

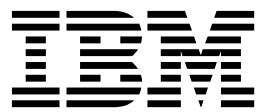
IBM Spectrum Scale

Big Data and Analytics Guide



IBM Spectrum Scale

Big Data and Analytics Guide



Note

Before using this information and the product it supports, read the information in "Notices" on page 363.

This edition applies to version 5 release 0 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 xxi. 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	vii
About this information ix	
Prerequisite and related information	xx
Conventions used in this information.	xx
How to send your comments	xxi
Summary of changes. xxiii	
Chapter 1. Hadoop Scale Storage	
Architecture	1
Elastic Storage Server (ESS)	1
Share Storage (SAN-based storage)	2
File Placement Optimizer (FPO)	2
Deployment model	3
1st model: Remote mount with all Hadoop nodes as Spectrum Scale nodes	3
2nd model: Remote mount with limited Hadoop nodes as Spectrum Scale nodes	4
3rd model: Single cluster with all Hadoop nodes as Spectrum Scale nodes	5
4th model: Single cluster with limited Hadoop nodes as Spectrum Scale nodes	7
Additional supported features about storage.	8
Chapter 2. IBM Spectrum Scale support for Hadoop 9	
HDFS transparency	9
Supported IBM Spectrum Scale storage modes.	10
Hadoop cluster planning	10
License planning	10
Node roles planning	13
Hardware and software requirements.	14
Hadoop service roles	15
Dual network interfaces	16
Installation and configuration of HDFS transparency	17
Installation of HDFS transparency	17
Configuration of HDFS transparency	17
Start and stop the service manually	24
Connector health check	24
Cluster and file system information configuration	24
Application interaction with HDFS transparency	25
Application interface of HDFS transparency	26
Command line for HDFS Transparency	26
Upgrading the HDFS Transparency cluster	27
Removing IBM Spectrum Scale Hadoop connector	27
Upgrading the HDFS Transparency cluster	29
Rolling upgrade for HDFS Transparency	30
Rolling upgrade for HDFS Transparency 2.7-x	30
Rolling upgrade for HDFS Transparency 3.x	31
Security	32
Advanced features	32
High availability configuration	32
Short-circuit read configuration.	38
Short circuit write	41
Multiple Hadoop clusters over the same file system	42
Docker support	43
Hadoop Storage Tiering	48
Hadoop distcp support	84
Automatic Configuration Refresh	86
Ranger support	86
Rack locality support for shared storage.	97
Accumulo support	99
Multiple Spectrum Scale File System support	101
Zero shuffle support	102
Native HDFS encryption	103
Hadoop distribution support	104
HortonWorks HDP 2.6.x and HDP 3.0 support	104
IBM BigInsights IOP support	104
Limitations and differences from native HDFS	104
Snapshot support	104
Hadoop ACL and Spectrum Scale Protocol	104
NFS/SMB	106
The difference between HDFS Transparency and native HDFS	106
Problem determination	107
Chapter 3. IBM Spectrum Scale Hadoop performance tuning guide 109	
Overview.	109
Introduction.	109
Hadoop over IBM Spectrum Scale	109
Spark over IBM Spectrum Scale	109
Hadoop and Software Stack Version.	109
Performance overview	110
MapReduce	110
Hive	112
Hadoop Performance Planning over IBM Spectrum Scale	112
Storage model	112
Hardware configuration planning	114
Performance guide	115
How to change the configuration for tuning	115
System tuning	116
HDFS Transparency Tuning.	117
Hadoop/Yarn Tuning	120
Performance sizing	125
Teragen	126
TeraSort	127
DFSIO.	128
TPC-H and TPC-DS for Hive	128
HBase/YCSB	131
Spark	133
Workloads/Benchmarks information	134

Chapter 4. Hortonworks Data Platform	
3.X	137
Planning	137
Hardware requirements	137
Preparing the environment	137
Installation	146
ESS setup	146
Adding Services	146
Create HDP cluster	147
Establish a IBM Spectrum Scale cluster on the Hadoop cluster	151
Configure remote mount access	152
Install Mpack package	152
Deploy the IBM Spectrum Scale service	153
Verifying installation	158
Upgrading and uninstallation	158
Upgrading IBM Spectrum Scale service MPack	158
Upgrading HDFS Transparency	161
Upgrading IBM Spectrum Scale file system	163
Upgrading from HDP 2.6.x to HDP 3.0.0	164
Uninstalling IBM Spectrum Scale Mpack and service	167
Configuration	167
Setting up High Availability [HA]	167
IBM Spectrum Scale configuration parameter checklist	168
Dual-network deployment	169
Manually starting services in Ambari	170
Setting up local repository	170
Configuring LogSearch	175
Hadoop Kafka/Zookeeper and IBM Spectrum Scale Kafka/Zookeeper	176
Create Hadoop local directories in IBM Spectrum Scale	176
Deploy HDP or IBM Spectrum Scale service on pre-existing IBM Spectrum Scale file system	177
Deploy FPO	179
Hadoop Storage Tiering	180
Limited Hadoop nodes as IBM Spectrum Scale nodes	180
Configuring multiple file system mount point access	181
Administration	184
IBM Spectrum Scale-FPO deployment	184
Ranger	187
Kerberos	193
Short-circuit read (SSR)	198
Disabling short circuit write	198
IBM Spectrum Scale service management	199
Ambari node management	205
Restricting root access	219
IBM Spectrum Scale management GUI	222
IBM Spectrum Scale versus Native HDFS	223
Limitations	225
Limitations and information	225
Problem determination	228
Snap data collection	228
General	230
Service fails to start	236
Service check failures	239
Chapter 5. Open Source Apache Hadoop	241
Apache Hadoop 3.0.x Support	241
Chapter 6. BigInsights 4.2.5 and Hortonworks Data Platform 2.6	243
Planning	243
Hardware requirements	243
Preparing the environment	243
Preparing a stanza file	245
Installation	247
Set up	248
Installation of software stack	254
BigInsights value-add services on IBM Spectrum Scale	276
Upgrading software stack	277
Upgrading IBM Spectrum Scale service MPack	277
Upgrading HDFS Transparency	280
Upgrading IBM Spectrum Scale file system	282
Migrating IOP to HDP	283
Configuration	287
Setting up High Availability [HA]	287
IBM Spectrum Scale configuration parameter checklist	287
Dual-network deployment	288
Manually starting services in Ambari	289
Setting up local repository	289
Configuring LogSearch	294
Setting IBM Spectrum Scale configuration for BigSQL	295
Hadoop Kafka/Zookeeper and IBM Spectrum Scale Kafka/Zookeeper	295
Administration	295
IBM Spectrum Scale-FPO deployment	295
Ranger	299
Kerberos	304
Short-circuit read (SSR)	309
Disabling short circuit write	310
IBM Spectrum Scale service management	311
Ambari node management	319
Restricting root access	327
IBM Spectrum Scale management GUI	331
IBM Spectrum Scale versus Native HDFS	332
Troubleshooting	335
Snap data collection	335
Limitations	336
Limitations and information	336
FAQ	340
General	340
Service fails to start	353
Service check failures	357
Accessibility features for IBM Spectrum Scale	361
Accessibility features	361
Keyboard navigation	361
IBM and accessibility	361

Notices	363
Trademarks	364
Terms and conditions for product documentation	365
IBM Online Privacy Statement.	365
Glossary	367
Index	373

Tables

1.	IBM Spectrum Scale library information units	x		16.	Tuning MapReduce2	120
2.	Conventions	xx		17.	Configurations for tuning Yarn.	122
3.	Spectrum Scale License requirement	11		18.	Hive's Tuning	129
4.	Generating internal configuration files	24		19.	HBase Configuration Tuning	131
5.	initmap.sh script command syntax	24		20.	IBM Spectrum Scale Tuning.	132
6.	Internal configuration files and location information	25		21.	YCSB Configuration Tuning.	132
7.	Ranger directory permission issues.	97		22.	Hadoop distribution support matrix	137
8.	Sharing nothing -vs- Shared storage	114		23.	IBM Spectrum Scale partitioning function matrix	186
9.	Hardware configuration for FPO model	114		24.	NATIVE HDFS AND IBM SPECTRUM SCALE DIFFERENCES	224
10.	Hardware configuration for IBM Spectrum Scale client node in shared storage model	115		25.	Hadoop distribution support matrix	244
11.	System Memory Allocation	116		26.	Preferred Simple Format and Standard Format.	245
12.	How to change the memory size	117		27.	IBM Spectrum Scale partitioning function matrix	298
13.	Tuning configurations for Transparency over IBM Spectrum Scale FPO.	118		28.	NATIVE HDFS AND IBM SPECTRUM SCALE DIFFERENCES	334
14.	Tuning configurations for Transparency over IBM ESS or shared storage	119				
15.	Configurations in hdfs-site.xml.	120				

About this information

This edition applies to IBM Spectrum Scale™ version 5.0.3 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 x.

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
<i>IBM Spectrum Scale: Concepts, Planning, and Installation Guide</i>	<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 • Introduction to file audit logging • Introduction to watch folder • 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 • Planning for Cloud services • Firewall recommendations • Considerations for GPFS applications 	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

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

Information unit	Type of information	Intended users
<i>IBM Spectrum Scale: Concepts, Planning, and Installation Guide</i>	<p>Installing</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 • Installing file audit logging • Installing watch folder • Steps to permanently uninstall GPFS and/or Protocols 	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

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

Information unit	Type of information	Intended users
<i>IBM Spectrum Scale: Concepts, Planning, and Installation Guide</i>	<p>Upgrading</p> <ul style="list-style-type: none"> • IBM Spectrum Scale supported upgrade paths • Upgrading to IBM Spectrum Scale 5.0.x from IBM Spectrum Scale 4.2.y • Upgrading to IBM Spectrum Scale 4.2.y from IBM Spectrum Scale 4.1.x • Upgrading to IBM Spectrum Scale 4.1.1.x from GPFS V4.1.0.x • Upgrading from GPFS 3.5 • Online upgrade support for protocols and performance monitoring • Upgrading AFM and AFM DR • Upgrading object packages • Upgrading SMB packages • Upgrading NFS packages • Upgrading call home • Manually upgrading the performance monitoring tool • Manually upgrading pmswift • Manually upgrading the IBM Spectrum Scale management GUI • Upgrading Cloud services • Upgrading to IBM Cloud Object Storage software level 3.7.2 and above • Upgrading file audit logging authentication • Upgrading watch folder callbacks • Upgrading IBM Spectrum Scale components with the installation toolkit • Migrating from Express Edition to Standard Edition • Completing the upgrade to a new level of IBM Spectrum Scale • Reverting to the previous level of IBM Spectrum Scale • Coexistence considerations • Compatibility considerations • Considerations for IBM Spectrum Protect™ for Space Management • GUI user role considerations • Applying maintenance to your GPFS system • Guidance for upgrading the operating system on IBM Spectrum Scale nodes 	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

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

Information unit	Type of information	Intended users
<i>IBM Spectrum Scale: Administration Guide</i>	<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 • Ensuring high availability of the GUI service • Configuring and tuning your system for Cloud services • Configuring file audit logging • 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 	System administrators or programmers of IBM Spectrum Scale systems

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

Information unit	Type of information	Intended users
<i>IBM Spectrum Scale: Administration Guide</i>	<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 • Managing file audit logging • Performing a watch with watch folder • Highly-available write cache (HAWC) • Local read-only cache • Miscellaneous advanced administration • GUI limitations 	System administrators or programmers of IBM Spectrum Scale systems

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

Information unit	Type of information	Intended users
<i>IBM Spectrum Scale: Problem Determination Guide</i>	<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 remote cluster through GUI • Monitoring file audit logging <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 • File audit logging issues • Troubleshooting watch folder • Maintenance procedures • Recovery procedures • Support for troubleshooting • References 	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>

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>	<p>This guide provides the following information:</p> <p>Command reference</p> <ul style="list-style-type: none"> • gpfs.snap command • mmaddcallback command • mmaddir command • mmaddnode command • mmadquery command • mmafmconfig command • mmafmctl command • mmafmlocal command • mmapplypolicy command • mmaudit command • mmauth command • mmbackup command • mmbackupconfig command • mmblock command • mmbuildgpl command • mmcachectl 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
<i>IBM Spectrum Scale: Command and Programming Reference</i>	<ul style="list-style-type: none"> • mmdefedquota command • mmdefquotaoff command • mmdefquotaon command • mmdefragfs command • mmdelacl command • mmdelcallback command • mmdeldisk command • mmdelfileset command • mmdelfs command • mmdelnnode 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 • mmlinkfilesset command • mmlsattr command • mmlscallback command • mmlscluster command • mmlsconfig command • mmlsdisk command • mmlsfilesset command • mmlsfs command • mmlslicense command • mmlsmgr 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>	<ul style="list-style-type: none"> • mmlsmount command • mmlsnodeclass command • mmlsnsd command • mmlspolicy command • mmlspool command • mmlsqos command • mmlsquota command • mmlssnapshot command • mmmigratefs command • mmmount command • mmmsgqueue command • mmnetverify command • mmnfsl command • mmnsdiscover command • mmobj command • mmperfmon command • mmpmon command • mmprotocoltrace command • mmptsnap command • mmputacl command • mmquotaoff command • mmquotaon command • mmremotecluster command • mmremotefs command • mmrepquota command • mmrestoreconfig command • mmrestorefs command • mmrestripefile command • mmrestripesfs command • mmrpldisk command • mmsdrrestore command • mmsetquota command • mmshutdown command • mmsmb command • mmsnapdir command • mmstartup command • mmtracectl command • mmumount command • mmunlinkfileset command • mmuserauth command • mmwatch command • mmwinservcctl 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 • Watch folder API 	<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
<i>IBM Spectrum Scale: Big Data and Analytics Guide</i>	<p>This guide provides the following information:</p> <p>Hadoop Scale Storage Architecture</p> <ul style="list-style-type: none"> • Elastic Storage Server (ESS) • Share Storage (SAN-based storage) • File Placement Optimizer (FPO) • Deployment model • Additional supported features about storage <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>IBM Spectrum Scale Hadoop performance tuning guide</p> <ul style="list-style-type: none"> • Overview • Performance overview • Hadoop Performance Planning over IBM Spectrum Scale • Performance guide 	<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: Big Data and Analytics Guide</i>	<p>Hortonworks Data Platform 3.X</p> <ul style="list-style-type: none">• Planning• Installation• Upgrading and uninstallation• Configuration• Administration• Limitations• Problem determination <p>Open Source Apache Hadoop</p> <ul style="list-style-type: none">• Apache Hadoop 3.0.x Support <p>BigInsights 4.2.5 and Hortonworks Data Platform 2.6</p> <ul style="list-style-type: none">• Planning• Installation• Upgrading software stack• Configuration• Administration• Troubleshooting• Limitations• FAQ	<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/gpfscustersfaq.html).

Conventions used in this information

Table 2 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	<p>Bold words or characters represent system elements that you must use literally, such as commands, flags, values, and selected menu options.</p> <p>Depending on the context, bold typeface sometimes represents path names, directories, or file names.</p>

Table 2. Conventions (continued)

Convention	Usage
<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: <code>mkcondition -r IBM.FileSystem -e "PercentTotUsed > 90" \ -E "PercentTotUsed < 85" -m p "FileSystem space used"</code>
{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 IBM Spectrum Scale Big Data and Analytics support section.

Summary of changes as updated, December 2018

Changes in Mpack version 2.7.0.1

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

HDFS Transparency 3.1.0-0

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

| Summary of changes as updated, October 2018

| Changes in Mpack version 2.4.2.7

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

| HDFS Transparency 2.7.3-4

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

Summary of changes as updated, August 2018

Changes in Mpack version 2.7.0.0

- Supports HDP 3.0.

Changes in HDFS Transparency version 3.0.0-0

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

New documentation sections

- Hadoop Scale Storage Architecture

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

Summary of changes as updated, July 2018

Changes in Mpack version 2.4.2.6

- HDP 2.6.5 is supported.
- Mpack installation resumes from the point of failure when the installation is re-run.
- The **Collect Snap Data** action in the IBM 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.

Summary of changes as updated, May 2018

Changes in HDFS Transparency 2.7.3-3

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

Changes in Mpack version 2.4.2.5

- HDP 2.6.5 is supported.

Summary of changes as updated, February 2018

Changes in HDFS Transparency 2.7.3-2

- Snapshot from a remote mounted file system is supported.
- IBM 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 2.4.2.4

- 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.

Summary of changes as updated, January 2018

Changes in Mpack version 2.4.2.3

- HDP 2.6.3 is supported.

Summary of changes as updated, December 2017

Changes in Mpack version 2.4.2.2

- The Mpack version 2.4.2.2 does not support migration from IOP to HDP 2.6.2. For migration, use the Mpack version 2.4.2.1.
- From IBM 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.syncBuffsPerIteration defaults to 100. Earlier it was 1.
Percentage of Pagepool for Prefetch defaults to 60 now. Earlier it was 20.
gpfs.maxStatCache defaults to 512 now. Earlier it was 100000.
- The default maximum log file size for IBM Spectrum Scale has been increased to 16 MB from 4 MB.

Summary of changes as updated, October 2017

Changes in Mpack version 2.4.2.1 and HDFS Transparency 2.7.3-1

- The GPFS Ambari integration package is now called the IBM 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.

Chapter 1. Hadoop Scale Storage Architecture

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

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

- Centralized Storage Mode:
 - IBM Elastic Storage Server (ESS)
 - Shared Storage (SAN-based storage)
- Local Storage Mode:
 - File Placement Optimizer (FPO)

Elastic Storage Server (ESS)

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

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

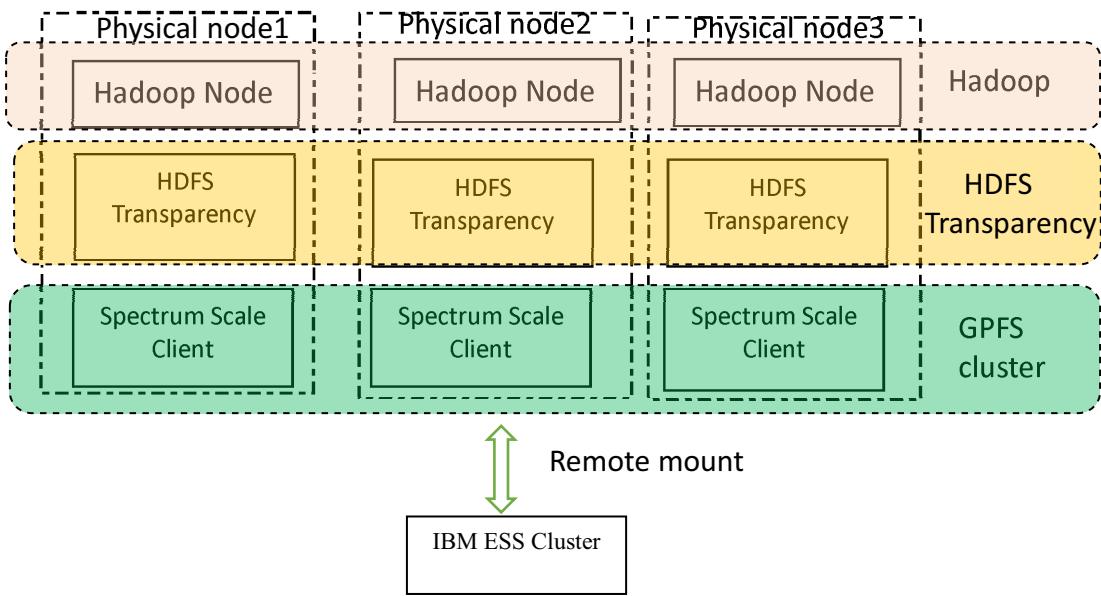


Figure 1. HDFS Transparency for IBM ESS

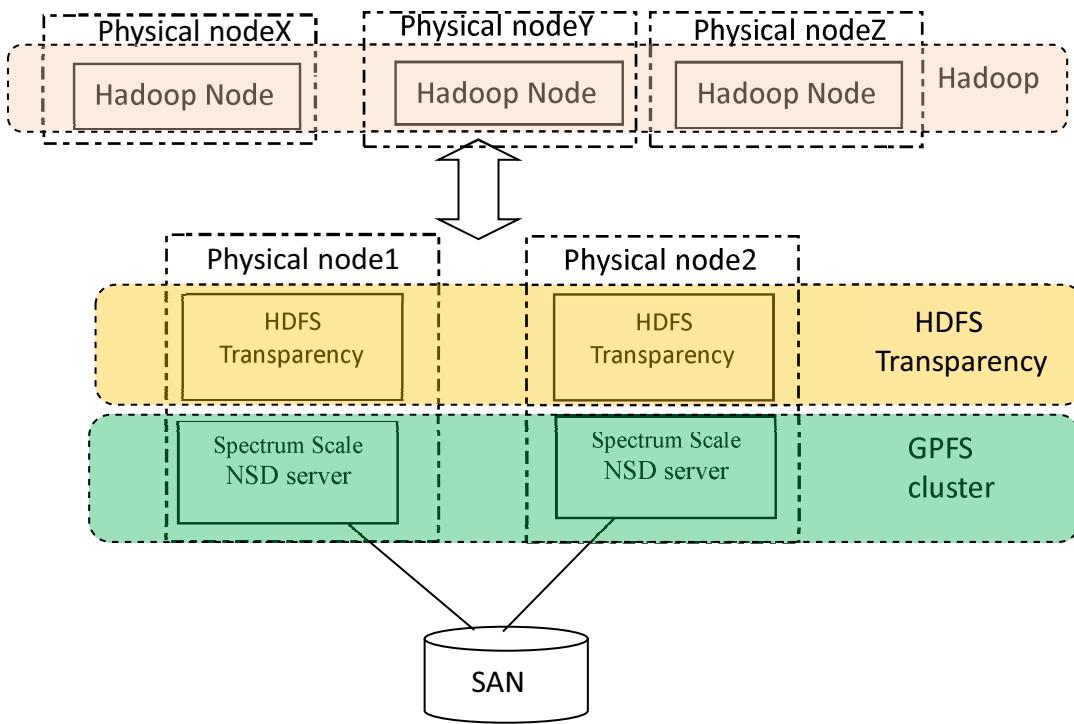
Refer IBM Elastic Storage Server Knowledge Center for more details.

b1bda09

Share Storage (SAN-based storage)

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

This is illustrated in the following figure:



b1ttda10

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

File Placement Optimizer (FPO)

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

This is illustrated in the following figure:

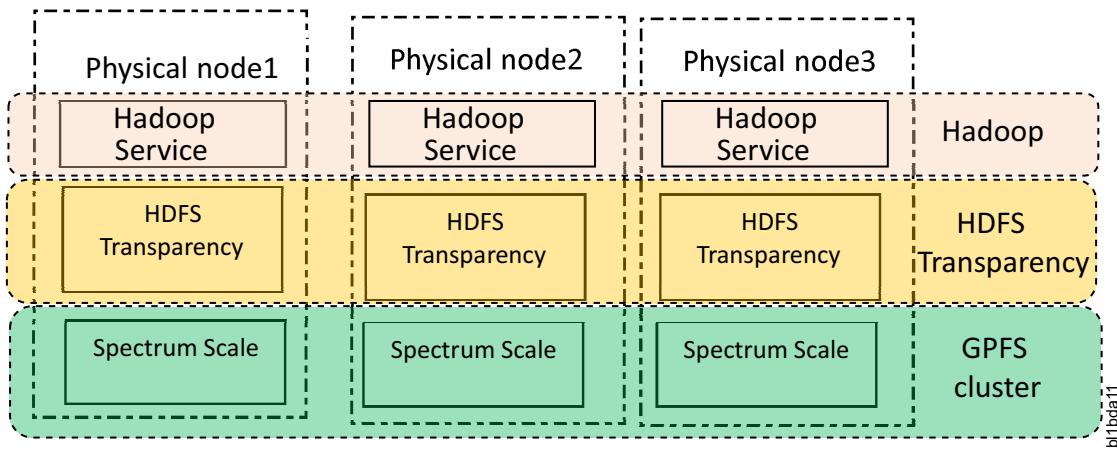


Figure 3. HDFS Transparency over IBM Spectrum Scale FPO

Refer IBM Spectrum Scale FPO for more details.

Deployment model

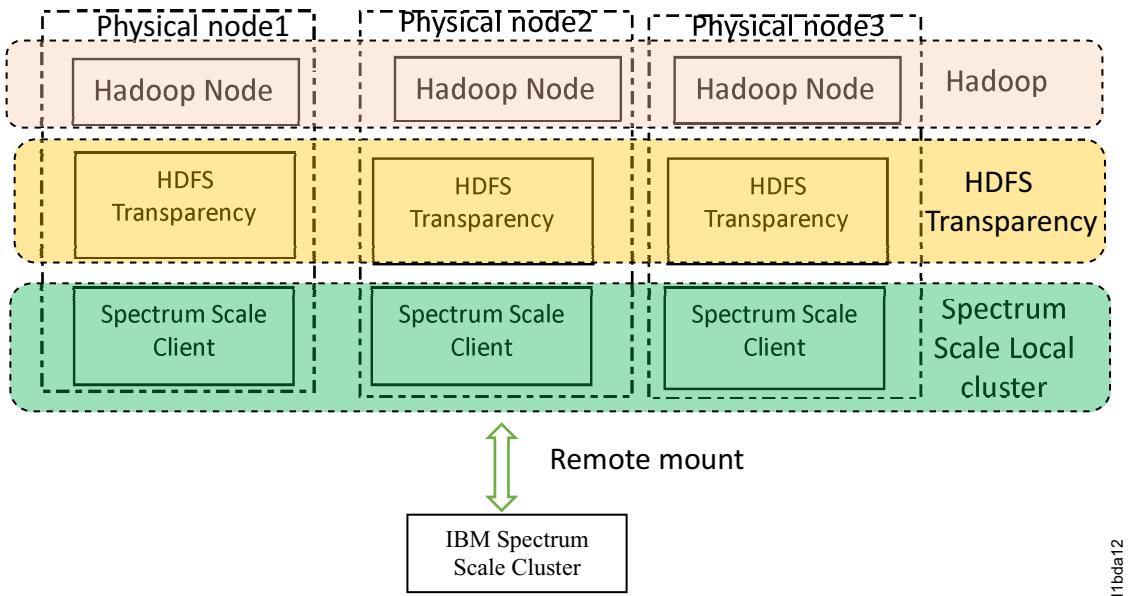
Deployment model should be considered from two levels: Spectrum Scale level and HDFS Transparency level.

From Spectrum Scale level, there are two deployment models available: remote mount mode and single cluster mode. From HDFS Transparency level, there are two deployment models available: all Hadoop nodes as Spectrum Scale nodes and limited Hadoop nodes as Spectrum Scale nodes.

1st model: Remote mount with all Hadoop nodes as Spectrum Scale nodes

This is recommended if you use IBM ESS storage and you have small Hadoop node size, typically less than 50 Hadoop nodes.

This is illustrated in the following figure:



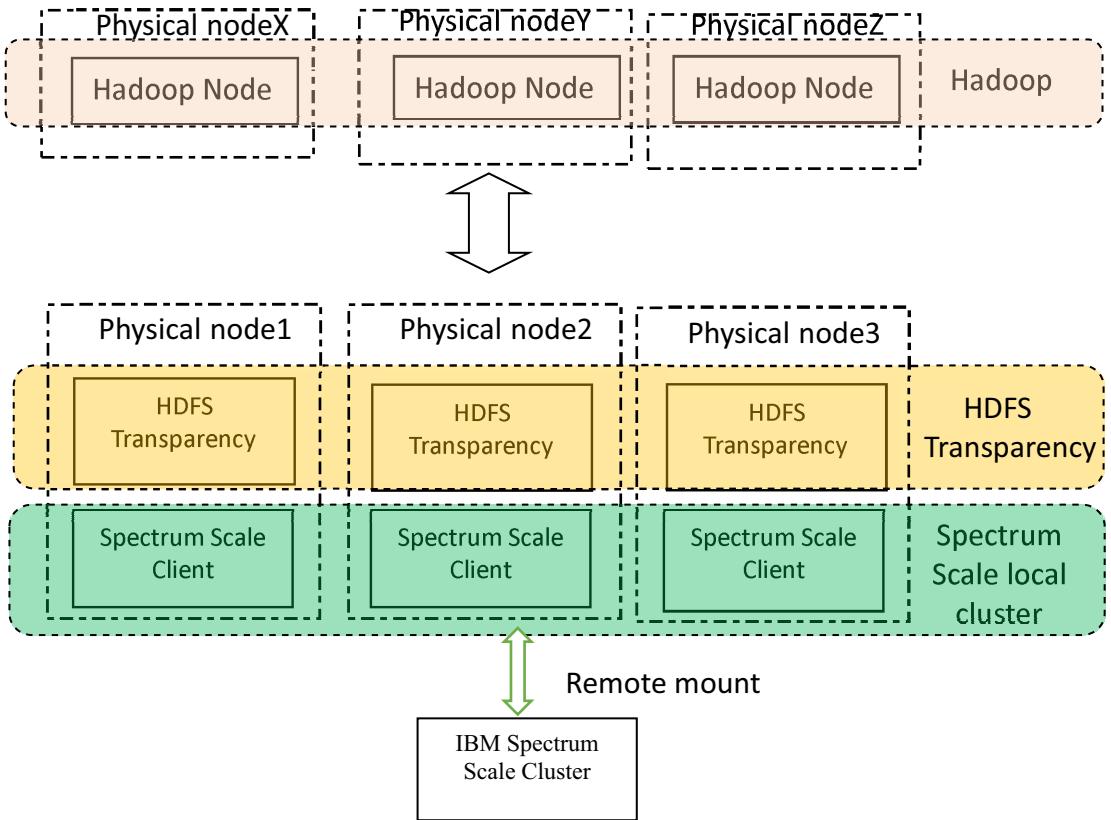
b1bda12

Figure 4. Remote mount with all Hadoop nodes as Spectrum Scale nodes

- | In this model, the IBM ESS storage is one IBM Spectrum Scale cluster then configure all the IBM Spectrum Scale clients as one cluster on the Hadoop cluster so that it can remote mounted the ESS IBM Spectrum Scale file system. All the Hadoop/Spark services will run on the IBM Spectrum Scale Hadoop local cluster.
- | With this model, one IBM ESS storage can be shared with different groups and the remote mount mode can isolate the storage management from the IBM Spectrum Scale local cluster. Some operations from local clusters (for example, `mmshutdown -a`) will not impact the storage side Spectrum Scale. Meanwhile, one can enable Hadoop Short Circuit Read/Write to gain better IO performance for Hadoop/Spark jobs.

| **2nd model: Remote mount with limited Hadoop nodes as Spectrum Scale nodes**

- | This is recommended if you use IBM ESS storage and you have huge Hadoop node size, typically more than 1000 Hadoop nodes.
- | This is illustrated by the following figure:



b11bda13

Figure 5. Remote mount with limited Hadoop nodes as Spectrum Scale nodes

This deployment model is used for large number of nodes in the Hadoop cluster (for example, more than 1000 nodes). Creating a large IBM Spectrum Scale™ cluster requires careful planning and increased demands on the network. The deployment model in Figure 5 limits the IBM Spectrum Scale deployment to just the nodes running the HDFS Transparency service rather than the entire Hadoop cluster. 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/write configuration△ does not help the data reading performance.

3rd model: Single cluster with all Hadoop nodes as Spectrum Scale nodes

- | If you use IBM Spectrum Scale FPO, this deployment model is recommended.
- | This is illustrated in the following figure:

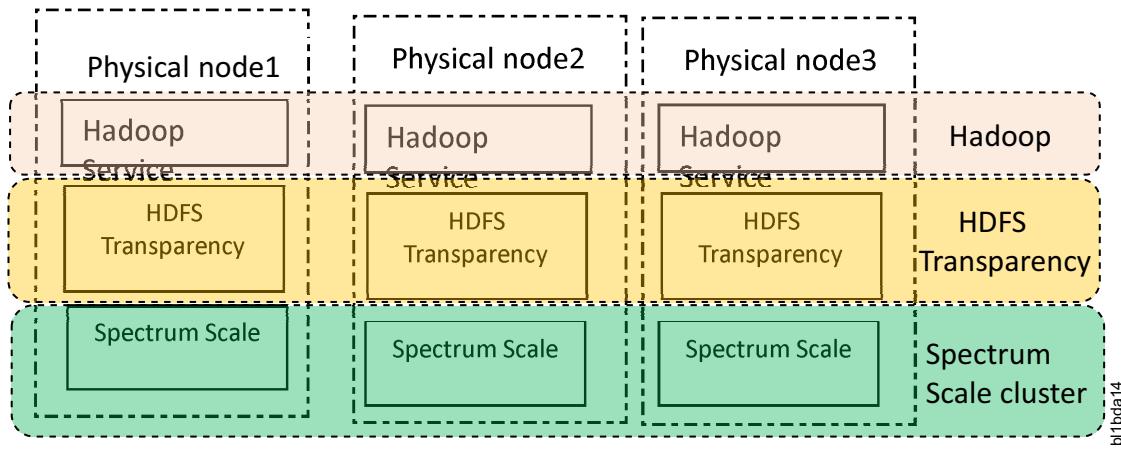


Figure 6. Single cluster with all Hadoop nodes as Spectrum Scale nodes

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

If you take IBM ESS storage, you can take this model too, illustrated by Figure 7:

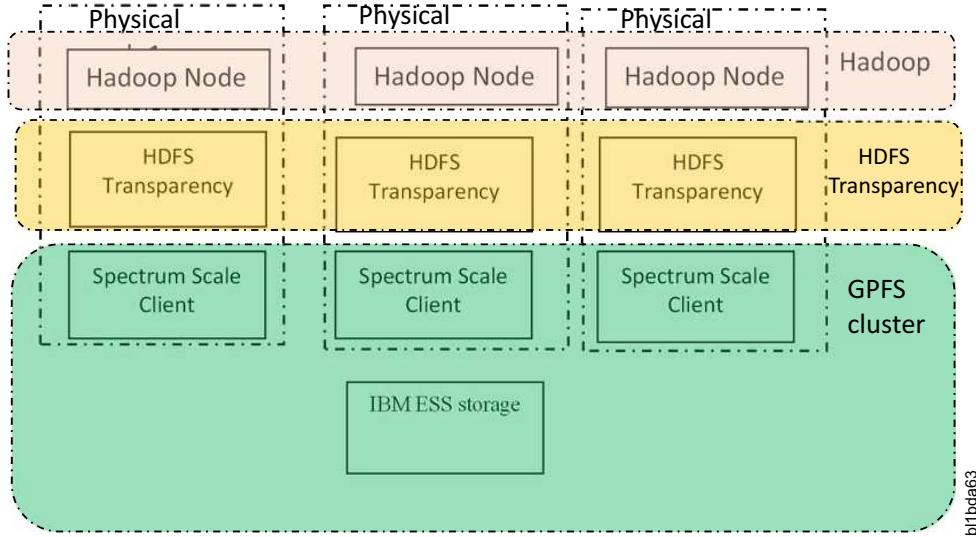
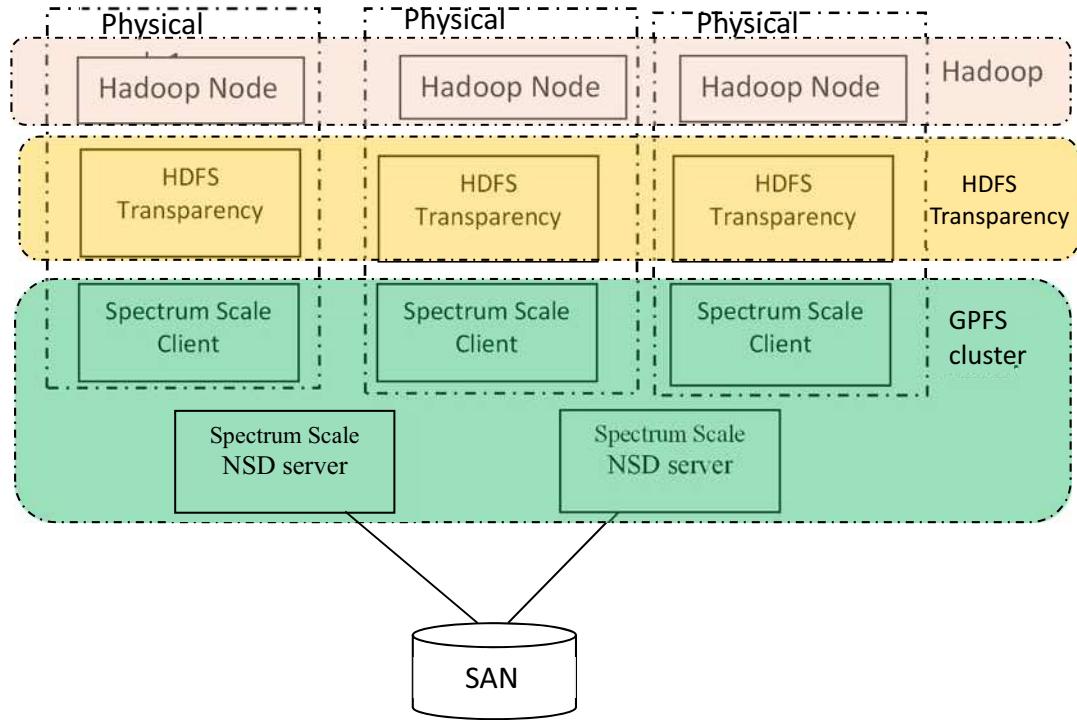


Figure 7. Single cluster with all Hadoop nodes as Spectrum Scale nodes (ESS)

If you take SAN-based storage, you can take this model too, illustrated by Figure 8 on page 7:

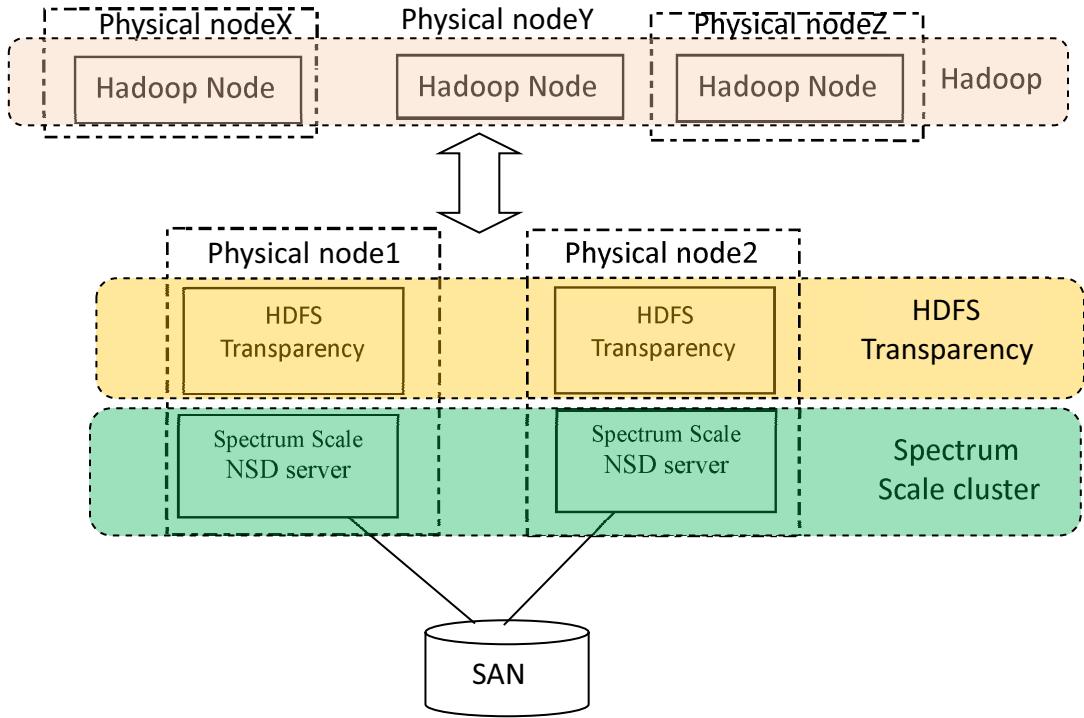


b11bda64

Figure 8. Single cluster with all Hadoop nodes as Spectrum Scale nodes (SAN storage)

4th model: Single cluster with limited Hadoop nodes as Spectrum Scale nodes

- | If you use SAN-based storage, this deployment model is recommended.
- | This is illustrated in the following figure:



b1bda15

Figure 9. Single cluster with limited Hadoop nodes as Spectrum Scale nodes

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

Additional supported features about storage

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

Hadoop Storage Tiering

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

Multiple Spectrum Scale file system support

If have multiple Spectrum Scale clusters and want to access them from the local Scale Hadoop cluster, follow the “Multiple Spectrum Scale File System support” on page 101 section. If have Ambari, follow the “Configuring multiple file system mount point access” on page 181 section.

