var/opt/hadoop/hdfs/namenode,/var/spool/hadoop/hdfs/
    namenode,/var/tmp/hadoop/hdfs/namenode
    ```

    DFS DataNode:

    ```
    /hadoop/hdfs/data,/.snapshots/hadoop/hdfs/data,/opt/hadoop/hdfs/
    data,/srv/hadoop/hdfs/data,/usr/local/hadoop/hdfs/data,/var/cache/
    hadoop/hdfs/data,/var/crash/hadoop/hdfs/data,/var/lib/libvirt/images/
    hadoop/hdfs/data,/var/lib/machines/hadoop/hdfs/data,/var/lib/mailman/
    hadoop/hdfs/data,/var/lib/mariadb/hadoop/hdfs/data,/var/lib/mysql/hadoop/
    hdfs/data,/var/lib/named/hadoop/hdfs/data,/var/lib/pgsql/hadoop/hdfs/
    data,/var/log/hadoop/hdfs/data,/var/opt/hadoop/hdfs/data,/var/spool/hadoop/
    hdfs/data,/var/tmp/hadoop/hdfs/data
    ```

    Even though the HDFS Transparency does not use the NameNode and DataNode listed above, the native HDFS will use them.

    The default directory path is only /hadoop/hdfs/namenode and /hadoop/hdfs/data. All other directories are not needed.

12. Ambari 2.7.x - Cannot find a valid baseurl for repo.

    For Ambari 2.7.x, Ambari writes empty baseurl values to the repo files when using a local repository causing the stack installation failures.

    Solution:

    See AMBARI-25069/SPEC-58/BUG-116328 workaround:

    For Ambari 2.7.0.0: https://docs.hortonworks.com/HDPDocuments/Ambari-2.7.0.0/bk_ambari-release-notes/content/ambari_relnotes-2.7.0.0-known-issues.html

    For Ambari 2.7.1.0: https://docs.hortonworks.com/HDPDocuments/Ambari-2.7.1.0/bk_ambari-release-notes/content/ambari_relnotes-2.7.1.0-known-issues.html

    For Ambari 2.7.3.0: https://docs.hortonworks.com/HDPDocuments/Ambari-2.7.3.0/ambari-release-notes/content/known_issues.html

13. Unable to stop Scale service in Mpack 2.7.0.4.

    In Mpack 2.7.0.4, if **gpfs.storage.type** is set to shared, stopping the Scale service from Ambari will report a failure in the UI even if the operation had succeeded internally.

    Solution:

    To workaround this issue:

    a. Before stopping IBM Spectrum Scale do a STOP All, set the Scale service to maintenance mode.

   b. On the command line, stop IBM Spectrum Scale using the **mmshutdown** command.

      ```
      # /usr/lpp/mmfs/bin/mmshutdown -a
      ```

   c. Put the Scale service out of maintenance mode.

   d. Start the IBM Spectrum Scale service or do Start All via Ambari.

14. During the upgrade process, while trying to perform unintegrate IBM Spectrum Scale service, Ambari failed with `HTTP 500 (Internal Server Error)` error.

    Solution:

    a. Check the Ambari log file, `/var/log/ambari-server/ambari-server.log`, to confirm that the failure was because the write permission for the `/var/run/ambari-server/stack-recommendations/` directory was missing.

    b. Change the owner/permission for the directory to ensure that the Ambari admin user can access the directory.

       i) Set permission to 755 as follows:

       ```
       chmod 755 /var/run/ambari-server/stack-recommendations/
       ```

       ii) If Ambari is running with root privileges, then set the owner and group as follows:

       ```
       chown root:root /var/run/ambari-server/stack-recommendations/
       ```

       iii) If Ambari is running with non root privileges, then set the owner and group as follows:

       ```
       chown AMBARI_USER:AMBARI_USER_GROUP /var/run/ambari-server/stack-recommendations/
       ```

15. Upgrade failures from Mpack 2.7.0.3 or earlier to Mpack 2.7.0.4 - 2.7.0.6.

    When trying to upgrade Mpack 2.7.0.3 and earlier to Mpack 2.7.0.4 and later will result in the following error:

    ```
    2020-04-28 11:21:48,488 WARN [ambari-client-thread-6537] HttpChannel:585 -
    /api/v1/clusters/DSL1/credentials/kdc.admin.credentialcom.google.gson.JsonSyntaxException:
    java.lang.IllegalStateException: Expected a string but was BEGIN_OBJECT at line 1 column 2
    ```

    The issue is from Ambari 2.7.4.0 used for HDP 3.1.4. The REST payload used in older Ambari releases cannot be wrapped with a single quote (') character. For more information, see https://issues.apache.org/jira/browse/AMBARI-9016. Therefore, Mpack 2.7.0.4 **`SpectrumScale_UpgradeIntegrationPackage --preEU`** command cannot run from the `upgrade_Mpack` directory.

    Solution

    a. Execute the **`SpectrumScale_UpgradeIntegrationPackage --preEU`** command from the `currently_installed_Mpack` directory.

    b. Copy the generated files created by the **`SpectrumScale_UpgradeIntegrationPackage --preEU`** command in the `currently_installed_Mpack` directory to the `upgrade_Mpack` directory. These files are later used by the **`SpectrumScale_UpgradeIntegrationPackage --postEU`** command.

       Generated files to be copied:

       ```
       gpfs-master-node.txt
       gpfs-nodes.txt
       gpfs-site.json
       gpfs-filesystem.json
       gpfs-advance.json
       gpfs-ambari-server-env.json
       gpfs-env.json
       ```

16. When you are uninstalling the Mpack, the Mpack uninstaller script might throw a Kerberos credential error even when the correct credentials were provided.

```
$ cd /root/GPFS_Ambari/currently_installed_Mpack
$ ./SpectrumScaleMPackUninstaller.py
***Starting the Spectrum Scale Mpack Uninstaller v2.7.0.3***
...
Enter kdc principal:  root/admin@IBM.COM
Enter kdc password:
INFO: Kerberos is Enabled. Proceeding with Configuration
WARN: {
  "status" : 400,
  "message" : "Invalid Request: Malformed Request Body.
An exception occurred parsing the request body: Unexpected character
(''' (code 39)): expected a valid value (number, String, array, object,
'true', 'false' or 'null')\n at [Source: java.io.StringReader@4f83866d;
line: 1, column: 3]"
}
MISSING_CREDENTIALS
WARN: Kerberos authentication not successful. Enter correct credentials of Kerberos KDC
Server in Spectrum Scale configuration.
INFO: Spectrum Scale service is not added to Ambari.
INFO: Spectrum Scale MPack exists. Removing the MPack.
...
INFO: Ambari server restart completed successfully.
INFO: Spectrum Scale MPack removal completed successfully.
```

Solution:

This error can occur if the Mpack is not compatible with the currently installed version of Ambari as per the Mpack support matrix.

The error messages can be ignored. The Uninstaller script can continue execution and successfully uninstall the existing Mpack.

Service fails to start

1. HDFS does not start after adding IBM Spectrum Scale service or after running an Integrate_Transparency or an Unintegrate_Transparency UI actions in HA mode.

   Solution:

   After Integrate_Transparency or Unintegrate_Transparency in HA mode, if the HDFS service or its components (for example, NameNodes) do not come up during start, then do the following:

   - Check if the zkfc process is up by running the following command on each NameNode host:

     ```
     # ps -eaf | grep zkfc
     ```

     If the zkfc process is up, kill the zkfc process from the NameNode host by running the **`kill -9`** command on the **pid**.
   - Once the zkfc process is not running in any NameNode host, go into the HDFS service dashboard and do a **`Start the HDFS service`**.

2. In non-root Ambari environment, IBM Spectrum Scale fail to start due to NFS mount point not being accessible by root.

   Solution: For example, the `/usrhome/am_agent` is a NFS mount point with permission set to 700. The following error is seen:

   ```
   2017-04-04 15:42:49,901 - ========== Check for changes to the configuration. ===========
   2017-04-04 15:42:49,901 - Updating remote.copy needs service reboot.
   2017-04-04 15:42:49,901 - Values don't match for gpfs.remote.copy.
   running_config[gpfs.remote.copy]:
   sudo wrapper in use; gpfs_config[gpfs.remote.copy]: /usr/bin/scp
   2017-04-04 15:42:49,902 - Updating remote.shell needs service reboot.
   2017-04-04 15:42:49,902 - Values don't match for gpfs.remote.shell.
   running_config[gpfs.remote.shell]:
   /usr/lpp/mmfs/bin/sshwrap; gpfs_config[gpfs.remote.shell]: /usr/bin/ssh
   2017-04-04 15:42:49,902 - Shutdown all gpfs clients.
   2017-04-04 15:42:49,902 - Run command: sudo /usr/lpp/mmfs/bin/mmshutdown -N
   k-001,k-002,k-003,k-004
   2017-04-04 15:44:03,608 - Status: 0, Output:
   ```

```
Tue  4 Apr 15:42:50 CEST 2017: mmshutdown: Starting force unmount of GPFS file systems
k-003.gpfs.net:  mmremote: Invalid current working directory detected: /usrhome/am_agent
```

To resolve this issue: Change the permissions of the home directory of the GPFS non-root user to at least 711.

