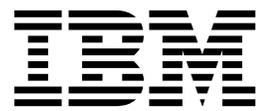


IBM Spectrum Scale
Version 4.2.3

Big Data and Analytics Guide



IBM Spectrum Scale
Version 4.2.3

Big Data and Analytics Guide



Note

Before using this information and the product it supports, read the information in “Notices” on page 223.

This edition applies to version 4 release 2 modification 3 of the following products, and to all subsequent releases and modifications until otherwise indicated in new editions:

- IBM Spectrum Scale ordered through Passport Advantage (product number 5725-Q01)
- IBM Spectrum Scale ordered through AAS/eConfig (product number 5641-GPF)
- IBM Spectrum Scale for Linux on Z (product number 5725-S28)
- IBM Spectrum Scale for IBM ESS (product number 5765-ESS)

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 xix. 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.

© Copyright IBM Corporation 2017, 2018.

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Tables	v
-------------------------	----------

About this information	vii
---	------------

Prerequisite and related information	xviii
Conventions used in this information	xviii
How to send your comments	xix

Summary of changes	xxi
-------------------------------------	------------

Chapter 1. IBM Spectrum Scale support for Hadoop 1

HDFS transparency	1
Supported IBM Spectrum Scale storage modes	2
Local Storage Mode	2
Shared Storage Mode	2
IBM ESS storage	5
Hadoop cluster planning	7
License planning	7
Node roles planning	10
Hardware and software requirements.	12
Hadoop service roles	13
Dual network interfaces	13
Installation and configuration of HDFS transparency 14	
Installation of HDFS transparency	14
Configuration of HDFS transparency	15
Start and stop the service manually	21
Connector health check	22
Cluster and file system information configuration 22	
Application interaction with HDFS transparency	23
Application interface of HDFS transparency	23
Command line for HDFS transparency	23
Upgrading the HDFS Transparency cluster	24
Removing IBM Spectrum Scale Hadoop connector	24
Manually upgrading the IBM Spectrum Scale HDFS Transparency connector	26
Rolling upgrade for HDFS Transparency.	26
Upgrading HDFS Transparency NameNode	27
Security	27
Advanced features	28
High availability configuration	28
Short-circuit read configuration.	33
Short circuit write	36
Multiple Hadoop clusters over the same file system	37
Docker support	38
Hadoop Storage Tiering	43
Hadoop distcp support	79
Automatic Configuration Refresh	81
Ranger support	81
Rack locality support for shared storage.	92
Accumulo support	94
Multiple Spectrum Scale File System support	96
Zero shuffle support	97
Hadoop distribution support	98

Replacing native HDFS service with HDFS transparency	98
HortonWorks HDP 2.6.x support	99
IBM BigInsights IOP support	99
Limitations and differences from native HDFS.	99
Snapshot support	99
Hadoop ACL and Spectrum Scale Protocol NFS/SMB	101
The difference between HDFS Transparency and native HDFS	101
Problem determination	102

Chapter 2. BigInsights 4.2.5 and Hortonworks Data Platform 2.6 103

Planning	103
Hardware requirements	103
Preparing the environment	103
Preparing a stanza file	105
Installation	107
Set up	107
Installation of software stack	114
BigInsights value-add services on IBM Spectrum Scale	135
Upgrading software stack	136
Migrating from BI IOP to HDP	136
Upgrading IBM Spectrum Scale service MPack 140	
Upgrading HDFS Transparency	142
Upgrading IBM Spectrum Scale file system	144
Upgrading to BI IOP 4.2.5	145
Configuration	149
Setting up High Availability [HA]	149
IBM Spectrum Scale configuration parameter checklist	149
Dual-network deployment	150
Manually starting services in Ambari	150
Setting up local repository	151
Configuring LogSearch	156
Setting IBM Spectrum Scale configuration for BigSQL	157
Administration	157
IBM Spectrum Scale-FPO deployment	157
Ranger	161
Kerberos	166
Short-circuit read (SSR)	171
Disabling short circuit write	172
IBM Spectrum Scale service management	173
Ambari node management	180
Restricting root access	188
IBM Spectrum Scale management GUI	192
IBM Spectrum Scale versus Native HDFS	193
Troubleshooting	196
Snap data collection	196
Limitations	197
Limitations and information	197
FAQ	201

General	201
Service fails to start	214
Service check failures.	217

Accessibility features for IBM

Spectrum Scale	221
Accessibility features	221
Keyboard navigation	221
IBM and accessibility.	221

Notices	223
Trademarks	224
Terms and conditions for product documentation	225
IBM Online Privacy Statement.	225

Glossary	227
---------------------------	------------

Index	233
------------------------	------------

Tables

1.	IBM Spectrum Scale library information units	viii	7.	Preferred Simple Format and Standard Format.	105
2.	Conventions	xix	8.	IBM Spectrum Scale partitioning function matrix	160
3.	List of changes in documentation	xxix	9.	NATIVE HDFS AND IBM SPECTRUM SCALE DIFFERENCES	195
4.	IBM Spectrum Scale License requirement	8			
5.	Ranger directory permission issues.	91			
6.	Hadoop distribution support matrix	104			

About this information

This edition applies to IBM Spectrum Scale™ version 4.2.3 for AIX®, Linux, and Windows.

IBM Spectrum 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 Spectrum Scale is running on a particular AIX node, enter:

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

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

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

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

To use these information units effectively, you must be familiar with IBM Spectrum 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 Spectrum Scale library information units

Information unit	Type of information	Intended users
<p><i>IBM Spectrum Scale: Concepts, Planning, and Installation Guide</i></p>	<p>This guide provides the following information:</p> <p>Product overview</p> <ul style="list-style-type: none"> • Overview of IBM Spectrum Scale • GPFS architecture • Protocols support overview: Integration of protocol access methods with GPFS • Active File Management • AFM-based Asynchronous Disaster Recovery (AFM DR) • Data protection and disaster recovery in IBM Spectrum Scale • Introduction to IBM Spectrum Scale GUI • IBM Spectrum Scale management API • Introduction to Cloud services • IBM Spectrum Scale in an OpenStack cloud deployment • IBM Spectrum Scale product editions • IBM Spectrum Scale license designation • Capacity based licensing • IBM Spectrum Storage™ Suite <p>Planning</p> <ul style="list-style-type: none"> • Planning for GPFS • Planning for protocols • Considerations for GPFS applications • Firewall recommendations • Planning for cloud services 	<p>System administrators, analysts, installers, planners, and programmers of IBM Spectrum Scale clusters who are very experienced with the operating systems on which each IBM Spectrum Scale cluster is based</p>

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

Information unit	Type of information	Intended users
<p><i>IBM Spectrum Scale: Concepts, Planning, and Installation Guide</i></p>	<p>Installing and upgrading</p> <ul style="list-style-type: none"> • Steps for establishing and starting your IBM Spectrum Scale cluster • Installing IBM Spectrum Scale on Linux nodes and deploying protocols • Installing IBM Spectrum Scale on AIX nodes • Installing IBM Spectrum Scale on Windows nodes • Installing cloud services on IBM Spectrum Scale nodes • Installing and configuring IBM Spectrum Scale management API • Installing Active File Management • Installing and upgrading AFM-based Disaster Recovery • Installing call home • Migration, coexistence and compatibility • Steps to permanently uninstall GPFS and/or Protocols 	<p>System administrators, analysts, installers, planners, and programmers of IBM Spectrum Scale clusters who are very experienced with the operating systems on which each IBM Spectrum Scale cluster is based</p>

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

Information unit	Type of information	Intended users
<p><i>IBM Spectrum Scale: Administration Guide</i></p>	<p>This guide provides the following information:</p> <p>Configuring</p> <ul style="list-style-type: none"> • Configuring the GPFS cluster • Configuring the CES and protocol configuration • Configuring and tuning your system for GPFS • Parameters for performance tuning and optimization • Configuring and tuning your system for Cloud services • Configuring Active File Management • Configuring AFM-based DR • Tuning for Kernel NFS backend on AFM and AFM DR <p>Administering</p> <ul style="list-style-type: none"> • Performing GPFS administration tasks • Verifying network operation with the mmnetverify command • Managing file systems • File system format changes between versions of IBM Spectrum Scale • Managing disks • 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 • Considerations for GPFS applications • Accessing a remote GPFS file system 	<p>System administrators or programmers of IBM Spectrum Scale systems</p>

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

Information unit	Type of information	Intended users
<p><i>IBM Spectrum Scale: Administration Guide</i></p>	<ul style="list-style-type: none"> • Information lifecycle management for IBM Spectrum 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 • 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 • Highly-available write cache (HAWC) • Local read-only cache • Miscellaneous advanced administration • GUI limitations 	<p>System administrators or programmers of IBM Spectrum Scale systems</p>

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

Information unit	Type of information	Intended users
<p><i>IBM Spectrum Scale: Problem Determination Guide</i></p>	<p>This guide provides the following information:</p> <p>Monitoring</p> <ul style="list-style-type: none"> • Performance monitoring • Monitoring system health through the IBM Spectrum Scale GUI • Monitoring system health by using the mmhealth command • Monitoring events through callbacks • Monitoring capacity through GUI • Monitoring AFM and AFM DR • GPFS SNMP support • Monitoring the IBM Spectrum Scale system by using call home • Monitoring the health of cloud services <p>Troubleshooting</p> <ul style="list-style-type: none"> • Best practices for troubleshooting • Understanding the system limitations • Collecting details of the issues • Managing deadlocks • Installation and configuration issues • Upgrade issues • Network issues • File system issues • Disk issues • Security issues • Protocol issues • Disaster recovery issues • Performance issues • GUI issues • AFM issues • AFM DR issues • Transparent cloud tiering issues • Recovery procedures • Support for troubleshooting • References 	<p>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 <i>IBM Spectrum Scale: Concepts, Planning, and Installation Guide</i></p>

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

Information unit	Type of information	Intended users
<p><i>IBM Spectrum Scale: Command and Programming Reference</i></p>	<p>This guide provides the following information:</p> <p>Command reference</p> <ul style="list-style-type: none"> • gpfs.snap command • mmaddcallback command • mmadddisk command • mmaddnode command • mmadquery command • mmafmconfig command • mmafmctl command • mmafmlocal command • mmapplypolicy command • mmauth command • mmbackup command • mmbackupconfig command • mmblock command • mmbuildgpl command • mmcallhome command • mmces command • mmcesdr 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 • mmclone command • mmcloudgateway command • mmcrcluster command • mmcrfileset command • mmcrfs command • mmcrnodeclass command • mmcrnsd command • mmcrsnapshot command 	<ul style="list-style-type: none"> • System administrators of IBM Spectrum Scale systems • Application programmers who are experienced with IBM Spectrum Scale systems and familiar with the terminology and concepts in the XDSM standard

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

Information unit	Type of information	Intended users
<p><i>IBM Spectrum Scale: Command and Programming Reference</i></p>	<ul style="list-style-type: none"> • mmdefquota 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 • mmfsctl command • mmgetacl command • mmgetstate command • mmhadoopctl command • mmhealth command • mmimgbackup command • mmimgrestore command • mmimportfs command • mmkeyserv command • mmlinkfileset command • mmlsattr command • mmlscallback command • mmlscluster command • mmlsconfig command • mmlsdisk command 	<ul style="list-style-type: none"> • System administrators of IBM Spectrum Scale systems • Application programmers who are experienced with IBM Spectrum Scale systems and familiar with the terminology and concepts in the XDSM standard

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

Information unit	Type of information	Intended users
<p><i>IBM Spectrum Scale: Command and Programming Reference</i></p>	<ul style="list-style-type: none"> • 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 • mmppmon command • mmprotocoltrace command • mmprsnap command • mmputacl command • mmquotaoff command • mmquotaon command • mmremotefluster command • mmremotefs command • mmrepquota command • mmrestoreconfig command • mmstorefs command • mmrestripefile command • mmrestripefs command • mmrpldisk command • mmsdrrestore command • mmsetquota command • mmshutdown command • mmsmb command • mmsnapdir command • mmstartup command • mmtracectl command • mmumount command • mmunlinkfileset command • mmuserauth command • mmwinservctl command • spectrumscale command 	<ul style="list-style-type: none"> • System administrators of IBM Spectrum Scale systems • Application programmers who are experienced with IBM Spectrum Scale systems and familiar with the terminology and concepts in the XDSM standard

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

Information unit	Type of information	Intended users
<i>IBM Spectrum Scale: Command and Programming Reference</i>	Programming reference <ul style="list-style-type: none"> • IBM Spectrum Scale Data Management API for GPFS information • GPFS programming interfaces • GPFS user exits • IBM Spectrum Scale management API commands 	<ul style="list-style-type: none"> • System administrators of IBM Spectrum Scale systems • Application programmers who are experienced with IBM Spectrum Scale systems and familiar with the terminology and concepts in the XDSM standard

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

Information unit	Type of information	Intended users
<p><i>IBM Spectrum Scale: Big Data and Analytics Guide</i></p>	<p>This guide provides the following information:</p> <p>IBM Spectrum Scale support for Hadoop</p> <ul style="list-style-type: none"> • HDFS transparency • Supported IBM Spectrum Scale storage modes • Hadoop cluster planning • Installation and configuration of HDFS transparency • Application interaction with HDFS transparency • Upgrading the HDFS Transparency cluster • Rolling upgrade for HDFS Transparency • Security • Advanced features • Hadoop distribution support • Limitations and differences from native HDFS • Problem determination <p>BigInsights® 4.2.5 and Hortonworks Data Platform 2.6</p> <ul style="list-style-type: none"> • Planning <ul style="list-style-type: none"> – Hardware requirements – Preparing the environment – Preparing a stanza file • Installation <ul style="list-style-type: none"> – Set up – Installation of software stack – BigInsights value-add services on IBM Spectrum Scale • Upgrading software stack <ul style="list-style-type: none"> – Migrating from BI IOP to HDP – Upgrading IBM Spectrum Scale service MPack – Upgrading HDFS Transparency – Upgrading IBM Spectrum Scale file system – Upgrading to BI IOP 4.2.5 	<ul style="list-style-type: none"> • System administrators of IBM Spectrum Scale systems • Application programmers who are experienced with IBM Spectrum Scale systems and familiar with the terminology and concepts in the XDSM standard

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

Information unit	Type of information	Intended users
<p><i>IBM Spectrum Scale: Big Data and Analytics Guide</i></p>	<ul style="list-style-type: none"> • Configuration <ul style="list-style-type: none"> – Setting up High Availability [HA] – IBM Spectrum Scale configuration parameter checklist – Dual-network deployment – Manually starting services in Ambari – Setting up local repository – Configuring LogSearch – Setting IBM Spectrum Scale configuration for BigSQL • Administration <ul style="list-style-type: none"> – IBM Spectrum Scale-FPO deployment – Ranger – Kerberos – Short-circuit read (SSR) – Disabling short circuit write – IBM Spectrum Scale service management – Ambari node management – Restricting root access – IBM Spectrum Scale management GUI – IBM Spectrum Scale versus Native HDFS • Troubleshooting <ul style="list-style-type: none"> – Snap data collection • Limitations <ul style="list-style-type: none"> – Limitations and information • FAQ <ul style="list-style-type: none"> – General – Service fails to start – Service check failures 	<ul style="list-style-type: none"> • System administrators of IBM Spectrum Scale systems • Application programmers who are experienced with IBM Spectrum Scale systems and familiar with the terminology and concepts in the XDSM standard

Prerequisite and related information

For updates to this information, see IBM Spectrum Scale in IBM Knowledge Center(www.ibm.com/support/knowledgecenter/STXKQY/ibmspectrumscale_welcome.html).

For the latest support information, see the IBM Spectrum Scale FAQ in IBM Knowledge Center(www.ibm.com/support/knowledgecenter/STXKQY/gpfsclustersfaq.html).

Conventions used in this information

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

Note: Users of IBM Spectrum 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.
<u>bold underlined</u>	<u>bold underlined</u> keywords are defaults. These take effect if you do not specify a different keyword.
constant width	Examples and information that the system displays appear in constant-width typeface. Depending on the context, constant-width typeface sometimes represents path names, directories, or file names.
<i>italic</i>	<i>Italic</i> words or characters represent variable values that you must supply. <i>Italics</i> 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 <i>Enter</i> .
\	In command examples, a backslash indicates that the command or coding example continues on the next line. For example: <pre>mkcondition -r IBM.FileSystem -e "PercentTotUsed > 90" \ -E "PercentTotUsed < 85" -m p "FileSystem space used"</pre>
{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 <i>synopsis</i> statements, vertical lines separate a list of choices. In other words, a vertical line means <i>Or</i> . 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 Spectrum 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 Spectrum Scale development organization, send your comments to the following e-mail address:

gpfs@us.ibm.com

Summary of changes

This topic summarizes changes to the IBM Spectrum Scale licensed program and the IBM Spectrum Scale library. Within each information unit in the library, a vertical line (|) to the left of text and illustrations indicates technical changes or additions that are made to the previous edition of the information.

| **Summary of changes**
| **for IBM Spectrum Scale version 4 release 2.3**
| **as updated, June 2018**

| This release of the IBM Spectrum Scale licensed program and the IBM Spectrum Scale library includes the following improvements:

| **Added support for OpenStack Mitaka packages**

| Support for OpenStack Mitaka packages has been added for the object protocol. For more information, see *Protocols support overview: Integration of protocol access methods with GPF* in *IBM Spectrum Scale: Concepts, Planning, and Installation Guide*.

| **Authentication considerations changes**

| The following changes are done:

- | • Authentication support matrix has been divided to separate out the File and object protocols and accordingly, the corresponding explanation is modified.
- | • The matrix is further divided based on the authentication service that is used.
- | • A diagram is added to explain the high-level flow of authentication for File protocols.
- | • "Authentication for file access" topic is renamed to "Authentication and ID mapping for file access".

| For more information, see the *Authentication considerations* topic in the *IBM Spectrum Scale: Concepts, Planning, and Installation Guide*.

| **Big data and analytics changes**

| Big data and analytics has the following updates:

| **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 HDFS Transparency 2.7.3-2**

- | • Snapshot from a remote mounted file system is supported.
- | • IBM Spectrum Scale fileset-based snapshot is supported.
- | • HDFS Transparency and IBM Spectrum 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 2.4.2.6**

- | • 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 Spectrum Scale service in the Ambari GUI can capture the Ambari agents' logs into a tar package under the `/var/log/ambari.gpfs.snap*` directory.
- Use cases where the Ambari server and the GPFS Master are co-located on the same host but are configured with multiple IP addresses are handled within the IBM Spectrum Scale service installation.
- On starting IBM Spectrum Scale from Ambari, if a new kernel version is detected on the IBM Spectrum Scale node, the GPFS portability layer is automatically rebuilt on that node.
- On deploying the IBM Spectrum Scale service, the Ambari server restart is not required. However, the Ambari server restart is still required when executing the **Service Action > Integrate Transparency** or **Unintegrate Transparency** from the Ambari UI.

Changes in Mpack version 2.4.2.5

- HDP 2.6.5 is supported.

Changes in Mpack version 2.4.2.4

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

Changes in Mpack version 2.4.2.3

- HDP 2.6.3 is supported.

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 Spectrum 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 Spectrum 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*

- 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.syncBufsPerIteration** 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 Spectrum Scale has been increased to 16 MB from 4 MB.

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 Spectrum 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 Spectrum 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 Spectrum Scale file systems/deployment models under one Hadoop cluster/Ambari management. (For HDP only)

This allows you to have a combination of Spectrum 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 data nodes are running on the same node.

Directory preallocation

In environments in which many files are added to and removed from a directory in a short time, you can improve performance by setting the minimum compaction size of the directory. The minimum compaction size is the number of directory slots, including both full and empty slots, that a directory is allowed to retain when it is compacted. For more information, see *gpfs_prealloc()* subroutine, *mmchattr* command, and *mmilsattr* command in *IBM Spectrum Scale: Command and Programming Reference*.

Express Edition no longer available

IBM Spectrum Scale Express Edition is no longer available. For information on migrating from IBM Spectrum Scale Express Edition 4.2.2.x or earlier to IBM Spectrum Scale Standard Edition 4.2.3.x, see *Migrating from Express Edition to Standard Edition* in *IBM Spectrum Scale: Concepts, Planning, and Installation Guide*.

FPO enhancements

FPO performs the following functions:

- Provides QoS support for auto recovery
- Supports locality-aware data copy
- Uses the **mmrestripefile** command to check whether the replicas of data blocks are matched for one file

Installation toolkit support for gpfs.adv and gpfs.crypto packages

The installation toolkit now supports installation, deployment, and upgrade of `gpfs.adv` and `gpfs.crypto` packages.

Installation toolkit support for populating cluster definition file

The installation toolkit now supports populating the cluster definition file with the current cluster state. For more information, see *Populating cluster definition file with current cluster state using the installation toolkit* in *IBM Spectrum Scale: Concepts, Planning, and Installation Guide*.

Installation toolkit support for Red Hat Enterprise Linux 7.4 and 7.5

The installation toolkit now also supports Red Hat Enterprise Linux 7.4 and 7.5 on x86_64, PPC64, and PPC64LE architectures. For more information, see *Installation prerequisites* in *IBM Spectrum Scale: Concepts, Planning, and Installation Guide*.

IBM Spectrum Scale GUI changes

The following main changes are made in the IBM Spectrum Scale GUI:

- Supports mounting and unmounting of file systems on selected nodes or group of nodes using GUI. For more information, see *Mounting a file system through GUI* and *Unmounting a file system through GUI* topics in *IBM Spectrum Scale: Administration Guide*.
- Added new **Storage > Pools** page. The Pools page provides details about configuration, health, capacity, and performance aspects of storage pools.
- Added new **Files > Active File Management** page. This new GUI page helps to view the configuration, health status, and performance of AFM, AFM DR, and gateway nodes.
- Added new **Monitoring > Tips** page. The tip events give recommendations to the user to avoid certain issues that might occur in the future. A tip disappears from the GUI when the problem behind the tip event is resolved.

- Added option to select events of type “tip” in the **Settings > Event Notifications > Email Recipients** page. You can configure whether to send email to the recipients if a tip event is reported in the system.
- Added detailed view in the **Files > Filesets** page. You can access the detailed view of individual filesets either by double-clicking the individual filesets or by selecting **View Details** option.
- Modified the **Storage > NSDs** page to list the rack, position, and node of the NSD in an FPO-enabled environment. This helps to sort the NSDs based on these parameters. The failure group definition is also modified to accommodate these new parameters.
- Added the **Customize the number of replicas** option in the **Files > Information Lifecycle** page to specify the number of replicas in a file placement rule.
- Modified the **Settings > Event Notifications** page to accept both IP address and host name for the email server.
- Added **Nodes** and **File Systems** tabs in the detailed view that is available in the **Files > Transparent Cloud Tiering** page.
- Added a separate **Properties** tab in the detailed view that is available in the **Monitoring > Nodes , Files > File Systems ,** and **Storage > NSDs** pages.

IBM Spectrum Scale functionality to support GDPR requirements

To understand the requirements of EU General Data Protection Regulation (GDPR) compliance that are applicable to unstructured data storage and how IBM Spectrum Scale helps to address them, see the IBM Spectrum Scale functionality to support GDPR requirements technote.

Introduction of IBM Spectrum Scale management API Version 2

The architecture and syntax of IBM Spectrum Scale management API is changed. The new implementation is based on the GUI stack. The GUI server is managing and processing the API requests and commands. Version 2 has the following features:

- Reuses the GUI deployment's backend infrastructure, which makes introduction of new API commands easier.
- No separate configuration is required as the GUI installation takes care of the basic deployment.
- Fixes scalability issues and introduces new features such as filter parameter, field parameter, and paging.
- Supports large clusters with thousands of nodes.
- All POST, PUT, and DELETE requests are completed asynchronously. A "jobs" object is created immediately when such a request is submitted.
- The APIs are driven by the same WebSphere® server and object cache that is used by the IBM Spectrum Scale GUI.
- The **mmrest** command is no longer required for configuring the management API. The IBM Spectrum Scale GUI installation and configuration takes care of the API infrastructure configuration. For more information on how to configure IBM Spectrum Scale management API Version 2, see *Configuring IBM Spectrum Scale management API* in IBM Spectrum Scale:Administration Guide.

As the syntax and architecture are changed for the API, modified the entire set of commands, which were available in the Version 1. New API commands are also added for improved flexibility. For more information about the available commands, see *IBM Spectrum Scale management API commands* in *IBM Spectrum Scale: Command and Programming Reference*. You can also access the documentation corresponding to each API command from the GUI itself. The API documentation is available in the GUI at: `https://<IP address or host name of API server>:<port>/ibm/api/explorer/`. For example: `https://scalegui.ibm.com:443/ibm/api/explorer`.

Linux on Z enhancements

The following changes are made:

- IBM Spectrum Scale for Linux on Z now supports Remote Cluster Mount (Multi-cluster).
- SLES 12.2 and RHEL 7.3 are now supported by IBM Spectrum Scale for Linux on Z.

mmcallhome command: Addition of --long option to mmcallhome group list command

The **--long** option displays the long admin node names. For more information, see *mmcallhome command* in *IBM Spectrum Scale: Command and Programming Reference*.

mmchconfig command: Setting an InfiniBand partition key

The **--verbsRdmaPkey** attribute specifies an InfiniBand partition key for a connection between a node and an InfiniBand server that is included in an InfiniBand partition. For more information, see *mmchconfig command* in the *IBM Spectrum Scale: Command and Programming Reference*.

mmdiag command: Status and queue statistics for NSD queues

The **--nsd** parameter displays the status and queue statistics for NSD queues.

For more information, see *mmdiag command* in *IBM Spectrum Scale: Command and Programming Reference*.

mmfsck command: Severity of errors

The command displays a summary of the errors that were found that includes the severity of each error: **CRITICAL**, **NONCRITICAL**, or **HARMLESS**. You must specify the verbose or semi-verbose parameter to get this output. For more information, see *mmfsck command* in *IBM Spectrum Scale: Command and Programming Reference*.

mmhealth command: Addition of new options to command

Addition of **AFM** and **THRESHOLD** options to the **mmhealth node show** and **mmhealth cluster show** commands. The **AFM** option displays the health status of a gateway node or cluster. The **THRESHOLD** option monitors whether the node-related thresholds rules evaluation is running as expected, and if the health status has changed as a result of the threshold limits being crossed.

Addition of **--clear** option to the **mmhealth node eventlog** command. This option clears the event log's database.

Addition of **threshold add** and **threshold delete** option to the **mmhealth** command. This option allows users to create and delete threshold rule.

Addition of **event hide**, **event unhide**, and **list hidden** options to the **mmhealth** command. The **event hide** option hides the specified TIP events, while the **event unhide** option reveals all TIP events that were previously hidden. The **list hidden** option shows all the TIP events that are added to the list of hidden events.

Addition of **config interval** option to the **mmhealth** command. The **config interval** option allows you to set the interval for monitoring the whole cluster.

For more information, see *mmhealth command* in *IBM Spectrum Scale: Command and Programming Reference*.

mmkeyserv command: Updating a certificate or a connection

You can now get a fresh certificate from an Remote Key Management (RKM) server without rebuilding the connection. You can also temporarily update a connection by adding backup servers, reordering the list of backup servers, or changing the timeout, number of retries, or retry interval. For more information, see *mmkeyserv command* in *IBM Spectrum Scale: Command and Programming Reference*.

mmlicense command: Displaying disk and cluster size information

You can now get information about disk and cluster size with the **mmlicense** command. For more information, see *mmlicense command* in *IBM Spectrum Scale: Command and Programming Reference*.

mmnetverify command: Enhancements

Several enhancements increase the capabilities of the **mmnetverify** command. Network checks are added to measure the total bandwidth, to check connectivity with the CTDB port, and to check connectivity with servers that are used with the Object protocol. If there are multiple local nodes,

the command is run on all the local nodes in parallel. The lists of local nodes and target nodes accept node classes. The **--ces-override** parameter causes the command to consider all the nodes in the configuration to be CES-enabled. For more information, see *mmnetverify command* in *IBM Spectrum Scale: Command and Programming Reference*.

mmrestripefile command: Fix inconsistencies between file data and replicas

The **-c** option compares the data of individual files with their replicas and attempts to fix any inconsistencies. For more information, see *mmrestripefile command* in *IBM Spectrum Scale: Command and Programming Reference*.

Monitoring of AFM and AFM DR

Using commands:

- Functionality added to **mmhealth**, **mmdiag**, and **mmperfmon**.

Using IBM Spectrum Scale GUI:

- Added new **Files > Active File Management** page. This new GUI page helps to view the configuration, health status, and performance of AFM, AFM DR, and gateway nodes.

Mount options specific to IBM Spectrum Scale: syncnfs is now the default on Linux nodes

In the mount options specific to IBM Spectrum Scale, **syncnfs** is now the default on Linux nodes. On AIX nodes, **nosyncnfs** is the default. For more information, see *Mount options specific to IBM Spectrum Scale* in *IBM Spectrum Scale: Command and Programming Reference*.

Protocol support on remotely mounted file systems

You can create an NFS/SMB export on a file system that is mounted from a remote cluster. For more information, see the *Using NFS/SMB protocol over remote cluster mounts* topic in the *IBM Spectrum Scale: Administration Guide*.

Tip added to event status to inform users when a configuration is not optimal

A new event type **TIP** is added to system health monitoring. A **Tip** is similar to a state-changing event, but can be hidden by the user. Like state-changing events, a tip is removed automatically if the problem is resolved. For more information on **Tip**, see *Event type and monitoring status for system health* in the *IBM Spectrum Scale: Problem Determination Guide*.

Quality of Service for I/O operations (QoS): Detailed statistics

You can now display more detailed statistics about IOPS rates for the QoS programs that are running on each node. The statistics are intended to be used as input for programs that analyze and display data. For more information, see *mmchqos command* and *mmlsqos command* in *IBM Spectrum Scale: Command and Programming Reference*.

Support for Samba 4.5

Transparent cloud tiering enhancements.

The following changes are done:

- Support for configuring and deploying WORM solutions. Your files will remain WORM-compliant, both in the file system and on the cloud object storage. For more information, see the *Deploying WORM solutions* topic in the *IBM Spectrum Scale: Administration Guide*.
- Support for configuring Transparent cloud tiering with a proxy server.
- Support for configuring cloud retention time, which overrides the default value.
- Support for restoring only the file stubs from the cloud storage tier in situations where files are deleted from the local file system.
- Support for Power8 Little Endian platform.

Note: This feature is available from 4.2.3.1 onwards.

- Substantial improvement in the performance when files are transparently recalled from the storage tier.

- Support for manually deleting orphaned cloud objects before retention time expires. For more information, see the *Manually deleting cloud objects before retention time* topic in the *IBM Spectrum Scale: Administration Guide*.
- Support for migrating files in the co-resident state, by which applications can directly access data without performing any recall operation. For more information, see the *Pre-migrating files to the cloud storage tier* topic in the *IBM Spectrum Scale: Administration Guide*

-Y option

Added the -Y option to the following commands:

• mmblock	• mmhealth	• mmlsfileset	• mmlsnodeclass	• mmnetverify
• mmcloudgateway	• mmkeyserv	• mmlsfs	• mmlsnsd	• mmnfs
• mmdf	• mmlscluster	• mmlslicense	• mmlspolicy	• mmrepquota
• mmdiag	• mmlsconfig	• mmlsmgr	• mmlsquota	• mm smb
• mmgetstate	• mmlsdisk	• mmlsmount	• mmlssnapshot	• mmuserauth

Documented commands, structures, and subroutines

The following lists the modifications to the documented commands, structures, and subroutines:

New commands

The following commands are new:

- **mmcli decode**

New structures

There are no new structures.

New subroutines

There are no new subroutines.

Changed commands

The following commands were changed:

- **mmadquery**
- **mmbackup**
- **mmblock**
- **mmcallhome**
- **mmces**
- **mmcesdr**
- **mmchattr**
- **mmchconfig**
- **mmchqos**
- **mmcloudgateway**
- **mmcrnsd**
- **mmdf**
- **mmdiag**
- **mmfsck**
- **mmgetstate**
- **mmhadoopctl**
- **mmhealth**
- **mmimgbackup**
- **mmimgrestore**
- **mmkeyserv**

- **mmisattr**
- **mmiscluster**
- **mmisconfig**
- **mmisdisk**
- **mmisfileset**
- **mmisfs**
- **mmislicense**
- **mmismgr**
- **mmismount**
- **mmisnodeclass**
- **mmisnsd**
- **mmispolicy**
- **mmisqos**
- **mmisquota**
- **mmisnapshot**
- **mmnetverify**
- **mmnfs**
- **mmprotocoltrace**
- **mmrepquota**
- **mm smb**
- **mmuserauth**
- **spectrumscale**

Changed structures

The following structures were changed:

- **gpfs_iattr64_t**

Changed subroutines

The following subroutines were changed:

- **gpfs_prealloc**

Deleted commands

mmrest

Deleted structures

There are no deleted structures.

Deleted subroutines

There are no deleted subroutines.

Messages

The following are the new, changed, and deleted messages:

New messages

6027-1525, 6027-1756, 6027-2392, 6027-2393, 6027-2503, 6027-2504, and 6027-3258

Changed messages

6027-1023, 6027-1725

Deleted messages

None.

Changes in documentation

Big data and analytics support

Moved the entire big data and analytics support information to a new section. See the topic *Big data and analytics support* in *IBM Spectrum Scale: Big Data and Analytics Guide*.

Restructured events page

The events page was split up into 19 different pages, with a separate page for each component. See the topic *.Events* in the *IBM Spectrum Scale: Problem Determination Guide*.

Renamed “REST API” to “IBM Spectrum Scale management API” in the documentation.

List of documentation changes in product guides and respective Knowledge Center sections

The following is a list of documentation changes including changes in topic titles, changes in placement of topics, and deleted topics:

Table 3. List of changes in documentation

Guide	Knowledge center section	List of changes
Concepts, Planning, and Installation Guide	Product overview	Under <i>IBM Spectrum Scale management API</i> <ul style="list-style-type: none"> Moved the IBM Spectrum Scale management API topics from the <i>Administering</i> section to the <i>IBM Spectrum Scale management API</i> section.
	Planning	Under <i>Planning for protocols</i> → <i>Authentication considerations</i> <ul style="list-style-type: none"> Changed the title <i>Authentication for file access</i> to <i>Authentication and ID mapping for file access</i> <p>Under <i>Planning for protocols</i> → <i>Planning for SMB</i></p> <ul style="list-style-type: none"> The <i>SMB share limitations</i> topic under <i>Administering</i> → <i>Managing protocol data exports</i> → <i>Managing SMB shares</i> has been removed. Limitations from the <i>SMB share limitations</i> topic have been added in the <i>SMB limitations</i> topic.
	Installing and upgrading	<ul style="list-style-type: none"> Removed <i>Installing the Scale Management server (REST API)</i> section. Moved <i>Manually upgrading pmswift</i> and <i>Manually upgrading the Performance Monitoring tool</i> from <i>Manually installing the Performance Monitoring tool</i> to the <i>Migration, coexistence and compatibility</i> section. Moved <i>Upgrading IBM Spectrum Scale components with the installation toolkit</i> from <i>Using the spectrumscale installation toolkit to perform installation tasks: Explanations and examples</i> to the <i>Migration, coexistence and compatibility</i> section. Created the <i>Upgrading Object packages</i> section: <ul style="list-style-type: none"> Moved the <i>Upgrading Object packages to version 4.2.2.x</i> from <i>4.2.2.x</i> topic from <i>Migrating to IBM Spectrum Scale 4.2.2.x</i> from <i>IBM Spectrum Scale 4.2.0.x</i> or later to the <i>Upgrading Object packages</i> section. Added the <i>Upgrading Object packages to version 4.2.3.x</i> from <i>4.2.2.x</i> topic.

Table 3. List of changes in documentation (continued)

Guide	Knowledge center section	List of changes
Administration Guide	Configuring	<ul style="list-style-type: none"> Removed the <i>Configuring and starting the Scale Management server (REST API)</i> section. Removed the <i>Enabling Cloud services performance monitoring metrics on the GUI</i> topic under the <i>Configuring</i> → <i>Configuring and tuning your system for Cloud services</i> section.
	Administering	<p>Under <i>File Placement Optimizer</i></p> <ul style="list-style-type: none"> Added <i>Data locality based copy</i> in <i>IBM Spectrum Scale: Administration Guide</i> section. Added <i>mmgetlocation</i> in <i>IBM Spectrum Scale: Administration Guide</i> <i>Data locality restore</i> section is renamed to <i>Data locality</i> in <i>IBM Spectrum Scale: Administration Guide</i>. Added the maintenance steps for IBM Spectrum Scale FPO. Added the performance tuning steps for IBM Spectrum Scale Sharing Nothing Cluster.
Problem Determination Guide	Monitoring	<p>Under <i>Monitoring AFM and AFM DR</i>, renamed the following topics:</p> <ul style="list-style-type: none"> <i>Fileset states for AFM</i> to <i>Monitoring fileset states for AFM</i> <i>Fileset states for AFM DR</i> to <i>Monitoring fileset states for AFM DR</i> <i>Callback events for AFM and AFM DR</i> to <i>Monitoring callback events for AFM and AFM DR</i> <i>Prefetch</i> to <i>Monitoring prefetch</i> Moved the <i>Monitoring callback events for AFM and AFM DR</i> topic under <i>Monitoring health and events</i> Moved the <i>Monitoring with mmpmon</i> topic under <i>Monitoring performance</i> <p>Restructured the following topics:</p> <ul style="list-style-type: none"> <i>Monitoring system health by using the mmhealth command</i> This topic has been split into four topics: <ul style="list-style-type: none"> Monitoring the health of a node Event type and monitoring status for system health Threshold monitoring for system health Use cases <i>Monitoring the IBM Spectrum Scale system by using call home</i> This topic has been split into four topics: <ul style="list-style-type: none"> Understanding call home Configuring call home to enable manual and automated data upload Monitoring, uploading, and sharing collected data with IBM Support Use cases <i>List of performance metrics</i> This topic has been split into three topics: <ul style="list-style-type: none"> Linux metrics GPFS metrics Protocol metrics
	Troubleshooting	<ul style="list-style-type: none"> Under <i>SMB issues</i>, added a new topic <i>Slow access to SMB caused by contended access to files or directories</i>.

Table 3. List of changes in documentation (continued)

Guide	Knowledge center section	List of changes
Command and Programming Reference	Command reference	<ul style="list-style-type: none"> Removed the mmrest command man page.
	Programming reference	<ul style="list-style-type: none"> Added documentation for each IBM Spectrum Scale management API Version 2 command.

Changes in the Library and related publications section

- Under *Library and related publications*, the following topics were updated:
 - *Redbooks*[®], *Redpapers*[™], and *Blueprints*: Six new links added.
 - *ISV links*
 - *Applying IBM Spectrum Scale → Using AFM with object*
- Under *AFM-based Disaster Recovery* section, the *Failback of multiple filesets* use case was added.

Chapter 1. IBM Spectrum Scale support for Hadoop

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

Different Hadoop connectors

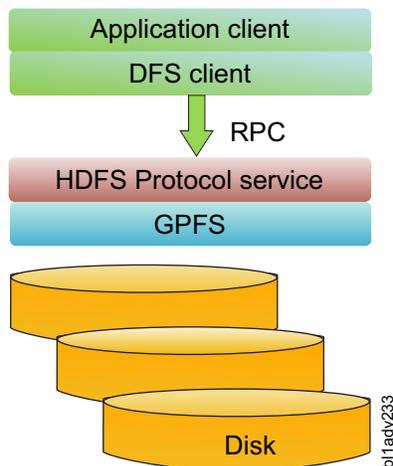
- 2nd generation HDFS Transparency
 - IBM Spectrum Scale HDFS Transparency (aka, HDFS Protocol) offers a set of interfaces that allows applications to use HDFS Client to access IBM Spectrum Scale through HDFS RPC requests. HDFS Transparency implementation integrates both the NameNode and the DataNode services and responds to the request as if it were HDFS.
- 1st generation Hadoop connector
 - The IBM Spectrum Scale Hadoop connector implements Hadoop file system APIs and the FileContext class so that it can access the IBM Spectrum Scale.

HDFS transparency

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

- HDFS compliant APIs or shell-interface command.
- Application client isolation from storage. Application Client may access data in IBM Spectrum 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.



For more information, visit the IBM Spectrum Scale developerWorks® wiki website and download the HDFS Transparency package.

Supported IBM Spectrum Scale storage modes

Local Storage Mode

HDFS transparency allows big data applications to access IBM Spectrum Scale local storage - File Placement Optimizer (FPO) mode and enables the support for shared storage mode (such as SAN-based storage, ESS) starting with `gpfs.hdfs-protocol.2.7.0-1` package.

In 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 is recommended to improve the access efficiency.

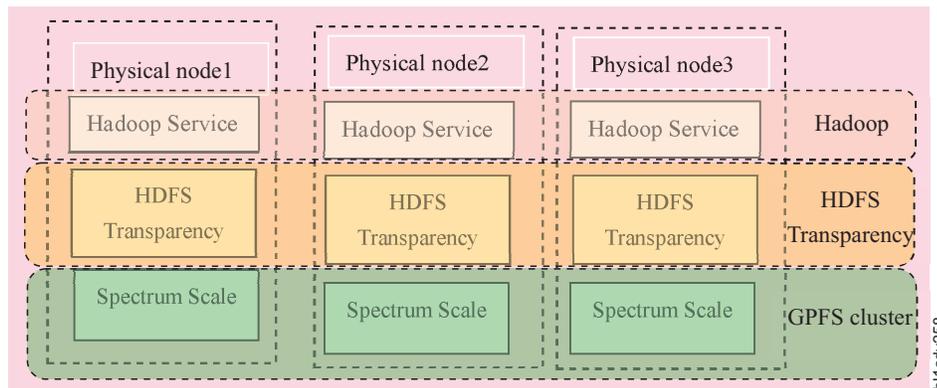


Figure 1. HDFS Transparency over IBM Spectrum Scale FPO

Shared Storage Mode

HDFS transparency allows big data applications to access data stored in shared storage mode, such as SAN-based storage.

In this mode, data is stored in shared storage systems which offer better storage efficiency than the local storage. RAID and other technologies can be used to protect hardware failure instead of using data replication.

DFS clients access data through the HDFS protocol remote procedure call (RPC). When a DFS Client requests to write blocks to IBM Spectrum Scale, the HDFS transparency NameNode randomly selects a DataNode for this request. When the DFS Client is located on a DataNode, that node will be selected for this request. When the DFS Client requests to **getBlockLocation** of an existing block, NameNode randomly selects a DataNodes for this request. For example, if the **dfs.replication** parameter is set to 3, three DataNodes are returned for a **getBlockLocation** request. If the **dfs.replication** parameter is set to 1, a single DataNode is returned for the **getBlockLocation** request.

HDFS transparency allows Hadoop application to access data stored in both a local IBM Spectrum Scale file system and a remote IBM Spectrum Scale file system from multiple cluster.

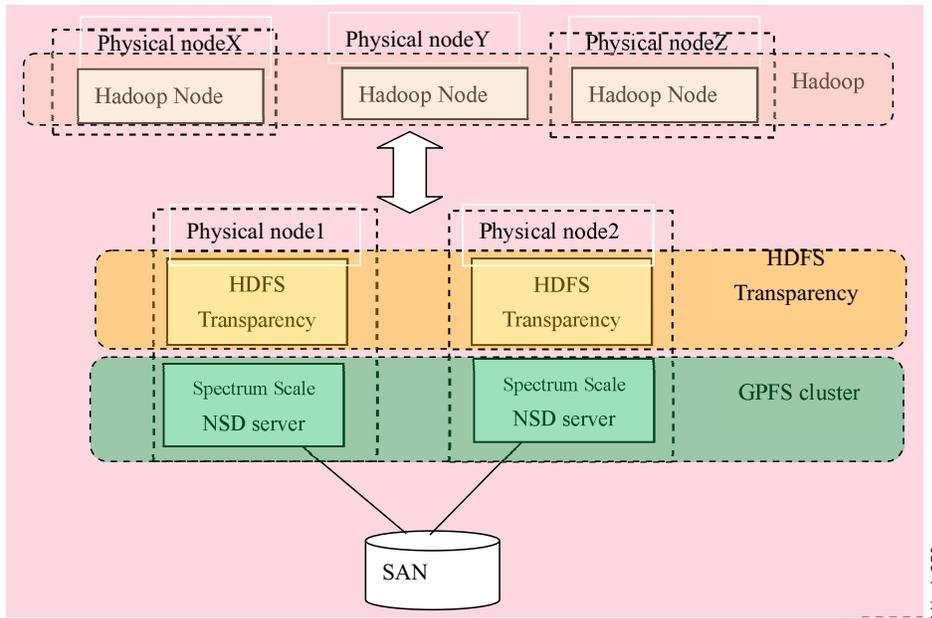
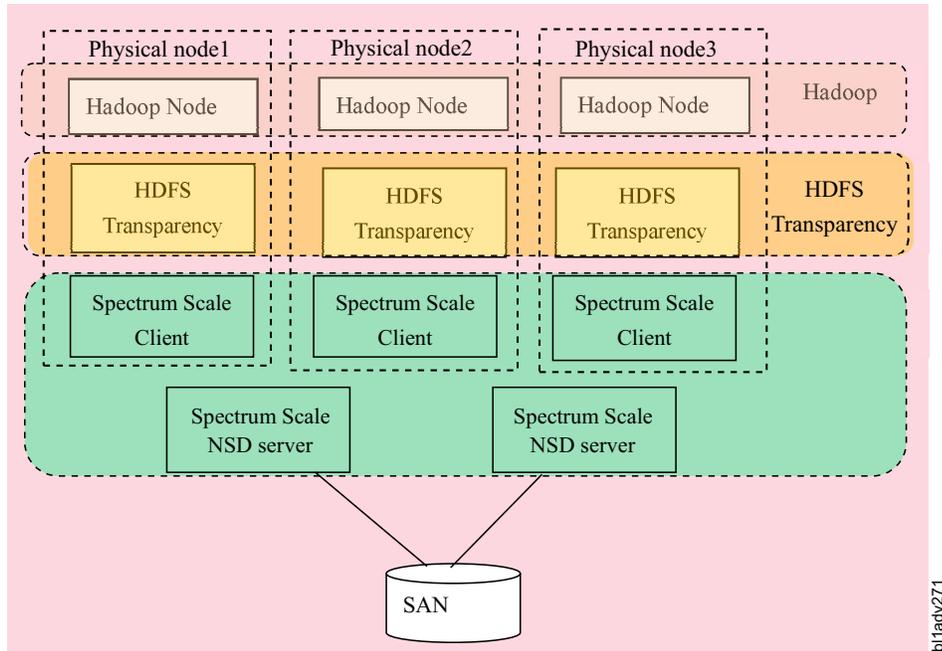


Figure 2. HDFS Transparency over Spectrum Scale NSD servers for shared storage

The Figure 2 is the recommended deployment for SAN-based storage. In this mode, HDFS Transparency is deployed over IBM Spectrum Scale NSD servers and all Hadoop components will take HDFS client to talk with HDFS Transparency over HDFS RPC. The data traffic will go from Hadoop client nodes, HDFS Transparency nodes/Spectrum Scale server nodes and then into/from SAN storage. If you are not considering short circuit read, this mode will give you the best performance.