Chapter 2. IBM Spectrum Scale support for Hadoop

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

If you plan to use a Hadoop distribution with the Hadoop connector, see the chapter that corresponds to your Hortonworks or BigInsights version or the Open Source Apache Hadoop chapter under the Big data and analytics support documentation.

Different Hadoop connectors

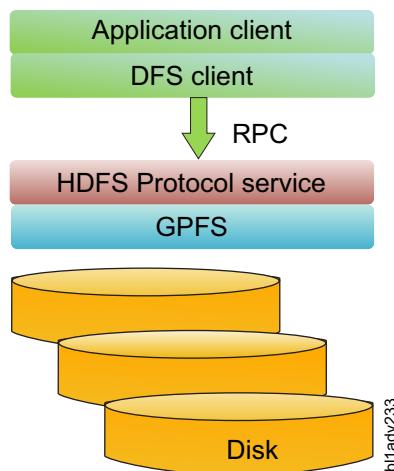
- 2nd generation HDFS Transparency
 - IBM Spectrum Scale HDFS Transparency (aka, HDFS Protocol) offers a set of interfaces that allows applications to use HDFS Client to access IBM Spectrum Scale through HDFS RPC requests. HDFS Transparency implementation integrates both the NameNode and the DataNode 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.

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

Supported IBM Spectrum Scale storage modes

For information on the supported storage modes, refer to Chapter 1, “Hadoop Scale Storage Architecture,” on page 1.

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 32. 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 2.7.x and 3.0.x 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.

In a production cluster, it is recommended to configure NameNode HA: one active NameNode and one standby NameNode. Active NameNode and standby NameNode must be located in two different nodes. For a small test or a POC cluster, such as 2-node or 3-node cluster, you can configure one node as NameNode and DataNode. However, in a production cluster, it is not recommended to configure the same node as both NameNode and DataNode.

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

License planning

HDFS Transparency does not require additional license. If you have Spectrum Scale license, you can get it from Spectrum Scale package or download the latest package from IBM developerWorks Wiki.

As for IBM Spectrum Scale license, any IBM Spectrum Scale license works with HDFS Transparency. However, you should take 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.

First go through the “Supported IBM Spectrum Scale storage modes” on page 10 section, select the mode you will take and then refer the license requirements in the following table:

Table 3. Spectrum Scale License requirement

Storage Category	Deployment Mode	License requirement
Spectrum Scale FPO	Illustrated in Figure 6 on page 6	<ul style="list-style-type: none">• 3+ 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, maximally up to 8 quorum nodes in one cluster)• All other nodes take Spectrum Scale FPO license <p>Note: If you purchase IBM Spectrum Scale capacity license, you do not need to purchase additional licenses mentioned above.</p>

Table 3. Spectrum Scale License requirement (continued)

Storage Category	Deployment Mode	License requirement
Spectrum Scale + SAN storage	Illustrated in Figure 8 on page 7	<ul style="list-style-type: none"> • All NSD servers must be with 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) <p>Note: If you purchase IBM Spectrum Scale capacity license, you do not need to purchase additional licenses mentioned above.</p>
	Illustrated in Figure 7 on page 6 (configure one Spectrum Scale cluster)	<ul style="list-style-type: none"> • 2+ Spectrum Scale server license for quorum/NSD servers with tiebreak disks for quorum. If you want to tolerate more quorum node failure, configure more Spectrum Scale NSD servers/quorum nodes <ul style="list-style-type: none"> – All HDFS Transparency nodes should take Spectrum Scale server license under this configuration. <p>Note: If you purchase IBM Spectrum Scale capacity license, you do not need to purchase additional licenses mentioned above.</p>
	Illustrated in Figure 5 on page 5 (configure Spectrum Scale Multi-cluster)	<ul style="list-style-type: none"> • For home Spectrum Scale cluster(NSD server cluster), 2+ NSD servers (Spectrum Scale server license) configured with tiebreak disks for quorum. If you want to tolerate more quorum node failure, configure more Spectrum Scale NSD servers/quorum nodes • For local Spectrum Scale cluster (all as Spectrum Scale clients), 3+ Spectrum Scale server license for quorum/manager (configure more Spectrum Scale server license node to tolerate more quorum node failure); All HDFS Transparency nodes take Spectrum Scale server license. other nodes could take Spectrum Scale client license. <p>Note: If you purchase IBM Spectrum Scale capacity license, you do not need to purchase additional licenses mentioned above.</p>
	Illustrated in Figure 4 on page 4 (configure Spectrum Scale Multi-cluster)	<ul style="list-style-type: none"> • For home Spectrum Scale cluster (NSD server cluster), 2+ NSD servers (Spectrum Scale server license) configured with tiebreak disks for quorum. If you want to tolerate more quorum node failure, configure more Spectrum Scale NSD servers/quorum nodes

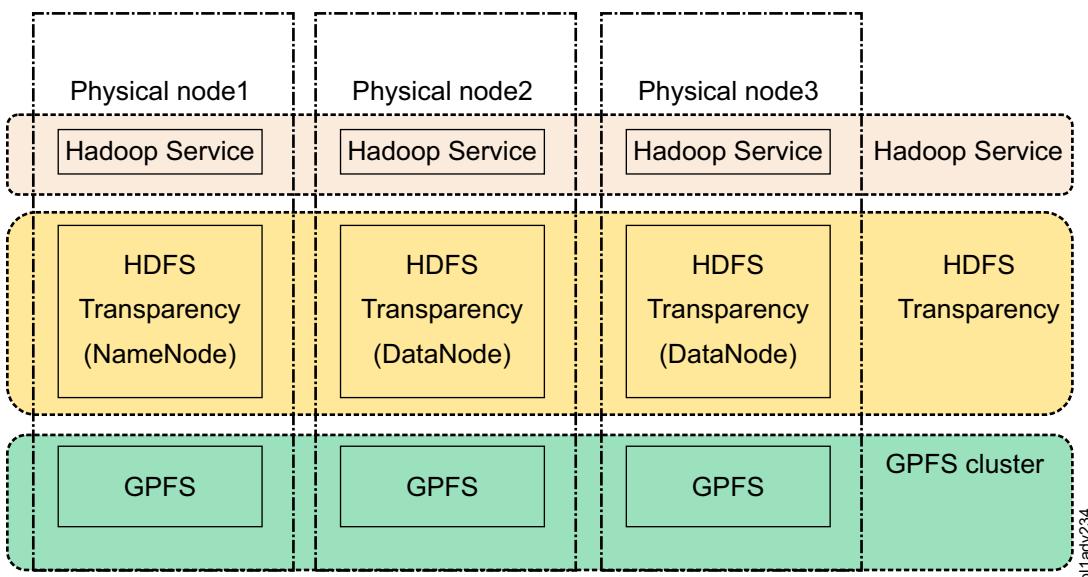
Note: If you plan to configure IBM Spectrum Scale protocol, you need to configure IBM Spectrum Scale services over nodes with IBM Spectrum Scale server license but no NSD disks in the file system. If you purchase IBM Spectrum Scale capacity license, you do not need to purchase additional licenses for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

Node roles planning

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

Node roles planning in FPO mode

In the FPO mode, all nodes are IBM 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

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

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

After HDFS transparency nodes are selected, follow the “Installation and configuration of HDFS transparency” on page 17 section to configure HDFS Transparency on these nodes.

Integration with Hadoop distributions

If you deploy HDFS transparency with a Hadoop distribution, such as IBM BigInsights® IOP or HortonWorks HDP, configure the native HDFS NameNode as the HDFS Transparency NameNode and configure native HDFS DataNodes as HDFS Transparency DataNodes. Add these nodes into IBM Spectrum Scale cluster. This setup results in fewer configuration changes. Therefore, before installing Hadoop distribution, you need to plan the nodes as NameNode and the nodes as DataNodes.

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

Hardware and software requirements

Hardware & OS Matrix support

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+
3.0.0-x	RHEL7.2+	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

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

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 (for HDFS Transparency 2.7.3-x) or /var/mmfs/hadoop/etc/hadoop/gpfs-site.xml (for HDFS Transparency 3.0.x)).

As for SAN-based storage or IBM ESS, the number of Hadoop nodes required for scaling depends on the workload types. If the workload is I/O sensitive, you could calculate the Hadoop node number according to the bandwidth of ESS head nodes and the bandwidth of Hadoop node. For example, if the network bandwidth from your ESS head nodes is 100Gb and if your Hadoop node is configured with 10Gb network, for I/O sensitive workloads, 10 Hadoop nodes (100Gb/10Gb) will drive all network bandwidth for your ESS head nodes. Considering that most Hadoop workloads are not pure I/O reading/writing workloads, you can take 10~15 Hadoop nodes in this configuration.

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 AmbariTM. 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-<version>.<arch>.rpm
```

This package has the following dependencies:

- libacl
- libattr
- openjdk

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
```

From HDFS Transparency 3.0.0, /usr/lpp/mmfs/hadoop/etc has been moved into /var/mmfs/hadoop/etc/ and therefore the contents of the directory /usr/lpp/mmfs/hadoop is the following:

```
# ls /usr/lpp/mmfs/hadoop/  
bin lib libexec license README sbin share template
```

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 Chapter 4, "Hortonworks Data Platform 3.X," on page 137.

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 HDFS Transparency 2.7.3-x) or /var/mmfs/hadoop/etc/hadoop (for HDFS Transparency 3.0.x) for any Hadoop distribution. Configuration files for Hadoop distribution are located in different locations. For example, /etc/hadoop/conf for 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.

If the file system is local, HDFS Transparency provides the following options for root 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 and to itself. 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.

Note: For the local cluster, follow one of the above options for the root password-less requirement. For the remote file system, follow one of the options listed below.

3. Option 3:

By default, HDFS Transparency NameNodes require root password-less access to at least one of the contact nodes (the 1st contact node is recommended if you cannot configure all contact nodes as password-less access) from the remote cluster.

For example, in the following cluster, `ess01-dat.gpfs.net` and `ess02-dat.gpfs.net` are contact nodes. `ess01-dat.gpfs.net` is the first contact node because it is listed first in the property **Contact nodes**:

```
# /usr/lpp/mmfs/bin/mmremotecluster show all
Cluster name: test01.gpfs.net
Contact nodes: ess01-dat.gpfs.net,ess02-dat.gpfs.net
SHA digest: abe321118158d045f5087c00f3c4b0724ed4cfb8176a05c348ae7d5d19b9150d
File systems: latestgpfs (gpfs0)
```

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

4. Option 4:

From HDFS Transparency 2.7.3-3, HDFS Transparency supports non-root password-less access to one of the contact nodes as a common user (instead of root user).

First, on HDFS Transparency NameNodes, configure password-less access for the root user as a non-privileged user to the contact nodes (at least one contact node and recommend the first contact node) from the remote cluster. Here, the `gpfsadm` user is used as an example.

Add the following into the `/usr/lpp/mmfs/hadoop/etc/hadoop/gpfs-site.xml` (for HDFS Transparency 2.7.3-x) or `/var/mmfs/hadoop/etc/hadoop/gpfs-site.xml` (for HDFS Transparency 3.0.x) file on HDFS Transparency NameNodes.

```
<property>
  <name>gpfs.ssh.user</name>
  <value>gpfsadm</value>
</property>
```

On one of the contact nodes (the first contact node is recommended), edit `/etc/sudoers` using `visudo` and add the following to the sudoers file.

```

gpfsadm  ALL=(ALL)      NOPASSWD: /usr/lpp/mmfs/bin/mm1sfs, /usr/lpp/mmfs/bin/mm1scluster,
          /usr/lpp/mmfs/bin/mm1snsd, /usr/lpp/mmfs/bin/mm1sfileset, /usr/lpp/mmfs/bin/mm1ssnapshot,
          /usr/lpp/mmfs/bin/mmcrsnapshot, /usr/lpp/mmfs/bin/mmde1snapshot, /usr/lpp/mmfs/bin/ts1sdisk

```

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/mm1sfs <fs-name>` to confirm that it works.

5. Option 5:

Manually generate the internal configuration files from the contact node and copy them onto the local nodes so that you do not require root or user password-less ssh to the contact nodes.

From HDFS transparency 2.7.3-2, you can configure `gpfs.remotecluster.autorefresh` as *false* in `/usr/lpp/mmfs/hadoop/etc/hadoop/gpfs-site.xml` (for HDFS Transparency 2.7.3-x) or `/var/mmfs/hadoop/etc/hadoop/gpfs-site.xml` (for HDFS Transparency 3.0.x).

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 24 section.

Note: If `gpfs.remotecluster.autorefresh` is configured as *false*, the snapshot from Hadoop interface is not supported against the remote mounted file system.

If the IBM Spectrum Scale cluster is configured as `adminMode=central` (check by executing `mm1sconfig 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 `mmhadoopct1` 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 NTP to synchronize the clock

For distributed cluster, configure NTP to synchronize the clock in the cluster. If the cluster can access the internet, take the public NTP servers to synchronize the clock on the nodes. However, if the cluster does not have access to the internet, then configure one of the node as the NTP server so that all the other nodes will synchronize the clock according to that NTP server.

Refer to the CONFIGURING NTP USING NTPD to configure NTP on RHEL 7.

HDFS Transparency requires the clock to be synchronized on all nodes. Otherwise potential issues will occur.

Configure Hadoop nodes

On HortonWorks HDP, you could configure Hadoop on Ambari GUI. If you are not familiar with HDFS/Hadoop, set up the native HDFS first by seeing the Hadoop cluster setup guide. Setting up the HDFS Transparency to replace the native HDFS is easier after you set up HDFS/Hadoop.

Hadoop and HDFS Transparency must take the same **core-site.xml**, **hdfs-site.xml**, **slaves** (Hadoop 2.7.x) or **workers** (Hadoop 3.0.x), **hadoop-env.sh** and **log4j.properties** for both Hadoop nodes and HDFS Transparency. This means, native HDFS in Hadoop and HDFS Transparency must take the same NameNodes and DataNodes.

Note:

1. For HortonWorks HDP, the configuration files above are located under **/etc/hadoop/conf**. For open source Apache Hadoop, the configuration files are located under **\$YOUR_APACHE_HADOOP_HOME/etc/hadoop** and **/usr/lpp/mmfs/hadoop/etc/hadoop** for HDFS Transparency 2.7.3-x.
2. For HDFS Transparency 2.7.3-3, the configurations are located under **/usr/lpp/mmfs/hadoop/etc/hadoop**. From HDFS Transparency 3.0.0, the configurations are located under **/var/mmfs/hadoop/etc/hadoop**.

If your native HDFS NameNodes are different than HDFS Transparency NameNodes, you need to update **fs.defaultFS** in your Hadoop configuration (for HortonWorks HDP it is located under **/etc/Hadoop/conf**. If it is open source Apache Hadoop, it is located under **\$YOUR_HADOOP_PREFIX/etc/hadoop/**):

```
<property>
<name>fs.defaultFS</name>
<value>hdfs://hs22n44:8020</value>
</property>
```

For HDFS Transparency 2.7.0-x, 2.7.2-0, 2.7.2-1, do not export the Hadoop environment variables on the HDFS Transparency nodes because this can lead to issues when the HDFS Transparency uses the Hadoop environment variables to map to its own environment. The following Hadoop environment variables can affect HDFS Transparency:

- **HADOOP_HOME**
- **HADOOP_HDFS_HOME**

- **HADOOP_MAPRED_HOME**
- **HADOOP_COMMON_HOME**
- **HADOOP_COMMON_LIB_NATIVE_DIR**
- **HADOOP_CONF_DIR**
- **HADOOP_SECURITY_CONF_DIR**

For HDFS Transparency versions 2.7.2-3+, 2.7.3-x and 3.0.x, the environmental variables listed above can be exported except for **HADOOP_COMMON_LIB_NATIVE_DIR**. This is because HDFS Transparency uses its own native .so library.

For HDFS Transparency versions 2.7.2-3+ and 2.7.3-x:

- If you did not export HADOOP_CONF_DIR, HDFS Transparency will read all the configuration files under /usr/lpp/mmfs/hadoop/etc/hadoop such as the gpfs-site.xml file and the hadoop-env.sh file.
- If you export HADOOP_CONF_DIR, HDFS Transparency will read all the configuration files under \$HADOOP_CONF_DIR. As gpfs-site.xml is required for HDFS Transparency, it will only read the gpfs-site.xml file from the /usr/lpp/mmfs/hadoop/etc/hadoop directory.

For questions or issues with HDFS Transparency configuration, send an email to scale@us.ibm.com.

Configure HDFS transparency nodes

This section provides information on configuring HDFS transparency nodes.

Hadoop configurations files:

This topic lists the Hadoop configuration files.

By default, HDFS Transparency 2.7.3-x uses the following configuration files located under /usr/lpp/mmfs/hadoop/etc/hadoop:

- core-site.xml
- hdfs-site.xml
- slaves
- log4j.properties
- hadoop-env.sh

HDFS Transparency 3.0.0+ uses the following configuration files located under /var/mmfs/hadoop/etc/hadoop:

- core-site.xml
- hdfs-site.xml
- workerslog4j.properties
- hadoop-env.sh
- 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 the storage mode: local or shared. Local is for IBM Spectrum Scale FPO file system and shared is for IBM Spectrum Scale over SAN storage or IBM ESS storage or remote mounted file system. This is a required configuration parameter and the `gpfs-site.xml` configuration file must be synchronized with all the HDFS Transparency nodes after the modification.

Update other configuration files:

Use this procedure to update the configuration files.

Note: To configure Hadoop HDFS, Yarn, etc. refer to the hadoop.apache.org website.

Configuring Apache Hadoop

Modify the `/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
```

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` (for HDFS Transparency 2.7.3-x) or `/var/mmfs/Hadoop/etc/hadoop/hadoop-env.sh` (for HDFS Transparency 3.0.0+) and other files as necessary.
2. Sync the changes to all the HDFS Transparency nodes (refer “Sync HDFS Transparency configurations”):

Sync HDFS Transparency configurations:

Usually, all the HDFS Transparency nodes take the same configurations. So, if you change the configurations of HDFS Transparency on one node, you need to run `/usr/lpp/mmfs/bin/mmhadoopctl` on the node to sync the changed configurations into all other HDFS Transparency nodes.

For example, `hdfs_transparency_node1` is the node where you update your HDFS Transparency configurations:

For HDFS Transparency 2.7.3-x:

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

For HDFS Transparency 3.0.0+:

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

Note: If you are using HDP with Mpack 2.4.2.1 or later, ensure you change configurations through Ambari only. If you change configurations by using `mmhadoopctl syncconf`, the changes get overwritten by Ambari integration after HDFS service restart.

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 started successfully for the first time, it executes a script called `initmap.sh` to automatically generate internal configuration files which contain the GPFS cluster information, disk-to-hostname map information and ip-to-hostname map information.

Table 4. Generating internal configuration files

HDFS Transparency version	Generating internal configuration files
HDFS Transparency 2.7.3-0 and earlier	If new disks are added in the file system or if the file systems are recreated, the <code>initmap.sh</code> script must be executed manually on the HDFS Transparency NameNode so that internal configuration files are updated with the new information.
HDFS Transparency 2.7.3-1 and later	The NameNode will run the <code>initmap.sh</code> script every time it starts so the script does not need to be run manually.
HDFS Transparency 2.7.3-2 and later	The internal configuration files will be generated automatically if they are not detected and will be synched to all other HDFS Transparency nodes when HDFS Transparency is started.

Table 5. initmap.sh script command syntax

HDFS Transparency version	initmap.sh script command syntax
HDFS Transparency 2.7.3-1 and earlier	<code>/usr/lpp/mmfs/hadoop/sbin/initmap.sh <fsName> diskinfo nodeinfo clusterinfo</code>
HDFS Transparency 2.7.3-2	<code>/usr/lpp/mmfs/hadoop/sbin/initmap.sh true <fsName> diskinfo nodeinfo clusterinfo</code>

Table 5. `initmap.sh` script command syntax (continued)

HDFS Transparency version	initmap.sh script command syntax	
HDFS Transparency 2.7.3-3+ and 3.0.0+	Local cluster mode	<code>/usr/lpp/mmfs/hadoop/sbin/initmap.sh -d -i all [fsName]</code>
	Remote Mount mode	<code>/usr/lpp/mmfs/hadoop/sbin/initmap.sh -d -r -u [gpfs.ssh.user] -i all [fsName]</code>

Note: `-u [gpfs.ssh.user]` is not necessary if `gpfs.ssh.user` is not set in `/usr/lpp/mmfs/Hadoop/etc/hadoop/gpfs-site.xml` (HDFS Transparency 2.7.3.3) or in `/var/mmfs/hadoop/etc/hadoop/gpfs-site.xml` (HDFS Transparency 3.0.0+).

Table 6. Internal configuration files and location information

HDFS Transparency version	Internal configuration files and location
HDFS Transparency 2.7.3-0 and earlier	Generated internal configuration files <code>diskid2hostname</code> , <code>nodeid2hostname</code> and <code>clusterinfo4hdfs</code> are under the <code>/var/mmfs/etc/</code> directory.
HDFS Transparency 2.7.3-1 and later	Generated internal configuration files <code>diskid2hostname.<fs-name></code> , <code>nodeid2hostname.<fs-name></code> and <code>clusterinfo4hdfs.<fs-name></code> are under the <code>/var/mmfs/etc/hadoop</code> for HDFS Transparency 2.7.3-x or under <code>/var/mmfs/hadoop/init</code> for HDFS Transparency 3.0.0+.

The following are examples of the internal configuration files:

```
# pwd
/var/mmfs/hadoop/init

# cat clusterinfo4hdfs.latestgpfs
clusterInfo:test01.gpfs.net:8833372820164647001
fsInfo:1:2:1048576:8388608
remoteInfo:gpfs0:ess01-dat.gpfs.net,ess02-dat.gpfs.net

# cat diskid2hostname.latestgpfs
1:ess01-dat.gpfs.net:30.1.1.11
2:ess01-dat.gpfs.net:30.1.1.11
3:ess01-dat.gpfs.net:30.1.1.11
4:ess01-dat.gpfs.net:30.1.1.11
5:ess02-dat.gpfs.net:30.1.1.12
6:ess02-dat.gpfs.net:30.1.1.12
7:ess02-dat.gpfs.net:30.1.1.12
8:ess02-dat.gpfs.net:30.1.1.12

# cat nodeid2hostname.latestgpfs
1:ess01-dat.gpfs.net:30.1.1.11
2:ess02-dat.gpfs.net:30.1.1.12
```

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

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

The HDFS shell command line can be used with HDFS Transparency.

Command Interface	Sub-Commands	Comments
hdfs dfs -xxx	Supported	hdfs dfs -du does not report exact value. Need to fix this in future release.
hadoop dfs -xxx		
hdfs envvars	Supported	
hdfs getconf	Supported	
hdfs groups	Supported	
hdfs jmxget	Supported	
hdfs haadmin	Supported	
hdfs zkfc	Supported	
hdfs crypto	Supported since HDFS Transparency 3.0.0+	hdfs -listZones might not list all encryption zones on HDFS Transparency. It only listed those zones you ever accessed.
hdfs httpfs	Not tested	Take HDFS WebHDFS for REST API
hdfs lsSnapshottableDir	Not supported	
hdfs oev	Not supported	
hdfs oiv	Not supported	
hdfs oiv_legacy	Not supported	
hdfs snapshotDiff	Not supported	
hdfs balancer	Not supported	
hdfs cacheadmin	Not supported	
hdfs diskbalancer	Not supported	
hdfs ec	Not supported	
hdfs journalnode	Not supported	
hdfs mover	Not supported	
hdfs namenode	Not supported	
hdfs nfs3	Not supported	
hdfs portmap	Not supported	
hdfs secondarynamenode	Not supported	
hdfs storagepolicies	Not supported	
hdfs dfsadmin	Not supported	

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 `libgpfshadoop.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 `mm1scallbacks 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-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 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 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.

Upgrading the HDFS Transparency cluster

If you could stop the HDFS Transparency cluster, execute the following steps to upgrade. If you cannot stop the HDFS Transparency cluster, refer “Rolling upgrade for HDFS Transparency” on page 30.

Procedure for upgrading HDFS Transparency 3.0.x into new 3.1.x release:

1. Back up the /var/mmfs/hadoop/etc/hadoop configuration directory.
2. Disable short circuit if it is currently enabled.
3. Stop the HDFS protocol service on all nodes by running the following command:
`/usr/lpp/mmfs/bin/mmhadoopctl connector stop`

Upgrade the RPM package on each node by running the command: `rpm -U gpfs.hdfs-protocol-<3.x version>.<arch>.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.

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

Procedure for upgrading HDFS Transparency 2.7.x into 3.x release:

1. Back up the /usr/lpp/mmfs/hadoop/etc/hadoop configuration directory.
2. Disable short circuit if it is currently enabled. If you take HortonWorks Ambari, disable short circuit read from **Ambari GUI > HDFS > Configs**. If you take open source Apache, disable short circuit read according to “Short-circuit read configuration” on page 38 section.
3. Stop the HDFS protocol service on all nodes by running the following command:
`/usr/lpp/mmfs/bin/mmhadoopctl connector stop`
4. Upgrade the RPM package on each node by running the command: `rpm -e gpfs.hdfs-protocol` and `rpm -ivh gpfs.hdfs-protocol-3.x version`. In HDFS Transparency 2.7.x, the configuration files are located under /usr/lpp/mmfs/hadoop/etc/hadoop directory. However, in HDFS Transparency 3.x release, the configuration files are located under /var/mmfs/hadoop/etc/hadoop. When installing HDFS Transparency 3.x package, it will check whether the configurations from HDFS Transparency 2.7.x are there under /usr/lpp/mmfs/hadoop/etc/hadoop. If yes, installation will automatically copy the old configurations from /usr/lpp/mmfs/hadoop/etc/hadoop into /var/mmfs/hadoop/etc/hadoop.
5. Start the HDFS Transparency service on all nodes by running the following command:
`/usr/lpp/mmfs/bin/mmhadoopctl connector start`
6. Enable short circuit again if it was previously disabled in step 2. If you take HortonWorks Ambari, enable short circuit read from **Ambari GUI > HDFS > Configs**. If you take open source Apache, enable short circuit read according to section “Short-circuit read configuration” on page 38.

Procedure for upgrading HDFS Transparency 2.7.x into new 2.7.x release:

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 protocol service on all nodes by running the following command:
`/usr/lpp/mmfs/bin/mmhadoopctl connector stop`
4. Upgrade the RPM package on each node by running the command: `rpm -U gpfs.hdfs-protocol-2.7.<x>-<y>.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 be removed during the upgrade.
5. Start the HDFS Transparency service on all nodes by running the following command:

```
/usr/lpp/mmfs/bin/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 in the same major release (For example, HDFS Transparency 2.7.x, or HDFS Transparency 3.x). However, rolling upgrade is not supported from HDFS Transparency 2.7.x to HDFS Transparency 3.0.0+ because of HDFS RPC compatibility between two major releases.

When you perform a rolling upgrade, perform the upgrade procedure for each HDFS Transparency Datanode and the upgrade procedure for the HDFS Transparency Namenodes.

Rolling upgrade for HDFS Transparency 2.7-x

Rolling upgrade for HDFS Transparency 2.7.x DataNode

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 older 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, follow the steps under “Rolling upgrade HDFS Transparency 2.7.x NameNode.”

Rolling upgrade HDFS Transparency 2.7.x 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.

Rolling upgrade for HDFS Transparency 3.x

Rolling upgrade for HDFS Transparency 3.x DataNode

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:
`/usr/lpp/mmfs/hadoop/bin/hdfs --config /var/mmfs/hadoop/etc/hadoop/ --daemon stop datanode`
2. Run **rpm -e gpfs.hdfs-protocol** to uninstall the older 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:
`/usr/lpp/mmfs/hadoop/bin/hdfs --config /var/mmfs/hadoop/etc/hadoop/ --daemon start datanode`

Rolling upgrade HDFS Transparency 3.x 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:
`/usr/lpp/mmfs/hadoop/bin/hdfs --config /var/mmfs/hadoop/etc/hadoop/ --daemon stop 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 older 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:
`/usr/lpp/mmfs/hadoop/bin/hdfs --config /var/mmfs/hadoop/etc/hadoop/ --daemon 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:
`/usr/lpp/mmfs/hadoop/bin/hdfs --config /var/mmfs/hadoop/etc/hadoop/ --daemon stop namenode`
5. Upgrade the HDFS Transparency NameNode in Step 4. Run **rpm -e gpfs.hdfs-protocol** to uninstall the older 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:
`/usr/lpp/mmfs/hadoop/bin/hdfs --config /var/mmfs/hadoop/etc/hadoop/ --daemon start namenode`

Note: When you upgrade the HDFS Transparency NameNode, if HA is not configured, HDFS Transparency service is interrupted.

Security

HDFS transparency supports full Kerberos and it is verified over Hortonworks Data Platform 2.6 and 3.0 and 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

For HortonWorks HDP, you could configure HA from Ambari GUI directly. For open source Apache Hadoop, refer the following sections.

Manual HA switch configuration

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

If your HDFS Transparency version is 2.7.0-2+, configure automatic HA according to “Automatic NameNode service HA” on page 35.

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-->
<!--Shared directory should be under gpfs.mnt.dir but not gpfs.data.dir-->
<name>dfs.namenode.shared.edits.dir</name>
<value>file:///<gpfs.mnt.dir>/HA</value>
</property>
<property>
<name>dfs.ha.standby.checkpoints</name>
<value>false</value>
</property>
<property>
<name>dfs.client.failover.proxy.provider.mycluster</name>
<value>org.apache.hadoop.hdfs.server.namenode.ha.ConfiguredFailoverProxyProvider</value>
</property>
The native HDFS uses the directory specified by dfs.namenode.shared.edits.dir configuration parameter to save information shared between the active namenode and the standby namenode. HDFS Transparency 2.7.3-4 and 3.1.0-0 uses this directory to save Kerberos delegation token related information. If you revert from HDFS Transparency back to the native HDFS, please revert dfs.namenode.shared.edits.dir configuration parameter back to the one used for the native HDFS. In Ambari Mpack 2.4.2.7 and Mpack 2.7.0.1, the dfs.namenode.shared.edits.dir parameter is set automatically when integrating or unintegrating IBM Spectrum Scale service.
The dfs.namenode.shared.edits.dir configuration parameter must be consistent with gpfs.mnt.dir defined in /usr/lpp/mmfs/hadoop/etc/hadoop/gpfs-site.xml. You can create the directory /<gpfs.mnt.dir>/HA and change the ownership to hdfs:hadoop before starting the HDFS transparency services.
For HDFS Transparency releases earlier than 2.7.3-4 and also the 3.0.0 release, the dfs.ha.standby.checkpoints must be set as false. If not, you will see a log of exceptions in the standby NameNode logs. However, for HDFS Transparency 2.7.3-4 and 3.1.0-0, the dfs.ha.standby.checkpoints must be set as true
For example,
ERROR ha.StandbyCheckpoint (StandbyCheckpoint.java:doWork(371)) - Exception in doCheckpoint
If you are using Ambari, search under HDFS > Configs to see if the dfs.ha.standby.checkpoints field is set to false. If the field is not found, add the field to the Custom hdfs-site > Add property and set it to false. Save Config, and restart HDFS and any services that you might see with restart icon in Ambari.
Note: This disables the exception written to the log.
To not show the alert from Ambari, click on the alert and disable the alert from Ambari GUI.
HDFS transparency does not have fsImage and editLogs. Therefore, do not perform checkpoints from the standby NameNode service.

```

| After 2.7.3-4 of 2.x and 3.1.0-0 of 3.x, HDFS transparency has minimal EditLogs support. Therefore,
| need checkpoints (dfs.ha.standby.checkpoints=true setting) to clean up outdated EditLogs. The 3.0.0
| release of HDFS transparency does not support EditLog (dfs.ha.standby.checkpoints=false setting).

The **dfs.client.failover.proxy.provider.mycluster** configuration parameter must be changed according to the name service ID. In the above example, the name service ID is configured as mycluster in core-site.xml. Therefore, the configuration name is **dfs.client.failover.proxy.provider.mycluster**.

Note: If you enable Short Circuit Read in the Short Circuit Read Configuration section, the value of the configuration parameter must be

org.apache.hadoop.gpfs.server.namenode.ha.ConfiguredFailoverProxyProvider.

3. Follow the guide in the Sync Hadoop configurations section to synchronize core-site.xml and hdfs-site.xml from the Hadoop distribution to any one node that is running HDFS transparency services. For example, HDFS_Transparency_node1.
4. For HDFS Transparency 2.7.0-x, on **HDFS_Transparency_node1**, modify /usr/lpp/mmfs/hadoop/etc/hadoop/hdfs-site.xml :

```
<property>
<name>dfs.client.failover.proxy.provider.mycluster</name>
<value>org.apache.hadoop.gpfs.server.namenode.ha.ConfiguredFailoverProxyProvider</value>
</property>
```

With this configuration, WebHDFS service functions correctly when NameNode HA is enabled.

Note: On HDFS transparency nodes, the configuration value of the key **dfs.client.failover.proxy.provider.mycluster** in hdfs-site.xml is different from that in Step2.

Note: This step should not be performed from HDFS Transparency 2.7.2-x.

5. On **HDFS_Transparency_node1**, run the command as the root user to synchronize the HDFS Transparency configuration to all the HDFS transparency nodes:
mmhadoopctl connector syncconf /usr/lpp/mmfs/hadoop/etc/hadoop
6. If you are using HDFS Transparency version 2.7.3-4 or 3.1.0-0, before you start HDFS transparency for the first time, run the /usr/lpp/mmfs/hadoop/bin/hdfs namenode -initializeSharedEdits command to initialize the shared edits directory. Run this command only after configuring HA and before starting the service.
7. Start the HDFS transparency service by running the **mmhadoopctl** command:
mmhadoopctl connector start
8. After the service starts, both NameNodes are in the standby mode by default. You can activate one NameNode by using the following command so that it responds to the client:
/usr/lpp/mmfs/hadoop/bin/gpfs haadmin -transitionToActive --forceactive [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 nn1If 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 Hortonworks Data Platform, the zookeeper service is deployed by default.

If you select open-source Apache Hadoop, you must set up the Zookeeper service by following the instruction on the zookeeper website.

After you set up the Zookeeper service, perform the following steps to configure automatic NameNode HA.

Note: In the following configuration example, HDFS Transparency NameNode service ID is *mycluster* and NameNode IDs are *nn1* and *nn2*. Zookeeper server *zk1.gpfs.net*, *zk2.gpfs.net* and *zk3.gpfs.net* are configured to support automatic NameNode HA. The ZooKeeper servers must be started before starting the HDFS Transparency cluster.

1. Define the NameNode service ID in the core-site.xml that is used by your Hadoop distribution.

Note: If you are using IBM BigInsights IOP or Hortonworks HDP, you can change this configuration in Ambari GUI and restart the HDFS services to synchronize it with all the Hadoop nodes.

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

2. Configure the *hdfs-site.xml* file used by your Hadoop distribution:

Note: If you are using IBM BigInsights IOP or Hortonworks HDP, you can change this configuration in Ambari GUI and restart the HDFS services to synchronize it with all the Hadoop nodes.