Example: For the `/usrhome/am_agent` directory, set the directory with at least a 711 permission set or **`rwx--x—x`**.

Where, 7= rwx for the user itself, 1= x for the group, 1= x for others; x will allow users to cd into the home directory.

This is because the IBM Spectrum Scale command does a cd into the home directory of the user. Therefore, the permission should be set to at least 711.

3. Accumulo Tserver failed to start.

   Solution: Accumulo Tserver might go down. Ensure that the block size is set to the IBM Spectrum Scale file system value.

   - In **Accumulo** > **Configs** > **Custom accumulo-site**, set the *tserver.wal.blocksize* to <GPFS File system block size of the data pool>.

     For example, *tserver.wal.blocksize* = 2097152.

     ```
     [root@c902f05x04 ~]# mmlsfs /dev/bigpfs -B
     flag value description
     ------------------- ------------------------ -----------------------------------
     B 262144 Block size (system pool)
     2097152 Block size (other pools)
     [root@c902f05x04 ~]#
     ```

   - Restart Accumulo service.
   - Run Accumulo service check.

     From **Ambari GUI** > **Accumulo** > **Service Actions** > **Run Service Check**.

   For additional failures, see What to do when the Accumulo service start or service check fails?.

4. Hive fails to start, with the default HDP and Ambari recommendations

   Solution:

   There is bug in HDP3.0 - https://hortonworks.jira.com/projects/SPEC/issues/SPEC-18 which causes Accumulo to use the same port number as HiveServer2 leading to port binding conflict. As a workaround, use the following configuration:

   a. Put accumulo and hiveserver2 on different hosts or
   b. Use non-default port for either of them.

5. Kafka service fails to start if Kafka is added after IBM Spectrum Scale.

   Solution:

   Check the Kafka configuration log directory to see if the Kafka log directory contains the IBM Spectrum Scale shared mount point (`/<gpfs mount point>/kafka-logs`). Remove the share mount point from the directory list and restart the Kafka service. For more information, see "Adding Services" on page 360.

6. HBase service fails to start.

   Solution:

   If IBM Spectrum Scale is integrated, Hbase Master fails to start or goes down. This could be because of stale znodes in Zookeeper created for Hbase.

   To clean znodes of Hbase, perform the following:

   a. Log in to zookeeper by executing the following command:

```
/usr/hdp/current/zookeeper-server/bin/zkCli.sh -server <any zookeeper hostname>
```

b. `rmr /hbase-unsecure` or `rmr /hbase-secure` (depending on kerberos enabled/disabled).

7. Start All services fails because **`zkfc`** fails to start. Therefore, putting the NameNodes into standby mode.

   The sequence of steps when this error occurs is to enable NameNode HA in native HDFS then integrate IBM Spectrum Scale service to use HDFS Transparency and later enable Kerberos. During the period when Kerberos is enabling, **`zkfc`** fails to start. This leads to both NameNodes being in the standby mode. Therefore, HDFS cannot be used. As a result, the **Start All services** fails.

   Solution:

   Restart HDFS or restart all services from Ambari.

8. NameNode and DataNodes failed to start with `mapreduce.tar.gz error`.

   For Mpack version 2.7.0.0, the NameNode and DataNode might fail to start with the following error message when the data directory (`gpfs.data.dir`) is specified through the Ambari IBM Spectrum Scale UI: `Failed to replace mapreduce.tar.gz with Transparency jars`.

   Solution:

   Follow these steps to set the `mapreduce.tar.gz` into a proper directory for the NameNode/DataNode to start:

   Check if the `/<gpfs.mnt.dir>/<gpfs.data.dir>/hdp/apps/<hdp-version>/mapreduce/` directory exists. If not, create the directory by running the following command:

   ```
   mkdir -p /<gpfs.mnt.dir>/<gpfs.data.dir>/hdp/apps/<hdp-version>/mapreduce/
   ```

   Copy the `mapreduce.tar.gz` from HDP to the Scale directory by running the following command:

   ```
   cp /usr/hdp/<hdp-version>/hadoop/mapreduce.tar.gz
   /<gpfs.mnt.dir>/<gpfs.data.dir>/hdp/apps/<hdp-version>/mapreduce/mapreduce.tar.gz
   ```

   where, **`<gpfs.mnt.dir>`** is the IBM Spectrum Scale mount point **`<gpfs.data.dir>`** is the IBM Spectrum Scale data directory **`<hdp-version>`** is the HDP version and can be obtained by running hdp-select versions. For example,

   ```
   mkdir -p /bigpfs/datadir1/hdp/apps/2.6.5.0-292/mapreduce/
   cp /usr/hdp/2.6.5.0-292/hadoop/mapreduce.tar.gz
   /bigpfs/datadir1/hdp/apps/2.6.5.0-292/mapreduce/mapreduce.tar.gz
   ```

   Restart the failed HDFS components.

9. The zookeeper failover controller (ZKFC) fails during the **`Start All`** operation after integrating IBM Spectrum Scale service with NameNode High Availability for the first time.

   There is a timing issue during the formatting of the zookeeper directory which is shared by both ZKFC in HA mode on which ZKFC should be started first.

   Solution:

   Rerun the **`Start All`** operation to get the services back up.

10. The zkfc fails to start when Kerberos is enabled.

    The zkfc might fail to start with `Can't set priority for process` error if IBM Spectrum Scale is first added to an HA enabled HDP cluster before adding Kerberos. The `hdfs_jaas.conf` file might not be generated during the kerberos enablement action.

    Solution:

    a. Create the `hdfs_jaas.conf` file in the `/etc/hadoop/conf/secure` directory, on both the NameNodes.

       For example,

```
# cat /etc/hadoop/conf/secure/hdfs_jaas.conf

Client {
      com.sun.security.auth.module.Krb5LoginModule required
      useKeyTab=true
      storeKey=true
      useTicketCache=false
      keyTab="/etc/security/keytabs/nn.service.keytab"
      principal="nn/c902f09x13.gpfs.net@IBM.COM";
};
```

Note: Ensure that you change the keyTab and principal values based on your environment.

b. If `/etc/hosts` is used for hostname resolution instead of DNS in your environment, use the FQDN hostname in `/etc/hosts`.

Ensure that the output from the command **`hostname`** matches the following:

i) Hostname specified in the Ambari wizard.

ii) IP/hostname used for DNS.

Check the same for all the hosts in the cluster and restart HDFS.

11. When the Scale service is unintegrated, the Active NameNode starts whereas the standby NameNode fails to start with `Failed to start namenode.java.io.FileNotFoundException: No valid image files found` error message in the `/var/log/hadoop/hdfs/hadoop-hdfs-namenode-<standby_namenode>.log` file:

```
ERROR namenode.NameNode (NameNode.java:main(1774)) -
Failed to start namenode.java.io.FileNotFoundException:
No valid image files found at
org.apache.hadoop.hdfs.server.namenode.FSImageTransactionalStorageInspector.
getLatestImages(FSImageTransactionalStorageInspector.java:165)
```

This is because the `dfs.namenode.name.dir` (default path: `/hadoop/hdfs/namenode`) directory is empty.

Solution:

Because the Active NameNode is up and running, run the following steps to start the Standby NameNode:

a. Run the following commands only on the Standby NameNode:

```
# su - hdfs
# hdfs namenode -bootstrapStandby
```

Note: Do not run this command on the Active NameNode.

This command tries to recover all the metadata on the Standby NameNode.

b. Restart both the ZKFailover Controllers from Ambari.

c. Restart the Standby NameNode from Ambari.

12. On SLES environment, the NameNode might fail to start due to Out of Memory error with the following error message: `Exiting with status 1: java.lang.OutOfMemoryError: unable to create new native thread.`

Solution:

Increase the `NameNode Heap Size` to at least 2GB in Ambari HDFS configuration and restart the NameNodes.

13. In SLES environment, the Zeppelin Notebook service stop action can be stuck for a long period of time.

Solution:

Stop and start the Zeppelin Notebook service to get out of the hang situation.

14. ZKFC fails to start due to hdfs_jaas.conf file missing when Kerberos is enabled when IBM Spectrum Scale is integrated.

    Error message:

    ```
    2019-05-08 13:34:44,595 WARN zookeeper.ClientCnxn (ClientCnxn.java:startConnect(1014)) -
    SASL configuration failed: javax.security.auth.login.LoginException: Zookeeper client
    cannot
    authenticate using the Client section of the supplied JAAS configuration:
    '/usr/hdp/3.1.0.0-78/hadoop/conf/secure/hdfs_jaas.conf' because of a RuntimeException:
    java.lang.SecurityException: java.io.IOException: /usr/hdp/3.1.0.0-78/hadoop/conf/secure/
    hdfs_jaas.conf
    (No such file or directory) Will continue connection to Zookeeper server without SASL
    authentication,
    if Zookeeper server allows it.
    ```

    Solution:

    a. Copy the `/etc/hadoop/conf/secure/hdfs_jaas.conf` into `/usr/hdp/3.1.0.0-78/hadoop/conf/secure/hdfs_jaas.conf` on all the NameNodes.
    b. Restart ZKFC.