Note: HDFS Transparency daemons are light weight processes. For more information, see Recommended Hardware Resource Configuration.



b11adv271

Figure 3. HDFS Transparency over limited Spectrum Scale client for shared storage

If the deployment in Figure 2 on page 3 is not accepted, the deployment in Figure 3 is recommended: install Spectrum Scale clients on all Hadoop nodes. The data traffic in Figure 4 on page 5 will go from Hadoop node, local lo (loop) network adapter, HDFS Transparency nodes/Spectrum Scale Clients, Spectrum Scale NSD servers and SAN storages. If short circuit read is enabled for this mode, the network traffic cost for local lo (loop) network adapter could be avoided.

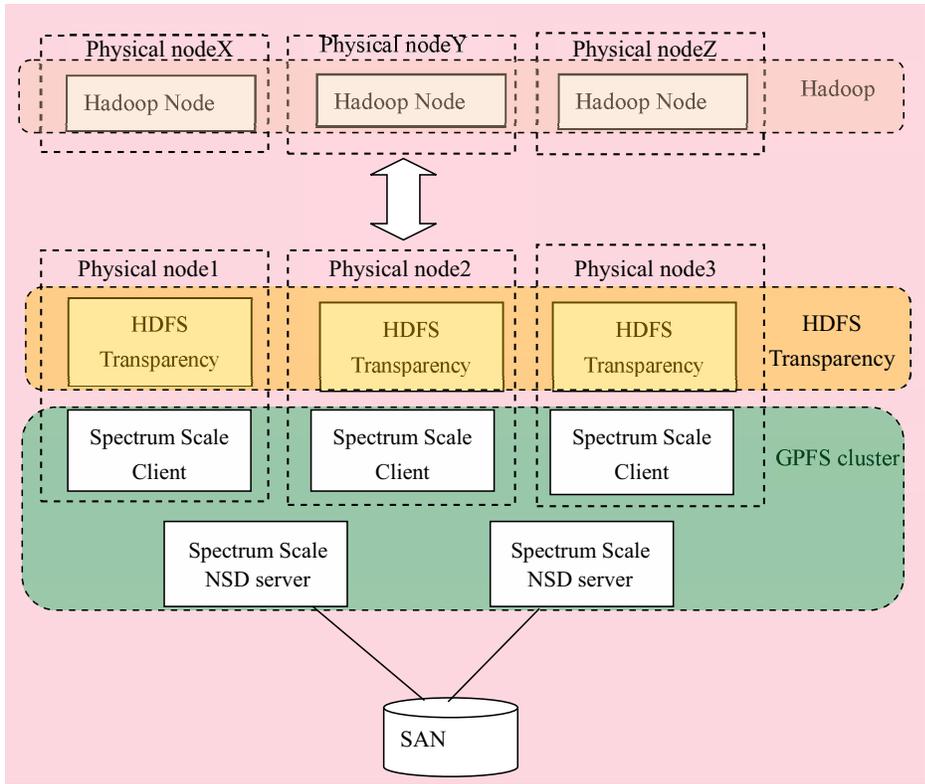


Figure 4. HDFS Transparency over Spectrum Scale clients for shared storage

In this deployment of Figure 4, the data traffic will go from Hadoop nodes, network RPC, HDFS Transparency nodes/Spectrum Scale Clients, network RPC, Spectrum Scale NSD servers and SAN storage. Short circuit read will not help data reading performance.

In Figure 4 or Figure 3 on page 4, the Spectrum Scale Clients and Spectrum Scale NSD servers could be configured with multi-cluster.

IBM ESS storage

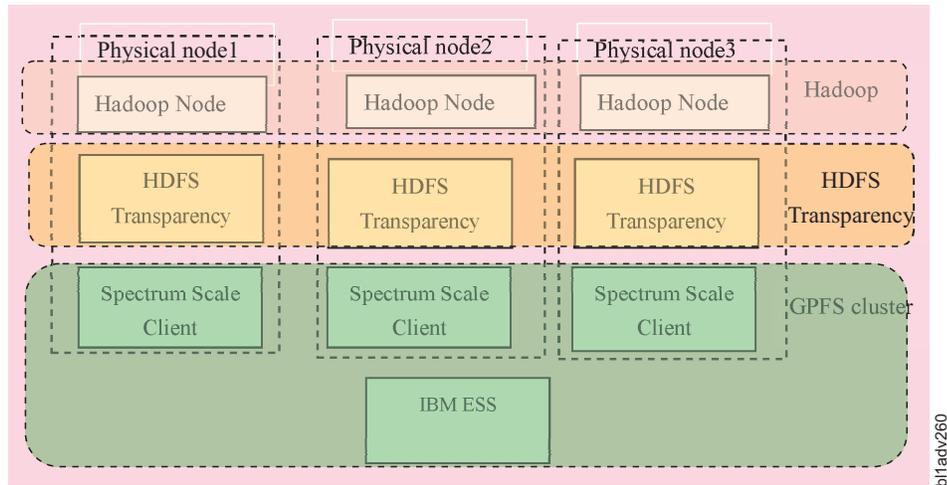


Figure 5. HDFS Transparency over IBM ESS

Figure 5 is the recommended deployment for IBM ESS if you have manageable Hadoop nodes (For example, Hadoop node number is less than 50). With short circuit read and short circuit write enabled, the RPC network latency could be obviously reduced. If RDMA is supported by the network, it further accelerates the data transfer between IBM ESS nodes and Spectrum Scale clients.

If there are a large number of nodes in the Hadoop cluster (for example, more than 1000 nodes), it is recommended to use the deployment pattern in Figure 6. Creating a large IBM Spectrum Scale cluster requires careful planning and increased demands on the network. The deployment pattern in Figure 6 limits the IBM Spectrum Scale deployment to just the nodes running the HDFS Transparency service rather than the entire Hadoop cluster.

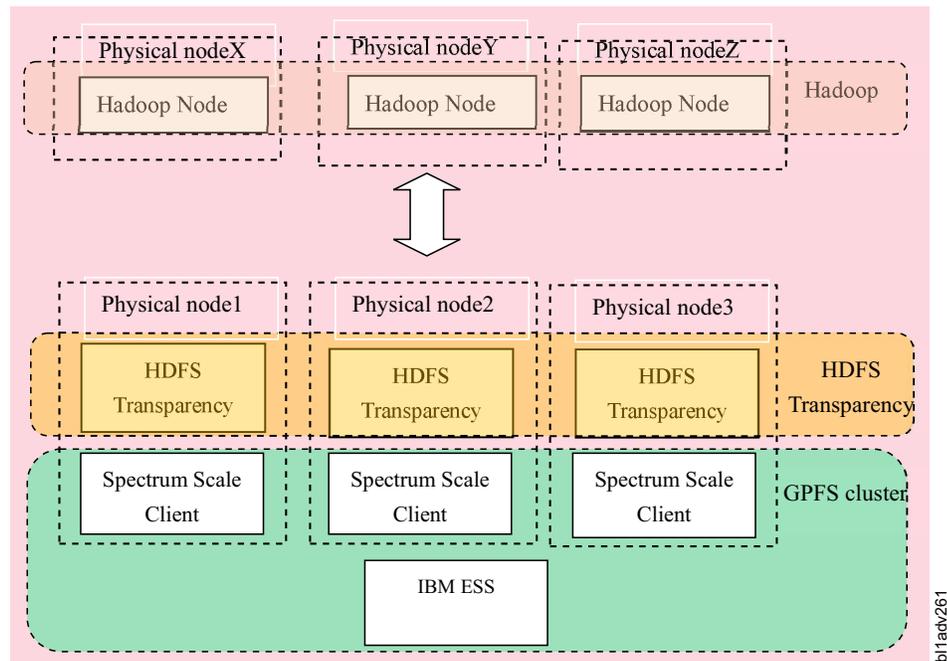


Figure 6. HDFS Transparency over limited Spectrum Scale Client for IBM ESS

- | In this deployment of Figure 6 on page 6, the data traffic will go from Hadoop nodes, network RPC, HDFS Transparency nodes/Spectrum Scale Clients, network RPC, Spectrum Scale NSD servers and SAN storage. “Short-circuit read configuration” on page 33 does not help the data reading performance.

Hadoop cluster planning

In an Hadoop cluster that runs the HDFS protocol, a node can take on the roles of 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 Spectrum Scale cluster or where only some of the nodes belong to an IBM Spectrum 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 28. The NameNode must be a part of an IBM Spectrum 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 in Hadoop 2.4, 2.5 and 2.7 releases.

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 or EditLog like state information.

DataNode

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

DFS Client

The DFS Client can be a part of an IBM Spectrum Scale cluster. When the DFS Client is a part of an IBM Spectrum Scale cluster, it can read data from IBM Spectrum Scale through an RPC or use the short-circuit mode. Otherwise, the DFS Client can access data from IBM Spectrum 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.

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

License planning

- | IBM Spectrum Scale Ambari Management Pack and HDFS Transparency do not require any additional license. If you have IBM Spectrum Scale license, you can get HDFS Transparency from IBM Spectrum Scale package or download the latest package from IBM developerWorks Wiki. You can also download the IBM Spectrum Scale Ambari Management Pack from IBM developerWorks Wiki

- | As for IBM Spectrum Scale license, any IBM Spectrum Scale license works with HDFS Transparency and IBM Spectrum Scale Ambari Management Pack. However, it is recommended to use IBM Spectrum Scale Standard Edition or IBM Spectrum Scale Advanced Edition or Data Management Edition so that you could leverage some advanced enterprise features (such as IBM Spectrum Scale storage pool, fileset, encryption or AFM) to power your Hadoop data platform.

- | Go through the Supported IBM Spectrum Scale storage modes section and select the mode you want to take and then refer to the license requirements in the following table:

Table 4. IBM Spectrum Scale License requirement

Storage Category	Deployment Mode	License requirement
IBM Spectrum Scale FPO	Illustrated in Figure 1 on page 2	<ul style="list-style-type: none"> • 3+ IBM Spectrum 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, maximum up to 8 quorum nodes in one cluster). • All other nodes take IBM Spectrum Scale FPO license.
IBM Spectrum Scale + SAN storage	Illustrated in Figure 2 on page 3	<ul style="list-style-type: none"> • All NSD servers must be with IBM Spectrum 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).
	Illustrated in Figure 3 on page 4 (configure one IBM Spectrum Scale cluster)	<ul style="list-style-type: none"> • 2+ IBM Spectrum Scale server license for quorum/NSD servers with tiebreak disks for quorum. If you want to tolerate more quorum node failure, configure more IBM Spectrum Scale NSD servers/quorum nodes. • All others take IBM Spectrum Scale clients.
	Illustrated in Figure 3 on page 4 (configure IBM Spectrum Scale Multi-cluster)	<ul style="list-style-type: none"> • For home IBM Spectrum Scale cluster (NSD server cluster), 2+ NSD servers (IBM Spectrum Scale server license) configured with tiebreak disks for quorum. If you want to tolerate more quorum node failure, configure more IBM Spectrum Scale NSD servers/quorum nodes. • For local IBM Spectrum Scale cluster (all as IBM Spectrum Scale clients), 3+ IBM Spectrum Scale server license for quorum/manager (configure more IBM Spectrum Scale server license node to tolerate more quorum node failure). All other take IBM Spectrum Scale client license.

Table 4. IBM Spectrum Scale License requirement (continued)

Storage Category	Deployment Mode	License requirement
IBM Spectrum Scale + SAN storage	Illustrated in Figure 4 on page 5 (configure one IBM Spectrum Scale cluster)	<ul style="list-style-type: none"> • 2+ IBM Spectrum Scale server license for quorum/NSD servers with tiebreak disks for quorum. If you want to tolerate more quorum node failure, configure more IBM Spectrum Scale NSD servers/quorum nodes. • All HDFS Transparency nodes take IBM Spectrum Scale clients.
	Illustrated in Figure 4 on page 5 (configure IBM Spectrum Scale Multi-cluster)	<ul style="list-style-type: none"> • For home IBM Spectrum Scale cluster (NSD server cluster), 2+ NSD servers (IBM Spectrum Scale server license) configured with tiebreak disks for quorum. If you want to tolerate more quorum node failure, configure more IBM Spectrum Scale NSD servers/quorum nodes. • For local IBM Spectrum Scale cluster (all as IBM Spectrum Scale clients), 3+ IBM Spectrum Scale server license for quorum/manager (configure more IBM Spectrum Scale server license node to tolerate more quorum node failure). All other HDFS Transparency nodes take IBM Spectrum Scale client license.

Table 4. IBM Spectrum Scale License requirement (continued)

Storage Category	Deployment Mode	License requirement
IBM ESS	Illustrated in Figure 5 on page 6 (configure one IBM Spectrum Scale cluster)	<ul style="list-style-type: none"> Take ESS nodes as quorum (you do not need to purchase new IBM Spectrum Scale license after you purchase IBM ESS). All other nodes take IBM Spectrum Scale client license.
	Illustrated in Figure 5 on page 6 (configure IBM Spectrum Scale Multi-cluster)	<ul style="list-style-type: none"> Create ESS nodes as home cluster (you do not need to purchase new IBM Spectrum Scale license after you purchase IBM ESS). For local IBM Spectrum Scale cluster (all as IBM Spectrum Scale clients), 3+ IBM Spectrum Scale server license for quorum/manager (configure more IBM Spectrum Scale server license node to tolerate more quorum node failure); all other take IBM Spectrum Scale client license.
	Illustrated in Figure Figure 6 on page 6 (configure one IBM Spectrum Scale cluster)	<ul style="list-style-type: none"> Take ESS nodes as quorum (you do not need to purchase new IBM Spectrum Scale license after you purchase IBM ESS). All HDFS Transparency nodes take IBM Spectrum Scale client license.
	Illustrated in Figure 6 on page 6 (configure IBM Spectrum Scale Multi-cluster)	<ul style="list-style-type: none"> Create ESS nodes as home cluster (you do not need to purchase new IBM Spectrum Scale license after you purchase IBM ESS). For local IBM Spectrum Scale cluster(all as IBM Spectrum Scale clients), 3+ IBM Spectrum Scale server license for quorum/manager(configure more IBM Spectrum Scale server license node to tolerate more quorum node failure); all other HDFS Transparency nodes take IBM Spectrum Scale client license.

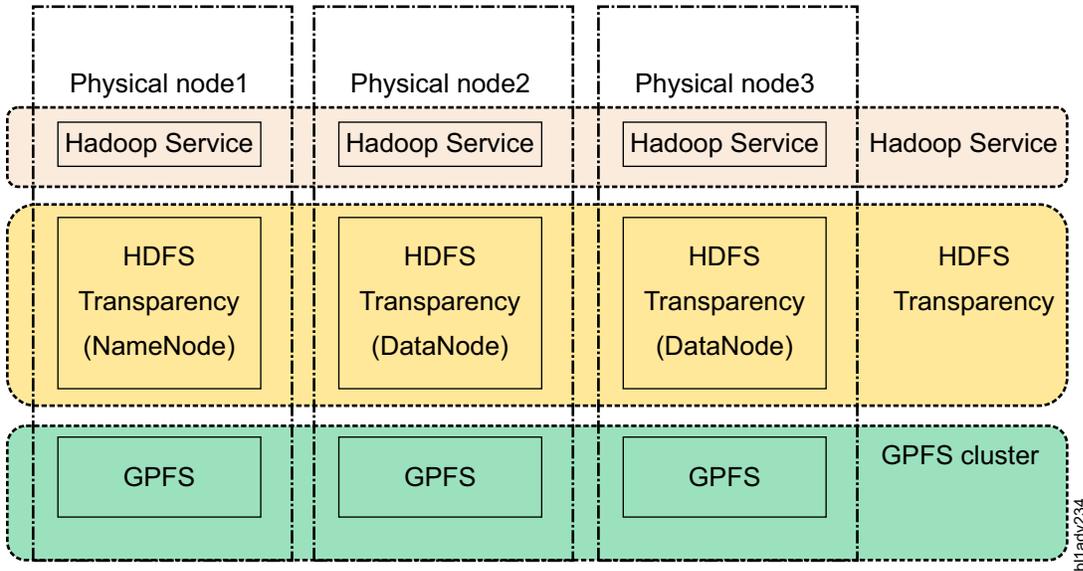
Note: If you plan to configure IBM Spectrum Scale protocol, you need to configure IBM Spectrum Scale protocol services over nodes with IBM Spectrum Scale server license but not any NSD disks in the file system.

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 Spectrum 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 Spectrum 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 Spectrum Scale cluster.

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

The shared storage mode or IBM ESS

According to the sections “Shared Storage Mode” on page 2, “IBM ESS storage” on page 5 and “License planning” on page 7 you can decide the HDFS Transparency nodes. Among these nodes, you need to define at least one NameNode and one DataNode.

If NameNode is configured, you need at least two nodes for NameNode HA and one node for DataNode. In production, you need at least two DataNodes to tolerate DataNode failure.

After HDFS transparency nodes are selected, follow the “Installation and configuration of HDFS transparency” on page 14 section to configure HDFS Transparency on these nodes.

Integration with Hadoop distributions

Finalize the HDFS Transparency NameNode host and DataNode hosts before deploying Hadoop distribution.

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 add this node into the IBM Spectrum Scale cluster. This setup results in a fewer configuration changes. Referring to Supported IBM Spectrum Scale storage modes, the deployment mode in Figure 1 on page 2,

Figure 3 on page 4 and Figure 5 on page 6 is feasible for this kind of integration. For other modes, if you want to take Ambari to manage HDFS Transparency and IBM Spectrum Scale services running on HDFS Transparency nodes, when you install IBM BigInsights IOP or HortonWorks HDP, you need to assign the node roles for native HDFS according to HDFS Transparency nodes.

For example, for the deployment mode of Figure 6 on page 6, if Physical node 1 is HDFS Transparency NameNode, Physical node 2 and Physical node 3 are HDFS Transparency DataNodes. When you install HortonWorks HDP over native HDFS, you should assign Physical node 1 as HDFS NameNode, Physical node 2 and Physical node 3 as HDFS DataNodes. This makes it easy when you add IBM Spectrum Scale into Ambari for integration.

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

Hardware and software requirements

Hardware & OS Matrix support

This section describes the hardware and software requirements for hadoop nodes and HDFS transparency.

The following table lists the Hardware and OS Matrix supported by HDFS Transparency:

HDFS Transparency	X86_64	ppc64	ppc64le
2.7.0-x	RHEL6+ RHEL7+ SLES11 SP3 SLES12+ Ubuntu14.04+	RHEL7+	RHEL7+ SLES12+ Ubuntu14.04+
2.7.2-0	RHEL6.7+ RHEL7.2+	RHEL7.2+	RHEL7.2+ SLES12+ Ubuntu14.04+
2.7.3-x	RHEL 6.7+ RHEL 7.2+	RHEL7.2+	RHEL7.2+ SLES12+ Ubuntu 14.04+

Note:

1. OpenJDK 1.8+ or Oracle JDK 1.8+ is required for HDFS Transparency versions 2.7.0-3+, 2.7.2-0+ and 2.7.3-0+. For HDFS Transparency 2.7.2-0 on RHEL7/7.1, ensure that the glibc version is at level 2.14 or later. Use the `rpm -qa | grep glibc` command to check the glibc version.
2. The version number RHEL 6.7+ means RHEL 6.7 and later. Others are similar.
3. If the Hadoop solution does not require management through Ambari, then the HDFS Transparency allows running a Hadoop cluster with mixed CPU architectures (for example, PPC64 or PPC64LE and x86_64). Careful planning is recommended for such deployments because the installation and management of such a mixed configuration is very complex.
4. For Hadoop solutions that are managed through Ambari (for example, HDP), deploying only on a homogeneous CPU architecture is supported. Currently, deployments on a mixed CPU architecture are not supported.

Recommended Hardware Resource Configuration

In 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).

As for memory requirement, refer to the following table:

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]: 1024bytes * inode number	2GB

Note:

1. Refer to the Disable Ranger Support section to disable the Ranger support from HDFS Transparency. By default, it is enabled. If you do not run Apache Ranger, you can disable it.
2. 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).

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.

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 Spectrum Scale, it is recommended that up to 25% of the physical memory be reserved for GPFS pagepool with a maximum of 20GB. 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 100GB 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 Spectrum 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 Spectrum 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, 192.168.1.0). Configure the second network interface as another subnet address (for example, 192.168.2.0).
2. Create the IBM Spectrum 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=192.168.2.0 -N all`.

Note: 192.168.2.0 is the subnet used for IBM Spectrum 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 Spectrum 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 Spectrum Scale cluster must be created using one network interface. The second network interface must be used for GPFS data traffic.

IBM ESS or SAN-based storage

For IBM ESS or SAN-based 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 Spectrum Scale over the bond adapters.
- The second option is to configure one adapter for IBM Spectrum Scale cluster and HortonWorks HDP and configure another adapter as subnets of IBM Spectrum Scale. For more information on subnets, see *GPFS and network communication* in the *IBM Spectrum Scale: Concepts, Planning, and Installation Guide*. Perform the following steps:
 1. Configure the first network interface with one subnet address (for example, 192.168.1.0). Configure the second network interface as another subnet address (for example, 192.168.2.0).
 2. Create the IBM Spectrum 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=192.168.2.0 -N all`.

Note: 192.168.2.0 is the subnet used for IBM Spectrum Scale data traffic.

Installation and configuration of HDFS transparency

This section describes the installation and configuration of HDFS transparency with IBM Spectrum Scale.

Note: For details on installing and configuring IBM Spectrum Scale, see the IBM Spectrum Scale Concepts, Planning and Installation Guide and the IBM Spectrum Scale Advanced Administration Guide in the IBM Knowledge Center. Also, see the best practices guide on the IBM DeveloperWorks GPFS Wiki.

Installation of HDFS transparency

IBM Spectrum Scale HDFS transparency must be installed on nodes that serve as NameNodes or DataNodes.

Use the following command to install the HDFS transparency RPM:

```
# rpm -hiv gpfs.hdfs-protocol-2.7.0-0.x86_64.rpm
```

This package has the following dependencies:

- libacl
- libattr
- openjdk 7.0+

HDFS transparency files are installed under the `/usr/lpp/mmfs/hadoop` directory. To list the contents of this directory, use the following command:

```
#ls /usr/lpp/mmfs/hadoop/  
bin etc lib libexec license logs README run sbin share
```

The following directories can be added to the system shell path for convenience:

- `/usr/lpp/mmfs/hadoop/bin` and
- `/usr/lpp/mmfs/hadoop/sbin`

Configuration of HDFS transparency

The following configurations are for manually configuring HDFS Transparency. For example, set up HDFS Transparency for open source Apache, or if you just want to set up HDFS Transparency service instead of HortonWorks HDP stack. If you take HortonWorks HDP and manage HDFS Transparency/IBM Spectrum Scale services from Ambari GUI, see BigInsights 4.2.5 and HortonWorks Data Platform 2.6.

For configuring, Hadoop distribution must be installed under `$YOUR_HADOOP_PREFIX` on each machine in the Hadoop cluster. The configurations for IBM Spectrum Scale HDFS transparency are located under `/usr/lpp/mmfs/hadoop/etc/hadoop` for any Hadoop distribution. Configuration files for Hadoop distribution are located in different locations. For example, `/etc/hadoop/conf` for IBM BigInsights IOP.

The `core-site.xml` and `hdfs-site.xml` configuration files should be synchronized between all the nodes and kept identical for the IBM Spectrum Scale HDFS transparency and Hadoop distribution. The `log4j.properties` configuration file can differ between the IBM Spectrum Scale HDFS transparency and the native 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 set up, HDFS transparency fails to start for HDFS transparency version earlier than 2.7.3-2.

For 2.7.3-2, HDFS Transparency supports only the root password-less ssh access. Since 2.7.3-3, support of non-root password-less ssh access is added.

If the file system is local, HDFS Transparency provides the following options for password-less requirement:

1. Option 1:

By default, HDFS Transparency requires root password-less access between any two nodes in the HDFS Transparency cluster.

2. Option 2:

If Option 1 is not feasible, you need at least one node with root password-less access to all the other HDFS Transparency nodes. In such a case, `mmhadoopctl` 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 Spectrum Scale cluster in admin central mode (**mmchconfig adminMode=central**), you can configure HDFS Transparency NameNodes on the IBM Spectrum 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.

1. Option 1:

By default, HDFS Transparency NameNodes require root password-less access to at least one of the contact nodes from the remote cluster.

Note: HDFS Transparency DataNodes do not require root password-less access to the contact nodes.

2. Option 2:

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` 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/mmlnsd, /usr/lpp/mmfs/bin/mmlsfileset, /usr/lpp/mmfs/bin/mmlsnapshot,
/usr/lpp/mmfs/bin/mmcrcsnapshot, /usr/lpp/mmfs/bin/mmdelnsnapshot, /usr/lpp/mmfs/bin/tlsdisk
```

The *gpfsadm* user can run these IBM Spectrum 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.

3. Option 3:

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.remoteclasscluster.autorefresh** as *false* in `/usr/lpp/mmfs/hadoop/etc/hadoop/gpfs-site.xml`.

Manually copy the **initmap.sh** script into one of the contact nodes. Log on the contact node as root and run the **initmap.sh** command. Copy the generated internal configuration files to all the HDFS Transparency nodes. For the **initmap.sh** script command syntax and generated internal configuration files, see the “Cluster and file system information configuration” on page 22 section.

| **Note:** If `gpfs.remoteclass.autorefresh` is configured as *false*, the snapshot from Hadoop interface
| is not supported against the remote mounted file system.
| If the IBM Spectrum Scale cluster is configured as `adminMode=central` (check by executing `mmfsconfig`
| `adminMode`), HDFS Transparency NameNodes can be configured on the management nodes of the IBM
| Spectrum Scale cluster.
| If the IBM Spectrum Scale cluster is configured in sudo wrapper mode, IBM Spectrum Scale requires
| the user to have password-less root access to all the other nodes as a common user (that means, log in
| as a root user in the node and execute `ssh <non-root>@<other-node>` in the password-less mode).
| However, in this mode of IBM Spectrum Scale, HDFS Transparency Namenodes requires the node that
| is configured with the user root password-less access to all the other nodes as the user root and
| `mmhadoopctl` can only be run on these nodes with the user root password-less access to all other nodes
| as a user root.

OS tuning for all nodes

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.

Note: This must be done on all nodes including the Hadoop client nodes and the HDFS Transparency nodes.

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 Hadoop nodes

This example describes the configuration of Hadoop nodes.

For configuring Hadoop nodes, the following files need to be updated for open source Apache Hadoop:

- `core-site.xml`
- `slaves`

In the following example, the hostname of NameNode service is `hs22n44`.

In `$(HADOOP_PREFIX)/etc/hadoop/core-site.xml`, ensure that the `fs.defaultFS` parameter is set to point to the HDFS Transparency NameNode:

```
<property>
<name>fs.defaultFS</name>
<value>hdfs://hs22n44:9000</value>
</property>
```

Replace `hs22n44:9000` with the hostname of your NameNode service and preferred port number.

User can customize other configuration parameters like service ports. For more information, see Welcome to Apache™ Hadoop®!.

In `$YOUR_HADOOP_PREFIX/etc/hadoop/slaves` file, ensure that all the DataNodes are listed in the file. For example:

```
# cat $HADOOP_PREFIX/etc/hadoop/slaves
hs22n44
hs22n54
hs22n45
```

As for `hdfs-site.xml` and other detailed configuration settings in `core-site.xml`, follow Welcome to Apache™ Hadoop®! to configure Hadoop nodes. The following must be configured to avoid unexpected exceptions from Hadoop:

```
<property>
<name>dfs.namenode.handler.count</name>
<value>600</value>
</property>

<property>
<name>dfs.datanode.handler.count</name>
<value>400</value>
</property>

<property>
<name>dfs.datanode.max.transfer.threads</name>
<value>8192</value>
</property>
```

After the configuration, synchronize them to all Hadoop nodes.

Note: For a Hadoop distribution, like IBM BigInsights or Hortonworks Data Platform® (HDP), configure the Hadoop components (for example, HBase, Hive, oozie, etc) in the management GUI. For example, Ambari for IBM BigInsights or HDP.

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+ and 2.7.3-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.

Synchronization of Hadoop configurations:

By default, HDFS transparency uses the core-site.xml and hdfs-site.xml configuration files from the Hadoop distribution, along with the gpfs-site.xml located under /usr/lpp/mmfs/hadoop/etc/hadoop directory.

The following configuration files need to be distributed to all the HDFS transparency nodes:

- core-site.xml
- hdfs-site.xml
- slaves
- log4j.properties

If the HDFS Transparency nodes are also running Hadoop, you can use the following command to sync the configuration files to the rest of the HDFS Transparency nodes. Run this command on one of the HDFS Transparency nodes (For example, hdfs_transparency_node1) after ensuring that the HDFS Transparency service is running:

```
hdfs_transparency_node1#/usr/lpp/mmfs/hadoop/sbin/mmhadoopctl connector syncconf <hadoop-conf-dir>
```

If the HDFS transparency nodes are not running Hadoop, use a tool like scp (secure copy) to distribute the following files to the /usr/lpp/mmfs/hadoop/etc/hadoop/ directory on all the HDFS transparency nodes:

- <hadoop-conf-dir>/core-site.xml
- <hadoop-conf-dir>/hdfs-site.xml
- <hadoop-conf-dir>/log4j.properties

Configure the storage mode:

Use this procedure to configure the storage mode.

Modify the /usr/lpp/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 storage mode. The storage mode can be local or shared. This is a required configuration parameter and the gpfs-site.xml file must be synced to all the HDFS transparency nodes after the above 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 `/usr/lpp/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 `gpfs.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 Spectrum Scale, IBM Spectrum 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 Spectrum 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 `/usr/lpp/mmfs/hadoop/etc/hadoop/`. These two files must be consistent as the files used by Hadoop nodes.

You need to modify `/usr/lpp/mmfs/hadoop/etc/hadoop/slaves` to add all HDFS transparency DataNode hostnames and one hostname per line, for example:

```
# cat /usr/lpp/mmfs/hadoop/etc/hadoop/slaves
hs22n44
hs22n54
hs22n45
```

You might check `/usr/lpp/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 Spectrum Scale HDFS transparency nodes:

```
hdfs_transparency_node1# /usr/lpp/mmfs/hadoop/sbin/mmhadoopctl connector syncconf /usr/lpp/mmfs/hadoop/etc/hadoop
```

Configuring IBM BigInsights IOP

In IBM BigInsights IOP 4.0, IOP and IBM Spectrum Scale HDFS Transparency are integrated manually.

For IBM BigInsights IOP 4.1, if the deployment was done with the old Hadoop connector with IBM Spectrum Scale Ambari integration package `gpfs.ambari-iop_4.1-X.X.noarch.bin`, see Upgrade IOP 4.1 + Ambari from Hadoop connector to HDFS Transparency Guide.

If you deployed IOP 4.1 with the old Hadoop connector but without IBM Spectrum Scale Ambari integration package `gpfs.ambari-iop_4.1-X.X.noarch.bin`, perform the following steps to move to HDFS Transparency:

1. On `hdfs_transparency_node1`, run the following command to sync IBM BigInsights IOP configuration into IBM Spectrum Scale HDFS transparency configuration directory: `/usr/lpp/mmfs/hadoop/sbin/mmhadoopctl connector syncconf /etc/hadoop/conf/`
2. On `hdfs_transparency_node1`, create the `/usr/lpp/mmfs/hadoop/etc/hadoop/gpfs-site.xml` file, update the `/usr/lpp/mmfs/hadoop/etc/hadoop/slaves` and `/usr/lpp/mmfs/hadoop/etc/hadoop/log4j.properties` files.
3. On the node `hdfs_transparency_node1`, run the `/usr/lpp/mmfs/hadoop/sbin/mmhadoopctl connector syncconf /usr/lpp/mmfs/hadoop/etc/hadoop/` command to sync the `gpfs-site.xml`, `core-site.xml`, `hdfs-site.xml`, `slaves` and `log4j.properties` to all the IBM Spectrum Scale HDFS transparency nodes. For deploying IBM BigInsights 4.1/4.2, follow the corresponding Deploying BigInsights IBM Spectrum Scale HDFS Transparency with Ambari guide under the IBM developerWorks References wiki page.

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`.

In order to update this, follow these steps:

1. On the HDFS Transparency NameNode, modify the `/usr/lpp/mmfs/Hadoop/etc/hadoop/hadoop-env.sh` and other files as necessary.
2. Sync the changes to all the HDFS Transparency nodes by executing the following command:

```
#/usr/lpp/mmfs/hadoop/sbin/mmhadoopctl connector syncconf /usr/lpp/mmfs/hadoop/etc/hadoop
```

Note: For HDFS Transparency 2.7.2-3+ or 2.7.3-0+, `/usr/lpp/mmfs/hadoop/etc/hadoop/hadoop-env.sh` is synced by `mmhadoopctl connector syncconf`. For older versions of HDFS Transparency, you need to take `scp` to copy the updated `hadoop-env.sh` to all other HDFS Transparency nodes.

Start and stop the service manually

To start and stop HDFS transparency services manually, you need to be a root user. You also need to keep native HDFS service down because HDFS transparency provides the same services. If you keep both the services running, a conflict is reported in the service network port number. You need to restart all other Hadoop services, such as Yarn, Hive, HBase, etc after you replace native HDFS with HDFS transparency.

To start the HDFS transparency service from any Transparency node, use the following command:

```
/usr/lpp/mmfs/hadoop/sbin/mmhadoopctl connector start
```

To stop the HDFS transparency service from any Transparency node, use the following command:

```
/usr/lpp/mmfs/hadoop/sbin/mmhadoopctl connector stop
```

Connector health check

Any user can conduct a connector health check of the Hadoop service.

To conduct a health check of the service, issue the following command:

```
/usr/lpp/mmfs/hadoop/bin/mmhadoopctl connector getstate  
# hadoop dfs -ls /
```

If you see the configured nodes running HDFS Transparency NameNode and DataNode services and there are files output from the **hadoop dfs -ls /** command, then the setup is successful.

Cluster and file system information configuration

After HDFS Transparency is successfully started for the first time, it executes a script called **initmap.sh** to automatically generate the internal configuration files that contain the GPFS cluster information, disk-to-hostname map information and ip-to-hostname map information.

For HDFS transparency 2.7.3-0 and earlier, if the new disks are added in the file system or if the file system is recreated, you need to manually create the **initmap.sh** script on the HDFS Transparency Namenode for updating the internal configuration files with the new information.

For HDFS Transparency 2.7.3-1 and later, the namenode runs the **initmap.sh** script every time it starts. Therefore, you do not need to run the script manually.

From HDFS Transparency 2.7.3-2 and later, the internal configuration files are automatically generated if they are not detected and are synched to all the other HDFS Transparency nodes when HDFS Transparency is started.

For the **initmap.sh** script to generate the proper internal configuration files, see the “Password-less ssh access” on page 15 section.

Unexpected exceptions can be seen in the HDFS Transparency logs if the internal configuration files are not properly created.

Note:

- The internal configuration files can be removed from all the nodes and regenerated if the HDFS Transparency is restarted or if the **initmap.sh** command is executed depending on your HDFS Transparency version.
- If the internal configuration files are missing, HDFS transparency re-runs the script and will take longer to start.

Following are the command syntaxes for the **initmap.sh** script for the various HDFS Transparency releases:

- HDFS Transparency 2.7.3-0 and earlier:

```
/usr/lpp/mmfs/hadoop/sbin/initmap.sh <fsName> diskinfo nodeinfo clusterinfo
```

Generated internal configuration files **diskid2hostname**, **nodeid2hostname** and **clusterinfo4hdfs** are under the **/var/mmfs/etc/** directory.

- HDFS Transparency 2.7.3-1 and later:

- HDFS Transparency 2.7.3-1:

```

| /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:
| /usr/lpp/mmfs/hadoop/sbin/initmap.sh -d -r -u [gpfs.ssh.user] -i all [fsName]
|
| Note:
| - The -d option propagates the generated files to all the nodes. You should use this option only
| when running on the Namenode.
| - The -r option requests to run the necessary commands on the contact nodes to generate the
| config files if this is a remote mounted filesystem.
| - The -u [gpfs.ssh.user] is an optional parameter used only if gpfs.ssh.user is set.
| - The -i all option generates all the required configuration files.
| Generated internal configuration files diskid2hostname.<fs-name>, nodeid2hostname.<fs-name> and
| clusterinfo4hdfs.<fs-name> are under the /var/mmfs/etc/hadoop directory.

```

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 Spectrum Scale filesystem 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 slave files. However, HDFS transparency can use a set of configuration files that are different from yarn. In that case, slave 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

You can use the HDFS shell command line with the HDFS transparency.

You can access commands from the HDFS command shell:

`$YOUR_HADOOP_PREFIX/bin/hdfs`

Usage: `hdfs [--config confdir] COMMAND`

where `COMMAND` is one of:

<code>dfs</code>	run a filesystem command on the file systems supported in Hadoop.
<code>namenode -format</code>	format the DFS filesystem
<code>secondarynamenode</code>	run the DFS secondary namenode
<code>namenode</code>	run the DFS namenode
<code>journalnode</code>	run the DFS journalnode
<code>zkfc</code>	run the ZK Failover Controller daemon
<code>datanode</code>	run a DFS datanode
<code>dfsadmin</code>	run a DFS admin client
<code>haadmin</code>	run a DFS HA admin client
<code>fsck</code>	run a DFS filesystem checking utility
<code>balancer</code>	run a cluster balancing utility
<code>jmxget</code>	get JMX exported values from NameNode or DataNode.
<code>mover</code>	run a utility to move block replicas across storage types
<code>oiv</code>	apply the offline fsimage viewer to an fsimage

oiv_legacy	apply the offline fsimage viewer to an legacy fsimage
oev	apply the offline edits viewer to an edits file
fetchdt	fetch a delegation token from the NameNode
getconf	get config values from configuration
groups	get the groups which users belong to
snapshotDiff	diff two snapshots of a directory or diff the current directory contents with a snapshot
lsSnapshottableDir	list all snapshottable dirs owned by the current user Use -help to see options
portmap	run a portmap service
nfs3	run an NFS version 3 gateway
cacheadmin	configure the HDFS cache
crypto	configure HDFS encryption zones
storagepolicies	list/get/set block storage policies
version	print the version

Most commands print help when invoked without parameters.

Note: All commands from **hdfs dfs** are supported (hdfs dfs -du and hdfs dfs -df are not exact in the output in HDFS Transparency 2.7,0-x. However, these issues have been fixed in HDFS Transparency version 2.7.2-0). Other commands from HDFS interface are not supported because these commands are not needed for IBM Spectrum Scale. For example, hdfs namenode -format.

Upgrading the HDFS Transparency cluster

Before upgrading the HDFS transparency, you need to remove the older IBM Spectrum Scale Hadoop connector.

Removing IBM Spectrum Scale Hadoop connector

The following sections describe how to remove IBM Spectrum Scale Hadoop connector based on the version of the Hadoop connector and the IBM Spectrum Scale version.

Refer to the section that pertain to your setup environment.

Removing IBM Spectrum Scale Hadoop connector 2.4 over IBM Spectrum Scale 4.1.0.4, 4.1.0.5, 4.1.0.6 or 4.1.0.7 releases

For users who are using IBM Spectrum Scale Hadoop connector 2.4 over IBM Spectrum Scale 4.1.0.4, 4.1.0.5, 4.1.0.6 or 4.1.0.7 releases, this section explains the steps required to remove the old connector over each node in the cluster.

If you are using Hadoop 1.x, IBM Spectrum Scale Hadoop Connector 2.7 does not support Hadoop 1.x and you need to upgrade your Hadoop version first.

1. Remove any links or copies of the `hadoop-gpfs-2.4.jar` file from your Hadoop distribution directory. Also, remove any links or copies of the `libgpfs_hadoop.64.so` file from your Hadoop distribution directory.

Note: For IBM BigInsights IOP 4.0, the distribution directory is `/usr/iop/4.0.0.0`.

2. Stop the current connector daemon:

```
# ps -elf | grep gpfs-connector-daemon
# kill -9 <pid-of-connector-daemon>
```

3. Run the following commands, to remove callbacks from IBM Spectrum Scale:

```
cd /usr/lpp/mmfs/fpo/hadoop-2.4/install_script
./gpfs-callbacks.sh --delete
```

Run the **mm1scalbacks all** command to check whether connector-related callbacks, such as callback ID `start-connector-daemon` and `stop-connector-daemon`, are removed. The IBM Spectrum Scale Hadoop connector callbacks are cluster-wide and this step is required to be done over any one of nodes.

4. Remove the following files:

```
# rm -f /var/mmfs/etc/gpfs-callbacks.sh
# rm -f /var/mmfs/etc/gpfs-callback_start_connector_daemon.sh
# rm -f /var/mmfs/etc/gpfs-callback_stop_connector_daemon.sh
# rm -f /var/mmfs/etc/gpfs-connector-daemon
```

5. Remove the IBM Spectrum Scale Scale-specific configuration from your Hadoop `core-site.xml` file. Modify the **fs.defaultFS** into an HDFS schema format after removing the following configurations:

```
fs.AbstractFileSystem.gpfs.impl,
fs.AbstractFileSystem.hdfs.impl, fs.gpfs.impl, fs.hdfs.impl,
gpfs.mount.dir, gpfs.supergroup
```

Install and set up the HDFS transparency, see the “Manually upgrading the IBM Spectrum Scale HDFS Transparency connector” on page 26 for more information.

Removing IBM Spectrum Scale Hadoop connector 2.4 over IBM Spectrum Scale 4.1.0.7 or 4.1.0.8 releases

For users who are using IBM Spectrum Scale Hadoop connector 2.4 over IBM Spectrum Scale 4.1.0.7 or 4.1.0.8 releases, this section explains the steps required to remove the old connector over each node in the cluster.

If you are using Hadoop 1.x, IBM Spectrum Scale Hadoop Connector 2.7 does not support Hadoop 1.x and you need to upgrade your Hadoop version first.

1. To stop the connector services, run **mmhadoopctl connector stop** on all nodes.
2. To detach the connector, run **mmhadoopctl connector detach --distribution BigInsights** on any one node.
3. To uninstall the connector, run **rpm -e gpfs.hadoop-2-connector** on all nodes.
4. Remove the IBM Spectrum Scale-specific configuration from your Hadoop `core-site.xml` file. Modify the **fs.defaultFS** into an HDFS schema format by removing the following configurations:

```
fs.AbstractFileSystem.gpfs.impl,
fs.AbstractFileSystem.hdfs.impl, fs.gpfs.impl, fs.hdfs.impl,
gpfs.mount.dir, gpfs.supergroup
```

Install and set up the HDFS transparency, see the “Manually upgrading the IBM Spectrum Scale HDFS Transparency connector” on page 26 for more information.

Removing IBM Spectrum Scale Hadoop connector 2.4 or 2.5 over IBM Spectrum Scale 4.1.1 and later releases

For users who are using IBM Spectrum Scale Hadoop connector 2.4 or 2.5 in IBM Spectrum Scale 4.1.1 and later, this section explains the steps required to remove the old connector over each node in the cluster.

If you are using Hadoop 1.x, IBM Spectrum Scale Hadoop Connector 2.7 does not support Hadoop 1.x and you need to upgrade your Hadoop version first.

1. To stop the connector service, run **mmhadoopctl connector stop** on all nodes.
2. To detach the connector, run **mmhadoopctl connector detach --distribution BigInsights** on all nodes.
3. To detach the connector, run **rpm -e gpfs.hadoop-2-connector** on all nodes.
4. Remove the IBM Spectrum Scale-specific configuration from your Hadoop `core-site.xml` file. Modify the **fs.defaultFS** into an HDFS schema format by removing the following configurations:

```
fs.AbstractFileSystem.gpfs.impl,  
fs.AbstractFileSystem.hdfs.impl, fs.gpfs.impl, fs.hdfs.impl,  
gpfs.mount.dir, gpfs.supergroup
```

Install and set up the HDFS transparency, see the “Manually upgrading the IBM Spectrum Scale HDFS Transparency connector” for more information.

Removing IBM Spectrum Scale Hadoop connector 2.7 (earlier release) over IBM Spectrum Scale 4.1.0.7, 4.1.0.8, or 4.1.1 and later releases

For users who are using IBM Spectrum Scale Hadoop connector 2.7 (earlier release) over IBM Spectrum Scale 4.1.0.7, 4.1.0.8, or 4.1.1 and later releases, this section explains the steps required to remove the old connector over each node in the cluster.

If you are using Hadoop 1.x, IBM Spectrum Scale Hadoop Connector 2.7 does not support Hadoop 1.x and you need to upgrade your Hadoop version first.

1. **mmhadoopctl connector stop**
2. **mmhadoopctl connector detach --distribution BigInsights**
3. **rpm -e gpfs.hadoop-2-connector**
4. Remove the IBM Spectrum Scale-specific configuration from your Hadoop core-site.xml file. Modify the **fs.defaultFS** into an HDFS schema format by removing the following configurations:

```
fs.AbstractFileSystem.gpfs.impl,  
fs.AbstractFileSystem.hdfs.impl, fs.gpfs.impl, fs.hdfs.impl,  
gpfs.mount.dir, gpfs.supergroup
```

For more information on installing and setting up the HDFS transparency, see the “Manually upgrading the IBM Spectrum Scale HDFS Transparency connector.”

Manually upgrading the IBM Spectrum Scale HDFS Transparency connector

Perform the following procedure to upgrade the HDFS transparency cluster.

1. Back up the `/usr/lpp/mmfs/hadoop/etc/hadoop` configuration directory.
2. Disable short circuit if it is currently enabled.
3. Stop the HDFS transparency service on all nodes by running the following command:
`/usr/lpp/mmfs/hadoop/sbin/mmhadoopctl connector stop.`
4. Upgrade the RPM on each node by running the following command: **`rpm -U gpfs.hdfs-protocol-2.7.0-<x>.x86_64.rpm`**. It does not update any configuration files under the `/usr/lpp/mmfs/hadoop/etc/hadoop` directory.

The `core-site.xml`, `hdfs-site.xml` and `slaves` files are not removed during the upgrade.

5. Start HDFS transparency service on all nodes by running the following command:
`/usr/lpp/mmfs/hadoop/sbin/mmhadoopctl connector start.`
6. Enable short circuit again if it was previously disabled in step 2.

Rolling upgrade for HDFS Transparency

- | HDFS Transparency supports rolling upgrade.
- | When you perform a rolling upgrade, perform the upgrade procedure for each HDFS Transparency Datanode and the upgrade procedure for the HDFS Transparency Namenodes.
- | Run the following steps for each HDFS Transparency DataNode:
 - | 1. Stop the DataNode from Ambari/GUI (**HDFS > Summary > Click DataNodes**) and stop the target DataNode. If you are not using Ambari, run the following command on the bash console of target node as the user root:

```

| 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/bin/gpfs stop datanode
| 2. Run rpm -e gpfs.hdfs-protocol to uninstall the old version of HDFS Transparency.
| 3. Run rpm -ivh <path-to-new-HDFS-Transparency-rpm> to install the new version of HDFS
| Transparency.
| 4. Start the DataNode on the target node: Run the following command on the bash console of target
| DataNode:
| 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/bin/gpfs start datanode
|
| Note: After all the DataNodes are upgraded, run the following procedure to upgrade the HDFS
| Transparency NameNode.

```

Upgrading HDFS Transparency NameNode

Run the following steps for Namenode:

```

| 1. If you configure NameNode HA, stop the standby HDFS Transparency NameNode from Ambari GUI
| or from bash console with the following commands:
| 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/bin/gpfs start namenode
| If you do not configure HDFS Transparency NameNode HA, go to Step 4
| 2. Upgrade the standby HDFS Transparency NameNode. Run rpm -e gpfs.hdfs-protocol to uninstall
| the old version of HDFS Transparency and run rpm -ivh <path-to-new-HDFS-Transparency-rpm> to
| install the new version of HDFS Transparency.
| 3. Start the standby NameNode from Ambari GUI or run the following command from the bash console:
| 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/bin/gpfs start namenode
| 4. Stop the active HDFS Transparency NameNode. You can stop active HDFS Transparency NameNode
| from the Ambari GUI or run the following command to stop it from the bash console of the active
| 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/bin/gpfs stop namenode
| 5. Upgrade the HDFS Transparency NameNode in Step 4. Run rpm -e gpfs.hdfs-protocol to uninstall
| the old version of HDFS Transparency and run rpm -ivh <path-to-new-HDFS-Transparency-rpm> to
| install the new version of the HDFS Transparency.
| 6. Start the NameNode in Step 4 from Ambari GUI or run the following command from the bash
| console:
| 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/bin/gpfs start namenode
|
| Note: When you upgrade the HDFS Transparency NameNode, if HA is not configured, HDFS
| Transparency service is interrupted. If you configure HA for NameNode, your running jobs are
| interrupted when you stop the active NameNode if the HDFS Transparency is older than 2.7.3-1.

```

Security

HDFS transparency supports full Kerberos and it is verified over IBM BigInsights IOP 4.1.0.2 and 4.2.

HDFS Transparency is certificated with IBM Security Guardium® DAM (Database Activity Monitoring) to monitor the Hadoop Data Access over IBM Spectrum Scale. For more information, see Big data security and auditing with IBM InfoSphere® Guardium.

Advanced features

This section describes the advanced features of Hadoop.

High availability configuration

This section describes the high availability configuration for HDFS transparency.

Manual HA switch configuration

High Availability (HA) is implemented in HDFS Transparency by using a shared directory in the IBM Spectrum Scale filesystem.

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 name nodes ID for HA-->
<name>dfs.ha.namenodes.mycluster</name>
<value>nn1,nn2</value>
</property>

<property>
<!--Actual hostname and rpc address of name node ID-->
<name>dfs.namenode.rpc-address.mycluster.nn1</name>
<value>c8f2n06.gpfs.net:8020</value>
</property>

<property>
<!--Actual hostname and rpc address of name node ID-->
<name>dfs.namenode.rpc-address.mycluster.nn2</name>
<value>c8f2n07.gpfs.net:8020</value>
</property>

<property>
<!--Actual hostname and http address of name node ID-->
<name>dfs.namenode.http-address.mycluster.nn1</name>
<value>c8f2n06.gpfs.net:50070</value>
</property>

<property>
<!--Actual hostname and http address of name node ID-->
<name>dfs.namenode.http-address.mycluster.nn2</name>
<value>c8f2n07.gpfs.net:50070</value>
</property>
```

```

<property>
<!--Shared directory used for status sync up-->
<name>dfs.namenode.shared.edits.dir</name>
<value>/<gpfs.mnt.dir>/<gpfs.data.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 **dfs.namenode.shared.edits.dir** configuration parameter must be consistent with `gpfs.mnt.dir` and `gpfs.data.dir` defined in `/usr/lpp/mmfs/hadoop/etc/hadoop/gpfs-site.xml`. You can create the directory `/<gpfs.mnt.dir>/<gpfs.data.dir>/HA` and change the ownership to `hdfs:hadoop` before starting the HDFS transparency services.

The **dfs.ha.standby.checkpoints** must be set to *false*. Otherwise, you will see a log of exceptions in the standby NameNode logs, such as:

```
ERROR ha.StandbyCheckpointer (StandbyCheckpointer.java:doWork(371)) - Exception in doCheckpoint
```

In HDFS transparency, NameNode does not maintain a state such as `fsImage` or `editLogs` as in native HDFS. Therefore, there is no need to perform checkpoints from the standby NameNode service.

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.

- 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`.
- 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.

- 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
```
- Start the HDFS transparency service by running the **mmhadoopctl** command:

```
# mmhadoopctl connector start
```
- 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 [name node 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 [name node ID]
```

You could check your `/usr/lpp/mmfs/hadoop/etc/hadoop/hdfs-site.xml` or run the following commands to figure out the [name node 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 IBM BigInsights IOP, 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 name nodes ID for HA-->
<name>dfs.ha.namenodes.mycluster</name>
<value>nn1,nn2</value>
</property>

<property>
<!--Actual hostname and rpc address of name node ID-->
<name>dfs.namenode.rpc-address.mycluster.nn1</name>
<value>c8f2n06.gpfs.net:8020</value>
</property>

<property>
<!--Actual hostname and rpc address of name node ID-->
<name>dfs.namenode.rpc-address.mycluster.nn2</name>
<value>c8f2n07.gpfs.net:8020</value>
</property>

<property>
<!--Actual hostname and http address of name node ID-->
<name>dfs.namenode.http-address.mycluster.nn1</name>
<value>c8f2n06.gpfs.net:50070</value>
</property>

<property>
<!--Actual hostname and http address of name node ID-->
<name>dfs.namenode.http-address.mycluster.nn2</name>
<value>c8f2n07.gpfs.net:50070</value>
</property>

<property>
<!--Shared directory used for status sync up-->
<name>dfs.namenode.shared.edits.dir</name>
<value>/<gpfs.mnt.dir>/<gpfs.data.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 configuration **dfs.namenode.shared.edits.dir** must be consistent with `gpfs.mnt.dir` and `gpfs.data.dir` defined in `/usr/lpp/mmfs/hadoop/etc/hadoop/gpfs-site.xml`. You could create the directory `<gpfs.mnt.dir>/<gpfs.data.dir>/HA` and change the ownership to `hdfs:hadoop` before starting the HDFS transparency services.

The **dfs.ha.standby.checkpoints** must be set as `false`. If not, you will see a log of exceptions in the standby NameNode logs. 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.
```

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 33, 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, see Sync Hadoop configurations.
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.

5. On `HDFS_Transparency_node1`, run the following command as the root user to synchronize HDFS Transparency configuration with all HDFS transparency nodes:

```
mmhadoopctl connector synconf /usr/lpp/mmfs/hadoop/etc/hadoop
```

6. Start the HDFS Transparency service by running the **mmhadoopctl** command:

```
mmhadoopctl connector start
```

7. Format the zookeeper data structure:

```
/usr/lpp/mmfs/hadoop/bin/gpfs --config /usr/lpp/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.

8. Start the zkfc daemon:

```
/usr/lpp/mmfs/hadoop/sbin/hadoop-daemon.sh start zkfc -formatZK
```

Run `jps` on the `nn1` and `nn2` name nodes 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`

9. Check the state of the services

Run the following command to check that all NameNode services and DataNode services are up:

```
# mmhadoopctl connector getstate
```

10. Run the following command to check the state of NameNode services:

```
/usr/lpp/mmfs/hadoop/bin/gpfs haadmin -getServiceState [name node ID]
```

You could check your `/usr/lpp/mmfs/hadoop/etc/hadoop/hdfs-site.xml` or run the following commands to figure out the `[name node 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 co-located 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.

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 Spectrum 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 Spectrum 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 Spectrum 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

Short-circuit Read configurations in this section is only applicable to Apache Hadoop 2.7.1+. For Apache Hadoop, you could take the following steps to enable it. For HortonWorks HDP, you could enable/disable short circuit read from the Ambari GUI. Perform the following steps to enable the DFS client and ensure that `hdfs-site.xml` is configured with the correct **dfs.client.read.shortcircuit** and **dfs.domain.socket.path** values.

Configuring short-circuit local read:

Note: For configuring short-circuit local read, glibc version must be at least version 2.14.

To configure short-circuit local reads, enable `libhadoop.so` and use the DFS client shipped by the IBM Spectrum 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 Spectrum 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. Back up the **hadoop-hdfs-2.7.3.jar** by running `$ mv $YOUR_HADOOP_PREFIX/share/hadoop/hdfs/hadoop-hdfs-2.7.3.jar $YOUR_HADOOP_PREFIX/share/hadoop/hdfs/hadoop-hdfs-2.7.3.jar.backup`
2. Link **hadoop-gpfs-2.7.2.jar** to classpath using `$ ln -s /usr/lpp/mmfs/hadoop/share/hadoop/hdfs/hadoop-gpfs-2.7.3.jar $YOUR_HADOOP_PREFIX/share/hadoop/hdfs/hadoop-gpfs-2.7.3.jar`
3. Update the `core-site.xml` file with the following information:

```
<property>
  <name>fs.hdfs.impl</name>
  <value>org.apache.hadoop.hdfs.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 the changes in the entire cluster and restart the Hadoop cluster to ensure that all services are aware of this configuration change. Restart the HDFS Transparency cluster or follow the section “Automatic Configuration Refresh” on page 81 to refresh the configuration without interrupting the HDFS Transparency 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.

Note: For Apache Hadoop, if you run other components, such as Oozie, Solr, Spark, these components will be packaged as `hadoop-hdfs-<version>.jar` into their packages. When enabling short circuit write, we need to re-package them with the `hadoop-hdfs-2.7.3.jar` from HDFS Transparency. When disabling short circuit write, we need to re-package them with the `hadoop-hdfs-2.7.3.jar` from Apache Hadoop.

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 Spectrum 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.

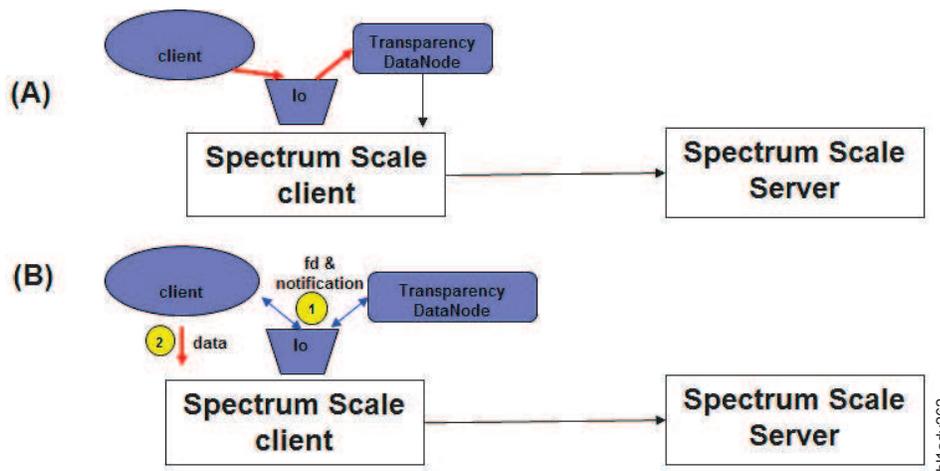


Figure 7. Short Circuit Write Logic

In Figure 7, (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 Spectrum Scale file system.

If you want to enable this feature, refer “Short-circuit read configuration” on page 33 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 Spectrum Scale service from Ambari.

If you take open source Apache Hadoop, change this in /usr/lpp/mmfs/hadoop/etc/hadoop/gpfs-site.xml and run /usr/lpp/mmfs/bin/mmhadoopctl connector syncconf /usr/lpp/mmfs/hadoop/etc/hadoop 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 Spectrum Scale file system. For each Hadoop cluster, you need one HDFS transparency cluster to provide the filesystem service.

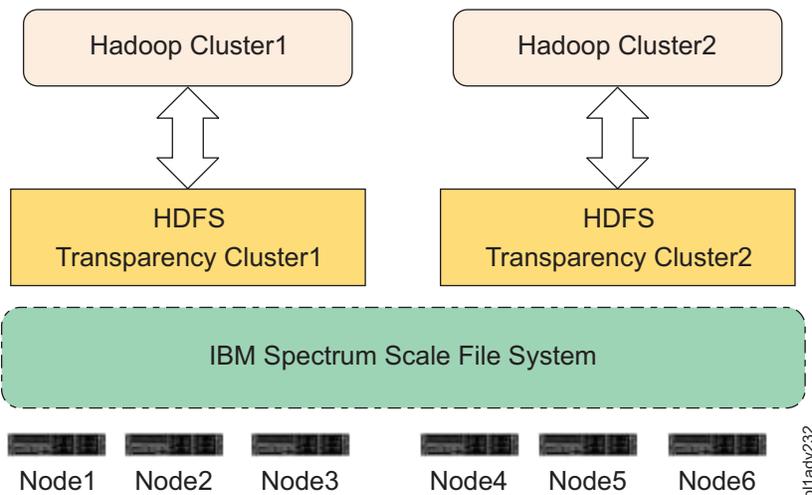


Figure 8. Two Hadoop Clusters over the same IBM Spectrum Scale file system

You can configure Node1 to Node6 as an IBM Spectrum 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:

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** 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** to synchronize the gpfs-site.xml from Step3 to all other nodes in HDFS transparency cluster2.
5. Restart the HDFS transparency services.

Docker support

HDFS transparency supports running the Hadoop Map/Reduce workload inside Docker containers.

See the Docker website for Docker technology.

One HDFS Transparency cluster on a set of physical nodes

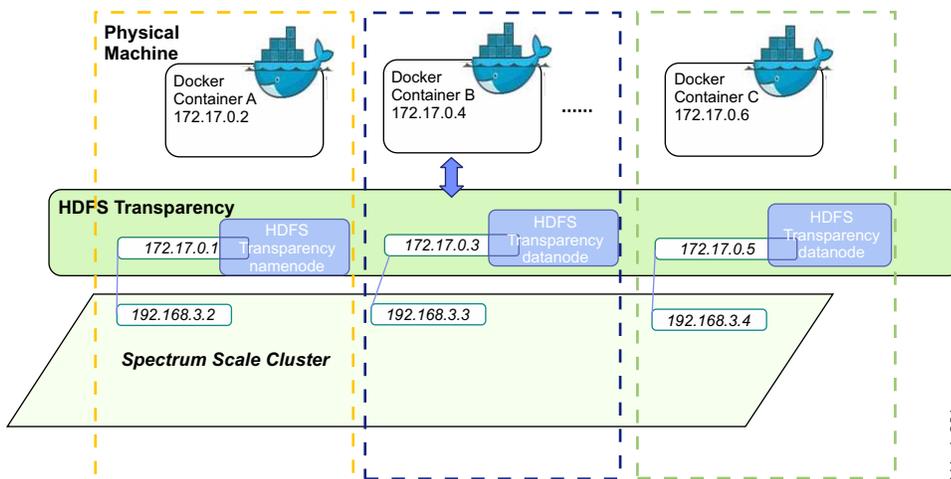
HDFS Transparency must be a Spectrum Scale node (Either a Spectrum Scale client or a Spectrum Scale server).

It is recommended that one physical node only belongs to one HDFS Transparency cluster. This section explains the steps to configure a set of physical nodes as HDFS Transparency cluster and this HDFS Transparency cluster will provide the data access for Hadoop running inside containers. If you have multiple Hadoop clusters running in different containers, you have to configure one HDFS Transparency cluster for each Hadoop cluster to isolate the data between the different Hadoop clusters. For example:

Physical node 1/2/3 configured as HDFS Transparency cluster1 for Hadoop cluster1.

Physical node 4/5/6 configured as HDFS Transparency cluster2 for Hadoop cluster2.

With HDFS transparency, you can run Hadoop Map/Reduce jobs in Docker and take IBM Spectrum Scale as the uniform data storage layer over the physical machines.



You can configure different Docker instances from different physical machines as one Hadoop cluster and run Map/Reduce jobs on the virtual Hadoop clusters. All Hadoop data is stored in the IBM Spectrum Scale file system over the physical machines. The 172.17.0.x IP address over each physical machine is a network bridge adapter used for network communication among Docker instances from different physical machines. HDFS transparency services must be configured to monitor the network bridge and process the requests from Docker instances. After receiving the requests from Hadoop jobs running in Docker instances, HDFS transparency handles the I/O requests for the mounted IBM Spectrum Scale file system on the node.

Configuring the Docker instance and HDFS transparency:

This topic provides information to configure the docker instance and HDFS transparency.

1. Docker (version 1.9+) requires Redhat7+. Modify the Redhat Yum Repos to upgrade the selinux-policy and device-mapper-libs by running the following commands:
 - # yum upgrade selinux-policy
 - # yum upgrade device-mapper-libs

2. To install Docker engine (version 1.9+), see Get Docker for Red Hat Enterprise Linux.
3. Configure the network bridge adapter on physical machines. There can be only one network bridge adapter on one machine.

Note: These configurations must be changed under `/etc/sysconfig/network-scripts/`:

```
[root@c3m3n04 network-scripts]# cat ifcfg-br0
DEVICE=br0
TYPE=Bridge
BOOTPROTO=static
IPADDR=172.17.0.1
NETMASK=255.255.255.0
ONBOOT=yes

[root@c3m3n04 network-scripts]# cat ifcfg-enp11s0f0
# Generated by dracut initrd
DEVICE="enp11s0f0"
ONBOOT=yes
NETBOOT=yes
UUID="ca481ab0-4cdf-482e-b5d3-82be13a7621c"
IPV6INIT=yes
BOOTPROTO=static
HWADDR="e4:1f:13:be:5c:28"
TYPE=Ethernet
NAME="enp11s0f0"
IPADDR=192.168.3.2
BROADCAST=192.168.255.255
NETMASK=255.255.255.0
```

Note: You must modify the IPADDR, BROADCAST, and NETMASK according to your network configuration.

In this example, the br0 bridge adapter is bundled with the enp11s0f0 physical adapter. You must modify the above configuration for all the physical machines on which the Docker instances must be run.

4. Modify the Docker service script and start the Docker engine daemons on each node:

```
# vim /usr/lib/systemd/system/docker.service
ExecStart=/usr/bin/docker daemon -b br0 -H fd://

# service docker stop
3 service docker start
```

5. Configure the network route table on each physical machine:

```
route add -net 172.17.1.0/24 gw <replace-physical-node-ip-here> dev enp11s0f0
where <replace-physical-node-ip-here> is the IP address of your machine.
```

6. The IP addresses of the nodes must be different so that the Docker instances from one physical node can access the Docker instances in another physical node. Check if you can connect to the br0 IP address from another node. If you cannot, you have problems in the network configuration and you need to fix them first.
7. Configure HDFS transparency and start the HDFS transparency services. Modify `/usr/lpp/mmfs/hadoop/etc/hadoop/core-site.xml` and `/usr/lpp/mmfs/hadoop/etc/hadoop/slaves`. You must select the IP address from Docker network bridge adapter. Pull the Hadoop Docker image on each node:

```
# docker pull sequenceiq/hadoop-docker:2.7.0
```

Note: We have selected the Hadoop Docker image from sequenceiq.

8. Start all Docker instances on each physical node by running the following command:

```
# docker run -h <this-docker-instance-hostname> -it
sequenceiq/hadoop-docker:2.7.0 /etc/bootstrap.sh -bash
```

You can start multiple Docker instances over the same physical node. This command starts a Docker instance with the hostname *<this-docker-instance-hostname>*.

9. For each Docker instance, change the `/etc/hosts` to map the Docker instance IP addresses to the hostname:

```
#vi /etc/hosts
172.17.0.2 node1docker1.gpfs.net node1docker1
172.17.0.4 node2docker1.gpfs.net node2docker1
172.17.0.6 node3docker1.gpfs.net node3docker1
```

Note: This must be done on the console of each Docker instance. You must add all Docker instances here if you want to set them up as one Hadoop cluster.

After a Docker instance is stopped, all changes are lost and you will have to make this change again after a new Docker instance has been started.

10. Select a Docker instance and start the Yarn ResourceManager on it:

```
#cd /usr/local/hadoop-2.7.0/sbin ./start-yarn.sh
```

You cannot run two ResourceManagers in the same Hadoop cluster. Therefore, you run this ResourceManager in the selected Docker instance.

11. Start Yarn NodeManager on other Docker instances by running the following command:

```
#!/usr/local/hadoop-2.7.0/sbin/yarn-daemon.sh --config /usr/local/hadoop/etc/hadoop/
start nodemanager
```

12. Run `hadoop dfs -ls /` to check if you can run Map/Reduce jobs in Docker now. To stop the Yarn services running in Docker, perform the following steps:

```
->on Yarn ResourceManager Docker instance:
cd /usr/local/hadoop-2.7.0/sbin ./stop-yarn.sh
->on Yarn NodeManager Docker instances:
/usr/local/hadoop-2.7.0/sbin/yarn-daemon.sh --config /usr/local/hadoop/etc/hadoop/
stop nodemanager
```

Note: While selecting HDFS transparency with Docker instances, data locality is not supported for the Map/Reduce jobs.

Multiple HDFS Transparency clusters on the same set of physical nodes

If you have limited physical nodes or you have too many Hadoop clusters running inside containers, then you might have to set up multiple HDFS Transparency clusters on the same set of physical nodes.

For example, configure HDFS Transparency cluster1 on the physical node 1/2/3 and configure HDFS Transparency cluster2 on the same physical node 1/2/3. This is supported since HDFS Transparency version 2.7.3-1.

Running multiple HDFS Transparency clusters on the same set of physical nodes will require configuration changes, especially network port number assigned to different HDFS Transparency clusters. This section will explain the steps to configure two HDFS Transparency clusters.

In the following example, it will configure two HDFS Transparency clusters on the same physical nodes `gpfstest1/2/6/7/9/10/11/12` with `gpfstest2` as the NameNode. In this environment, Kerberos is not enabled and the configurations for 1st HDFS Transparency cluster is from the HortonWorks HDP cluster.

1. Configure the `/usr/lpp/mmfs/hadoop/etc/hadoop` to bring the first HDFS Transparency cluster up. The `gpfstest1/2/6/7/9/10/11/12` is configured as the HDFS transparency cluster1:

```
[root@gpfstest2 ~]# mmhadoopctl connector getstate
gpfstest2.cn.ibm.com: namenode running as process 6699.
gpfstest2.cn.ibm.com: datanode running as process 8425.
gpfstest9.cn.ibm.com: datanode running as process 13103.
gpfstest7.cn.ibm.com: datanode running as process 9980.
gpfstest10.cn.ibm.com: datanode running as process 6420.
gpfstest11.cn.ibm.com: datanode running as process 83753.
```

gpfstest1.cn.ibm.com: datanode running as process 22498.
gpfstest12.cn.ibm.com: datanode running as process 52927.
gpfstest6.cn.ibm.com: datanode running as process 48787.

Note: This setup will be configured by HortonWorks HDP through Ambari and the gpfstest2 is configured as the NameNode.

2. Select any one node from these nodes, and change the configurations:

In this example, the gpfstest1 node is selected.

Step 3 to step 10 are done on the gpfstest1 node as the node selected in step 2.

3. Copy the following configurations from /usr/lpp/mmfs/hadoop/etc/hadoop to /usr/lpp/mmfs/hadoop/etc/hadoop2.

Note: /usr/lpp/mmfs/hadoop/etc/Hadoop is the configuration location for the 1st HDFS Transparency cluster and /usr/lpp/mmfs/hadoop/etc/hadoop2 is the configuration location for the 2nd HDFS Transparency cluster.

```
-rw-r--r-- 1 root root 2187 Oct 28 00:00 core-site.xml
-rw----- 1 root root 393 Oct 28 00:00 gpfs-site.xml
-rw----- 1 root root 6520 Oct 28 00:00 hadoop-env.sh
-rw----- 1 root root 2295 Oct 28 00:00 hadoop-metrics2.properties
-rw----- 1 root root 2490 Oct 28 00:00 hadoop-metrics.properties
-rw----- 1 root root 1308 Oct 28 00:00 hadoop-policy.xml
-rw----- 1 root root 6742 Oct 28 00:00 hdfs-site.xml
-rw----- 1 root root 10449 Oct 28 00:00 log4j.properties
-rw----- 1 root root 172 Oct 28 00:00 slaves
-rw----- 1 root root 884 Oct 28 00:00 ssl-client.xml
-rw----- 1 root root 1000 Oct 28 00:00 ssl-server.xml
-rw-r--r-- 1 root root 17431 Oct 28 00:00 yarn-site.xml
```

4. Change the fs.defaultFS value in core-site.xml

```
In /usr/lpp/mmfs/hadoop/etc/hadoop/core-site.xml: fs.defaultFS=hdfs://
gpfstest2.cn.ibm.com:8020
```

```
In /usr/lpp/mmfs/hadoop/etc/hadoop2/core-site.xml: fs.defaultFS=hdfs://
gpfstest2.cn.ibm.com:8021
```

5. Change values in the hdfs-site.xml file:

```
In /usr/lpp/mmfs/hadoop/etc/hadoop/hdfs-site.xml:
```

```
dfs.datanode.address=0.0.0.0:50010
dfs.datanode.http.address=0.0.0.0:50075
dfs.datanode.https.address=0.0.0.0:50475
dfs.datanode.ipc.address=0.0.0.0:8010
dfs.https.port=50470
dfs.journalnode.http-address=0.0.0.0:8480
dfs.journalnode.https-address=0.0.0.0:8481
dfs.namenode.http-address=gpfstest2.cn.ibm.com:50070
dfs.namenode.https-address=gpfstest2.cn.ibm.com:50470
dfs.namenode.rpc-address=gpfstest2.cn.ibm.com:8020
dfs.namenode.secondary.http-address=gpfstest10.cn.ibm.com:50090
```

```
In /usr/lpp/mmfs/hadoop/etc/hadoop2/hdfs-site.xml:
```

```
dfs.datanode.address=0.0.0.0:50011
dfs.datanode.http.address=0.0.0.0:50076
dfs.datanode.https.address=0.0.0.0:50476
dfs.datanode.ipc.address=0.0.0.0:8011
dfs.https.port=50471
dfs.journalnode.http-address=0.0.0.0:8482
dfs.journalnode.https-address=0.0.0.0:8483
dfs.namenode.http-address=gpfstest2.cn.ibm.com:50071
dfs.namenode.https-address=gpfstest2.cn.ibm.com:50471
dfs.namenode.rpc-address=gpfstest2.cn.ibm.com:8021 <== match the port number in step4
dfs.namenode.secondary.http-address=gpfstest10.cn.ibm.com:50091
```

Note: Check that the network port numbers for the different HDFS Transparency clusters require to be different. If not, when starting the HDFS Transparency, it will report network port conflicts.

6. Change values in the `hadoop-env.sh` file.

```
In /usr/lpp/mmfs/hadoop/etc/hadoop/hadoop-env.sh:
```

```
HADOOP_PID_DIR=/var/run/hadoop/$USER
```

```
HADOOP_LOG_DIR=/var/log/hadoop/$USER
```

```
In /usr/lpp/mmfs/hadoop/etc/hadoop2/hadoop-env.sh:
```

```
HADOOP_PID_DIR=/var/run/hadoop/hdfstransparency2
```

```
HADOOP_LOG_DIR=/var/log/hadoop/hdfstransparency2
```

Change the `$USER` in `HADOOP_JOBTRACKER_OPTS`, `SHARED_HADOOP_NAMENODE_OPTS`, `HADOOP_DATANODE_OPTS` into value `"hdfstransparency2"`.

Note: HDFS Transparency can only be started as the root user. If the first HDFS Transparency cluster takes the `$USER` as `root`. The 2nd HDFS Transparency cluster, you need to change `$USER` into a different string value by setting it to `hdfstransparency2` to make the 2nd HDFS Transparency able to write logs there.

7. Change `hadoop-metrics2.properties`:

```
In /usr/lpp/mmfs/hadoop/etc/hadoop/hadoop-metrics2.properties:
```

```
namenode.sink.timeline.metric.rpc.client.port=8020
```

```
In /usr/lpp/mmfs/hadoop/etc/hadoop2/hadoop-metrics2.properties:
```

```
namenode.sink.timeline.metric.rpc.client.port=8021 <== match the namenode port number in step 4
```

8. Update `/usr/lpp/mmfs/hadoop/etc/hadoop2/gpfs-site.xml`, especially for the `gpfs.data.dir` field.

- Configure different `gpfs.data.dir` values for the different HDFS Transparency cluster.
- Configure different `gpfs.mnt.dir` values if you have multiple file systems.

9. Sync the 2nd transparency cluster configuration from node `gpfstest1` (selected in step2):

```
export HADOOP_GPFS_CONF_DIR=/usr/lpp/mmfs/hadoop/etc/hadoop2
```

```
mmhadoopctl connector syncconf /usr/lpp/mmfs/hadoop/etc/hadoop2
```

10. Start the end transparency cluster:

```
export HADOOP_GPFS_CONF_DIR=/usr/lpp/mmfs/hadoop/etc/hadoop2
```

```
export HADOOP_CONF_DIR=/usr/lpp/mmfs/hadoop/etc/hadoop2/
```

```
mmhadoopctl connector start
```

```
mmhadoopctl connector getstate:
```

```
[root@gpfstest1 hadoop2]# mmhadoopctl connector getstate
```

```
gpfstest2.cn.ibm.com: namenode running as process 18234.
```

```
gpfstest10.cn.ibm.com: datanode running as process 29104.
```

```
gpfstest11.cn.ibm.com: datanode running as process 72171.
```

```
gpfstest9.cn.ibm.com: datanode running as process 94872.
```

```
gpfstest7.cn.ibm.com: datanode running as process 28627.
```

```
gpfstest2.cn.ibm.com: datanode running as process 25777.
```

```
gpfstest6.cn.ibm.com: datanode running as process 30121.
```

```
gpfstest12.cn.ibm.com: datanode running as process 36116.
```

```
gpfstest1.cn.ibm.com: datanode running as process 21559.
```

11. Check the 2nd transparency cluster:

On any node, run the following commands:

```
hdfs --config /usr/lpp/mmfs/hadoop/etc/hadoop2 dfs -put /etc/passwd /
```

```
hdfs --config /usr/lpp/mmfs/hadoop/etc/hadoop2 dfs -ls /
```

12. Configure `hdfs://gpfstest2.cn.ibm.com:8020` and `hdfs://gpfstest2.cn.ibm.com:8021` to different Hadoop clusters running inside the container.

Hadoop Storage Tiering

IBM Spectrum Scale HDFS Transparency, also known as HDFS Protocol, offers a set of interfaces that allows applications to use HDFS clients to access IBM Spectrum Scale through HDFS RPC requests. For more information about HDFS Transparency, see Chapter 1, “IBM Spectrum Scale support for Hadoop,” on page 1 documentation.

Currently, if the jobs running on the native HDFS cluster plan to access data from IBM Spectrum Scale, the option is to use **distcp** or Hadoop Storage Tiering mode.

Using Hadoop **distcp** requires the data to be copied between the native HDFS and the IBM Spectrum 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 79.

If you are using Hadoop Storage Tiering, 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 “Open Source Apache viewfs support” on page 71.

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 Spectrum Scale architecture is shown in Figure 9 and Figure 10 on page 44:

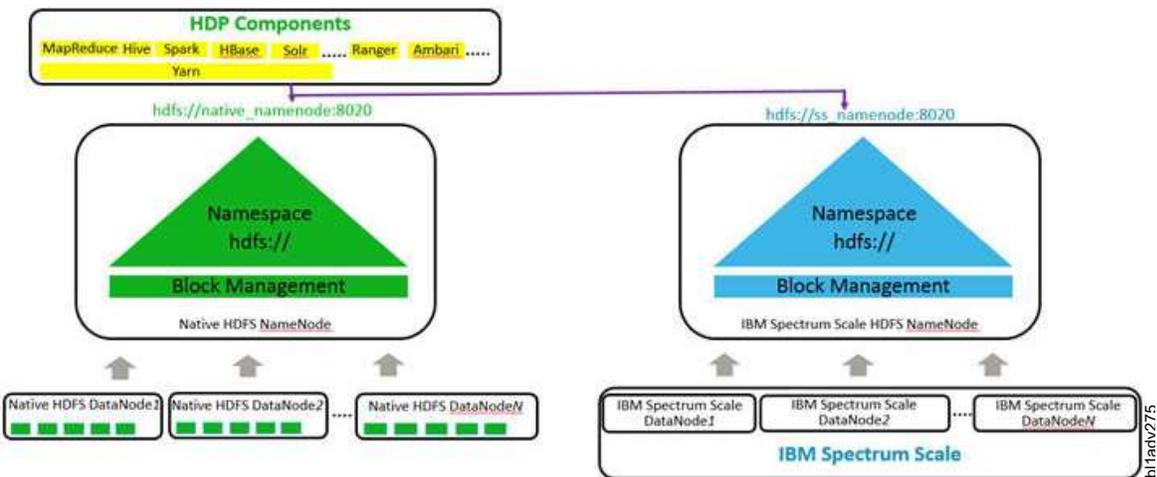


Figure 9. Hadoop Storage Tiering with IBM Spectrum Scale with single HDP cluster

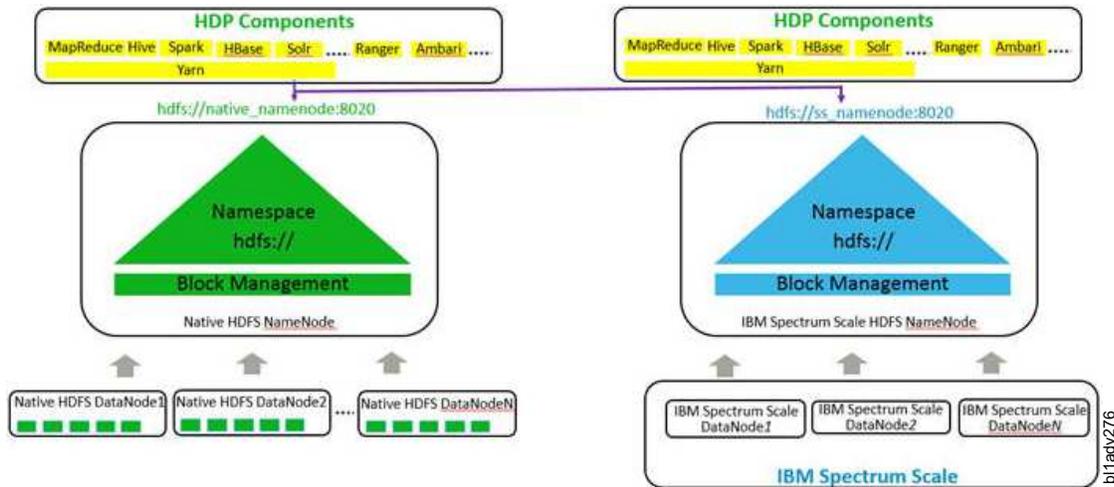


Figure 10. Hadoop Storage Tiering with IBM Spectrum Scale with 2 HPD clusters

The architecture for the Hadoop Storage Tiering has a native HDFS cluster (local cluster), seen on the left hand side, and a IBM Spectrum 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 Spectrum 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 Spectrum Scale HDFS Transparency cluster side is not a requirement for Hadoop Storage Tiering with IBM Spectrum Scale solution. This Hadoop cluster deployed on the IBM Spectrum Scale HDFS Transparency cluster side shows that a Hadoop cluster can access data via HDFS or POSIX from the IBM Spectrum 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-Utills	1.1.0.22
IBM Spectrum Scale & HDFS Transparency cluster	IBM Spectrum Scale	5.0.0
	HDFS Transparency	2.7.3-2
	IBM Spectrum 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 Spectrum Scale HDFS Transparency cluster, follow the steps listed below:

Option 1: IBM Spectrum Scale and HDFS Transparency cluster

This configuration is just for storage and does not have any Hadoop components.

1. Follow the “Installation and configuration of HDFS transparency” on page 14 to set up the IBM Spectrum Scale HDFS Transparency cluster.
2. Refer to “Enable Kerberos” on page 47 section to setup Kerberos and “Enable Ranger” on page 49 section to setup Ranger in a Hadoop Storage Tiering configuration.

Option 2: HDP with IBM Spectrum Scale and HDFS Transparency integrated cluster

1. Follow the “Installation” on page 107 topic to setup HDP and IBM Spectrum Scale HDFS Transparency cluster.
2. Refer to “Enable Kerberos” on page 47 section to setup Kerberos and “Enable Ranger” on page 49 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 Spectrum 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@cl6f1n07 ~]# 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@cl6f1n07 ~]# 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@cl6f1n07 ~]# 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 51 on how to test and leverage Hadoop Storage Tiering with IBM Spectrum Scale.

Enable Kerberos:

To enable Kerberos on the native HDFS cluster, the native HDFS cluster and the remote IBM Spectrum 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:

- Follow the “Setting up KDC server and enabling Kerberos” on page 169, 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 Spectrum Scale HDFS Transparency cluster.

For cluster without Ambari:

- 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 2c 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 name node 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 2c.
 - f. On the HDFS Transparency node selected in 2c, 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 Spectrum Scale service is integrated” on page 168 to enable Kerberos on IBM Spectrum Scale HDFS Transparency cluster.

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 name node, *c16f1n03.gpfs.net* is the remote HDFS Transparency cluster name node).

```
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/192.168.172.8:8050
18/03/05 22:29:27 INFO client.AHSPProxy: Connecting to Application History server at
c16f1n08.gpfs.net/192.168.172.8:10200
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: "c16f1n07/192.168.172.7";
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 Spectrum 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 Spectrum Scale HDFS Transparency cluster, see “Enabling Ranger” on page 161.

For cluster without Ambari:

- 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 name node 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`.
- d. Add the following to the `hadoop_env.sh` on the HDFS Transparency NameNode:


```
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
```
- 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 name node, *c16f1n03.gpfs.net* is the remote HDFS Transparency cluster name node).

- 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.q1.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$RangerAccessControlEnforcer.checkDefaultEnforcer
(RangerHdfsAuthorizer.java:428)
at org.apache.ranger.authorization.hadoop.RangerHdfsAuthorizer$RangerAccessControlEnforcer.
checkPermission(RangerHdfsAuthorizer.java:304)

```

- d. Log into 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 Spectrum Scale HDFS Transparency cluster
	3	Run the MapReduce WordCount job with input from the native HDFS and generate output to IBM Spectrum Scale HDFS Transparency cluster.
	4	Run the MapReduce WordCount job with input from the IBM Spectrum 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 Spectrum Scale HDFS Transparency cluster.
	3	Run the Spark LineCount/ WordCount job with input from the native HDFS and generate output to the IBM Spectrum Scale HDFS Transparency cluster.
	4	Run the Spark LineCount/ WordCount job with input from the IBM Spectrum 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 localinpath 2. INSERT into table 3. INSERT Overwrite TABLE	1	Drop remote database if EXISTS cascade.
	2	Create <i>remote_db</i> with Hive warehouse on the IBM Spectrum Scale HDFS Transparency cluster.
	3	Create internal nonpartitioned table on <i>remote_db</i> .
	4	LOAD data local inpath into table created in the above step.
	5	Create internal nonpartitioned table on the remote IBM Spectrum Scale HDFS Transparency cluster.
	6	LOAD data local inpath into table created in the above step.
	7	Create internal transactional table on <i>remote_db</i> .
	8	INSERT into table from internal nonpartitioned table.
	9	Create internal partitioned table on <i>remote_db</i> .
	10	INSERT OVERWRITE TABLE from internal nonpartitioned table.
	11	Create external nonpartitioned table on <i>remote_db</i> .
	12	Drop local database if EXISTS cascade.
	13	Create <i>local_db</i> with, Hive warehouse on local DAS Hadoop cluster.
	14	Create internal nonpartitioned table on <i>local_db</i> .
	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 <i>local_db</i> .
	19	INSERT into table from internal nonpartitioned table.
	20	Create internal partitioned table on <i>local_db</i> .
	21	INSERT OVERWRITE TABLE from internal nonpartitioned table.
	22	Create external nonpartitioned table on <i>local_db</i> .

Test case name	Step	Descriptions
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.
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.

Test case name	Step	Descriptions
JOIN tables between <i>local_db</i> and <i>remote_db</i>	1	JOIN <i>local_db</i> external nonpartitioned table with <i>remote_db</i> internal nonpartitioned table.
	2	JOIN <i>local_db</i> internal nonpartitioned table with <i>remote_db</i> internal nonpartitioned remote table.
	3	JOIN <i>local_db</i> internal nonpartitioned remote table with <i>remote_db</i> internal partitioned table.
	4	JOIN <i>local_db</i> internal partitioned table with <i>remote_db</i> internal transactional table.
	5	JOIN <i>local_db</i> internal transactional table with <i>remote_db</i> external nonpartitioned table.
Local temporary table from <i>remote_db</i> table	1	Create temporary table on <i>local_db</i> AS select query from <i>remote_db</i> table.
	2	Query data from temporary table.
IMPORT and EXPORT operations	1	EXPORT <i>local_db</i> internal partitioned table to the remote IBM Spectrum 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 <i>local_db</i> 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 <i>local_db</i> table created by IMPORT operation.
	6	EXPORT <i>remote_db</i> 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 <i>remote_db</i> 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 <i>remote_db</i> table created by preceding IMPORT operation.

Test case name	Step	Descriptions
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.

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/t101.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.sh to 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 Spectrum 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 Spectrum 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 `tpcd-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 Spectrum Scale HDFS Transparency cluster with <code>hadoop dfs</code> interface.
	4	Switch to hdp-user1 in Hadoop client node and query data from the local native HDFS cluster and the remote IBM Spectrum Scale HDFS Transparency cluster.
	5	Put local file <code>/etc/redhat-release</code> on HDP (local native HDFS) file system with <code>hadoop dfs -put</code> .
	6	Put local file <code>/etc/redhat-release</code> on IBM Spectrum Scale HDFS Transparency cluster with <code>hadoop dfs -put</code> .
	7	Run MapReduce WordCount job with input from HDP (local native HDFS) and generate output to IBM Spectrum Scale HDFS Transparency cluster.
	8	Run MapReduce WordCount job with input from IBM Spectrum 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 Spectrum Scale HDFS Transparency cluster.
	3	Create home directory and assign permission for hdp-user2 in the local native and the remote IBM Spectrum 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 Spectrum Scale HDFS Transparency cluster.
	6	Run MapReduce WordCount job with input from HDP (local native HDFS) and generate output to the IBM Spectrum Scale HDFS Transparency cluster.
	7	Run MapReduce WordCount job with input from IBM Spectrum 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 Spectrum Scale HDFS Transparency cluster.
	2	Create directory RESTRICT_ACCESS on remote IBM Spectrum Scale HDFS Transparency cluster.
	3	Create hdp-user1 on all the nodes of both the Hadoop cluster (HDP local HDFS) and IBM Spectrum 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 Spectrum Scale.
	2	Assign RWX on /user/hdp-user1 directory for hdp-user1 on HDP (local HDFS) and IBM Spectrum 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 Spectrum Scale HDFS Transparency cluster.
	5	Run MapReduce WordCount job with input from HDP (local HDFS), and generate output to the remote IBM Spectrum Scale HDFS Transparency cluster.
	6	Run MapReduce WordCount job with input from IBM Spectrum 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 Spectrum Scale HDFS Transparency cluster.
	3	Run Spark LineCount/WordCount job with input from primary HDP (local HDFS) and generate output to the IBM Spectrum Scale HDFS Transparency cluster.
	4	Run Spark LineCount/WordCount job with input from remote IBM Spectrum 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 Spectrum Scale HDFS Transparency cluster using Ranger UI.

Test case name	Step	Description
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 Spectrum 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 Spectrum 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 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.
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.

Test case name	Step	Description
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.
IMPORT and EXPORT operations	1	EXPORT local_db internal partitioned table to remote IBM Spectrum 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.