```
<property>
<!--define dfs.nameservices ID-->
<name>dfs.nameservices</name>
<value>mycluster</value>
</property>

<property>
<!--define 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-->
| <!--Shared directory should be under gpfs.mnt.dir but not gpfs.data.dir-->
| <name>dfs.namenode.shared.edits.dir</name>
| <value>file:///<gpfs.mnt.dir>/HA</value>
| </property>

<property>
<name>dfs.ha.standby.checkpoints</name>
<value>false</value>
</property>

<property>
<name>dfs.client.failover.proxy.provider.mycluster</name>
<value>org.apache.hadoop.hdfs.server.namenode.ha.ConfiguredFailoverProxyProvider</value>
</property>

<property>
<name>dfs.ha.fencing.methods</name>
<value>shell(/bin/true)</value>
</property>

<property>
<name>dfs.ha.automatic-failover.enabled</name>
<value>true</value>
</property>

<property>
<name>ha.zookeeper.quorum</name>
<value>zk1.gpfs.net:2181,zk2.gpfs.net:2181,zk3.gpfs.net:2181</value>
</property>

| The native HDFS uses the directory specified by dfs.namenode.shared.edits.dir configuration
| parameter to save information shared between the active namenode and the standby namenode.
| HDFS Transparency 2.7.3-4 and 3.1.0-0 uses this directory to save Kerberos delegation token related
| information. If you revert from HDFS Transparency back to the native HDFS, please revert
| dfs.namenode.shared.edits.dir configuration parameter back to the one used for the native HDFS.
| In Ambari Mpack 2.4.2.7 and Mpack 2.7.0.1, the dfs.namenode.shared.edits.dir parameter is set
| automatically when integrating or unintegrating IBM Spectrum Scale service.

```

The configuration **dfs.namenode.shared.edits.dir** must be consistent with `gpfs.mnt.dir` defined in `/usr/lpp/mmfs/hadoop/etc/hadoop/gpfs-site.xml` (for HDFS Transparency 2.7.x) or `/var/mmfs/hadoop/etc/hadoop/gpfs-site.xml` (for HDFS Transparency 3.x). You could create the `<gpfs.mnt.dir>/HA` directory and change the ownership to `hdfs:hadoop` before starting the HDFS transparency services.

For HDFS Transparency releases earlier than 2.7.3-4 of HDP 2.x and also the 3.0.0 of HDP 3.x release, the **dfs.ha.standby.checkpoints** must be set as *false*. If not, you will see a log of exceptions in the standby NameNode logs. However, for the 2.7.3-4 of HDP 2.x and 3.1.0-0 of HDP 3.x, the **dfs.ha.standby.checkpoints** must be set as *true*.

For example,

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

HDFS transparency does not have `fsImage` and `editLogs`. Therefore, do not perform checkpoints from the standby NameNode service.

After 2.7.3-4 for HDFS Transparency 2.7.x release or 3.1.0-0 for HDFS Transparency 3.x release, HDFS transparency has minimal `EditLogs` support. Therefore, need checkpoints (`dfs.ha.standby.checkpoints=true` setting) to clean up outdated `EditLogs`. The HDFS transparency release earlier than 2.7.3-4 or 3.0.0-0 do not support `EditLog` (`dfs.ha.standby.checkpoints=false` setting).

The configuration name `dfs.client.failover.proxy.provider.mycluster` must be changed according to the nameservice ID. In the above example, the nameservice ID is configured as `mycluster` in `core-site.xml`. Therefore, the configuration name is `dfs.client.failover.proxy.provider.mycluster`.

Note: If you enable Short Circuit Read in the “Short-circuit read configuration” on page 38, the value of this configuration must be `org.apache.hadoop.gpfs.server.namenode.ha.ConfiguredFailoverProxyProvider`.

3. To synchronize `core-site.xml` with `hdfs-site.xml` from your Hadoop distribution to any one node that is running HDFS transparency services, you could run `scp` to copy them into one of your HDFS Transparency node, assuming it is `HDFS_Transparency_node1`.
4. For HDFS Transparency 2.7.0-x, on `HDFS_Transparency_node1`, modify the `/usr/lpp/mmfs/hadoop/etc/hadoop/hdfs-site.xml`:

```
<property>
<name>dfs.client.failover.proxy.provider.mycluster</name>
<value>org.apache.hadoop.gpfs.server.namenode.ha.ConfiguredFailoverProxyProvider</value>
</property>
```

The WebHDFS service functions properly when NameNode HA is enabled.

Note: On HDFS transparency nodes, the above configuration value in `hdfs-site.xml` is different as that in Step2.

Note: This step should not be performed from HDFS Transparency 2.7.2-0 and later.

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

For HDFS Transparency 2.7.3-x, run the following command:

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

For HDFS Transparency 3.0.x or 3.1.x, run the following command:

```
mmhadoopctl connector syncconf /var/mmfs/hadoop/etc/hadoop
```

6. Before you start HDFS transparency for the first time, you should run the `/usr/lpp/mmfs/hadoop/bin/hdfs namenode -initializeSharedEdits` command.
- This initializes the shared edits directory.

Note: Run this command just for once after configuring HA and before staring the service.

7. Start the HDFS Transparency service by running the `mmhadoopctl` command:

- ```
mmhadoopctl connector start
```
8. Format the zookeeper data structure:
 

For HDFS Transparency 2.7.3-x:

```
/usr/lpp/mmfs/hadoop/bin/gpfs --config /usr/lpp/mmfs/hadoop/etc/hadoop/ zkfc -formatZK
```

For HDFS Transparency 3.0.x or 3.1.x:

```
/usr/lpp/mmfs/hadoop/bin/hdfs --config /var/mmfs/hadoop/etc/hadoop/ zkfc -formatZK
```

This step is only needed when you start HDFS Transparency service for the first time. After that, this step is not needed when restarting HDFS Transparency service.
  9. Start the zkfc daemon:
 

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

Run jps on the nn1 and nn2 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

10. Check the state of the services

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

```
mmhadoopctl connector getstate
```

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

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

You could check your /usr/lpp/mmfs/hadoop/etc/hadoop/hdfs-site.xml (for HDFS Transparency 2.7.3-x) or /var/mmfs/hadoop/etc/hadoop/hdfs-site.xml (for HDFS Transparency 3.0.x) or run the following commands to figure out the [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. For HortonWorks and IBM Spectrum Scale Mpack (Ambari integration), you could enable or disable short circuit read from Ambari GUI. For Apache Hadoop, refer the following sections.

Short-circuit reads provide a substantial performance boost to many applications.

### For HDFS Transparency version 2.7.0-x

Short-circuit local read can only be enabled on Hadoop 2.7.0. HDFS Transparency versions 2.7.0-x does not support this feature in Hadoop 2.7.1/2.7.2. IBM BigInsights IOP 4.1 uses Hadoop version 2.7.1.

Therefore, short circuit cannot be enabled over IBM BigInsights IOP 4.1 if HDFS Transparency 2.7.0-x is used. For more information on how to enable short-circuit read on other Hadoop versions, contact scale@us.ibm.com.

#### Configuring short-circuit local read:

To configure short-circuit local reads, enable libhadoop.so and use the DFS Client shipped by the IBM 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 for HDFS Transparency 2.7.3-x:

1. Back up the `hadoop-hdfs-client-<version>.jar` by running `$ mv $YOUR_HADOOP_PREFIX/share/hadoop/hdfs/hadoop-hdfs-client-<version>.jar $YOUR_HADOOP_PREFIX/share/hadoop/hdfs/hadoop-hdfs-client-<version>.jar.backup`
2. Link `hadoop-hdfs-client-<version>.jar` to classpath using `$ ln -s /usr/lpp/mmfs/hadoop/share/hadoop/hdfs/hadoop-hdfs-client-<version>.jar $YOUR_HADOOP_PREFIX/share/hadoop/hdfs/hadoop-gpfs-client-<version>.jar`

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 86 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 in Hadoop 2.7.x or hadoop-hdfs-client-<version>.jar into their packages in Hadoop 3.0.x. When enabling short circuit write, we need to re-package them with the hadoop-hdfs-2.7.3.jar from HDFS Transparency 2.7.3-x or hadoop-hdfs-client-<version>.jar from HDFS Transparency 3.0.x. When disabling short circuit write, we need to re-package them with the hadoop-hdfs-2.7.3.jar from Apache Hadoop 2.7.x or hadoop-hdfs-client-<version>.jar from Apache Hadoop 3.0.x.

## 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.

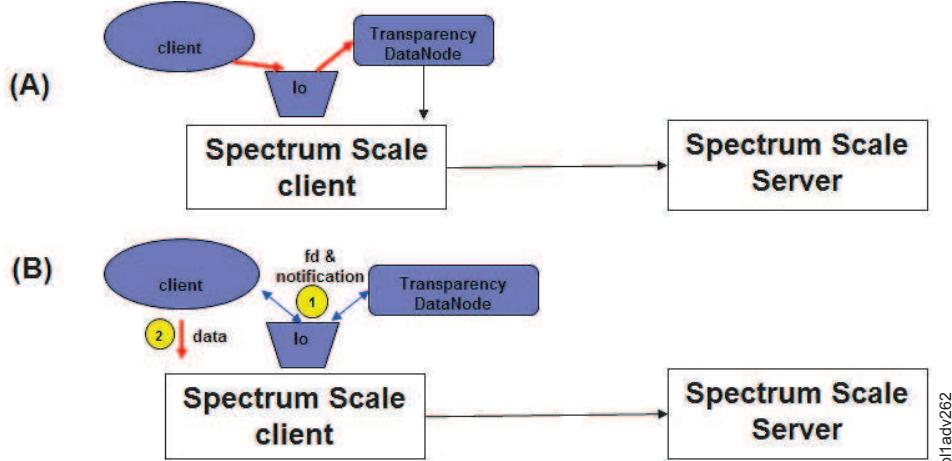


Figure 10. Short Circuit Write Logic

In Figure 10, (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 38 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` (for HDFS Transparency 2.7.3-x) or `/var/mmfs/hadoop/etc/hadoop/gpfs-site.xml` (for HDFS Transparency 3.0.x) and run `/usr/lpp/mmfs/bin/mmhadoopctl connector syncconf /usr/lpp/mmfs/hadoop/etc/hadoop` (for HDFS Transparency 2.7.3-x) or `/usr/lpp/mmfs/bin/mmhadoopctl connector syncconf /var/mmfs/hadoop/etc/hadoop` (for HDFS Transparency 3.0.x) to sync the change to all HDFS Transparency nodes.

## Multiple Hadoop clusters over the same file system

By using HDFS transparency, you can configure multiple Hadoop clusters over the same IBM Spectrum Scale file system. For each Hadoop cluster, you need one HDFS transparency cluster to provide the filesystem service.

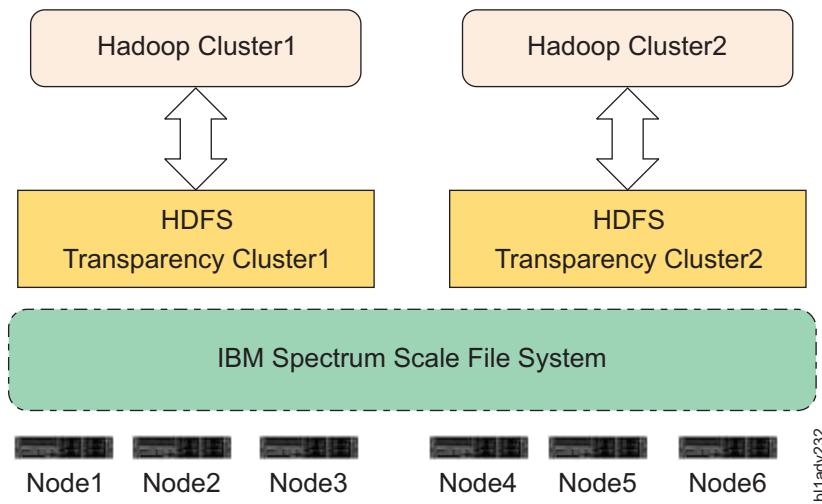


Figure 11. Two Hadoop Clusters over the same IBM Spectrum Scale file system

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` (for HDFS Transparency 2.7.3-x) or `/var/mmfs/hadoop/etc/hadoop/gpfs-site.xml` (for HDFS Transparency 3.0.x):

1. Change the `gpfs-site.xml` for HDFS transparency cluster1 to store the data under `/<gpfs-mount-point>/<hadoop1>` (`gpfs.data.dir=hadoop1` in `gpfs-site.xml`).
2. Run `mmhadoopctl connector syncconf /usr/lpp/mmfs/hadoop/etc/hadoop` (for HDFS Transparency 2.7.3-x) or `mmhadoopctl connector syncconf /var/mmfs/hadoop/etc/hadoop` (for HDFS Transparency 3.0.x) to synchronize the `gpfs-site.xml` from Step1 to all other nodes in HDFS transparency cluster1.
3. Change the `gpfs-site.xml` for HDFS transparency cluster2 to store the data under `/<gpfs-mount-point>/<hadoop2>` (`gpfs.data.dir=hadoop2` in `gpfs-site.xml`).
4. Run `mmhadoopctl connector syncconf /usr/lpp/mmfs/hadoop/etc/hadoop` (for HDFS Transparency 2.7.3-x) or `mmhadoopctl connector syncconf /var/mmfs/hadoop/etc/hadoop` (for HDFS Transparency 3.0.0) to synchronize the `gpfs-site.xml` from Step3 to all other nodes in HDFS transparency cluster2.
5. Restart the HDFS transparency services.

## 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

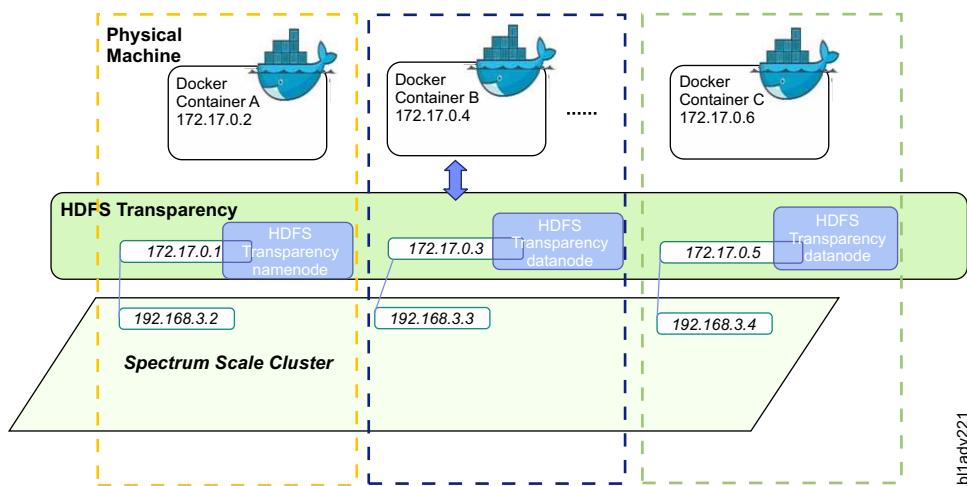
HDFS Transparency must be a Spectrum Scale node (Either a Spectrum Scale client or a Spectrum Scale server).

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-enp1s0f0
Generated by dracut initrd
DEVICE="enp1s0f0"
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 amenode 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 2, “IBM Spectrum Scale support for Hadoop,” on page 9 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 84.

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 76.

**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 12 and Figure 13:

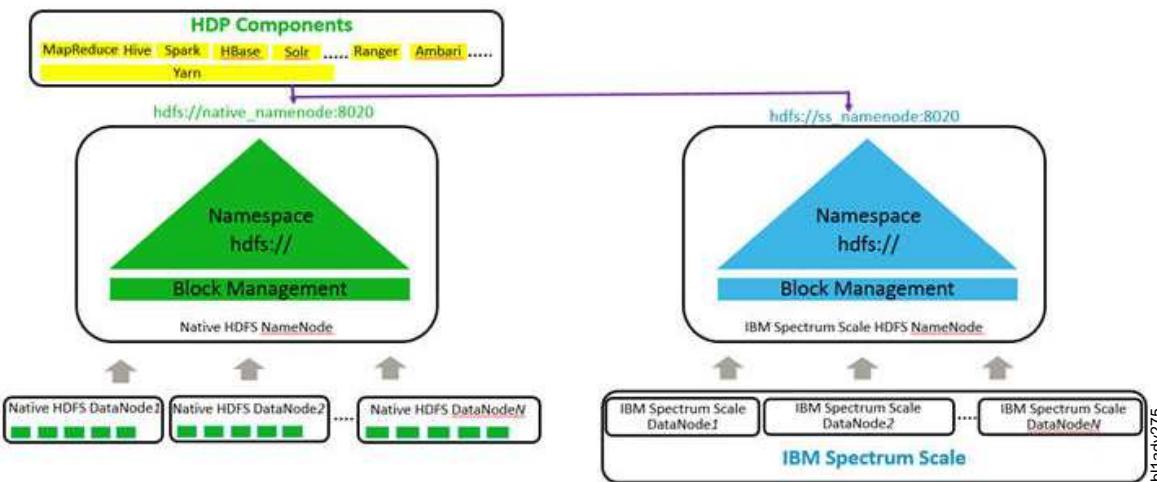


Figure 12. Hadoop Storage Tiering with IBM Spectrum Scale with single HDPE cluster

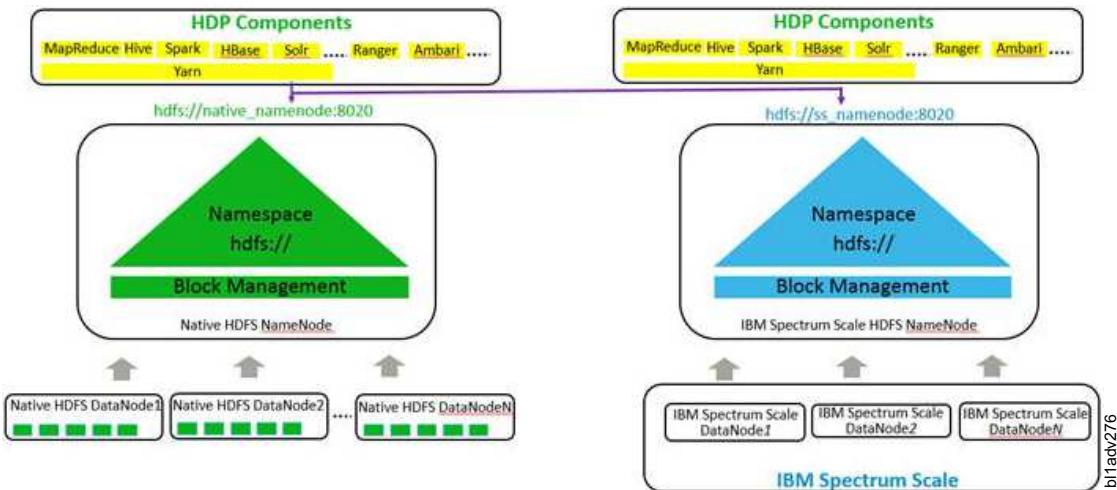


Figure 13. Hadoop Storage Tiering with IBM Spectrum Scale with 2 HDPE clusters

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:

<b>Clusters</b>	<b>Stack</b>	<b>Version</b>
HDP cluster	Ambari	2.6.1.0
	HDP	2.6.4.0
	HDP-Utils	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 17 to set up the IBM Spectrum Scale HDFS Transparency cluster.
2. Refer to “Enable Kerberos” on page 52 section to setup Kerberos and “Enable Ranger” on page 55 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 146 topic to setup HDP and IBM Spectrum Scale HDFS Transparency cluster.
2. Refer to “Enable Kerberos” on page 52 section to setup Kerberos and “Enable Ranger” on page 55 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:

- 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.

- Save the specific tag version cluster environment into the `cluster_env.curl` file by running the following command:

```
curl -u <ambari-user>:<ambari-password> -H "X-Requested-By: ambari" -X
GET http://<ambari-server>:<ambari-port>/api/v1/clusters/<cluster-name>/configurations?type=cluster-env&
tag=<tag_version_found> > cluster_env.curl
```

For example, running the command on the Ambari server host:

```
[root@c16f1n07 ~]# curl -u admin:admin -H "X-Requested-By: ambari" -X
GET "http://localhost:8080/api/v1/clusters/hdfs264/configurations?type=cluster-env&tag=version1"
```

- Copy the `cluster_env.curl` file into `cluster_env.curl_new` and modify the `cluster_env.curl_new` with the following information:

- Set the **manage\_hive\_fsroot** field to *false*.
- If Kerberos is enabled, set the **security\_enabled** field to *true*.
- 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.

- Remove the last symbol ] and } at the end of the `cluster_env.curl_new` file.

- Run the following command after replacing the “/path/to” with the real path to the `cluster_env.curl_new` file to POST the update:

```
curl -u <ambari-user>:<ambari-password> -H "X-Requested-By: ambari" -X
POST "http://<ambari-server>:<ambari-port>/api/v1/clusters/<cluster-name>/configurations"
--data @/path/to/cluster_env.curl_new
```

For example, on the Ambari server host, run:

```
[root@c16f1n07 ~]# curl -u admin:admin -H "X-Requested-By: ambari" -X
POST "http://localhost:8080/api/v1/clusters/hdfs264/configurations" --data
@cluster_env.curl_new
```

- Run the following command to PUT the update:

```

curl -u <ambari-user>:<ambari-password> -H "X-Requested-By: ambari" -X
PUT "http://<ambari-server>:<ambari-port>/api/v1/clusters/<cluster-name>" -d '${
 "Clusters": {
 "desired_config": {
 "type": "cluster-env",
 "tag": "version2"
 }
 }
}'

```

For example, on the Ambari server host, run:

```

[root@c16f1n07 ~]# curl -u admin:admin -H "X-Requested-By: ambari" -X
PUT "http://localhost:8080/api/v1/clusters/hdfs264" -d '${
 "Clusters": {
 "desired_config": {
 "type": "cluster-env",
 "tag": "version2"
 }
 }
}'

```

#### *Verifying environment:*

Refer to the “Hadoop test case scenarios” on page 57 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 307, 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 on page 52.

- f. On the HDFS Transparency node selected in 2c on page 52, 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 306 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 /etc/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.AHSProxy: 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 GSSEException: 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 GSSEException: 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 299.

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.ql.exec.DDLTask.
MetaException(message:java.security.AccessControlException:
Permission denied: user=hdp-user1, access=WRITE, inode="/user/hive":root:root:drwxr-xr-x
at org.apache.hadoop.hdfs.server.namenode.FSPermissionChecker.check(FSPermissionChecker.java:319)
at org.apache.hadoop.hdfs.server.namenode.FSPermissionChecker.checkPermission(FSPermissionChecker.java:219)
at org.apache.hadoop.hdfs.server.namenode.GPFSPermissionChecker.checkPermission(GPFSPermissionChecker.java:86)
at org.apache.ranger.authorization.hadoop.RangerHdfsAuthorizer$RangerAccessControlEnforcer.checkDefaultEnforcer
(RangerHdfsAuthorizer.java:428)
at org.apache.ranger.authorization.hadoop.RangerHdfsAuthorizer$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	Drop remote database if EXISTS cascade.
1. LOAD data localinpath	2	Create <i>remote_db</i> with Hive warehouse on the IBM Spectrum Scale HDFS Transparency cluster.
2. INSERT into table	3	Create internal nonpartitioned table on <i>remote_db</i> .
3. INSERT Overwrite TABLE	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> .

<b>Test case name</b>	<b>Step</b>	<b>Descriptions</b>
DML operations	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 <b>tpcds-build.sh</b> to build TPC-DS data generator.
	3	Run <b>tpcds-setup</b> 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 <b>query52.sqlscript</b> .
	3	Run <b>query55.sqlscript</b> .
	4	Run <b>query91.sqlscript</b> .
	5	Run <b>query42.sql script</b> .
	6	Run <b>query12.sqlscript</b> .
	7	Run <b>query73.sqlscript</b> .
	8	Run <b>query20.sqlscript</b> .
	9	Run <b>query3.sqlscript</b> .
	10	Run <b>query89.sqlscript</b> .
	11	Run <b>query48.sqlscript</b> .

*Running TPC-DS test:*

This topic lists the steps to run a TPC-DS test.

1. Prepare Hive-testbench by running the `tpcds-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 <b>hadoop dfs</b> 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 /etc/redhat-release on HDP (local native HDFS) file system with <b>hadoop dfs -put</b> .
	6	Put local file /etc/redhat-release on IBM Spectrum Scale HDFS Transparency cluster with <b>hadoop dfs -put</b> .
	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).

<b>Test case name</b>	<b>Step</b>	<b>Description</b>
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.

<b>Test case name</b>	<b>Step</b>	<b>Description</b>
DDL operations	1	Drop remote database if EXISTS cascade.
1. LOAD data local inpath	2	Create <code>remote_db</code> with hive warehouse on remote IBM Spectrum Scale HDFS Transparency cluster.
2. INSERT into table	3	Create internal nonpartitioned table on <code>remote_db</code> .
3. INSERT Overwrite TABLE	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 <code>remote_db</code> .
	8	INSERT into table from internal nonpartitioned table.
	9	Create internal partitioned table on <code>remote_db</code> .
	10	INSERT OVERWRITE TABLE from internal nonpartitioned table.
	11	Create external nonpartitioned table on <code>remote_db</code> .
	12	Drop local database if EXISTS cascade.
	13	Create <code>local_db</code> with hive warehouse on local native HDFS cluster.
	14	Create internal nonpartitioned table on <code>local_db</code> .
	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 <code>local_db</code> .
	19	INSERT into table from internal nonpartitioned table.
	20	Create internal partitioned table on <code>local_db</code> .
	21	INSERT OVERWRITE TABLE from internal nonpartitioned table.
	22	Create external nonpartitioned table on <code>local_db</code> .

<b>Test case name</b>	<b>Step</b>	<b>Description</b>
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.

<b>Test case name</b>	<b>Step</b>	<b>Description</b>
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 <b>table-level statistics</b> command on external nonpartitioned table.
	2	Run <b>DESCRIBE EXTENDED</b> to check the statics of the nonpartitioned table.
	3	Run <b>column-level statistics</b> command on internal partitioned table.
	4	Run <b>DESCRIBE EXTENDED</b> to check the statics of the partitioned table.

*DistCp in Kerberized and non-Kerberized cluster cases:*

Test Case	Step	Description
Distcp	1	Use <b>distcp</b> to copy sample file from native HDFS to remote IBM Spectrum Scale/HDFS Transparency cluster.
	2	Use <b>distcp</b> 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 GSSEException: 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
```

3. hive> CREATE database remote\_db2 COMMENT 'Holds all the tables data in remote HDFS Transparency cluster' LOCATION hdfs://c16f1n13.gpfs.net:8020/user/hive/remote\_db2;  
FAILED: Execution Error, return code 1 from org.apache.hadoop.hive.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":"AAAAAAAABAAAAAAA","cc_rec_start_date":"1998-01-01",
"cc_rec_end_date":"","cc_closed_date_sk":null,"cc_open_date_sk":2450952,"cc_name":"NY Metro",
"cc_class":"large","cc_employees":135,"cc_sq_ft":76815,"cc_hours":"8AM-4PM","cc_manager":"Bob Belcher",
"cc_mkt_id":6,"cc_mkt_class":"More than other authori","cc_mkt_desc":"Shared others could not count
fully dollars. New members ca","cc_market_manager":"Julius Tran","cc_division":3,"cc_division_name":
"pri","cc_company":6,"cc_company_name":"cally","cc_street_number":"730","cc_street_name":"Ash Hill",
"cc_street_type":"Boulevard","cc_suite_number":"Suite 0","cc_city":"Fairview","cc_county":
"Williamson County","cc_state":"TN","cc_zip":"35709","cc_country":"United States","cc_gmt_offset":
-5.0,"cc_tax_percentage":0.11}
at org.apache.hadoop.hive.ql.exec.MapOperator.process(MapOperator.java:565)
at org.apache.hadoop.hive.ql.exec.mr.ExecMapper.map(ExecMapper.java:163)
... 8 more
Caused by: org.apache.hadoop.hive.ql.metadata.HiveException: org.apache.hadoop.hive.ql.metadata.HiveException:
java.io.IOException: Failed on local exception: java.io.IOException:
org.apache.hadoop.security.AccessControlException: Client cannot authenticate via:[TOKEN, KERBEROS];
Host Details : local host is: "c16f1n07.gpfs.net/192.168.172.7"; 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.localtbl1 as select * from remote_db_gpfs_ranger.passwd_int_part;
Query ID = hdp-user1_20180319000550_ba08afa2-bbc3-4636-a6dd-c2c9564bfaf3
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1520923434038_0018)

 VERTICES STATUS TOTAL COMPLETED RUNNING PENDING FAILED KILLED

Map 1 SUCCEEDED 1 1 0 0 0 0

VERTICES: 01/01 [======>>] 100% ELAPSED TIME: 4.07 s

Moving data to directory hdfs://c16f1n07.gpfs.net:8020/user/hive/local_db_hdfs_ranger/localtbl1
Table local_db_hdfs_ranger.localtbl1 stats: [numFiles=1, numRows=84, totalSize=3004, rawDataSize=2920]
OK
Time taken: 6.584 seconds

hive> create table local_db_hdfs_ranger.localtbl2 as select * from local_db_hdfs_ranger.passwd_int_part;
Query ID = hdp-user1_20180319000658_90631a73-e34e-4919-a30a-05a66769ab41
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1520923434038_0018)

 VERTICES STATUS TOTAL COMPLETED RUNNING PENDING FAILED KILLED

Map 1 SUCCEEDED 1 1 0 0 0 0

VERTICES: 01/01 [======>>] 100% ELAPSED TIME: 6.16 s

Moving data to directory hdfs://c16f1n07.gpfs.net:8020/user/hive/local_db_hdfs_ranger/localtbl2
Table local_db_hdfs_ranger.localtbl2 stats: [numFiles=1, numRows=84, totalSize=3004, rawDataSize=2920]
OK
Time taken: 7.904 seconds

```

## Open Source Apache viewfs support

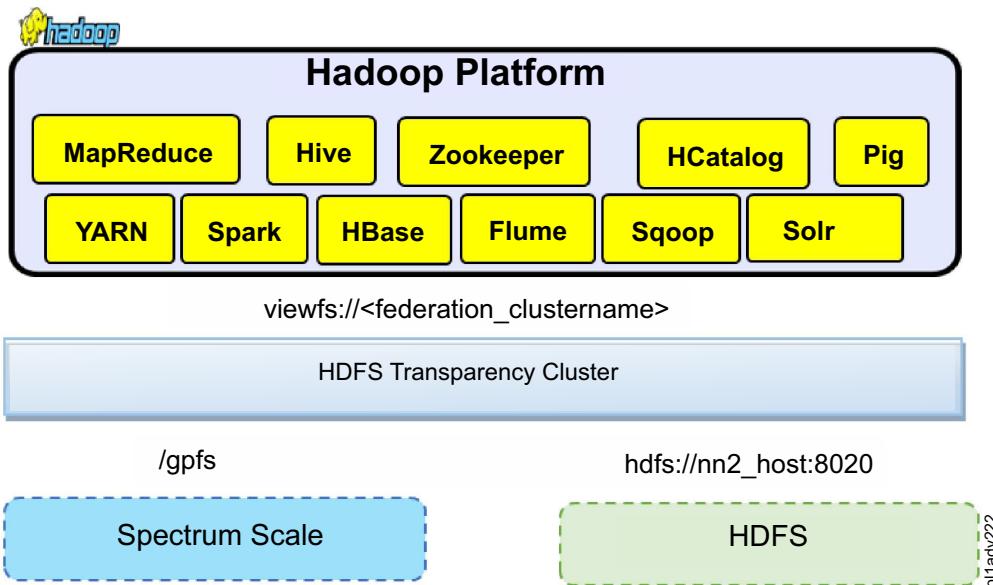
Federation was added to HDFS to improve the HDFS NameNode horizontal scaling. In HDFS transparency, federation is used to co-exist IBM Spectrum Scale file systems and HDFS file system. The Hadoop applications can get input data from the native HDFS, analyze the input and write the output to the IBM Spectrum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

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 called federating clusterB with clusterA. If you want your applications running in clusterB to process data from clusterA, you need to update the configuration for federation in clusterB. This is called federating clusterA with clusterB.

### Single viewfs namespace between IBM Spectrum Scale and native HDFS:

Before configuring viewfs support, ensure that you have configured the HDFS Transparency cluster (see “Hadoop cluster planning” on page 10 and “Installation and configuration of HDFS transparency” on page 17).



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

#### *Single viewfs namespace between IBM Spectrum Scale and native HDFS –Part I:*

This topic describes the steps to get a single namespace by federating HDFS transparency namespace into native HDFS namespace. If you take HortonWorks HDP, you could change the configurations from the **Ambari GUI > HDFS > Configs**.

In this mode, `nn1_host` is HDFS Transparency NameNode. You have another native HDFS cluster. After you federate native HDFS into HDFS Transparency, your applications can access the data from both HDFS Transparency and native HDFS with the schema `fs.defaultFS` defined in the HDFS Transparency cluster. All configuration changes are done on the HDFS Transparency side and your Hadoop client nodes. This mode does not need configurations change on native HDFS cluster.

1. Shut down the HDFS Transparency cluster daemon by running the following command from one of the HDFS transparency nodes in the cluster:

```
mmhadoopctl connector stop
```

2. On the `nn1_host`, add the following configuration settings in `/usr/lpp/mmfs/hadoop/etc/hadoop/core-site.xml` (for HDFS Transparency 2.7.3-x) or `/var/mmfs/hadoop/etc/hadoop/core-site.xml` (for HDFS Transparency 3.0.x):

```
<configuration>
<property>
<name>fs.defaultFS</name>
<value>viewfs://<viewfs_clustername></value>
<description>The name of the namespace</description>
</property>

<property>
<name>fs.viewfs.mounttable.<viewfs_clustername>.link.<viewfs_dir1></name>
<value>hdfs://nn1_host:8020/<mount_dir></value>
<description>The name of the Spectrum Scale file system</description>
</property>

<property>
<name>fs.viewfs.mounttable.<federation_clustername>.link.<viewfs_dir2></name>
<value>hdfs://nn2_host:8020/<mount_dir></value>
<description>The name of the hdfs file system</description>
</property>
</configuration>
```

**Note:** Change <viewfs\_clustername> and <mount\_dir> according to your cluster configuration. In this example, the *nn1\_host* refers to the HDFS transparency NameNode and the *nn2\_host* refers to the native HDFS NameNode.

Once the federation configuration changes are in effect on the node, the node will only see the directories that are specified in the core-site.xml file. For the above configurations, you can only see the two directories /<viewfs\_dir1> and /<viewfs\_dir2>.

3. On *nn1\_host*, add the following configuration settings in /var/mmfs/hadoop/etc/hadoop/hdfs-site.xml (for HDFS Transparency 3.0.x).

```
<configuration>
<property>
 <name>dfs.nameservices</name>
 <value>nn1,nn2</value>
</property>

<property>
 <name>dfs.namenode.rpc-address.nn1</name>
 <value>nn1-host:8020</value>
</property>

<property>
 <name>dfs.namenode.rpc-address.nn2</name>
 <value>nn2-host:8020</value>
</property>

<property>
 <name>dfs.namenode.http-address.nn1</name>
 <value>nn1-host:50070</value>
</property>

<property>
 <name>dfs.namenode.http-address.nn2</name>
 <value>nn2-host:50070</value>
</property>
</configuration>
```

4. On *nn1\_host*, synchronize the configuration changes with the other HDFS transparency nodes by running the following command:

For HDFS Transparency 2.7.3-x:

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

For HDFS Transparency 3.0.x:

```
mmhadoopctl connector syncconf /var/mmfs/hadoop/etc/hadoop/
```

**Note:** The following output messages from the above command for the native HDFS NameNode, *nn2-host*, can be seen:

```
scp: /usr/lpp/mmfs/hadoop/etc/hadoop//: No such file or directory
```

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.

For HDFS Transparency 3.0.x:

```
#mmhadoopctl connector syncconf /var/mmfs/hadoop/etc/hadoop
```

5. On *nn1\_host*, start all the HDFS transparency cluster nodes by running the following command:  
**# mmhadoopctl connector start**

**Note:** The following warning output messages from the above command for the native HDFS NameNode, *nn2-host* can be seen:

```
nn2-host: bash: line 0: cd: /usr/lpp/mmfs/hadoop: No such file or directory
nn2-host: bash: /usr/lpp/mmfs/hadoop/sbin/hadoop-daemon.sh: No such file or directory
```

These messages are displayed because HDFS Transparency is not installed on the native HDFS NameNode. Therefore, these messages can be ignored.

To avoid the above messages, run the following commands:

- a. On *nn1-host*, run the following command as root to start the HDFS Transparency NameNode:

```
cd /usr/lpp/mmfs/hadoop; /usr/lpp/mmfs/hadoop/sbin/hadoop-daemon.sh
--config /usr/lpp/mmfs/hadoop/etc/hadoop
--script /usr/lpp/mmfs/hadoop/sbin/gpfs start namenode
```

- b. On *nn1-host*, run the following command as root to start the HDFS Transparency DataNode:

```
cd /usr/lpp/mmfs/hadoop; /usr/lpp/mmfs/hadoop/sbin/hadoop-daemons.sh
--config /usr/lpp/mmfs/hadoop/etc/hadoop
--script /usr/lpp/mmfs/hadoop/sbin/gpfs start datanode
```

**Note:** If you deployed IBM BigInsights IOP, the 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 77 on the node *nn2-host* with the correct path for *core-site.xml* and the *hdfs-site.xml* according to the Hadoop distribution.

If running with the open source Apache Hadoop, the location of the *core-site.xml* and the *hdfs-site.xml* is in *\$YOUR\_HADOOP\_PREFIX/etc/hadoop/*. The *\$YOUR\_HADOOP\_PREFIX* is the location of the Hadoop package. If running with IBM BigInsights IOP, then Ambari currently does not support viewfs configuration. You will have to manually update the configurations under */etc/hadoop/conf/*.

**Note:** If you want to see all the directories from the native HDFS shown up in viewfs, define all the native HDFS directories in the core-site.xml.

If you have a secondary NameNode configured in native HDFS, update the following configuration in the hdfs-site.xml:

```
<property>
 <name>dfs.namenode.secondary.http-address.nn2-host</name>
 <value>secondaryNameNode-host:50090</value>
</property>

<property>
 <name>dfs.secondary.namenode.keytab.file.nn2-host</name>
 <value>/etc/security/keytabs/nn.service.keytab</value>
</property>
```

**Note:** If you have deployed IBM BigInsights IOP, it will generate the key `dfs.namenode.secondary.http-address` and `dfs.secondary.namenode.keytab.file` by default. For viewfs, modify the `hdfs-site.xml` file with the correct values according to your environment.

3. Synchronize the updated configurations from the `nn2-host` node to all the other native HDFS nodes and start the native HDFS services.

If running with open source Apache Hadoop, you need to use the `scp` command to synchronize the `core-site.xml` and the `hdfs-site.xml` from the host `nn2-host` to all the other native HDFS nodes. Start the native HDFS service by running the following command:

```
$YOUR_HOME_PREFIX/sbin/start-dfs.sh
```

If IBM BigInsights IOP is running, synchronize the updated configurations manually to avoid the updated viewfs configurations overwritten by Ambari.

**Note:** Check the configurations under `/etc/hadoop/conf` to ensure that all the changes have been synchronized to all the nodes.

4. Start the native HDFS service.

If you are running open source Hadoop, start the native HDFS service on the command line:

```
$YOUR_HADOOP_PREFIX/bin/start-dfs.sh
```

If you deployed IBM BigInsights IOP, Ambari does not support viewfs configuration. Therefore, you must start the native HDFS services manually.

- a. Start native HDFS NameNode.

Log in to `nn2-host` as root, run `su - hdfs` to switch to the `hdfs` UID and then run the following command:

```
/usr/iop/current/hadoop-client/sbin/hadoop-daemon.sh --config
/usr/iop/current/hadoop-client/conf start namenode
```

- b. Start the native HDFS DataNode.

Log in to the DataNode, run `su - hdfs` to switch to the `hdfs` UID and then run the following command:

```
/usr/iop/current/hadoop-client/sbin/hadoop-daemon.sh --config
/usr/iop/current/hadoop-client/conf start datanode
```

**Note:** Run the above command on each DataNode.

Log in to the 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 viewfs. If the open source Apache Hadoop is running, the location of `core-site.xml` and `hdfs-site.xml` is in `$YOUR_HADOOP_PREFIX/etc/hadoop/`. The `$YOUR_HADOOP_PREFIX` is the location of the Hadoop package. If another Hadoop distro is running, see “Known limitations” on page 84.

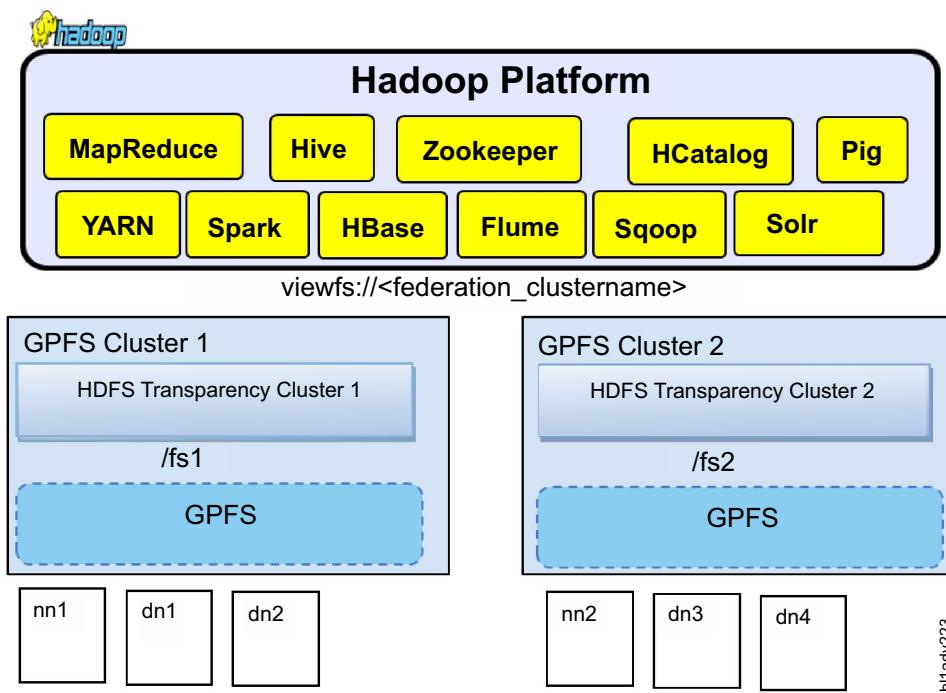
If IBM BigInsights IOP is running, `core-site.xml` and `hdfs-site.xml` are located in `/etc/hadoop/conf/`.

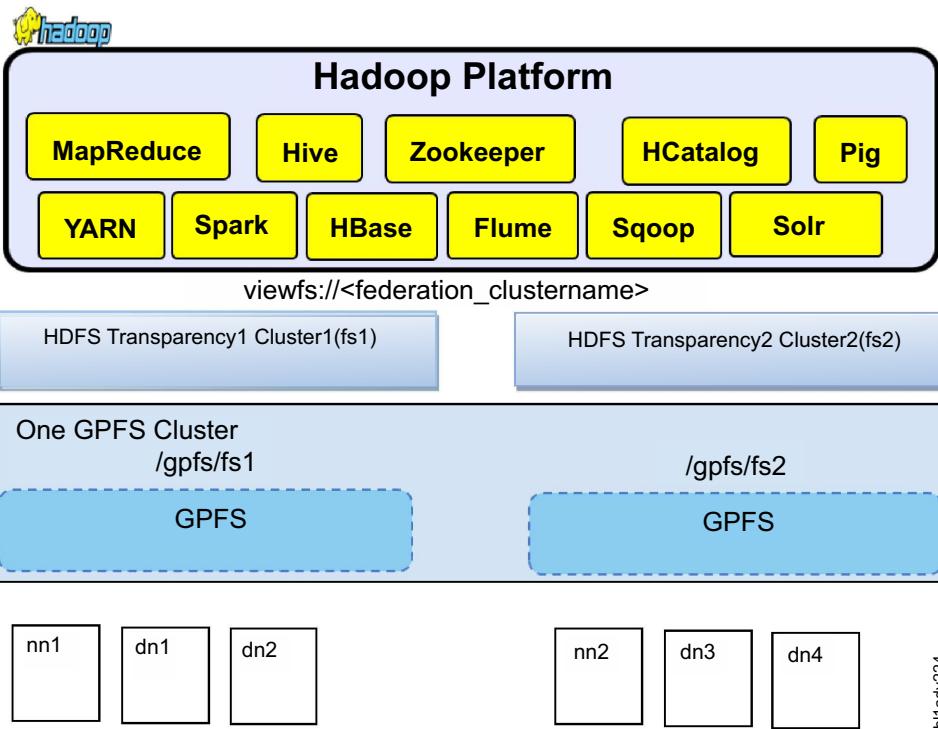
6. To ensure that the viewfs file system is functioning correctly, run the following command:  
`hadoop fs -ls /`

#### Single viewfs namespace between two IBM Spectrum Scale file systems:

You can get a single viewfs 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 10 and “Installation and configuration of HDFS transparency” on page 17), 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 10 and “Installation and configuration of HDFS transparency” on page 17 to configure HDFS transparency cluster 1 and HDFS transparency cluster 2 for each file system.

1. To stop the HDFS transparency services, run the **mmhadoopctl connector stop** on both HDFS transparency clusters.
2. On the *nn1* host, add the following configuration settings in */usr/lpp/mmfs/hadoop/etc/hadoop/core-site.xml* (for HDFS Transparency 2.7.3-x) or */var/mmfs/hadoop/etc/hadoop/core-site.xml* (for HDFS Transparency 3.0.x):
 

```
<configuration>
<property>
<name>fs.defaultFS</name>
<value>viewfs://<viewfs_clustername></value>
<description>The name of the viewfs file system</description>
</property>

<property>
<name>fs.viewfs.mounttable.<viewfs_clustername>.link.<mount_dir></name>
<value>hdfs://nn1_host:8020/<mount_dir></value>
<description>The name of the gpfs file system</description>
</property>

<property>
<name>fs.viewfs.mounttable.<viewfs_clustername>.link.<mount_dir></name>
```

```

<value>hdfs://nn2_host:8020/<mount_dir></value>
<description>The name of the hdfs file system</description>
</property>
</configuration>

```

**Note:** Change <viewfs\_clustername> and <mount\_dir> according to your cluster. Change *nn1\_host* and *nn2\_host* accordingly.

3. On *nn1\_host*, add the following configuration settings in `hdfs-site.xml`.

```

<configuration>
<property>
 <name>dfs.nameservices</name>
 <value>nn1,nn2</value>
</property>

<property>
 <name>dfs.namenode.rpc-address.nn1</name>
 <value>nn1:8020</value>
</property>

<property>
 <name>dfs.namenode.rpc-address.nn2</name>
 <value>nn2:8020</value>
</property>

<property>
 <name>dfs.namenode.http-address.nn1</name>
 <value>nn1:50070</value>
</property>

<property>
 <name>dfs.namenode.http-address.nn2</name>
 <value>nn2:50070</value>
</property>
</configuration>

```

4. On *nn1\_host*, take `mmhadoopctl connector syncconf /usr/lpp/mmfs/hadoop/etc/hadoop` (for HDFS Transparency 2.7.3-x) or `mmhadoopctl connector syncconf /var/mmfs/hadoop/etc/hadoop` (for HDFS Transparency 3.0.x) to synchronize the configurations from *nn1\_host* to all other HDFS Transparency nodes.

**Note:** The above must be done for each node in the HDFS Transparency Cluster 1. For example, change the hostX accordingly and run it for each node in the HDFS Transparency Cluster1.

5. On *nn2\_host*, perform Step 1 through Step 4.

**Note:** If you only want to federate HDFS Transparency Cluster2 into HDFS Transparency Cluster1, Step 5 is not needed.

6. On *nn1\_host*, start the HDFS transparency cluster:

- a. On nn1, run the following command as root to start HDFS Transparency Cluster1 NameNode:

```
#cd /usr/lpp/mmfs/hadoop; /usr/lpp/mmfs/hadoop/sbin/hadoop-daemon.sh
--config /usr/lpp/mmfs/hadoop/etc/hadoop --script /usr/lpp/mmfs/hadoop/sbin/gpfs start namenode
```

- b. On nn1, run the following command as root to start HDFS Transparency Cluster1 DataNode:

```
cd /usr/lpp/mmfs/hadoop; /usr/lpp/mmfs/hadoop/sbin/hadoop-daemons.sh
--config /usr/lpp/mmfs/hadoop/etc/hadoop --script /usr/lpp/mmfs/hadoop/sbin/gpfs start datanode
```

**Note:** If you deployed IBM BigInsights IOP, IBM 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."
9. Restart the Hadoop applications on both clusters.

**Note:** You should always keep the native HDFS service non-functional if you select HDFS Transparency.

10. To ensure that the viewfs is functioning correctly, run the `hadoop fs -ls /` command.

#### Known limitations:

This topic lists the known limitations for viewfs support.

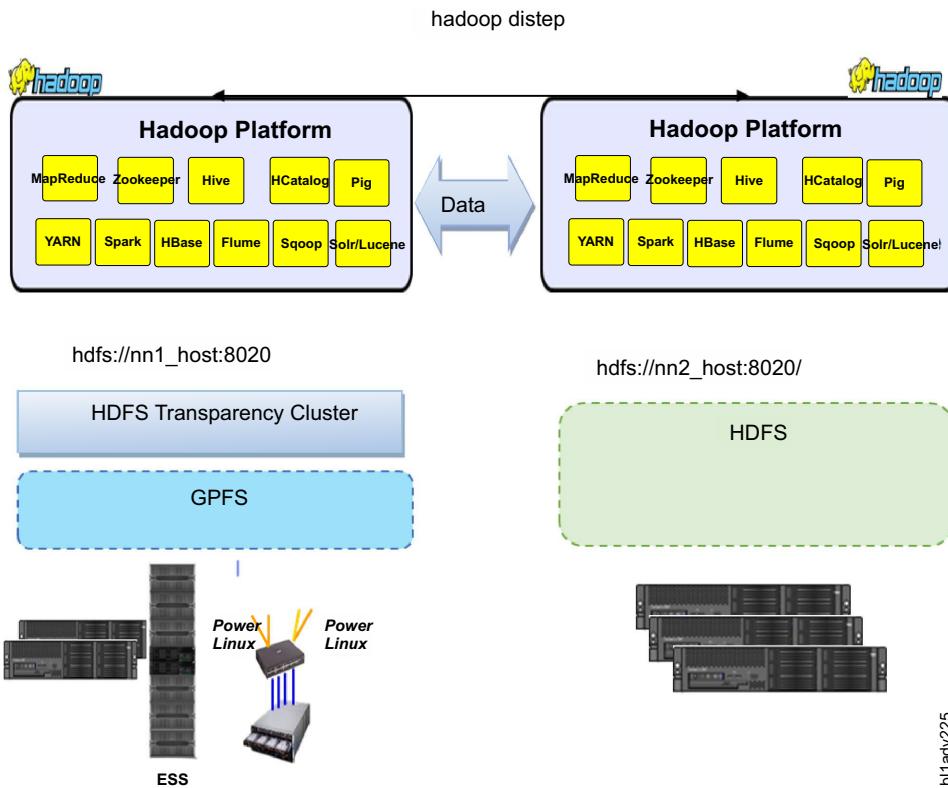
- All the changes in `/usr/lpp/mmfs/hadoop/etc/hadoop/core-site.xml` and `/usr/lpp/mmfs/hadoop/etc/hadoop/hdfs-site.xml` must be updated in the configuration file that is used by the Hadoop distributions. However, Hadoop distributions occasionally manage their configuration and the management interface might not support the key used for viewfs, 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 viewfs.
- The native HDFS and HDFS transparency cannot be run over the same node because of the network port number conflict.
- If you select to join both the native HDFS and HDFS transparency, configure the native HDFS cluster and make the native HDFS service function. Configure the viewfs for native HDFS and HDFS transparency.

For a new native HDFS cluster, while starting the service for the first time, DataNode registers itself with the NameNode. The HDFS Transparency NameNode does not accept any registration from the native HDFS DataNode. Therefore, an exception occurs if you configure a new native HDFS cluster, federate it with HDFS transparency, and then try to make both clusters (one native HDFS cluster and another HDFS Transparency cluster) function at the same time.

- Start and stop the native HDFS cluster or the HDFS Transparency cluster separately if you want to maintain them.

## 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. This is applicable only for HDFS Transparency 2.7.3-x. From HDFS Transparency 3.0.x, the configuration `gpfs.supergroup` has been removed from HDFS Transparency.

## **Issue 2: Access time exception while copying files from IBM 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** (for HDFS Transparency 2.7.3-x) or **/var/mmfs/hadoop/etc/hadoop** (for HDFS Transparency 3.0.0) 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/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.

### Configuring MySQL for Ranger:

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

1. Create a HDP 2.6/3.0 or 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<sup>TM</sup> 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**.



3. On the Choose Services page, select **Ranger**.

The system displays the Ranger Requirements page.

**Ranger Requirements**

- 1. You must have an **MySQL/Oracle/Postgres/MSSQL/SQL Anywhere Server** database instance running to be used by Ranger.
- 2. 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)
- 3. Ensure that the access for the DB Admin user is enabled in DB server from any host.
- 4. 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={/jdb
c/driver/path}
```

**Cancel** **Proceed**

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**.

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 host
<input type="button" value="MYSQL"/>	c8f2n07.gpfs.net
Ranger DB name	Driver class name for a JDBC Ranger database
ranger	com.mysql.jdbc.Driver
Ranger DB username	Ranger DB password
rangeradmin	*****
JDBC connect string	
jdbc:mysql://c8f2n07.gpfs.net/ranger	

b11adm052

- 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	Database Administrator (DBA) password
rangerdba	*****
JDBC connect string for root user	
jdbc:mysql://c8f2n07.gpfs.net	
<input type="button" value="Test Connection"/>	Connection OK

b11adm053

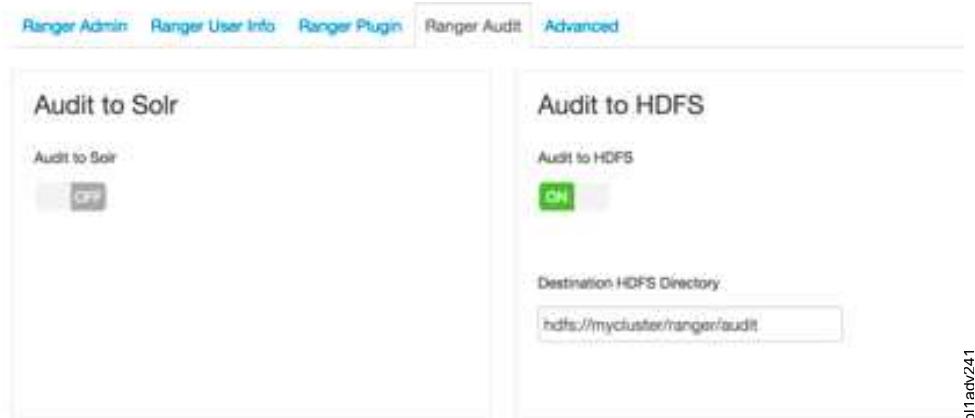
- For IOP 4.1/4.2, set the Ranger Audit DB username value to *rangerlogger*, and for the Ranger Audit DB password, set the value to *YES*.

## Audit to DB



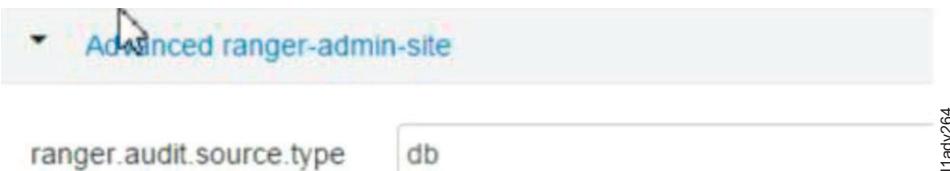
The screenshot shows the 'Audit to DB' section of the Ranger Audit configuration. A green 'ON' button is selected. To its right is a 'Ranger Audit DB name' field containing 'ranger\_audit'. Below this are fields for 'Ranger Audit DB username' ('rangerlogger') and 'Ranger Audit DB password' (two masked fields). A small text 'b1adv263' is visible on the right.

- For IOP 4.2.5/HortonWorks 2.6/HortonWorks 3.0, set the Ranger Audit DB username value to `rangerlogger` and Ranger Audit DB password value to `YES`.



The screenshot shows the 'Audit to Solr' and 'Audit to HDFS' sections of the Ranger Audit tab. The 'Audit to Solr' section has a greyed-out 'ON' button. The 'Audit to HDFS' section has a green 'ON' button. Below it is a 'Destination HDFS Directory' field containing 'hdfs://mycluster/ranger/audit'. A small text 'b1adv241' is visible on the right.

- 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`.



The screenshot shows the 'Advanced ranger-admin-site' configuration page. A dropdown menu is open next to the 'ranger.audit.source.type' field, which contains the value 'db'. A small text 'b1adv264' is visible on the right.

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

Database Administrator (DBA) username  
rangerdba

Database Administrator (DBA) password  
\*\*\*\*\*

JDBC connect string for root user  
jdbc:mysql://c8f2n07.gpf.net

Test Connection Connection OK ✓

b1adm055

**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

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

There are 2 configuration changes in 1 service [Show Details](#)

Ranger Admin Ranger User Info Ranger Plugin Ranger Audit Advanced

**Ranger Plugin**

**HDFS Ranger Plugin** ON

**Hbase Ranger Plugin** OFF

b1adv242

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

**Dependent Configurations**

**Recommended Changes**

Based on your configuration changes, Ambari is recommending the following dependent configuration changes.  
Ambari will update all checked configuration changes to the Recommended Value. Uncheck any configuration to retain the Current Value.

Property	Service	Config Group	File Name	Current Value	Recommended Value
ranger-hdfs-plugin-enabled	HDFS	Default	ranger-hdfs-plugin-provider	No	Yes
dfs.namenode.inode.attributes.provider.class	HDFS	Default	dfs-site	Property undefined	org.apache.ranger.authorization.hadoop.RangerHdfsAuthorizer

Cancel OK

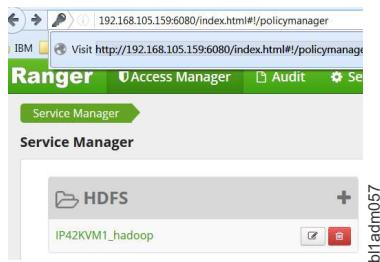
b1adv274

- 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



## 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 17.

### 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 for HDFS Transparency 2.7.3-x or /var/mmfs/hadoop/etc/hadoop/hdfs-site.xml for HDFS Transparency 3.0.0 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 (for HDFS Transparency 2.7.3-x) or /var/mmfs/hadoop/etc/hadoop (for HDFS Transparency 3.0.0) on all the NameNode and DataNodes: ranger-hdfs-audit.xml, ranger-hdfs-security.xml, ranger-policymgr-ss1.xml, ranger-security.xml from the path /etc/hadoop/conf.
3. Edit the /usr/lpp/mmfs/hadoop/etc/hadoop/hadoop-env.sh (for HDFS Transparency 2.7.3-x) or /var/mmfs/hadoop/etc/hadoop/hadoop-env.sh (for HDFS Transparency 3.0.0) 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
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 IOP 4.2.5:

Change the above version string 4.2.0.0 into “4.2.5.0-0000”.

#### For HortonWorks 2.6 and 3.X:

```
for f in /usr/hdp/<your-HDP-version>/ranger-hdfs-plugin/lib/*.jar;
do
 export HADOOP_CLASSPATH=$HADOOP_CLASSPATH:$f
done

export HADOOP_CLASSPATH=$HADOOP_CLASSPATH:/usr/share/java/mysql-connector-java.jar
```

```

for f in /usr/hdp/<your-HDP-version>/hadoop/client/jersey-client.jar;
do
 export HADOOP_CLASSPATH=$HADOOP_CLASSPATH:$f
done

```

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

```
[root@c8f2n03kvm2 lib]# service mysql 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).

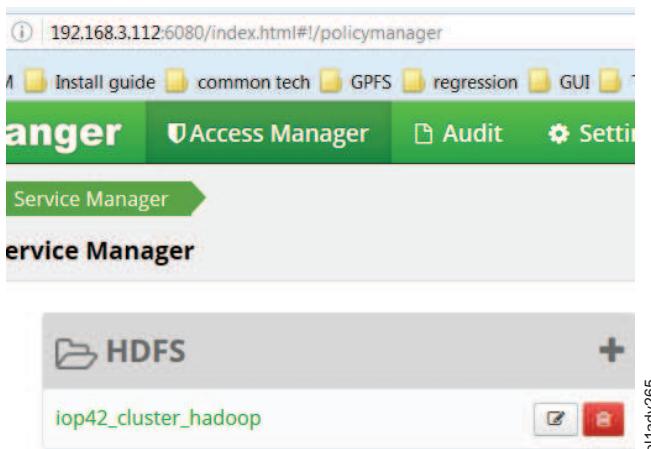


Figure 14. Service manager

2. Go to Service Manager > iop42\_cluster\_hadoop > Add New Policy.

**Note:** The **iop42\_cluster\_hadoop** is the Ranger service name created by the user.

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.

Label	Description
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 15. 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
```

Permissions	Select Group	Select User	Permissions	Delegate Admin
Select Group	u	Write	<input type="checkbox"/>	<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/gpfs-site.xml`.

To disable Ranger support, modify the `/usr/lpp/mmfs/hadoop/etc/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 94 section was not followed during Ranger setup.

*Table 7. Ranger directory permission issues*

User name	Directory Permission in Ranger	directory permission on native HDFS	Results
fvtuser (not super user, not the owner of the directory)	r--	--x	Expectation for user fvtuser 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.
fvtuser (not super user, not the owner of the directory)	--x	-w-	Expectation for user fvtuser 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.
fvtuser (not super user, not the owner of the directory)	--x	r--	Expectation for user fvtuser 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:

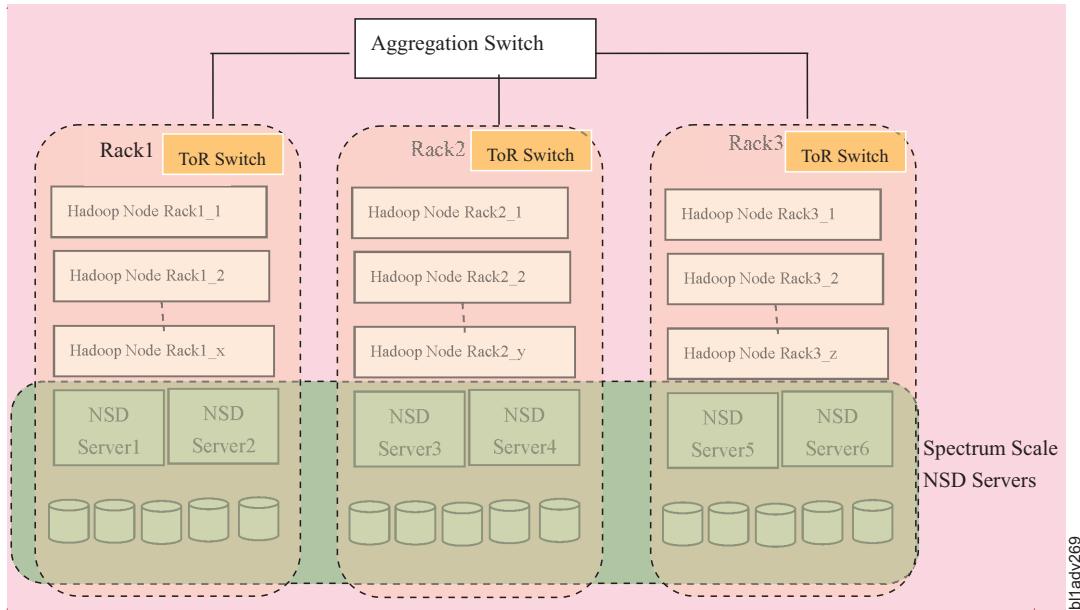


Figure 16. Topology of rack awareness locality for shared storage

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 16. 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` (for HDFS Transparency 2.7.x) or `/var/mmfs/hadoop/etc/hadoop/core-site.xml` (for HDFS Transparency 3.0.x):

```
<property>
 <name>net.topology.table.file.name</name>
 <value>/usr/lpp/mmfs/hadoop/etc/hadoop/topology.data</value>
</property>
<property>
 <name>net.topology.node.switch.mapping.impl</name>
 <value>org.apache.hadoop.net.TableMapping</value>
</property>
```

3. On the HDFS Transparency NameNode, create the topology in `/usr/lpp/mmfs/hadoop/etc/hadoop/topology.data` (for HDFS Transparency2.7.x) or `/var/mmfs/hadoop/etc/hadoop/topology.data` (for HDFS Transparency 3.0.x):

```
vim topology.data
192.168.200.57 /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 16 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` (for HDFS Transparency 2.7.x) or `/var/mmfs/hadoop/etc/hadoop/gpfs-site.xml` (for HDFS Transparency 3.0.x):