15. When Kerberos is enabled on RH 7.5, the ZKFController fails with the following errors:

    ```
    2019-05-06 06:10:09,974 ERROR client.ZooKeeperSaslClient
    (ZooKeeperSaslClient.java:createSaslToken(388)) -
    An error: (java.security.PrivilegedActionException: javax.security.sasl.SaslException: GSS
    initiate failed
    [Caused by GSSException: No valid credentials provided (Mechanism level: Ticket expired
    (32) - PROCESS_TGS)])
    occurred when evaluating Zookeeper Quorum Member's  received SASL token.
    Zookeeper Client will go to AUTH_FAILED state.

    2019-05-06 06:10:09,974 ERROR zookeeper.ClientCnxn (ClientCnxn.java:run(1059)) - SASL
    authentication with Zookeeper
    Quorum member failed: javax.security.sasl.SaslException: An error:
    (java.security.PrivilegedActionException:
    javax.security.sasl.SaslException: GSS initiate failed [Caused by GSSException: No valid
    credentials provided
    (Mechanism level: Ticket expired (32) - PROCESS_TGS)]) occurred when evaluating Zookeeper
    Quorum Member's
    received SASL token. Zookeeper Client will go to AUTH_FAILED state.

    2019-05-06 06:10:10,081 ERROR ha.ActiveStandbyElector
    (ActiveStandbyElector.java:fatalError(719)) -
    Unexpected Zookeeper watch event state: AuthFailed

    2019-05-06 06:10:10,081 ERROR ha.ZKFailoverController
    (ZKFailoverController.java:fatalError(379)) -
    Fatal error occurred:Unexpected Zookeeper watch event state: AuthFailed

    2019-05-06 06:10:10,081 FATAL tools.DFSZKFailoverController
    (DFSZKFailoverController.java:main(197)) -
    DFSZKFailOverController exiting due to earlier exception java.io.IOException:
    Couldn't determine existence of znode '/hadoop-ha/nn'

    2019-05-06 06:10:10,083 INFO  util.ExitUtil (ExitUtil.java:terminate(210)) - Exiting with
    status 1:
    java.io.IOException: Couldn't determine existence of znode '/hadoop-ha/nn'

    2019-05-06 06:10:10,085 INFO  tools.DFSZKFailoverController (LogAdapter.java:info(49)) -
    SHUTDOWN_MSG:
    /************************************************************
    SHUTDOWN_MSG: Shutting down DFSZKFailoverController at dn01-dat.gpfs.net/30.1.1.15
    ************************************************************/
    ```

    Solution:

    The default KDC version on RHEL7.5 has a known bug. You need to upgrade the krb-server packages to 1.1.15.1-19 + version.

    Steps:

    a. Check the installed version of krb on all the hosts.

```
# yum list installed | grep krb
```

b. Stop Kerberos.

```
# systemctl stop krb5kdc
# systemctl stop kadmin
```

c. Upgrade krb-server, libs and workstation to 1.15.1-19 on the ambari-server and all the ambari-agent nodes.

For example:

```
# rpm -Uvh krb5-workstation-1.15.1-19.el7.ppc64le.rpm krb5-
libs-1.15.1-19.el7.ppc64le.rpm
libkadm5-1.15.1-19.el7.ppc64le.rpm
```

d. Start Kerberos.

```
# systemctl start krb5kdc
# systemctl start kadmin
```

e. Restart Ambari server.

```
# Ambari-server restart
```

f. Restart ZKFController in Ambari.

For additional information, see 2nd generation HDFS Protocol troubleshooting DataNode reports exceptions after Kerberos is enabled on RHEL7.5.

16. Yarn Timeline Service 2.0 fails to start.

In HDP 3.0: The Timeline Service 2.0 in Yarn fails to start.

Solution:

There is a new implementation of Timeline service in HDP 3.0 named Timeline Service 2.0. It can run in 2 modes (Embedded mode or System service mode) depending on the cluster capacity. To check which mode is being set, filter the search for **`is_hbase_system_service_launch`** under YARN configuration. If this value is checked, it is running in the system service mode. If it is running in the system service mode, follow the set of best practices from the Enable System Service Mode.

The following important steps should be performed after Integrating/UnIntegrating the IBM Spectrum Scale service and Enabling/Disabling Kerberos: Remove ats-hbase before switching between clusters.

If you get the ERROR `client.ApiServiceClient: Failed to destroy service ats-hbase, because it is still running` error above, perform the following steps:

a. Check the status of the ats-hbase service by executing the following command:

```
yarn app -status ats-hbase
```

b. If the state is STOPPED, then perform the following steps:

Get the application_ID from the ResourceManager UI in the Ambari GUI and run:

```
yarn -kill -appId <application_ID>
yarn app -destroy ats-hbase
```

Might need to remove the `/<gpfs.mnt.dir value>/<gpfs.data.dir value>/user/yarn-ats/{stack-version}` directory.

For example,

```
rm -rf /gpfs/datadir_1/user/yarn-ats/{stack-version}
```

c. Run all the service checks to ensure that all the services are successful.

Service check failures

1. MapReduce service check fails

   Solution:

   a. If the MapReduce service check failed with `/user/ambari-qa` not found:

      Look for the `ambari-qa` folder in the DFS user directory. If it does not exist, create it. If this step is skipped, MapReduce service check will fail with the `/user/ambari-qa path not found` error.

      As root:

      - **`mkdir <gpfs mount>/user/ambari-qa`**
      - **`chown ambari-qa:hadoop /gpfs/hadoopfs/user/ambari-qa`**

   b. If the MapReduce service check time out or job failed due to permission failure for yarn:

      For Yarn, ensure that the **`yarn yarn.nodemanager.local-dirs`** and **`yarn.nodemanager.log-dirs`** are writable for the user yarn. If this is not the case, add write permission to the **`yarn.nodemanager.local-dirs`** and **`yarn.nodemanager.log-dirs`** directories for the user yarn.

2. What to do when the Accumulo service start or service check fails?

   Solution:

   Note:

   - Ensure that the HDFS and Zookeeper services are running before you proceed.
   - If it is non root environment, run the commands in the workaround steps after logging in with non root user only.
   - If GPFS is unintegrated, remove the **`tserver.wal.blocksize`** entry from Accumulo. From Ambari, go to **Accumulo** > **Configs** > **Custom accumulo-site** and remove the **`tserver.wal.blocksize`** value and save the configuration.
   - If GPFS is integrated, follow the workaround for **`tserver.wal.blocksize`** as mentioned in the FAQ Accumulo Tserver failed to start.

   If the problem still exists, contact IBM service.

3. Atlas service check fails.

   Solution:

   a. Restart the Ambari Infra service.

   b. Restart the Hbase service.

   c. Re-run the Atlas service check.

4. Falcon service check fails.

   Solution:

   This is a known issue with HDP. For information on resolving this issue, go to Falcon Web UI is inaccessible(HTTP 503 error) and Ambari Service Check for Falcon fails: "ERROR: Unable to initialize Falcon Client object".

5. What to do when the Hive service check fails with the following error:

   ```
   Templeton Smoke Test (ddl cmd): Failed. : {"error":"Unable to access program:
   /usr/hdp/${hdp.version}/hive/bin/hcat"}http_code <401>
   ```

   Solution:

   HDP is not able to properly parse the **`${hdp.version}`** value. To set the HDP version, execute the following steps:

   a. Get the HDP version for your environment by running the **`/usr/bin/hdp-select versions`** command on any Ambari node.

b. In the Ambari GUI, click **HIVE** > **Configs**. Find the **`templeton.hcat`** field under the **Advanced webhcat-site**.

c. Replace the **`${hdp.version}`** in the **`templeton.hcat`** field with the hardcoded **`hdp.version`** value found in `step a`.

   For example, if the value of **`hdp.version`** is *2.6.5.0-292*, set the **`templeton.hcat`** value from `/usr/hdp/${hdp.version}/hive/bin/hcat` to `/usr/hdp/2.6.5.0-292/hive/bin/hcat`.

d. Restart the HIVE service components.

e. Re-run the HIVE service check.

Chapter 6. Apache Hadoop

Apache Hadoop 3.0.x Support

Apache Hadoop 3.0.x is supported with HDFS Transparency 3.0.0. When you use Apache Hadoop, the configuration files of HDFS Transparency are located under `/var/mmfs/hadoop/etc/hadoop`. By default, the logs of HDFS Transparency are located under `/var/log/transparency/`.

If you want to run Apache Hadoop with HDFS Transparency 3.0, execute the following steps:

1. Set **`ulimit nofile`** to *64K* on all the nodes.
2. Set up the `ntp server` to synchronize the time on all nodes.
3. Root password-less access from NameNodes to all other DataNodes.