Test case name	Step	Description
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 Spectrum 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
```

FAQ:

1. ERROR: Requested user hdfs is banned while running MapReduce jobs as user *hdfs* in native HDFS cluster.

Solution:

<https://community.hortonworks.com/content/supportkb/150669/error-requested-user-hdfs-is-banned-while-running.html>

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:

Require to 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
```

- 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.q1.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 192.168.172.8)

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=*
```

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.q1.metadata.HiveException: Hive Runtime Error while
processing row {"cc_call_center_sk":1,"cc_call_center_id":"AAAAAAAAABAAAAAA","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":"AAAAAAAAABAAAAA","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/192.168.172.7"; 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.localtb11 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/localtb11
Table local_db_hdfs_ranger.localtb11 stats: [numFiles=1, numRows=84, totalSize=3004, rawDataSize=2920]
OK
Time taken: 6.584 seconds
```

```
hive> create table local_db_hdfs_ranger.localtb12 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/localtb12
Table local_db_hdfs_ranger.localtb12 stats: [numFiles=1, numRows=84, totalSize=3004, rawDataSize=2920]
OK
Time taken: 7.904 seconds
```

| **Open Source Apache views support**

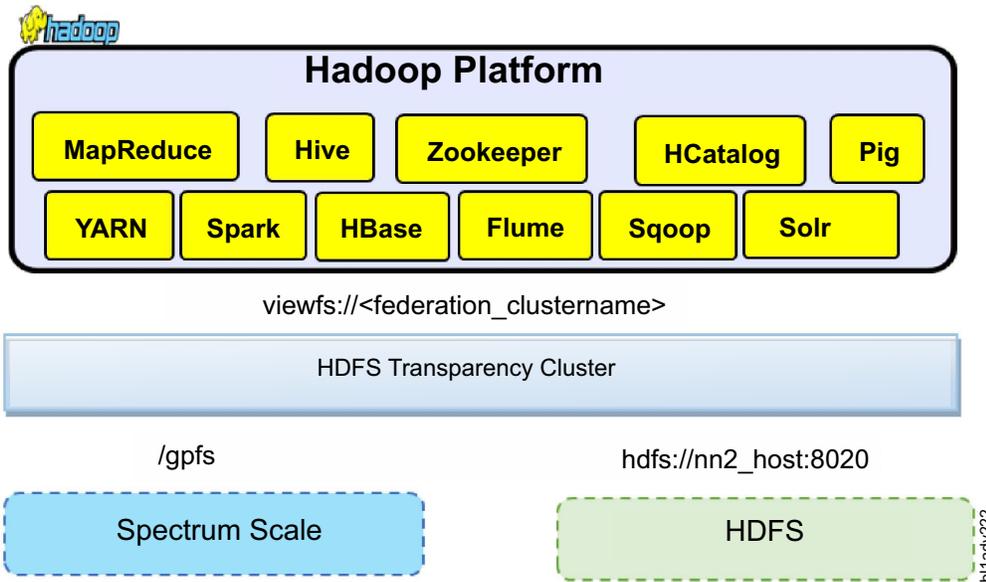
| Federation was added to HDFS to improve the HDFS NameNode horizontal scaling. In HDFS transparency, federation is used to co-exist IBM Spectrum Scale file systems and HDFS file system. The Hadoop applications can get input data from the native HDFS, analyze the input and write the output to the IBM Spectrum Scale file system. This feature is available in HDFS transparency version 2.7.0-2 (gpfs.hdfs-protocol-2.7.0-2) and higher.

| Also, the HDFS transparency federation can allow two or more IBM Spectrum Scale file systems to act as one uniform file system for Hadoop applications. These file systems can belong to the same cluster or be a part of different IBM Spectrum Scale clusters. For example, you need to read data from an existing file system, analyze it, and write the results to a new IBM Spectrum Scale file system.

| **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 views namespace between IBM Spectrum Scale and native HDFS:**

| Before configuring views support, ensure that you have configured the HDFS Transparency cluster (see "Hadoop cluster planning" on page 7 and "Installation and configuration of HDFS transparency" on page 14).



See the following sections to configure viewsfs for IBM Spectrum Scale and Native HDFS.

Single viewsfs namespace between IBM Spectrum Scale and native HDFS –Part I:

This topic describes the steps to get a single viewsfs namespace by joining native HDFS namespace with HDFS transparency namespace.

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`:

```
<configuration>
<property>
<name>fs.defaultFS</name>
<value>viewsfs://<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 `/usr/lpp/mmfs/hadoop/etc/hadoop/hdfs-site.xml`.

```
<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:

```
# mmhadoopctl connector syncconf /usr/lpp/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
```

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: `slaves`, `log4j.properties`, `hdfs-site.xml`, `hadoop-policy.xml`, `hadoop-metrics.properties`, `hadoop-metrics2.properties`, `core-site.xml`, and `gpfs-site.xml`.

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 Spectrum 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 are able to 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 client, verify that the viewfs directories are available by running the following command:

```
hadoop dfs -ls /
```

Single viewfs namespace between IBM Spectrum 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 Spectrum Scale and native HDFS –Part I” on page 72 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 viewsfs, 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 viewsfs 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 viewsfs 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 SecondaryNameNode, run `su - hdfs` to switch to the `hdfs` UID and run the following command to start SecondaryNameNode:

```
/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 viewsfs. 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 79.

If IBM BigInsights IOP is running, `core-site.xml` and `hdfs-site.xml` are located in `/etc/hadoop/conf/`.

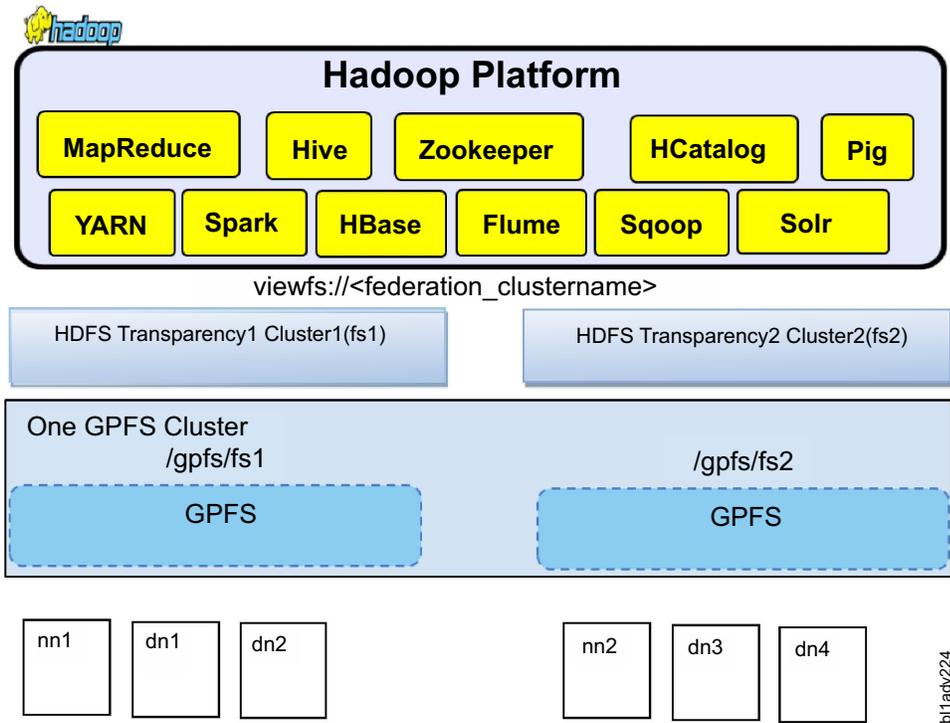
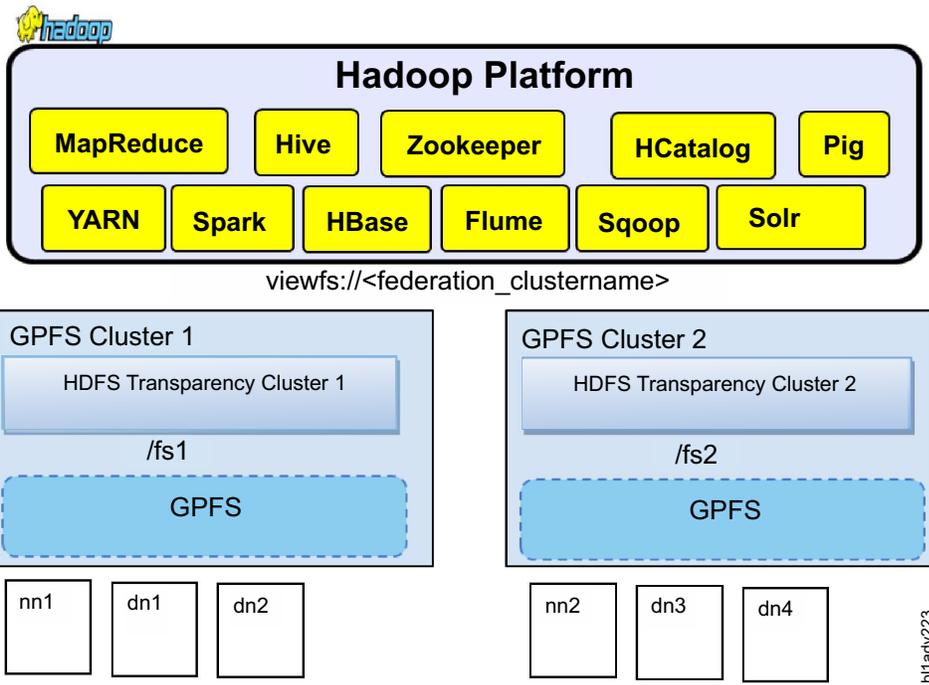
6. To ensure that the viewsfs file system is functioning correctly, run the following command:

```
hadoop fs -ls /
```

Single viewsfs namespace between two IBM Spectrum Scale file systems:

You can get a single viewsfs namespace joining two IBM Spectrum 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 Spectrum
- Scale file system (refer the “Hadoop cluster planning” on page 7 and “Installation and configuration of
- HDFS transparency” on page 14), 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 Spectrum Scale file systems.

| Before configuring the viewfs, see “Hadoop cluster planning” on page 7 and “Installation and configuration of HDFS transparency” on page 14 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`:

```
| <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*, synchronize the configuration change to another HDFS transparency node.

Note: You cannot take `mmhadoopctl connector syncconf /usr/lpp/mmfs/hadoop/etc/hadoop/` to synchronize the updated configurations because this might overwrite the configurations on the NameNode `nn2-host` in another cluster.

Take the following commands to synchronize the updated configurations from HDFS Transparency `nn1-host` with all the other nodes in the same HDFS Transparency cluster:

```
#login the HDFS Transparency Cluster1 NameNode nn1 as root:  
cd /usr/lpp/mmfs/Hadoop/etc/hadoop  
scp * <hostX>:/usr/lpp/mmfs/Hadoop/etc/hadoop/
```

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 Spectrum 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 Spectrum 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 79.

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 viewsfs support.

- All the changes in `/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 viewsfs, such as 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 viewsfs.

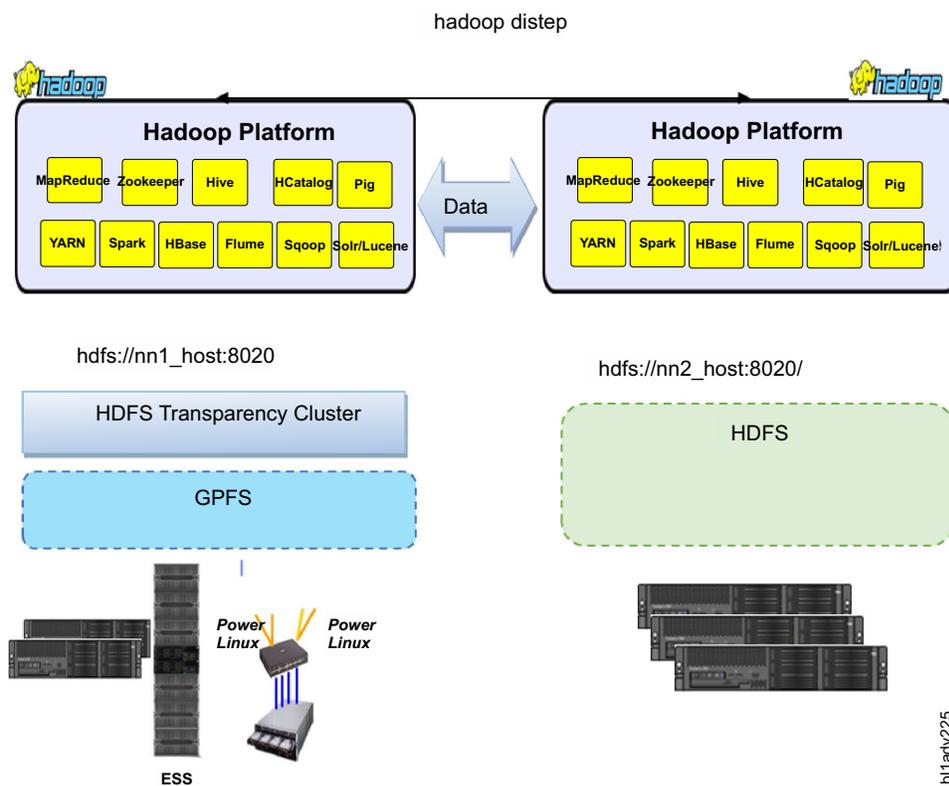
- The native HDFS and HDFS transparency cannot be run over the same node because of the network port number conflict.
- If you select to join both the native HDFS and HDFS transparency, configure the native HDFS cluster and make the native HDFS service function. Configure the viewsfs for native HDFS 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.

Hadoop distcp support

The `hadoop distcp` command is used for data migration from HDFS to the IBM Spectrum Scale file system and between two IBM Spectrum 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.

Issue 2: Access time exception while copying files from IBM Spectrum 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. See <https://issues.apache.org/jira/browse/HADOOP-11724> for more detail.

Note: This will not impact your data copy.

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` or in the IBM Spectrum 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.

Ranger support

HDFS authorization can use POSIX style permissions (also known as HDFS ACLs) or use Apache Ranger.

Apache Ranger (<http://hortonworks.com/hadoop/ranger/>) is a centralized security administration solution for Hadoop that enables administrators to create and enforce security policies for HDFS and other Hadoop platform components.

Ranger requires to be installed in native HDFS then configure for HDFS Transparency.

Installing Ranger in native HDFS

This section lists the ways in which you can install ranger in native HDFS.

There are three steps to install Ranger using IOP 4.2 and Ambari 2.2 User Interface (UI) :

- Configuring MySQL for Ranger (installation prerequisites)
- Installing Ranger
- Enabling the Ranger HDFS plugin

Configuring MySQL for Ranger:

Prepare the environment by configuring MySQL to be used for Ranger.

1. Create an IOP 4.2 Hadoop cluster, run the service check to ensure that the environment is running properly.
2. Configure MySQL for Ranger:
 - a. Create a non-root user to create the Ranger databases. In this example, the username `rangerdba` with password `rangerdba` is used.
 - 1) 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.

- 2) 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.

b. Check MySQL Java™ connector

- 1) 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
```

- 2) 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
```

c. Configure audit for Ranger

- 1) 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 *rangerlogger* user with password *YES* and grant the user adequate privileges:

```
CREATE USER 'rangerlogger'@'localhost' IDENTIFIED BY 'YES';
GRANT ALL PRIVILEGES ON *.* TO 'rangerlogger'@'localhost';
CREATE USER 'rangerlogger'@'%' IDENTIFIED BY 'YES';
GRANT ALL PRIVILEGES ON *.* TO 'rangerlogger'@'%' ;
GRANT ALL PRIVILEGES ON *.* TO 'rangerlogger'@'localhost' WITH GRANT OPTION;
GRANT ALL PRIVILEGES ON *.* TO 'rangerlogger'@'%' WITH GRANT OPTION;
FLUSH PRIVILEGES;
```

After setting the privileges, use the **exit** command to exit MySQL.

- 2) Reconnect to the database as user *rangerdba* by using the following command:

```
mysql -u rangerlogger -pYES
```

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**.



- On the Choose Services page, select **Ranger**.
The system displays the Ranger Requirements page.

Ranger Requirements

- You must have an **MySQL/Oracle/Postgres/MSSQL/SQL Anywhere Server** database instance running to be used by Ranger.
- In Assign Masters step of this wizard, you will be prompted to specify which host for the Ranger Admin. On that host, you **must have DB Client installed** for Ranger to access to the database. (Note: This is applicable for only Ranger 0.4.0)
- Ensure that the access for the DB Admin user is enabled in DB server from any host.
- Execute the following command on the Ambari Server host. Replace `database-type` with **mysql|oracle|postgres|mssql|sqlanywhere** and `/jdbc/driver/path` based on the location of corresponding JDBC driver:


```
ambari-server setup --jdbc-db={database-type} --jdbc-driver={/jdbc/driver/path}
```

Cancel Proceed

b11adm050

I have met all the requirements above.

b11adm051

Ensure that you have met all the installation requirements, then check the box for "I have met all the requirements above" before clicking **Proceed**.

- 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*.

Ranger Admin

DB FLAVOR

Ranger DB name

Ranger DB username

JDBC connect string

Ranger DB host

Driver class name for a JDBC Ranger database

Ranger DB password

bl1adm052

- For the Database Administrator (DBA) username, set the value to *rangerdba*.
- For the Database Administrator (DBA) password, set the value to *rangerdba*.
- Click on the Test Connection button and ensure that the connection result is OK.

Database Administrator (DBA) username

JDBC connect string for root user

Database Administrator (DBA) password

Connection OK

bl1adm053

- For IOP 4.1/4.2, set the Ranger Audit DB username value to *rangerlogger*, and for the Ranger Audit DB password, set the value to *YES*.

Audit to DB

Audit to DB

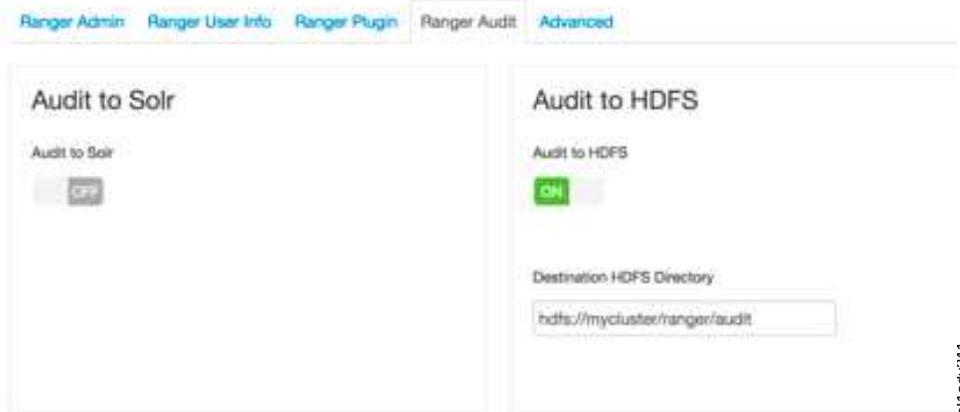
Ranger Audit DB username

Ranger Audit DB name

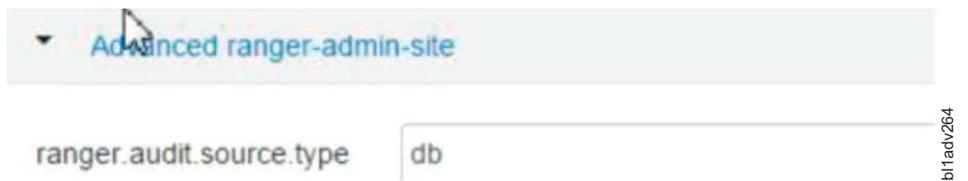
Ranger Audit DB password

bl1adv263

- For IOP 4.2.5/HortonWorks 2.6, set the Ranger Audit DB username value to *rangerlogger* and Ranger Audit DB password value to *YES*.



- In the Ranger Audit tab, ensure that the Audit to Solr option is disabled.
- Click **Advanced** tab > **Advanced ranger-admin-site** and set the value of `ranger.audit.source.type` to `db`.



5. 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**.
6. Test the connection.
 - On the Ranger dashboard, go to **Configs** > **Ranger Admin** > **Test connection**.

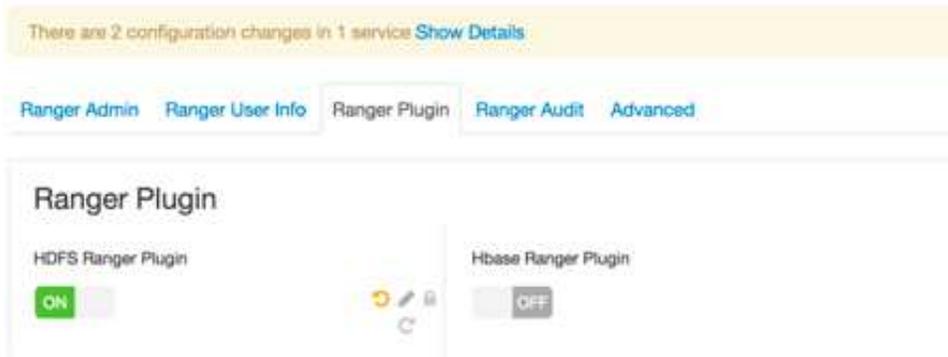


Note: After you install ranger, enable the ranger hdfs plugin and restart HDFS.

Enabling Ranger HDFS plug-in:

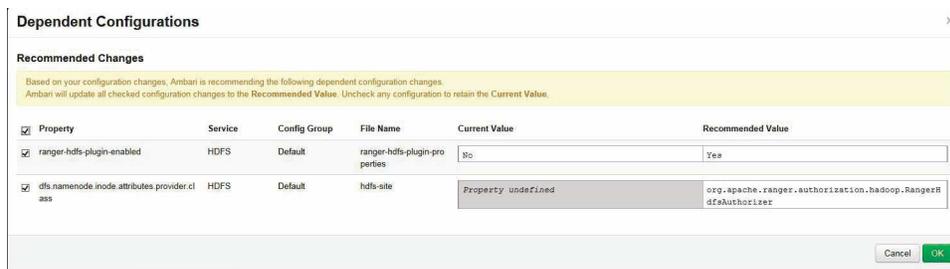
This topic lists the steps to enable Ranger HDFS plug-in

1. From the dashboard, click **Ranger** > **Configs** > **Ranger Plugin**, and switch on the **HDFS Ranger Plugin**.



bl1adv242

You will get the following screen once you enable the HDFS Ranger Plugin. Click **Ok** to accept the recommended changes.



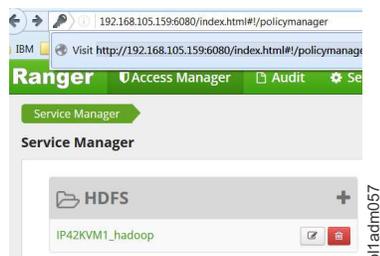
bl1adv274

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. 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



bl1adm057

Configuring Ranger with HDFS Transparency

Ranger configuration is based on the installation and configuration of HDFS Transparency. Therefore, HDFS transparency must be installed before configuring Ranger. To install HDFS transparency, see “Installation and configuration of HDFS transparency” on page 14.

Configuring Ranger

1. Check that `/etc/hadoop/conf/hdfs-site.xml` contains the value `org.apache.ranger.authorization.hadoop.RangerHdfsAuthorizer` for the **dfs.namenode.inode.attributes.provider.class**.

```
<property>
  <name>dfs.namenode.inode.attributes.provider.class</name>
  <value>org.apache.ranger.authorization.hadoop.RangerHdfsAuthorizer</value>
</property>
```

Synchronize `/usr/lpp/mmfs/hadoop/etc/hadoop/hdfs-site.xml` to all the NameNodes and DataNodes.

```
mmhadoopctl connector syncconf /etc/hadoop/conf/hdfs-site.xml
```

2. Copy the following four files to `/usr/lpp/mmfs/hadoop/etc/hadoop` on all the NameNode and DataNodes: `ranger-hdfs-audit.xml`, `ranger-hdfs-security.xml`, `ranger-policymgr-ssl.xml`, `ranger-security.xml` from the path `/etc/hadoop/conf`.
3. Edit the `/usr/lpp/mmfs/hadoop/etc/hadoop/hadoop-env.sh` on the NameNode and add these two classes to CLASSPATH:

For IOP 4.2:

```
/usr/iop/4.2.0.0/ranger-hdfs-plugin/lib/*.jar
/usr/share/java/mysql-connector-java.jar
for f in /usr/iop/4.2.0.0/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 IOP4.2.5:

Change the above version string 4.2.0.0 into "4.2.5.0-0000".

For HortonWorks 2.6:

```
/usr/hdp/4.2.0.0/ranger-hdfs-plugin/lib/*.jar
/usr/share/java/mysql-connector-java.jar
for f in /usr/hdp//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
```

4. Ensure that the DB service is running on the DB host node. Run the command `service mariadb restart` or `service mysqld restart` if the database service is not running.

```
[root@c8f2n03kvm2 lib]# service mysqld status
mysqld (pid 2774) is running...
```

On the Ranger DB Host node, ensure that the `rangerlogger` user exists.

```
mysql -u rangerlogger -pYES
```

Testing the Ranger policy for HDFS Transparency

1. Log in to the Ranger UI `http://<gateway>:6080` (admin/admin).

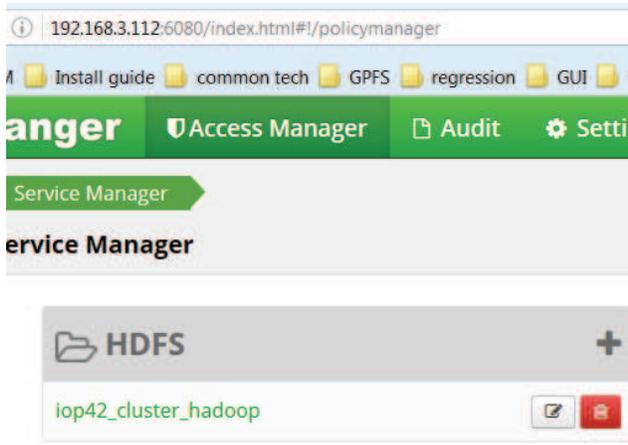


Figure 11. Service manager

2. Go to **Service Manager > iop42_cluster_hadoop > Add New Policy**.
3. Type the following values in the fields displayed on the Add New Policy page:

Label	Description
Policy Name	Type the policy name. This name is cannot be duplicated for the same Service type (HDFS). This field is mandatory.
Resource path	Define the resource path for folder/file. You can add wildcard characters like /home* to avoid writing the full path as well as to enable the policy for all sub folders and files.
Description	Type the description for the policy you are creating.
Recursive	Indicate if all files or folders within the existing folder are valid for the policy. Can be used instead of wildcard characters.
Audit Logging	Indicate if this policy will be audited.
Group Permissions	From a user group list, pick a particular group and choose permissions for that group.
Enable/disable	By default, the policy is enabled. You can disable a policy to restrict user/group access for that policy
User Permissions	From a user list, pick a particular user and choose permissions for that user.
Delegate Admin	When a policy is assigned to a user or a group of users, those users become the delegated admin. The delegated admin can update and delete the policies. It can also create child policies based on the original policy (base policy).

4. Set the policy.

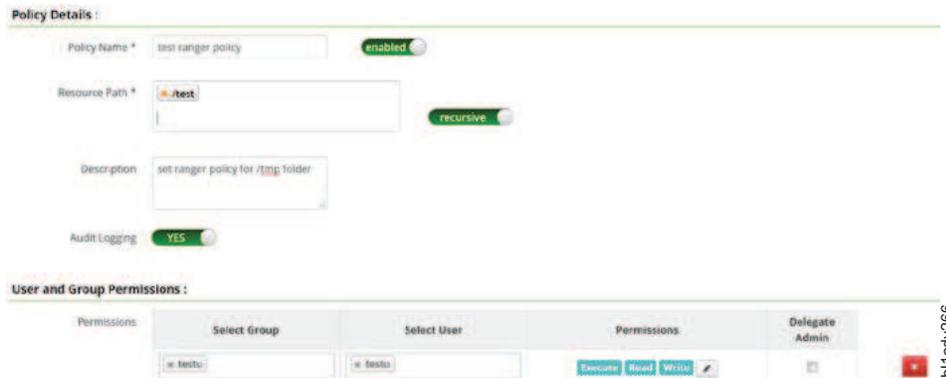


Figure 12. Policy details

5. Test if the user *testu* has the RWX access for path or test.

Using Ranger to secure HDFS

Apache Ranger offers a federated authorization model for HDFS.

Note:

1. The Ranger plugin for HDFS checks for Ranger policies. If a policy exists, access is granted to the user.
2. If a policy does not exist in Ranger, Ranger defaults to the native permissions model in HDFS (POSIX or HDFS ACL).

After Apache Ranger and Hadoop have been installed, administrators must perform the following steps:

1. Change HDFS umask to 077 from 022. This will prevent any new files or folders to be accessed by anyone other than the owner. To change the umask, from the **HDFS dashboard > Configs tab > search for umask**, and change the value from 022 to 077.



2. Know which directory is managed by Ranger and which directory is managed by POSIX/HDFS/ACL. Let HDFS manage the permissions for the /tmp and the /user folders.
3. Do not configure a file to be controlled by both Ranger and POSIX/HDFS/ACL permissions. This creates confusion in permission control.
4. Do not deny permission to the owner if the file is controlled by Ranger.

Note:

- a. Root is super user in GPFS mode.
- b. If you want to do some operation (such as delete) to one file, ensure that you have the corresponding access (wx) to its parent directory.
- c. When one user (such as u) deletes one file or file folder, ensure that /user/u exists.

```
drwx----- - u u 0 2016-09-21 04:05 /user/u
```

If not, you can create the /user/u manually and **chown u:u /user/u**.

Example

Root user creates one file, common user u wants to delete this file but without the w access, we can add this w access through adding a policy through the Ranger UI.

1. For /fycheng/hosts, u user just has r-x access and have no w access, and cannot delete the hosts.

```
[root@c8f2n04 hadoop]# hdfs dfs -ls -d /fycheng
drwxr-xr-x - root root 0 2016-10-12 23:29 /fycheng
```

```
[root@c8f2n04 hadoop]# hdfs dfs -ls /fycheng
```

Found 1 items

```
-rw-r--r-- 2 root root 158 2016-10-12 23:29 /fycheng/hosts
```

2. The u user wants to delete the /fycheng/hosts. In order to do this, follow these steps:

- Make sure that the /user/u exists.
- Check that u has the rwx access to /fycheng.
- Give correct policy access to /fycheng.
- As u user, do the delete operation to the files under /fycheng.

Here is a list of sequence based on the steps above:

Check if /user/u exists

```
[root@c8f2n04 hadoop]# su - u
Last login: Wed Oct 12 23:06:42 EDT 2016 on pts/1
```

```
[root@c8f2n04 hadoop]# hdfs dfs -ls /user/u
ls: ~/user/u': No such file or directory
```

The screenshot shows the Ranger UI configuration for a policy. The 'Policy Name' is 'u delete /fycheng/hosts', 'Resource Path' is '/fycheng', and 'Description' is 'u want to delete the file in /fycheng'. The 'Permissions' are set to 'wrm'. The 'Audit Logging' is set to 'YES'. Below the form, there is a section for 'User and Group Permissions' with a table:

Permissions	Select Group	Select User	Permissions	Delegate Admin
	Select Group	u	wrm	<input type="checkbox"/>

Delete operation to files under /fycheng.

Note: This command will fail.

```
[u@c8f2n04 ~]$ hdfs dfs -rmr /fycheng
rmr: DEPRECATED: Please use 'rm -r' instead.
16/10/12 23:07:22 INFO fs.TrashPolicyDefault: Namenode trash configuration:
Deletion interval = 360 minutes, Emptier interval = 0 minutes.
16/10/12 23:07:22 WARN fs.TrashPolicyDefault: Can't create trash directory:
hdfs://c8f2n04.gpfs.net:8020/user/u/.Trash/Current
org.apache.hadoop.security.AccessControlException: Permission denied: user=u,
access=WRITE, inode="/user/u/.Trash/Current":hdfs:hadoop:drwxr-xr-x
```

Create the /user/u manually, and chown u:u

```
[root@c8f2n04 hadoop]# hdfs dfs -ls -d /user/u /user/u
drwxr-xr-x - u u 0 2016-10-12 23:36 /user/u
```

Delete operation to the files under /fycheng will fail due to permission

```
[u@c8f2n04 ~]$ hdfs dfs -rmr /fycheng/hosts
rmr: DEPRECATED: Please use 'rm -r' instead.
16/10/12 23:38:02 INFO fs.TrashPolicyDefault: Namenode trash configuration:
Deletion interval = 360 minutes, Emptier interval = 0 minutes.
Rmr: Failed to move to trash: hdfs://c8f2n04.gpfs.net:8020/fycheng/hosts:
Permission denied: user=u, access=WRITE, inode="/fycheng/hosts":root:root:drwxr-xr-x
```

```
# Give correct policy access to /fycheng
In Ranger UI, add policy w access for u to /fycheng.
# Delete operation to files under /fycheng. Now the command will succeed.
[u@c8f2n04 ~]$ hdfs dfs -rmr /fycheng/hosts
rmr: DEPRECATED: Please use 'rm -r' instead.
16/10/12 23:42:48 INFO fs.TrashPolicyDefault: Namenode trash configuration:
Deletion interval = 360 minutes, Emptier interval = 0 minutes.
Moved: 'hdfs://c8f2n04.gpfs.net:8020/fycheng/hosts' to trash at:
hdfs://c8f2n04.gpfs.net:8020/user/u/.Trash/Current
```

Enabling Ranger auditing

This section lists the steps to enable ranger auditing.

For information on enabling and configuring Ranger auditing, see Enable Ranger Auditing.

Note:

1. In order to enable Audit to Solr for the Ranger plugins, ensure that the **xasecure.audit.destination.solr.zookeepers** field is set to `<host>:2181/solr`.
2. If you get the Unable to connect to the Audit store! message in the Ranger UI, see the FAQ Not able to view Solr audits in Ranger to remove the write locks from HDFS.

Disabling Ranger support

Ranger is supported by default since HDFS Transparency version 2.7.2-X. From HDFS Transparency version 2.7.2-1, Ranger support can be disabled by configuring the **gpfs.ranger.enabled** property field in the `/usr/lpp/mmfs/hadoop/etc/hadoop/gpfs-site.xml`.

To disable Ranger support, modify the `/usr/lpp/mmfs/hadoop/etc/hadoop/gpfs-site.xml` file on one of the HDFS transparency nodes to false:

```
<property>
  <name>gpfs.ranger.enabled</name>
  <value>>false</value>
  <description>Set false to disable Ranger mechanism</description>
</property>
```

Synchronize the modified `gpfs-site.xml` into all the other HDFS Transparency nodes and restart the HDFS Transparency. When Ranger support is in disabled mode, Ranger will not work over HDFS Transparency.

If you did not install or are not using the Apache Ranger over HDFS Transparency version 2.7.2-1+, set the `gpfs.ranger.enabled` field value to false to get better performance over HDFS Transparency.

Known issues

Directory permission issues might be hit when Ranger is enabled if the “Using Ranger to secure HDFS” on page 89 section was not followed during Ranger setup.

Table 5. Ranger directory permission issues

User name	Directory Permission in Ranger	directory permission on native HDFS	Results
fvuser (not super user, not the owner of the directory)	r--	--x	Expectation for user fvuser is to have r and x permission. However, the user has no permission to read the directory. To correct this issue, grant r-x access through the ranger UI or in HDFS.

Table 5. Ranger directory permission issues (continued)

User name	Directory Permission in Ranger	directory permission on native HDFS	Results
fvuser (not super user, not the owner of the directory)	--x	-w-	Expectation for user fvuser is to have x and w permission. However, the user has no permission to create file under the directory. To correct this issue, grant -wx access in the ranger UI or in HDFS.
fvuser (not super user, not the owner of the directory)	--x	r--	Expectation for user fvuser is to have x and r permission. However, the user has no permission to read the directory. To correct this issue, grant r-x access in the ranger UI or in HDFS.

Note: The issues in this table are for both native HDFS and HDFS Transparency.

- | **Restriction:** If the uid or gid value is larger than 8388607, Hadoop will report that the uid or gid is too large during the permission checking in Ranger before HDFS Transparency 2.7.3-3. This issue is fixed in HDFS Transparency 2.7.3-3.

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 Spectrum 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:

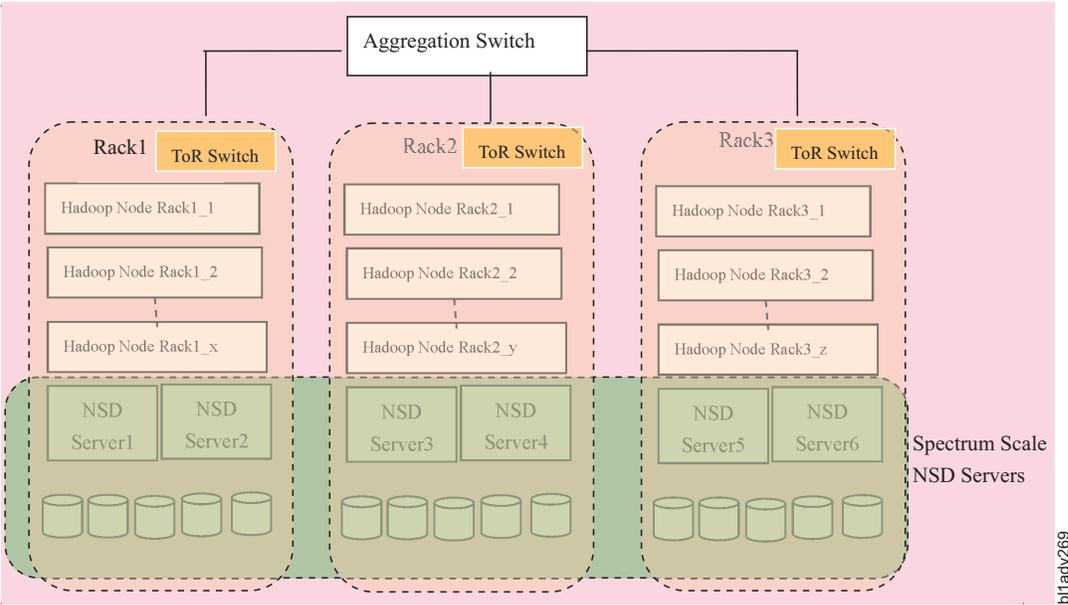


Figure 13. Topology of rack awareness locality for shared storage

For IBM Spectrum Scale over shared storage or IBM ESS, there is no data locality in the file system. The maximal file system block size from IBM Spectrum 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 Spectrum 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 13. 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 Spectrum 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`:

```
<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`:

```
# vim topology.data

192.168.200.57      /dc1/rack1
192.168.200.58      /dc1/rack1
192.168.80.129     /dc1/rack1
192.168.80.130     /dc1/rack1
192.168.172.6      /dc1/rack2
192.168.172.7      /dc1/rack2
192.168.172.8      /dc1/rack2
192.168.172.15     /dc1/rack2
```

Note: The topology.data file uses IP addresses. To configure two IP addresses, see the “Dual network interfaces” on page 13 section. The IP addresses here must be the IP addresses used for Yarn services and the IBM Spectrum Scale NSD server.

Also, it is required to specify the IP addresses for the IBM Spectrum™ scale NSD servers. For figure 8.9.1, specify the IP and corresponding rack information for NSD Server1/2/3/4/5/6.

4. On the HDFS Transparency NameNode, modify the `/usr/lpp/mmfs/hadoop/etc/hadoop/gpfs-site.xml`:

```
<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` 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 Spectrum Scale NSD servers and an IBM Spectrum 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 15 section.

Accumulo support

Special configuration on IBM Spectrum 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 Spectrum 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 `mm1spool <fs-name> all -L` command to check the value.

| **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 Spectrum 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

```

| Multiple Spectrum Scale File System support

This feature is available since HDFS Transparency version 2.7.3-1.

The federation feature from Hadoop supports federating two file systems into one logical file system. However, federation is not officially supported by HortonWorks HDP nor is it certificated with Hive in the Hadoop community. The multiple Spectrum 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` file.

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 Spectrum Scale file system support.

Note: Currently Multiple Spectrum Scale file system only supports 2 file systems.

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 Spectrum 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 `allowWriteAffinity` 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

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 Spectrum 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 Spectrum Scale file system with replica 1. If you are taking zero shuffle, the Map/Reduce jobs will store shuffled data into IBM Spectrum 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
<code>mapreduce.task.io.sort.mb</code>	<code><=1024</code>
<code>mapreduce.map.speculative</code>	<code>false</code>
<code>mapreduce.reduce.speculative</code>	<code>false</code>
<code>mapreduce.job.map.output.collector.class</code>	<code>org.apache.hadoop.mapred.SharedFsPlugins\$MapOutputBuffer</code>
<code>mapreduce.job.reduce.shuffle.consumer.plugin.class</code>	<code>org.apache.hadoop.mapred.SharedFsPlugins\$Shuffle</code>

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` 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.

Hadoop distribution support

Only the open source Apache packages, IBM BigInsights IOP 4.0/4.1/4.2 and HortonWorks HDP 2.6.x are officially supported. Cloudera is not supported.

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

Replacing native HDFS service with HDFS transparency

1. Install Hadoop distribution over native HDFS.

If you are using `host1/NameNode`, all other hosts are `DataNodes`.

2. Configure Hadoop configuration from your Hadoop distribution GUI.

Ensure that `dfs.client.read.shortcircuit` in HDFS service is disabled if you are using Hadoop 2.7.1/2.7.2. By default, it is disabled by most Hadoop distributions.

3. Set up an IBM Spectrum Scale cluster.

Note: `Host1` used for the HDFS `NameNode` service must be added to the IBM Spectrum Scale cluster.

4. Download and install the HDFS Transparency cluster according to the link.

All nodes in HDFS transparency cluster must be installed with the GPFS packages and must be able to mount the GPFS file system.

Configure the native HDFS `NameNode` as the default `NameNode` for HDFS Transparency. This will make the configuration changes simpler.

5. Set up the HDFS Transparency cluster.

- Copy the `/etc/hadoop/conf/core-site.xml`, `hdfs-site.xml` from the Hadoop distro `host1/NameNode` into `host1:/usr/lpp/mmfs/hadoop/etc/hadoop/`.
- Modify the `host1:/usr/lpp/mmfs/hadoop/etc/hadoop/slaves` to add the `DataNode` service for HDFS Transparency.

- Modify the `host1:/usr/lpp/mmfs/hadoop/etc/hadoop/gpfs-site.xml` according to the comments for each xml property.
 - On host1, run the command `/usr/lpp/mmfs/bin/mmhadoopctl connector syncconf /usr/lpp/mmfs/hadoop/etc/hadoop`.
6. On host1, run the `/usr/lpp/mmfs/bin/mmhadoopctl connector start` command to start the HDFS Transparency services.
 7. Run `hadoop dfs -ls /` to ensure that the system displays the correct output.
 8. Start the Hadoop distro services, hbase, yarn, and the other services.

Note: The native HDFS service must not be functioning in the Hadoop distribution GUI.

HortonWorks HDP 2.6.x support

HortonWorks HDP 2.6.x is verified over HDFS Transparency 2.7.3-x. Short Circuit Read can be enabled.

In HortonWorks HDP 2.6.x, some services, such as Hive, will take the user name “anonymous”(whose group name is “anonymous”) to do some service check. Therefore, we need to create the group and user in advance if they are not existing. The gid/uid of “anonymous” on all nodes should be of the same. For example,

```
# mmdsh -N all id anonymous
c35f1m4n15.gpfs.net: uid=2825(anonymous) gid=2825(anonymous) groups=2825(anonymous)
c35f1m4n14.gpfs.net: uid=2825(anonymous) gid=2825(anonymous) groups=2825(anonymous)
c35f1m4n13.gpfs.net: uid=2825(anonymous) gid=2825(anonymous) groups=2825(anonymous)
c35f1m4n16.gpfs.net: uid=2825(anonymous) gid=2825(anonymous) groups=2825(anonymous)
```

IBM BigInsights IOP support

This topic describes the support for IBM BigInsights IOP.

IBM BigInsights IOP 4.1.0.2 is verified over HDFS Transparency 2.7.0-x. Short Circuit Read is enabled by default and you must disable it on the Ambari GUI.

IBM BigInsights IOP 4.2 is verified over HDFS Transparency 2.7.2-x. Short Circuit Read can be enabled.

IBM BigInsights IOP 4.2.5 is verified over HDFS Transparency 2.7.3-x. Short Circuit Read can be enabled.

In IBM BigInsights IOP 4.2/4.2.5, some services, such as Hive, will take the user name *anonymous* (whose group name is *anonymous*) to do some service check. Therefore, we need to create the group and user in advance if they are not existing. The gid/uid of *anonymous* on all nodes should be of the same. For example

```
# mmdsh -N all id anonymous
c35f1m4n15.gpfs.net: uid=2825(anonymous) gid=2825(anonymous) groups=2825(anonymous)
c35f1m4n14.gpfs.net: uid=2825(anonymous) gid=2825(anonymous) groups=2825(anonymous)
c35f1m4n13.gpfs.net: uid=2825(anonymous) gid=2825(anonymous) groups=2825(anonymous)
c35f1m4n16.gpfs.net: uid=2825(anonymous) gid=2825(anonymous) groups=2825(anonymous)
```

Limitations and differences from native HDFS

This topic lists the limitations and the differences from the native HDFS.

Snapshot support

In native HDFS, it can create snapshot against one directory. Spectrum 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 Spectrum Scale fileset (check the column "Path" from the output of `mmfsfileset <fs-name>`) and if the fileset is an independent fileset, the `mmfsfileset` 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 Spectrum 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 Spectrum 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 Spectrum 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 Spectrum Scale level, only the root user and the owner of the linked directory of independent fileset can create snapshot for IBM Spectrum 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 Spectrum 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 Spectrum Scale File Systems.
- Snapshot for remote mounted file system is not supported if `gpfs.remotecuster.autorefresh (/usr/lpp/mmfs/hadoop/etc/hadoop/gpfs-site.xml)` is configured as `false`. By default, it is true.

Hadoop ACL and Spectrum Scale Protocol NFS/SMB

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 Spectrum Scale `-k` option as `all` or `posix`. If not, the ACL operations from Hadoop will report exceptions.

If you want to run Spectrum 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 Spectrum Scale Protocol SMB, SMB requires IBM Spectrum 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 Spectrum Scale file system. However, even with this workaround, you cannot take the SMB client to control the ACL of the files/directories from IBM Spectrum Scale. It is verified that the SMB ACL does not work properly over directories with the file system configured as `-k all`.

The difference between HDFS Transparency and native HDFS

The configuration that differ from HDFS in IBM Spectrum Scale.

Property name	Value	New definition or limitation
<code>dfs.permissions.enabled</code>	True/false	For HDFS protocol, permission check is always done.
<code>dfs.namenode.acls.enabled</code>	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 Spectrum Scale, HDFS protocol will not save the meta data. When ACL checking is on, the ACL will be set and stored in the IBM Spectrum Scale file system. If the admin turns ACL checking off, the ACL entries set before are still stored in IBM Spectrum Scale and remain effective. This will be improved in the next release.
<code>dfs.blocksize</code>	Long digital	Must be an integer multiple of the IBM Spectrum Scale file system blocksize (<code>mm1sfs -B</code>), the maximal value is <code>1024 * file-system-data-block-size (mm1sfs -B)</code>
<code>dfs.namenode.fs-limits.max-xattr-per-inode</code>	INT	Does not apply to HDFS Transparency.
<code>dfs.namenode.fs-limits.max-xattr-size</code>	INT	Does not apply to HDFS Transparency.