```
<property>
 <name>gpfs.storage.type</name>
 <value>rackaware</value>
</property>
```

5. On the HDFS Transparency NameNode, run the `mmhadoopctl connector syncconf` `/usr/lpp/mmfs/hadoop/etc/hadoop` (for HDFS Transparency 2.7.x) or `mmhadoopctl connector syncconf` `/var/mmfs/hadoop/etc/hadoop` (for HDFS Transparency 3.0.x) command to synchronize the configurations to all the HDFS Transparency nodes.

**Note:** If you have HDP with Ambari Mpack 2.4.2.1 and later, the **connector syncconf** cannot be executed. Ambari manages the configuration syncing through the database.

6. **(optional):** To configure multi-cluster between IBM 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 18 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> a11 -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.syncconclose 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.syncconclose
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 (for HDFS Transparency 2.7.3-x) or /var/mmfs/hadoop/etc/hadoop/gpfs-site.xml (for HDFS Transparency 3.0.x).

Configure the HDFS Transparency on the primary file system as the 1st file system defined in the **gpfs.mnt.dir** field.

Once the primary file system configuration is working properly, enable the multiple 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 `mm1spool <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>&lt;=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` (for HDFS Transparency 2.7.3-x) or `/usr/lpp/mmfs/hadoop/share/hadoop/hdfs/hadoop-hdfs-client-<version>.jar` (for HDFS Transparency 3.0.x) into your mapreduce class path.

#### Important:

- Zero shuffle does not impact teragen-like workloads because this kind of workloads do not involve using shuffle.
- `mapreduce.task.io.sort.mb` should be `<=1024`. Therefore, the data size for each map task must not be larger than 1024MB.
- Zero shuffle creates one file from each map task for each reduce task. Assuming your job has 1000 map tasks and 300 reduce tasks, it will create at least 300K intermediate files. Considering spilling, it might create around one million intermediate inodes and remove them after the job is done. Therefore, if the `reduce-task-number*map-task-number` is more than 300,000, it is not recommended to use zero shuffle.

## Native HDFS encryption

| This is supported since HDFS Transparency 3.0.0-0 and 2.7.3-4. This requires Ranger and Ranger KMS and this has only been tested over HortonWorks stack. If you plan to enable this for open source Apache, | you should enable it on the native HDFS first and confirm it is working before you switch native HDFS | into HDFS Transparency.

To enable native HDFS encryption, configure `gpfs.ranger.enabled=true` in `gpfs-site.xml` and configure the following value for `gpfs-site.xml` from Ambari GUI:

Configuration	Value (default)
<code>gpfs.encryption.enabled</code>	true (false)
<code>gpfs.ranger.enabled</code>	true (true)

On HDP 3.0, since Ranger-KMS does not allow root account by default, you need to execute some steps to enable it:

- Add a new group in all Hadoop nodes and ensure that the group-id is same (for example, `kms_supergroup`).
- Add `kms_supergroup` to the root supplementary group list by `usermod -G`.
- On Ranger admin web UI, create a new profile and select `kms_supergroup` and assign the same permissions as hdfs user profile, ensure that the permissions include **Get Metadata** and **Generate EEK**.
- Add `root` to `hadoop.kms.blacklist.DECRYPT_EEK` in `dbks-site.xml` on Ambari GUI. The user `hdfs` and `root` are super user. By default, the user `hdfs` is configured in `hadoop.kms.blacklist.DECRYPT_EEK`. Adding the user `root` there to prohibit the user `root` to read any encrypted data through HDFS interface.

5. Save the changes and restart NameNode.

#### Known limits:

- The output of hdfs crypto -listZones might be not a full list because it only lists the encryption zones you have accessed.
- Files in HDFS snapshot against encryption zone files are not supported for read/write.

---

## Hadoop distribution support

Only the open source Apache packages, IBM BigInsights IOP 4.0/4.1/4.2, HortonWorks HDP 2.6.x and HDP 3.0 are officially supported. Cloudera is not supported.

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

## HortonWorks HDP 2.6.x and HDP 3.0 support

HortonWorks HDP 3.0 is verified over HDFS Transparency 3.0.0. Short Circuit Read can be enabled on both HDP 2.6.x and 3.0.

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 **mmlsfileset <fs-name>**) and if the fileset is an independent fileset, the **mmlsfileset** command creates the snapshot against the independent fileset. For example, if /path/to/directory is linked with fileset1 and fileset1 is an independent fileset, the above command creates snapshot against fileset1. If not, Transparency checks /path/to, then checks /path followed by / (which is /gpfs.mnt.dir/gpfs.data.dir from IBM 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/1pp/mmfs/hadoop/etc/hadoop/gpfs-site.xml and **dfs.permissions.superusergroup** in /usr/1pp/mmfs/hadoop/etc/hadoop/dfs-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/1pp/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.remotecluster.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 $1024 * \text{file-system-data-block-size}$ ( <code>mm1sfs -B</code> )
<code>dfs.namenode.fs-limits.max-xattrs-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
<code>dfs.namenode.fs-limits.max-component-length</code>	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 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.
- In HDFS, files can only be appended. If a file is uploaded into the same location with the same file name to overwrite the existing file, then HDFS can detect this according to the inode change. However, IBM Spectrum Scale supports POSIX interface and other protocol interfaces (for example, NFS/SMB) and one file could be changed from non HDFS interface. Therefore, files loaded for Hadoop services to process cannot be modified until the process completes. Else the service or job fails.

---

## Problem determination

When hitting exceptions from HDFS Transparency, you could refer “Configuration of HDFS transparency” on page 17 to check whether you have configured ulimits correctly and whether you have configured NTP to synchronize clock in the clusters first.

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



---

## | **Chapter 3. IBM Spectrum Scale Hadoop performance tuning guide**

### | **Overview**

#### | **Introduction**

- | The Hadoop ecosystem consists of many open source projects. One of the central components is the Hadoop Distributed File System (HDFS).
  - | HDFS is a distributed file system designed to run on the commodity hardware. Other related projects facilitate workflow and the coordination of jobs, support data movement between Hadoop and other systems, and implement scalable machine learning and data mining algorithms. HDFS lacks the enterprise class functions necessary for reliability, data management, and data governance. IBM's General Parallel File System (GPFS) is a POSIX compliant file system that offers an enterprise class alternative to HDFS.
  - | There are many similar tuning guides available for native HDFS. However, when you apply those tuning steps over IBM Spectrum Scale, usually you cannot get the best performance because of natural design difference between HDFS and IBM Spectrum Scale. With this section the customers can tune different Hadoop components when they run Hadoop over IBM Spectrum Scale and HDFS Transparency so that they get good performance.

#### | **Hadoop over IBM Spectrum Scale**

- | By default, Hadoop uses HDFS schema (`hdfs://<namenode>:<portnumber>`) for all the components to read data from HDFS or write data into HDFS.
- | Hadoop also supports local file schema (`file:///`) and other HCFS schema for Hadoop components to read data from or write data into other distributed file systems.
- | IBM Spectrum Scale HDFS Transparency follows the Hadoop HCFS specification and provide the HDFS RPC level implementation for Hadoop components to read data from or write data into IBM Spectrum Scale. It also supports Kerberos, federation, and distcp.

#### | **Spark over IBM Spectrum Scale**

- | Apache Spark is a fast and general engine for large scale data processing.
- | Spark supports multiple ways (such as `hdfs://`, `file://`) for applications running over Spark to access the data in distributed file systems.

#### | **Hadoop and Software Stack Version**

- | All tuning configurations mentioned in this section are against Hadoop 2.7.x and Hadoop 3.0.x. Therefore, this tuning could be applied to IBM BigInsights IOP 4.1 (Hadoop 2.7.1), IOP 4.2 (Hadoop 2.7.2), IOP 4.3 (Hadoop 2.7.3), HortonWorks 2.5/2.6 (Hadoop 2.7.x) and HortonWorks 3.0 (Hadoop 3.0.x).
- | If you are running open source community Hadoop 2.4.x/2.5.x/2.6.x, some configurations mentioned in this section might be new deprecated configurations. Therefore, refer to the Deprecated Properties section to confirm the effective configuration in Hadoop 2.4.x/2.5.x/2.6.x.

## Performance overview

This section describes the type of configurations that impact job executions from MapReduce and Hive.

## MapReduce

This section describes Yarn's job management and execution flow so that we know the impact on MapReduce job performance.

The topology of job management in Yarn is illustrated by Figure 17. The picture is from Apache Hadoop YARN website:

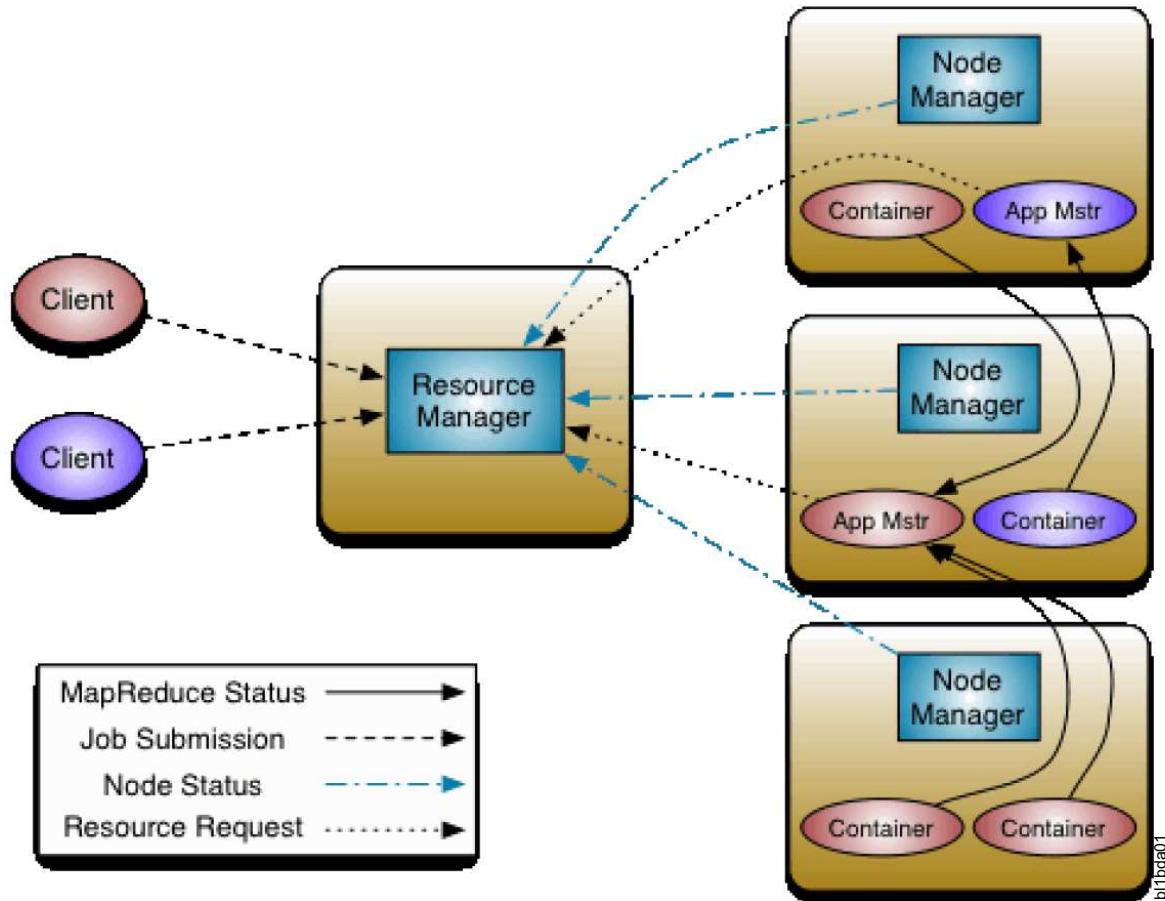


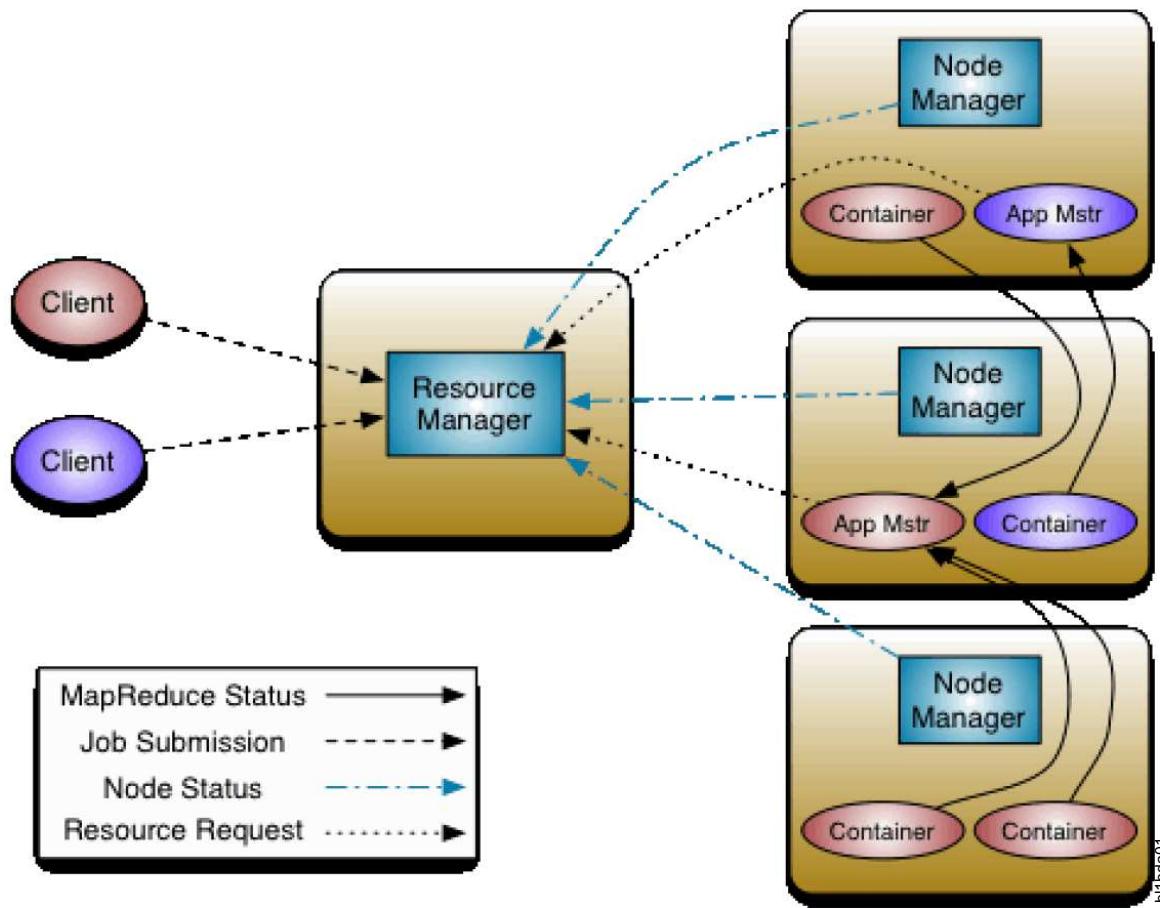
Figure 17. Topology of Job management in Yarn

After a client submits one job into Yarn, the Resource Manager receives the request and puts it into Resource Manager queue for scheduling. For each job, it has one App Master that coordinates with the Resource Manager and the Node Manager to request the resource for other tasks of the job and start these tasks after their resource is allocated.

From Figure 17, all tasks are run over the nodes on which there is Node Manager service. Only these tasks will read data from distributed file system or write data into distributed file systems. For native HDFS, if HDFS datanode services are not running over these Node Manager nodes, all the Map/Reduce jobs cannot leverage data locality for high read performance because all data have to be transferred over network.

Also, App Master of one job takes configured CPU resources and memory resources.

| To one specific map/reduce job, the execution flow is illustrated by Figure 18:



*Figure 18. MapReduce Execution Flow*

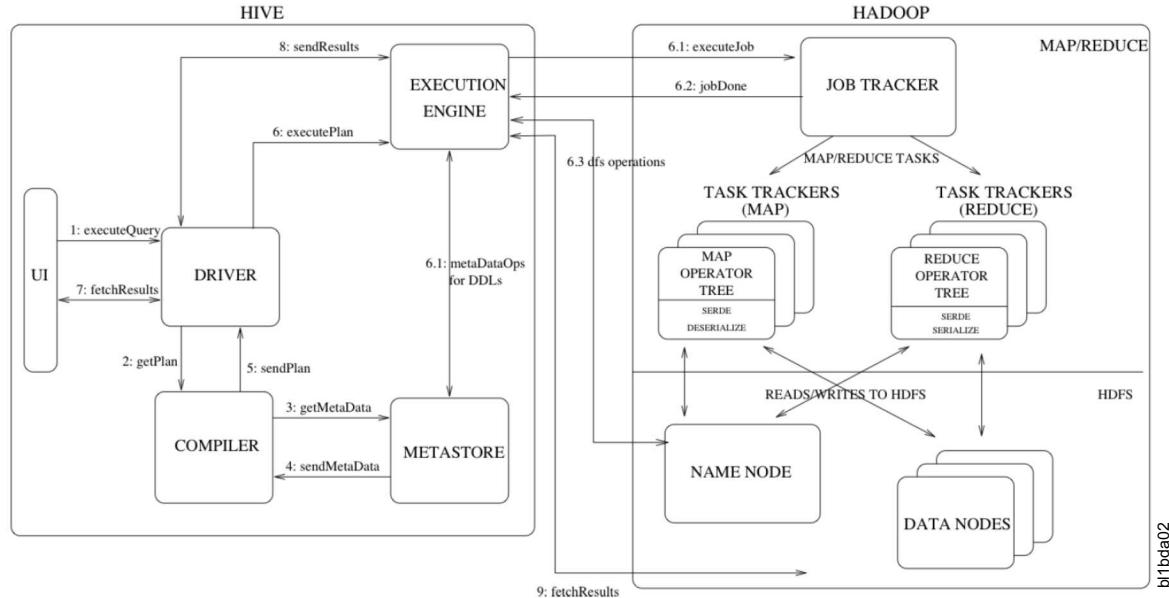
- | For each map task, it reads one split of input data from distributed file system. The split size is controlled  
| by `dfs.blocksize`, `mapreduce.input.fileinputformat.split.minsize` and  
| `mapreduce.input.fileinputformat.split.maxsize`. If data locality could be met, this improves the data  
| read time. Map task spills the input data into Yarn's local directories when its buffer is filled up  
| according to Yarn's configuration (controlled by `mapreduce.task.io.sort.mb` and  
| `mapreduce.map.sort.spill.percent`). For spilled out data, map task needs to read them into memory,  
| merge them and write out again into Yarn's local directories as one file. This is controlled by  
| `mapreduce.task.io.sort.factor` and `mapreduce.map.combine.minspills`. Therefore, if a spill happens, we  
| need to write them, read them and write them again. It is 3x times of IO time. Therefore, we need to tune  
| Yarn's configuration to avoid spill.
- | After map task is done, its output is written onto the local directories. After that, if there are reduce tasks  
| up, reduce task will take HTTP requests to fetch these data into local directories of the node on which the  
| reduce task is running. This phase is called copy. If thread number of copy is too small, the performance  
| will be impacted. The thread number of copy is controlled by `mapreduce.reduce.shuffle.parallelcopies`  
| and `mapreduce.tasktracker.http.threads`.
- | At the merge phase in reduce task, it keeps fetching data from different map task output into its local  
| directories. If these data fill up reduce task's buffer, this data will be written into local directories (this is  
| controlled by `mapreduce.reduce.shuffle.input.buffer.percent` and  
| `mapreduce.reduce.shuffle.merge.percent`). Also, if the spilled data file number on local disks exceed the  
| `mapreduce.reduce.merge.inmem.threshold`, reduce task will also merge these files into larger ones. After

| all map data are fetched, reduce task will enter the sort phase which merges spilled files maintaining their sort order. This is done in rounds and the file number merged per round is controlled by `mapreduce.task.io.sort.factor`. Finally, one reduce task will generate one file in distributed file system.

## | **Hive**

| Apache Hive is designed for traditional data warehousing tasks and it is not for OLTP workloads.

| Apache Hive comprises of several components illustrated in Figure 19:



| *Figure 19. Framework of Hive (from Apache Hive website)*

| In Hadoop distro, such as IBM BigInsights, Hive Metastore server provides metadata related operations (such as database, table, partition). WebHCat server provides web interface for clients to execute queries against Hive. HiveServer2 server is a server interface that enables remote clients to execute queries against Hive and retrieve the results.

---

## | **Hadoop Performance Planning over IBM Spectrum Scale**

### | **Storage model**

#### | **Internal disks (FPO)**

| Similar to Hadoop HDFS, IBM Spectrum Scale FPO takes the sharing nothing framework. Every server in the cluster is a computing node and also a storage node. IBM Spectrum Scale manages the local disks from each server and unifies them as one distributed file system for applications running over these nodes.

| Figure 20 on page 113 illustrates a typical deployment:

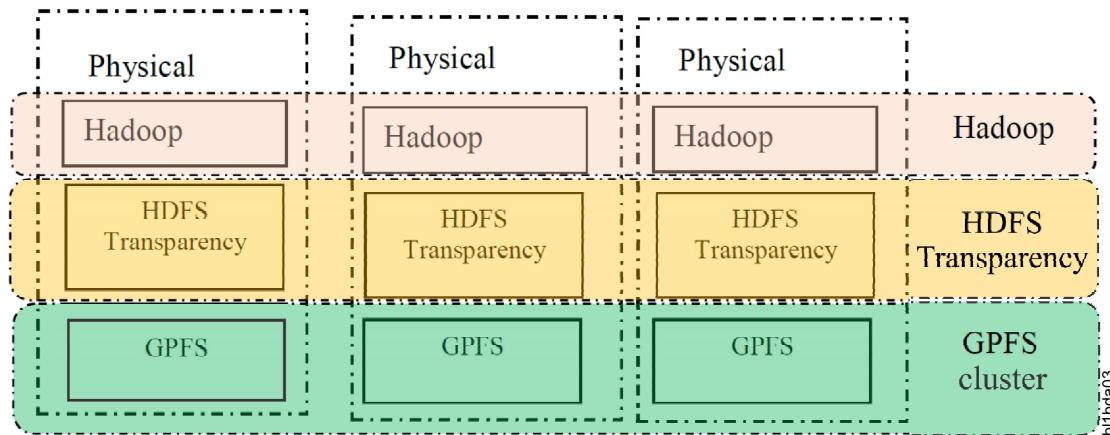


Figure 20. Hadoop over Spectrum Scale FPO

If you are deploying Hadoop over IBM Spectrum Scale FPO, execute the following steps for better performance cluster:

1. IBM Spectrum Scale FPO.

Refer to the *Spectrum Scale Shared Nothing Cluster Performance Tuning Guide* in the IBM developerworks wiki.

Some important configurations, such as, data replica, meta data replica, disk layout, failure group definition must be decided in this step. Also, you must tune the cluster configuration accordingly.

You need to consider the shuffle directory for intermediate data.

**Note:**

- Because of limited network bandwidth, do not run IBM Spectrum Scale FPO over 1Gb ethernet adapter in the production environment.
- 8~12 SAS or SATA disks are recommended disks per node with one 10Gb ethernet adapter. For meta data, SSD is recommended, especially for file/directory operation sensitive workloads.

2. HDFS Transparency nodes.

All FPO nodes must be installed with HDFS Transparency. Also, you should avoid introducing Transparency nodes and other Hadoop services (especially for Yarn's NodeManager, Resource Manager, Hive's Hiveserver2, HBase Master and Region Servers) over IBM Spectrum Scale clients because this will make all the data for these servers go over network. Therefore, degrading the performance.

Assign HDFS Transparency NameNode over IBM Spectrum Scale node with meta disks.

3. Server nodes for Yarn, HBase, Hive and Spark

Assign Yarn ResourceManager over the node running with HDFS Transparency. Assign HBase Master and HDFS Transparency NameNode on different nodes because of memory allocation.

HBase Master needs to be allocated with large memory for better performance. If you have many files, then assign the Transparency Namenode with more memory. If the node running Transparency Namenode does not have enough memory for HBase Master, assign them over different nodes.

4. Server nodes for other components.

Try for each node to have even I/O workloads and CPU %.

## ESS or shared storage

For Hadoop over shared storage (such as SAN storage), use the deployment in Figure 9 on page 8.

In this mode, Transparency reads the data from IBM Spectrum Scale NSD servers and sends them directly through RPC to Hadoop nodes. All data reading/writing on Transparency service nodes goes directly

- into file system through local disk paths (usually, it is Fiber Channel connections) and you get a better performance. In this mode, the data read flow is: **FC/local disks > NSD server > Transparency > network > Hadoop client**.
- Note:** In this mode, we do not deploy any Hadoop component over IBM Spectrum Scale NSD servers.
- If you cannot install Transparency over IBM Spectrum Scale NSD servers (for example, ESS), you must install Transparency over IBM Spectrum Scale clients. If so, the deployment in Figure 4 on page 4 is recommended.
- As compared to Figure 9 on page 8, the data read in Figure 4 on page 4 will be: **FC channel/disks > NSD server > network > Transparency > lo adapter > Hadoop client**.

## Storage model Consideration

This topic lists the advantages and disadvantages of different storage models.

*Table 8. Sharing nothing -vs- Shared storage*

Storage models	Advantages	Disadvantages
Sharing Nothing model	<ul style="list-style-type: none"> <li>Scaling computing and storage together by adding more nodes</li> <li>High I/O bandwidth</li> <li>Data locality for fast read</li> </ul>	<ul style="list-style-type: none"> <li>Additional network bandwidth from data protection against disk failure or node failure</li> <li>Low storage efficiency from 3 replica(~33% storage utility)</li> </ul>
Shared storage model	<ul style="list-style-type: none"> <li>No additional network bandwidth from data protection against disk/node failure</li> <li>1 replica, high data efficiency</li> <li>Scaling computing and storage separately/flexibly</li> </ul>	<ul style="list-style-type: none"> <li>No data locality</li> <li>Limited IO bandwidth</li> </ul>

- Consider the following factors while selecting a storage model:
  - If your data is larger than 1 PB, select shared storage or ESS. If you take sharing nothing framework, scanning such a huge data will take up a long time when protecting data against disk failure or node failure.
  - If the IBM Spectrum Scale cluster size is greater than 100 nodes, select shared storage or ESS. More nodes, the high possibility of node being down and more data stripe for protection.
  - If your data size is less than 500 TB but will be scaled to 1 PB+ in short term, select IBM Spectrum Scale sharing nothing model.
  - If your data size is between 500 TB and 1 PB, refer to Table 8 and your workloads for decision making.

## Hardware configuration planning

This topic describes the hardware configuration to be used for FPO model and IBM Spectrum Scale client node in shared storage model.

*Table 9. Hardware configuration for FPO model*

Configuration	Recommended configuration
Network	At least one 10Gb+ ethernet adapter or IB adapters
Disk	SAS disks, 8~12 disks per node
Memory	100+ GB per node
CPU	8+ logic processors

| Note: For X86\_64, with hyper thread on (by default), one physical core is mapped to two logic processors. For ppc64le, SMT 4 is recommended and one physical core is mapped to four logic processors.

| *Table 10. Hardware configuration for IBM Spectrum Scale client node in shared storage model*

Configuration	Recommended configuration
Network	At least one 10 Gb+ ethernet adapter or IB adapters
Disk	SAS disks, ~ 6 disks for shuffle if possible
Memory	100+ GB per node
CPU	8+ logic processors

| Note: It is better to take the same hardware configurations for all Hadoop nodes.

---

## | Performance guide

### | How to change the configuration for tuning

| This topic lists the steps to change the configuration for tuning.

| Refer to the following table to find the correct configuration directory and take the corresponding steps to update the configuration for tuning:

	Configuration Location	How to change the configuration
HortonWorks HDP	/etc/hadoop/conf	<ul style="list-style-type: none"> <li>• Change the configuration from Ambari for HDFS, Yarn, MapReduce2, Hive, etc.</li> <li>• Restart the services to sync the configuration into /etc/Hadoop/conf on each Hadoop node.</li> </ul>
Community Apache Hadoop	\$HADOOP_HOME/etc/hadoop	<ul style="list-style-type: none"> <li>• Modify the configuration xml files directly.</li> <li>• Take scp to sync the configurations into all other nodes.</li> </ul>
Standalone Spark Distro	Refer the guide from your Spark distro	Usually, if taking HDFS schema as data access in Spark, the <code>hdfs-site.xml</code> and <code>core-site.xml</code> are defined by <code>HADOOP_CONF_DIR</code> in <code>\$SPARK_HOME/spark-env.sh</code> . If so, you need to modify <code>hdfs-site.xml</code> and <code>core-site.xml</code> accordingly for tuning and sync the changes to all other Spark nodes.

	Configuration Location	How to change the configuration
HDFS Transparency	/usr/lpp/mmfs/hadoop/etc/hadoop (for HDFS Transparency 2.7.x) or /var/mmfs/hadoop/etc/hadoop (for HDFS Transparency 3.0.x)	<ul style="list-style-type: none"> <li>hdfs-site.xml and core-site.xml should be the same as the configuration for Hadoop or Spark.</li> <li>If you take HortonWorks HDP, modify the configuration for gpfs-site.xml from Ambari/IBM Spectrum Scale and restart the HDFS service to sync the changes to all HDFS Transparency nodes.</li> <li>If you take community Apache Hadoop, manually update gpfs-site.xml on one HDFS Transparency node and then run mmhadoopctl connector syncconf /usr/lpp/mmfs/hadoop/etc/hadoop (for HDFS Transparency 2.7.x) or mmhadoopctl connector syncconf /var/mmfs/hadoop/etc/hadoop (for HDFS Transparency 3.0.x) to sync the changes to all HDFS Transparency nodes.</li> </ul>

## System tuning

### Tuning over IBM Spectrum Scale FPO

To tune IBM Spectrum Scale FPO, refer the IBM Spectrum Scale Shared Nothing Cluster Performance Tuning Guide.

### Tuning over ESS/Shared Storage

If the customers deploy ESS, it is tuned after it is setup by the IBM team.

### Memory tuning

This topic describes the memory tuning.

Refer the following table to plan your system memory per node:

*Table 11. System Memory Allocation*

Total Memory per Node	Recommended Reserved System Memory	Recommended Reserved HBase Memory	Spectrum Scale Pagepool	Transparency NameNode Heap Size	Transparency DataNode Heap Size
16 GB	2 GB	2 GB	4GB	2GB+	2GB
24 GB	4 GB	4 GB	6GB	2GB+	2GB
48 GB	6 GB	8 GB	12GB	2GB+	2GB
64 GB	8 GB	8 GB	16GB	2GB+	2GB
72 GB	8 GB	8 GB	18GB	2GB+	2GB
96 GB	12 GB	16 GB	20GB	2GB+	2GB
128 GB	20 GB	24 GB	20GB	2GB+	2GB
256 GB	32 GB	32 GB	20GB	2GB+	2GB
512 GB	64 GB	64 GB	20GB	2GB+	2GB

HDFS Transparency DataNode service is a light weight daemon and does not need a lot of memory.

- | For HDFS Transparency NameNode, when ranger support is enabled (by default, it's enabled and you could turn it off by setting `gpfs.ranger.enabled=false` in `gpfs-site.xml`), Transparency NameNode will cache inode information. Therefore, if your Transparency NameNode heap size is very small, JVM garbage collection is executed frequently. Usually, it is 2GB and you could increase this up to 4GB if you have a large set of files in your file system.
- | **Note:** Transparency NameNode does not manage FSImage as native HDFS does. It does not need large memory for large number of files as native HDFS.
- | The pagepool (memory cache for IBM Spectrum Scale) size for IBM Spectrum Scale is recommended for most cases in production. However, if you mainly run HBase and want HBase of the best performance, follow section 4.6 Tuning HBase/YCSB.

| *Table 12. How to change the memory size*

<b>Configuration</b>	<b>Guide</b>
IBM Spectrum Scale pagepool size	<pre>mmchconfig pagepool=XG -N &lt;node1,node2...&gt;</pre> <p>Need to restart IBM Spectrum Scale daemon to make the change effective.</p>
Transparency NameNode Heap Size	<p><b>Ambari GUI &gt; HDFS &gt; Configs</b>, change the value of <b>NameNode Java heap size</b>.</p> <p>If you take community Hadoop, modify the variable <b>HADOOP_NAMENODE_OPTS</b> in <code>\$HADOOP_HOME_DIR/etc/hadoop/hadoop-env.sh</code>. For example, add <code>-Xms2048m -Xmx2048m</code> to set the heap size as 2GB. If the option <b>-Xms</b> and <b>-Xmx</b> have been there, you can modify the number (For example, 2048 for the example) directly.</p> <p>Need to restart HDFS Transparency to make the change effective.</p>
Transparency DataNode Heap size	<p><b>Ambari GUI &gt; HDFS &gt; Configs</b>, change the value of <b>DataNode maximum Java heap size</b>.</p> <p>If you take community Hadoop, modify the variable <b>HADOOP_DATANODE_OPTS</b> in <code>\$HADOOP_HOME_DIR/etc/hadoop/hadoop-env.sh</code>. For example, add <code>-Xms2048m -Xmx2048m</code> to set the heap size as 2GB. If the option <b>-Xms</b> and <b>-Xmx</b> have been there, you could modify the number (For example, 2048 for the example) directly.</p> <p>Need to restart HDFS Transparency to make the change effective.</p>

## | **HDFS Transparency Tuning**

### | **Tuning for IBM Spectrum Scale FPO**

- | If you deploy the Hadoop cluster through Ambari (for both HortonWorks and IBM BigInsights IOP), Ambari will do some default tuning according to your cluster and you do not need to do special tuning for HDFS Transparency.

- | The following table lists the most important configurations that we need to check for running HDFS Transparency over IBM Spectrum Scale FPO:

| Table 13. Tuning configurations for Transparency over IBM Spectrum Scale FPO

Configurations	Default value	Recommended Value	Comments
<b>dfs.replication</b>	3	Keep consistent with your file system default data replica	Usually, it will be 3.
<b>dfs.blocksize</b>	134217728	Chunk size	It should be blockGroupFactor * blocksize. Usually, it is recommended as 128 * 2MB or 256 * 2MB.
<b>io.file.buffer.size</b>	4096	Data block size of your file system	It should be the value of your data storage pool. Check this by <b>mm1spool &lt;your-file-system&gt; all -L</b> .
<b>dfs.datanode.handler.count</b>	10	200	If you have more than 10 physical disks per node, you could increase this. For example, if you have 20 physical disks per node, increase it to 400.
<b>dfs.namenode.handler.count</b>	10	Refer the comments	This depends on the resource of NameNode and the Hadoop node number. If taking IBM Spectrum Scale Ambari Integration, $100 * \log_e(\text{DataNode-Number})$ is used to calculate the value for <b>dfs.namenode.handler.count</b> .  If not taking IBM Spectrum Scale Ambari integration, you could take 400 for ~10 Hadoop nodes; take 800 or higher for ~20 Hadoop nodes.
<b>dfs.ls.limit</b>	1000	100000	
<b>dfs.client.read.shortcircuit.streams.cache.size</b>	4096	Refer the comments	Change it as the IBM Spectrum Scale file system data blocksize.
<b>dfs.datanode.transferTo.allowed</b>	true	false	If this is true, the IO will be 4K mmap () for gpfs.

| Note: If **dfs.datanode.handler.count** is very small, you might see socket time when HDFS Client connects to the datanode.

| If **nofile** and **noproc** from **ulimit** is less than 64K, you might see socket connection timeout. By default, **dfs.client.socket-timeout** is *60000 ms*. If your cluster is busy (for example, doing benchmark), you could configure this into *300000 ms* and configure **dfs.datanode.socket.write.timeout** into *600 seconds* (by default it is *480000ms*).

| The above tuning should also be done for Hadoop HDFS client. If you take HortonWorks HDP, change the above configuration on Ambari GUI. After these changes, restart all services and ensure that these changes are synced into */etc/hadoop/conf/hdfs-site.xml* and */usr/lpp/mmfs/hadoop/etc/hadoop/hdfs-site.xml* (for HDFS Transparency 2.7.x) or */var/mmfs/hadoop/etc/hadoop/hdfs-site.xml* (for HDFS Transparency 3.0.x). If you take open source Apache Hadoop, you need to update these configurations for

| Hadoop clients (`$HADOOP_HOME/etc/hadoop/hdfs-site.xml`) and take `/usr/lpp/mmfs/bin/mmhadoopctl` to sync your changes to HDFS Transparency configuration on all HDFS Transparency nodes.

## | **Tuning for Spectrum Scale over shared storage or ESS**

| If you deploy the Hadoop cluster through Ambari (for both HortonWorks and IBM BigInsights IOP), Ambari will do some default tuning according to your cluster.

| The following table lists the most important configurations that we need to check for running HDFS Transparency over IBM Spectrum Scale ESS or shared storage:

| *Table 14. Tuning configurations for Transparency over IBM ESS or shared storage*

Configurations	Default value	Recommended Value	Comments
<code>dfs.replication</code>	1	Keep consistent with your file system default data replica	Usually, it will be 1.
<code>dfs.blocksize</code>	134217728	536870912	
<code>io.file.buffer.size</code>	4096(bytes)	Data block size of your file system	It should be the value of your data storage pool. Check this by <code>mm1spool &lt;your-file-system&gt; all -L</code> . <b>Note:</b> The highest value should be less than 16777216 bytes (16MB). If ESS data block size is 16MB, configure this as 8388608 bytes (8MB).
<code>dfs.datanode.handler.count</code>	10	Refer the comments	Calculate this according to Hadoop node number and Transparency DataNode number: $(40 * \text{HadoopNodes}) / \text{TransparencyDataNodeNumber}$
<code>dfs.namenode.handler.count</code>	10	Refer the comments	This depends on the resource of NameNode and the Hadoop node number. If taking IBM Spectrum Scale Ambari Integration, $100 * \log_e(\text{DataNodeNumber})$ is used to calculate the value for <code>dfs.namenode.handler.count</code> .  If not taking IBM Spectrum Scale Ambari integration, you could take 400 for ~10 Hadoop nodes; take 800 or higher for ~20 Hadoop nodes.
<code>dfs.ls.limit</code>	1000	100000	
<code>dfs.client.read.shortcircuit.streams.cache.size</code>	4096	Refer the comments	Change it as the IBM Spectrum Scale file system data blocksize.
<code>dfs.datanode.transferTo.allowed</code>	true	false	If this is true, the IO will be 4K mmap() for gpfs.

The above tuning should also be done for Hadoop HDFS client. If you take HortonWorks HDP, change the above configuration on Ambari GUI. After these changes, you need to restart all services and ensure that these changes are synced into /etc/hadoop/conf/hdfs-site.xml and /usr/lpp/mmfs/hadoop/etc/hadoop/hdfs-site.xml (for HDFS Transparency 2.7.x) or /var/mmfs/hadoop/etc/hadoop/hdfs-site.xml (for HDFS Transparency 3.0.x). If you take open source Apache Hadoop, you need to update these configurations for Hadoop clients (\$HADOOP\_HOME/etc/hadoop/hdfs-site.xml) and take /usr/lpp/mmfs/bin/mmhadoopctl to sync your changes to HDFS Transparency configuration on all HDFS Transparency nodes.

## **Special tuning for IBM Spectrum Scale**

For better performance, if all the Hadoop nodes are in IBM Spectrum Scale cluster (as IBM Spectrum Scale client or NSD server or FPO nodes), you should enable Hadoop short circuit read from HDFS Transparency 2.7.3 as this will bring data read performance. From HDFS Transparency 2.7.3-1, short circuit write is enabled to improve the data write performance when short circuit read is enabled.

If you take Hadoop distro, enable short circuit write from Ambari/HDFS. If you take open source Apache or Spark standalone distro, see Chapter 2, “IBM Spectrum Scale support for Hadoop,” on page 9 to enable the short circuit read.

If you do not run Apache Ranger, disable Ranger support and see Chapter 2, “IBM Spectrum Scale support for Hadoop,” on page 9.

If you enable HA, and the IO stress from Hadoop cluster is heavy, configure new service RPC port to avoid unnecessary active NameNode switch. Change the following configurations in hdfs-site.xml:

*Table 15. Configurations in hdfs-site.xml*

HA mode	Configuration	Recommendation	Comment
HA or non HA	<b>dfs.namenode.service.handler.count</b>	80	
Non HA	<b>dfs.namenode.servicerpc-address</b>	<namenode>:8060	
NameNode HA	<b>dfs.namenode.servicerpc-address.&lt;yourcluster&gt;.nn1</b>	<namenode1>:8060	Change <yourcluster> into your HA cluster ID and <b>nn1</b> should be matched with <b>dfs.ha.namenodes.&lt;yourcluster&gt;</b> .
	<b>dfs.namenode.servicerpc-address.&lt;yourcluster&gt;.nn2</b>	<namenode2>:8060	Change <yourcluster> into your HA cluster ID and <b>nn2</b> should be matched with <b>dfs.ha.namenodes.&lt;yourcluster&gt;</b> .

## **Hadoop/Yarn Tuning**

### **Common Tuning**

Tuning for Yarn comprises of two parts: tuning MapReduce2 and tuning Yarn.

Follow the configurations in to tune MapReduce2 and Table 16 to tune Yarn.

*Table 16. Tuning MapReduce2*

Configurations	Comments
<b>mapreduce.map.memory.mb</b>	Default: 1024 MB Recommended: 2048 MB

| *Table 16. Tuning MapReduce2 (continued)*

<b>Configurations</b>	<b>Comments</b>
<b>mapreduce.reduce.memory.mb</b>	Default: 1024 MB Recommended: 4096 MB or larger The value could be considered according to the <b>yarn.nodemanager.resource.memory-mb</b> and the current task number on one node. For example, if you configure 100 GB for <b>yarn.nodemanager.resource.memory-mb</b> and you have
<b>mapreduce.map.java.opts</b>	Recommended: 75% * mapreduce.map.memory.mb or 80% * mapreduce.map.memory.mb
<b>mapreduce.reduce.java.opts</b>	Recommended: 75% * mapreduce.reduce.memory.mb or 80% * mapreduce.reduce.memory.mb.
<b>mapreduce.job.reduce.slowstart.completedmaps</b>	Default: 0.05 Different Yarn jobs could take different value for this configuration. You could specify this value when submitting Yarn job if your job wants to take different value for this.
<b>mapreduce.map.cpu.vcores</b>	1
<b>mapreduce.reduce.cpu.vcores</b>	1
<b>mapreduce.reduce.shuffle.parallelcopies</b>	Default: 5 Recommend: 30+
<b>mapreduce.tasktracker.http.threads</b>	Default: 40 If your cluster has more than 40 nodes, you could increase this to ensure that the reduce task on each host could have at least 1 thread for shuffle data copy.
<b>yarn.app.mapreduce.am.job.task.listener.thread-count</b>	Default: 30 If you have larger cluster for job (for example. your cluster is larger than 20 nodes and 16 logic processors per node) you could increase this to try.
<b>mapreduce.task.io.sort.mb</b>	Default: 100(MB) Recommended: 70% * mapreduce.map.java.opts
<b>mapreduce.map.sort.spill.percent</b>	Default: 80 Take default value and not change this.
<b>mapreduce.client.submit.file.replication</b>	Default: 10 Change it as the default replica of your IBM Spectrum Scale file system (check this by <code>mm1sfs &lt;your-fs-name&gt; -r</code> ).
<b>mapreduce.task.timeout</b>	Default: 300000ms Change it into 600000s if you are running benchmark.

| The following configurations are not used by Yarn and you do not need to change them:

- | `mapreduce.jobtracker.handler.count`
- | `mapreduce.cluster.local.dir`
- | `mapreduce.cluster.temp.dir`

## Tuning Yarn

This section describes the configurations to be followed for tuning Yarn.

*Table 17. Configurations for tuning Yarn*

Configurations	Default	Recommended value	Comments
Resource Manager Heap Size (resourcemanager_heapsize)	1024	1024	
NodeManager Heap Size (nodemanager_heapsize)	1024	1024	
<b>yarn.nodemanager.resource.memory-mb</b>	8192	Refer comments	The total memory that could be allocated for Yarn jobs.
<b>yarn.scheduler.minimum-allocation-mb</b>	1024	Refer comments	This value should not be greater than <b>mapreduce.map.memory.mb</b> and <b>mapreduce.reduce.memory.mb</b> . And, <b>mapreduce.map.memory.mb</b> and <b>mapreduce.reduce.memory.mb</b> must be the multiple times of this value. For example, if this value is 1024MB, then, you cannot configure <b>mapreduce.map.memory.mb</b> as 1536 MB.
<b>yarn.scheduler.maximum-allocation-mb</b>	8192	Refer comments	This value should not be smaller than <b>mapreduce.map.memory.mb</b> and <b>mapreduce.reduce.memory.mb</b> .
<b>yarn.nodemanager.local-dirs</b>	<code> \${hadoop.tmp.dir}/nm-local-dir</code>	Refer comments	<code>hadoop.tmp.dir</code> is <code>/tmp/hadoop-\$user.name</code> by default. This will impact the shuffle performance. If you have multiple disks/partitions for shuffle, you could configure this as: <code>/hadoop/local/sd1</code> , <code>/hadoop/local/sd2</code> , <code>/hadoop/local/sd3</code> . If you have more disks for this configuration, you will have more IO bandwidth for Yarn's intermediate data.
<b>yarn.nodemanager.log-dirs</b>	<code> \${yarn.log.dir}/userlogs</code>	Refer comments.	These directories are for Yarn job to write job task logs. It does not need a lot of bandwidth and therefore you can configure one directory for this configuration. For example, <code>/Hadoop/local/sd1/logs</code> .

| *Table 17. Configurations for tuning Yarn (continued)*

Configurations	Default	Recommended value	Comments
<b>yarn.nodemanager.resource.cpu-vcores</b>	8	Logic processor number	Configure this as the logic processor number. You could check the logic processor number according to /proc/cpuinfo. This is the common rule. However, if you run the job which takes all vcores of all nodes and the CPU% is not higher than 80%, you could increase this (for example, 1.5 X logic-processor-number).  If you will run CPU sensitive workloads, keep the ratio of physical_cpu/vcores as 1:1. If you will run IO bound workloads, you could change this as 1:4. If you do not know your workloads, keep this as 1:1.
<b>yarn.scheduler.minimum-allocation-vcores</b>	1	1	
<b>yarn.scheduler.maximum-allocation-vcores</b>	32	1	
<b>yarn.app.mapreduce.am.resource.mb</b>	1536	Refer the comments	Configure this as the value for <b>yarn.scheduler.minimum-allocation-mb</b> . Usually, 1GB or 2GB is enough for this. <b>Note:</b> This is under MapReduce2.
<b>yarn.app.mapreduce.am.resource.cpu-vcores</b>	1	1	
<b>Compression</b>			
<b>mapreduce.map.output.compress</b>	false	true	Make the output of Map tasks compressed. Usually, this means to compress the Yarn's intermediate data.
<b>mapreduce.map.output.compress.codec</b>	<b>org.apache.hadoop.io.compress.DefaultCodec</b>	<b>org.apache.hadoop.io.compress.SnappyCodec</b>	
<b>mapreduce.output.fileoutputformat.compress</b>	false	true	Make the job output compressed.
<b>mapreduce.output.fileoutputformat.compress.codec</b>	<b>org.apache.hadoop.io.compress.DefaultCodec</b>	<b>org.apache.hadoop.io.compress.GzipCodec</b>	
<b>mapreduce.output.fileoutputformat.compress.type</b>	RECORD	BLOCK	Specify the
<b>Scheduling</b>			

| *Table 17. Configurations for tuning Yarn (continued)*

Configurations	Default	Recommended value	Comments
<code>yarn.scheduler.capacity.resource-calculator</code>	<code>org.apache.hadoop.yarn.util.resource.DefaultResourceCalculator</code>	Refer the comments	<p>By default, it is <code>org.apache.hadoop.yarn.util.resource</code>. <code>DefaultResourceCalculator</code> which only schedules the jobs according to memory calculation. If you want to schedule the job according to memory and CPU, you could enable CPU scheduling from <b>Ambari &gt; Yarn &gt; Config</b>. After you enable CPU scheduling, the value will be <code>org.apache.hadoop.yarn.util.resource</code>. <code>DominantResourceCalculator</code>.</p> <p>If you find that your cluster node has 100% CPU% after taking default configuration, you could try to limit the concurrent tasks by enabling CPU scheduling. If not, no need to change this.</p>

## Specific Tuning of Yarn for ESS/Shared Storage

This section describes the Yarn configurations to be tuned for ESS/Shared storage.

If you are running Hadoop over shared storage or ESS, the following must be tuned:

Configurations	default	Recommended value	Comments
<code>yarn.scheduler.capacity.node-locality-delay</code>	40	-1	<p>Change this as -1 to disable the locality-based scheduling. If not changing this, you will only have 40 concurrent tasks running over the cluster no matter the number of nodes in your cluster.</p> <p>Change this from <b>Ambari GUI &gt; Yarn &gt; Config &gt; Advanced &gt; Scheduler; Capacity Scheduler</b>.</p>

## Maximal Map and Reduce Task Calculation

This topic describes the calculations for Maximal Map and Reduce Task.

Value	Calculation
<code>MaxMapTaskPerWave_mem</code>	$(yarn.nodemanager.resource.memory-mb * YarnNodeManagerNumber - yarn.app.mapreduce.am.resource.mb) / mapreduce.map.memory.mb$

Value	Calculation
<b>MaxMapTaskPerWave_vcore</b>	(yarn.nodemanager.resource.cpu-vcores * YarnNodeManagerNumber - yarn.app.mapreduce.am.resource.cpu)/ yarn.scheduler.minimum-allocation-vcores
<b>TotalMapTaskPerWave</b>	Equal to MaxMapTaskPerWave_mem by default
<b>MaxReduceTaskPerNode_mem</b>	(yarn.nodemanager.resource.memory * YarnNodeManagerNumber - yarn.app.mapreduce.am.resource.mb)/ mapreduce.reduce.memory.mb
<b>MaxReduceTaskPerNode_vcore</b>	(yarn.nodemanager.resource.cpu-vcores * YarnNodeManagerNumber - yarn.app.mapreduce.am.resource.cpu)/ yarn.scheduler.minimum-allocation-vcores
<b>TotalReduceTaskPerWave</b>	Equal to MaxReduceTaskPerNode_mem by default

If `yarn.scheduler.capacity.resource-calculator` is not changed, by default, `MaxMapTaskPerWave_mem` will take effective. Under this situation, if the `MaxMapTaskPerWave_vcore` is more than 2 times of `MaxMapTaskPerWave_mem`, you still have a lot of CPU resource and you could increase `MaxMapTaskPerWave_mem` by either increasing `yarn.nodemanager.resource.memory-mb` or decreasing `mapreduce.map.memory.mb`. However, if `MaxMapTaskPerWave_vcore` is smaller than `MaxMapTaskPerWave_mem`, it means more than 1 task will run on the same logic processor and this might bring additional context switch cost.

It is similar for `MaxReduceTaskPerNode_mem` and `MaxReduceTaskPerNode_vcore`.

`TotalMapTaskPerWave` is the totally concurrent task number that you could run in one wave.  
`TotalReduceTaskPerWave` is the totally concurrent task number that you could run in one wave.

## Performance sizing

A lot of factors, such as logic processor number, memory size, network bandwidth, storage bandwidth and the IBM Spectrum Scale deployment mode, can impact performance sizing. This section gives a brief throughput estimate sizing guide for teragen and terasort workloads as the query and transaction workload types (HBase, Hive) have too many factors to be able to give sizing rules.

Sizing the throughput from HDFS Transparency cluster could be done by two steps:

1. Sizing the throughput from IBM Spectrum Scale POSIX interface.
2. Calculating the throughput from HDFS Transparency.

To size the throughput of IBM Spectrum Scale POSIX interface, use the open source IOR benchmark to get the throughput of reads and writes from the POSIX interface. If you are not able to use IOR benchmark, estimate the throughput for IBM Spectrum Scale POSIX interface as follows:

- For IBM ESS, get the throughput number from the IBM product guide.
- For IBM Spectrum Scale FPO:
  - If the network bandwidth is greater than (Disk-number-per-node \* disk-bandwidth), calculate using:  

$$((Disk-number * Disk-bandwidth / Replica-number) * 0.7)$$
  - If network bandwidth is smaller than (Disk-number-per-node \* disk-bandwidth), calculate using:  

$$((Network-bandwidth-per-node * node-number) * 0.7)$$

Usually, it is recommended to take SSD for metadata so that the metadata operations in IBM Spectrum Scale FPO do not become the bottleneck. Under this condition, HDFS Transparency interface will yield

- | approximately 70% to 80% throughput based on the POSIX interface throughput value. The benchmark throughput is impacted by the number of Hadoop nodes and the Hadoop-level configuration settings.
- | As for how to size Hadoop node number,
  - | • For ESS:
    - | Calculate Hadoop node number by using the ESS official throughput value and the client network bandwidth:
      - | For example, if using ESS GL4s and the throughput is 36GB from IBM product guide, and the client has 10Gb network bandwidth, then will need  $36\text{GB}/((10\text{Gb}/8) * 0.8)$  clients to drive the throughput.
      - | OR
      - | Calculate based only on the network bandwidth from the ESS configuration and the client network adapter throughput:
        - | For example, ESS configuration network bandwidth (e.g. 100GB) and the client network adapter throughput (e.g. 10GB), then will need  $(100\text{GB}/10\text{GB}) = 10$  clients.
  - | • For FPO:
    - | All Hadoop nodes should be Spectrum Scale FPO nodes.

## | **Teragen**

- | When benchmarking Teragen, execute the following command:
 