   For more details, see "Password-less ssh access" on page 53.
4. Install HDFS Transparency 3.0.0-x (`gpfs.hdfs-protocol-3.0.0-x.<arch>.rpm`) on all HDFS Transparency nodes.
5. **ssh** to TransparencyNode1.
6. Update the `/var/mmfs/hadoop/etc/hadoop/core-site.xml` with your NameNode hostname.

   ```
   <configuration>
     <property>
       <name>fs.defaultFS</name>
       <value>hdfs://c8f2n04:8020</value>
     </property>
   </configuration>
   ```
7. Update the `/var/mmfs/hadoop/etc/hadoop/hdfs-site.xml` according to your configuration.

Configuration	Default	Recommendation	Comment
`dfs.replication`	1	1 or 2 or 3	Check your file system by **`mmlsfs <fs-name> -r`** and update this configuration according to the value from **`mmlsfs`**.
`dfs.blocksize`	N/A	134217728 or 268435456 or 536870912	Usually, 128MB (134217728) is used and 512MB (536870912) might be used for IBM ESS file system.
`dfs.client.read.shortcircuit`	false	true	See "Short-circuit read (SSR)" on page 427.

   For other configurations, take the default value.
8. Update the `/var/mmfs/hadoop/etc/hadoop/gpfs-site.xml` for the `gpfs.mnt.dir`, `gpfs.data.dir` and `gpfs.storage.type` configurations.
9. Update the `/var/mmfs/hadoop/etc/hadoop/hadoop-env.sh` and change **`export JAVA_HOME=`** into **`export JAVA_HOME=<your-real-JDK8-Home-Dir>`**.
10. Update the `/var/mmfs/hadoop/etc/hadoop/workers` to add DataNodes. One DataNode hostname per line.
11. Synchronize all these changes into other DataNodes by executing the following command:

```
/usr/lpp/mmfs/bin/mmhadoopctl connector syncconf /var/mmfs/hadoop/etc/hadoop
```

12. Start HDFS Transparency by executing **mmhadoopctl**:

    ```
    /usr/lpp/mmfs/bin/mmhadoopctl connector start
    ```

13. Check the service status of HDFS Transparency by executing **mmhadoopctl**:

    ```
    /usr/lpp/mmfs/bin/mmhadoopctl connector getstate
    ```

    Note: If HDFS Transparency is not up on some nodes, login to those nodes and check the logs located under /var/log/transparency. If you do not get any errors, HDFS Transparency should be up by now.

    If you want to configure the Yarn, execute the following steps:

    a. Download Apache Hadoop 3.0.x from Apache Hadoop website.

    b. Unzip the packages to /opt/Hadoop-3.0.x on HadoopNode1.

    c. Log in to HadoopNode1.

    d. Copy the **hadoop-env.sh**, **hdfs-site.xml**, and workers from /var/mmfs/hadoop/etc/hadoop on HDFS Transparency node to HadoopNode1:/opt/hadoop-3.0.x/etc/hadoop/.

    e. Copy /usr/lpp/mmfs/hadoop/template/mapred-site.xml.template and /usr/lpp/mmfs/hadoop/template/yarn-site.xml.template from HDFS Transparency node into HadoopNode1:/opt/hadoop-3.0.x/etc/hadoop as mapred-site.xml and yarn-site.xml.

    f. Update /opt/hadoop-3.0.x/etc/hadoop/mapred-site.xml with the correct path location for **yarn.app.mapreduce.am.env**, **mapreduce.map.env**, and **mapreduce.reduce.env** configurations.

       For example, change the value from HADOOP_MAPRED_HOME=/opt/hadoop-3.0.2 to HADOOP_MAPRED_HOME=/opt/hadoop-3.0.x

       Note: /opt/hadoop-3.0.x is the real location for Hadoop.

    g. Update /opt/hadoop-3.0.x/etc/hadoop/yarn-site.xml. Especially configuring the correct hostname for **yarn.resourcemanager.hostname**.

    h. Synchronize /opt/hadoop-3.0.x from HadoopNode1 to all other Hadoop nodes and keep the same location for all hosts.

    i. On the Resource Manager node, run the following command to start the Yarn service:

       ```
       #cd /opt/hadoop-3.0.x/sbin/
       #export YARN_NODEMANAGER_USER=root
       #export YARN_RESOURCEMANAGER_USER=root
       #./start-yarn.sh
       ```

       Note: By default, the logs for Yarn service will be under /opt/hadoop-3.0.x/logs. If you plan to start Yarn services with other user name, you could change the user root in the above command to your target user name.

    j. Run the following command to submit word count jobs:

       ```
       #/opt/hadoop-3.0.x/bin/hadoop dfs -put /etc/passwd /passwd
       #/opt/hadoop-3.0.x/bin/hadoop
       jar /opt/hadoop-3.0.x/share/hadoop/mapreduce/hadoop-mapreduce-examples-3.0.2.jar
       wordcount /passwd /results
       ```

       The Yarn service works well if the word count job executed successfully.

       For more information, see Chapter 3, “IBM Storage Scale Hadoop performance tuning guide,” on page 253.

Enabling Kerberos with Apache Hadoop and CES HDFS

This section lists the steps to enable Kerberos with Apache Hadoop and CES HDFS Transparency.

Setting up the Kerberos server

This topic lists the steps to set up the Kerberos server.

Before following these steps, see the “Prerequisites” on page 81 topic.

1. Install and configure the Kerberos server.

   ```
   yum install krb5-server krb5-libs krb5-workstation
   ```

2. Create `/etc/krb5.conf` with the following contents:

   ```
   [logging]
   default = FILE:/var/log/krb5libs.log
   kdc = FILE:/var/log/krb5kdc.log
   admin_server = FILE:/var/log/kadmind.log

   [libdefaults]
   default_realm = IBM.COM
   dns_lookup_realm = false
   dns_lookup_kdc = false
   ticket_lifetime = 24h
   renew_lifetime = 7d
   forwardable = true
   default_realm = IBM.COM

   [realms]
   IBM.COM =  {
       kdc = {KDC_HOST_NAME}
       admin_server = {KDC_HOST_NAME}
       }

   [domain_realm]
       .ibm.com = IBM.COM
       ibm.com = IBM.COM
   ```

   Note: The KDC_HOST_NAME, KDC_HOST_NAME and IBM.COM should reflect the correct host and REALM based on your environment.

3. Set up the server.

   ```
       kdb5_util create -s

       systemctl start krb5kdc
       systemctl start kadmin
       chkconfig krb5kdc on
       chkconfig kadmin on
   ```

4. Add the admin principal, and set the password.

   ```
   kadmin.local -q "addprinc root/admin"
   ```

   Check the `kadm5.acl` to ensure that the entry is correct.

   ```
   cat /var/kerberos/krb5kdc/kadm5.acl
   */admin@IBM.COM

   systemctl restart krb5kdc.service

   systemctl restart kadmin.service
   ```

5. Ensure that the password is working by running the following command:

   ```
   kadmin -p root/admin@IBM.COM
   ```

Setting up Kerberos for HDFS Transparency nodes

This topic lists the steps to set up the Kerberos clients on the HDFS Transparency nodes. These instructions work for both Cloudera Private Cloud Base and Apache Hadoop distributions.

Before following these steps, see the "Prerequisites" on page 81 topic.

1. Install the Kerberos clients package on all the HDFS Transparency nodes.

   ```
   yum install -y krb5-libs krb5-workstation
   ```

2. Copy the `/etc/krb5.conf` file to the Kerberos client hosts on the HDFS Transparency nodes.
3. Create a directory for the keytab directory and set the appropriate permissions on each of the HDFS Transparency node.

   ```
   mkdir -p /etc/security/keytabs/
   chown root:root /etc/security/keytabs
   chmod 755 /etc/security/keytabs
   ```

4. Create KDC principals for the components, corresponding to the hosts where they are running, and export the keytab files as follows:

Service	User:Group	Daemons	Principal	Keytab File Name
HDFS	root:root	NameNode	nn/<NN_Host_FQDN>@<REALM-NAME>	nn.service.keytab
		NameNode HTTP	HTTP/<NN_Host_FQDN>@<REALM-NAME>	spnego.service.keytab
		NameNode HTTP	HTTP/<CES_HDFS_Host_FQDN>@<REALM-NAME>	spnego.service.keytab
		DataNode	dn/<DN_Host_FQDN>@<REALM-NAME>	dn.service.keytab

Replace the < NN_Host_FQDN > with the HDFS Transparency NameNode hostname and the <DN_Host_FQDN> with the HDFS Transparency DataNode hostname. Replace the <CES_HDFS_Host_FQDN> with the CES hostname configured for your CES HDFS cluster.

You need to create one principal for each HDFS Transparency DataNode and two principals for each HDFS Transparency NameNode.

Note: If you are using CDP Private Cloud Base, Cloudera Manager creates the principals and keytabs for all the services except the IBM Storage Scale service. Therefore, you can skip the create service principals section below and go directly to step a.