Property name	Value	New definition or limitation
dfs.namenode.fs-limits.max-component-length	Not checked	Does not apply to HDFS Transparency; the file name length is controlled by Spectrum Scale. Refer Spectrum Scale FAQ for file name length limit (255 unicode-8 chars)
Native HDFS encryption	Not supported	Customers should take native Spectrum Scale encryption.
Native HDFS caching	Not supported	Spectrum Scale has its own caching mechanism
NFS Gateway	Not supported	Spectrum Scale provides POSIX interface and taking Spectrum Scale protocol could give your better performance and scaling

Functional limitations

- The maximum number of Extended Attributes (EA) is limited by IBM Spectrum Scale and the total size of the EA key. Also, the 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. Also, if one file is smaller than inode size (by default, it is 4Kbytes per inode), Spectrum Scale will store the file as data-in-inode. For these kind 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 Spectrum Scale is 3.
- **hdfs fsck** does not work against HDFS Transparency. Instead, run **mmfsck**.
- Spectrum 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.

Problem determination

For known problem determination, see 2nd generation HDFS Protocol troubleshooting.

Chapter 2. BigInsights 4.2.5 and Hortonworks Data Platform 2.6

This section describes the deployment of IBM BigInsights™ 4.2.5 with Apache© Spark and Apache© Hadoop on the IBM Spectrum Scale™ file system or Hortonworks Data Platform (HDP®) 2.6 by using the Apache© Ambari framework.

Planning

Hardware requirements

- | This section specifies the hardware requirements to install IBM BigInsights Ambari IOP or 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 HDFS Transparency Guide.

Preparing the environment

This section describes how to prepare the environment to install IBM BigInsights Ambari IOP or Hortonworks Data Platform (HDP®), IBM Spectrum Scale Ambari management pack, and HDFS Transparency on IBM Spectrum Scale.

Note: The GPFS Ambari integration package is now called the IBM Spectrum Scale Ambari management pack (in short, management pack or MPack).

The IBM Spectrum Scale Ambari management pack can be used for both Ambari 2.4.2 for BigInsights, and Ambari 2.5 for Hortonworks to setup the IBM Spectrum Scale Service.

The IBM Spectrum Scale Ambari management pack installation process attempts to detect an existing IBM Spectrum Scale file system. For IBM Spectrum Scale FPO, which is a multi-node, just-a-bunch-of-disk/JBOD configuration, the installer can set up a new clustered file system if the hostnames of all the nodes and disk devices are available at each node by a stanza file. The installer designates manager roles and quorum nodes, and creates NSDs and the file system. The best practices for the Hadoop configuration are automatically implemented.

For installation on an existing file system, the Hadoop integration components for IBM Spectrum Scale are deployed. There will be no validation-checking done on the pre-existing IBM Spectrum Scale configuration.

See the current best practices for installation at [Big Data Best Practice wiki page](#).

The Hadoop distribution (HDP or BI) requires specific version combinations for the Mpack and HDFS Transparency. The IBM Spectrum Scale file system is independent of the versioning for the Mpack and HDFS Transparency. Hortonworks has been certified to work with IBM Spectrum Scale 4.2.3 and later.

Table 6. Hadoop distribution support matrix

IBM Spectrum Scale Ambari management pack (Mpack)	HDFS Transparency	Hadoop distribution	Ambari version	RHEL x86_64	RHEL ppc64le
Mpack 2.4.2.6	HDFS Transparency 2.7.3-1+	HDP 2.6.5	Ambari 2.6.2.0	7.2/7.3/7.4	7.2/7.3/7.4
Mpack 2.4.2.5	HDFS Transparency 2.7.3-1+	HDP 2.6.5	Ambari 2.6.2.0	7.2/7.3/7.4	7.2/7.3/7.4
Mpack 2.4.2.4	HDFS Transparency 2.7.3-1+	HDP 2.6.4	Ambari 2.6.1.0	7.2/7.3/7.4	7.2/7.3/7.4
Mpack 2.4.2.3	HDFS Transparency 2.7.3-1+	HDP 2.6.3	Ambari 2.6.0.0	7.2/7.3/7.4 ¹	7.2/7.3/7.4
Mpack 2.4.2.2	HDFS Transparency 2.7.3-1+	HDP 2.6.2	Ambari 2.5.2.0	7.2/7.3	7.2/7.3
Mpack 2.4.2.1	HDFS Transparency 2.7.3-1+	HDP 2.6.2	Ambari 2.5.2.0	7.2/7.3	7.2/7.3
Mpack 2.4.2.0	HDFS Transparency 2.7.3-0+	HDP 2.6.0 HDP 2.6.1	Ambari 2.5.0.3 Ambari 2.5.2.0	7.2/7.3	7.2/7.3
Mpack 2.4.2.1	HDFS Transparency 2.7.3-1+	BI 4.2.5 ²	Ambari 2.4.2.0	6.8/7.2/7.3	7.2/7.3
Mpack 2.4.2.0	HDFS Transparency 2.7.3-0+	BI 4.2.5	Ambari 2.4.2.0	6.8/7.2/7.3	7.2/7.3

Note:

1. HDP 2.6.3 with Scale is certified with x86_64 RH 7.4 even though the Hortonworks documentation currently might not be updated with this change.
2. Mpack 2.4.2.1 is the last supported release for BI 4.2.5
3. Additional support information:
 - 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+,5.0.0+ on pc64le and x86_64.
 - 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+.
 - Remote mount and Multiple file systems is only supported for HDP 2.6.2.
 - Mpack 2.4.2.1 supports migration from IOP to HDP.
 - The lzo compression package must be installed during the Ambari server setup before applying the IBM Spectrum Scale Mpack if you are using Mpack version 2.4.2.4. IBM Spectrum Scale service requires the lzo libraries when you are using Mpack version 2.4.2.4.

```

| # ambari-server setup
| ....
| Checking GPL software agreement...
| GPL License for LZO: https://www.gnu.org/licenses/old-licenses/gpl-2.0.en.html.
| Enable Ambari Server to download and install GPL Licensed LZO packages [y/n] (n)? y
| If lzo is selected as no, the IBM Spectrum Scale service installation will fail with get_lzo_packages
| error:
| File "/var/lib/ambari-agent/cache/common-services/HDFS/2.1.0.2.0/package/scripts/params_linux.py", line 46, in <mo
| from resource_management.libraries.functions.get_lzo_packagesimport get_lzo_packages
| ImportError: No module named get_lzo_packages

```

Review the Limitations section before starting the installation or upgrade procedures.

Ensure that you review the Hortonworks documentation for all system requirements.

For a list of the IBM Open Platform with Apache Spark and Apache Hadoop see the open source software version section in the BigInsights document.

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 Spectrum Scale Standard Configs panel.

- The sample files for GPFS policy, nsd, hadoop cache, rack configuration and share configuration file are located in `/var/lib/ambari-server/resources/mpacks/SpectrumScaleExtension-MPack-<version>/extensions/SpectrumScaleExtension/<version>/services/GPFS/package/templates` directory. The `<version>` is the version of the Mpack.
- Copy sample files to `/var/lib/ambari-server/resources`.

```

$ cd /var/lib/ambari-server/resources/mpacks/SpectrumScaleExtension-MPack-2.4.2.0/
extensions/SpectrumScaleExtension/2.4.2.0/services/GPFS/package/templates
$ ls
gpfs_fs.pol.sample gpfs_nsd.sample hadoop_cache.sample racks.sample shared_gpfs_node.cfg.sample
$ cp * /var/lib/ambari-server/resources

```

Note: Ambari for deploying a new IBM Spectrum Scale cluster is only supported for FPO mode.

Two types of NSD files are supported for file system auto-creation. One is the preferred simple format, and the other is the standard IBM Spectrum Scale NSD format intended for experienced IBM Spectrum Scale administrators.

Table 7. Preferred Simple Format and Standard Format

Preferred Simple Format	Standard Format
Ambari selects the correct metadata and data ratios.	The GPFS administrator is responsible for the storage arrangement and configuration.
If possible, Ambari creates partitions on some disks for Hadoop intermediate data to enhance performance.	A policy file is also required.
One system pool and one data pool are created.	Storage pools and block sizes can be defined as needed.
NSD file must be located at <code>/var/lib/ambari-server/resources/</code> on the Ambari server.	
Only <code>/dev/sdX</code> and <code>/dev/dx-X</code> devices are supported.	

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/sgd
DISK|compute002.private.dns.zone:/dev/sdb,/dev/sdc,/dev/sdd,/dev/sde,/dev/sdf,/dev/sgd
DISK|compute003.private.dns.zone:/dev/sdb,/dev/sdc,/dev/sdd,/dev/sde,/dev/sdf,/dev/sgd
DISK|compute005.private.dns.zone:/dev/sdb,/dev/sdc,/dev/sdd,/dev/sde,/dev/sdf,/dev/sgd
DISK|compute006.private.dns.zone:/dev/sdb,/dev/sdc,/dev/sdd,/dev/sde,/dev/sdf,/dev/sgd
DISK|compute007.private.dns.zone:/dev/sdb,/dev/sdc,/dev/sdd,/dev/sde,/dev/sdf,/dev/sgd
```

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

```

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'
```

Installation

Set up

This section gives the set up information for BigInsights® IOP or Hortonworks Data Platform (HDP) and IBM Spectrum Scale Hadoop integration support.

Base packages

The following packages must be installed on all IBM Spectrum Scale nodes:

```
$ yum -y install kernel-devel cpp gcc gcc-c++ binutils ksh
libstdc++ libstdc++-devel compat-libstdc++ imake make nc
```

For HDP: `libtirpc-devel` (From RH optional packages) and 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.

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.

BigInsights IOP (BI IOP)

This topic helps in the preparation to install BI IOP.

If installing the BigInsights IOP package:

Follow the BigInsights IOP with Apache Spark and Apache Hadoop documentation in the IBM Knowledge center for BI IOP installation preparation. For BigInsights overview, see Reference Architecture.

Setup prerequisites

- Preparing to install IBM Open Platform with Apache Spark and Apache Hadoop
 - Get ready to install
 - Preparing your environment
 - Obtaining software for the IBM Open Platform with Apache Spark and Apache Hadoop

IBM Open Platform (IOP) and BigInsights support installation by reading from the IBM-hosted yum repositories or the local mirror repositories. Reading from the local mirror repositories is faster for multi-node clusters because each node performs its own download of repository code. This document follows the procedure to create a local Ambari repository for installation.

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, 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 site for the HDP version you are using.

| **Note:** Ensure that the smartsense package is in a repository that is resolvable by Ambari during
| installation.

| The example in this document uses a local repository. For more information, see the *Using a local
| repository* section in the *Apache Ambari Installation* guide, Version 2.6.2, at the Hortonworks Documentation
| site

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.

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 IBM Spectrum Scale HDFS Transparency from HDFS Transparency Download.

- | Follow the Download and Merge process section in the HDFS Transparency developerWorks wiki to
- | combine the part 1 and part 2 of the HDFS Transparency rpm into a single HDFS Transparency rpm.

The module name is `gpfs.hdfs-protocol-2.7.3-(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.

IBM Spectrum Scale file system

This topic helps in the preparation to install IBM Spectrum Scale file system.

For IBM Spectrum Scale overview, see Product overview.

If you have purchased the IBM Spectrum Scale license, you can download the Spectrum Scale base installation package files from the IBM Passport Advantage® web site.

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 internal IBM users, customer POC, and trail licenses, follow the instructions on the Spectrum Scale Sales Wiki Software Evaluation - Spectrum Scale Trial license page.

To order IBM Spectrum Scale, see IBM Spectrum Scale Knowledge Center Question 1.1.

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:

This topic gives information about kernel.

- On all nodes, confirm that the output includes the following:

```
kernel-headers
kernel-devel
kernel
```

If any 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
```

- Check the installed kernel rpms. Unlike HDFS, IBM Spectrum 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 7.1.

Note: 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.

```
[root@c902f05x01 ~]# uname -r
3.10.0-327.el7.x86_64<== Find kernel-devel and kernel-headers to match this
```

```
[root@c902f05x01 ~]# rpm -qa | grep kernel
kernel-devel-3.10.0-327.el7.x86_64<== kernel-devel matches
kernel-3.10.0-327.el7.x86_64
```

```
kernel-tools-3.10.0-327.el7.x86_64
kernel-tools-libs-3.10.0-327.el7.x86_64
kernel-headers-3.10.0-327.el7.x86_64<== kernel-headers matches
```

SELinux and NTP:

This topic gives information about SELinux and NTP.

- SELinux must be in disabled mode.
- Network Time Protocol (NTP)

It is recommended that NTP be configured on all 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.

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
```

Network validation:

While using a private network for Hadoop data nodes, 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
| Spectrum Scale service is installed. It should be a part of the administrator nodes set. All the Spectrum
| Scale cluster wide administrative commands including those for creation of the 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 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: IBM Spectrum Scale Mpact Version 2.4.2.4 and later supports the admin mode central configuration of Spectrum Scale (adminMode configuration attribute).

In this configuration, one or more hosts could be designated as 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 (192.168.10.9)' 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.

- 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. If you plan to use quotas on the ESS storage, you need to either create the same users on the ESS cluster or setup ID mapping.
- 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 example:

```
groupadd --gid 1000 hadoop
groupadd --gid 1016 rddcached #optionally align rddcached GID with UID
groupadd --gid 10013 anonymous # Use for Hive

useradd -g hadoop -u 1001 ams
useradd -g hadoop -u 1002 hive
useradd -g hadoop -u 1003 oozie
useradd -g hadoop -u 1004 ambari-qa
useradd -g hadoop -u 1005 flume
useradd -g hadoop -u 1006 hdfs
useradd -g hadoop -u 1007 solr
useradd -g hadoop -u 1008 Knox
useradd -g hadoop -u 1009 spark
useradd -g hadoop -u 1010 mapred
useradd -g hadoop -u 1011 hbase
useradd -g hadoop -u 1012 zookeeper
useradd -g hadoop -u 1013 sqoop
useradd -g hadoop -u 1014 yarn
useradd -g hadoop -u 1015 hcat
useradd -g rddcached -u 1016 rddcached #optionally align rddcached GID with UID
useradd -g hadoop -u 1017 kafka
useradd -g anonymous -u 10013 anonymous # Use for Hive
```

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

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 starts and stops the HDFS Transparency Namenodes and Datanodes.

Download the management pack from the IBM Spectrum Scale wiki page for BI IOP 4.2.5 or HDP 2.6 support at the following link:

- BI 4.2.5 and HDP 2.6

From the download section, download the management pack into the Ambari server node as root.

For example:

- Create the /root/GPFS_Ambari directory.
- Unzip the packages there. This document uses the /root/GPFS_Ambari directory.

The management pack version 2.4.2.0 contains the following files:

- SpectrumScaleIntegrationPackageInstaller-2.4.2.0.bin
- SpectrumScaleMPackInstaller.py
- SpectrumScaleMPackUninstaller.py
- SpectrumScale_UpgradeIntegrationPackage-BI425 [For BigInsights only]

| The management pack version 2.4.2.1 contains the following files:

- | • SpectrumScaleIntegrationPackageInstaller-2.4.2.1.bin
- | • SpectrumScaleMPackInstaller.py

- | • SpectrumScaleMPackUninstaller.py
- | • SpectrumScale_UpgradeIntegrationPackage [Upgrade Spectrum Scale Mpack]

| The management pack version 2.4.2.3 contains the following files:

- | • SpectrumScaleIntegrationPackageInstaller-2.4.2.3.bin
- | • SpectrumScaleMPackInstaller.py
- | • SpectrumScaleMPackUninstaller.py
- | • SpectrumScale_UpgradeIntegrationPackage [Upgrade Spectrum Scale Mpack]

| The management pack version 2.4.2.4 contains the following files:

- | • SpectrumScaleIntegrationPackageInstaller-2.4.2.4.bin
- | • SpectrumScaleMPackInstaller.py
- | • SpectrumScaleMPackUninstaller.py
- | • SpectrumScale_UpgradeIntegrationPackage [Upgrade Spectrum Scale Mpack]

Note:

- | • Ensure all the packages reside in the same directory before executing the executables.
- | • For CentOS, create the /etc/redhat-release file to simulate a RedHat environment when you are using IBM Spectrum Scale Ambari Mpack version lower than 2.4.2.4. Otherwise, the IBM Spectrum Scale Ambari deployment fails. This workaround is no longer needed if you are using IBM Spectrum Scale Management pack 2.4.2.4 and later. See General section under the Limitations topic.

| **Installation of software stack**

| This section describes the installation and deployment of BI IOP or HDP and IBM Spectrum Scale Hadoop integration that consists of the management pack and HDFS Transparency connector.

| **Overview**

| This chapter shows the examples of installing BigInsights IOP 4.2.5 or HDP 2.6 with IBM Spectrum Scale Ambari management pack version 2.4.2.

Note:

- | • Before starting the software deployment, ensure all the Preparing the environment sections are reviewed and completed.
- | • Ensure that the correct management pack is used for installation for the specific Hadoop distribution.
- | • For Mpack 2.4.2.4 and earlier, if the node in the cluster contains a banner during ssh, the Ambari service advisor does not work properly when you deploy the IBM Spectrum Scale service. If the banner cannot be suppressed on the node, ensure that the IBM Spectrum Scale configuration, especially for an existing configuration, is correct. Also, ensure that the **yarn.nodemanager.local-dirs** field is set up properly. For more information, see the FAQ Environment with ssh banner enabled during IBM Spectrum Scale service deployment.
- | • To install the IBM Spectrum Scale service, an existing HDFS cluster is required. This can be created by installing the BI Ambari IOP with native HDFS, or the HDP stack with native HDFS.
- | • Manual Kerberos setup must be disabled in Ambari GUI before you deploy the IBM Spectrum Scale service.
- | • This chapter describes how to add IBM Spectrum Scale service as a root. If you plan to restrict root access, review the Restricting root access section first.

| **Adding Services**

| Ambari services can be added before or after the IBM Spectrum Scale™ service is installed.

| **Note:** 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 Spectrum Scale or vice versa, the service cannot provide the data that you entered before
| switching the file system.

| The following are the minimum services required to be installed before you install the IBM Spectrum
| Scale service:

- | • HDFS
- | • Yarn
- | • Mapreduce
- | • Zookeeper
- | • SmartSense (HDP)
- | • IBM BigSQL (Optional)

| **Note:**

- | • Kerberos: To manually set up Kerberos, you must disable the Ambari Kerberos before installing the
| IBM Spectrum Scale service. For more details, see the Kerberos section.
- | • For HDP: To install BigSQL on HDP, refer to IBM BigSQL Knowledge Center. For IBM Spectrum Scale
| management pack version 2.4.2.0, if the IBM Spectrum Scale service is added after adding IBM BigSQL,
| refer to the FAQ IBM Spectrum Scale service missing from Ambari when installed after HDP and
| BigSQL version 5.0.
- | • For management pack 2.4.2.1, see the Setting IBM Spectrum Scale configuration for BigSQL topic to
| correct the impersonation issues.
- | • For BI 4.2.5: BigInsights value-add services on IBM Spectrum Scale.

| **BI IOP or HDP installation**

| This section lists the installation procedure that is required for BI IOP/HDP and IBM Spectrum Scale
| environment, assuming that a local Ambari repo is used, and starting from installing the ambari-server.

| Before starting the software deployment, ensure to review the installation of the software stack from the
| Overview section.

| Follow the BigInsights installation procedure from the BI 4.2.5 documentation

| Follow the Hortonworks Data Platform installation procedure from the HDP documentation. The
| installation procedure is based on your platform: x86 or Power® LE.

| **Note:** You do not need to create a local partition file system for HDFS if you are deploying a new IBM
| Spectrum Scale FPO through Ambari. You can use a directory name that is not already hosting the
| Hadoop cluster.

| 1. Setup the Ambari repo file in /etc/yum.repos.d on the Ambari server by following the procedure
| listed in the Creating a mirror repository for the IBM Open Platform with Apache Hadoop software
| topic in the IBM BigInsights Knowledge Center or Using a local repository topic in the Hortonworks
| documentation.

| 2. Install Ambari by running the following command:

| `yum -y install ambari-server`

| 3. 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`

| For BI:

| `openjdk1.8.url`

| For HDP:

| 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.

| **Note:** For HDP, ensure that the correct JDK is setup based on your architecture. For x86, Oracle JDK
| is used. For PowerLE, IBM Power Open Source JDK is used.

| 4. 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 are not enough in Ambari, the system might
| suffer a heartbeat loss and the datanodes might going down. The Ambari GUI might not be able to
| start if not enough threads are 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

| 5. For HDP, if you are planning to install Falcon, the Berkeley DB JAR file is required to be installed. See
| Hortonwork Prerequisite to Installing or Upgrading Falcon page.

| From the HDP page, after the ambari-server setup --jdbc-db=bdb --jdbc-driver=/usr/share/je-
| 5.0.73.jar step, do not restart the ambari-server.

| 6. In HDP, to set up Ambari, execute **ambari-server setup** or **ambari-server setup -j \$JAVA_HOME**.

| 7. Start Ambari: **ambari-server start**

| 8. Configure Ambari:

| a. Log into Ambari.

| b. For BI, select a local repository and enter the IOP and IOP-UTILS paths for your environment.

| For HDP, select a local repository and enter the HDP and HDP-UTILS paths for your environment.

| c. Enter the target hosts and SSH private keys:

| In the Target Hosts section, Ambari requires a list of fully qualified domain names (FQDNs) of the
| nodes in the cluster.

| In HDP, 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 an existing file system, verify that the list of 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.

| • If this is an ESS, the ESS I/O servers must not be a part of the Ambari cluster.

| • Ensure that the Ambari server node is also the Ambari agent node and the GPFS Master node.

| d. In the **Confirm Hosts** panel, if the cluster has pre-existing UID or GID for the Hadoop
| components then a warning is encountered during the precheck. This host check warning for the
| user ID can be ignored, and the pop-up panel can be closed after verifying that all the other
| pre-requisites have passed. You can continue if you have only received a warning about the user
| ID. However, if there are other errors, then you must fix the errors to ensure that the other
| pre-requisites have been met.

| e. In the **Choose Services** panel, review the Adding Services. Choose the services you want to
| deploy.

- f. In the **Assign Slaves and Clients** panel, the `/etc/hadoop/conf/slaves` file is derived from the hosts that have the Datanode service.

Note: When creating a new IBM Spectrum Scale FPO cluster through Ambari when IBM Spectrum Scale is integrated in Adding the IBM Spectrum Scale service, the GPFS master is required to be set to be the Ambari server. The GPFS master is a GPFS Node. The HDFS Transparency node is required to be a Hadoop Datanode, a NodeManager, and a GPFS Node. Therefore, ensure that the Ambari server nodes have the **DataNode** and **NodeManager** option checked in the columns under **Assign Slaves and Clients** panel. Checking this option allows the new FPO cluster to create partitioning. If you do not want jobs to be scheduled on the Ambari server, remove the NodeManager after deploying IBM Spectrum Scale service.

- g. In the **Customize Services** panel, go to each red dot to enter the correct information:

- 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.

- 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 all the mounted directories on all the nodes in the cluster, or you must manually find all the places in the Ambari installation configuration and set it to a local directory. Otherwise BI IOP or HDP will not start or run correctly as the nodes in the cluster are accessing the same directories.

List of services' configurations that need the local directory to be configured:

- `yarn.nodemanager.log-dirs` (YARN Advanced)
Ex. Use `/hadoop/yarn/logs`
- `yarn.nodemanager.local-dirs` (YARN Advanced)
Ex. Use `/hadoop/yarn/local`
- `yarn.timeline-service.leveldb-timeline-store.path` (YARN Advanced)
Ex. `/hadoop/yarn/timeline`
- HBase local directory (HBase Advanced, under Advanced `hbase-site`)
- Oozie Data Dir (Oozie)
Ex. `/hadoop/oozie/data`
- ZooKeeper directory (ZooKeeper)
Ex. `/hadoop/zookeeper`
- `log.dirs` (Kafka)
Ex. `/kafka-logs`
- NameNode directories
Ex. `/hadoop/hdfs/namenode`
- DataNode directories
Ex. `/hadoop/hdfs/data`

- h. Ambari might give recommended configuration values. You can follow the suggestion and modified the values as requested. Even after modification, Ambari might still give the configuration message, so hit **Proceed Anyway** to go to the next step.

- i. Install, Start and Test have warnings:

- Continue and figure out which component failed and try to restart them manually.
- 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 **Service Actions > Run Service Check**.

IBM Spectrum Scale service installation

This section adds the IBM Spectrum Scale Ambari management pack into Ambari, and configures the IBM Spectrum Scale service.

| Before you start the software deployment, ensure to review the *Installation of software stack* topic from the Overview section.

| **Note:**

- | • Before starting the software deployment, make sure the Preparing the environment sections are reviewed and completed.
 - | – Ensure that password-less SSH is set up on every node.
 - | – Ensure all the user ID and group ID and Hadoop service user ID and group are same.
 - | – For pre-existing IBM Spectrum Scale, ensure that IBM Spectrum Scale is active and mounted.
 - | – Manual Kerberos setup requires Kerberos to be disabled in Ambari before deploying IBM Spectrum Scale mpack.
- | • If the IBM Spectrum Scale cluster has been created, a quorum node must be selected as the IBM Spectrum Scale Master node.
- | • Ambari only supports creating an IBM Spectrum Scale FPO file system.
- | • Namenode is configured as the host name by `fs.defaultFS` in the `core-site.xml` file in Hadoop version 2.4, 2.5, and 2.7.
- | • 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.

| **Pre-existing IBM Spectrum Scale cluster:**

| This section lists the steps to install IBM Spectrum Scale service on a pre-existing IBM Spectrum Scale cluster.

| If installing IBM Spectrum Scale service on a pre-existing IBM Spectrum Scale cluster:

- | 1. Ensure that IBM Spectrum Scale is set to auto-mount on reboot by running the following command:
| `/usr/lpp/mmfs/bin/mmchfs <device> -A yes`
- | 2. 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`
- | 3. Mount the file system over all nodes by running the following command:
| `/usr/lpp/mmfs/bin/mmmount <fs-name> -a`
- | 4. Ensure that IBM Spectrum Scale is active and mounted.

| **Note:**

- | • 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.
- | • 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:

- | a. 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

```

- b. 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/*

```

- c. 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 directory
|      // 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

```

Note: After installation, set the Yarn's configuration **yarn.nodemanager.local-dirs** as */hadoop/yarn/local* by clicking **Ambari GUI > Yarn > Configs setting**.

Installing the IBM Spectrum Scale Ambari management pack:

This topic lists the steps to install the management pack.

Note: Before you proceed, ensure that you review the IBM Spectrum Scale service section.

1. Ensure that the management pack, IBM Spectrum Scale service, is downloaded and unzipped into a local directory on the Ambari server node. This example uses the */root/GPFS_Ambari* directory.

2. Stop all services:

Log into 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-2.4.2.0.bin* executable:

- 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-2.4.2.0.bin
|
| 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.
|
| Ensure the input values are known before running the installer script.
|
| # cd /root/GPFS Ambari
| # ./SpectrumScaleIntegrationPackageInstaller-2.4.2.0.bin
| International License Agreement for Non-Warranted Programs
|
| ...
| c. wasted management time or lost profits, business, revenue, goodwill, or anticipated savings.
|
| Z125-5589-05 (07/2017)
|
| 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 : 172.16.1.11
| 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: Adding Spectrum Scale MPack : ambari-server install-mpack
| --mpack=SpectrumScaleExtension-MPack-2.4.2.0.tar.gz -v
| INFO: Spectrum Scale MPack Successfully Added. Continuing with Ambari Server Restart...
| INFO: Performing Ambari Server Restart.
| INFO: Ambari Server Restart Completed Successfully.
| INFO: Running command - curl -u admin:***** -H
| 'X-Requested-By: ambari' -X POST -d '{"ExtensionLink":
| {"stack_name":"BigInsights", "stack_version": "4.2.5",
| "extension_name": "SpectrumScaleExtension", "extension_version":
| "2.4.2.0"}}' http://c902f10x13:8080/api/v1/links/
| INFO: Extension Link Created Successfully.
| INFO: Starting Spectrum Scale Changes.
| INFO: Spectrum Scale Changes Successfully Completed.
| INFO: Performing AmbariServer restart.
| INFO: Ambari Server restarted successfully.
| Done.
|
| • This script automatically restarts the Ambari server.

```

Adding the IBM Spectrum Scale service:

This topic lists the steps to add an IBM Spectrum Scale service.

Note: Before you proceed, ensure that you review the Limitations section.

The steps are as follows:

1. Log back into Ambari to add the IBM Spectrum Scale service. Click **Ambari > Actions > Add services**.
2. On the Add Service Wizard, choose services panel, select the Spectrum Scale package and click **Next**.
3. In the Spectrum Scale UI configuration panel:
 - a. Co-locate the GPFS Master component to the same host as the Ambari-server.

- b. Select the GPFS Node components check box on the ALL hosts on the **Assign Slaves and Agents** page. This is a best practice. At a minimum, all the hosts which run the Namenodes and Datanodes should also have the GPFS Node running on them.

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.
- All HDFS Namenode and Datanodes are required to be GPFS Nodes.
- HDFS Transparency data node is required to be a Hadoop Datanode, a NodeManager, and a GPFS Node.
- If deploying 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 idle disks formatted as a local file system. 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.
- It is recommended to assign the metadata disks to the HDFS transparency Namenode running over a GPFS node.
- It is recommended to assign Yarn ResourceManager on the node running HDFS Transparency NameNode.

- c. Change the following in the Spectrum Scale **Standard and Advanced** Tabs:

- 1) In the Standard tab, adjust the parameters by using the slider bars and drop-down menus. The Advanced tab contains parameters that do not need to be changed frequently. 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. Therefore, verify all the parameters with the lock icon before proceeding to the next step. Further, while every attempt is made to detect the correct values from the cluster, verify that the parameters are imported properly and make corrections as needed.
Review the IBM Spectrum Scale configuration parameter checklist.
- 2) Review the parameters for Max Data Replicas and Max Metadata Replicas, as these values cannot be changed after the file system is created. If you decrease the values from the default of three, ensure that it is what you wanted to do. Also, setting the value of **Max Data Replicas**, **Max Metadata Replicas**, **Default Data Replicas**, and **Default Metadata Replicas** to 3 implies that there are at least three failure groups in the cluster (at least three nodes with disks) or the file system creation will fail.
- 3) Ambari credentials: Under **Advanced > Advanced gpfs-ambari-server-env** set the user ID and password of the Ambari server. Default is *ambari/ambari*.
- 4) GPFS Repo: Under **Advanced > Advanced gpfs-ambari-server-env** set the **GPFS_REPO_URL** field to the repo directory of where the IBM Spectrum Scale rpms are located.

Note: There must be no leading or trailing spaces in the repo name.

Example:

`http://c902mnx09-ug.pok.stglabs.ibm.com/repos/GPFS/4.2.2.3/gpfs_rpms`

This directory should contain the HDFS Transparency rpm.

- 5) If creating a new FPO cluster, do the following:

- Configuration fields on both standard and advanced tabs are populated with values taken from the Deploying a big data solution using IBM Spectrum Scale- **Hadoop Best Practices White Paper**.
- 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.
- 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.

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.

If the cluster has a partitioned file system, only a Standard NSD file can be used.

If the 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.p01`, must be created and placed into the `/var/lib/ambari-server/resources` directory. Add the policy filename, `gpfs_fs.p01`, into the **GPFS policy file** field in the Standard Config tab.
 - See Policy file on how to create a policy file.

For more information on each of the set-up points for standard NSD file, see Preparing a stanza File and IBM Spectrum Scale-FPO Deployment.

- 6) If you have a pre-existing IBM Spectrum Scale with remote cluster mount file system, follow the Configuring remote mount access onto an existing local Spectrum Scale cluster section.
- 7) If you want to configure multiple Spectrum Scale file system for Hadoop by using the pre-existing local and/or remote mounted file systems, follow the Configuring multiple file system mount point access section.
- 8) If you have a pre-existing IBM Spectrum Scale (FPO, ESS, Shared) cluster, then do the following:

- Ensure that IBM Spectrum Scale is active and mounted. If you have not started the IBM Spectrum Scale cluster but are 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 and that it is set to shared for shared file system (ESS).
- 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.

Note: If you accidentally place a value in that field, and then try to remove it, you must leave in a “blank” character for Ambari to proceed.

- For ESS only, create the /var/lib/ambari-server/resources/shared_gpfs_node.cfg file on the Ambari server. 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.

Note: For IBM Spectrum Scale Ambari management pack version 2.4.2.0 and GPFS Ambari integration module version 4.2.1 and earlier, the gpfs.gss package for monitoring needs to be installed but not configured on the node that is specified in the shared_gpfs_node.cfg file for Ambari to setup the shared mode for ESS correctly.

- [Optional] For shared storage, create the local cache disk for Hadoop usage. Create the Hadoop local cache disk stanza file, `hadoop_disk`, in /var/lib/ambari-server/resources directory. Add the filename, `hadoop_disk`, to the **Hadoop local cache disk stanza file** field in the Standard config tab.

Hadoop local cache disk stanza file:

```
[root@compute000 GPFS]# 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
```

Type the `hadoop_disk` file name in the **Hadoop local cache disk stanza file** field in the Standard Config tab.

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.supergroup=hadoop,root (Appears as hdfs,root from Mpack version 2.4.2.2 and later)
- | – gpfs.storage.type=shared (ESS or Shared)
- | or
- | gpfs.storage.type=local (FPO)

| **Note:** 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 Spectrum Scale cluster.

| 9) Kerberos settings:

| The Kerberos principal and password can be set through the IBM Spectrum Scale service in Ambari.

| 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.

| 10) Click **Deploy**. If deployment is successful, go to the next step. If not, fix the problem and click on the **Retry** button.

| 4. Once the deployment completes, restart the Ambari server.

| On the Ambari server, run **ambari-server restart**.

| This is a mandatory step after deploying the IBM Spectrum Scale service.

| **Note:** Restarting the Ambari server is not required if you are using Mpack 2.4.2.6 and later.

| After the restart, 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.

| 5. Log into Ambari GUI.

| a. Start all services from Ambari. Click **Ambari GUI > Actions > Start All**. If some services did not start properly, start them by going to the host dashboard, and restarting each service individually.

| b. Once all the services are up, HDFS will get alerts on the Namenodes.

| This is because HDFS Transparency does not do the checkpointing because IBM Spectrum Scale is stateless. Disable the alert since Namenode checkpoint is not relevant. From **HDFS panel > Alert > NameNode Last Checkpoint > State:Enabled > Confirmation panel > Confirm Disable**.

| c. If Accumulo fails to start, see the FAQ Accumulo Tserver failed to start.

| If Atlas fails to start, see the FAQ Atlas Metadata server failed to start or the Web UI cannot be accessed.

| d. If IBM Spectrum Scale service missing from Ambari when installed after HDP and BigSQL, see the FAQ IBM Spectrum Scale service missing from Ambari when installed after HDP and BigSQL version 5.0.

| 6. If any configuration in the gpfs-site is changed in the Spectrum Scale dashboard in Ambari, a restart required alert is displayed for the Spectrum Scale service and the HDFS service. Check your environment to ensure that the changes made are in effect.

| **Important:** IBM Spectrum Scale service must be restarted and only then can the HDFS service be restarted. To restart the Spectrum Scale service, on the Spectrum Scale dashboard, select **Service Actions > Stop > Start options**.

- Restart all affected components with Stale Configs when the dashboard displays the request. For IBM Spectrum Scale, restart from the **Spectrum Scale dashboard > Service Actions > Stop and Start options**.

Configuring Remote Mount Access onto an existing local Spectrum Scale cluster:

This topic lists the steps to set up and configure remote mount file system access.

Note: Starting from IBM Spectrum Scale Ambari management pack version 2.4.2.1 along with HDFS Transparency version 2.7.3.1, you can configure one remote Spectrum Scale file system for Hadoop use. For remote mount access, only the HDP Hadoop distribution supports the IBM Spectrum Scale Ambari management pack version 2.4.2.1 and higher.

- An existing local IBM Spectrum Scale cluster is required. This cluster must be a different cluster from the ESS-based cluster.
However, ensure that the version of Spectrum Scale on the local cluster is higher than or same as the version on the file system owning the cluster.

Note: 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.

Otherwise, 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 is not the same as the ESS cluster, then on the local cluster, run the following command:

```
/usr/lpp/mmfs/bin/mmchconfig maxblocksize=<ESS maxblocksize value>
```

- One or more file systems from the ESS are already mounted onto the local IBM Spectrum Scale cluster.

Note: Ensure that the local clusters have passwordless ssh to the first node listed in the contact node list. To see the contact node list, run the **mmremoteccluster show all** command.

- After the mounting of a remote GPFS file system is completed, run the **mmremotefs** command in your local cluster.

```
[root@mn01 ~]# /usr/lpp/mmfs/bin/mmremotefs show all
Local Name Remote Name Cluster name Mount Point Mount Options Automount Drive Priority
essfs gpfs0 c902mnp05-ess.gpfs.net /essfs rw no - 0
```

The above example shows the following:

- A file system that is called *gpfs0* exists on the ESS.
 - The above file system from ESS is remote mounted as *essfs* on the local cluster.
 - The local mount point name of the remote mounted file system is */essfs*.
- To add the IBM Spectrum Scale service to the existing local IBM Spectrum Scale cluster with remote mount access, follow the Add the Spectrum Scale service section for pre-existing IBM Spectrum Scale, but with the following changes:
 - Do not set the ESS `/var/lib/ambari-server/resources/shared_gpfs_node.cfg` configuration file for remote mount setup.
 - In the IBM Spectrum Scale service configuration UI window, the following fields are required to be set for remote mount setup:

Definition

<code>gpfs.storage.type:</code>	remote
---------------------------------	--------

GPFS file system Name:	Local name of the remote mounted file system.
gpfs.mnt.dir:	Only one local name can be configured. Local mount point name. Only one mount point name can be configured.

For example, the fields would then be set as follows:

gpfs.storage.type:	remote
GPFS file system Name:	essfs
gpfs.mnt.dir:	/essfs

Showing how the entries would look from the IBM Spectrum Scale GUI window:

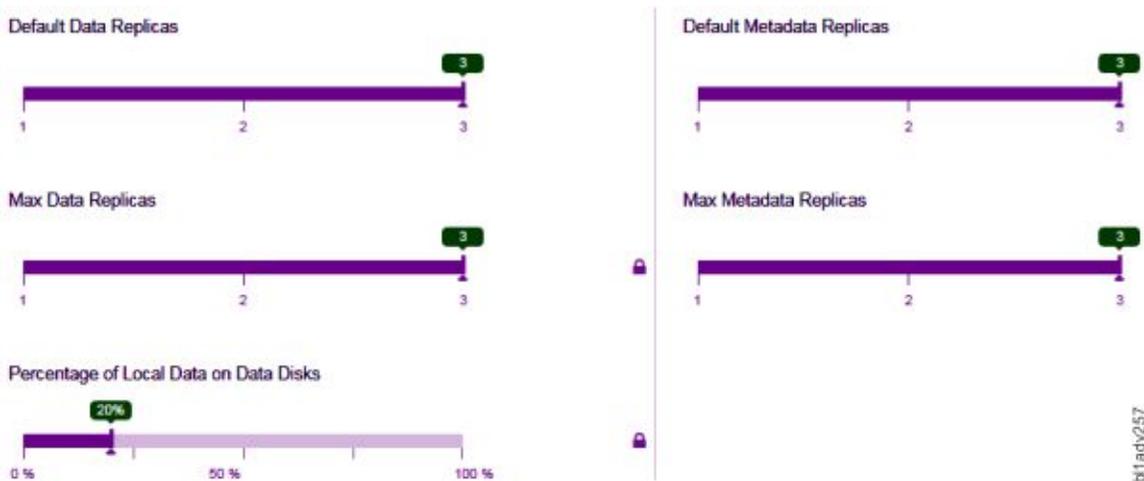


5. Click **Deploy** after you verify all the configuration fields for the local cluster.

Note: The fields in Ambari for the local cluster are not configurable for the remote file system.

For example,

- The default replica values are not configurable for the remote file system.
- The remote file system replica is to be configured locally on the remote file system.



Important: After you deploy the IBM Spectrum Scale service onto the local cluster, ensure that all the **Restart Required** icon is processed. There should not be any **Restart Required** icon that is seen in Ambari. Especially for the Spectrum Scale service. This must be done before you run the Starting All services.

6. Continue to follow the Add the Spectrum Scale service section.

For HDFS Transparency version 2.7.3-0 and 2.7.3-1, all the transparency nodes that are located in the local cluster are required to configure password-less ssh root access to one of the contact nodes that belongs to the ESS cluster.

For HDFS Transparency version 2.7.3-2 and later, the internal configuration files will be automatically generated if they are not detected.

In HDFS Transparency, if you configure remote mounted file system support, you need to configure password-less ssh root access from HDFS Transparency NameNode to at least one of the contact nodes from the remote cluster.

For more information on setting up password-less ssh access and how the internal configuration files are generated based on the HDFS Transparency version, see the “Password-less ssh access” on page 15 and “Cluster and file system information configuration” on page 22 sections.

If password-less ssh access configuration cannot be set up, starting from HDFS transparency 2.7.3-2, you can configure **gpfs.remoteccluster.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 as root.

- a. If you are using Ambari, add the **gpfs.remoteccluster.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 15 section.

Note:

- a. Ambari does not automatically detect a remote mounted file system to propagate values into the IBM Spectrum Scale service configuration window.
- b. 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 `/usr/lpp/mmfs/hadoop/etc/hadoop/gpfs-site.xml` is set as *shared*. Ensure that you update only through Ambari.
- c. The IBM Spectrum Scale cluster running the application requires to have consistent uid/gid configuration. The owning cluster of the remote mount does not require uid/gid setup. For more information, see User and group ids.

Configuring multiple file system mount point access:

Starting with management pack version 2.4.2.1 along with HDFS Transparency version 2.7.3.1, multiple IBM Spectrum Scale file systems can be configured for Hadoop use. Only the HDP Hadoop distribution supports the IBM Spectrum Scale Ambari management pack version 2.4.2.1 and later for multiple file system mount point access.

Currently, in management pack version 2.4.2.1, the GUI supports up to two file systems.

Follow these steps to configure multiple file system support for Hadoop usage:

1. Ensure that you have deployed the management pack version 2.4.2.1 and HDFS Transparency version 2.7.3.1.
2. 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.
 - 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 *Mounting a remote GPFS file system* topic in the *IBM Spectrum Scale: Administration Guide*.
3. 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. In management pack version 2.4.2.1, only the following combination of file system values is supported: gpfs.storage.type=local,remote gpfs.storage.type=remote,remote
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 FileSystem 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.

4. Follow the instructions based on the type of deployment model that you have:

- a. 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** to stop all the services.
- On the owning Spectrum Scale cluster, run the **/usr/lpp/mmfs/bin/mmfsmlount all** command to ensure that the file system is mounted.

This step is needed for the Spectrum Scale deploy wizard to automatically detect the existing file systems.

- Update the 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 Spectrum Scale service.
 - Restart any service with **Restart Required** icon.
 - Click **Ambari > Actions > Start All** to start all the services.
- b. 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** to stop all the services.
- Start Spectrum Scale service cluster.
On the local 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 Spectrum Scale cluster, run the `/usr/lpp/mmfs/bin/mmgetstate -a` command to ensure it is started.

This step is needed for the Spectrum Scale deploy wizard to automatically detect the existing file systems.

- Deploy the Spectrum Scale service on the pre-existing file system.

During deployment, the wizard would detect both the file systems and would populate the 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
```

- Restart the Ambari server from the command line.

Note: Restarting the Ambari server is not required if you are using Mpack 2.4.2.6 and later.

- Click **Ambari > Actions > Start All** to start all the services.
- c. 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** to stop all the services.
- Start Spectrum Scale service cluster.
On the local 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 Spectrum Scale cluster, run the `/usr/lpp/mmfs/bin/mmgetstate -a` command to ensure it is started.

This step is needed for the Spectrum Scale deploy wizard to automatically detect the existing file systems.

- Deploy the Spectrum Scale service.

During deployment, the wizard would detect both the file systems and would populate the 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 Spectrum Scale service.
- Restart any service with the **Restart Required** icon.
- On the Ambari UI, click **Actions > Start All** to start all the services.

- d. 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** to stop all the services.

- Start Spectrum Scale cluster.

On the local 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 Spectrum Scale cluster, run the `/usr/lpp/mmfs/bin/mmgetstate -a` command to ensure it is started.

This step is needed for the Spectrum Scale deploy wizard to automatically detect the existing file systems.

- Deploy the Spectrum Scale service.

During deployment, the wizard would detect both the file systems and would populate the 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
```

- Restart the Ambari server from the command line.

Note: Restarting the Ambari server is not required if you are using Mpack 2.4.2.6 and later.

- On the Ambari UI, click **Actions > Start All** to start all the services.

Verifying and testing the installation

After the BigInsights or Hortonworks with IBM Spectrum Scale service is deployed, verify the installation setup.

Note: To run IBM Spectrum Scale commands, add the `/usr/lpp/mmfs/bin` directory to the environment `PATH`.

1. For an initial installation through Ambari, the UID and GID of the users is consistent over all nodes. However, if you deploy it for the second time, or part of the nodes have been created with the same UID or GID, the UID and GID of these users might not be consistent over all nodes, as per the AMBARI-10186 issue, from the Ambari community.

After the deployment and during verification of system, check by using `mmdsh -N all id <user-name>` to see whether the UID is consistent across all nodes.

2. After the Ambari deployment, 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. Check user ID, `ambari-qa`, and user access to the file system.

```
HDFS commands:
[ambari-qa@c902f05x01 bigpfs]$ hadoop fs -ls /user
Found 1 items
drwxrwx--- - ambari-qa root          0 2017-05-31 14:45 /user/ambari-qa
[ambari-qa@c902f05x01 bigpfs]$
```

```
POSIX commands:
[ambari-qa@c902f05x01 user]$ pwd;ls -ltr
/bigpfs/user
total 0
drwxrwx--- 2 ambari-qa root 4096 May 31 14:45 ambari-qa
[ambari-qa@c902f05x01 user]$
```

```
[ambari-qa@c902f05x01 ambari-qa]$ pwd
```

```

/bigpfs/user/ambari-qa
[ambari-qa@c902f05x01 ambari-qa]$ hadoop fs -ls
[ambari-qa@c902f05x01 ambari-qa]$

[ambari-qa@c902f05x01 ambari-qa]$ echo "My test" > mytest
[ambari-qa@c902f05x01 ambari-qa]$ cat mytest
My test
[ambari-qa@c902f05x01 ambari-qa]$

[ambari-qa@c902f05x01 ambari-qa]$ hadoop fs -cat mytest
My test
[ambari-qa@c902f05x01 ambari-qa]$

[ambari-qa@c902f05x01 ambari-qa]$ rm mytest
[ambari-qa@c902f05x01 ambari-qa]$ ls -ltr
total 0
[ambari-qa@c902f05x01 ambari-qa]$

```

4. Run wordcount as user.

This example uses the BI IOP word count.

1. Copy the mywordcountfile file to be used as input to the wordcount program.