```
| hadoop jar /usr/hdp/current/hadoop-mapreduce-client/hadoop-mapreduce-examples.jar
| teragen -Dmapreduce.job.maps=<JOB_MAP_NUMBER> -Ddfs.blocksize=<BLOCKSIZE>
| <DATA_RECORDS> /<OUTPUT_PATH>
```
- | In the above command, **<DATA\_RECORDS>** is used to specify the number of records in your evaluation. One record is 100 bytes. So, 1000 000 000 0 records mean the data size  $1000\ 000\ 000\ 0 * 100$  bytes, which is closed to 1TB.
- | **<OUTPUT\_PATH>** is the data output directory. Change this accordingly.
- | Teragen has no Reduce task and the value **<JOB\_MAP\_NUMBER>** is the one you need to plan with some efforts. Refer the **yarn.nodemanager.resource.memory-mb**, **yarn.scheduler.minimum-allocation-vcores**, **yarn.app.mapreduce.am.resource.cpu-vcores**, **yarn.app.mapreduce.am.resource.mb** in Table 17 on page 122.
- | The **<JOB\_MAP\_NUMBER>** should be equal to  $(\text{MaxTaskPerNode\_mem} * \text{YarnNodeManagerNumber} - 1)$ . Also, the value  $((\text{DATA\_RECORDS} * 100) / (1024 * 1024)) / <\text{JOB\_MAP\_NUMBER}>$  should not be very small and it should be close to **dfs.blocksize** or multiple times of **dfs.blocksize**. If the value  $((\text{DATA\_RECORDS} * 100) / (1024 * 1024)) / <\text{JOB\_MAP\_NUMBER}>$  is very small, your **<DATA\_RECORDS>** is very small for your cluster.
- | If **yarn.scheduler.capacity.resource-calculator** is changed by enabling CPU scheduling from Ambari, the smaller value between **MaxTaskPerNode\_mem** and **MaxTaskPerNode\_vcore** will be effective. Under this situation, you could try to make **MaxTaskPerNode\_vcore** and **MaxTaskPerNode\_mem** close to each other. If not, that usually indicates you have free resources that are not yet utilized.
- | If you take the same block size (**dfs.blocksize** from **core-site.xml**) for your TeraGen job, you do not need to specify **-Ddfs.blocksize=<BLOCKSIZE>**. If you want to take different **dfs.blocksize** for the job, you could specify the **-Ddfs.blocksize=<BLOCKSIZE>** option.

## TeraSort

- | TeraSort is a typical Map/Reduce job. It will take map tasks and reduce tasks.
- | Take the following command to run TeraSort:

```
| hadoop jar /usr/hdp/current/hadoop-mapreduce-client/hadoop-mapreduce-examples.jar terasort \
| -Dmapreduce.job.reduces=<REDUCE_TASKS> \
| -Ddfs.blocksize=<DFS_BLOCKSIZE> \
| -Dmapreduce.input.fileinputformat.split.minsize=<DFS_BLOCKSIZE> \
| -Dio.file.buffer.size=<IO_BUFFER_SIZE> \
| -Dmapreduce.map.sort.spill.percent=0.8 \
| -Dmapreduce.reduce.shuffle.merge.percent=0.96 \
| -Dmapreduce.reduce.shuffle.input.buffer.percent=0.7 \
| -Dmapreduce.reduce.input.buffer.percent=0.96 \
| /<TERAGEN_DATA_INPUT> \
| /<TERASORT_DATA_OUTPUT>
```
- | <IO\_BUFFER\_SIZE> must be equal to your IBM Spectrum Scale data pool block size (check this by `mm1spool <fs-name> all -L`).
- | <TERAGEN\_DATA\_INPUT> and <TERASORT\_DATA\_OUTPUT> must be specified according to your requirements.
- | <DFS\_BLOCKSIZE> could be the default `dfs.blocksize` from `hdfs-site.xml`. If you want to take different block size, specify it here. For IBM Spectrum Scale ESS or shared storage, <DFS\_BLOCKSIZE> cannot be equal to the data pool block size (usually, 1GB block size can give you a good performance). For IBM Spectrum Scale FPO, the <DFS\_BLOCKSIZE> must be equal to your file system data blocksize \* `blockGroupFactor` (check these two values from `mm1spool <fs-name> all -L`).
- | <DFS\_BLOCKSIZE> impacts the map task number in your cluster. For the above command (`mapreduce.input.fileinputformat.split.minsize` is specified as <DFS\_BLOCKSIZE>), the final map task number is calculated according to <DFS\_BLOCKSIZE>. If you have only one file with size 512MB and you specify 500MB as <DFS\_BLOCKSIZE>, you will have two splits or map tasks. If you have two 512MB files and you specify 500MB as <DFS\_BLOCKSIZE>, you will get four splits or map tasks.
- | **Note:** The total split number is the final map task number and cannot be changed by options from TeraSort. If you do not follow the above guide, you might get incorrect split number and therefore impact the map phase in Terasort.
- | The ideal case for each map task, is that the to-be-processed data size is close but not larger than (70% \* `mapreduce.map.java.opts` \* 80%) and this could keep the intermediate data size as small as possible in the shuffle of job.
- | Very small <DFS\_BLOCKSIZE> makes you have more map tasks. Map task number should be proper for the cluster to execute them in one wave or two waves. You should not execute map tasks in three or more waves because this will slow down the performance.
- | If the map task number is ((`MaxTaskPerNode_mem` \* `YarnNodeManagerNumber`) - 1), all these map tasks can be handled in one wave. If the map task number is larger than ((`MaxTaskPerNode_mem` \* `YarnNodeManagerNumber`) - 1), it should be between 1.75 \* (`MaxTaskPerNode_mem` \* `YarnNodeManagerNumber`) and 1.9 \* (`MaxTaskPerNode_mem` \* `YarnNodeManagerNumber`). You could try different map task number by changing file number from <TERAGEN\_DATA\_INPUT> and <DFS\_BLOCKSIZE>.
- | Usually, <REDUCE\_TASKS> should be executed in one wave. That means, <REDUCE\_TASKS> should be equal to (`TotalReduceTaskPerWave` - 1).

## | DFSIO

- | DFSIO is shipped with Hadoop distro.
  - | The following are the options for DFSIO:

```
$yarn jar /usr/hdp/current/hadoop-mapreduce-client/hadoop-mapreduce-client-jobclient.jar TestDFSIO
Usage: TestDFSIO [genericOptions] -read [-random | -backward | -skip [-skipSize Size]] | -write | -append | -truncate | -clean [-compression codecClassName] [-nrFiles N] [-size Size[B|KB|MB|GB|TB]] [-resFile resultFileName] [-bufferSize Bytes] [-rootDir]
```
  - | Usually, we care only about the -read, -write, -nrFiles, -size and -bufferSize options.
  - | The option -read is used to evaluate the read performance. The option -write is used to evaluate the write performance. The option -nrFiles is used to specify the number of files you want to generate. The option -size is used to specify the file size for each file.
  - | So, the total data that TestDFSIO will read/write is (nrFiles \* size). DFSIO is simple in logic and it will start the nrFiles map tasks running over the whole Hadoop/Yarn cluster.
  - | 1st tuning guide: nrFiles and task number
    - | While evaluating TestDFSIO, we need to consider the Yarn's configuration. If the maximum tasks per wave is **TotalMapTaskPerWave**, your nrFiles should be **TotalMapTaskPerWave**.
    - | If it is IBM Spectrum Scale FPO, the file size -size should be at least 512MB or more (you could try 1GB, 2GB and 4GB). If it is shared storage or ESS, the file size -size should be 1GB or more (try 1GB, 2GB and 4GB).
    - | Usually, according to experience, we take as many map tasks as possible for DFSIO read. For DFS write, we recommend to try the map tasks according to logic processors even if you have more free memory.
  - | 2nd tuning guide: nrFiles \* size
    - | The total data size (nrFiles \* size) should be at least 4 times of the total physical memory size of all HDFS nodes. For example, if you want to compare the performance data of DFSIO between native HDFS and IBM Spectrum Scale, if you have 10 nodes for native HDFS DataNode (100GB physical memory size per DataNode), then your (nrFiles \* size) over native HDFS should be 4 \* (100GB per node \* 10 DataNodes), ~ 4000GB. And then, the same (nrFiles \* size) for IBM Spectrum Scale.
  - | 3rd tuning guide: -bufferSize
    - | Try -bufferSize with the block size of IBM Spectrum Scale. This is the IO buffer size for task to write/read data.
- ## | TPC-H and TPC-DS for Hive
- ### | Tuning for Hive
- | Hive is Hadoop's SQL interface over HDFS. Therefore, the tuning is very similar for Hive as native HDFS.
  - | For more information, refer Hive's Manual Guide.
  - | The following tunings will be related with IO and therefore impacted by different distributed file system:

| Table 18. Hive's Tuning

Configuration	Default Value	Recommended Value	Comments
<b>hive.exec.compress.output</b>	False	True	Compress the Hive's execution results before writing it out.
<b>hive.exec.compress.intermediate</b>	False	True	Compress the Hive's intermediate data before writing it out.
<b>hive.intermediate.compression.codec</b>	N/A	<b>org.apache.hadoop.io.compress.SnappyCodec</b>	
<b>hive.intermediate.compression.type</b>	N/A	BLOCK	
<b>hive.exec.reducer.max</b>	1009	Variable	This should be decided according to your cluster and Yarn's configuration.

| If the Hive's job invokes a lot of Map/Reduce and generates a lot of intermediate data or output data, configuring the above will improve the Hive's job execution.

| If you configure **hive.exec.compress.output** as *true*, you need to check the following configuration in Hadoop Yarn:

- | • **mapreduce.output.fileoutputformat.compress=true**
- | • **mapreduce.output.fileoutputformat.compress.codec=org.apache.hadoop.io.compress.SnappyCodec**
- | • **mapreduce.output.fileoutputformat.compress.type=BLOCK**

| You can modify `hive-site.xml` and restart Hive service to make it effective for all the Hive jobs. Else, you can set them in Hive console and they will only be effective for the commands/jobs invoked from the Hive console:

```
| #hive
| hive> set hive.exec.compress.output=true;
| hive> set mapreduce.output.fileoutputformat.compress=true;
| hive> set mapreduce.output.fileoutputformat.compress.codec=org.apache.hadoop.io.compress.SnappyCodec;
| hive> set mapreduce.output.fileoutputformat.compress.type=BLOCK;
| hive>
```

## | Running TPC-H/Hive

| This section lists the steps for running TPC-H for Hive.

| Execute the following steps to run TPC-H for Hive:

- | 1. Download TPC-H from the official website and `TPC-H_on_Hive` from Running TPC-H queries on Hive.
- |     Download `TPC-H_on_Hive_2009-08-14.tar.gz`.
- | 2. Untar the above `TPC-H_on_Hive_2009-08-14.tar.gz` into `$TPC_H_HIVE_HOME`
- | 3. Download DBGEN from the TPC-H website:

```

| Note: Register your information and download the TPC-H_Tools_v<version>.zip.
| 4. Untar TPC-H_Tools_v<version>.zip and build dbgen:
| #unzip TPC-H_Tools_v<version>.zip
| Assuming it is located under $TPCH_DBGEN_HOME
| #cd $TPCH_DBGEN_HOME
| #cd dbgen
| #cp makefile.suite makefile
| Update the following values in $TPCH_DBGEN_HOME/dbgen/makefile accordingly:
| #vim makefile
|
| CC = gcc
| DATABASE = SQLSERVER
| MACHINE=LINUX
| WORKLOAD = TPCH
|
| Modify the $TPCH_DBGEN_HOME/dbgen/tpcd.h:
| vim $TPCH_DBGEN_HOME/dbgen/tpcd.h
|
| #ifdef SQLSERVER
| #define GEN_QUERY_PLAN "EXPLAIN;"
| #define START_TRAN "START TRANSACTION;\n"
| #define END_TRAN "COMMIT;\n"
| #define SET_OUTPUT ""
| #define SET_ROWCOUNT "limit %d;\n"
| #define SET_DBASE "use %s;\n"
| #endif
|
| Execute make to build dbgen:
| #cd $TPCH_DBGEN_HOME/dbgen/
| #make
| ...
| gcc -g -DDBNAME=\"dss\" -DLINUX -DSQLSERVER -DTPCH -DRNG_TEST
| -D_FILE_OFFSET_BITS=64 -O -o qgen build.o bm_utils.o qgen.o
| rnd.o varsub.o text.o bcd2.o permute.o speed_seed.o rng64.o -lm
|
| 5. Generate the transaction data:
| #cd $TPCH_DBGEN_HOME/dbgen/
| #./dbgen -s 500 △this is to generate 500GB data, change the value
| accordingly for your benchmark
|
| The generated data is stored under $TPCH_DBGEN_HOME/dbgen/ and all files are named with the suffix
| .tbl:
| # ls -la *.tbl
| -rw-r--r-- 1 root root 24346144 Jul 5 08:41 customer.tbl
| -rw-r--r-- 1 root root 759863287 Jul 5 08:41 lineitem.tbl
| -rw-r--r-- 1 root root 2224 Jul 5 08:41 nation.tbl
| -rw-r--r-- 1 root root 171952161 Jul 5 08:41 orders.tbl
| -rw-r--r-- 1 root root 118984616 Jul 5 08:41 partsupp.tbl
| -rw-r--r-- 1 root root 24135125 Jul 5 08:41 part.tbl
| -rw-r--r-- 1 root root 389 Jul 5 08:41 region.tbl
| -rw-r--r-- 1 root root 1409184 Jul 5 08:41 supplier.tbl
|
| Note: You will need all the above files. If you find any file missing, you need to regenerate the data.
| 6. Prepare the data for TPC-H:
| #cd $TPCH_DBGEN_HOME/dbgen/
| #mv *.tbl $TPC_H_HIVE_HOME/data/
|
| #cd $TPC_H_HIVE_HOME/data/
| #!/bin/bash
| ./tpch_prepare_data.sh

```

**Note:** Before executing `tpch_prepare_data.sh`, you need to ensure that IBM Spectrum Scale is active (`/usr/lpp/mmfs/bin/mmgetstate -a`), file system is mounted (`/usr/lpp/mmfs/bin/mm1smount <fs-name> -L`) and HDFS Transparency is active (`/usr/lpp/mmfs/bin/mmhadoopctl connector getstate`).

7. Check your prepared data:  

```
#hadoop dfs -ls /tpch
```

Check that all the files listed in Step 5 are present.
8. Run TPC-H:  

```
cd $TPC_H_HIVE_HOME/
#export HADOOP_HOME= /usr/hdp/current/hadoop-client
#export HADOOP_CONF_DIR=/etc/Hadoop/conf
#export HIVE_HOME= /usr/hdp/current/hive-client

vim benchmark.conf □ change the variables, such as LOG_FILE, if you want
#./tpch_prepare_data.sh
```

## HBase/YCSB

- HBase YCSB is a benchmark developed by Yahoo to test the performance of HBase.
- When running YCSB to evaluate the HBase performance over IBM Spectrum Scale, take the tuning accordingly as explained in the following sections.

### Tuning for YCSB/HBase

#### HBase configuration

1. Change the `hbase-site.xml` from Ambari if you take HortonWorks or IBM BigInsights. If you take open source HBase, you could modify `$HBASE_HOME/conf/hbase-site.xml` directly.

*Table 19. HBase Configuration Tuning*

Configuration	Default Value	Recommended Value	Comments
Java Heap	N/A	Refer the “Memory tuning” on page 116 section.	HBase Master server Heap Size; HBase Region Server Heap Size
<code>hbase.regionserver.handler.count</code>	30	60	
<code>zookeeper.session.timeout</code>	N/A	180000	
<code>hbase.hregion.max.filesize</code>	10737418240	10737418240	Check the default value. If it is not 10GB, change it into 10GB.
<code>hbase.hstore.blockingStoreFiles</code>	10	50	
<code>hbase.hstore.compaction.max</code>	10	10	
<code>hbase.hstore.compaction.max.size</code>	LONG.MAX_VALUE	Variable	If you see a lot of compaction, you could set this to 1GB to exclude those HFiles from compaction.
<code>hbase.hregion.majorcompaction</code>	604800000	0	Turn off the major compaction when running benchmark to ensure that the results are stable. In production, this should not be changed.
<code>hbase.hstore.compactionThreshold</code>	3	3	

| *Table 19. HBase Configuration Tuning (continued)*

Configuration	Default Value	Recommended Value	Comments
<code>hbase.hstore.compaction.max</code>	10	3	

| *Table 20. IBM Spectrum Scale Tuning*

Configuration	Default Value	Recommended Value	Comments
<code>pagepool</code>	1GB	30% of physical memory	30% of physical memory

| **Note:** 30% of physical memory is only for running HBase/YCSB. In production, you need to consider the memory allocation for other workloads. If you run Map/Reduce jobs, Hive jobs over the same cluster, you need to trace off the performance for these different workloads. If you allocate more memory for pagepool because of HBase, you will have fewer memory for Map/Reduce jobs and therefore degrade the performance for Map/Reduce jobs.

| *Table 21. YCSB Configuration Tuning*

Configuration	Default Value	Recommended Value	Comments
<code>writebuffersize</code>	12MB	12MB	
<code>clientbuffering</code>	False	True	For benchmark, keep this the same as what you use to run YCSB over native HDFS.
<code>recordcount</code>	1000	1000000	
<code>operationcount</code>	N/A	N/A	Depends on the number of operations you want to benchmark. For example, 20M operations
<code>threads</code>	N/A	Variable	Depends on the number of threads you want to benchmark.
<code>requestdistribution</code>	zipfian	Not changed	
<code>recordszie</code>	100*10	Not changed	YCSB for HBase takes 100 bytes per field and 10 fields for one record.

| **Important:** While creating the HBase table before running YCSB, you need to pre-split the table accordingly. For example, you need to pre-split the table into 100 partitions for ~10 HBase Region servers. If it is more than 10 HBase Region servers, you need to increase the pre-split partition number.

| If you do not pre-split the table, all requests are handled by limited HBase Region servers and therefore the performance of YCSB is impacted.

## | **Running YCSB/HBase**

| This section lists the steps to run YCSB/ HBASE.

- | 1. Download YCSB from YCSB 0.14.0
- | 2. Untar the YCSB package into `$YCSB_HOME`.
- | 3. Remove all the libraries shipped by YCSB:  
`#rm -fr $YCSB_HOME/hbase10-binding/lib/*`
- | 4. Copy the libraries from your Hadoop distro:
- |     The following is for HortonWorks HDP 2.6:

```

| cp /usr/hdp/2.6.0.3-8/hbase/lib/* $YCSB_HOME/ hbase10-binding/lib/
| cp /usr/hdp/2.6.0.3-8/Hadoop/*.jar $YCSB_HOME/ hbase10-binding/lib/
| 5. Create the following script accordingly:
| The script for loading:
| vim ycsb_workload_load.sh
| #!/bin/bash
| set -x

| # <script> <thread-number> <YCSB_RESULT_HOME>
|
| ${YCSB_HOME}/bin/ycsb load hbase10 -P ${YCSB_HOME}/workloads/workloada
| -p columnfamily=family -p recordcount=${RECORD_COUNT} -s -threads $1
| -p measurementtype=timeseries -p timeseries.granularity=2000 2>&1 >
| ${YCSB_RESULT_HOME}/$2/workload_load.output.thread-$1-.`date "+%y%m%d_%H%M%S"``

| In the above script, RECORD_COUNT is the record number you want to load into HBase. RECORD_COUNT
| should be 1M. <thread-number> depends on your running.
| If you want to change writebuffersize and clientbuffering, you could add -p <writebuffersize>
| -p <clientbuffering> for the above YCSB command.
| The script for YCSB workload A/B/C/D/E/F:
| vim ycsb_workload_a.sh
| #!/bin/bash
| # <script> <workloadA-recordcount> <thread-number> <result sub dir>

| ${YCSB_HOME}/bin/ycsb run hbase10 -P ${YCSB_HOME}/workloads/workloada
| -p columnfamily=family -p operationcount=${OPERATION_COUNT}
| -p recordcount=$1 -s -threads $2 -p measurementtype=timeseries
| -p timeseries.granularity=2000 2>&1 |
| tee ${YCSB_RESULT_HOME}/$3/workload_a.output.thread-$2.`date "+%y%m%d_%H%M%S"``

| Similarly create the other scripts for workload B/C/D/E/F.
| You need to first run the ycsb_workload_load.sh script. After it loads data into HBase, run
| ycsb_workload_a.sh or other scripts.

Spark

Spark Tuning (HortonWorks or IBM Spectrum Conductor)

| If you run Spark standalone distro (for example, community Spark, or IBM Spectrum Conductor with
| Spark), it is recommended to take IBM Spectrum Scale POSIX interface.

| If you run Spark standalone distro on IBM Spectrum Scale FPO, refer the “Tuning for IBM Spectrum
| Scale FPO” on page 117 section.

| If you run Spark standalone distro on IBM ESS, no need to tune further.

| If you take Hadoop distro (for example, HortonWorks HDP), it is recommended to take IBM Spectrum
| Scale HDFS Transparency. Refer the “System tuning” on page 116 and “HDFS Transparency Tuning” on
| page 117 sections.

| At Spark level, the following two should be tuned to make Spark work well on IBM Spectrum Scale:
```

Configuration	Default value	Recommended value
<b>spark.shuffle.file.buffer</b> (\$SPARK_HOME/conf/spark-defaults.conf)	32K	IBM Spectrum Scale data blocksize  <code>spark_shuffle_file_buffer=\$(./usr/lpp/mmfs/bin/mmlsfs &lt;filesystem_name&gt; -B   tail -1   awk '{ print \$2 }')</code>  If the blocksize of file system is larger than 2MB, configure 2MB for <b>spark.shuffle.file.buffer</b> .
<b>spark.local.dir</b>	/tmp	Configure the local directory for this (not configure this with Spectrum Scale directory).
<b>spark.hadoop.mapreduce.fileoutputcommitter.algorithm.version</b>	1	Changing this into 2 can make Spark job commit fast.

| As for other tuning for Spark, refer Spark configuration and Tuning Spark for Spark level tuning.

## | **Benchmarking Spark**

| For benchmarking Spark, follow the guide from some popular benchmarks, such as spark-bench, BigDataBench-Spark.

## | **Workloads/Benchmarks information**

| This topic describes the Workloads/Benchmarks information.

### | **Teragen**

| Shipped with Hadoop release.

### | **Terasort**

| Terasort is shipped with Hadoop package. It is located in \$BI\_HOME/IHC/hadoop-example.jar.

### | **TPC-H**

| Usually, TPC-H for big data is done with Hive. Also, some users will do TPC-H for big data with Pig.  
| The former one is more widely used.

| TPC-H data generation (DBGEN) should be downloaded from TPC-H website:

| TPC-H over hive:

| Running TPC-H queries on Hive

| TPC-H-Hive

| Imperative and Declarative Hadoop: TPC-H in Pig and Hive

| Hive is shipped with BigInsights. For more information, see APACHE HIVE TM.

### | **YCSB**

- | YCSB (Yahoo! Cloud System Benchmark) is widely used to benchmark some no SQL db, such as hbase.
- | You can download it from YCSB.

- | **Hibench**

- | Hibench is a workload-mixed benchmark. It consists of nine different workloads (such as TeraGen, Terasort, DFSIO, Hive etc). It is used to evaluate the cluster performance under many different workloads.

- | **HiBench**

- | **Hive**

- | Shipped with Biginsights (v2.1). You can download the benchmark for Hive from Running TPC-H queries on Hive.
- | For more information, see APACHE HIVE TM.



---

## Chapter 4. Hortonworks Data Platform 3.X

- | This section describes the deployment of Hortonworks Data Platform (HDP<sup>®</sup>) 3.X on the IBM Spectrum Scale™ file system by using the Apache© Ambari framework.
  - | This section specifies the HDP deviations requirements for IBM Spectrum Scale integration and IBM Spectrum Scale specifics information and should be read beforehand to understand any deviation required when following the Hortonworks installation documentation for your specific platform.
- 

### Planning

- | Review the Chapter 1, "Hadoop Scale Storage Architecture," on page 1 on which the configuration setup is to be used in your environment.

### Hardware requirements

- | This section specifies the hardware requirements to install Hortonworks Data Platform (HDP<sup>®</sup>), IBM Spectrum Scale Ambari management pack and HDFS Transparency on IBM Spectrum Scale.
- | In addition to the normal operating system, IBM Spectrum Scale and Hadoop requirements, the Transparency connector has minimum hardware requirements of one CPU (processor core) and 4GB to 8GB physical memory on each node where it is running. This is a general guideline and might vary. For more planning information, see Chapter 2, "IBM Spectrum Scale support for Hadoop," on page 9.

### Preparing the environment

- | This section describes how to prepare the environment to install Hortonworks Data Platform (HDP<sup>®</sup>), IBM Spectrum Scale Ambari management pack, and HDFS Transparency on IBM Spectrum Scale.
- | The IBM Spectrum Scale Ambari management pack can be used for Ambari 2.7 for Hortonworks HDP to setup the IBM Spectrum Scale Service.
- | HDP requires specific version combinations for the Ambari, Mpack and HDFS Transparency. The IBM Spectrum Scale file system is independent of the versioning for HDP, Mpack and HDFS Transparency. Hortonworks has been certified to work with IBM Spectrum Scale 4.2.3 and later.

### Support Matrix

- | This section describes the Hadoop distribution support matrix.

| *Table 22. Hadoop distribution support matrix*

IBM Spectrum Scale Ambari management pack (Mpack)	HDFS Transparency	Hadoop distribution	Ambari version	RHEL x86_64	RHEL ppc64le
Mpack 2.7.0.1	HDFS Transparency 3.1.0-0	HDP 3.0.1	Ambari 2.7.0.1	7.2/7.3/7.4/7.5	7.4/7.5
Mpack 2.7.0.0	HDFS Transparency 3.0.0-0	HDP 3.0.0	Ambari 2.7.0.0	7.2/7.3/7.4/7.5	7.4/7.5

#### Note:

- | • HDP and Mpack requires Python 2.7.

- HDP Java™ Development Kits (JDKs): OpenJDK on ppc64le and Oracle JDK on x86\_64.
- IBM Spectrum Scale file system release supported: 4.1.1.3+, 4.2.2.3+, 4.2.3.1+, 5.0.X+ on pc64le and x86\_64.
- RH 7.5 is supported from V4.2.3.9 and V5.0.1.1.
- IBM Spectrum Scale Management GUI function requires RHEL 7.2+ at IBM Spectrum Scale version 4.2.X+.
- IBM Spectrum Scale snap data for Hadoop function requires RHEL 7.2+ at IBM Spectrum Scale version 4.2.2.X+.
- Preserving Kerberos token delegation during Namenode failover with HDP is supported when Mpack 2.7.0.1 and HDFS Transparency 3.1.0-0 are applied together.

## **Set up**

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

### **Local Repository server**

- Set up a local repository server to be used for Ambari, IBM Spectrum Scale, and for the OS repository.
  1. Set up the Mirror repository server.
  2. Set up the Local OS repository if needed.

### **Base packages**

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

```
$ yum -y install kernel-devel cpp gcc gcc-c++ binutils
```
- For more information, see IBM Spectrum Scale *Software requirements* topic in the *IBM Spectrum Scale: Concepts, Planning, and Installation Guide*.

### **HDP packages:**

- Install libtirpc-devel package (From RH optional packages) on all nodes.
- MySQL community edition
  - For new database install option through HDP for Hive Metastore, the MySQL community would require internet access or have a local repository setup to deploy on the Hive Metastore host. For more information, see MySQL Community Edition repository.

The following recommended packages can be downloaded to all nodes:

- acl, libacl – to enable Hadoop ACL support
- libattr – to enable Hadoop extended attributes
- java-<version>-openjdk-devel – Development tool-kit required for short circuit

Some of these packages are installed by default while installing the operating system.

### **Create the anonymous user id**

- If the anonymous user id does not exist already, create it for all the nodes in the IBM Spectrum Scale cluster on the Hadoop cluster.
- The anonymous user is a mandatory user id while starting the IBM Spectrum Scale Service.

| Ensure that the UID and GID for the anonymous user have the same value for all the nodes in the IBM Spectrum Scale cluster.

| The anonymous user id is used for Hive.

| For more information, see “User and group ids” on page 142.

## | **Hortonworks Data Platform (HDP)**

| This topic helps in the preparation to install Hortonworks Data Platform (HDP).

| If you are installing the HDP package, follow the *Getting Ready, Meet Minimum System Requirements, Prepare the Environment, Using a Local Repository* and *Obtaining Public Repositories* sections from the installation guide of your platform. For the *Apache Ambari Installation for IBM Power Systems™* guide and *Apache Ambari Installation* guide for the x86 platform, see Hortonworks Documentation site for the HDP version you are using.

### | **Scale integration deviation requirement:**

| • Maximum Open Files Requirements

| Set the maximum number of open file descriptors to 65535.

| • When HDP is integrated with IBM Spectrum Scale, there is no Secondary Namenode.

| • See “IBM Spectrum Scale file system” for Kernel, SELinux and NTP, password-less ssh and User/group id for specific changes.

## | **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-3.0.0-(version)`.

| Save this module in the IBM Spectrum Scale repository.

| **Note:** Ensure that there is only one package of the transparency in the IBM Spectrum Scale repository.

| Rebuild the repository by executing the `createrepo .` command to update the repository metadata.

## | **IBM Spectrum Scale file system**

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

| For IBM Spectrum Scale overview, see the *Product overview* section in the *IBM Spectrum Scale: Concepts, Planning, and Installation Guide*.

| 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 trial 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 of `rpm -qa | grep kernel` 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 7.4.
- | **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
3.10.0-693.11.6.el7.x86_64 <== Find kernel-devel and kernel-headers to match this

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

#### | **SELinux and NTP:**

- | This topic gives information about SELinux and NTP.
  - | • SELinux must be in disabled mode.
  - | • Network Time Protocol (NTP)
- | Configure NTP on all the nodes in your system to ensure that the clocks of all the nodes are synchronized.
- | **On Red Hat Enterprise Linux nodes**

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

#### | **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:** BDA Ambari integration supports the admin mode central configuration of IBM Spectrum Scale (*adminMode configuration attribute* topic in the *IBM Spectrum Scale: Administration Guide*).
- | 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.

| The recommendation is to create the uid/gid manually on all the nodes for the service before deploying the service. This is because Ambari creates inconsistent uid/gid across the cluster after the initial deployment on a fresh cluster. See Jira AMBARI-12616. For a list of the service user and group account uid/gid see the *Default user accounts* and *Default group accounts* sections under Understanding service users and groups.

|     **Note:** Ensure that these user and group accounts uid and gid have the same values across all the IBM Spectrum Scale nodes.

- | • If you are using LDAP, create the IDs and groups on the LDAP server and ensure that all nodes can authenticate the users.
- | • If you are using local IDs, the IDs must be the same on all nodes with the same ID and group values across the nodes.
- | • If you setup remote mount access for IBM Spectrum Scale, the owning cluster does not require to have the Hadoop uid and gid configured because there are no applications running on those nodes.
- | However, if the owning cluster have other applications from non Hadoop clients, they need to ensure

- | that the uid and gid used by the Hadoop cluster are not the same as the one used by the non Hadoop clients. 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 more information, see "Create the anonymous user id" on page 138 section.
- | **Note:** UID or GID is the common way for a Linux system to control access from users and groups. For example, if the user Yarn UID=100 on node1 generates data and the user Yarn UID=200 on node2 wants to read this data, the read operation fails because of permission issues.
- | Keeping a consistent UID and GID for all users on all nodes is important to avoid unexpected issues.
- | For the initial installation through Ambari, the UID or GID of users are consistent across all nodes. However, if you deploy the cluster for the second time, the UID or GID of these users might be inconsistent over all nodes (as per the AMBARI-10186 issue that was reported to the Ambari community).
- | After deployment, check whether the UID is consistent across all nodes. If it is not, you must fix it by running the following commands on each node, for each user or group that must be fixed:
  - | ##### Change UID of one account:  
`| usermod -u <NEWUID><USER>`
  - | ##### Change GID of one group:  
`| groupmod -g <NEWGID><GROUP>`
  - | ##### Update all files with old UID to new UID:  
`| find / -user <OLDDUID> -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.
    - | Ensure there is a Mirror repository server created before proceeding.
    - | Setup the Local OS repository if needed.
    - | Setup the Local IBM Spectrum Scale repository. This section helps you to set up the IBM Spectrum Scale and HDFS Transparency local repository.
- | **IBM Spectrum Scale service (Mpack)**
  - | The IBM Spectrum Scale Ambari management pack is an Ambari service for IBM Spectrum Scale.

- | For traditional Hadoop clusters that use HDFS, an HDFS service is displayed in the Ambari console to provide a graphical management interface for the HDFS configuration (`hdfs-site.xml`) and the Hadoop cluster (`core-site.xml`). Through the Ambari HDFS service, you can start and stop the HDFS service, make configuration changes, and implement the changes across the cluster.
- | The management pack creates an Ambari IBM Spectrum Scale service to start, stop, and make configuration changes to IBM Spectrum Scale and HDFS Transparency. Once the IBM Spectrum Scale HDFS Transparency is integrated, the HDFS service manages the HDFS Transparency Namenodes and Datanodes instead of the native HDFS components.
- | Download the management pack from the IBM Spectrum Scale wiki page for HDP 3.X support at the following link:
  - HDP 3.X
- | The management pack version 2.7.x.x contains the following files:
  - `SpectrumScaleIntegrationPackageInstaller-2.7.x.x.bin`
  - `SpectrumScaleMPackInstaller.py`
  - `SpectrumScaleMPackUninstaller.py`
  - `SpectrumScale_UpgradeIntegrationPackage` [Upgrade Spectrum Scale Mpact]
  - `mpack_utils.py`
  - `sum.txt`
- | **Note:** Ensure that all the packages reside in the same directory before executing the executables.

## | **Preparing for FPO environment**

- | This section provides the information for preparing for a FPO deployment. In order to create a FPO cluster, a NSD file needs to be created beforehand.

### | **Preparing a stanza file**

- | The Ambari install process can install and configure a new IBM Spectrum Scale™ cluster file system and configure it for Hadoop workloads. To support this task, the installer must know the disks available in the cluster and how you want to use them. If you do not indicate preferences, intelligent defaults are used. The stanza file is used for new FPO deployment through Ambari by setting the GPFS NSD file field under the Spectrum Scale Standard Configs panel.
- | See “Preparing a stanza file” on page 245.

### | **Simple NSD File:**

- | This section describes a simple NSD file with an example.
- | Simple NSD file can only be used for full disk that have not already been partitioned as input to Ambari.
- | All disks of GPFS NSD server nodes requires to be listed in the NSD stanza file.
- | The following is an example of a preferred simple IBM Spectrum Scale NSD file:
  - | There are 7 nodes, each with 6 disk drives to be defined as NSDs. All information must be continuous with no extra spaces
  - | 

```
$ cp /var/lib/ambari-server/resources/gpfs_nsd.sample /var/lib/ambari-server/resources/gpfs_nsd
$ cat /var/lib/ambari-server/resources/gpfs_nsd
```
  - | `DISK|compute001.private.dns.zone:/dev/sdb,/dev/sdc,/dev/sdd,/dev/sde,/dev/sdf,/dev/sdg`

```

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

```

- | If you want to select disks such as SSD drives for metadata , add the label -meta to those disks.
- | In a simple NSD file, add the label meta for the disks that you want to use as metadata disks, as shown in the following example. If -meta is used, the Partition algorithm is ignored.
 