If you are using Apache Hadoop, you need to create service principals for YARN and Mapreduce services as shown in the following table:

Service	User:Group	Daemons	Principal	Keytab File Name
YARN	yarn:hadoop	ResourceManager	rm/ <Resource_Manager_FQDN>@<REALM-NAME>	rm.service.keytab
		NodeManager	nm/ <Node_Manager_FQDN>@<REALM-NAME>	nm.service.keytab
Mapreduce	mapred:hadoop	MapReduce Job History Server	jhs/ <Job_History_Server_FQDN>@<REALM-NAME>	jhs.service.keytab

Replace the <Resource_Manager_FQDN> with the Resource Manager hostname, the <Node_Manager_FQDN> with the Node Manager hostname and the <Job_History_Server_FQDN> with the Job History Server hostname.

a. Create service principals for each service. Refer to the sample table above.

```
kadmin.local -q "addprinc -randkey  -maxrenewlife 7d +allow_renewable {Principal}"
```

For example:

```
kadmin.local -q "addprinc -randkey -maxrenewlife 7d +allow_renewable nn/
nn01.gpfs.net@IBM.COM"
```

b. Create host principals for each Transparency host.

```
kadmin.local -q "addprinc -randkey host/{HOST_NAME}@<Realm Name>"
```

For example:

```
kadmin.local -q "addprinc -randkey host/nn01.gpfs.net@IBM.COM"
```

c. If you are using RHEL 9.1+ for Power LE, update the principals to include the `+requires_preauth` attribute.

For all the host and service principals created under the previous steps 4.a and 4.b, update the principals to include the `+requires_preauth` flag, as shown in the following example:

```
# kadmin.local: modify_principal +requires_preauth nn/nn01.gpfs.net@IBM.COM
Principal nn/nn01.gpfs.net@IBM.COM modified
```

d. For each service on each Transparency host, create a keytab file by exporting its service principal into a keytab file:

```
kadmin.local ktadd -k
/etc/security/keytabs/{SERVICE_NAME}.service.keytab {Principal}
```

For example:

DataNode:

```
kadmin.local ktadd -k /etc/security/keytabs/dn.service.keytab dn/dn01.gpfs.net@IBM.COM
```

NameNode:

```
kadmin.local ktadd -k /etc/security/keytabs/nn.service.keytab nn/nn01.gpfs.net@IBM.COM
```

NameNode HTTP:

The keytab for this service needs an additional step as it contains entries for two principals – one corresponding to the actual NameNode hostname and another for the CES IP hostname.

- First create the keytab file for HTTP service corresponding to the NameNode host.

  ```
  kadmin.local ktadd -k /etc/security/keytabs/spnego.service.keytab HTTP/
  nn01.gpfs.net@IBM.COM
  ```

- Create a temporary keytab file for HTTP service corresponding to the CES HDFS IP hostname.

  ```
  kadmin.local ktadd -norandkey -k /etc/security/keytabs/myceshdfs.service.keytab HTTP/
  myceshdfs.gpfs.net@IBM.COM
  ```

- Merge the above two keytabs with kutil utility to create an updated spnego.service.keytab:

  ```
  #ktutil
  ktutil: rkt /etc/security/keytabs/myceshdfs.service.keytab
  ktutil: wkt /etc/security/keytabs/spnego.service.keytab
  exit
  ```

  Note: **myceshdfs.gpfs.net** is an example of the CES IP hostname configured for your CES HDFS service.

- Repeat the “4.a” on page 493, “4.b” on page 493, and “4.d” on page 493 steps for every required keytab file.

Note:

- The filename for a service is common (for example, **dn.service.keytab**) across hosts but the contents would be different because every keytab would have a different host principal component.
- After a keytab is generated, move the keytab to the appropriate host immediately or move it into a different location to avoid the keytab from getting overwritten.

5. For CES HDFS NameNode HA, an HDFS admin user and its Kerberos user principal and keytab are required to be created and setup for the CES NameNodes. These credentials are used by the CES framework to elect an active NameNode.

   This principal should map to an existing OS user on the NameNode hosts.

   In this example, the OS user is *hdfs*. You will configure this principal/keytab into hadoop-env.sh in step 8.

   a. First create a Hadoop supergroup.

   Set the **dfs.permissions.superusergroup** parameter to *supergroup* by running the following command:

   ```
   /usr/lpp/mmfs/hadoop/sbin/mmhdfs config set hdfs-site.xml -k
   dfs.permissions.superusergroup=supergroup
   ```

   b. Create the *hdfs* user on all the HDFS Transparency nodes that belongs to the *supergroup* Hadoop super group by using the supplied **gpfs_create_hadoop_users_dirs.py** command.

   The command ensures that the custom user/group is created with consistent UID/GID across all the nodes.

   ```
   /usr/lpp/mmfs/hadoop/scripts/gpfs_create_hadoop_users_dirs.py --create-custom-hadoop-
   user-group hdfs:supergroup
   ```

   Note: If you are going to use CDP, you can skip this step. You will create this user as part of the CDP specific configuration workflow.

   c. Create the user principal.

   ```
   # kadmin.local "addprinc -randkey -maxrenewlife 7d +allow_renewable ces-
   <clustername>@IBM.COM"
   ```

```
# kadmin.local "ktadd -k /etc/security/keytabs/ces-<clustername>.headless.keytab ces-
<clustername>@IBM.COM"
```

where, **`<clustername>`** is the name of your CES HDFS cluster. In case there are multiple CES HDFS clusters sharing a common KDC server, having the cluster name as part of the principal helps to create a user principal unique to each CES HDFS cluster.

d. Copy the `/etc/security/keytabs/ces-<clustername>.headless.keytab` file to all the NameNodes and change the owner permission of the file to root:

```
# chown root:root /etc/security/keytabs/ces-<clustername>.headless.keytab
# chmod 400 /etc/security/keytabs/ces-<clustername>.headless.keytab
```

6. Copy the appropriate keytab file to each host. If a host runs more than one component (for example, both NameNode and DataNode), copy the keytabs for both these components.
7. Set the appropriate permissions for the keytab files.

   On the HDFS Transparency NameNode hosts:

```
chown root:root /etc/security/keytabs/nn.service.keytab
chmod 400 /etc/security/keytabs/nn.service.keytab
chown root:root /etc/security/keytabs/spnego.service.keytab
chmod 440 /etc/security/keytabs/spnego.service.keytab
```

   On the HDFS Transparency DataNode hosts:

```
chown root:root /etc/security/keytabs/dn.service.keytab
chmod 400 /etc/security/keytabs/dn.service.keytab
```

   On the Yarn resource manager hosts:

```
chown yarn:hadoop /etc/security/keytabs/rm.service.keytab
chmod 400 /etc/security/keytabs/rm.service.keytab
```

   On the Yarn node manager hosts:

```
chown yarn:hadoop /etc/security/keytabs/nm.service.keytab
chmod 400 /etc/security/keytabs/nm.service.keytab
```

   On Mapreduce job history server hosts:

```
chown mapred:hadoop /etc/security/keytabs/jhs.service.keytab
chmod 400 /etc/security/keytabs/jhs.service.keytab
```

8. Update the HDFS Transparency configuration files and upload the changes.

   - Get the config files

```
mkdir /tmp/hdfsconf
mmhdfs config export /tmp/hdfsconf core-site.xml,hdfs-site.xml,hadoop-env.sh
```

   - Configurations in `core-site.xml` and `hdfs-site.xml` are different for HDFS Transparency 3.1.x and HDFS Transparency 3.2.2-x/3.3.x. The configurations are as follows:

     – For HDFS Transparency 3.1.x use the following configurations in `core-site.xml` and `hdfs-site.xml`:

       `File: core-site.xml`

```
<property>
   <name>hadoop.security.authentication</name>
   <value>kerberos</value>
</property>

<property>
   <name>hadoop.rpc.protection</name>
   <value>authentication</value>
</property>
```

If you are using Cloudera Private Cloud Base cluster, create the following rules:

```
<property>
  <name>hadoop.security.auth_to_local</name>
  <value>
  RULE:[2:$1/$2@$0](nn/.*@.*IBM.COM)s/.*/hdfs/
  RULE:[2:$1/$2@$0](dn/.*@.*IBM.COM)s/.*/hdfs/
  RULE:[1:$1@$0](ces-<clustername>@IBM.COM)s/.*/hdfs/
  RULE:[1:$1@$0](.*@IBM.COM)s/@.*//
  DEFAULT
  </value>
</property>
```

Otherwise, if you are using Apache Hadoop, create the following rules:

```
<property>
  <name>hadoop.security.auth_to_local</name>
  <value>
    RULE:[2:$1/$2@$0](nn/.*@.*IBM.COM)s/.*/hdfs/
    RULE:[2:$1/$2@$0](dn/.*@.*IBM.COM)s/.*/hdfs/
    RULE:[2:$1/$2@$0](nm/.*@.*IBM.COM)s/.*/yarn/
    RULE:[2:$1/$2@$0](rm/.*@.*IBM.COM)s/.*/yarn/
    RULE:[2:$1/$2@$0](jhs/.*@.*IBM.COM)s/.*/mapred/
    RULE:[1:$1@$0](ces-<clustername>@IBM.COM)s/.*/hdfs/
    DEFAULT
  </value>
</property>
```

In the above example, replace IBM.COM with your Realm name and *<clustername>* parameter with your actual CES HDFS cluster name.