```

For HDP:
[ambari-qa@c902f10x13 ambari-qa]$ yarn jar
/usr/hdp/2.6.2.0-205/hadoop-mapreduce/hadoop-mapreduce-examples-2.7.3.2.6.2.0-205.jar wordcount mycountfile wc_output
17/10/13 15:16:08 INFO client.RMProxy: Connecting to ResourceManager at c902f10x14.gpfs.net/172.16.1.93:8050
17/10/13 15:16:08 INFO client.AHSPProxy: Connecting to Application History server at c902f10x14.gpfs.net/172.16.1.93:10200
17/10/13 15:16:08 INFO input.FileInputFormat: Total input paths to process : 1
17/10/13 15:16:09 INFO mapreduce.JobSubmitter: number of splits:1
17/10/13 15:16:09 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1507902950551_0006
17/10/13 15:16:09 INFO impl.YarnClientImpl: Submitted application application_1507902950551_0006
17/10/13 15:16:09 INFO mapreduce.Job: The url to track the job:
http://c902f10x14.gpfs.net:8088/proxy/application_1507902950551_0006/
17/10/13 15:16:09 INFO mapreduce.Job: Running job: job_1507902950551_0006

```

```

[ambari-qa@c902f05x01 ambari-qa]$ pwd
/bigpfs/user/ambari-qa
[ambari-qa@c902f05x01 ambari-qa]$
[ambari-qa@c902f05x01 ambari-qa]$ cp /etc/passwd mycountfile
[ambari-qa@c902f05x01 ambari-qa]$

```

2. Run the wordcount program.

```

For BI IOP:
[ambari-qa@c902f05x01 ambari-qa]$ yarn jar
/usr/iop/4.2.5.0-0000/hadoop-mapreduce/hadoop-mapreduce-examples-2.7.3-IBM-29.jar
wordcount mycountfile wc_output
17/05/31 14:51:30 INFO impl.TimelineClientImpl: Timeline service address:
http://c902f05x02.gpfs.net:8188/ws/v1/timeline/
17/05/31 14:51:30 INFO client.RMProxy: Connecting to ResourceManager at
c902f05x02.gpfs.net/172.16.1.13:8050
17/05/31 14:51:31 INFO input.FileInputFormat: Total input paths to process : 1
17/05/31 14:51:31 INFO mapreduce.JobSubmitter: number of splits:1
17/05/31 14:51:31 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1496256368436_0001
17/05/31 14:51:31 INFO impl.YarnClientImpl: Submitted application application_1496256368436_0001
17/05/31 14:51:31 INFO mapreduce.Job: The url to track the job:
http://c902f05x02.gpfs.net:8088/proxy/application_1496256368436_0001/
....

```

3. Check the output in directory.

```

[ambari-qa@c902f05x01 ambari-qa]$ hadoop fs -ls wc_output
Found 2 items
-rw-r--r--  3 ambari-qa root          0 2017-05-31 14:51 wc_output/_SUCCESS
-rw-r--r--  3 ambari-qa root    43139 2017-05-31 14:51 wc_output/part-r-00000
[ambari-qa@c902f05x01 ambari-qa]$

[ambari-qa@c902f05x01 ambari-qa]$ pwd; ls -ltr wc_output
/bigpfs/user/ambari-qa

```

```
total 192
-rw-r--r-- 1 ambari-qa root 43139 May 31 14:51 part-r-00000
-rw-r--r-- 1 ambari-qa root    0 May 31 14:51 _SUCCESS
[ambari-qa@c902f05x01 ambari-qa]$
```

5. Check the Hadoop GUI.

6. For more validation runs, see BigInsights install validation in the IBM Knowledge Center or for HDP, see Smoke Test MapReduce job.

Uninstalling IBM Spectrum Scale Mpack and service

Before upgrading Mpack on an HDP cluster, see the Preparing the environment topic 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. Uninstalling only the IBM Spectrum Scale service

From Ambari GUI > IBM Spectrum Scale service > Service Actions > Unintegrate_Transparency Restart Ambari server.

From Ambari GUI > IBM Spectrum Scale service > Service Actions > Delete Service. The IBM Spectrum Scale MPack is preserved. Add Service can be used to add back the IBM Spectrum Scale service.

2. Uninstalling the management pack and the IBM Spectrum Scale service

```
$ python SpectrumScaleMPackUninstaller.py
```

| The SpectrumScaleMPackUninstaller.py script is in the download package along with the Mpack and
| license bin executables.

| The SpectrumScaleMPackUninstaller.py script verifies the settings before it uninstalls the Mpack and
| services. If the services are still running, and if the HDFS Transparency is not unintegrated, the
| SpectrumScaleMPackUninstaller.py script will exit, and request user action.

| Follow the action, and rerun the SpectrumScaleMPackUninstaller.py script.

| For example: The management pack is installed but the IBM Spectrum Scale service is not added.

```
| # python SpectrumScaleMPackUninstaller.py
| 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 : 172.16.1.11
| 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: Removing Spectrum Scale MPack.
| INFO: Performing Ambari Server Restart.
| INFO: Ambari Server Restart Completed Successfully.
| INFO: Spectrum Scale Mpack Removal Successfully Completed.
| #
```

| For example: The management pack is installed and the IBM Spectrum Scale service is added. The
| services are not stopped, and the HDFS Transparency is not unintegrated.

```
| # python SpectrumScaleMPackUninstaller.py
| 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 : 172.16.1.11
| Enter Ambari Server Username, default=admin : admin
| Enter Ambari Server Password :
| INFO: Verifying Ambari Server Address, Username and Password.
| INFO: Verification Successful.
| INFO: Spectrum Scale Service is added in Ambari.
| ERROR: Please stop all services, unintegrate the service and then retry this operation.
| #
```

| For example: The management pack is installed, and the IBM Spectrum Scale service is added. The
| services are stopped, and HDFS Transparency is unintegrated.

```
| python 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 : c902f10x13
| 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: Running command - curl -u admin:***** -X
| GET http://c902f10x13:8080/api/v1/clusters/nampatra/services/KERBEROS
| INFO: It is not a manual Kerberos setup.
| Enter kdc principal: root/admin@IBM.COM
| Enter kdc password:
| INFO: Kerberos is Enabled. Proceeding with Configuration
| OK
| INFO: Kerberos Authentication done Successfully. Proceeding with Component Addition Step.
| INFO: Spectrum Scale Service is added in Ambari.
| Deleting the Spectrum Scale Service for removing the mpack. Would you like to continue?(y/n): y
| 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.
```

| **Note:** This 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.

| Uninstalling Ambari stack

| This topic lists the steps to uninstall Ambari stack.

| For BI, follow the steps in the Cleaning up nodes before reinstalling software section to remove the BI IOP stack.

| For HDP, follow the steps in the Uninstalling HDP section.

| **Note:** If the IBM Spectrum Scale service has been installed, the IBM Spectrum Scale packages and directories are not removed. If the NSD and partitions have been created, they are not removed. To clean up IBM Spectrum Scale, see IBM Spectrum Scale documentation in the Knowledge Center.

| BigInsights value-add services on IBM Spectrum Scale

Several value-add services from BigInsights can be installed by using the Ambari GUI.

Any of these services can be optionally installed, and do not explicitly depend on one another.

- The BigInsights home service provides a web UI which serves as a launching pad for the web UI's of the Data Server Manager, BigSheets, and Text Analytics services.
- When you install Big SQL, BigSheets, or Text Analytics, install the BigInsights Home service.
- While adding the Big SQL service, the **bigsql_user_password** must be set to *bigsql*.
- BigSQL and BigR services check the workarounds after unintegrating the HDFS Transparency. For more information, see the General section under Troubleshooting Value Add Services.
- You must manually copy the BigSQL biga-hbase-index.jar file into the GPFS path.

Installation

Use this procedure to install for BigInsights value-add services.

1. Perform the preparation steps for BigInsights value-adds: IBM BigInsights 4.2 documentation - Preparing to install the BigInsights value-add services
2. Install the BigInsights value-add package as stated in the BigInsights Knowledge Center web page: IBM BigInsights 4.2 documentation - Installing the BigInsights value-add packages

Note: BigSQL automatically determines the number of GPFS NSD servers, sets the number of worker threads to that number, and runs them on the NSD nodes. However, in the case of a shared storage system like ESS, BigSQL reports an error because there are only a limited number of NSD servers, and usually they are not part of the Hadoop cluster.

To correctly configure the number of worker threads for BigSQL in a shared storage system or in a remote mounted environment, the following workaround must be set in the BigSQL bigsql-conf.xml configuration file:

```
<property>  
  <name>scheduler.dataLocationCount</name>  
  <value>max:8</value>  
  <description>Set this to max:number-of-worker-nodes in gpfs-shared-disk environment</description>  
</property>
```

This specifies the number of worker threads that BigSQL must use, and BigSQL does not enforce the worker threads to run only on the GPFS NSD server nodes.

3. For BigSQL environment, additional steps are required on IBM Spectrum Scale file system.

- | On the BigSQL head node or on the master host, copy the BigSQL jar file, biga-hbase-index.jar, into the GPFS path as follows:

```
$ cp /usr/ibmpacks/bigsql/4.2.5.0/bigsql/lib/java/hbase-coprocessors/biga-hbase-index.jar  
    <gpfs.mnt.dir>/<gpfs.data.dir>/biginsights/bigsql/hbase-coprocessors/biga-hbase-index.jar
```

```
$ chown hdfs:hadoop <gpfs.mnt.dir>/<gpfs.data.dir>/biginsights/  
$ chown -R bigsql:hadoop <gpfs.mnt.dir>/<gpfs.data.dir>/biginsights/*
```

To find the **<gpfs.mnt.dir>** and **<gpfs.data.dir>** values, log into **Ambari GUI > Spectrum Scale > Config > Search field**, and type in *gpfs.mnt.dir* or *gpfs.data.dir*.

Troubleshooting value-add services

Use this procedure to troubleshoot value-add services.

1. The BigInsights Home webpage is blank.

Solution: Configure the Knox service and restart the BigInsights service to see the home webpage.

- a. Enabled Demo LDAP in Knox.

Logon to **Ambari GUI > Knox > Service Actions > Start Demo LDAP**.

- b. Go to `/usr/ibmpacks/bin/<version>` directory of the BI installation.
- c. Execute: `./knox_setup.sh -u admin -p admin -x 8080`, and follow the prompts.
- d. Restart the Knox service in the Ambari GUI.
- e. Restart BIGINSIGHTS_HOME service in the Ambari GUI.
- f. Verify the BigInsights Home web page: `https://<bi_home_host>:8443/gateway/default/BigInsightsWeb/index.html`.

For more details, see Enable Knox for BigInsights value-add services in the IBM Knowledge Center.

2. Big SQL and Big R service check limitation.

Solution: On the Ambari dashboard, select **Spectrum Scale > Service Actions > Unintegrate Transparency**.

Create the user directories for BigSQL and BigR. Otherwise, the service check for those services might fail.

```
# su - hdfs -c " hadoop fs -mkdir /user/bigsql "  
# su - hdfs -c " hadoop fs -chown bigsql:hadoop /user/bigsql "  
# hadoop fs -ls /user
```

```
# su - hdfs -c " hadoop fs -mkdir /user/bigr "  
# su - hdfs -c " hadoop fs -chown bigr:hadoop /user/bigr "  
# su - hdfs -c " hadoop fs -chmod 777 /user/bigr "  
# hadoop fs -ls /user
```

Upgrading software stack

This section lists the upgrading and uninstallation procedures for BigInsights IOP/Hortonworks HDP and IBM Spectrum Scale service.

Migrating from BI IOP to HDP

- | You can migrate BigInsights IOP to Hortonworks Data Platform if you have BI IOP version 4.2 or 4.2.5.

- | If you do not have BI IOP version 4.2 or 4.2.5, see Upgrading BigInsights IOP and IBM Spectrum Scale service to upgrade to the supported versions.

Migrating to HDP 2.6.2

- | This section describes migration from BI IOP 4.2 or BI IOP 4.2.5 with IBM Spectrum Scale service to HDP 2.6.2 with IBM Spectrum Scale service.

| IBM Spectrum Scale Ambari management pack version 2.4.2-1 supports migration from BI IOP to HDP.
| Only the express upgrade method is supported for migrating to HDP with IBM Spectrum Scale service.
| HDFS Transparency 2.7.3-1 is required for management pack version 2.4.2-1.

| You must plan a cluster maintenance window and prepare for cluster downtime during the migration.

| **Note:**

- | • Download the management pack version 2.4.2-1 as root to a directory on the Ambari server node.
| This example uses the /root/GPFS_Ambari directory.
| Gunzip and unzip the management pack into the /root/GPFS_Ambari directory.
| The **SpectrumScale_UpgradeIntegrationPackage** script used for upgrade and migration is run from the
| /root/GPFS_Ambari directory.
- | • Migrating to HDP with IBM Spectrum Scale service does not affect the IBM Spectrum Scale file system.
- | • You do not need to unconfigure HDFS HA when you are migrating the services for IBM Spectrum
| Scale.

| **Procedure**

- | 1. Log in to Ambari.
- | 2. Stop HBase services manually. The **Stop All** process in Ambari does not stop the HBase components
| in the correct order. To stop HBase, go to **Services > Hbase** and stop the HBase master. After you
| have stopped the HBase master, stop the Hbase region servers. If the HBase services were not
| stopped in the correct sequence, HBase might fail to start after migrating to HDP. If HBase failed to
| start after migrating to HDP, follow the FAQ HBase fail to start after migrating from BI to HDP.
- | 3. Stop all the services. Click **Ambari > Actions > Stop All**.
- | 4. 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.

- | 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. On the Ambari server node as root, run the **SpectrumScale_UpgradeIntegrationPackage** script with
| the **preEU** option:

```
| $ cd /root/GPFS_Ambari  
| $ ./SpectrumScale_UpgradeIntegrationPackage --preEU
```

| 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.

```
| $ ./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. If you are migrating from BI IOP 425, issue the following command:
|
| $./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 : 127.0.0.1
| 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. Start all services. Click Ambari > Actions > Start All.
|
| Wait for all the services to start. At this stage, native HDFS is used.
|
| 9. To migrate from BI IOP to HDP, you need to follow the procedure given in the Hortonworks IOP to
| HDP Migration guide.
|
| 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.
|
| 10. After BI IOP is successfully migrated to HDP, stop all services.
|
| Click Ambari > Actions > Stop All.
|
| Wait until all services have stopped. Ensure that the native HDFS has stopped running.
|
| 11. HDP supports HDFS Transparency version 2.7.3 and later.
|
| If you are migrating from BI 4.2, add the HDFS Transparency version 2.7.3 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.
|
| 12. Add the IBM Spectrum Scale service.
|
| 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.
|
| $ cd /root/GPFS_Ambari
| $ ./SpectrumScale_UpgradeIntegrationPackage --postEU
|
| 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.
|
| $ ./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 :
| 172.16.1.17
| 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:
| ...
| 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.
| *****

```

13. Update the HDFS Transparency package to all the GPFS nodes.

HDP requires HDFS Transparency version 2.7.3. If your BI IOP version is 4.2, update the HDFS Transparency package before you start any services.

Ensure that the HDFS Transparency package, gpfs.hdfs-protocol-2.7.3.X, is updated in the IBM Spectrum Scale repository as stated in step 10.

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](#).

14. Start all services.

Click **Ambari > Actions > Start All**.

Restart all components by using the **Restart** icon.

Note:

- If the 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 IBM Spectrum Scale service MPack

This section describes the IBM Spectrum Scale MPack upgrade process.

Ambari Management Packs (MPack) was introduced in IBM Spectrum Scale Ambari management pack version 2.4.2.0.

You must plan a cluster maintenance window and prepare for cluster downtime when you upgrade the IBM Spectrum Scale MPack.

Note:

- The cluster must be at management pack version 2.4.2.0 or later.
- Download a management pack that, at a higher PTF version than the current installed version of IBM Spectrum Scale service, acts as root to a directory on the Ambari server node. This example uses the /root/GPFS_Ambari directory.
- The management pack contains the upgrade script to upgrade the MPack.
- For example: The current installed cluster is at management pack version 2.4.2.0 and is planned to upgrade to version 2.4.2.1.
- Upgrading MPack does not affect the IBM Spectrum Scale file system.
- You do not need to unconfigure HDFS HA when you migrate the services for IBM Spectrum Scale.

Procedure

1. Log in to Ambari.
2. Stop all the services. Click **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. If the IBM Spectrum Scale service is not already stopped, stop the IBM Spectrum Scale service by clicking **Ambari > Spectrum Scale > Service Actions > Stop**.
5. As root on the Ambari server node, run the **SpectrumScale_UpgradeIntegrationPackage** script:

```
$ cd /root/GPFS_Ambari
$ ./SpectrumScale_UpgradeIntegrationPackage --preEU
```

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.

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
$ ./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:
| ...
| 6. Run the MPack uninstaller script from the currently installed GPFS Ambari integration package to
| remove the existing MPack link.
|
| $./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 : 127.0.0.1
| 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.
| 7. HDP is now in the native HDFS mode.
|
| • If you plan to upgrade HDP to a newer level, follow the Hortonworks documentation process to
| upgrade the HDP and 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.
| 8. 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.
|
| $ cd /root/GPFS_Ambari
| $ ./SpectrumScale_UpgradeIntegrationPackage --postEU
|
| 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.
|
| $ ./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 :
| 172.16.1.17
| 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:
| ...
| 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.
| *****

```

9. Start all the services.

Click **Ambari > Actions > Start All**.

Restart all the components by using the restart icon.

Note:

- If the 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 the HDFS Transparency through Ambari. The IBM Spectrum Scale update package and the HDFS Transparency is upgraded separately.

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: Only one version of the HDFS Transparency must be in the repo.

2. Go to the IBM Spectrum Scale yum directory, and rebuild the yum database by running the **createrepo** command.

```

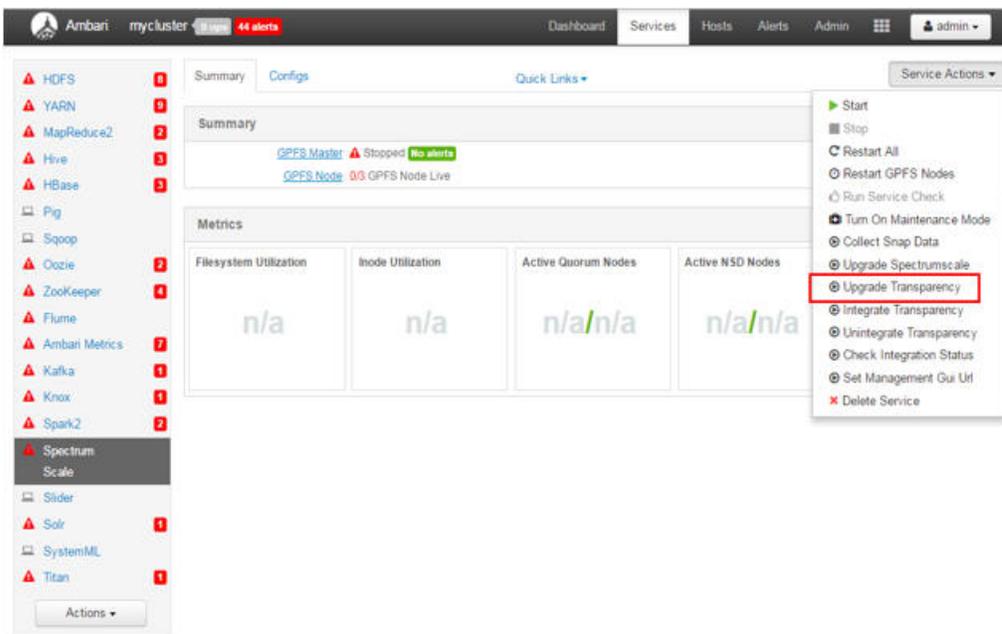
$ cd /var/www/html/repos/GPFS/4.2.2.3/gpfs_rpms
$ createrepo .
Spawning worker 0 with 2 pkgs
Spawning worker 1 with 2 pkgs
Spawning worker 2 with 2 pkgs
Spawning worker 3 with 2 pkgs
Workers Finished
Saving Primary metadata
Saving file lists metadata
Saving other metadata
Generating sqlite DBs
Sqlite DBs complete

```

- From the dashboard, select **Actions > Stop All** to stop all the services.

Note: To upgrade the HDFS Transparency, the IBM Spectrum Scale file system does not need 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 **Service Actions > Stop** for all, except the IBM Spectrum Scale service.

- From the dashboard, click **Spectrum Scale > Service Actions > Upgrade Transparency**.



- 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-2.7.3-X.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:

- Remove the existing HDFS Transparency package by running the following command:

```

$ yum erase gpfs.hdfs-protocol<old version>
$ yum clean all

```

- yum install the new package.

```

$ yum install gpfs.hdfs-protocol-2.7.3-X.<OS>.rpm

```

- On the dashboard, click **Actions > Start All**.
- Check 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.

You can update the IBM Spectrum Scale packages through the Ambari GUI. This function will upgrade the IBM Spectrum Scale packages and run 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.

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.

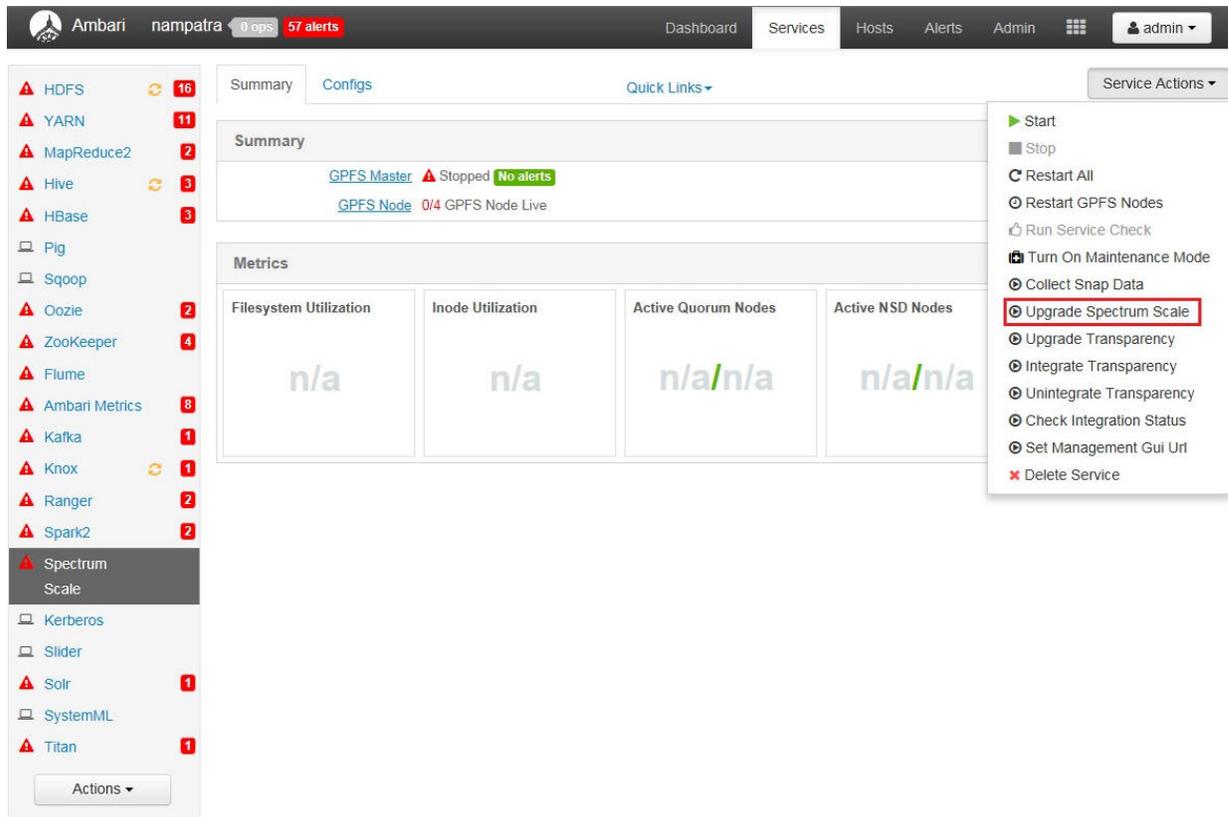
You must put all the update packages (PTF) into a yum repository. If the Yum repository is not the existing IBM Spectrum Scale yum repository path specified in Ambari, add the yum repository URL to Ambari Spectrum Scale configuration. For information on updating the yum repository, see GPFS yum repo 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** to stop all services.
3. From the dashboard, select **Spectrum Scale > Service Actions > Upgrade Spectrum Scale**.



Upgrading to BI IOP 4.2.5

Only the BigInsights express upgrade method is supported for upgrading BI IOP.

You must plan a cluster maintenance window, and prepare for the cluster downtime when upgrading BigInsights IOP with the IBM Spectrum Scale service (GPFS Ambari integration module) and HDFS Transparency to BI IOP 4.2.5 with IBM Spectrum Scale Ambari management pack 2.4.2.0 and HDFS Transparency.

Follow the section that pertain to your current environment when upgrading to BI IOP 4.2.5 and IBM Spectrum Scale Ambari management pack version 2.4.2.0 and HDFS transparency.

Upgrading from BI IOP 4.1/4.2 with GPFS Ambari integration HDFS Transparency to BI IOP 4.2.5

This section describes upgrading from BI IOP 4.1 or BI IOP 4.2 with IBM Spectrum Scale Ambari management pack HDFS Transparency to BI IOP 4.2.5 with GPFS Ambari integration HDFS Transparency.

Note:

- Download the management pack, IBM Spectrum Scale service, as root to a directory on the Ambari server node. This example uses the `/root/GPFS_Ambari` directory.
- Upgrading the BI IOP and the IBM Spectrum Scale service does not affect the IBM Spectrum Scale file system.
- IBM Spectrum Scale Ambari management pack version 2.4.2 requires HDFS Transparency package version 2.7.3-0 and above.
- You do not need to unconfigure HDFS HA when upgrading the services for IBM Spectrum Scale.

- On all the GPFS nodes ensure that the `/usr/lpp/mmfs/etc/hadoop/etc/slaves` file have the same Datanode entries as the HDFS Datanodes from Ambari. Validation during installation of the GPFS Master and GPFS Node components will fail if the Datanode entries do not match.

Procedure

1. Log into Ambari.
2. Stop all the services. Click **Ambari > Actions > Stop All**.
3. Once all the services have stopped, unintegrate the transparency. Follow the steps in Unintegrating Transparency, and ensure that the **ambari-server restart** is executed. Do not start services.
4. Ensure that the IBM Spectrum Scale service is stopped. If not, stop the IBM Spectrum Scale service. Click **Ambari > Spectrum Scale > Service Actions > Stop**.
5. Run the following command on the ambari server node acting as root:


```
$ cd /root/GPFS_Ambari
$ ./SpectrumScale_UpgradeIntegrationPackage-BI425 --preEU
```

The `--preEU` option saves the existing Spectrum Scale service information into JSON files in the local directory where the script was executed, and removes the Spectrum Scale service from the Ambari cluster so that the BI cluster can be upgrade properly. This does not affect the IBM Spectrum Scale file system.

Review the questions for the upgrade script below, and have the information for your environment handy before preceding. If Kerberos is enabled, additional inputs are required.

```
$ cd /root/GPFS_Ambari
$ ./SpectrumScale_UpgradeIntegrationPackage-BI425 --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:
...

```
6. Start all services. Click **Ambari > Actions > Start All**.
Wait for all the services to start. At this stage, native HDFS is used.
7. Update Ambari, BigInsights IOP and the other value add services.
Follow the BI express upgrade method as described in the BigInsights upgrade documentation:
 - a. Prepare for an express upgrade
 - b. Upgrade Ambari
 - c. Restart components to refresh configurations
 - d. Make configuration changes
 - e. Regenerate Kerberos keytabs

Note: BigInsights has a Snappy devel version that might not be compatible with the version that you currently have installed on your nodes. You can remove the Snappy package from all your nodes in the cluster and let BI reinstall the version that it uses. BI uses Snappy version 1.0.5-1.el6 (on RH7).

```
$ yum -y remove snappy*
```

- f. Follow the steps in Preparing for stack upgrade

Note: Perform the pre-requisites check carefully.

Note: Ensure that the HCAT client is added to the WebHCAT server node. If not, add the HCAT client to the WebHCAT server node. The express upgrade of IOP process fails during the UPGRADE install check phase if the HCAT client is not added to the WebHCAT server.

- g. Express upgrade of IOP
 - h. Upgrade Ambari Metrics
 - i. Upgrade Ambari alerts
 - j. Upgrade the IBM BigInsights value-add services
8. Stop all services. Click **Ambari > Actions > Stop All**. Wait till all services have stopped.
 9. Add the HDFS Transparency 2.7.3-0 into the GPFS repo directory.

Note: Ensure that the older HDFS Transparency version is removed from the repo directory. Only 1 HDFS Transparency rpm can reside in the GPFS repo directory.

Run **createrepo .** to update the repo metadata.

For more information, see Upgrading HDFS Transparency.

10. Add the IBM Spectrum Scale service.

Run the following command as root with the --preEU option on the Ambari server node, in the directory where the --preEU step was run, and the JSON configuration was stored.

```
$ cd /root/GPFS_Ambari
$ ./SpectrumScale_UpgradeIntegrationPackage-BI425 --postEU
```

For the --postEU option, review the questions and have the information for your environment handy before proceeding. If Kerberos is enabled, additional inputs are required.

```
$ ./SpectrumScale_UpgradeIntegrationPackage-BI425 --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 : 172.16.1.17
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:
...
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: Before starting all services in Ambari, ensure to run the
"Upgrade Transparency" service action in the Spectrum Scale service UI panel.
Prior to that, ensure that Repository for Spectrum Scale has been updated
to contain the version 2.7.3.0 of the gpfs.hdfs-protocol rpm package.
*****
```

Note: The script will trigger the installation and configuration of the Spectrum Scale service through the REST API calls, and will start the service after the installation completes. Monitor the Ambari UI for the progress of the service deployment.

Operations	Start Time	Duration	Show:
✓ Installing the GPFS Service using REST CALL	Today 20:17	83.23 secs	All (10)

When the process calls are completed in the UI, enter Y to complete the upgrade script. Entering Y will restart the Ambari server, complete the upgrade process and integrate the Spectrum Scale service.

Monitor to ensure that the Spectrum Scale service REST call process is completed from the Ambari GUI before entering Y to complete the Spectrum Scale service upgrade process. Entering Y will restart the Ambari server, complete the upgrade process, and integrate the IBM Spectrum Scale service.

11. Upgrade the HDFS transparency. This will apply the new HDFS Transparency from the GPFS repo to the HDFS Transparency nodes. For more information, see Upgrading HDFS Transparency.
12. Go to the <Spectrum Scale mount point>/<data directory>/iop/apps/ folder and ensure that the 4.2.5.0-0000 directory is the only directory existing under it. Delete or move other files or directories from this path.
13. Start all services. Click **Ambari > Actions > Start All**.
Restart all components using the **restart** icon.

Note: If the Spectrum Scale service is restarted 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 from BI IOP 4.1 with GPFS Ambari integration Hadoop Connector (1st gen) to BI IOP 4.2.5

This topic lists the steps to upgrade BI IOP 4.1 with GPFS Ambari integration Hadoop connector (1st gen) to BI IOP 4.2.5.

To upgrade to HDFS transparency (2nd gen) before attempting to upgrade to BI IOP 4.2.5, refer to the Upgrade IOP 4.1 + Ambari from Hadoop connector to HDFS Transparency Guide section in the IBM Spectrum Scale developerWorks Reference wiki.

After upgrading to HDFS Transparency, follow the Upgrading from BI IOP 4.1/4.2 with GPFS Ambari integration HDFS Transparency to BI IOP 4.2.5 process to upgrade to BI IOP 4.2.5 with IBM Spectrum Scale Ambari management pack and HDFS Transparency.

Upgrading from BI IOP 4.1 with Hadoop Connector (1st gen) to BI IOP 4.2.5

This topic lists the steps to upgrade BI IOP 4.1 with Hadoop connector (1st gen) to BI IOP 4.2.5.

If you have BI IOP 4.1 without GPFS Ambari integration with the Hadoop connector, contact scale@us.ibm.com for upgrade information.

Configuration

Setting up High Availability [HA]

You can set up high availability to protect against planned and unplanned events.

The process sets up a standby Namenode configuration so that failover can happen automatically.

Follow these steps to configure High Availability option when IBM Spectrum Scale service is integrated:

1. Log into the Ambari GUI.
2. If the Ambari GUI has IBM Spectrum Scale service deployed, and HDFS Transparency is integrated, follow the steps to Unintegrating Transparency. Ensure that the **ambari-server restart** is run.
Verify that the IBM Spectrum Scale HDFS Transparency integration state is in unintegrated state. See, Verifying Transparency integration state.
3. From the Ambari dashboard, click the HDFS service.
4. Select **Service Actions > Enable NameNode HA** and follow the steps.

Note: Namenode needs to be deployed onto a GPFS Node.

For more information on setting up the Namenode HA, see .

5. If the Ambari GUI has IBM Spectrum Scale service deployed, follow the steps under Integrating HDFS Transparency. Ensure that the **ambari-server restart** is run.

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
FileSystem Name		Advanced gpfs-advance: gpfs.quorum.nodes	The node number must be odd
FileSystem Mount Point			
NSD stanza file	See Preparing a stanza File		
Policy file	See Policy File		
Hadoop local cache disk stanza file	For more information, see create the local cache disk for Hadoop usage.		
Default Metadata Replicas	<= Max Metadata Replicas		
Default Data Replicas	<= Max Data Replicas		
Max Metadata Replicas			
Max Data Replicas			

Dual-network deployment

The following section is only applicable for IBM Spectrum Scale FPO (local storage) mode, and does not impact Hadoop clusters running over a shared storage configuration like a SAN-based cluster, or ESS.

If the FPO cluster has a dual 10 GB network, you have two configuration options. The first option is to bond the two network interfaces, and deploy the IBM Spectrum 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 Spectrum Scale to use for data traffic. This configuration can minimize interference between disk I/O and application communication.

For the second option, perform the following steps to ensure that the Hadoop applications can exploit data locality for better performance.

- Configure the first network interface with one subnet address, for example 192.168.1.0. Configure the second network interface with another subnet address, for example 192.168.2.0.
- Create the IBM Spectrum Scale cluster and NSDs with the IP or host name from the first network interface.
- Install the Hadoop cluster and the HDFS Transparency services by using the IP addresses or host names from the first network interface.
- Run the following command:

```
mmchconfig subnets=192.168.2.0 -N all
```

Note: 192.168.2.0 is the subnet used for IBM Spectrum Scale data traffic.

For Hadoop map and reduce jobs, the scheduler Yarn checks the block location. HDFS transparency returns the host name which is used to create an IBM Spectrum Scale cluster as a block location to Yarn. Yarn checks the host name within the NodeManager host list. If Yarn cannot find the host name within the NodeManager list, it cannot schedule the tasks according to data locality. The suggested configuration can ensure that the host name for block location is found in the Yarn NodeManager list, and Yarn can schedule the task according to data locality.

For a Hadoop distribution like IBM BigInsights IOP and HDP[®], all Hadoop components are managed by Ambari[™]. In this scenario, all Hadoop components, HDFS transparency, and the IBM Spectrum Scale cluster must be created by using one network interface. Use the second network interface for GPFS.

For more information, see [Deploying a big data solution using IBM Spectrum Scale](#).

- | For information on setting up the HDFS service Quicklinks Namenode UI access, see the [FAQ Quicklinks Namenode GUI](#) are not accessible from HDFS service in multihomed network environment.

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 Spectrum 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`

The following example uses RHEL 7.1:

```
# rpm -qa |grep httpd
httpd-tools-2.4.6-31.el7.x86_64
httpd-2.4.6-31.el7.x86_64

# service httpd start

# service httpd status
Redirecting to /bin/systemctl status httpd.service
httpd.service - The Apache HTTP Server
  Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled)
  Active: active (running) since Thu 2015-10-29 21:26:07 EDT; 6 months 6 days ago
  Process: 7400 ExecReload=/usr/sbin/httpd $OPTIONS -k graceful (code=exited, status=0/SUCCESS)
 Main PID: 8998 (httpd)
  Status: "Total requests: 0; Current requests/sec: 0; Current traffic:  0 B/sec"
   CGroup: /system.slice/httpd.service
├─ 6963 /usr/sbin/httpd -DFOREGROUND
├─ 6964 /usr/sbin/httpd -DFOREGROUND
├─ 7028 /usr/sbin/httpd -DFOREGROUND
├─ 8998 /usr/sbin/httpd -DFOREGROUND
├─15377 /usr/sbin/httpd -DFOREGROUND
├─19914 /usr/sbin/httpd -DFOREGROUND
├─19915 /usr/sbin/httpd -DFOREGROUND
├─20097 /usr/sbin/httpd -DFOREGROUND
├─20100 /usr/sbin/httpd -DFOREGROUND
├─20101 /usr/sbin/httpd -DFOREGROUND
├─21482 /usr/sbin/httpd -DFOREGROUND
....

# systemctl enable httpd
```

Local OS repository

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

1. Create the repository path:

```
mkdir /var/www/html/repos/<rhel_0Slevel>
```

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

```
cd /var/www/html/repos/<rhel_0Slevel>
```

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_0Slevel>
```

4. Create the repository for this node:

```
createrepo -v /var/www/html/repos/<rhel_0Slevel>
```

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_0Slevel>.repo.

7. Copy this file to all nodes. The contents of this file must look like the following:

```
[local_rhel7.2]
name=local_rhel7.2
enabled=1
baseurl=http://<internal IP that all nodes can reach>/repos/<rhel_0Slevel>
gpgcheck=0
```

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

Local IBM Spectrum Scale repository

IBM Spectrum Scale Express Edition can be used only if it is installed and configured manually before installing Ambari.

The following list of rpm packages for IBM Spectrum Scale v4.1.1 and later can help verify the edition of IBM Spectrum Scale:

IBM Spectrum Scale Edition	rpm package list
Express Edition Available up to version 4.2.2	gpfs.base gpfs.gpl gpfs.docs gpfs.gskit gpfs.msg.en_US gpfs.platform
Data Management Available from version 4.2.3 and above.	This edition provides identical functionality as IBM Spectrum Scale Advanced Edition under capacity-based licensing. For more information, see Capacity-based licensing.
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

The following example uses IBM Spectrum Scale 4.2.2.3 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/4.2.2.3
```

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 4.2.2.3 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-4.2.2.3-x86_64-Linux-install
```

```
./Spectrum_Scale_Advanced-4.2.2.3-x86_64-Linux-install --silent --dir /var/www/html/repos/GPFS/4.2.2.3
```

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/4.2.2.3`. Without specifying the `--dir` option, the default location is `/usr/lpp/mmfs/gpfs_rpms/4.2.X`.

3. If the packages are extracted into the IBM Spectrum Scale default directory, `/usr/lpp/mmfs/4.2.X/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/4.2.X/gpfs_rpms
```

```
cp -R * /var/www/html/repos/GPFS/4.2.2.3/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`
- `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-2.7.2-(version) /var/www/html/repos/GPFS/4.2.2.3/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:

```
createrepo /var/www/html/repos/GPFS/4.2.2.3/gpfs_rpms
```

```
# cd /var/www/html/repos/GPFS/4.2.2.3/gpfs_rpms  
# createrepo .
```

8. Access the repository at http://<yum-server>/repos/GPFS/4.2.2.3/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.

```
[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. Go to **HDFS Service > Configure > Advanced hdfs-logsearch-conf**

a. Replace the following:

```

{{default('/configurations/hadoop-env/hdfs_log_dir_prefix',
'/var/log/hadoop')}}/{{default('configurations/hadoop-env/hdfs_user',
'hdfs')}}/hadoop-{{default('configurations/hadoop-env/hdfs_user',
'hdfs')}}-namenode-*.log

```

with

```

{{default('/configurations/hadoop-env/hdfs_log_dir_prefix',
'/var/log/hadoop')}}/root/hadoop-root-namenode-*.log

```

b. Replace the following:

```

{{default('/configurations/hadoop-env/hdfs_log_dir_prefix',
'/var/log/hadoop')}}/{{default('configurations/hadoop-env/hdfs_user',
'hdfs')}}/hadoop-{{default('configurations/hadoop-env/hdfs_user',
'hdfs')}}-datanode-*.log

```

with

```

{{default('/configurations/hadoop-env/hdfs_log_dir_prefix',
'/var/log/hadoop')}}/root/hadoop-root-datanode-*.log

```

c. If HA is enabled, replace the following:

```
{{default('/configurations/hadoop-env/hdfs_log_dir_prefix',
'/var/log/hadoop')}}/{{default('configurations/hadoop-env/hdfs_user',
'hdfs')}}/hadoop-{{default('configurations/hadoop-env/hdfs_user',
'hdfs')}}-zkfc-*.log
```

with

```
{{default('/configurations/hadoop-env/hdfs_log_dir_prefix',
'/var/log/hadoop')}}/root/hadoop-root-zkfc-*.log
```

2. Perform **LOGSEARCH** service restart.
3. Perform **Ambari Server** restart.

Setting IBM Spectrum Scale configuration for BigSQL

In Ambari , the **gpfs.supergroup** is used to set the superuser for IBM Spectrum Scale service.

For BigSQL, the **gpfs.supergroup** requires to be set as *hdfs,root*. Otherwise, BigSQL impersonation fails.

Deploying IBM Spectrum Scale service

1. For new IBM Spectrum Scale service installation, during deployment at the installation configuration panel for IBM Spectrum Scale, find the **gpfs.supergroup** field and change the default values from *hadoop,root* to *hdfs,root*.
2. Proceed reviewing all the other fields to create the IBM Spectrum Scale deployment when returning back to Adding the IBM Spectrum Scale service section.

Existing IBM Spectrum Scale service

1. For existing IBM Spectrum Scale service installation, on the Ambari UI , click **Spectrum Scale > Configs > gpfs.supergroup** and change the default values of *hadoop,root* to *hdfs,root*.
2. Save Configuration.
3. Restart IBM Spectrum Scale service.
4. Restart HDFS service.

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 data node. 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 four, each node is on a different rack.
2. If the node number is greater than five, and node number is greater than 10, every two nodes are put in one virtual rack.
3. If the node number is greater than ten and node number is less than 21, every three nodes are put in one virtual rack.
4. If the node number is less than 22, 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
```

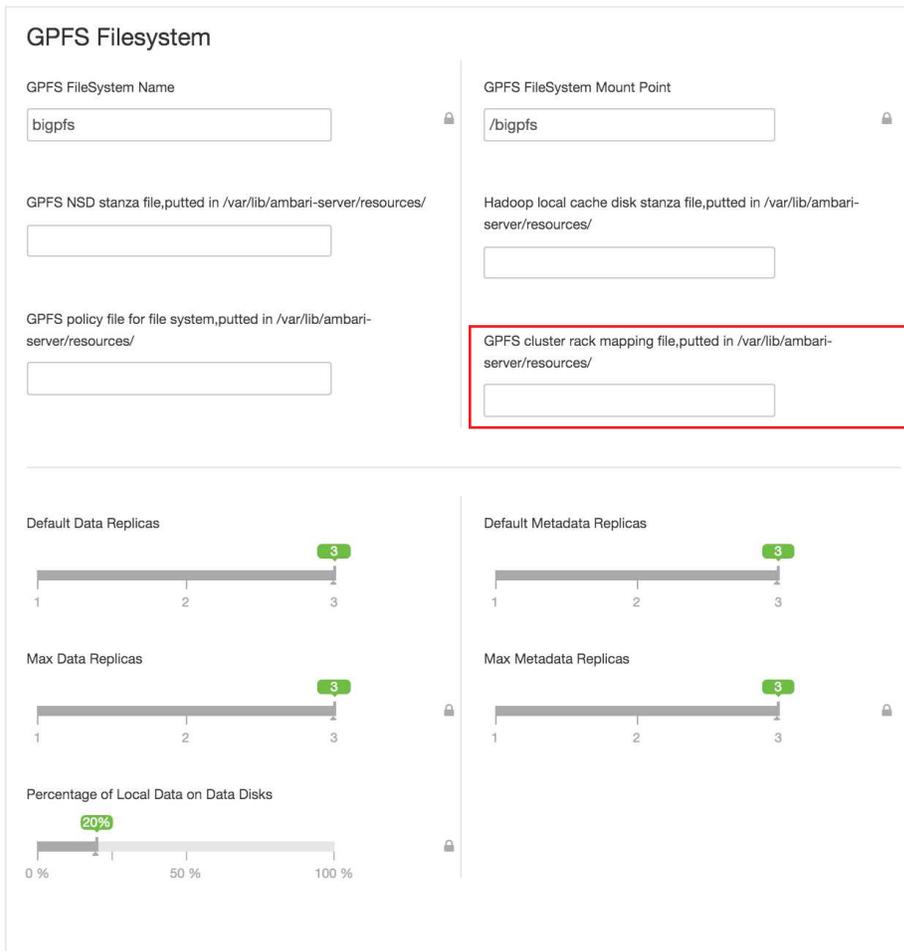


Figure 14. AMBARI RACK MAPPING

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 8. 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, found in the attachments at the bottom of the IBM Open Platform with Apache Hadoop wiki page, to partition the disks first, and specify the partition that is used for GPFS NSD in a simple NSD file.

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

Apache Ranger (<http://ranger.incubator.apache.org/>) is a centralized security administration solution for Hadoop.

Ranger enables administrators to create and enforce security policies for HDFS and other Hadoop platform components.

For more information on Ranger, see the Spectrum Scale HDFS Transparency Guide.

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
```

3. Configure audit for Ranger

- 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 `rangerlogger` user with password `YES` and grant the user adequate privileges:

```
CREATE USER 'rangerlogger'@'localhost' IDENTIFIED BY 'YES';
GRANT ALL PRIVILEGES ON *.* TO 'rangerlogger'@'localhost';
CREATE USER 'rangerlogger'@'%' IDENTIFIED BY 'YES';
GRANT ALL PRIVILEGES ON *.* TO 'rangerlogger'@'%';
GRANT ALL PRIVILEGES ON *.* TO 'rangerlogger'@'localhost' WITH GRANT OPTION;
GRANT ALL PRIVILEGES ON *.* TO 'rangerlogger'@'%' 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 rangerlogger -pYES
```

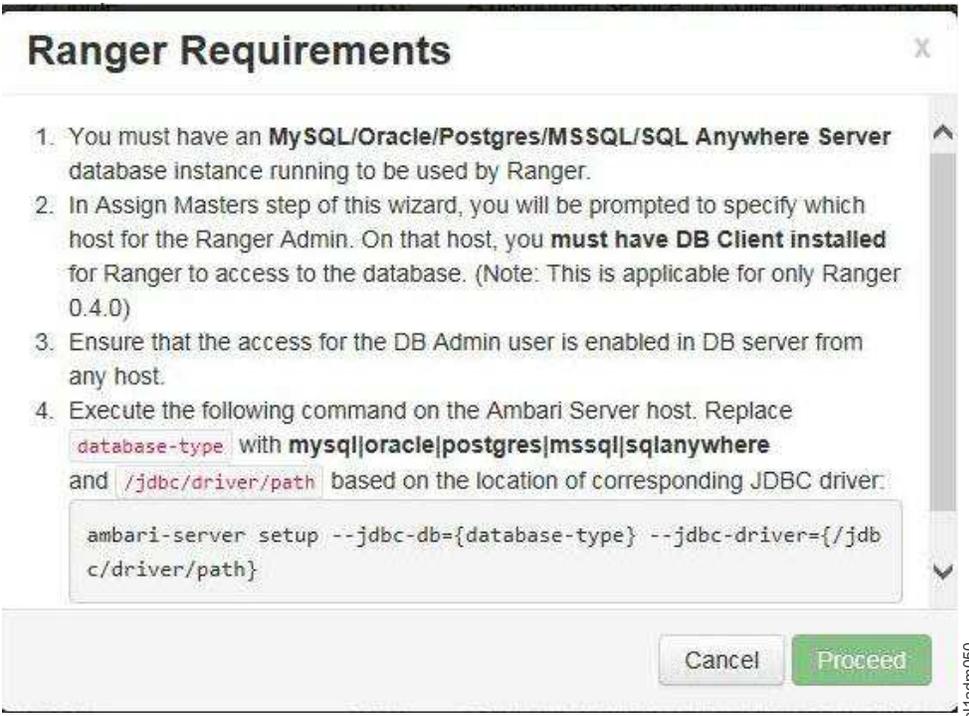
Installing Ranger through Ambari:

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**.