```
$ cat /var/lib/ambari-server/resources/gpfs_nsd
|
| DISK|compute001.private.dns.zone:/dev/sdb-meta,/dev/sdc,/dev/sdd
| DISK|compute002.private.dns.zone:/dev/sdb-meta,/dev/sdc,/dev/sdd
| DISK|compute003.private.dns.zone:/dev/sdb-meta,/dev/sdc,/dev/sdd
| DISK|compute005.private.dns.zone:/dev/sdb-meta,/dev/sdc,/dev/sdd
| DISK|compute006.private.dns.zone:/dev/sdb,/dev/sdc,/dev/sdd
| DISK|compute007.private.dns.zone:/dev/sdb,/dev/sdc,/dev/sdd
```
- | In the simple NSD file, /dev/sdb from compute001, compute002, compute003, and compute005 are specified as meta disks in the IBM Spectrum Scale file system.
- | The partition algorithm is ignored if the nodes listed in the simple NSD file do not match the set of nodes that will be used for the NodeManager service. If nodes that are not NodeManagers are in the NSD file or nodes that will be NodeManagers are not in the NSD file, no partitioning will be done.

#### | Standard NSD file:

- | This section describes a standard NSD file with an example.
- | The following is an example of a Standard IBM Spectrum Scale NSD File
 

```
%pool: pool=system blockSize=256K layoutMap=cluster allowWriteAffinity=no
%pool: pool=datapool blockSize=2M layoutMap=cluster allowWriteAffinity=yes writeAffinityDepth=1
blockGroupFactor=256

gpfstest9
%nsd: nsd=node9_meta_sdb device=/dev/sdb servers=gpfstest9 usage=metadataOnly failureGroup=101 pool=system
%nsd: nsd=node9_meta_sdc device=/dev/sdc servers=gpfstest9 usage=metadataOnly failureGroup=101 pool=system

%nsd: nsd=node9_data_sde2 device=/dev/sde2 servers=gpfstest9 usage=dataOnly failureGroup=1,0,1 pool=datapool
%nsd: nsd=node9_data_sdf2 device=/dev/sdf2 servers=gpfstest9 usage=dataOnly failureGroup=1,0,1 pool=datapool

gpfstest10
%nsd: nsd=node10_meta_sdb device=/dev/sdb servers=gpfstest10 usage=metadataOnly failureGroup=201 pool=system
%nsd: nsd=node10_meta_sdc device=/dev/sdc servers=gpfstest10 usage=metadataOnly failureGroup=201 pool=system

%nsd: nsd=node10_data_sde2 device=/dev/sde2 servers=gpfstest10 usage=dataOnly failureGroup=2,0,1 pool=datapool
%nsd: nsd=node10_data_sdf2 device=/dev/sdf2 servers=gpfstest10 usage=dataOnly failureGroup=2,0,1 pool=datapool

gpfstest11
%nsd: nsd=node11_meta_sdb device=/dev/sdb servers=gpfstest11 usage=metadataOnly failureGroup=301 pool=system
%nsd: nsd=node11_meta_sdc device=/dev/sdc servers=gpfstest11 usage=metadataOnly failureGroup=301 pool=system

%nsd: nsd=node11_data_sde2 device=/dev/sde2 servers=gpfstest11 usage=dataOnly failureGroup=3,0,1 pool=datapool
%nsd: nsd=node11_data_sdf2 device=/dev/sdf2 servers=gpfstest11 usage=dataOnly failureGroup=3,0,1 pool=datapool
```
- | **Note:** Starting with IBM Spectrum Scale version 5.0.0, the default block size is 4M.
- | Type the /var/lib/ambari-server/resources/gpfs\_nsd filename in the NSD stanza field.

| Because of the limitations of the Ambari framework, the NSD file must be copied to the Ambari server in the /var/lib/ambari-server/resources/ directory. Ensure that the correct file name is specified on the IBM Spectrum Scale Customize Services page.

| If you are using a standard NSD stanza file and only one datapool is defined, you can either specify the policy file or let it be done by IBM Spectrum Scale. However, if you have more than one data pool, you should specify a policy to define the location of the data in the data pool. If there is no policy specified, by default the data will be stored to the first data pool only.

#### | **Policy File:**

| This section describes a policy file with an example.

| The bigpfs.pol is an example of a policy file.

| RULE 'default' SET POOL 'datapool'

---

## | **Installation**

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

| This installation section will describe how to install ESS Remote mount with all Hadoop nodes as Spectrum Scale nodes as the 1st deployment model as described in the Chapter 1, “Hadoop Scale Storage Architecture,” on page 1 section.

| Before starting the software deployment, follow the Planning section to setup the environment and download the software.

#### | **Note:**

- | • This chapter describes how to add IBM Spectrum Scale service as a root user. If you plan to restrict root access, review the “Restricting root access” on page 327 section.
- | • To install the IBM Spectrum Scale service, an existing HDFS cluster is required. This can be created by installing the HDP stack with native HDFS.

| For other installation scenarios, see “Configuration” on page 167 section.

## | **ESS setup**

| The ESS is setup and tuned by IBM Lab Services.

| **Note:** Ensure that the IBM Spectrum Scale file system ACL setting is set to *ALL*.

| For more information see HDFS and Spectrum Scale filesystem ACL support section.

## | **Adding Services**

| HDFS and IBM Spectrum Scale are different file systems. If services are added only to one of the file systems, the other file system does not have the data for that service. Therefore, on switching from native HDFS to IBM Spectrum Scale or vice versa, the service cannot provide the data that you entered before switching the file system.

| Ensure that you follow the “User and group ids” on page 142 section for uid/gid consistency setup and verification.

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

- | • HDFS
  - | • Yarn
  - | • Mapreduce2
  - | • Zookeeper
  - | • SmartSense (HDP)
- | When adding new services to HDP, ensure that you review all configurations in the Customize Services wizard tabs. Especially, fields that set directories. This is because IBM Spectrum Scale is a shared file system. Ensure that any services do not use a shared file system mount point when it requires a local directory. See **Create HDP cluster>Installing, Configuring, and Deploying a Cluster >Customize Services>Directories** section.
- | If the service is already added with the remote mount point set as one of the directories, and you want to modify or remove the remote mount point directory from the service, ensure that you restart the service to pick up the new configuration changes.
- | If you are planning to install High Availability (HA), ensure that you setup the HA in native HDFS mode.
- | Do not install HA when IBM Spectrum Scale Service is integrated.
- | If IBM Spectrum Scale is integrated, see “HDFS Namenode High Availability [HA]” on page 167.

## | **Create HDP cluster**

- | Follow the Installing Ambari section in Hortonworks HDP installation documentation (<https://docs.hortonworks.com>) for your specific platform.
- | This section will mention the deviations required in the setup while integrating with IBM Spectrum Scale.
- | **Note:** Ambari uses the mount points it finds on the Ambari server to set up the services during deployment. If you do not want to use the mount points, ensure that you unmount the mount points on the Ambari server node before starting Ambari or find and change the mount point in the configuration setting for each service that uses it. See **Create HDP cluster>Installing, Configuring, and Deploying a Cluster >Customize Services>Directories** for a list of services that use the mount point value.
- | Ensure that you follow the “User and group ids” on page 142 section for uid/gid consistency setup and verification.

## | **Install Ambari bits**

- | Ensure that the ambari.repo is setup on /etc/yum.repos.d on the Ambari server host.
- | On the Ambari server host, run:
- ```
| yum install ambari-server
```
- | • Update the /etc/ambari-server/conf/ambari.properties file to point to the correct Open JDK and JCE files.


```
| $ vi /etc/ambari-server/conf/ambari.properties
| jdk1.8.jcpol-url to point to the correct jce_policy-8.zip file.
| jdk1.8.url to point to the correct jdk-8u112-linux-x64.tar.gz file.
```
 - | • Update the number of threads in /etc/ambari-server/conf/ambari.properties file.

| The size of the threadpool must be set to the number of logical cpus on the node on which the Ambari server is running. When the number of threads are not enough in Ambari, the system might suffer a heartbeat loss and the datanodes might go down. The Ambari GUI might not be able to start if enough threads are not available. This is especially true for Power system.
- | Threadpool values requiring to be modified in /etc/ambari-server/conf/ambari.properties:

```
| $ vi /etc/ambari-server/conf/ambari.properties  
| server.execution.scheduler.maxThreads=<number of logical cpu's>  
| client.threadpool.size.max=<number of logical cpu's>  
| agent.threadpool.size.max=<number of logical cpu's>  
| To calculate the number of logical cpus:  
| $ lscpu  
| Thread(s) per core: 8  
| Core(s) per socket: 1  
| Socket(s): 20  
| Number of logical cpu's = Thread(s) per core x Core(s) per socket x Socket(s) = 8 x 1 x 20 = 160
```

| **Set up the Ambari server**

- | This topic describes how to set up the Ambari server.
- | On the Ambari server host, run:
 - | ambari-server setup
- | **Note:** If you are using Hive MySQL, you also need to set up the MySQL connector and run
 - | ambari-server setup --jdbc-db=mysql --jdbc-driver=/path/to/mysql/mysql-connector-java.jar.
- | See Hortonworks Using Hive with MySQL.
- | If you plan to use a non-root user id, see “Restricting root access” on page 219.

| **Installing, Configuring, and Deploying a Cluster**

- | This section states the deviations from the Hortonworks documentation, Installing, Configuring, and Deploying a Cluster section when integrating IBM Spectrum Scale service.

| **Start the Ambari server:**

- | This section describes the procedure to start the Ambari server.
- | On the Ambari server host, run: ambari-server start.

| **Log in to Apache Ambari:**

- | Open <http://<your.ambari.server>:8080> in your web browser.

| **Select version:**

- | Select the software version and method of delivery for your cluster.
- | Using a Local Repository requires you to configure the software in a repository available in your network.

| For this deployment:

- | • Select Use Local Repository.
- | • Set HDP and HDP UTILS repository for redhat7/redhat-ppc7.

| **Install Options:**

- | The cluster wizard prompts for general information on how to setup the cluster.
- | In the Target hosts section:
 - | • The ESS I/O servers must not be a part of the Ambari cluster.

- Ambari requires a list of fully qualified domain names (FQDNs) of the nodes in the cluster.
- Verify that the list of the host names used in the Ambari Target Hosts section are the data network addresses that IBM Spectrum Scale uses for the cluster setup. Otherwise, during the installation of the IBM Spectrum Scale service, the installation fails and gives an Incorrect hostname error.
- Ensure that the hostname does not have mixed case. Otherwise, failures might occur when starting services. It is recommended to use all lower case.

| For this deployment, the ssh user is set to *root*.

| If you plan to use a non-root user id, see “Restricting root access” on page 219.

| **Choose Services:**

| Choose the services to install into the cluster.

| **Note:** For this configuration, Ranger is unchecked. You can add the Ranger at a later point.

| For Scale integration specific information regarding adding services, see “Adding Services” on page 146 section.

| **Assign masters:**

| The Cluster Install wizard assigns the master components for selected services.

| **Note:**

- GPFS Master node should be co-located on the Ambari server host.
- The native HDFS Namenode will also become the IBM Spectrum Scale HDFS Transparency Namenode.
- It is recommended to assign the Yarn Resource Manager onto the native HDFS Namenode.

| **Assign Slaves and Clients:**

| The Cluster Install wizard assigns the slave components to appropriate hosts in the cluster.

| **Note:** Select all nodes to be data nodes. This installs HDFS Transparency on all the IBM Spectrum Scale nodes on the Hadoop cluster.

| **Customize Services:**

| Customize services set of tabs to review and modify your cluster setup. The wizard sets defaults for each options.

| Ensure that you review these settings carefully.

| **Note:** Accumulo and HiveServer2 if installed on the same host will have port binding conflicts.

| To work around the port conflict when starting the services:

- Put Accumulo and HiveServer2 on different hosts or
- Use non-default port for either of the service

| **Hive Database**

| In Hive panel, if you are using MySQL you need to set up the MySQL connector.

| To use MySQL with Hive, you must download the MySQL Connector/J. On downloading to the Ambari Server host, run:

```
| ambari-server setup --jdbc-db=mysql --jdbc-driver=/path/to/mysql/com.mysql.jdbc.Driver
```

| Ambari SSL configuration

| The Knox gateway server uses the 8443 port by default. The Knox gateway fails to start if the Ambari HTTPS uses the same port. Ensure that the Ambari HTTPS port and the Knox gateway port are unique, and are not used by other processes.

| Directories

| If the cluster has a mounted file system, Ambari can select mounted paths other than / as the default directory value for some of its services, when the local file system must be used. This might include a GPFS mounted directory. Either unmount the mount points on the Ambari server before starting Ambari, or you must manually find all the places in the Ambari installation configuration and set it to a local directory. Otherwise HDP services will not start or run correctly as the nodes in the cluster are accessing the same directories.

| Following is the list of service configurations that need the local directory to be configured:

| Service | Field Name | GUI Field Name | Local directory setting |
|----------------|------------------------------------|------------------------------------|--|
| HDFS | dfs.namenode.name.dir | NameNode directories | /hadoop/hdfs/namenode |
| HDFS | dfs.datanode.data.dir | DataNode directories | /hadoop/hdfs/data |
| Yarn | yarn.nodemanager.local-dirs | YARN NodeManager Local directories | /hadoop/yarn/local |
| Yarn | yarn.nodemanager.log-dirs | YARN NodeManager Log directories | /hadoop/yarn/log |
| Ambari Metrics | hbase.rootdir | HBase root directory | file:///var/lib/ambari-metrics-collector/hbase |
| Kafka | log.dirs | Log directories | /kafka-logs |
| Oozie | oozie_data_dir | Oozie Data Dir | /hadoop/oozie/data |
| Zookeeper | dataDir | ZooKeeper directory | /hadoop/zookeeper |

| Install, Start and Test:

| Ambari installs, starts, and runs a simple test on each component.

| In the event of any failure during the initial cluster deployment, it is a good practice to go through each service one by one by running its service check command. Ambari runs all the service checks as part of the installation wizard, but if anything were to fail, Ambari might not have run all the service checks. On the dashboard page for each service in the Ambari GUI, go to each service's panel and click **Actions > Run Service Check**.

| Summary - Complete:

| After the summary page shows a list of accomplished tasks, choose **Complete** to open up the Ambari dashboard.

| Note:

- Ensure that all services are up.
- Ensure that all service checks passed.
- Ensure that all the UID and GID have the same value across all the IBM Spectrum Scale cluster hosts after HDP is deployed.

| Establish a IBM Spectrum Scale cluster on the Hadoop cluster

| Establish a local IBM Spectrum Scale cluster on the Hadoop cluster. This local IBM Spectrum Scale cluster
| accesses the ESS via remote mount. This creates a multi-cluster Scale environment and one IBM ESS
| storage can be shared with different groups where the remote mount mode can isolate the storage
| management from the IBM Spectrum Scale local cluster.

| Note:

- Ensure that the version of IBM Spectrum Scale on the local cluster is higher than or same as the
version on the file system owning the cluster (ESS).
- The **maxblocksize** value requires to be the same on the local IBM Spectrum Scale cluster and the ESS
cluster. The **maxblocksize** value can be set up during the installation of the local IBM Spectrum Scale
cluster to be the same value as the ESS cluster. If the **maxblocksize** is not set, it defaults to 1 MB for
releases prior to IBM Spectrum Scale version 5.0.0, however from 5.0.0 release onwards it is set to 4
MB.

| Steps

| 1. Ensure that the gpfs.repo is in the /etc/yum.repos.d directory on all the nodes in the Hadoop cluster.
| On each node, run: yum clean all; yum makecache.

| For example, the /etc/yum.repos.d/gpfs.repo file contains:

```
[GPFS-5.0.1]
name=gpfs-5.0.1
baseurl=http://60.2.0.229/repos/rhel/5.0.1/GPFS_5.0.1
enabled=1
gpgcheck=0
```

| 2. Install the IBM Spectrum Scale on your cluster. See the *Manually installing the IBM Spectrum Scale
software packages on Linux nodes* topic in the *IBM Spectrum Scale: Concepts, Planning, and Installation
Guide*.

| For example, on each of the Hadoop nodes, run:

```
yum -y install gpfs.adv* gpfs.base* gpfs.crypto* gpfs.ext* gpfs.gpl* gpfs.gskit* gpfs.lice* gpfs.msg*
```

| 3. Build the kernel portability layer on each node by issuing the following command:

```
/usr/lpp/mmfs/bin/mmbuildgp1
```

| 4. Follow the *Steps for establishing and starting your IBM Spectrum Scale cluster* topic in the *IBM Spectrum
Scale: Concepts, Planning, and Installation Guide* for your specific Scale version.

| Note:

- Do not create NSD (**mmcrnsd**) or a filesystem (**mmcrfs**) because this is a remote mount environment.
- Ensure that the Ambari server is set as the IBM Spectrum Scale quorum node. The IBM Spectrum
Scale Master node resides on the Ambari server node and requires to be set as a quorum node.

| 5. Check the **maxblocksize** value on the local cluster and ESS cluster by running the following
command:

```
/usr/lpp/mmfs/bin/mm1sconfig | grep maxblocksize
```

| If **maxblocksize** value on the local cluster is not set or not the same as the ESS cluster, then on the
local cluster, run the following command:

```
/usr/lpp/mmfs/bin/mmchconfigmaxblocksize=<ESSmaxblocksizevalue>
```

| 6. Start IBM Spectrum Scale by issuing the **mmstartup** command.

```
/usr/lpp/mmfs/bin/mmstartup -a
```

| 7. Ensure that all the IBM Spectrum Scale nodes are in active state.

```
/usr/lpp/mmfs/bin/mmgetstate -a
```

| 8. Tune the local cluster as an ESS client:

| For remote mount mode for Hadoop cluster (1st model: Remote mount with all Hadoop nodes as IBM
Spectrum Scale nodes), run the following commands:

| On ESS run:
| `scp /usr/lpp/mmfs/samples/gss/gssClientConfig.sh root@<Hadoop_local_scale_cluster_host>:</path-to-gssclient>
| On Hadoop local scale cluster host, run:
| `<path-to-gssclient>/gssClientConfig.sh all
| However, if the Spectrum Scale clients nodes and the ESS nodes are in the same cluster (3rd model: Single cluster with all Hadoop nodes as IBM Spectrum Scale nodes), then run the gssClientConfig.sh script from the ESS node with `<path-to-gssclient>/gssClientConfig.sh <gpfs-client-node1, gpfs-client-node2, gpfs-client-node3, ...>`. For additional information, see the *Adding IBM Spectrum Scale nodes to the ESS cluster* topic in the *Elastic Storage Server: Quick Deployment Guide*.
| After running this script, restart GPFS™ on the affected nodes for the optimized configuration settings to take effect.

| Configure remote mount access

| To configure remote mount access, an existing local IBM Spectrum Scale cluster is required. This cluster must be a different cluster from the ESS-based cluster.
| See “Establish a IBM Spectrum Scale cluster on the Hadoop cluster” on page 151 on how to create the local IBM Spectrum Scale cluster.

| Note:

- | • The Hadoop local IBM Spectrum Scale cluster is the accessingCluster.
 - | • The ESS IBM Spectrum Scale cluster is the owningCluster.
 - | • Ensure that the local clusters have password-less ssh to the first node or all the nodes listed in the contact node list. To see the contact node list, run the **mmremotecluster show all** command.
- | Follow the *Mounting a remote GPFS file system* topic in the *IBM Spectrum Scale: Administration Guide* to configure remote mount access between the multiple IBM Spectrum Scale clusters. After the remote mount is configured, ensure that the accessing cluster can read/write to the mount point using POSIX.

| Install Mpack package

| This topic lists the steps to install the management pack.

| **Note:** Before you proceed, ensure that you review the “Support Matrix” on page 137 section to download the correct package for your environment.

- | 1. Ensure that the management pack, “IBM Spectrum Scale service (Mpack)” on page 143, is downloaded and unzipped into a local directory on the Ambari server node. This example uses the /root/GPFS_Ambari directory.
| `\$ cd /root/GPFS_Ambari
| \$ tar -xvzf SpectrumScaleMPack-2.7.X.X.noarch.tar.gz
- | 2. Stop all services:
| Log 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-2.7.X.X.bin executable:

| **Note:** The Management Pack can only be executed as root.

- | • On the Ambari server node, run `cd /root/GPFS_Ambari` to enter the directory.
- | • Run the installer bin to accept the license. The Mpack will be automatically generated and installed on the Ambari server, and the Ambari server will be restarted after the executable completes.
| `\$ cd /root/GPFS_Ambari
| \$./SpectrumScaleIntegrationPackageInstaller-2.7.0.0.bin

| Ensure that you know the input values before running the installer script.

- Input fields:
 - Ambari server port number: The port that was set up during the Ambari installation.
 - Ambari server IP address: The Ambari server IP address used during Ambari installation. If a node has multiple networks, specifying the IP address guarantees that the address is used.
 - Ambari server username: The Ambari server admin user name.
 - Ambari server password: The Ambari server admin user password.
 - Enter kdc principal: The kdc server principal if Kerberos is enabled
 - Enter kdc password: The kdc password if Kerberos is enabled
- Note:** This script automatically restarts the Ambari server.
- To complete this installation, “Deploy the IBM Spectrum Scale service” in Ambari GUI.

Deploy the IBM Spectrum Scale service

This section lists the steps to add and deploy the IBM Spectrum Scale™ service through Ambari based on the architecture stated in the “Deployment model” on page 3 section.

Note:

- Before you proceed, ensure that you review the Preparing the environment section.
- Review the “Limitations” on page 225 section.
- The IBM Spectrum Scale cluster running the application requires to have consistent uid/gid configuration. Ensure all the user ID and group ID and Hadoop service user ID and group ID are the same across all the IBM Spectrum Scale nodes on the local Hadoop cluster. The owning cluster of the remote mount does not require uid/gid setup. For more information, see “User and group ids” on page 142.
- The user id anonymous is required. See “Create the anonymous user id” on page 138.
- If configured as non-root, see “Restricting root access” on page 327 section.
- For cluster with IBM Spectrum Scale installed:
 - Ensure that IBM Spectrum Scale is active and mounted.


```
/usr/lpp/mmfs/bin/mmgetstate -a  
/usr/lpp/mmfs/bin/mmlsmount all or  
/usr/lpp/mmfs/bin/mmmount <fs-name> -a
```
 - Ensure the local Hadoop IBM Spectrum Scale cluster set the IBM Spectrum Scale Master node (which is the Ambari server node) as a quorum node.
- Secondary Namenode:
 - The Secondary NameNode in native HDFS is not needed for HDFS Transparency because the HDFS Transparency Namenode is stateless and does not maintain FSImage-like or EditLog information.
 - The Secondary NameNode should not be shown in the HDFS service GUI when the IBM Spectrum Scale service is integrated.
- Ambari uses the mount points it finds on the Ambari server to set up the services during deployment. If you do not want to use the mount points, ensure that you unmount the mount points on the Ambari server node before starting Ambari or find and change the mount point in the configuration setting for each service that uses it. See **Create HDP cluster > Installing, Configuring, and Deploying a Cluster > Customize Services Directories** section.

Log in to Apache Ambari

Log back into Ambari to add the IBM Spectrum Scale service.

Note:

- All services should be down. If not, ensure to stop the services.

- For cluster with IBM Spectrum Scale file system installed, ensure IBM Spectrum Scale is active and mounted. Ensure that you review the Ambari mount point usage bullet under the “Deploy the IBM Spectrum Scale service” on page 153 section before proceeding.

| Click **Services > Add service**.

| **Choose Services**

| On the Add Service Wizard, choose services panel, select the Spectrum Scale package and click **Next**.

| **Note:** The actual version for IBM Spectrum Scale deployment is based on the IBM Spectrum Scale repository. Hence the value shown in the Choose Services panel will not correspond to your actual IBM Spectrum Scale version.

| **Assign Masters**

| Co-locate the GPFS™ Master component to the same host as the Ambari-server. Click **Next**.



| **Assign Slaves and Clients**

| Select the GPFS Node components check box on ALL hosts on the **Assign Slaves and Agents** page. Click **Next**.

| **Note:**

- For client-only nodes where you do not want IBM Spectrum Scale, do not select the GPFS Node option.
- Review the GPFS Node column for the Namenode and Datanodes hosts that are part of the HDFS cluster.
- Selecting the GPFS Node column for those nodes run the IBM Spectrum Scale and IBM Spectrum Scale HDFS Transparency.
- The GPFS Master node is a GPFS Node which is the Ambari Server node.
- HDFS Transparency data node is required to be a Hadoop Datanode, a NodeManager, and a GPFS Node.

| **Customize Services**

| Configuration fields on both standard and advanced tabs are populated with values taken from the IBM Spectrum Scale Hadoop performance tuning guide.

| For all setups, the parameters with a lock icon must not be changed after deployment. These include parameters like the cluster name, remote shell, file system name, and max data replicas. Review the IBM Spectrum Scale configuration parameter checklist.

| Review whether all the services configuration are correct, especially the ones listed in the “Create HDP cluster” on page 147 > “Installing, Configuring, and Deploying a Cluster” on page 148 > “Customize Services” on page 149 > Directories section. When adding Spectrum Scale services, the service configuration directories might change to include the mount points that are now mounted.

| **Note:**

- Assign the metadata disks to the HDFS transparency Namenode running over a GPFS node.
- Assign Yarn ResourceManager on the node running HDFS Transparency NameNode.

| Ensure that you check the values are correct for your environment before clicking **Next**.

| Standard Tab

| Review GPFS Environment Definition

| Field | Value |
|---------------------------------------|--|
| GPFS Cluster Name | Name of IBM Spectrum Scale on local Hadoop cluster

See the <i>mmlscluster command</i> topic in the <i>IBM Spectrum Scale: Command and Programming Reference</i> . |
| GPFS quorum nodes | Nodes that are Spectrum Scale quorum nodes.

See the <i>mmlscluster command</i> topic in the <i>IBM Spectrum Scale: Command and Programming Reference</i> . |
| gpfs.storage.type | remote (for ESS remote mount deployment) |
| gpfs.remotecluster.autorefresh | True (for ESS remote mount deployment) |
| gpfs.ssh.user | User id IBM Spectrum Scale configured as.

This example uses the root user id. |

| Review GPFS Filesystem Definition

| Field | Value |
|---|---|
| GPFS FileSystem Name | Local name of the remote mounted file system.

Only one local name can be configured. |
| gpfs.mnt.dir | Local mount point name. Only one mount point name can be configured. |
| Verify if the disks are already formatted as NSDs | A value of Yes specifies that the NSDs are to be created only if each disk has not been formatted as a NSD by a previous invocation of the mmcrnsd command. The default value is Yes. A value of No specifies that the disks are to be created irrespective of their previous state.

Important: Setting this field to No would result in data stored over those NSDs being erased during the service deployment process. Therefore, set this field to No only when intended. |

| For example, the fields would then be set as follows based on the “Configure remote mount access” on page 152 configuration settings:

| | |
|---------------------------|--------|
| gpfs.storage.type: | Remote |
| GPFS file system Name: | essfs |
| gpfs.mnt.dir: | /essfs |

| Note:

- In Ambari, the IBM Spectrum Scale configuration value for **gpfs.storage.type** is set as remote. However, the **gpfs.storage.type** value that is seen in the /var/mmfs/hadoop/etc/hadoop/gpfs-site.xml is set as shared. Ensure that you update only through Ambari.
- Do not set the GPFS NSD stanza file for ESS or Share mode.

| Create Hadoop local cache disks

- [Optional] For ESS/shared storage, create the local cache disk for Hadoop usage.
- 1. Create the Hadoop local cache disk stanza file, `hadoop_disk`, in the `/var/lib/ambari-server/resources` directory.
- Hadoop local cache disk stanza file example:

```
# cat /var/lib/ambari-server/resources/hadoop_disk
DISK|compute001.private.dns.zone:/dev/sdb,/dev/sdc,/dev/sdd,/dev/sde,/dev/sdf,
/dev/sdg,/dev/sdi,/dev/sdj,/dev/sdk,/dev/sd1,/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/sd1,/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/sd1,/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/sd1,/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/sd1,/dev/sdm,/dev/sdn,/dev/sdo,/dev/sdp
```
- 2. Add the filename, `hadoop_disk`, to the **Hadoop local cache disk stanza file** field in the **Standard config** tab.

Advanced Tab

Review Advanced gpfs-ambari-server-env Definition

| Field | Value |
|----------------------|--|
| AMBARI_USER_PASSWORD | Ambari user password |
| GPFS_REPO_URL | <p>There must be no leading or trailing spaces in the repo name. This directory should contain the IBM Spectrum Scale filesystem rpms and the HDFS Transparency rpm. Only one HDFS Transparency rpm should reside in this directory.</p> <p>For example: <code>http://60.2.0.229/repos/GPFS/x86_64/rhel/5.0.1</code></p> |

Kerberos setting

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

| | |
|------------------------|-----------------------------|
| KDC_Principal | Kerberos Principal |
| KDC_PRINCIPAL_PASSWORD | Kerberos Principal password |

- If Kerberos is disabled, ignore the **KDC_PRINCIPAL** and **KDC_PASSWORD** fields under the **Customize Services** panel.
- If Kerberos is already enabled, then enter the **KDC_PRINCIPAL** and **KDC_PASSWORD** fields under the **Customize Services** panel.
- In a Kerberos environment, verify all the configuration information in the Customize Services panel before clicking **NEXT** to go to configure the Configure Identities panel.
- Note:** Under Advanced gpfs-env, the IBM Spectrum Scale version and HDFS Transparency version will be set automatically once deployment is completed. These fields are set by Ambari whenever HDFS service is restarted. All the IBM Spectrum Scale nodes must have the same version of HDFS Transparency and IBM Spectrum Scale file system installed.

| Review

| Review the configuration before installation. Click **Deploy**.

| Install, Start and Test

| Selected services will be installed and started. Click **Next**.

| Note:

- If deployment fails, fix any issues and click on the **Retry** button.
- If some services failed to start, you can Click **Next** to complete the deployment and start the service up manually on the Ambari dashboard.

| Summary > Complete

| After the summary page shows a list of accomplished tasks. Choose **Complete** to open up the Ambari dashboard.

| If the Spectrum Scale mount point was unmounted on the Ambari server, after the ADD Service deployment completes, the Spectrum Scale mount point is automatically remounted.

| **Note:** After the IBM Spectrum Scale service is integrated, the HDFS scripts are modified so that the HDFS service in Ambari will operate on the IBM Spectrum Scale HDFS Transparency components instead of native HDFS.

| Post setup

| This section lists the steps to be performed post setup.

1. Log into Ambari GUI
2. On a separate console, check the `/usr/lpp/mmfs/bin/mmlsfs -r` replication value and update the HDFS config panel **dfs.replication** field in Ambari if the value does not match the `mmlsfs` command output. For more information, see the *mmlsfs command* topic in the *IBM Spectrum Scale: Command and Programming Reference*.

| **Note:** For IBM® ESS file system, if the file system replication is set to 1, the HDFS **dfs.replication** field should be set as 1 in the HDFS config panel in Ambari to ensure that the services do not get misleading error message for not having enough space. For example: spark2 thrift server down due to do not have enough space error.

3. IBM Spectrum Scale service is up with Restart Required icon.
- Restart Spectrum Scale service.
- Run service check for Spectrum Scale service.
4. HDFS Transparency is now integrated into HDFS service.
5. Start all other services from Ambari. Click **Ambari GUI > Services > Start All**. If some services did not start properly, start them by going to the host dashboard, and restarting each service individually.

| **Note:** HDFS will get alerts on the Namenodes. 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**.

| Important:

- Restart all affected components with Stale Configs when the dashboard displays the request.
- If any configuration in the `gpfs-site` is changed in the IBM Spectrum Scale dashboard in Ambari, a restart required alert is displayed for the IBM Spectrum Scale service and the HDFS service. Check your environment to ensure that the changes made are in effect.
- IBM Spectrum Scale service must be restarted and only then can the HDFS service be restarted.

Verifying installation

- After HDP with IBM Spectrum Scale service is deployed, verify the installation setup.
- Note:** To run IBM Spectrum Scale commands, add the `/usr/lpp/mmfs/bin` directory to the environment PATH.
- 1. Verify all uid/gid values for user to ensure that they are consistent across all IBM Spectrum Scale nodes. Check by using `mmdsh -N all id<user-name>` to see whether the UID is consistent across all nodes.
- 2. Check the IBM Spectrum Scale installed packages on all nodes by using `rpm -qa | grep gpfs` to verify that all base IBM Spectrum Scale packages have been installed.
- 3. As user, ambari-qa, check the user id and access to the file system.
 - HDFS commands:
`$hadoopfs -ls /user`
 - POSIX commands:
`$pwd; ls -ltr`
 - #Create a file with entry
`$ echo "My test" > mytest`
`$ cat mytest`
 - HDFS commands:
`$ hadoop fs -cat mytest`
 - POSIX commands:
`$ rm mytest`
`$ls -ltr`
- 4. Run wordcount as user.
 - a. Create a mywordcountfile.
 - b. Use the mywordcountfile file to be used as input to the wordcount program.
`$ yarn jar`
`/usr/hdp/3.0.0.0-1634/hadoop-mapreduce/ hadoop-mapreduce-examples-3.1.0.3.0.0.0-1634.jar`
`wordcount mywordcountfile wc_output`
 - c. Check the output in directory
`$ hadoop fs -ls wc_output`
- 5. Run teragen/terasort. See Hortonworks Smoke Test Mapreduce.

Upgrading and uninstallation

Upgrading IBM Spectrum Scale service MPack

- This section describes the IBM Spectrum Scale MPack upgrade process.
 - You must plan a cluster maintenance window and prepare for cluster downtime when you upgrade the IBM Spectrum Scale MPack.
- Note:**
- You must perform the Mpack upgrade only if the target Mpack version is supported on your HDP level. Ensure that you check the support matrix and verify whether the Mpack version is supported with your HDP level.
 - To see the default configuration modifications under Spectrum Scale Mpacks, refer to the Big data and analytics section under the Big Data and Analytics - summary of changes section.
 - The cluster must be at management pack version 2.7.0.0 or later.
 - Upgrading MPack does not affect the IBM Spectrum Scale file system.

- You do not need to unconfigure HDFS HA when you migrate the services for IBM Spectrum Scale.
- Ensure that the anonymous user id is created and have the same uid/gid in your cluster before upgrading. From Mpack 2.4.2.6, having an anonymous user id is mandatory. For more information, see the “Create the anonymous user id” on page 138 topic.

Procedure

1. As the root user, download a management pack at a higher PTF version, than the version of IBM Spectrum Scale service installed on your system, onto a directory on the Ambari server node. For information on downloading the management packs, see the “IBM Spectrum Scale service” on page 253 topic.

Note: The downloaded management pack should be stored and unzipped in a directory different than the currently installed version of the Mpack.

In this example, the downloaded management pack has been downloaded in the /root/GPFS_Ambari/upgrade_Mpack directory. The management pack contains the upgrade script to upgrade the MPack.

For example, if the currently installed Mpack is at 2.7.0.0 version then plan to upgrade to Mpack 2.7.0.1 version.

The **SpectrumScale_UpgradeIntegrationPackage** script used for upgrade and migration is run from the /root/GPFS_Ambari/upgrade_Mpack directory.

Ensure that the current Mpack installable package resides on a separate directory on the Ambari server node. This example uses the /root/GPFS_Ambari/currently_installed_Mpack directory.

The **SpectrumScaleMPackUninstaller.py** script used as part of this procedure would have to be run from the /root/GPFS_Ambari/currently_installed_Mpack directory.

2. Log in to Ambari.
3. Stop all the services. Click **Ambari > Actions > Stop All**¹.

¹For FPO cluster, do not run STOP ALL from the Ambari GUI. Refer to the Limitations > General sections on how to properly stop IBM Spectrum Scale.

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. As the root user on the Ambari server node, from the /root/GPFS_Ambari/upgrade_Mpack directory, run the **SpectrumScale_UpgradeIntegrationPackage** script with the **--preEU** option.

The **--preEU** option saves the existing IBM Spectrum Scale service information into JSON files in the local directory where the script was run. It also removes the IBM Spectrum Scale service from the Ambari cluster so that the BI cluster can be properly migrated. This does not affect the IBM Spectrum Scale file system.

Before you proceed, review the following questions for the upgrade script and have the information for your environment handy. If Kerberos is enabled, more inputs are required.

```
$ cd /root/GPFS_Ambari/upgrade_Mpack
$ ./SpectrumScale_UpgradeIntegrationPackage --preEU
Are you sure you want to upgrade the GPFS Ambari integration package (Y/N)? (Default Y):
*****
***STARTING WITH SPECTRUM SCALE EXPRESS UPGRADE PRE STEPS***
*****
Enter the Ambari server User:(Default admin ):
Enter the password for the Ambari server.
Password:
Retype password:
SSL Enabled (True/False) (Default False):
Enter the Ambari server Port. (Default 8080):
```

```

| ...
| # Note: If Kerberos is enabled, then the KDC principal and password information are required.
| Kerberos is Enabled. Proceeding with Configuration
| Enter kdc principal:
| Enter kdc password:
| ...
| 7. As a root user on the Ambari server, run the MPack uninstaller script,
| SpectrumScaleMPackUninstaller.py, from the currently installed Mpack directory, to remove the
| existing MPack link in Ambari.
| $ cd /root/GPFS_Ambari/currently_installed_Mpack
| $./SpectrumScaleMPackUninstaller.py
| INFO: ***Starting the MPack Uninstaller***

| Enter Ambari Server Port Number. If it is not entered, the uninstaller will take default port 8080:
| INFO: Taking default port 8080 as Ambari Server Port Number.
| Enter Ambari Server IP Address : 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. 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.

| 9. On the Ambari server node as root, from the /root/GPFS_Ambari/upgrade_Mpack directory, run the
|    SpectrumScale_UpgradeIntegrationPackage script with the --postEU option in the directory where
|    the --preEU step was run and where the JSON configurations were stored.

| Before you proceed, for the --postEU option, review the following questions and have the
| information for your environment handy. If Kerberos is enabled, more inputs are required.

| $ cd /root/GPFS_Ambari/upgrade_Mpack
| $ ./SpectrumScale_UpgradeIntegrationPackage --postEU
| Are you sure you want to upgrade the GPFS Ambari integration package (Y/N)? (Default Y):
| ****
| ***STARTING WITH SPECTRUM SCALE EXPRESS UPGRADE POST STEPS***
| ****
| Starting Post Express Upgrade Steps. Enter Credentials
| Enter the Ambari server User:(Default admin ):
| Enter the password for the Ambari server.
| Password:
| Retype password:
| SSL Enabled (True/False) (Default False):
| Enter the Ambari server Port. (Default 8080):
| ....
| # Accept License
| Do you agree to the above license terms? [yes or no]
| yes
| Installing...
| Enter Ambari Server Port Number. If it is not entered, the installer will take default port 8080 :
```

```
| INFO: Taking default port 8080 as Ambari Server Port Number.  
| Enter Ambari Server IP Address :  
| 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.  
| *****
```

Note: If the Mpack requires a corresponding HDFS Transparency update version, ensure that the process in the “Upgrading HDFS Transparency” on page 280 topic is done before doing a **Start All** in the next step.

10. Start all the services.

Click **Ambari** > **Actions** > **Start All**.

| Restart all the components by using the restart icon.

Note:

- If the **Start All** fails, try starting each of the services individually. Ensure that the manual starting services in the Ambari order is executed first. For more information, see the “Manually starting services in Ambari” on page 289 topic.
 - If the IBM Spectrum Scale service is restarted by using the restart icon, the HDFS service also needs to be restarted.
 - The NameNode Last Checkpoint alert can be ignored and can be disabled.
 - If the HBase master failed to start with `FileNotFoundException` 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.

- Go to the IBM Spectrum Scale yum directory, and rebuild the yum database by running the **createrepo** command.

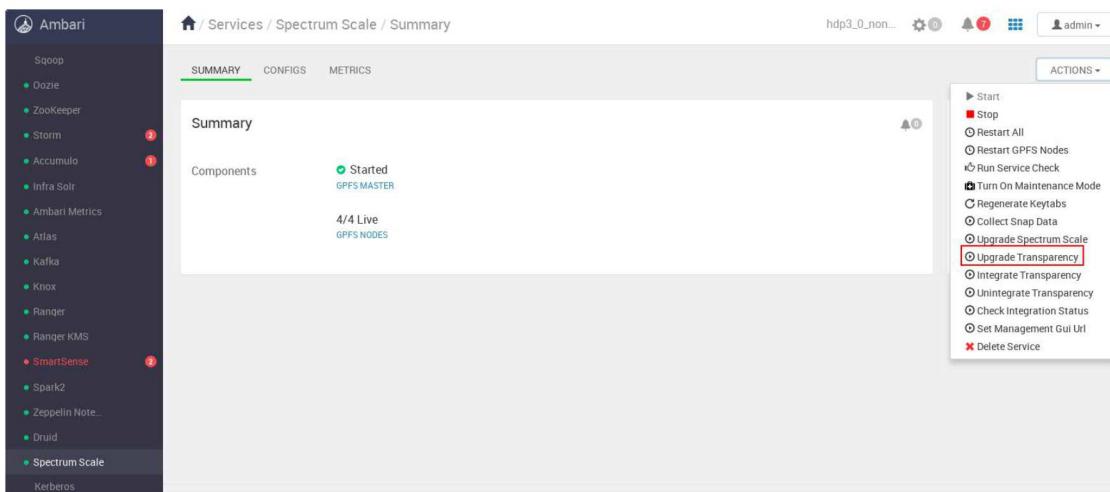
```
$ cd /var/www/html/repos/GPFS/<Scale_version>/gpfs_rpms
$ createrepo .
Spawning worker 0 with 2 pkgs
Spawning worker 1 with 2 pkgs
Spawning worker 2 with 2 pkgs
Spawning worker 3 with 2 pkgs
Workers Finished
Saving Primary metadata
Saving file lists metadata
Saving other metadata
Generating sqlite DBs
Sqlite DBs complete
```

- From the dashboard, select **Actions > Stop All**¹ to stop all the services.

¹For FPO cluster, do not run STOP ALL from the Ambari GUI. Refer to the Limitations > General sections on how to properly stop IBM Spectrum Scale.

Note: To upgrade the HDFS Transparency, the IBM Spectrum Scale file system does not need to be stopped. If you do not want to stop the IBM Spectrum Scale file system, do not select **Actions > Stop All**. Instead, stop all the services individually by going into each service panel, and clicking **Actions > Stop** for all, except the IBM Spectrum Scale service.

- From the dashboard, click **Spectrum Scale > 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-3.0.0-0.x86_64
```

If the HDFS Transparency version is not correct on a specific node, then manually install the correct version onto that node.