`File: hdfs-site.xml`

```
<property>
  <name>dfs.data.transfer.protection</name>
  <value>authentication</value>
</property>

<property>
  <name>dfs.datanode.address</name>
  <value>0.0.0.0:1004</value>
</property>

<property>
  <name>dfs.datanode.data.dir.perm</name>
  <value>700</value>
</property>

<property>
  <name>dfs.datanode.http.address</name>
  <value>0.0.0.0:1006</value>
</property>

<property>
  <name>dfs.datanode.kerberos.principal</name>
  <value>dn/_HOST@IBM.COM</value>
</property>

<property>
  <name>dfs.datanode.keytab.file</name>
  <value>/etc/security/keytabs/dn.service.keytab</value>
</property>

<property>
  <name>dfs.encrypt.data.transfer</name>
  <value>false</value>
</property>

<property>
  <name>dfs.namenode.kerberos.internal.spnego.principal</name>
  <value>HTTP/_HOST@IBM.COM</value>
</property>

<property>
  <name>dfs.namenode.kerberos.principal</name>
  <value>nn/_HOST@IBM.COM</value>
</property>
```

```
<property>
  <name>dfs.namenode.keytab.file</name>
  <value>/etc/security/keytabs/nn.service.keytab</value>
</property>

<property>
  <name>dfs.web.authentication.kerberos.keytab</name>
  <value>/etc/security/keytabs/spnego.service.keytab</value>
</property>

<property>
  <name>dfs.web.authentication.kerberos.principal</name>
  <value>*</value>
</property>
```

- For HDFS Transparency 3.2.2-x and 3.3.x use the following configurations in `core-site.xml` and `hdfs-site.xml`:

  `File: core-site.xml`

```
<property>
   <name>hadoop.security.authentication</name>
   <value>kerberos</value>
</property>

<property>
   <name>hadoop.rpc.protection</name>
   <value>authentication</value>
</property>

<property>
   <name>hadoop.http.authentication.type</name>
   <value>kerberos</value>
</property>

<property>
   <name>hadoop.http.authentication.kerberos.principal</name>
   <value>*</value>
</property>

<property>
   <name>hadoop.http.authentication.kerberos.keytab</name>
   <value>/etc/security/keytabs/spnego.service.keytab</value>
</property>
```

  If you are using Cloudera Private Cloud Base cluster, create the following rules:

```
<property>
  <name>hadoop.security.auth_to_local</name>
  <value>
   RULE:[2:$1/$2@$0](nn/.*@.*IBM.COM)s/.*/hdfs/
   RULE:[2:$1/$2@$0](dn/.*@.*IBM.COM)s/.*/hdfs/
   RULE:[1:$1@$0](ces-<clustername>@IBM.COM)s/.*/hdfs/
   RULE:[1:$1@$0](.*@IBM.COM)s/@.*//
   DEFAULT
   </value>
</property>
```

  Otherwise, if you are using Apache Hadoop, create the following rules:

```
<property>
  <name>hadoop.security.auth_to_local</name>
  <value>
    RULE:[2:$1/$2@$0](nn/.*@.*IBM.COM)s/.*/hdfs/
    RULE:[2:$1/$2@$0](dn/.*@.*IBM.COM)s/.*/hdfs/
    RULE:[2:$1/$2@$0](nm/.*@.*IBM.COM)s/.*/yarn/
    RULE:[2:$1/$2@$0](rm/.*@.*IBM.COM)s/.*/yarn/
    RULE:[2:$1/$2@$0](jhs/.*@.*IBM.COM)s/.*/mapred/
    RULE:[1:$1@$0](ces-<clustername>@IBM.COM)s/.*/hdfs/
    DEFAULT
  </value>
</property>
```

  In the above example, replace IBM.COM with your Realm name and *<clustername>* parameter with your actual CES HDFS cluster name.

`File: hdfs-site.xml`

```
<property>
  <name>dfs.data.transfer.protection</name>
  <value>authentication</value>
</property>

<property>
  <name>dfs.datanode.address</name>
  <value>0.0.0.0:1004</value>
</property>

<property>
  <name>dfs.datanode.data.dir.perm</name>
  <value>700</value>
</property>

<property>
  <name>dfs.datanode.http.address</name>
  <value>0.0.0.0:1006</value>
</property>

<property>
  <name>dfs.datanode.kerberos.principal</name>
  <value>dn/_HOST@IBM.COM</value>
</property>

<property>
  <name>dfs.datanode.keytab.file</name>
  <value>/etc/security/keytabs/dn.service.keytab</value>
</property>

<property>
  <name>dfs.encrypt.data.transfer</name>
  <value>false</value>
</property>

<property>
  <name>dfs.namenode.kerberos.internal.spnego.principal</name>
  <value>HTTP/_HOST@IBM.COM</value>
</property>

<property>
  <name>dfs.namenode.kerberos.principal</name>
  <value>nn/_HOST@IBM.COM</value>
</property>

<property>
  <name>dfs.namenode.keytab.file</name>
  <value>/etc/security/keytabs/nn.service.keytab</value>
</property>

<property>
  <name>dfs.block.access.token.enable</name>
  <value>true</value>
</property>
```

- `File: hadoop-env.sh`

  ```
  KINIT_KEYTAB=/etc/security/keytabs/ces-<clustername>.headless.keytab
  KINIT_PRINCIPAL=ces-<clustername>@IBM.COM
  ```

  where, *<clustername>* is the name of your CES HDFS cluster.

9. Stop the HDFS Transparency services for the cluster.

   a. Stop the DataNodes.

      On any HDFS Transparency node, run the following command:

      ```
      mmhdfs hdfs-dn stop
      ```

   b. Stop the NameNodes.

      On any CES HDFS NameNode, run the following command:

      ```
      mmces service stop HDFS -N <NN1>,<NN2>
      ```

10. Import the files.

    ```
    mmhdfs config import /tmp/hdfsconf core-site.xml,hdfs-site.xml,hadoop-env.sh
    ```

11. Upload the changes.

    ```
    mmhdfs config upload
    ```

12. Start the HDFS Transparency services for the cluster.

    a. Start the DataNodes.

       On any HDFS Transparency node, run the following command:

       ```
       mmhdfs hdfs-dn start
       ```

    b. Start the NameNodes.

       On any CES HDFS NameNode, run the following command:

       ```
       mmces service start HDFS -N <NN1>,<NN2>
       ```

    c. Verify that the services have started.

       On any CES HDFS NameNode, run the following command:

       ```
       mmhdfs hdfs status
       ```

Configuring YARN and MapReduce

This topic lists the steps to configure YARN and MapReduce with Kerberos.

For Apache Hadoop, Yarn and MapReduce needs to be installed on the clients. For information, see the "HDFS clients configuration" on page 500 and "MapReduce/YARN clients configuration" on page 501.

1. Update `yarn-site.xml`.

   ```
       <property>
         <name>yarn.resourcemanager.principal</name>
         <value>rm/_HOST@IBM.COM</value>
       </property>
       <property>
         <name>yarn.resourcemanager.keytab</name>
         <value>/etc/security/keytab/rm.service.keytab</value>
       </property>

       <property>
         <name>yarn.nodemanager.principal</name>
         <value>nm/_HOST@IBM.COM</value>
       </property>
       <property>
         <name>yarn.nodemanager.keytab</name>
         <value>/etc/security/keytab/nm.service.keytab</value>
       </property>
   ```

2. Update `mapreduce-site.xml`.

   ```
       <property>
         <name>mapreduce.jobhistory.keytab</name>
         <value>/etc/security/keytab/jhs.service.keytab</value>
       </property>
       <property>
         <name>mapreduce.jobhistory.principal</name>
         <value>jhs/_HOST@IBM.COM</value>
       </property>
   ```

3. Synchronize `/opt/hadoop-3.x.x` to all the other Hadoop nodes and keep the same location for all the hosts.

   Use **scp** to copy the configuration files from HADOOP_HOME to the other Hadoop nodes with the services installed.

4. On the Resource Manager node, run the following command to start the Yarn service:

   ```
   cd /opt/hadoop-3.0.x/sbin/
   export YARN_NODEMANAGER_USER=root
   export YARN_RESOURCEMANAGER_USER=root
   ./start-yarn.sh
   ```

5. Run the following command to submit the word count jobs:

   ```
   /opt/hadoop-3.0.x/bin/hadoop dfs -put /etc/passwd /passwd
   /opt/hadoop-3.0.x/bin/hadoop jar
   /opt/hadoop-3.0.x/share/hadoop/mapreduce/hadoop-mapreduce-examples-3.0.2.jar
   wordcount /passwd /results
   ```

   Note: The successful execution of the word count job indicates that the Yarn and MapReduce services are working properly.

HDFS clients configuration

HDFS clients must be configured in the following way to work with the CES IP failover mechanism.

The cluster name is the CES group name without the hdfs prefix.