3. On the Choose Services page, select **Ranger**.
The system displays the Ranger Requirements page.



I have met all the requirements above.

Ensure that you have met all the installation requirements, then check the box for "I have met all the requirements above" before clicking **Proceed**.

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*.

Ranger Admin

DB FLAVOR

Ranger DB name

Ranger DB username

JDBC connect string

Ranger DB host

Driver class name for a JDBC Ranger database

Ranger DB password

bi1adm052

- For the Database Administrator (DBA) username, set the value to *rangerdba*.
- For the Database Administrator (DBA) password, set the value to *rangerdba*.
- Click on the Test Connection button and ensure that the connection result is OK.

Database Administrator (DBA) username

JDBC connect string for root user

Database Administrator (DBA) password

Connection OK 

bi1adm053

- For the Ranger Audit DB username, set the value to *rangerlogger*, and for the Ranger Audit DB password, set the value to *YES*.
- In the Ranger Audit tab, ensure that the Audit to Solr option is disabled.
- Click **Advanced tab** > **Advanced ranger-admin-site** and set the value of **ranger.audit.source.type** to *db*.

Ranger Admin | Ranger User Info | Ranger Plugin | Ranger Audit | **Advanced**

Audit to Solr

Audit to Solr

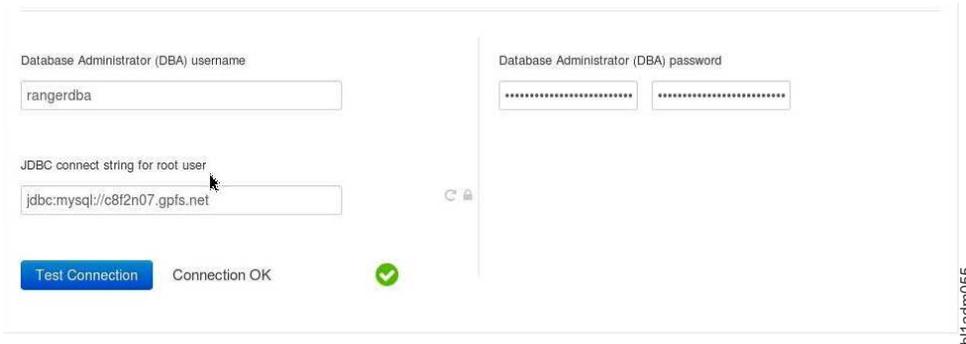
Audit to HDFS

Audit to HDFS

Destination HDFS Directory

bi1adv241

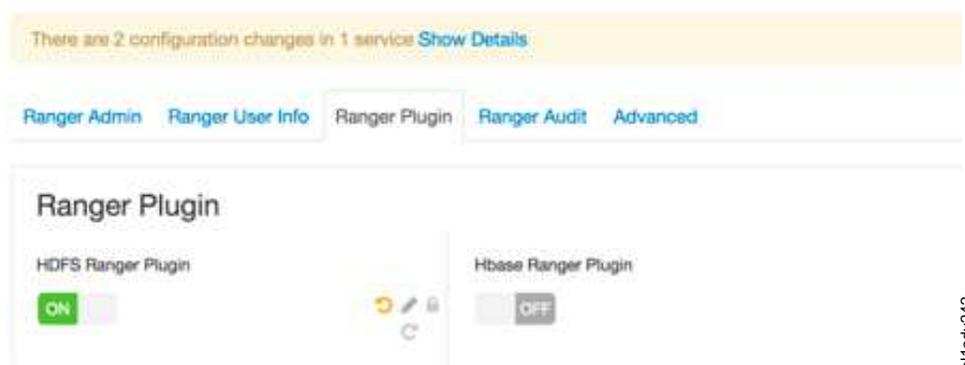
5. Deploy and complete the installation.
Assign the Ranger server to be on the same node as the HDFS Transparency namenode for better performance.
 - Select **Next > Next > Deploy**.
6. Test the connection.
 - On the Ranger dashboard, go to **Configs > Ranger Admin > Test connection**.



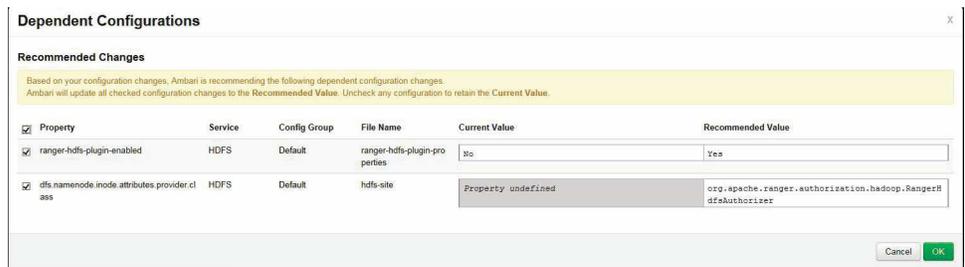
Enabling Ranger HDFS plug-in:

This topic lists the steps to enable Ranger HDFS plug-in

1. From the dashboard, click **Ranger > Configs > Ranger Plugin**, and switch on the **HDFS Ranger Plugin**.



You will get the following screen once you enable the HDFS Ranger Plugin. Click **Ok** to accept the recommended changes.



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. 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



Enabling Ranger auditing:

This section lists the steps to enable ranger auditing.

For information on enabling and configuring Ranger auditing, see [Enable Ranger Auditing](#).

Note:

1. In order to enable Audit to Solr for the Ranger plugins, ensure that the `xasecure.audit.destination.solr.zookeepers` field is set to `<host>:2181/solr`.
2. If you get the Unable to connect to the Audit store! message in the Ranger UI, see the FAQ Not able to view Solr audits in Ranger to remove the write locks from HDFS.

Disabling Ranger

If you do not plan to use Ranger, an option is provided to disable Ranger functionality in the HDFS Transparency to increase performance.

To disable Ranger in Ambari:

1. Log in to the Ambari GUI.
2. Select the **IBM Spectrum Scale service > Configs > Advanced > Custom gpfs-site > Add property** to set the key field to `gpfs.ranger.enabled`, and the value field 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.

Enabling Kerberos when Spectrum Scale service is not integrated:

IBM Spectrum Scale service (IBM Spectrum Scale Ambari management pack version) is not integrated into Ambari.

1. Follow Setting up KDC server and enabling Kerberos to enable Kerberos. This is before deploying IBM Spectrum Scale service in IBM Spectrum Scale service installation and Adding the IBM Spectrum Scale service to Ambari. 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

The screenshot shows the 'Customize Services' interface in Ambari. On the left is a sidebar with navigation options: 'Choose Services', 'Assign Masters', 'Assign Slaves and Clients', 'Customize Services' (selected), 'Configure Identities', 'Review', 'Install, Start and Test', and 'Summary'. The main area is titled 'Customize Services' and contains a list of services: HDFS, MapReduce2, YARN, Hive, HBase, Pig, Sqoop, Oozie, ZooKeeper, Flume, Titan, and Ambari Metrics. Below this is a 'Spectrum Scale' section with sub-sections: Kafka, Knox, Slider, Solr, Spark, and Misc. A 'Group' dropdown is set to 'Default (3)'. There are tabs for 'Standard' and 'Advanced'. Under the 'Advanced' tab, there are two expandable sections: 'Advanced gdfs-advance' and 'Advanced gdfs-ambari-server-env'. The 'Advanced gdfs-ambari-server-env' section contains several fields: 'AMBARI_USER' (admin), 'AMBARI_USER_PASSWORD' (masked), 'KDC_PRINCIPAL' (root/admin@IBM.COM, highlighted with a red box), 'KDC_PRINCIPAL_PASSWORD' (masked), and 'GPFS_REPO_URL' (http://c902mnx09.gpfs.net/repos/GPFS/4.2.2.3/gpfs_rpms). A vertical watermark 'bitadv243' is visible on the right side of the screenshot.

After all the required fields are set for the customized services panel, review all the fields in Customizing Services before clicking **NEXT** to view the Configure Identities panel.

The Configure Identities panel is displayed only in a Kerberos-enabled environment.

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.

Add Service Wizard

ADD SERVICE WIZARD

- Choose Services
- Assign Masters
- Assign Slaves and Clients
- Customize Services
- Configure Identities**
- Review
- Install, Start and Test
- Summary

Configure Identities

Configure principal name and keytab location for service users and hadoop service components.

General **Advanced**

Global

Admin principal: root/admin@IBM.COM

Admin password: [masked] [masked]

Save Admin Credentials ⓘ

Keytab Dir: /etc/security/keytabs

Realm: IBM.COM

Spnego Principal: HTTP/_HOST@\${realm}

Spnego Keytab: \${keytab_dir}/spnego.service keytab

bf1adv244

- Return to the previous step in Customizing Services tab to continue installation of the IBM Spectrum Scale service.

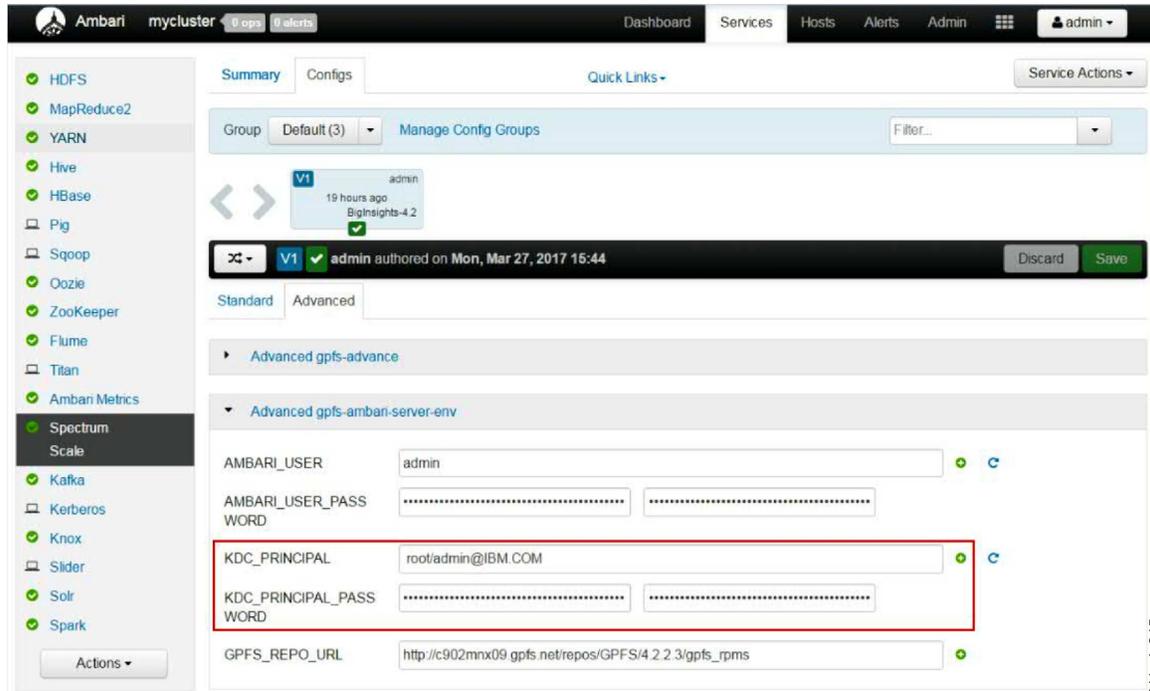
Enabling Kerberos when Spectrum Scale service is integrated:

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.

- Follow the steps in Setting up KDC server and enabling Kerberos section 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.

- 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.



3. Restart all the services. Click **Ambari panel > Service Actions > Stop All and Start All**.

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 BI, follow the BigInsights Setting up a KDC manually documentation.
 - 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.

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, ensure that you run this command 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.

Titan service check fails

If Titan service check fails:

1. On a Titan client, run the following command:
`/usr/bin/kinit -kt /etc/security/keytabs/hbase.service.keytab hbase/<HBASE_SERVER>@<REALM>`
For example, `/usr/bin/kinit -kt /etc/security/keytabs/hbase.service.keytab hbase/c902f05x04.gpfs.net@IBM.COM`
2. Remove the existing Titan table. For information on how to remove the existing Titan table, see the [How to fix the Titan service check failure?](#) section.

Journal nodes not installed in the native HDFS HA

If you are unintegrating from a Kerberos-enabled Namenode HA mode environment to native HDFS, see the [FAQ Journal nodes are not installed while unintegrating to native HDFS in a Kerberos-enabled enable namenode HA environment for a workaround for journal node.](#)

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 co-located 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.

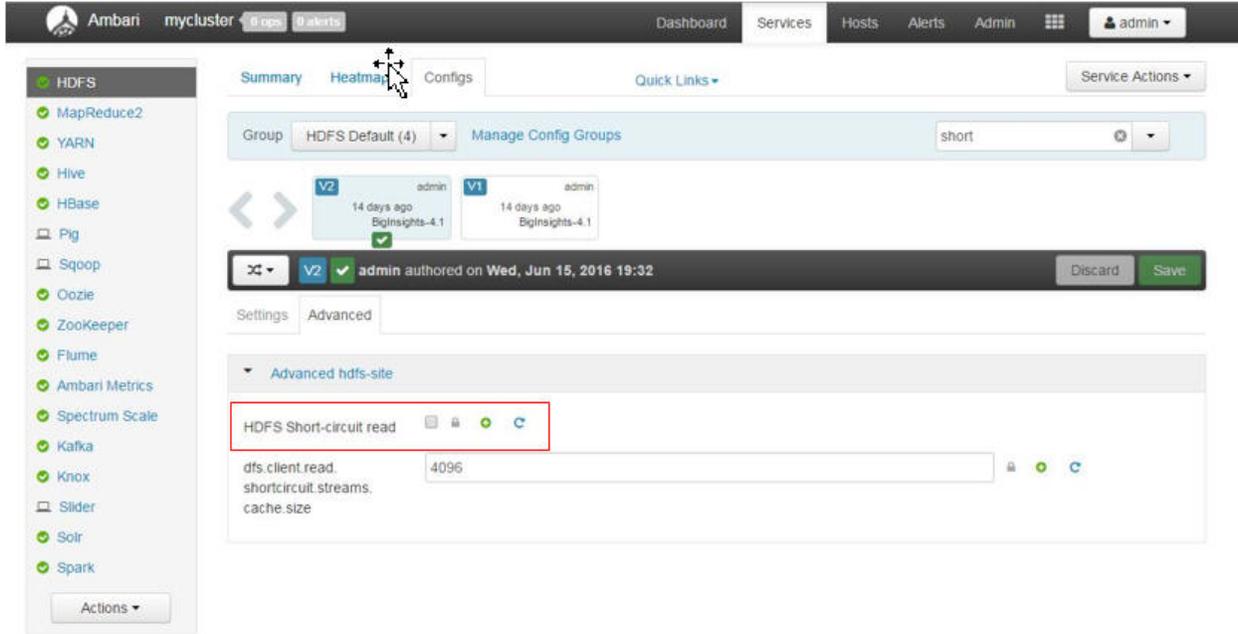
The short-circuit read is disabled by default in IBM Spectrum Scale Ambari management pack.

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 dash-board > 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**.



Note: When the short-circuit is enabled, SOLR generates a warning in the SOLR log (`/var/log/solr/solr.log`) on the SOLR server during the start of all processes if the SOLR `java.library.path` is not configured.

```
WARN - 2017-01-05 06:53:17.835; [c:titan s:shard2 r:core_node2 x:titan_shard2_replica1]
org.apache.hadoop.hdfs.shortcircuit.DomainSocketFactory; The short-circuit local reads feature
cannot be used because libhadoop cannot be loaded.
```

To fix the warning message in SOLR during the start process when SSR is enabled:

- Log into Ambari GUI
- Click **Solr service > Configs > Advanced solr-env > solr-env** template.
 - Add `-Djava.library.path=/usr/iop/4.2.5.0-0000/hadoop/lib/native` to the `SOLR_OPTS`

```
# Comment out the following SOLR_OPTS setting to config Solr to write its index and
transaction log files to local filesystem.
# Data (index and transaction log files) exists on HDFS will not be moved to local filesystem,
# after you change this config, they will not be available from local filesystem.
SOLR_OPTS="-Dsolr.directoryFactory=HdfsDirectoryFactory \
-Dsolr.lock.type=hdfs \
-Dsolr.hdfs.confdir=/etc/hadoop/conf \
-Dsolr.hdfs.home={{fs_root}}{{solr_hdfs_home_dir}} \
-Dsolr.hdfs.security.kerberos.enabled={{sole_kerberos_enabled}} \
-Dsolr.hdfs.security.kerberos.keytabfile={{solr_keytab}} \
-Dsolr.hdfs.security.kerberos.principal={{solr_principal}} \
-Dsolr.log4j.dir={{log_dir}} \
-Djava.library.path=/usr/iop/4.2.5.0-0/hadoop/lib/native"
```
 - Save the configuration.
- Restart SOLR from Ambari GUI.

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. Go to **Ambari GUI > HDFS > Custom hdfs-site**, add the **gpfs.short-circuit-write.enabled=false** property, and save the configuration.
3. Restart IBM Spectrum Scale service.
4. Restart HDFS service.
5. 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 to adapter when you run a teragen job.

IBM Spectrum Scale service management

Manage the IBM Spectrum Scale through the Spectrum Scale dashboard. The status and utilization information of IBM Spectrum Scale and HDFS Transparency can be viewed on this panel.

Service Actions dropdown list

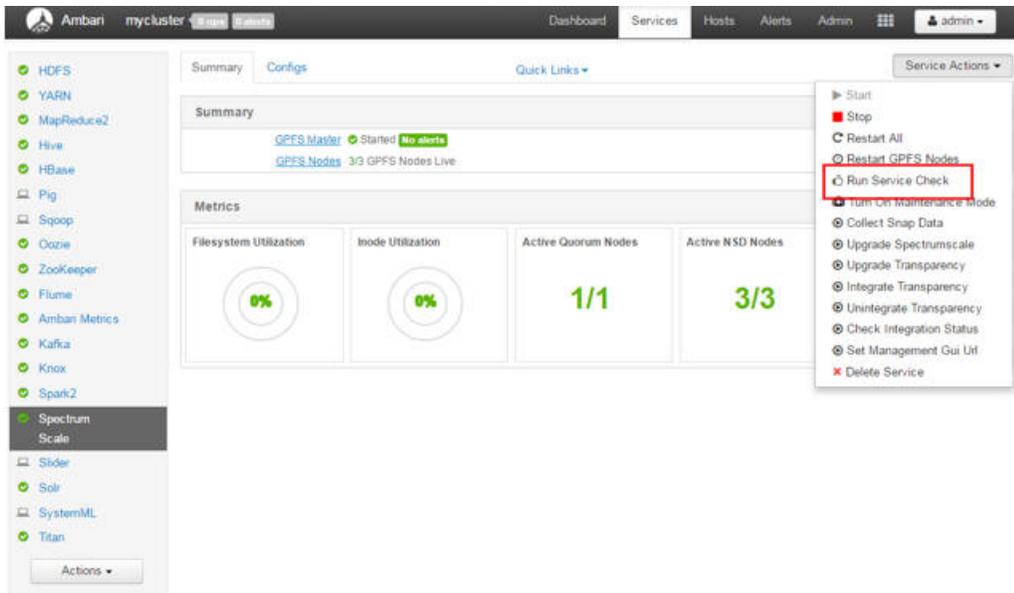
To go to the Service Actions dropdown list, click **Spectrum Scale > Service Action**.

The screenshot displays the Ambari GUI interface for managing the IBM Spectrum Scale service. The top navigation bar includes 'Ambari mycluster', 'Dashboard', 'Services', 'Hosts', 'Alerts', 'Admin', and a user profile 'admin'. The left sidebar lists various services, with 'Spectrum Scale' selected. The main content area shows the 'Summary' and 'Metrics' sections. The 'Summary' section indicates that the 'GPFS Master' is 'Started' with 'No alerts', and there are '3/3 GPFS Nodes Live'. The 'Metrics' section contains four gauges: 'Filesystem Utilization' at 0%, 'Inode Utilization' at 0%, 'Active Quorum Nodes' at 1/1, and 'Active NSD Nodes' at 3/3. A 'Service Actions' dropdown menu is open on the right, listing various actions such as Start, Stop, Restart All, Run Service Check, Turn On Maintenance Mode, Collect Snap Data, Upgrade Spectrumscale, Upgrade Transparency, Integrate Transparency, Unintegrate Transparency, Check Integration Status, Set Management Gui Url, and Delete Service.

Note: For the Delete Service, only the IBM Spectrum Scale service is deleted from Ambari. 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 Service 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 > Service Actions > Turn on Maintenance Mode.

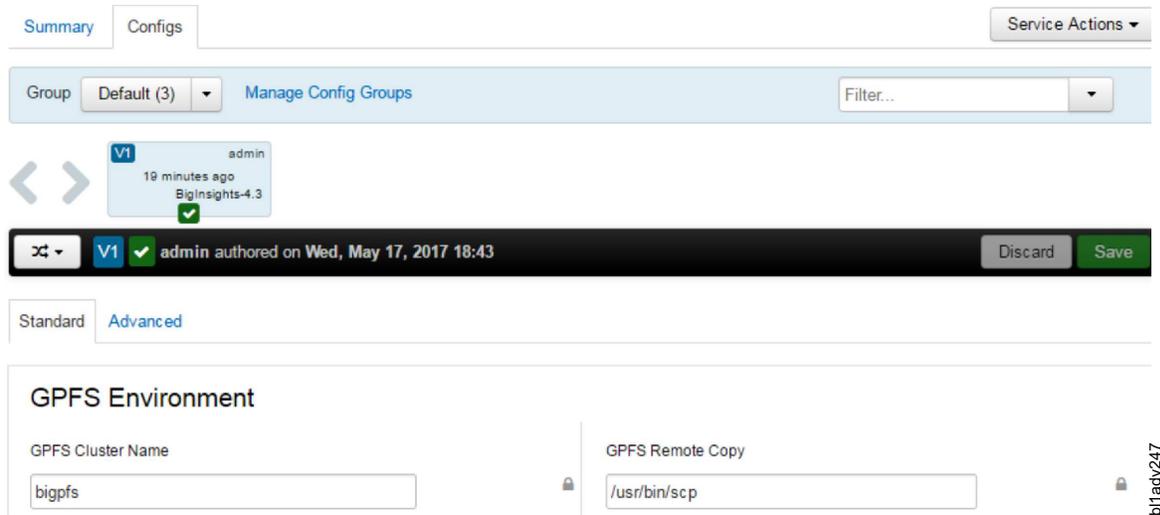
This prevents any Ambari actions from occurring on the service that is in maintenance mode.

To get out of Maintenance Mode, click Ambari GUI > Spectrum Scale service > Service Actions > Turn off Maintenance Mode.

Modifying IBM Spectrum Scale service configurations

The IBM Spectrum Scale service has standard and advanced configuration panels.

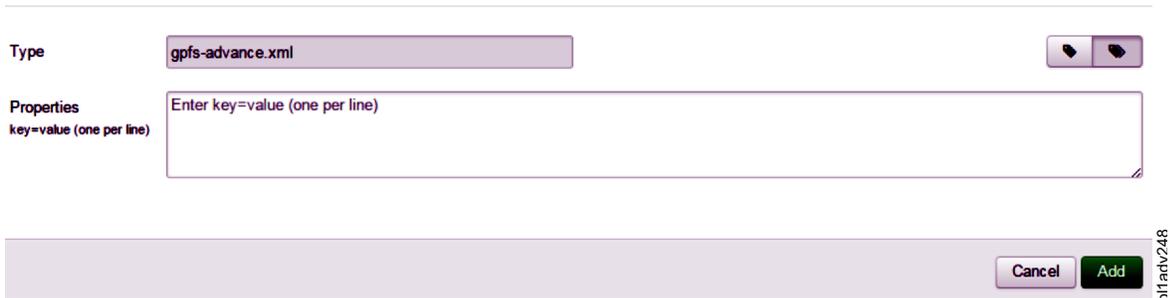
Click Ambari GUI > Spectrum Scale > Configs tab.



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



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. A reboot is required for recovery.

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_REPO_URL` in Ambari:

1. Log into 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/4.2.2/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

Starting from management pack 4.1-1 and later, the HDFS Transparency Namenode and Datanode are the same as the HDFS Namenode and Datanode in Ambari HDFS service.

When configuration is changed in IBM Spectrum Scale, the IBM Spectrum Scale service must be restarted first and then 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.

Note: Do not integrate the HDFS transparency more than once consecutively. Unpredictable errors will occur, which would cause the cluster to be in an unusable state. Contact scale@us.ibm.com if this occurs.

To integrate the HDFS Transparency (GPFS Transparency Node) with the native HDFS:

1. On the dashboard, click **Actions** > **Stop All** to stop all services.
2. On the IBM Spectrum Scale dashboard, click **Service Actions** > **Integrate Transparency**.
3. Verify that all services are stopped. If not, stop the services.

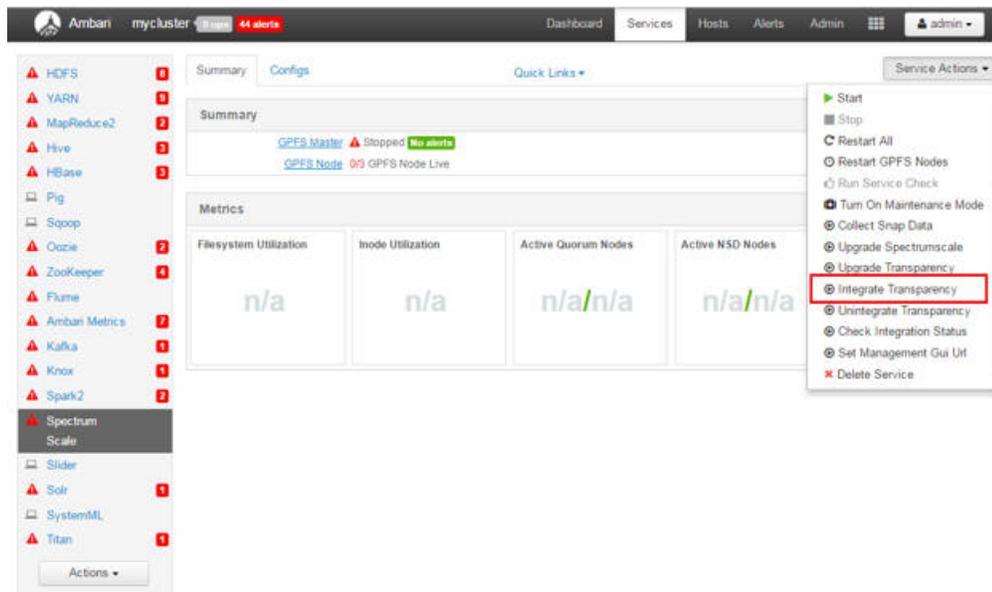
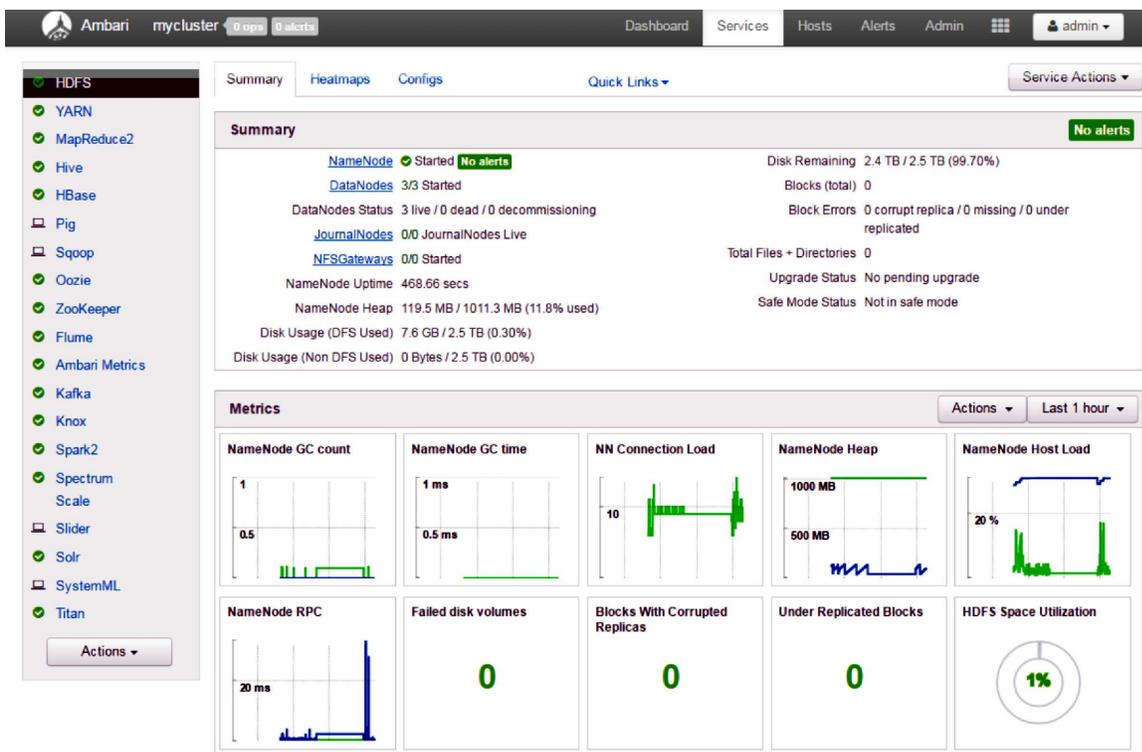


Figure 15. IBM SPECTRUM SCALE INTEGRATE TRANSPARENCY

4. On the Ambari server node, run the **ambari-server restart** command to restart the Ambari server.
5. Log back in to the Ambari GUI.
6. 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.



Command verification

To verify that the HDFS Transparency is available, use the following command to check the connector state:

```
[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

```
[root@c902f05x01 ~]#
```

```
[root@c902f05x01 ~]# /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.
```

```
[root@c902f05x01 ~]#
```

Check that Spectrum Scale is integrated

For more information on how to verify the HDFS transparency integration state, see “Verifying Transparency integration state” on page 179

Cluster environment

When the IBM Spectrum Scale service is deployed, IBM Spectrum Scale is used instead of HDFS. IBM Spectrum Scale inherits the native HDFS configuration, and adds the additional changes for IBM Spectrum Scale to function correctly.

After IBM Spectrum Scale is deployed, a new HDFS configuration set V2 is created, and is visible in the HDFS UI Panel > Configs tab.

Unintegrating 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.

Note: Do not run the Unintegrate HDFS Transparency more than once consecutively. Unpredictable errors will occur, which would cause the cluster to be in an unusable state. Contact scale@us.ibm.com if this occurs.

1. On the dashboard, click **Actions > Stop All** to stop all services.
2. Click **Spectrum Scale > Service Actions > Unintegrate Transparency**.

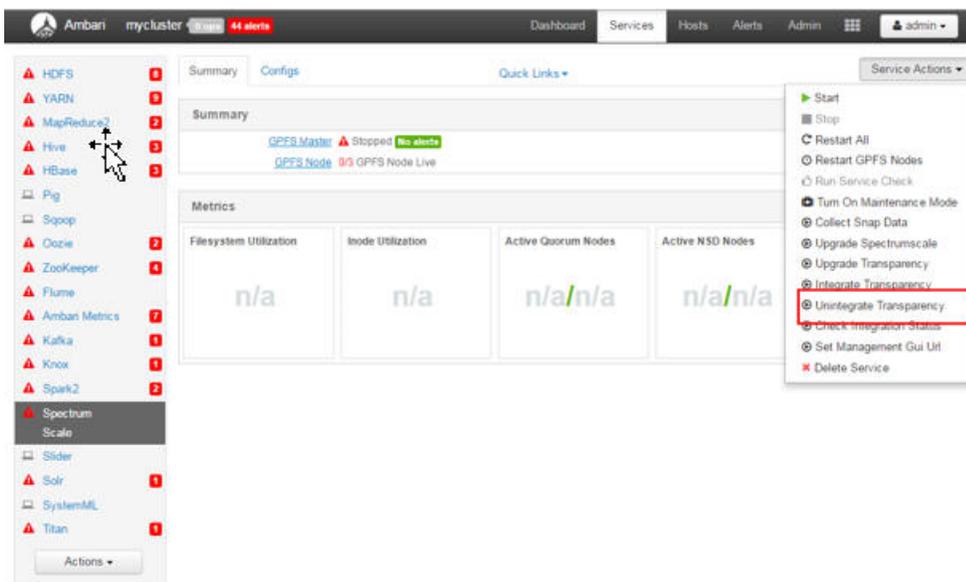


Figure 16. IBM SPECTRUM SCALE UNINTEGRATE TRANSPARENCY

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 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:

```
[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

```
[root@c902f05x01 ~]#
```

```
[root@c902f05x01 ~]# /usr/lpp/mmfs/hadoop/sbin/mmhadoopctl connector getstate
c902f05x01.gpfs.net: namenode is not running.
c902f05x03.gpfs.net: datanode is not running.
c902f05x02.gpfs.net: datanode is not running.
c902f05x01.gpfs.net: datanode is not running.
c902f05x04.gpfs.net: datanode is not running.
[root@c902f05x01 ~]#
```

```
# Check for unintegrated state
Verify IBM Spectrum Scale HDFS Transparency integration state
```

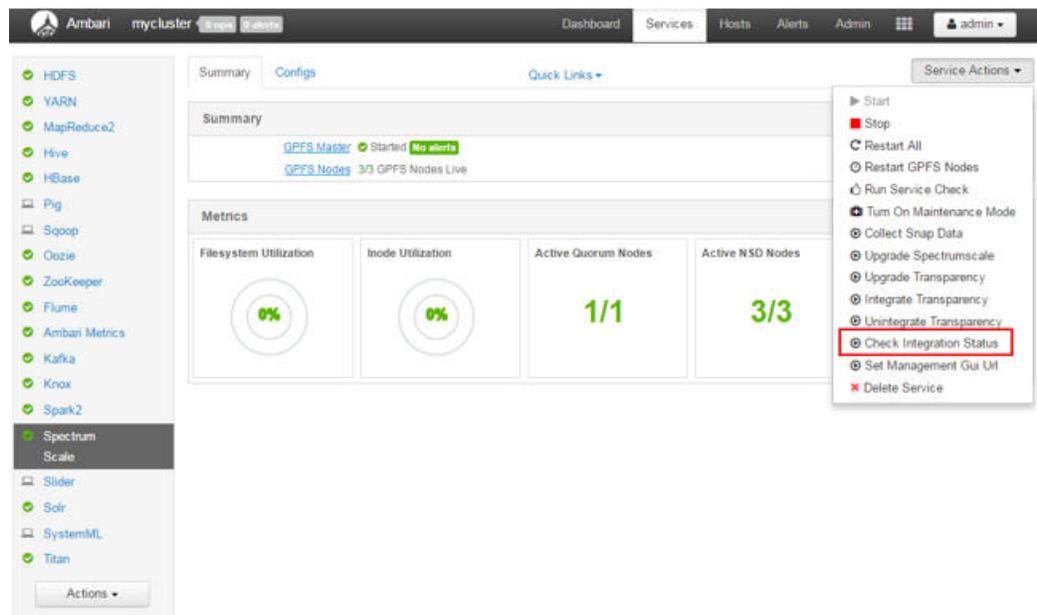
Cluster environment

After using the Spectrum Scale Unintegrate Transparency function, the native HDFS will be in effect. The configuration from IBM Spectrum Scale 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.”

Verifying Transparency integration state

To verify the HDFS Transparency integration state, click **Ambari GUI > Spectrum Scale > Service Actions > Check Integration Status**.



After the process completes, check the output log for the state information.

stderr: /var/lib/ambari-agent/data/errors-252.txt

None

stdout: /var/lib/ambari-agent/data/output-252.txt

2017-05-31 12:02:33,255 - ===== Spectrum Scale service is INTEGRATED. =====

Command completed successfully!

 Do not show this dialog again when starting a background operation

OK

b11adv254

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 node.

See Preparing the environment section to prepare the new nodes.

Note: If you are adding new nodes to an existing cluster, and if the nodes being added already have IBM Spectrum Scale installed on them, then 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](#).

Add the new nodes to the Ambari cluster by using the Ambari web interface.

1. From the dashboard, click **Hosts** > **Actions** > **Add New Hosts**.
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.

3. Click **next** when ready to go to the next step.
4. Select the services that you want to install on the new node in the Assign Slaves and Clients panel. For IBM Spectrum Scale and HDFS Transparency, the GPFS Node, and Data Node column is required to be checked.

Note:

- All HDFS namenodes and datanodes must be GPFS Nodes.
- HDFS Transparency Datanode is required to be a Hadoop Datanode, NodeManager, and GPFS Node.

5. Click **next** when ready to go to the next step.

6. If several configuration groups are created, select one of them for the new node, or use the default value.
7. Click **next** when ready to go to the next step.
8. Review the information, and start the deployment by clicking Deploy.
9. After selected services are installed and started with success, click **Next** to go to next step. If some services are not able to start, you can manually start them once you exited the Add Host Wizard panels.
10. Review the summary of the installed process, and click **Complete** to finish the Add Host Wizard.
11. A new host is added to the Ambari cluster.

From Hosts dashboard, verify the newly added node in the host list.

12. If the Ambari server is non-root, and the newly added host contains a datanode component, an HDFS restart is required to start the new datanodes. Plan a cluster maintenance window and prepare for cluster downtime when restarting HDFS service. From **HDFS > Service Actions > Restart All**.

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 in the ADD section in Deploying a big data solution using IBM Spectrum Scale.

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	172.16.1.11	c902f05x01.gpfs.net	quorum
2	c902f05x04.gpfs.net	172.16.1.17	c902f05x04.gpfs.net	quorum
3	c902f05x03.gpfs.net	172.16.1.15	c902f05x03.gpfs.net	quorum
4	c902f05x02.gpfs.net	172.16.1.13	c902f05x02.gpfs.net	
5	c902f05x05.gpfs.net	172.16.1.19	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/mmlnsd
```

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 ~]#

```

Deleting a host

This topic provides information on how to delete a node.

Decommissioning a Datanode is not supported by the Ambari GUI that is installed with the management pack version 4.1-X and later.

Starting from version 4.1-1, deleting a node is now supported in the management pack.

1. Stop all the components on the node to be deleted. For example: c902f05x05
2. From Ambari dashboard, click **Hosts tab**, and click **the host that must be removed > Host Actions > Stop All Components**.

Wait for the processes to be completed.

3. Issue the following commands to stop the Ambari agent on the host to be deleted.

```

[root@c902f05x05 ~]# 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@c902f05x05 ~]#

```

4. To delete the host, click **Host Actions > Delete Hosts**.

The system displays a Warning message.

5. Click **OK**.

The node is deleted from the Hosts list.

Note: The HDFS alert is displayed because the deleted node is not cleaned from the cluster yet. The DataNode is 4/4 started, but the DataNodes Status is 5 live nodes.

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 > Service Actions > Restart All**

A confirmation warning message will appear to confirm if you want to turn on the Maintenance mode.

Click **Confirm 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 nodes.

Note: This does not remove the Ambari packages and the IBM Spectrum Scale packages and disks. Follow the steps in the Uninstalling Ambari stack section to remove Ambari from the node. Follow the IBM Spectrum Scale documentation on removing disks and packages from the environment.

```
[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	down

```
[root@c902f05x01 ~]#
```

```
[root@c902f05x01 ~]# /usr/lpp/mmfs/hadoop/sbin/mmhadoopctl connector getstate
c902f05x01.gpfs.net: namenode running as process 3413.
c902f05x01.gpfs.net: datanode running as process 29488.
c902f05x04.gpfs.net: datanode running as process 24456.
c902f05x03.gpfs.net: datanode running as process 15439.
c902f05x02.gpfs.net: datanode running as process 17884.
[root@c902f05x01 ~]#
```

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 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.

Move Namenode in integrated state:

This section provides the steps to move Namenode into an integrated state.

Instructions for HA cluster:

This topic describes the steps to manually move the Namenode when HDFS Transparency is in integrate state.

1. From the dashboard, select **Actions > Stop All**.
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-2.4.2.0/extensions/  
SpectrumScaleExtension/2.4.2.0/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**.
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/  
MoveNameNodeTransparency.py
```

Follow the command prompts and type the required input.

```
$ python /var/lib/ambari-server/resources/mpacks/SpectrumScaleExtension-MPack-2.4.2.0/  
extensions/SpectrumScaleExtension/2.4.2.0/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 Transparency 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 an HA enabled environment, then follow the HA steps. 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'
```

 - a. Log in to Ambari.
 - b. From the dashboard, select **Actions > Start All**. The Hadoop cluster will now use the native HDFS.