The value of **`fs.defaultFS`** and **`dfs.nameservices`** should be configured as the cluster name (In this example, cluster). The cluster name for the HDFS client should be the same as the NameNodes or DataNodes.

For CES HDFS, there is only one NameNode in the HDFS client configuration. The hostname of the CES IP configured for CES group should be used as the NameNode value (In this example, `cesip.example.com`). This is same for HA and non-HA configuration.

For example, the Apache Hadoop is installed in `/usr/hadoop-3.1.3`, so the Hadoop configuration files are all located at `/usr/hadoop-3.1.3/etc/hadoop`.

For core-site.xml:

The values should be the same as the HDFS transparency configuration file on the NameNode.

```
<property>
  <name>fs.defaultFS</name>
  <value>hdfs://cluster</value>
</property>
```

For hdfs-site.xml:

Replace the HDFS transparency NameNode with the host name of the corresponding CES IP value.

```
<property>
  <name>dfs.nameservices</name>
  <value>cluster</value>
</property>
<property>
  <name>dfs.ha.namenodes.cluster</name>
  <value>nn1</value>
</property>
<property>
  <name>dfs.namenode.rpc-address.cluster.nn1</name>
  <value>cesip.example.com:8020</value>
</property>
<property>
  <name>dfs.namenode.http-address.cluster.nn1</name>
  <value>cesip.example.com:50070</value>
</property>
<property>
  <name>dfs.client.failover.proxy.provider.cluster</name>
  <value>org.apache.hadoop.hdfs.server.namenode.ha.ConfiguredFailoverProxyProvider</value>
</property>
<property>
   <name>dfs.ha.fencing.methods</name>
  <value>shell(/bin/true)</value>
</property>
```

Note: The NameNode configuration will contain properties for both the NameNodes while the HDFS clients will only define one NameNode property that contains the CES IP hostname. The HDFS clients in the CES HDFS environment will only know about one NameNode and will communicate with CES HDFS Transparency through this IP. High availability is achieved through failing over the IP to another NameNode. This is handled by CES and transparent for HDFS clients because they always talk to the same IP.

For hadoop_env.sh:

For Apache Hadoop, configure the properties with the values based on your host environment.

Then set the JAVA_HOME value in `haoop-env.sh` file.

```
export JAVA_HOME=/usr/lib/jvm/java-1.8.0-openjdk
```

MapReduce/YARN clients configuration

MapReduce and Yarn clients must be configured to launch MapReduce workload on Yarn so that it can read/write data from/into the IBM Storage Scale cluster.

MapReduce/YARN client configuration files are located in the same directory as the HDFS client.

For mapred-site.xml:

Add the following properties to the corresponding value according to your host environment:

```
<configuration>
  <property>
    <name>mapreduce.framework.name</name>
    <value>yarn</value>
  </property>
  <property>
    <name>yarn.app.mapreduce.am.env</name>
    <value>HADOOP_MAPRED_HOME=/usr/hadoop-3.1.3</value>
    <description>Change this to your hadoop location.</description>
  </property>
  <property>
    <name>mapreduce.map.env</name>
    <value>HADOOP_MAPRED_HOME=/usr/hadoop-3.1.3</value>
    <description>Change this to your hadoop location.</description>
 </property>
  <property>
    <name>mapreduce.reduce.env</name>
    <value>HADOOP_MAPRED_HOME=/usr/hadoop-3.1.3</value>
    <description>Change this to your hadoop location.</description>
  </property>
  <property>
    <name>mapreduce.map.memory.mb</name>
    <value>4096</value>
    <description>Change this according to your cluster configuration.</description>
  </property>
 <property>
   <name>mapreduce.reduce.memory.mb</name>
    <value>8192</value>
    <description>Change this according to your cluster configuration.</description>
  </property>
```

Note: If the Mapreduce job failed with return code 1, see Mapreduce container job exit with return code 1.

For yarn-site.xml:

Add the following properties to the corresponding value according to your host environment.

For example, `c16f1n11.gpfs.net` is the Resource Manager.

```
<configuration>
  <property>
    <name>yarn.resourcemanager.hostname</name>
    <value>c16f1n11.gpfs.net</value>
    <description>Configure resourcemanager hostname.</description>
  </property>
  <property>
    <name>yarn.nodemanager.aux-services</name>
    <value>mapreduce_shuffle</value>
  </property>

  <property>
    <name>yarn.nodemanager.resource.memory-mb</name>
    <value>24576</value>
  </property>
```

For workers, add the hostname which will act as the Node Manager.

For this example, c16f1n10.gpfs.net and c16f1n12.gpfs.net are the Node Managers.

```
cat workers
c16f1n10.gpfs.net
c16f1n12.gpfs.net
```

Start the Resource Manager and Node Manager to launch a MapReduce workload.

```
cd /usr/hadoop-3.1.2/sbin/
export YARN_NODEMANAGER_USER=root
export YARN_RESOURCEMANAGER_USER=root
./start-yarn.sh
/usr/hadoop-3.1.3/bin/yarn jar /usr/hadoop-3.1.3/share/hadoop/mapreduce/hadoop-mapreduce-
examples-3.1.3.jar teragen 1000 /gen
```

Add HDFS client to CES HDFS nodes

HDFS Transparency does not require you to have the Hadoop distribution installed onto the IBM Storage Scale HDFS Transparency nodes. However, if the HDFS client is not installed on the CES HDFS NameNodes and DataNodes, then functions like `distcp` will not work because HDFS Transparency does not include the **bin/hadoop** command.

To execute the **hadoop** command on the HDFS Transparency nodes, the HDFS client needs to be installed and configured on the HDFS Transparency nodes.

The setup and configuration is similar to the "HDFS clients configuration" on page 500. But all the configurations will stay on the Hadoop distribution path and the HDFS Transparency configurations under `/var/mmfs/hadoop/etc/hadoop` path will not be changed.

Steps to install and configure on the HDFS Transparency nodes:

1. Download the Apache Hadoop and extract the packages onto each node.
2. On one of the CES HDFS node, modify the downloaded Hadoop distribution path `HADOOP_HOME/etc/hadoop` configurations files based on the settings seen in the "HDFS clients configuration" on page 500.
3. Manually sync (scp) the `HADOOP_HOME/etc/hadoop` configurations files to all the other CES HDFS nodes.
4. Execute the **hadoop** command from the `HADOOP_HOME/etc/hadoop/bin` path.

   For example:

   `<HADOOP_HOME>/hadoop-3.1.3/bin/hadoop dfs -ls /`

   or

   `<HADOOP_HOME>/hadoop-3.1.3/bin/hadoop distcp hdfs://nn1:8020/fileA`

   `hdfs://nn2:8020/fileB`

Accessibility features for IBM Storage Scale

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 Storage Scale:

- 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 US. This material might be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

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, MD-NC119 Armonk, NY 10504-1785 US

For license inquiries regarding double-byte character set (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

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 jurisdictions 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 websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you provide 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 Director of Licensing IBM Corporation North Castle Drive, MD-NC119 Armonk, NY 10504-1785 US

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

The performance data discussed herein is presented as derived under specific operating conditions. Actual results may vary.

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.

Statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

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 actual people or business enterprises 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.

Each copy or any portion of these sample programs or any derivative work must include a copyright notice as follows:

© (your company name) (year).
Portions of this code are derived from IBM Corp.
Sample Programs. © Copyright IBM Corp. _enter the year or years_.

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

Red Hat, OpenShift®, 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 IBM Storage Scale.

The following cross-references are used in this glossary:

- *See* refers you from a nonpreferred 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 (www.ibm.com/software/globalization/terminology) (opens in new window).

B

block utilization
The measurement of the percentage of used subblocks per allocated blocks.

C

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

cluster configuration data
The configuration data that is stored on the cluster configuration servers.

Cluster Export Services (CES) nodes
A subset of nodes configured within a cluster to provide a solution for exporting GPFS file systems by using the Network File System (NFS), Server Message Block (SMB), and Object protocols.

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

Note: The cluster manager role is not moved to another node when a node with a lower node number becomes active.

clustered watch folder
Provides a scalable and fault-tolerant method for file system activity within an IBM Storage Scale file system. A clustered watch folder can watch file system activity on a fileset, inode space, or an entire file system. Events are streamed to an external Kafka sink cluster in an easy-to-parse JSON format. For more information, see the *mmwatch command* in the *IBM Storage Scale: Command and Programming Reference Guide*.

control data structures
Data structures needed to manage file data and metadata cached in memory. Control data structures include hash tables and link pointers for finding cached data; lock states and tokens to implement distributed locking; and various flags and sequence numbers to keep track of updates to the cached data.