If non-HA is enabled, perform the following steps, for example:

Name Node 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**

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.
 - The cluster must be at Mpack version 2.4.2.1 or later.
 - 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 into 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 > Service 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 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:
| ...
| 6. Run the MPack uninstaller script to remove the existing MPack link.
|   $./SpectrumScaleMPackUninstaller.py
| 7. Move the Ambari server to the new host as documented in the 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 :
| 172.16.1.17
| 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:
| ...
| 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
| Waiting for the Spectrum Scale service to be completely installed.
| Restarting Ambari server
| 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 found warnings. See /var/log/ambari-server/ambari-server-check-database.log for more details.
| *****
| Upgrade of the Spectrum Scale Service completed successfully.
| Make sure to restart the Ambari server if not done as part of this script after Spectrum Scale service installation.....
| *****
| *****
| IMPORTANT: You need to ensure that the HDFS Transparency package, gdfs.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.
| *****

```

12. Start all services by clicking **Ambari > Actions > Start All**.

Restart all components by using the restart icon.

Note:

- If the 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.

Adding GPFS node component

For an existing IBM Spectrum Scale, and the HDFS Transparency node where the GPFS Node column was not set for that host in the Assign Slaves and Clients panel, set the node to GPFS Node in Ambari.

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 > Service Actions > Stop > Start**.

Note: The NameNode is required to be restarted before datanodes.

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.

Follow the Planning section first, 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 will be used 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 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 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.

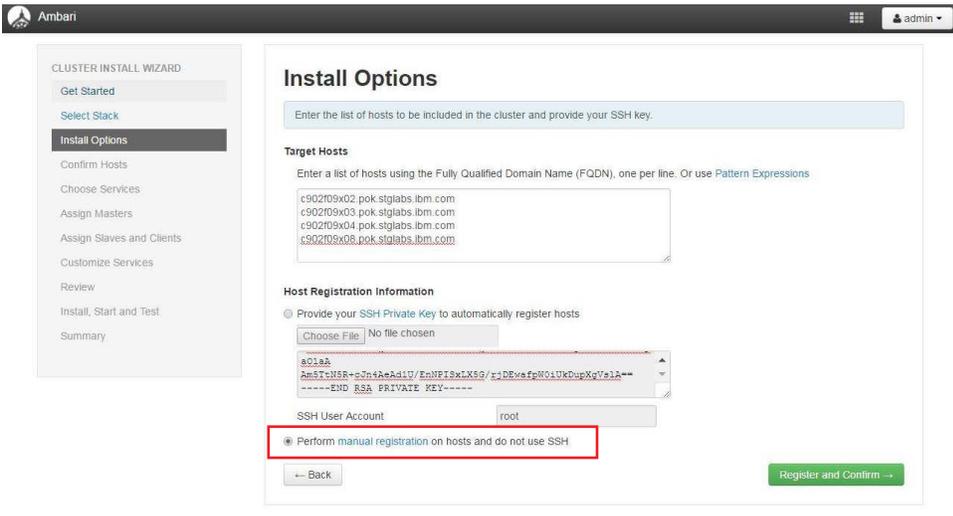
3. Set up an Ambari cluster as the non-root user.

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.



4. Configuring IBM Spectrum Scale without remote root by following the steps in IBM Spectrum Scale Administration and Programming Reference documentation under section Running IBM Spectrum Scale without remote root login - Configuring sudo in Running IBM Spectrum Scale without remote root login.

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 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
Cmd Alias BIGSQL_SERVICE_AGNT=/var/lib/ambari-agent/cache/stacks/BigInsights/*/services/
BIGSQL/package/scripts/*
Cmd Alias BIGSQL_SERVICE_SRVR=/var/lib/ambari-server/resources/stacks/BigInsights/*
/services/BIGSQL/package/scripts/*
Cmd 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-datasrvmgr/bin/*,
/usr/ibmpacks/bin/*/*
Cmd 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
GPFS_CMDPOR-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 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 accumulio *,/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 += "MMMMode 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 IBM Spectrum Scale service installation 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 Adding the IBM Spectrum Scale service to Ambari. This requires restarting Ambari as root. Exceptions occurs as non-root user. However, this issue is not shown on Ambari 2.5.0.3 when ambari-server restart with non root user.

Note:

- There might be an issue with HBase stopping in a non-root environment. For more information, see the FAQ section.
- In non-root Ambari environment, the Hive service check might fail. For resolution, see the FAQ section.

IBM Spectrum Scale management GUI

The IBM Spectrum Scale management GUI quick link is not a part of the GPFS Ambari integration module version 4.1-X and version 4.2-0.

The IBM Spectrum Scale management GUI can be manually installed and accessed.

Installation instructions for IBM Spectrum Scale management GUI are available on IBM Knowledge Center here: [Manually installing IBM Spectrum Scale management GUI](#).

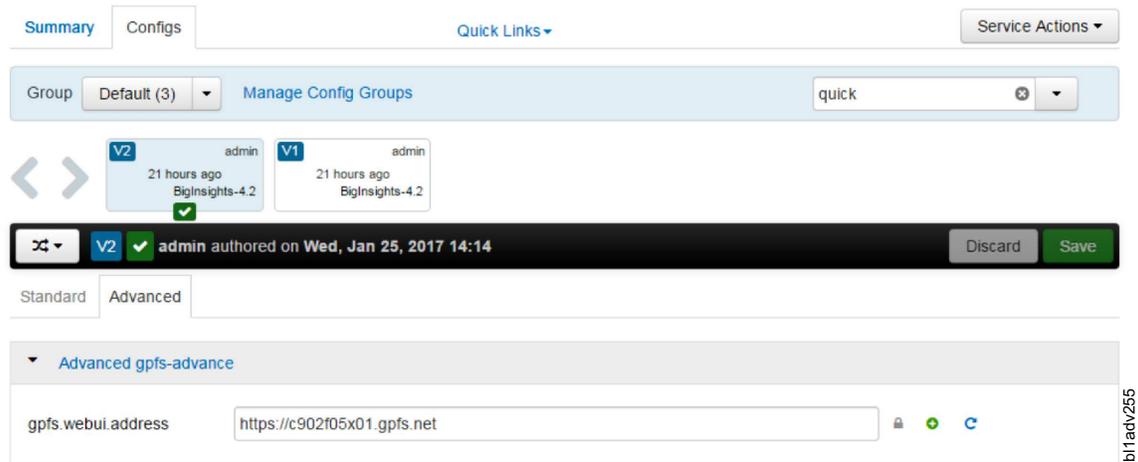
In GPFS Ambari integration module version 4.2-1 and later, the IBM Spectrum Scale management GUI quick link is available in Ambari.

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.

If you are running IBM Spectrum Scale 4.1, there is an Open Beta of the GUI available here: <https://www.ibm.com/developerworks/servicemanagement/tc/gpfs/evaluate.html>.

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.



3. Restart the IBM Spectrum Scale service
4. Sync the configuration.
Click **Ambari GUI > IBM Spectrum Scale service > Service Actions > Set Management GUI URL**.

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 Spectrum Scale is 3.
- 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.
- HDFS encryption is not supported by HDFS Transparency, instead use the IBM Spectrum Scale encryption.
- 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 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+ and 2.7.3-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 versions 2.7.2-3+ and 2.7.3-x:

- If you did not export `HADOOP_CONF_DIR`, then 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`, 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 `/usr/lpp/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.

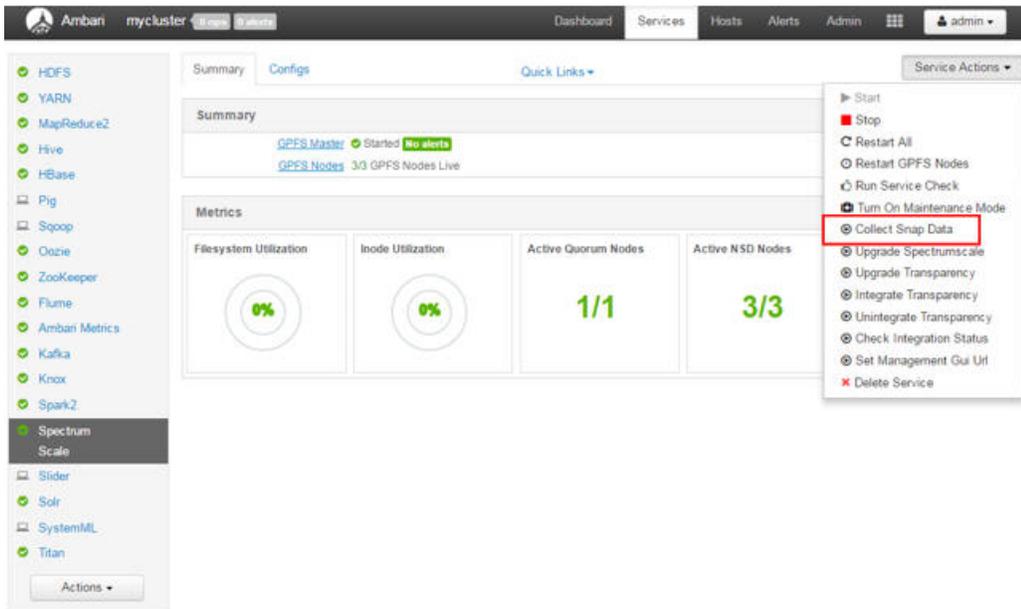
Table 9. NATIVE HDFS AND IBM SPECTRUM SCALE DIFFERENCES

Property name	Value	New definition or limitation
<code>dfs.permissions.enabled</code>	True/false	For HDFS protocol, the permission check is always done.
<code>dfs.namenode.acls.enabled</code>	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.
<code>dfs.blocksize</code>	Long digital	Must be a multiple of the IBM Spectrum Scale file system block size (<code>mmfsfs -B</code>). The maximal value is $1024 * \text{file-system-data-block-size}$ (<code>mmfsfs -B</code>).
<code>gpfs.data.dir</code>	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 <code>gpfs.mount.dir</code> .
<code>dfs.namenode.fs-limits.max-xattrs-per-inode</code>	INT	Does not apply to the HDFS protocol.
<code>dfs.namenode.fs-limits.max-xattr-size</code>	INT	Does not apply to the HDFS protocol.
<code>dfs.namenode.fs-limits.max-component-length</code>	INT	Does not apply to HDFS Transparency. The file name length is controlled by IBM Spectrum Scale. Refer to Spectrum Scale FAQ for file name length limit.
Native HDFS encryption	Not supported	Customers should take native Spectrum Scale encryption.
Native HDFS caching	Not supported	Spectrum Scale
NFS Gateway	Not supported	Spectrum Scale provides POSIX interface and taking Spectrum Scale protocol could give you better performance and scaling.

Troubleshooting

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:

- From GPFS Ambari integration module 4.2-1, if your cluster has IBM Spectrum Scale file system version 4.2.2.0 and later, **gpfs.snap** will include the `--hadoop` option.
- From Mpack 2.4.2.6, 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 > 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
```

| From management pack version 2.4.2, PTF6 snap data for ambari and its services is also captured with
| the **ambari_mpack_snap.sh** command. 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/



Figure 17. AMBARI COLLECT SNAP DATA

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 management pack 2.4.2.0 requires HDFS Transparency version 2.7.3.0 or later. The GPFS™ Ambari integration mpack 2.4.2.1 requires HDFS Transparency version 2.7.3.1 or later. The HDFS service must not be started until HDFS Transparency version 2.7.3.X is installed.
- 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.
- IBM Spectrum Scale management pack version 2.4.2.0 does not support Platform Symphony®.
- 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 HDFS Transparency Guide. For support on Federation for your environment, contact scale@us.ibm.com.

Note:

- BigInsights and Hortonworks do not support Federation.
- Ambari does not support Federation - JIRA AMBARI-10982.
- The latest JDK supported version for Ambari is 1.8.0.77. Currently, JDK versions after 1.8.0.77 broke existing JSP code, creating Ambari java exceptions.
- If not using the OpenJDK from IOP-UTILS, then the Java OpenJDK is required to be installed on all the nodes in the Hadoop cluster.
- 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 /usr/lpp/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 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 CentOS, create the /etc/redhat-release file to simulate a RedHat environment. Otherwise, the Ambari deployment fails.

For example:

```
# cat redhat-release
Red Hat Enterprise Linux Server release 7.1 (Maipo)
```

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 co-located. 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 co-located. If Ambari server and GPFS Master are not co-located, then click **Ambari GUI > Spectrum Scale > Service Actions > Delete Service** to remove the IBM Spectrum Scale service. Then redeploy the IBM Spectrum Scale service.
- 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 IOP or 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.
 - 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 IOP or 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.

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 `mmfsconfig` 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 Namenode CPU WIO metric in the Ambari GUI might not be displayed due to jira AMBARI-1614.
- The **IBM Spectrum Scale Service UI Panel > Service 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.
- The Datanode Unmounted Data Dir alert is activated because the native HDFS Datanode directories are not created in IBM Spectrum Scale. The alert can be ignored. See the FAQ Datanode Unmounted Data Dir alert after adding a host into the Spectrum Scale cluster.
- Decommissioning a Datanode is not supported in the management pack from version 4.1-X and above.
- Moving a Namenode from the Ambari HDFS UI when HDFS Transparency is integrated is not supported in the management pack version 4.1-X and above. 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 `mm1sfs <Device> -k` to ensure the `-k` value is set to `all`.
- For IBM Spectrum Scale Ambari management pack version 2.4.2.0 and GPFS Ambari integration module version 4.2.1 and earlier, the `gpfs.gss` package for monitoring needs to be installed but not configured on the node that is specified in the `shared_gpfs_node.cfg` file for Ambari to setup the shared mode for ESS correctly.

BI IOP

- Only the BI **express upgrade** procedure is supported with IBM Spectrum Scale service.

HDP

- The Manage JournalNodes is shown in **HDFS > Service Actions** submenu. This function should not be used when IBM Spectrum Scale service is deployed.
- The + is not supported when using `hftp://namenode:50070`.

FAQ

This section lists the FAQs.

For more issues, see Troubleshooting Ambari.

General

1. 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 IOP** over existing IBM Spectrum Scale cluster mode.

2. 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.

3. 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 IOP and IOP-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, IOP, or IOP-UTILS yum URL.

4. 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.

5. How to check the superuser and the supergroup?

Solution: If you are using the connector version `hadoop-gpfs-2.7.0-3` or later, additional security controls are added to support multiple user groups. Normally, just one super user “hdfs” and super group “hadoop” is used. The IDs that can access the distributed file system via HDFS is controlled by permissions and ACLs defined on `/var/run/ibm_bigpfs_gcd`. To see the superuser or the super group:

```
ls -alt /var/run/ibm_bigpfs_gcd
srw-----. 1 hdfs hadoop 0 Dec 10 21:17 /var/run/ibm_bigpfs_gcd
```

6. How to set user permissions in the file system?

Solution: Create some directories to support the new connector, if they do not already exist.

```
mkdir /var/mmfs/bi;  
chown hdfs:hadoop /var/mmfs/bi;  
chmod 660 /var/mmfs/bi
```

In this example, the HDFS superuser is *hdfs* and the super group is *hadoop*.

- a. To allow a specific set of users to access the DFS via ACLs, perform the following on all nodes:

Note: For HDFS ACL support, install the following rpm packages: *acl* and *libacl* to enable Hadoop ACL support, and *libattr* to enable Hadoop extended attributes on all nodes.

Note: Using fine grained control will require extensive testing for your applications. If a user ID is not authorized to see the DFS through HDFS APIs, the error will be:

```
java.io.IOException: GPFSC00023E: Unable to establish communication with file system  
at org.apache.hadoop.fs.gpfs.GeneralParallelFileSystem.lockNativeRootAction(GeneralParallelFileSystem.java:2786)  
at org.apache.hadoop.fs.gpfs.GeneralParallelFileSystem.getFileStatus(GeneralParallelFileSystem.java:799)
```

On every node, run:

```
yum install -y acl libacl libattr
```

To see the currently set ACLs, run:

```
getfacl /var/run/ibm_bigpfs_gcd  
# file: ibm_bigpfs_gcd  
# owner: root  
# group: root  
user::rwx  
group:---  
other:---
```

- b. To allow hdfs (super user in HDFS) to have full access to DFS, run the following command on all nodes:

```
setfacl -m "u:hdfs:rwx" /var/run/ibm_bigpfs_gcd
```

- c. To allow any service ID that is a member of hadoop group (For example, Hadoop service IDs) to have full access to DFS, run the following command on all nodes:

```
setfacl -m "g:hadoop:rwx" /var/run/ibm_bigpfs_gcd
```

7. Why is the Ambari GUI displaying the Service down message when the service process is active on the target node?

Solution:

- a. Check whether the */var/lib/ambari-agent/data/structured-out-status.json* file has a length of 0 bytes. If it does, remove the *structured-out-status.json* file.
- b. Check the space usage of the file system where the JSON file resides. Free some space on the file system if the file system is full.

8. 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
```

9. If the Ambari GUI stops functioning, how must I fix it?

Solution: The Ambari GUI might stop functioning due to unresolved exception handling in Ambari version 2.1.0 app.js.



A reason for this error can be a service sending an error where the app.js could not handle the error that was sent.

Perform one of the following actions to resolve this issue:

- Restart the Ambari server from the Ambari server node.

```
# /usr/sbin/ambari-server restart
```
- Use a different browser to log into the Ambari server.
- Restart Metrics by using Ambari REST APIs from the Ambari server node.

Replace the *admin*, *\$PASSWORD*, *AMBARI_SERVER_HOST*, and *CLUSTER_NAME* with the corresponding values in the environment.

Where *admin:\$PASSWORD* is the admin user ID and password.

To Stop

```
curl -u admin:$PASSWORD -i -H 'X-Requested-By: ambari' -X PUT -d '{"RequestInfo": {"context": "Stop Ambari Metrics via REST"}, "Body": {"ServiceInfo": {"state": "INSTALLED"}}}' http://AMBARI_SERVER_HOST:8080/api/v1/clusters/CLUSTER_NAME/services/AMBARI_METRICS
```

To Start

```
curl -u admin:$PASSWORD -i -H 'X-Requested-By: ambari' -X PUT -d '{"RequestInfo": {"context": "Start Ambari Metrics via REST"}, "Body": {"ServiceInfo": {"state": "STARTED"}}}' http://AMBARI_SERVER_HOST:8080/api/v1/clusters/CLUSTER_NAME/services/AMBARI_METRICS
```

10. Oozie service check fails.

Solution: Oozie service is green but the service check fails due to BI 4.1 Oozie issue in Ambari GUI.

To run the Oozie service check, run the executable in a script on the command line.

- Create the Oozie service check script by finding where the Oozie service check executable is located.

```
# cd /  
# find . -name oozie-env.sh -print  
/usr/iop/current/oozie-client/conf/oozie-env.sh  
# find . -name oozieSmoke2.sh -print  
/var/lib/ambari-agent/data/tmp/oozieSmoke2.sh
```

- Place the script in the directory of the user (For example, root or as non-root user ID, gpfsadm).

Example: On the Ambari server, create the `Oozie.service.check.sh` script

```
# cat Oozie.service.check.sh  
source /usr/iop/current/oozie-client/conf/oozie-env.sh ;  
/var/lib/ambari-agent/data/tmp/oozieSmoke2.sh redhat /usr/iop/current/oozie-client/conf  
/usr/iop/current/oozie-client/bin /usr/iop/current/hadoop-client/conf  
/usr/iop/current/hadoop-client/bin ambari-qa False
```

- Run the script.

Example: On the Ambari server, run the script as the Ambari installation user ID (For example, root or gpfsadm)

```
# ./oozie.service.check.sh
```

11. 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.

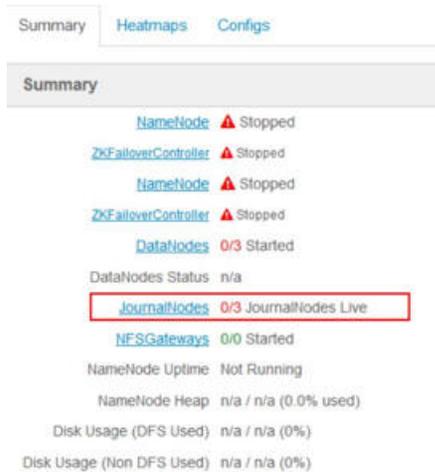
Run the following command on a HDFS Transparency host to tar up the HDFS Transparency directory into /tmp:

```
# cd /usr/lpp/mmfs/hadoop/etc
# tar -cvf /tmp/hdfs.transparency.hadoop.etc.tar hadoop
```

12. Journal nodes are not installed while unintegrating to native HDFS in a Kerberos-enabled Namenode HA environment.

Solution:

- a. On the HDFS dashboard, go to **Summary > JournalNodes**, and select one of the JournalNodes.

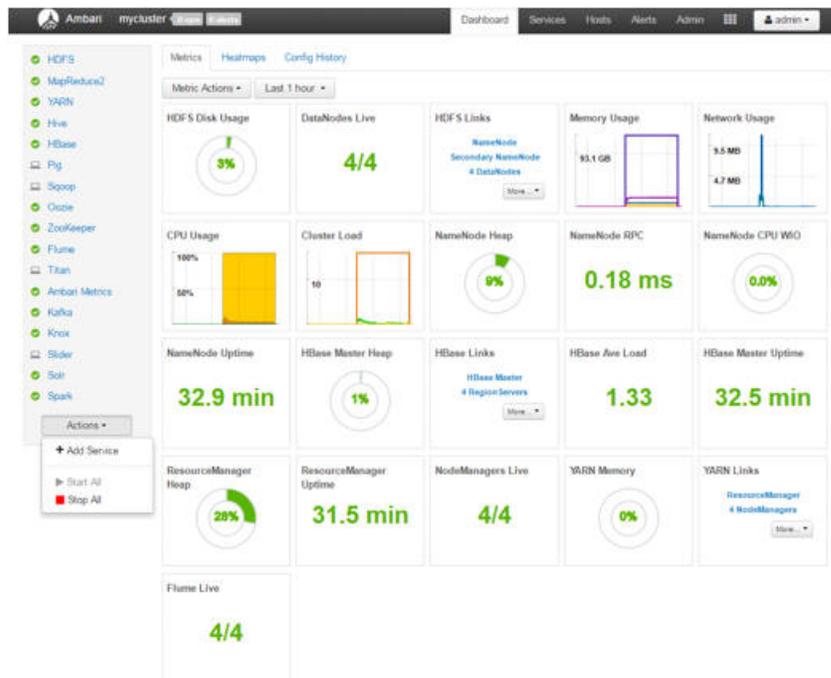


- b. On the host panel for the JournalNodes, select the JournalNodes component, click the drop-down menu and select **Reinstall**.



- c. Perform Step 1 and Step 2 for the rest of the Journal Nodes.
13. HDFS checkpoint confirmation warning message from **Actions > Stop All** 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**, 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 `c902f05x01.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'
```

Cancel

Next

14. Datanode Unmounted Data Dir alert after adding a host into the Spectrum Scale cluster.

Solution: The HDFS directory for the Datanode is not created. Therefore, an alert will be set. If the HDFS Transparency is unintegrated, then the native HDFS will create the directory. The directory is used by native HDFS.

DataNode Unmounted Data Dir Alert:

The screenshot displays the Ambari GUI with an 'Alerts for HDFS' dialog box open. The dialog lists the following alerts:

Alert Name	Status	Duration
DataNode Unmounted Data Dir Data dir(s) not found: /hadoop/hdfs/data	CRIT (1)	for 7 minutes
NameNode Host CPU Utilization 16 CPU, load 1.8%	OK	for 2 hours
HDFS Pending Deletion Blocks Pending Deletion Blocks: [0]	OK	for 2 hours
HDFS Upgrade Finalized State HDFS cluster is not in the upgrade state	OK	for 2 hours

The background shows the Ambari Summary page with a '1 alert' notification in the top right corner.

15. 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.

16. How to change HDFS and IBM HDFS Transparency configuration files?

Solution: Configuration changes must be done through Ambari so that HDFS and HDFS Transparency and Ambari database are all in synchronization.

The following fields only affect HDFS Transparency, and are not presented in the Ambari configuration panels:

```
/usr/lpp/mmfs/hadoop/etc/hadoop/log4j.properties
hdfs-log4j:log4j.logger.org.apache.hadoop.hdfs.server.namenode.top.window.RollingWindowManager
  /usr/lpp/mmfs/hadoop/etc/hadoop/hdfs-site.xml
dfs.ha.standby.checkpoints
dfs.namenode.shared.edits.dir
  /usr/lpp/mmfs/hadoop/etc/hadoop/hadoop-metrics2.properties
*.sink.timeline.slave.host.name
```

Note: These fields should not be modified by the user.

17. Kerberos authentication error during IBM Spectrum Scale unintegration

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 in GPFS Ambari integration version 4.2-1 and above, check to make sure that the KDC_PRINCIPAL and KDC_PRINCIPAL_PASSWORD values in **Spectrum Scale services > Configs > Advanced tab** have the correct values. Save the configuration changes.

18. As an Ambari non-root user trying to stop Ambari services, the HBase service might fail to stop with the following permission denied error.

```
resource_management.core.exceptions.Fail: Execution of 'rm -f
/var/run/hbase/hbase-hbase-regionserver.pid' returned 1. rm: cannot remove
'/var/run/hbase/hbase-hbase-regionserver.pid': Permission denied
```

Solution:

To resolve this issue: Remove the pid file that is stated in the error message.

19. Incorrect mount point (gpfs.mnt.dir) value detected in IBM Spectrum Scale service installation panel when deploying on an existing IBM Spectrum Scale cluster

Solution: This scenario will occur if the following sequence occurs:

- Install the management pack, then add the IBM Spectrum Scale service onto an existing IBM Spectrum Scale file system.
- Unintegrate the IBM Spectrum Scale service.
- Remove the IBM Spectrum Scale service by using the **curl-delete** command from Ambari.
- Try to add back the IBM Spectrum Scale service into Ambari. This will show the incorrect mount point in **gpfs.mnt.dir** field.

To resolve the incorrect mount point, before trying to add back the IBM Spectrum Scale service into Ambari in step d, run the following command after step c:

```
cp /var/lib/ambari-server/resources/stacks/BigInsights/4.2/services/stack_advisor.py.gpfs
/var/lib/ambari-server/resources/stacks/BigInsights/4.2/services/stack_advisor.py
```

20. Namenodes and Datanodes failed with the error Fail to replace Transparency jars with hadoop client jars when short-circuit is enabled.

c16f1n13.gpfs.net

Tasks ✔ Restart NameNode Copy Open

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 OK

Solution: To resolve this issue: Ensure that the Java OpenJDK is installed on all the nodes in the Hadoop cluster. See HDFS Transparency package.

21. BigSQL HBase fails when trying to write to a local directory /tmp/hbase-hbase/local/jars/tmp.

Solution: BigSQL will invoke the Hive or HBase interface to creates table.

This requires tmp local directory /file to be created with the correct owner, group and permission.

For example, the /tmp/hbase-hbase requires to be created and owned by hbase:hadoop with 755 permission. All subdirectories under /tmp/hbase-hbase requires to be set to the same permission.

22. ssh rejects additional ssh connections which causes the HDFS Transparency synconf 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
```

23. 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 > Service Actions > Restart All**

Click **Ambari GUI > Solr > Service Actions > Restart All**

24. On restarting the service that failed due to network port being in use, the Namenode is still up after doing a STOP ALL 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.

25. IBM Spectrum Scale service missing from Ambari when installed after HDP and BigSQL version 5.0.

Solution: In the management pack version 2.4.2.0, if the IBM Spectrum Scale service is missing from Ambari when installed after HDP and BigSQL version 5.0, the following manual steps are required to enable it:

Note: This example uses http. If your cluster uses https, replace the http with https.

Here, <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. The IBM Spectrum Scale Ambari management pack extension link is missing even though the management pack /var/lib/ambari-server/resources/mpacks/SpectrumScaleExtension-MPack-2.4.2.0/extensions/SpectrumScaleExtension directory exists.

Run the following 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 management pack extension is missing. Only the IBM-Big_SQL extension is seen.

- b. Create the IBM Spectrum Scale Ambari management pack 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 the extension link for management pack 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,

```
{
  "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 the Ambari server

Run the following command:

```
# ambari-server restart
```

e. Log into Ambari GUI, ensure that the IBM Spectrum Scale service is visible in the GUI.

26. 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. 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 synconf /usr/lpp/mmfs/hadoop/etc/hadoop
```

d. Start HDFS Transparency

```
mmhadoopctl connector start -N all
```

27. 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.

28. Why did the IBM Spectrum Scale service did 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 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.

29. Getting ERROR: Spectrum Scale is in integrated state and HDFS is running. Stop ALL services and start this operation again. message from the Ambari GUI Background Operations for IBM Spectrum Scale panel.

This can be a result of IBM Spectrum Scale being in the integrated state and running. However, HDFS is holding the active file handles on the file system.

Solution:

Stop HDFS and restart the IBM Spectrum Scale service.

30. 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 *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 by executing the `/usr/sbin/ambari-server restart` command.

31. Getting Spectrum Scale is stopped issue in Assign Slaves and Clients panel.

For IBM Spectrum Scale Mpack 2.4.2.2 and earlier, when you deploy the IBM Spectrum Scale service onto a pre-existing IBM Spectrum Scale cluster, the Assign Slaves and Clients panel shows that 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.](#)

Host	all none	all none	all none	all none

bitadv272

The **Click for details** will show a pop up panel that states 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 available.

Assign Slaves and Clients Issues

X

Assignment of slave and client components has the following issues

- Spectrum Scale is stopped. Please start Spectrum Scale daemons (mmstartup -a) before proceeding to 'Customize Services' page. It is required to populate the correct recommended configurations.

OK

bitadv273

Solution:

For a pre-existing IBM Spectrum Scale cluster, if the `mmgetstate -a` displays the nodes in the cluster as not active, ensure that you start IBM Spectrum Scale by executing `/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 a pre-existing IBM Spectrum Scale cluster, if the `mmgetstate -a` displays the nodes in the cluster as active and all the mount points are available on all the nodes, ignore this issue and continue with the deployment of the IBM Spectrum Scale service by clicking **OK**. This is a known bug and is being tracked internally.

32. Error while loading the Spectrum Scale configuration GUI.

Solution:

If accessing Ambari UI using firefox browser from Windows platform, sometimes the following error may be seen while the Spectrum Scale service is being deployed through the wizard:

```
Invalid Request: Malformed Request Body. An exception occurred parsing the request body:  
Unexpected character ('r' (code 114)): was expecting double-quote to start field name at  
[Source: java.io.StringReader@1bff44d8; line: 1, column: 131]
```

If you see this error, reload the configuration wizard to get past the problem.

33. IBM Spectrum Scale cannot start due to kernel extension errors.

When starting IBM Spectrum Scale, the following errors are seen in Ambari:

```
Status: 4352, Output:  
mmstartup: Starting GPFS ...  
<host> mmremote: startSubsys: The /lib/modules/3.10.0-327.62.4.el7.x86_64/extra/mmfslinux.ko  
kernel extension does not exist. Use mmbuildgpl command to create the needed kernel extension  
for your kernel or copy the binaries from another node with the identical environment.  
<host>: mmremote: startSubsys: Unable to verify kernel/module configuration.
```

Solution:

On each of the IBM Spectrum Scale node, run the `/usr/lpp/mmfs/bin/mmbuildgpl` command to build the GPFS portability layer.

For more information, see Building the GPFS portability layer on Linux nodes.

34. 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.

35. 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 2.7.3-2 and earlier.

Solution:

Check the Kerberos principal mapping `hadoop.security.auth_to_local` field in the `/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/
```

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
```

36. 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.

37. 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.

38. 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.

- | 39. Environment with ssh banner enabled during IBM Spectrum Scale service deployment.

| **Solution:**

| For Mpack version 2.4.2.4 and earlier, to prevent the Ambari stack advisor errors, deploy the IBM Spectrum Scale service without ssh banner enabled for all the nodes in the cluster.

| **Note:** When the banner is set in the environment, error messages like IBM Spectrum Scale is down and requires to be started or Consistency Check failed might occur for Mpack version 2.4.2.4 and earlier. For IBM Spectrum Scale down message, ensure that IBM Spectrum Scale is active and mounted. You can then ignore the message and continue. For the Consistency check message, ensure that the fields that you had set are valid and you can ignore the message and continue. From Mpack version 2.4.2.5, ssh banner suppression is handled in the IBM Spectrum Scale service deployment. Therefore, manually disabling ssh banner is not required.

- | 40. **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.

- | 41. 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 autoloading and automount on the IBM Spectrum Scale cluster (on the HDP cluster side).
- | c. If ESS is being used, enable autoloading on the ESS cluster.

| For more information, see IBM Spectrum Scale **mmchfs fsname -A yes** for automount and **mmchconfig autoloading=yes** commands.

| 42. 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*.

Service fails to start

1. Oozie fails to start after installation.

Solution:

Error Message:

```
resource_management.core.exceptions.Fail: Execution of 'cd /var/tmp/oozie &&  
/usr/iop/current/oozie-server/bin/oozie-start.sh'  
returned 255. WARN: Use of this script is deprecate-ed; use 'oozied.sh start' instead
```

Check that IOP-UTILS 1.1 package is used with the IOP 4.1 package.

Note: IOP-UTILS 1.2 is not compatible with IOP 4.1.

2. HDFS does not start after adding Spectrum Scale service or after running an Integrate_Transparency or a Unintegrate_Transparency UI actions in HA mode.

Solution:

After Integrate_Transparency or Unintegrate_Transparency in HA mode, if the HDFS service or its components (e.g 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** 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**.

3. In non-root Ambari environment, IBM Spectrum Scale fail to start due to NFS mount point permission 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.

- | 4. Atlas Metadata Server failed to start or the Web UI cannot be accessed.

| **Solution:**

- | a. From the Atlas logs (/var/log/atlas), see existing table error for atlas_titan:

- | • Remove the atlas_titan table from HBase. From the HBase master node:

| If the cluster is not kerberos enabled, run the following command:

```
| hbase zkcli  
| ls /hbase-unsecure/table  
| [hbase:meta, hbase:namespace, atlas_titan]  
| rmr /hbase-unsecure/table/atlas_titan
```

| If you get authorization issues (Authentication is not valid or NoAuthException) when deleting the znode table, follow the Zookeeper - Super User Authentication and Authorization workaround.

- | • Restart Atlas service. Go to **Ambari GUI > Atlas > Service Actions > Restart All**

- | • Run Atlas service check.

| Go to **Ambari GUI > Atlas > Service Actions > Run Service Check**.

- | b. If you see Table Exists error in /var/log/atlas/application.log for ATLAS_ENTITY_AUDIT_EVENTS table.

| Error:

| Caused by: org.apache.atlas.AtlasException:
| org.apache.hadoop.hbase.TableExistsException: ATLAS_ENTITY_AUDIT_EVENTS

| If ATLAS_ENTITY_AUDIT_EVENTS table is found in HBase, and you can destroy the table, then remove the ATLAS_ENTITY_AUDIT_EVENTS table from HBase using the same commands for removing the atlas_titan table. Otherwise, change the **atlas.audit.hbase.tablename** field to a different name from Ambari.

| To change the **atlas.audit.hbase.tablename** field:

| Go to **Ambari GUI > Atlas > Configs > find atlas.audit.hbase.tablename** in search bar:

| Change the existing value of ATLAS_ENTITY_AUDIT_EVENT to a new value.

| For example, ATLAS_ENTITY_AUDIT_EVENT_NEW

- | c. Save configuration.

- | 5. 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?.

6. HBase fails to start after migrating from BI to HDP.

Solution:

Case 1: When GPFS service is not yet integrated, perform the following steps:

- In HBase service, go to **config > advanced** and add the following to the **hbase-env** template:

```

export HBASE_MASTER_OPTS="$HBASE_MASTER_OPTS
{% if hbase_max_direct_memory_size %}
-XX:MaxDirectMemorySize={{hbase_max_direct_memory_size}}m
{% endif %}"
save the configuration and restart Hbase service

```

Note: This can occur only on the Power platform.

Case 2: When GPFS service is integrated, perform the following steps:

- Run the WALPlayer to playback the pending WALs by running the following:

```

bin/hbase org.apache.hadoop.hbase.mapreduce.WALPlayer
[options] <wal inputdir> <tables> [<tableMappings>]>

```

For example,

```

hbase org.apache.hadoop.hbase.mapreduce.WALPlayer /apps/hbase/data/WALs
"c902f10x10.gpfs.net,16020,1506057064277", "c902f10x12.gpfs.net,16020,1506057054610"

```

- Delete the WALs folders (WALs, oldWALs, MasterProcWALs).

```

hdfs dfs -rm -R /apps/hbase/data/WALs
hdfs dfs -rm -R /apps/hbase/data/oldWALs
hdfs dfs -rm -R /apps/hbase/data/MasterProcWALs

```

7. In a Ranger enabled environment, the HDFS Namenode failed to start.

If Ranger is enabled through Ambari, the HDFS Transparency Namenode might fail to start and no logging is seen.

Solution:

Restart the HDFS service through the HDFS Ambari UI.

If the Namenodes are still down, do the following:

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 two connector.py files that need to be patched to ensure that the CLASSPATH is set properly through Ambari.

- From Scale path: /var/lib/ambari-server/resources/mpacks/SpectrumScaleExtension-MPack-<version>/extensions/SpectrumScaleExtension/<version>/services/GPFS/package/scripts/connector.py
- From HDFS path: /var/lib/ambari-server/resources/common-services/HDFS/<version>/package/scripts/connector.py

To start the HDFS name node, perform the following:

- Save a 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
```

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. How to fix the Titan service check failure?

Solution: On the **hbase master**, as root:

- Check that the titan_solr table exists.
- If it exists then remove the titan_solr table.

```
# hbase zkcli
>ls /hbase-unsecure/table
>rmr /hbase-unsecure/table/titan_solr
> quit
```

- Restart the Titan Service.
- Run the Titan service check.
- If the Titan service check still fails, restart the HBase service, and then rerun the Titan service check.

Note: A scenario where the Titan service check can fail is if the IBM Spectrum Scale service is deployed in a non-Kerberized environment.

3. Hive service check fails.

Solution: To resolve this issue, do the following:

- a. Create a user called *anonymous* on all the nodes with the same uid and retry the service check.
For example,
\$ useradd -u 5000 anonymous
where, *5000* is an arbitrary uid of the user.
- b. Restart HDFS service.
- c. Re-run the Hive service check

4. 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.
- a. 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.
 - b. 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, follow the steps below.
 - c. Back up all the data that resides under <GPFS mount point>/apps/accumulo/data directory.
 - d. Remove the <GPFS mount point>/apps/accumulo/data directory by running **sudo -u hdfs hdfs dfs -rm -R /apps/accumulo/data**.
 - e. Reinitialize the Accumulo service, run **sudo -u accumulo ACCUMULO_CONF_DIR=/etc/accumulo/conf/server accumulo init --instance-name hdp-accumulo-instance --clear-instance-name**. Enter a valid password when asked for initial password for root.
 - f. Update the Accumulo root password and trace user password from the Ambari GUI. Set it to the same password as provided in the previous step.
 - g. Restart Accumulo.
 - h. After the Accumulo service gets started successfully and the service check passes, you can restore the data.

5. Atlas service check fails.

Solution:

For Non root HDP environment:

- a. Run **/usr/hdp/current/zookeeper-client/bin/zkCli.sh -server <any one zookeeper host>**.
- b. On the zookeeper CLI, run **rmr /infra-solr**.
- c. Restart **zookeeper** service.
- d. Restart **Ambari Infra** service.

For root HDP environment:

- a. Restart the Ambari Infra service.
 - b. Restart the Hbase service.
 - c. Re-run the Atlas service check.
6. What to do when the Hbase master fails to start or the Hbase service check fails?

Solution:

- a. On the HBase master, as a HBase user, run the following set of commands:

```
# hbase zkcli
>ls /hbase-unsecure
>rmr /hbase-unsecure
> quit
```

- b. Restart the HBase Service.
- c. Run the HBase service check.

Note: If the IBM Spectrum Scale service is deployed in a non-Kerberized environment, the HBase service check might fail.

7. 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".

8. 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.

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

Keyboard navigation

This product uses standard Microsoft Windows navigation keys.

IBM and accessibility

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

Notices

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

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

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

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

Terms and conditions for product documentation

Permissions for the use of these publications are granted subject to the following terms and conditions.

Applicability

These terms and conditions are in addition to any terms of use for the IBM website.

Personal use

You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative work of these publications, or any portion thereof, without the express consent of IBM.

Commercial use

You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not 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 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 may not download, export or re-export 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.

IBM Online Privacy Statement

IBM Software products, including software as a service solutions, ("Software Offerings") may use cookies or other technologies to collect product usage information, to help improve the end user experience, to tailor interactions with the end user or for other purposes. In many cases no personally identifiable information is collected by the Software Offerings. Some of our Software Offerings can help enable you to

collect personally identifiable information. If this Software Offering uses cookies to collect personally identifiable information, specific information about this offering's use of cookies is set forth below.

This Software Offering does not use cookies or other technologies to collect personally identifiable information.

If the configurations deployed for this Software Offering provide you as customer the ability to collect personally identifiable information from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent.

For more information about the use of various technologies, including cookies, for these purposes, See IBM's Privacy Policy at <http://www.ibm.com/privacy> and IBM's Online Privacy Statement at <http://www.ibm.com/privacy/details> the section entitled "Cookies, Web Beacons and Other Technologies" and the "IBM Software Products and Software-as-a-Service Privacy Statement" at <http://www.ibm.com/software/info/product-privacy>.

Glossary

This glossary provides terms and definitions for IBM Spectrum 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.

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 connection, 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 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

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.

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

A IP address used to communicate on a private network.

public IP address

A 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 involves 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 Spectrum Protect™ Hierarchical Storage Management (HSM).

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 Spectrum 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).

Index

A

- accessibility features for IBM Spectrum Scale 221
- accumulo support 94
- add
 - service 120
 - services 115
- advanced features 28
- Apache Hadoop
 - configuration 19
- Apache Ranger 81, 161
- Apache viewfs support 71

B

- BI
 - limitations 197
- BigInsights 4.2.5 103

C

- cluster and file system information
 - configuration 22
- configuration
 - automatic namenode HA 30
 - BigSQL 157
 - cluster and file system information 22
 - short-circuit read 33, 34
- configuration refresh
 - automatic 81
- configure
 - BigData and Analytics 149
 - logsearch 156
 - multiple file system 127
 - multiple mount point access 127
 - remote mount access 125
- configure MySQL 81, 161
- configure Ranger 86
- configuring 77
 - docker instance 38
 - HDFS transparency 38

D

- deploying
 - dual-network 150
 - FPO 157
- difference
 - HFDS transparency and native HDFS 101
 - IBM Spectrum Scale 195
 - native HDFS 195
- differences
 - functional 194
- differences from native HDFS 99
- disable
 - kerberos 166
 - short circuit write 172
- disk-partition 157
- distcp 68
- dual network interfaces 13

E

- enable
 - kerberos 166
 - Ranger HDFS plugin 85, 165
- enable kerberos 47
- enable Kerberos
 - GPFS Ambari integration version 4.2.1 167, 168
- enable ranger 49
- enable Ranger 161

F

- FAQ 201
 - general 201
 - Hadoop storage tiering 68
 - service check failures 217
 - service fails to start 214
- federate
 - IBM Spectrum Scale and HDFS 72
- federation 198
 - limitations 198
- file system support 96
- file systems 76
- fix hive schema 45
- FPO
 - dual network interfaces 13

G

- GPFS Ambari integration version 4.2.1
 - enable Kerberos 167, 168
- group ids 111
- GUI
 - management 192

H

- hadoop
 - distcp support 79
- Hadoop
 - configurations 19
 - configure nodes 17
 - connector 7
 - connector 2.4 25
 - connector 2.5 25
 - connector 2.7 26
 - health check 22
 - service health check 22
 - Hadoop connector
 - administer 157
 - Hadoop connector 2.4
 - removing 25
 - Hadoop connector 2.5
 - removing 25
 - Hadoop connector 2.7
 - removing 26
 - Hadoop distribution support 98
 - hadoop nodes
 - hardware and software requirements 12

- Hadoop nodes
 - configure 17
- hadoop storage tiering 43
- hadoop support 1
- Hadoop test case 52
- hardware requirements 103
- hardware resource configuration 12
- HDFS
 - configuration files 19
 - installation 14
 - storage mode 19
 - transparency cluster 26
 - update environment variables 19
 - update other configuration files 19
- HDFS protocol nodes
 - configure 19
- HDFS transparency
 - administer 157
 - application interaction 23
 - application interface 23
 - command line 23
 - configuration 14, 15
 - DataNode 7, 33, 171
 - DFS Client 7
 - docker support 38
 - FPO mode 11
 - FPO nodes 11
 - hardware and software requirements 12
 - high availability 28
 - high availability configuration 28, 30
 - installation 14
 - installation of 14
 - integration 176
 - limitations 197
 - Local storage mode 2
 - NameNode 7
 - overview 1
 - security 28
 - shared storage mode 2
 - short-circuit read configuration 33, 34, 171
 - start services 21
 - stop services 21
 - unintegration 178
 - update environment variables 21
 - upgrade 24
- HDFS Transparency
 - hadoop service roles 13
- HDFS transparency cluster
 - upgrade 26
- HDP 2.6 103
- HDP 2.6.x 99
- high availability 149
- hive on tez 57
- hive operations
 - import and export 58
- hive schema
 - local Hadoop cluster 45
- hive-mapreduce/tez 54

I

- IBM BigInsights IOP 99
 - configuration 19
- IBM ESS storage 5
- IBM Spectrum Scale 1, 2, 7, 11, 14, 15, 17, 19, 24, 25, 26, 28
 - administer 157
 - planning 103

- IBM Spectrum Scale (*continued*)
 - service installation 118
 - service management 173, 176
 - special configuration 94
 - storage mode 2
- IBM Spectrum Scale Ambari management pack 119
- IBM Spectrum Scale information units vii
- IBM Spectrum Scale service 113
- install 107
 - Apache Ranger 81, 161
 - IBM Spectrum Scale 118
 - Ranger 82, 161, 162
 - verify and test 131
- install Ambari
 - restrict root access 188
- install IBM Spectrum Scale
 - restrict root access 188
- install ranger 81
- installation
 - BI IOP or HDP 115
- installing
 - BigInsight value-add 135
- integration
 - hadoop distributions 11
- issues
 - kerberos enabled environment 170

J

- join
 - HDFS transparency with native HDFS 74
 - native HDFS with HDFS transparency 72

K

- KDC server 169
- kerberos 166
 - disable 171
- kerberos security 61
 - ranger policy 64
 - ranger policy cases 64
- kernel 109
- kinit
 - datanodes 170
 - namenodes 169
- known information
 - IBM Spectrum Scale and HDFS transparency
 - integration 198
- known issues 91
- known limitations
 - Hadoop storage tiering 69
 - HDFS transparency 79

L

- license planning 7
- limitation
 - functional 193
- limitations
 - IBM Spectrum Scale and HDFS transparency
 - integration 198
- limitations from native HDFS 99
- local native HDFS cluster 44
- local repository 113, 151, 152
- Local storage mode
 - FPO 2

log in
Ranger UI 86, 166

M

management GUI 192
MapReduce cases 52
migration
BI IOP to HDP 136
HDP 2.6.2 137
move
namenode 183
move namenode
integrated state 184
move Namenode
HA cluster 184
non-HA cluster 184
multiple hadoop clusters 37
multiple HDFS transparency clusters 40

N

native HDFS 95, 193
install ranger 81
network validation 110
NFS/SMB 101
node
add 180
Ambari GPFS 188
delete 182
node management 180
node roles 11
Nodes
OS tuning 17
NTP 110

O

OS repository 152
overview
hadoop tiering 43
software stack 114

P

parameter checklist 149
partitioning
function matrix 159
password-less
root 110
patch
GPFS Ambari 136
physical node
HDFS transparency cluster 38
policy file 107
pre-existing IBM Spectrum Scale cluster 118
problem determination 103

R

rack mapping 158
ranger
disable 91, 166
secure HDFS 89
ranger auditing 91, 166

ranger policy 63
remote HDFS transparency cluster 45
remove
IBM Spectrum Scale Hadoop connector 24
removing
Hadoop connector 2.4 24
replace
HDFS service with HDFS transparency 98
repository
MySQL community edition 154
restart order
HDFS and IBM Spectrum Scale 176
restrict root access 188
revert
native HDFS 185
rolling upgrade 26
root ssh access
password-less 15
rules
failure group 158
run
service check 174
run distcp
kerberized cluster test 68
non-kerberized cluster test 68
run hive 57
run MapReduce 52
run spark test 53

S

SELinux 110
service actions 173
service configurations
modify 174
service management 173
services
stop all 174
set
IBM Spectrum Scale configuration 157
set up
BigInsights IOP 107
HDFS Transparency package 107
Hortonworks Data Platform (HDP) 107
setup
BigInsights IOP 108
HDFS transparency package 108
Hortonworks Data Platform (HDP) 108
IBM Spectrum Scale file system 109
shared storage
rack locality 92
shared storage mode
node roles 11
short circuit write 36
simple NSD File 106
single namespace 74
single viewfs namespace 72
Snap data collection 196
snapshot support 100
software packages
mirroring repository server 151
software stack
installation 114
spark cases 52
standard NSD file 106
stanza file 105

- starting services
 - manually 150
- stop all 174

T

- TPC-DS cases 59
- TPC-DS test 59
- troubleshooting
 - Hadoop connector 196
 - HDFS transparency 196
 - value-add 136

U

- uninstall
 - ambari stack 135
 - IBM Spectrum Scale mpack and service 133
- updating
 - GPFS yum repo 136
- upgrade
 - BI IOP 4.1/4.2 to 4.2.5 145
 - BI IOP 4.2.5 145
 - HDFS transparency 142
 - HDFS Transparency NameNode 27
 - IBM Spectrum Scale file system 144
 - IBM Spectrum Scale service MPack 140
- upgrade to BI IOP 4.2.5 148, 149
- upgrading
 - BigInsights IOP 136
 - Hadoop connector 136
 - HDFS Transparency 136
- user ids 111

V

- value-add services
 - BigInsights 135
- verify
 - transparency integration 179
- verify environment 47

W

- workaround
 - kerberos enabled environment 170
- workarounds
 - IBM Spectrum Scale and HDFS transparency integration 198

Y

- yum repo 175

Z

- zero shuffle 97



Product Number: 5725-Q01
5641-GPF
5725-S28
5765-ESS

Printed in USA

SC27-4637-08