D

Data Management Application Program Interface (DMAPI)
The interface defined by the Open Group's XDSM standard as described in the publication *System Management: Data Storage Management (XDSM) API Common Application Environment (CAE) Specification C429*, The Open Group ISBN 1-85912-190-X.

deadman switch timer

A kernel timer that works on a node that has lost its disk lease and has outstanding I/O requests. This timer ensures that the node cannot complete the outstanding I/O requests (which would risk causing file system corruption), by causing a panic in the kernel.

dependent fileset

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

disk descriptor

A definition of the type of data that the disk contains and the failure group to which this disk belongs. See also *failure group*.

disk leasing

A method for controlling access to storage devices from multiple host systems. Any host that wants to access a storage device configured to use disk leasing registers for a lease; in the event of a perceived failure, a host system can deny access, preventing I/O operations with the storage device until the preempted system has reregistered.

disposition

The session to which a data management event is delivered. An individual disposition is set for each type of event from each file system.

domain

A logical grouping of resources in a network for the purpose of common management and administration.

E

ECKD

See *extended count key data (ECKD)*.

ECKD device

See *extended count key data device (ECKD device)*.

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*, *master encryption key*.

extended count key data (ECKD)

An extension of the count-key-data (CKD) architecture. It includes additional commands that can be used to improve performance.

extended count key data device (ECKD device)

A disk storage device that has a data transfer rate faster than some processors can utilize and that is connected to the processor through use of a speed matching buffer. A specialized channel program is needed to communicate with such a device. See also *fixed-block architecture disk device*.

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 connections, and could all become unavailable through a single hardware failure.

FEK

See *file encryption key*.

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.

file audit logging
Provides the ability to monitor user activity of IBM Storage Scale file systems and store events related to the user activity in a security-enhanced fileset. Events are stored in an easy-to-parse JSON format. For more information, see the *mmaudit command* in the *IBM Storage Scale: Command and Programming Reference Guide*.

file clone
A writable snapshot of an individual file.

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

file-management policy
A set of rules defined in a policy file that GPFS uses to manage file migration and file deletion. See also *policy*.

file-placement policy
A set of rules defined in a policy file that GPFS uses to manage the initial placement of a newly created file. See also *policy*.

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.

fixed-block architecture disk device (FBA disk device)
A disk device that stores data in blocks of fixed size. These blocks are addressed by block number relative to the beginning of the file. See also *extended count key data device*.

fragment
The space allocated for an amount of data too small to require a full block. A fragment consists of one or more subblocks.

G

GPUDirect Storage
IBM Storage Scale's support for NVIDIA's GPUDirect Storage (GDS) enables a direct path between GPU memory and storage. File system storage is directly connected to the GPU buffers to reduce latency and load on CPU. Data is read directly from an NSD server's pagepool and it is sent to the GPU buffer of the IBM Storage Scale clients by using RDMA.

global snapshot
A snapshot of an entire GPFS file system.

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 recovery log
A file that contains a record of metadata activity and exists for each node of a cluster. In the event of a node failure, the recovery log for the failed node is replayed, restoring the file system to a consistent state and allowing other nodes to continue working.

I

ill-placed file
A file assigned to one storage pool but having some or all of its data in a different storage pool.

ill-replicated file
A file with contents that are not correctly replicated according to the desired setting for that file. This situation occurs in the interval between a change in the file's replication settings or suspending one of its disks, and the restripe of the file.

independent fileset
A fileset that has its own inode space.

indirect block
A block containing 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.

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

J

journaled file system (JFS)
A technology designed for high-throughput server environments, which are important for running intranet and other high-performance e-business file servers.

junction
A special directory entry that connects a name in a directory of one fileset to the root directory of another fileset.

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.

M

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

MEK
See *master encryption key*.

metadata
Data structures that contain information that is needed to access file data. Metadata includes inodes, indirect blocks, and directories. Metadata is not accessible to user applications.

metanode
The one node per open file that is responsible for maintaining file metadata integrity. In most cases, the node that has had the file open for the longest period of continuous time is the metanode.

mirroring
The process of writing the same data to multiple disks at the same time. The mirroring of data protects it against data loss within the database or within the recovery log.

Microsoft Management Console (MMC)

A Windows tool that can be used to do basic configuration tasks on an SMB server. These tasks include administrative tasks such as listing or closing the connected users and open files, and creating and manipulating SMB shares.

multi-tailed
A disk connected to multiple nodes.

N

namespace
Space reserved by a file system to contain the names of its objects.

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 hex 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 may contain one or more nodes.

node descriptor
A definition that indicates how GPFS uses a node. Possible functions include: manager node, client node, quorum node, and nonquorum node.

node number
A number that is generated and maintained by GPFS 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 GPFS 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.

Non-Volatile Memory Express (NVMe)
An interface specification that allows host software to communicate with non-volatile memory storage media.

P

policy
A list of file-placement, service-class, and encryption rules that define characteristics and placement of files. Several policies can be defined within the configuration, but only one policy set is active at one time.

policy rule
A programming statement within a policy that defines a specific action to be performed.

pool
A group of resources with similar characteristics and attributes.

portability
The ability of a programming language to compile successfully on different operating systems without requiring changes to the source code.

primary GPFS cluster configuration server
In a GPFS cluster, the node chosen to maintain the GPFS cluster configuration data.

private IP address
An IP address used to communicate on a private network.

public IP address
An IP address used to communicate on a public network.

Q

quorum node
A node in the cluster that is counted to determine whether a quorum exists.

quota
The amount of disk space and number of inodes assigned as upper limits for a specified user, group of users, or fileset.

quota management
The allocation of disk blocks to the other nodes writing to the file system, and comparison of the allocated space to quota limits at regular intervals.

R

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.

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

replication
The process of maintaining a defined set of data in more than one location. Replication consists of copying designated changes for one location (a source) to another (a target) and synchronizing the data in both locations.

RKM server
See *remote key management server*.

rule
A list of conditions and actions that are triggered when certain conditions are met. Conditions include attributes about an object (file name, type or extension, dates, owner, and groups), the requesting client, and the container name associated with the object.

S

SAN-attached
Disks that are physically attached to all nodes in the cluster using Serial Storage Architecture (SSA) connections or using Fibre Channel switches.

Scale Out Backup and Restore (SOBAR)
A specialized mechanism for data protection against disaster only for GPFS file systems that are managed by IBM Storage Protect for Space Management.

secondary GPFS cluster configuration server
In a GPFS cluster, the node chosen to maintain the GPFS cluster configuration data in the event that the primary GPFS cluster configuration server fails or becomes unavailable.

Secure Hash Algorithm digest (SHA digest)
A character string used to identify a GPFS security key.

session failure
The loss of all resources of a data management session due to the failure of the daemon on the session node.

session node
The node on which a data management session was created.

Small Computer System Interface (SCSI)
An ANSI-standard electronic interface that allows personal computers to communicate with peripheral hardware, such as disk drives, tape drives, CD-ROM drives, printers, and scanners faster and more flexibly than previous interfaces.

snapshot
An exact copy of changed data in the active files and directories of a file system or fileset at a single point in time. See also *fileset snapshot*, *global snapshot*.

source node
The node on which a data management event is generated.

stand-alone client
The node in a one-node cluster.

storage area network (SAN)
A dedicated storage network tailored to a specific environment, combining servers, storage products, networking products, software, and services.

storage pool
A grouping of storage space consisting of volumes, logical unit numbers (LUNs), or addresses that share a common set of administrative characteristics.

stripe group
The set of disks comprising the storage assigned to a file system.

striping
A storage process in which information is split into blocks (a fixed amount of data) and the blocks are written to (or read from) a series of disks in parallel.

subblock
The smallest unit of data accessible in an I/O operation, equal to one thirty-second of a data block.

system storage pool
A storage pool containing file system control structures, reserved files, directories, symbolic links, special devices, as well as the metadata associated with regular files, including indirect blocks and extended attributes. The `system storage pool` can also contain user data.

T

token management
A system for controlling file access in which each application performing a read or write operation is granted some form of access to a specific block of file data. Token management provides data consistency and controls conflicts. Token management has two components: the token management server, and the token management function.

token management function
A component of token management that requests tokens from the token management server. The token management function is located on each cluster node.

token management server
A component of token management that controls tokens relating to the operation of the file system. The token management server is located at the file system manager node.

transparent cloud tiering (TCT)
A separately installable add-on feature of IBM Storage Scale that provides a native cloud storage tier. It allows data center administrators to free up on-premise storage capacity, by moving out cooler data to the cloud storage, thereby reducing capital and operational expenditures.

twin-tailed
A disk connected to two nodes.

U

user storage pool
A storage pool containing the blocks of data that make up user files.

V

VFS
See *virtual file system*.

virtual file system (VFS)
A remote file system that has been mounted so that it is accessible to the local user.

virtual node (vnode)
The structure that contains information about a file system object in a virtual file system (VFS).

W

watch folder API
Provides a programming interface where a custom C program can be written that incorporates the ability to monitor inode spaces, filesets, or directories for specific user activity-related events within IBM Storage Scale file systems. For more information, a sample program is provided in the following directory on IBM Storage Scale nodes: `/usr/lpp/mmfs/samples/util` called tswf that can be modified according to the user's needs.

Index

Part Number:
Product Number: 5641-DM1
5641-DM3
5641-DM5
5641-DA1
5641-DA3
5641-DA5
5737-F34
5737-I39
5765-DME
5765-DAE

(1P) P/N:

SC27-9284-14