To manually install the HDFS Transparency on a specific node:

- 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-3.0.0.0.<OS>.rpm
```

- | 6. On the dashboard, click **Actions > Start All**.
- | 7. 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.

| To upgrade to a newer version of IBM Spectrum Scale™, consider the version that you are upgrading from and then consider co-existence and compatibility issues. See the *Upgrading and IBM Spectrum Scale supported upgrade paths* topics in the *IBM Spectrum Scale: Concepts, Planning, and Installation Guide*.

| You can update the IBM Spectrum Scale packages through the Ambari GUI. This function will upgrade the IBM Spectrum Scale packages and build the GPFS portability layer to all the Ambari GPFS nodes. The packages will follow the same rules as specified in Local IBM Spectrum Scale repository.

| **Note:** Only offline upgrade of IBM Spectrum Scale is supported through this Ambari interface.

| Upgrading the IBM Spectrum Scale file system package and Upgrading HDFS Transparency are done separately.

| You can get the PTF packages from IBM Fix Central and extract the packages as stated in the README file.

| You must put all the update packages (PTF) into a yum repository. If the Yum repository is not the existing IBM Spectrum Scale yum repository path specified in Ambari, add the yum repository URL to Ambari Spectrum Scale configuration. For information on updating the yum repository, see GPFS yum repo directory.

| **Note:** Only root Ambari installation can upgrade the IBM Spectrum Scale function through Ambari. Under non-root Ambari installation, IBM Spectrum Scale file system must be upgraded manually as stated in the README file for the specific PTF package in Fix Central. Ensure that all services are stopped, and that no processes are accessing the file system before proceeding to do the upgrade.

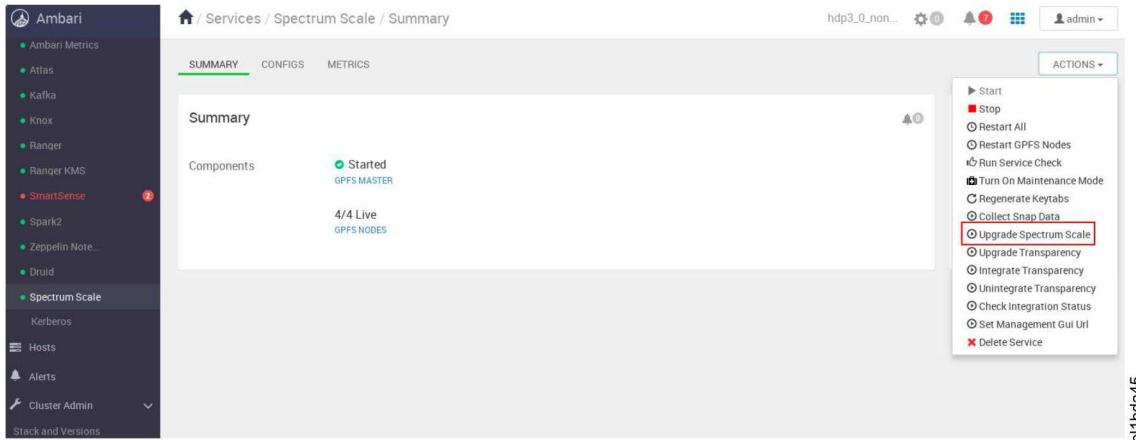
- | 1. Go to the IBM Spectrum Scale yum directory and rebuild the yum database by using the **createrepo** command.

```
$ createrepo .
Spawning worker 0 with 4 pkgs
Spawning worker 1 with 4 pkgs
Spawning worker 2 with 4 pkgs
Spawning worker 3 with 4 pkgs
Workers Finished
Saving Primary metadata
Saving file lists metadata
Saving other metadata
Generating sqlite DBs
Sqlite DBs complete
```

- | 2. From the dashboard, select **Actions > Stop All**¹ to stop all services.

| ¹For FPO cluster, do not run STOP ALL from the Ambari GUI. Refer to the Limitations > General sections on how to properly stop IBM Spectrum Scale.

- | 3. From the dashboard, select **Spectrum Scale > Actions > Upgrade Spectrum Scale**.



bl1bda45

- 4. Verify that the selected IBM Spectrum Scale PTF packages are installed on the nodes.

```
$ xdsh c902f05[x01-x04] "rpm -qa | grep gpfs"
```

Note: Check that the gpfs version installed on the nodes are the updated ones.

- 5. From the dashboard, select **Actions > Start All**

Note: IBM Spectrum Scale starts with the latest PTF packages. Verify that HDFS Transparency Namenode and Datanodes are functioning.

```
$ /usr/lpp/mmfs/hadoop/sbin/mmhadoopctl connector getstate
c902f05x01.gpfs.net: namenode running as process 18150.
c902f05x01.gpfs.net: datanode running as process 22958.
c902f05x04.gpfs.net: datanode running as process 15560.
c902f05x03.gpfs.net: datanode running as process 17275.
c902f05x02.gpfs.net: datanode running as process 26416.
```

Note: After all nodes are upgraded to the new level of IBM Spectrum Scale, use the cluster for a while with the new level installed. Then follow the *Completing the upgrade to a new level of IBM Spectrum Scale to finalize the upgrade* topic in the *IBM Spectrum Scale: Concepts, Planning, and Installation Guide*.

Upgrading from HDP 2.6.x to HDP 3.0.0

This section describes how to upgrade your HDP 2.6.x environment with IBM Spectrum Scale service to HDP 3.0. This involves upgrading the Ambari Server as well.

You must plan a cluster maintenance window and prepare for cluster downtime during the upgrade.

Note:

- HDFS Transparency 3.0.0-0 and IBM Spectrum Scale Mpack version 2.7.0.0 are required.
- 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.
- Ensure that an anonymous user id is created and has the same uid/gid in your cluster before upgrading.

Procedure

1. Download the management pack version 2.7.0.0 as root to a directory on the Ambari server node. This example uses the /root/GPFS_Ambari/upgrade_Mpack directory. Unzip the management pack into the /root/GPFS_Ambari/upgrade_Mpack directory.

Note: The downloaded management pack should be stored and unzipped in a directory different than the currently installed version of the Mpack.

The **SpectrumScale_UpgradeIntegrationPackage** script used for upgrade and migration is run from the `/root/GPFS_Ambari/upgrade_Mpack` directory.

Ensure that the current Mpack installable package (Mpack version 2.4.2.X) resides on a separate directory on the Ambari server node. This example uses the `/root/GPFS_Ambari/currently_installed_Mpack` directory.

The **SpectrumScaleMPackUninstaller.py** script used as part of this procedure would have to be run from the `/root/GPFS_Ambari/currently_installed_Mpack` directory.

2. Log in to Ambari and stop all the services of your HDP 2.6.x cluster with Spectrum Scale. Click **Ambari > Actions > Stop All**¹

If the IBM Spectrum Scale service is not already stopped, stop the IBM Spectrum Scale service by clicking **Ambari > Spectrum Scale > Actions > Stop**¹.

3. After all the services have stopped, unintegrate the transparency. Follow the steps in “Unintegrating HDFS Transparency” on page 203 and ensure that the **ambari-server restart** is run.

Note: Do not start any services yet.

4. HDP 3.0.0 requires HDFS Transparency version 3.0.0-0. If you are upgrading from HDP 2.6.x, add the HDFS Transparency version 3.0.0-0 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.

5. Update the HDFS Transparency package to all the GPFS nodes.

From Ambari GUI, go to **Upgrade Transparency service** in the Spectrum Scale service UI window to propagate the new package to all the GPFS Nodes. For more information, see “Upgrading HDFS Transparency” on page 161.

6. 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 Ambari so that the HDP cluster can be properly upgraded. 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/upgrade_Mpack
$ ./SpectrumScale_UpgradeIntegrationPackage --preEU
Are you sure you want to upgrade the GPFS Ambari integration package (Y/N)? (Default Y):
*****
***STARTING WITH SPECTRUM SCALE EXPRESS UPGRADE PRE STEPS***
*****
Enter the Ambari server User:(Default admin ):
Enter the password for the Ambari server.
Password:
Retype password:
SSL Enabled (True/False) (Default False):
Enter the Ambari server Port. (Default 8080):
...
# Note: If Kerberos is enabled, then the KDC principal and password information are required.
Kerberos is Enabled. Proceeding with Configuration
Enter kdc principal:
Enter kdc password:
...
```

7. On the Ambari server node, as root, uninstall the existing IBM Spectrum Scale MPack from the currently installed Mpack directory by running the following command:

```

cd /root/GPFS_Ambari/currently_installed_Mpack
$ ./SpectrumScaleMPackUninstaller.py
INFO: ***Starting the MPack Uninstaller***
Enter Ambari Server Port Number. If it is not entered, the uninstaller will take default port 8080:
INFO: Taking default port 8080 as Ambari Server Port Number.
Enter Ambari Server IP Address : 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 in Ambari. Click Ambari > Actions > Start All.
Wait for all the services to start. At this stage, native HDFS is used.

9. Follow the Hortonworks upgrade documentation.
For x86 upgrade, see Apache Ambari Upgrade.
For power upgrade, see Apache Ambari Upgrade for IBM Power Systems.

Note: Ensure that you upgrade the Ambari server and the Ambari agents before upgrading the HDP cluster to HDP 3.0.0.

10. After HDP is successfully upgraded to HDP 3.0.0, stop all services.
Click Ambari > Actions > Stop All1.
Wait until all services have stopped. Ensure that the native HDFS has stopped running.

11. 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.
Before you proceed, for the --postEU option, review the following questions, and have the information for your environment handy. If Kerberos is enabled, more inputs are required.

$ cd /root/GPFS_Ambari/upgrade_Mpack
$ ./SpectrumScale_UpgradeIntegrationPackage --postEU
Are you sure you want to upgrade the GPFS Ambari integration package (Y/N)? (Default Y):
*****
***STARTING WITH SPECTRUM SCALE EXPRESS UPGRADE POST STEPS***
*****
Starting Post Express Upgrade Steps. Enter Credentials
Enter the Ambari server User:(Default admin ):
Enter the password for the Ambari server.
Password:
Retype password:
SSL Enabled (True/False) (Default False):
Enter the Ambari server Port. (Default 8080):
....
# Accept License
Do you agree to the above license terms? [yes or no]
yes
Installing...
Enter Ambari Server Port Number. If it is not entered, the installer will take default port 8080:
INFO: Taking default port 8080 as Ambari Server Port Number.
Enter Ambari Server IP Address :
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.  
| *****  
| Upgrade of the Spectrum Scale Service completed successfully.  
| *****
```

| 12. Start all services.

| Click **Ambari > Actions > Start All**.

| Restart all components where the Restart icon Alert is shown.

| **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.

| ¹For FPO cluster, do not run STOP ALL from the Ambari GUI. Refer to the Limitations > General sections on how to properly stop IBM Spectrum Scale.

| **Uninstalling IBM Spectrum Scale Mpack and service**

| Before you upgrade Mpack on an HDP cluster, see “Preparing the environment” on page 137 section to check whether the new Mpack upgrade supports the HDP version installed on the cluster.

| This topic lists the steps to uninstall IBM Spectrum Scale Mpack and service.

| 1. Follow the steps in “Unintegrating HDFS Transparency” on page 203 and ensure that the ambari-server restart is run.

| 2. Uninstalling the management pack and the IBM Spectrum Scale service

| On the Ambari server host, as root, execute the following script:

```
| $ python SpectrumScaleMPackUninstaller.py
```

| Follow the script and enter the required details.

| The SpectrumScaleMPackUninstaller.py script is in the download package along with the Mpack and license bin executables.

| The SpectrumScaleMPackUninstaller.py uninstalls the IBM Spectrum Scale Mpack and the IBM Spectrum Scale service from Ambari. It verifies the settings before it uninstalls the Mpack and IBM Spectrum Scale service. If the services are still running, and if the HDFS Transparency is not unintegrated, the SpectrumScaleMPackUninstaller.py script will exit and request user action.

| **Note:** Running this script does not remove the IBM Spectrum Scale packages and disks. The IBM Spectrum Scale file system is preserved as is. For the FPO cluster created through Ambari, the mounted local disks /opt/mapred/local* and entries in /etc/fstab are preserved as is.

| **Configuration**

| **Setting up High Availability [HA]**

| This section contains information on how to setup high availability to protect against planned and unplanned events.

| **HDFS Namenode High Availability [HA]**

| This process sets up a standby Namenode configuration so that failover can happen automatically.

- In order to setup HA, IBM Spectrum Scale HDFS Transparency has to be unintegrated and revert back to native HDFS mode.
 - Follow these steps to configure High Availability option when IBM Spectrum Scale™ service is integrated:
 - Log into the Ambari GUI.
 - If the Ambari GUI has IBM Spectrum Scale service deployed, and HDFS Transparency is integrated, follow the steps to “Integrating HDFS transparency” on page 202.
 - Ensure that you run the **ambari-server restart**.
 - Verify that the IBM Spectrum Scale HDFS Transparency integration state is in unintegrated state. See Verifying Transparency integration state.
 - From the Ambari dashboard, click the HDFS service.
 - Select **Actions > Enable NameNode HA** and follow the steps.

Note: Namenode needs to be deployed onto a GPFS™ Node.
 - If the Ambari GUI has IBM Spectrum Scale service deployed, follow the steps under Integrating HDFS Transparency to integrate HDFS Transparency.
 - Ensure that you run the **ambari-server restart**.
 - For more information on setting up the Namenode HA, see Hortonworks Configuring Namenode HA.
- ### **Yarn Resource Manager HA**
- This process sets up a Resource Manager configuration so that failover can happen automatically.
 - This topic describes how to enable ResourceManager High Availability if Spectrum Scale service is already integrated with HDP.
 - Turn on Maintenance Mode on Spectrum Scale. Click **Ambari Spectrum Scale service > Actions > Turn On Maintenance Mode**. This will ensure that IBM Spectrum Scale is not stopped while Resource Manager HA is enabled in the next step.
 - Enable Resource Manager HA as per Configuring ResourceManager High Availability
 - Turn off Maintenance Mode on IBM Spectrum Scale service.
 - Restart HDFS service.

IBM Spectrum Scale configuration parameter checklist

- The IBM Spectrum Scale checklist shows the parameters that affect the system, the Standard, and Advanced tabs in the Ambari wizard.
- These are the important IBM Spectrum Scale parameters checklists:

| Standard tab | Rule | Advanced tab | Rule |
|------------------------|--|---|---|
| Cluster Name | | Advanced core-site:
fs.defaultFS | Ensure that
hdfs://localhost:8020 is
used |
| FileSystem Name | | Advanced gpfs-advance:
gpfs.quorum.nodes | The node number must be
odd.
Note: In SAN storage
deployment model with
tiebreaker disk, quorum
nodes can be even |
| FileSystem Mount Point | | | |
| NSD stanza file | See “Preparing for FPO
environment” on page 144 | | |

| Standard tab | Rule | Advanced tab | Rule |
|-------------------------------------|--|--------------|------|
| Policy file | See Policy File | | |
| Hadoop local cache disk stanza file | For more information, see “Deploy HDP or IBM Spectrum Scale service on pre-existing IBM Spectrum Scale file system” on page 177. | | |
| Default Metadata Replicas | <= Max Metadata Replicas | | |
| Default Data Replicas | <= Max Data Replicas | | |
| Max Metadata Replicas | | | |
| Max Data Replicas | | | |

HDFS and Spectrum Scale filesystem ACL support

- HDFS and IBM Spectrum Scale HDFS Transparency support only POSIX ACL.
- Ensure that you check the ACL value on the IBM Spectrum Scale file system and set it to all.
- Otherwise, the Hadoop command will fail with Operation not supported: but POSIX command can be executed successfully.

To check the ACL value, run:

```
# /usr/lpp/mmfs/bin/mmlsfs <filesystem> -k
flag      value          description
-----  -----
-k        nfs4           ACL semantics in effect
```

To change the ACL value to all, run:

```
# /usr/lpp/mmfs/bin/mmchfs <filesystem> -k all
```

Check the ACL value after modification, run:

```
# /usr/lpp/mmfs/bin/mmlsfs <filesystem> -k
flag      value          description
-----  -----
-k        all            ACL semantics in effect
```

Dual-network deployment

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 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 information on setting up the HDFS service Quicklinks Namenode UI access, see 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`

| For example:

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

| Local OS repository

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

| 1. Create the repository path:

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

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

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

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

| 3. Run the following command:

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

| 4. Create the repository for this node:

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

| 5. Ensure that all the firewalls are disabled or that you have the httpd service port open, because yum uses http to get the packages from the repository.

| 6. On all nodes in the cluster that require the repositories, create a file in /etc/yum.repos.d called `local_<rhel_OSlevel>.repo`.

| 7. Copy this file to all nodes. The contents of this file must look like the following:

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

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

| Local IBM Spectrum Scale repository

| The following list of rpm packages for IBM Spectrum Scale v4.1.1 and later can help verify the edition of IBM Spectrum Scale:

| IBM Spectrum Scale Edition | rpm package list |
|---|---|
| Data Management available from version 4.2.3 and later. | This edition provides identical functionality as IBM Spectrum Scale Advanced Edition under capacity-based licensing. For more information, see the <i>Capacity-based licensing</i> topic in the <i>IBM Spectrum Scale: Concepts, Planning, and Installation Guide</i> . |
| Standard Edition | <Express Edition rpm list> + gpfs.ext |

| IBM Spectrum Scale Edition | rpm package list |
|----------------------------|---|
| Advanced Edition | <Standard Edition rpm list> + gpfs.crypto
For IBM Spectrum Scale 4.2 release and later: Add gpfs.adv to the list above |

| **Note:** From IBM Spectrum Scale 5.0.2, the gpfs.ext package is not available.

| The following example uses IBM Spectrum Scale 5.0.1.1 version.

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

```
mkdir -p /var/www/html/repos/GPFS/5.0.1.1
```
- | 2. Obtain the IBM Spectrum Scale software. If you have already installed IBM Spectrum Scale manually, skip this step. Download the IBM Spectrum Scale package. In this example, IBM Spectrum Scale 5.0.1.1 is downloaded from Fix Central, the package is unzipped, and the installer is extracted.
For example, As root or a user with sudo privileges, run the installer to get the IBM Spectrum Scale packages into a user-specified directory via the --dir option:

```
chmod +x Spectrum_Scale_Advanced-5.0.1.1-x86_64-Linux-install
```

```
./Spectrum_Scale_Advanced-5.0.1.1-x86_64-Linux-install --silent --dir /var/www/html/repos/GPFS/5.0.1.1
```

| **Note:** The --silent option is used to accept the software license agreement, and the --dir option places the IBM Spectrum Scale rpms into the directory /var/www/html/repos/GPFS/5.0.1.1/gpfs_rpms. Without specifying the --dir option, the default location is /usr/lpp/mmfs/gpfs_rpms/5.0.1.1/gpfs_rpms.

- | 3. If the packages are extracted into the IBM Spectrum Scale default directory, /usr/lpp/mmfs/5.0.1.1/gpfs_rpms, copy all the IBM Spectrum Scale files that are required for your installation environment into the IBM Spectrum Scale repository path:

```
cd /usr/lpp/mmfs/5.0.1.1/gpfs_rpms
```

```
cp -R * /var/www/html/repos/GPFS/5.0.1.1/gpfs_rpms
```

- | 4. The following packages will not be installed by Ambari:

- gpfs.crypto
- gpfs.gui
- gpfs.scalemgmt
- gpfs.tct

| Ambari requires only the following packages:

- gpfs.base
- gpfs.gpl
- gpfs.docs
- gpfs.gskit
- gpfs.msg.en_US
- gpfs.ext (Not available from IBM Spectrum Scale 5.0.2 version)
- gpfs.crypto (if Advanced edition is used)
- gpfs.adv (if IBM Spectrum Scale 4.2 Advanced edition is used)

| The IBM Spectrum Scale repo will not install the protocol and transparent cloud tier (gpfs.tct) packages when installing through Ambari.

- | 5. Copy the HDFS Transparency package to the IBM Spectrum Scale repo path.

- | **Note:** The repo must contain only one HDFS Transparency package. Remove all old transparency packages.
- | cp gpfs.hdfs-protocol-3.0.0-(version) /var/www/html/repos/GPFS/5.0.1.1/gpfs_rpms
- | 6. Check for the IBM Spectrum Scale packages in the /root/ directory. If the package exists, relocate them to a subdirectory. There are known issues with IBM Spectrum Scale package in the /root that cause the Ambari installation to fail.
- | 7. Create the yum repository:
- | # cd /var/www/html/repos/GPFS/5.0.1.1/gpfs_rpms
- | # createrepo .
- | 8. Access the repository at http://<yum-server>/repos/GPFS/5.0.1.1/gpfs_rpms.

| MySQL community edition repository

- | If you are using the new database option for Hive MetaStore through HDP, HDP will create MySQL community edition repositories on the Hive Metastore host which will require internet access to download.
- | On a host with internet access, use the repo information to obtain a local copy of the packages to create a local repository.
- | If you have a local MySQL repo, create the mysql-community.repo file to point to the local repo on the Hive Metastore host.
- | For example:


```
[mysql56-community]
name=MySQL 5.6 Community Server
baseurl=http://<REPO_HOST>/repos/MySQL_community
enabled=1
gpgcheck=0
```
- | Only the following MySQL packages are required for HDP:


```
mysql-community-libs
mysql-community-common
mysql-community-client
mysql-community-server
```
- | HDP creates the following MySQL community repos:
 - | HDP Power®: Creates 1 repo file
 - | mysql-community.repo:


```
# Enable to use MySQL 5.6
[mysql56-community]
name=MySQL 5.6 Community Server
baseurl=http://s3.amazonaws.com/dev.hortonworks.com/
HDP-UTILS-1.1.0.21/repos/mysql-ppc64le/
enabled=1
gpgcheck=0
gpgkey=file:/etc/pki/rpm-gpg/RPM-GPG-KEY-mysql
```
 - | HDP x86: Creates 2 repo files
 - | mysql-community.repo:

```

[mysql-connectors-community]
name=MySQL Connectors Community
baseurl=http://repo.mysql.com/yum/mysql-connectors-community/el/7/$basearch/
enabled=1
gpgcheck=1
gpgkey=file:/etc/pki/rpm-gpg/RPM-GPG-KEY-mysql

[mysql-tools-community]
name=MySQL Tools Community
baseurl=http://repo.mysql.com/yum/mysql-tools-community/el/7/$basearch/
enabled=1
gpgcheck=1
gpgkey=file:/etc/pki/rpm-gpg/RPM-GPG-KEY-mysql

# Enable to use MySQL 5.5
[mysql55-community]
name=MySQL 5.5 Community Server
baseurl=http://repo.mysql.com/yum/mysql-5.5-community/el/7/$basearch/
enabled=0
gpgcheck=1
gpgkey=file:/etc/pki/rpm-gpg/RPM-GPG-KEY-mysql

# Enable to use MySQL 5.6
[mysql56-community]
name=MySQL 5.6 Community Server
baseurl=http://repo.mysql.com/yum/mysql-5.6-community/el/7/$basearch/
enabled=1
gpgcheck=1
gpgkey=file:/etc/pki/rpm-gpg/RPM-GPG-KEY-mysql

# Note: MySQL 5.7 is currently in development. For use at your own risk.
# Please read with sub pages: https://dev.mysql.com/doc/relnotes/mysql/5.7/en/
[mysql57-community-dmr]
name=MySQL 5.7 Community Server Development Milestone Release
baseurl=http://repo.mysql.com/yum/mysql-5.7-community/el/7/$basearch/
enabled=0
gpgcheck=1
gpgkey=file:/etc/pki/rpm-gpg/RPM-GPG-KEY-mysql

```

mysql-community-source.repo:

```

[mysql-connectors-community-source]
name=MySQL Connectors Community - Source
baseurl=http://repo.mysql.com/yum/mysql-connectors-community/el/7/SRPMS
enabled=0
gpgcheck=1
gpgkey=file:/etc/pki/rpm-gpg/RPM-GPG-KEY-mysql

[mysql-tools-community-source]
name=MySQL Tools Community - Source
baseurl=http://repo.mysql.com/yum/mysql-tools-community/el/7/SRPMS
enabled=0
gpgcheck=1
gpgkey=file:/etc/pki/rpm-gpg/RPM-GPG-KEY-mysql

[mysql55-community-source]
name=MySQL 5.5 Community Server - Source
baseurl=http://repo.mysql.com/yum/mysql-5.5-community/el/7/SRPMS
enabled=0
gpgcheck=1
gpgkey=file:/etc/pki/rpm-gpg/RPM-GPG-KEY-mysql

[mysql56-community-source]
name=MySQL 5.6 Community Server - Source
baseurl=http://repo.mysql.com/yum/mysql-5.6-community/el/7/SRPMS
enabled=0
gpgcheck=1
gpgkey=file:/etc/pki/rpm-gpg/RPM-GPG-KEY-mysql

[mysql57-community-dmr-source]
name=MySQL 5.7 Community Server Development Milestone Release - Source
baseurl=http://repo.mysql.com/yum/mysql-5.7-community/el/7/SRPMS
enabled=0
gpgcheck=1
gpgkey=file:/etc/pki/rpm-gpg/RPM-GPG-KEY-mysql

```

Configuring LogSearch

To setup LogSearch when IBM Spectrum Scale is integrated in an HDP environment, configuration changes are required to point to the correct NameNode, DataNode and ZKFC logs associated with IBM Spectrum Scale.

1. Click **Ambari GUI > Log Search service > Quick Links > Log Search UI.**
2. Login to **Log Search UI** and click on the top right corner button. Choose the **Configuration Editor** option.

3. In the **Configuration Editor** page, change the log path for the following components:

- a. hdfs_datanode
- b. hdfs_namenode
- c. hdfs_zkfc

Example (existing) entry for namenode:

```
"type": "hdfs_namenode",
"rowtype": "service",
"path": "/var/log/hadoop/hdfs/hadoop-hdfs-namenode-*.log"
```

change to:

```
"type": "hdfs_namenode",
"rowtype": "service",
"path": "/var/log/hadoop/root/hadoop-root-namenode-*.log"
```

Make similar changes for datanode and zkfc.

4. Restart HDFS service.

| **Hadoop Kafka/Zookeeper and IBM Spectrum Scale Kafka/Zookeeper**

- | IBM Spectrum Scale has new audit logging capability that uses its own libraries for Kafka and zookeeper.
- | To use the IBM Spectrum Scale Kafka and Zookeeper, and Hadoop Kafka and Zookeeper from HDP, you must install the Kafka and Zookeeper on different nodes for IBM Spectrum Scale and Hadoop.
- | In a new environment, IBM Spectrum Scale FPO is installed by the IBM Spectrum Scale Ambari Mpack.
- | The packages for Kafka and zookeeper can be installed when HDP is installed or after the Mpack is installed. If it is required to install IBM Spectrum Scale audit logging, follow the IBM Spectrum Scale documentation to install the Kafka and Zookeeper on different nodes other than the ones installed for the HDP Hadoop cluster.
- | When you are installing HDP, if IBM Spectrum Scale is already installed with audit logging, check where the zookeeper and Kafka are to be installed. Ensure that the HDP zookeeper and Kafka are on nodes that are not where the IBM Spectrum Scale Kafka and zookeeper reside. If needed, change to different hosts in Ambari.

| **Create Hadoop local directories in IBM Spectrum Scale**

- | If you did not create local disks to host the Yarn and Mapreduce local temporary files, and want to use the existing IBM Spectrum Scale file system, then you need to create directories under IBM Spectrum Scale file system for each node.
- | **Note:** Performance might be impacted if the local disks are not used.
- | Recommendation: Create two partitions, one for local directories and one for IBM Spectrum Scale.
- | If you need to use the IBM Spectrum Scale file system, then create a fileset within IBM Spectrum Scale to host the local directories.
- | This section details the steps for creating a fileset within IBM Spectrum Scale to host the local directories:
 - | 1. Ensure that IBM Spectrum Scale is active and mounted.
 - | Start the IBM Spectrum Scale cluster on the console of any one node in the IBM Spectrum Scale cluster, by running the following command:

```
/usr/lpp/mmfs/bin/mmstartup -a
```
 - | Mount the file system over all nodes by running the following command:

```
/usr/lpp/mmfs/bin/mount <fs-name> -a
```
 - | 2. Create a fileset in IBM Spectrum Scale and set a policy to use only one replica:

```
# Create a GPFS fileset
$ mkdir /bigpfs/hadoop
$ export PATH=$PATH:/usr/lpp/mmfs/bin
$ mmcrfileset bigpfs local
$ mmlinkfileset bigpfs local -J /bigpfs/hadoop/local

# Create policy file
$ vi hadoop.policy
rule 'fset_local' set pool 'datapool' REPLICATE (1,1) FOR FILESET (local)
rule 'default' set pool 'datapool'
$ mmchpolicy bigpfs hadoop.policy

# Verify fileset
$ cd /bigpfs/hadoop/local
$ dd if=/dev/zero of=log bs=1M count=100
$ mm1sattr -d -L log
# Verify the output for data replication is 1
```

- | 3. Create local directories for each host using the host name as the directory name for simplicity, and then change the permission. Run the following command to create the local directory from one of the IBM Spectrum Scale node.


```

      | for host in <your host name list>
      | do
      |   echo "$host"
      |   mkdir -p /bigpfs/hadoop/local/$host
      | done
      | chown -R yarn:hadoop /bigpfs/hadoop/local
      | chmod -R 755 /bigpfs/hadoop/local/*
      
```
- | 4. For each node, link a local directory to its corresponding node directory named with its host name:


```

      | for host in <your host name list>
      | do
      |   echo "$host"
      |   mmdsh -N $host "ln -s /bigpfs/hadoop/local/$host /hadoop/yarn/local"
      |   mmdsh -N $host "chown -R yarn:hadoop /hadoop/yarn/local"

      // If additional user created directories are configured under Yarn in the shared storage,
      // then ensure to create the corresponding user created directories in the local host and
      // link them to the share storage directories.
      // For example: yarn.nodemanager.log-dirs is set to /hadoop/yarn/log
      // mmdsh -N $host "mkdir -p /hadoop/yarn/local/log"
      // mmdsh -N $host "chown -R yarn:hadoop /hadoop/yarn/log"
      | done
      
```
- | 5. After installation, click **Ambari GUI > Yarn > CONFIFS** to set the Yarn's configuration **yarn.nodemanager.local-dirs** as **/hadoop/yarn/local**.

Deploy HDP or IBM Spectrum Scale service on pre-existing IBM Spectrum Scale file system

- | If you one of the following configurations, you can deploy HDP with IBM Spectrum Scale service:
 - | 1. Pre-existing Spectrum Scale cluster in FPO configuration.
 - | 2. Pre-existing Spectrum Scale cluster with remote mount filesystem configuration.
 - | 3. Pre-existing Spectrum Scale cluster in which the GPFS client nodes belongs to an ESS-based Spectrum Scale cluster.
- | The steps for deployment are as follows:
 - | 1. A quorum node on the pre-existing must be selected as the IBM Spectrum Scale Master node.
 - | 2. Ensure that IBM Spectrum Scale is active and mounted.


```
[root@c902f09x13 ~]# mmgetstate -a
```

| Node number | Node name | GPFS state |
|-------------|------------|------------|
| 1 | c902f09x13 | active |
| 2 | c902f09x16 | active |
| 3 | c902f09x14 | active |
| 4 | c902f09x15 | active |

```
[root@c902f09x13 ~]# mmllsmount bigpfs -L
```

File system bigpfs is mounted on 3 nodes:
 172.16.1.59 c902f09x13
 172.16.1.61 c902f09x14
 172.16.1.63 c902f09x15
 172.16.1.64 c902f09x16
- | **Note:** Ensure that the FPO or the local Hadoop IBM Spectrum Scale cluster is set to automount on reboot by running the following command:


```
/usr/lpp/mmfs/bin/mmchfs <filesystem name> -A yes
```

3. Follow “Create HDP cluster” on page 147.
4. Follow “Install Mpack package” on page 152.
5. Follow the “Deploy the IBM Spectrum Scale service” on page 153, with the following deviations:
 - If you have not started the IBM Spectrum Scale cluster on the Ambari Assign Slaves and Clients page, click the **Previous** button to go back to **Assign Master** page in Ambari. Then start the IBM Spectrum Scale cluster, and mount the file system onto all the nodes. Go back to the Ambari GUI to continue to the **Assign Slaves and Client** page.
 - Verify that the **gpfs.storage.type** is set to
 - *local* for FPO
 - *shared* for Single cluster with all Hadoop nodes as Spectrum Scale nodes
 - *remote* for Remote mount with all Hadoop nodes as Spectrum Scale nodes
 - Verify the **yarn.nodemanager.local-dirs** and **yarn.nodemanager.log-dirs** values are set to an available mounted local partitioned directories that already exist in your file system.

For example:

```
Mounted local partitioned directories - /opt/mapred/local<NUM>
yarn.nodemanager.local-dirs=/opt/mapred/local1/yarn, /opt/mapred/local2/yarn,
/opt/mapred/local3/yarn
yarn.nodemanager.log-dirs=/opt/mapred/local1/yarn/logs, /opt/mapred/local2/yarn/logs,
/opt/mapred/local3/yarn/logs
```

 - Do not set the GPFS NSD stanza file field.

For FPO, the IBM Spectrum Scale NSD stanza file is not required because the file system already exists. Because Ambari does not allow a blank value, leave the default value of IBM Spectrum Scale NSD stanza file.

Note: If you accidentally place a value in the GPFS NSD stanza file field which was originally blank, and then try to remove it, you must leave in a “blank” character for Ambari to proceed.

 - **Single Scale cluster configuration**

For **gpfs.storage.type=shared** the local cluster hosts with GPFS components (GPFS_Master or GPFS_Node) selected in the UI, are added on to the ESS/Shared IBM Spectrum Scale cluster.

 - Setting **gpfs.storage.type=shared** for Shared storage means this will create a single scale cluster configuration.
 - Setting **gpfs.storage.type=shared** for ESS and creating the `/var/lib/ambari-server/resources/shared_gpfs_node.cfg` file on the Ambari server will create a single scale cluster configuration. The file must contain only one FQDN of a node in the shared management host cluster, and password-less SSH must be configured from the Ambari server to this node. Ambari uses this one node to join the GNR/ESS cluster. Ensure that the file has at least 444 permission.
 - [Optional] To create local cache disks, see **Deploy the IBM Spectrum Scale service>>Customize Services>Create Hadoop local cache disks** section.

Note: If you are not using shared storage, you do not need this configuration, and you can leave this local cache disk parameter unchanged in the Ambari GUI.

 - Verify the following fields have the correct information that match your preinstalled IBM Spectrum Scale file system (GPFS) cluster.
 - GPFS cluster name
 - GPFS quorum nodes
 - GPFS File System Name
 - **gpfs.mnt.dir**
 - **gpfs.storage.type**

| Deploy FPO

| In File Placement Optimizer (FPO) mode, data blocks are stored in chunks in IBM Spectrum Scale, and | replicated to protect against disk and node failure. DFS clients run on the storage node so that they can | leverage the data locality for executing the tasks quickly.

| For the local storage mode configuration, “Short-circuit read (SSR)” on page 198 is recommended to | improve the access efficiency.

| **Note:** Ambari only supports creating an IBM Spectrum Scale FPO file system.

| Follow the Installation but to create a new FPO cluster, the following deviation is to be followed:

- | • Skip “ESS setup” on page 146 section
- | • Follow “Create HDP cluster” on page 147 section
- | • Skip “Establish a IBM Spectrum Scale cluster on the Hadoop cluster” on page 151 section
- | • Skip “Configure remote mount access” on page 152 section
- | • Follow “Install Mpack package” on page 152 section
- | • Follow the “Deploy the IBM Spectrum Scale service” on page 153 section with the following | deviations:

| **Under Assign Masters:**

- | • All the Yarn’s NodeManager nodes should be FPO nodes with the same number of disks for each node | specified in the NSD stanza.

| **Under Customize Services:**

- | • Configuration fields on both standard and advanced tabs are populated with values taken from the | Hadoop performance tuning guide.
- | • Verify that the **gpfs.storage.type** is set to *local*.
- | • If you do not plan to have a sub-directory under the IBM Spectrum Scale mount point, do not click on | the **gpfs.data.dir** field to preserve the field to not have any values set.
- | • Ensure the **yarn.nodemanager.local-dirs** and **yarn.nodemanager.local-logs** are set to a dummy local | directory initially. When a new FPO is deployed, partitioned local directories dynamically replace the | ones in **yarn.nodemanager.local-dirs** after the FPO system is created. Manually check to ensure that | the **yarn.nodemanager.local-logs** value is set correctly. For more information, see “Disk-partitioning | algorithm” on page 295.
- | • Create an NSD file, **gpfs_nsd**, and place it into the /var/lib/ambari-server/resources directory. Ensure | that the permission on the file is at least 444. Add the NSD filename, **gpfs_nsd**, to the GPFS File system | > GPFS NSD stanza file field in the Standard Config tab.

| Two types of NSD files are supported for file system auto creation. One is the preferred simple format | and another is the standard IBM Spectrum Scale NSD file format for IBM Spectrum Scale experts.

| **Simple NSD**

| If a simple NSD file is used, Ambari selects the proper metadata and data ratio for you. If possible, | Ambari creates partitions on some disks for the Hadoop intermediate data, which improves the Hadoop | performance. Simple NSD does not support existing partitioned disks in the cluster.

| • Disk partitioning under Ambari would happen only if the following conditions are met:

- | 1. The NSD stanza requires all GPFS nodes to be specified in the NSD stanza file.
- | 2. Each of those nodes should have the same number of disks specified in the stanza file.
- | 3. Number of host entries in the stanza file/NSD servers should be equal to the number of Node | managers. This requires all hosts running GPFS node to be set up as a Node manager too. If you

| do not want Hadoop jobs to run on a specific GPFS node host (For example, on the Ambari server host), you could remove the Node manager component from that host after deploying the IBM Spectrum™ scale service.

| For more details on disk partitioning, see the following:

- | • “Disk-partitioning algorithm” on page 185.
- | • “Partitioning function matrix in automatic deployment” on page 186.

| Standard NSD

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

| For standard IBM Spectrum Scale NSD file is used, administrators are responsible for the storage space arrangement.

| • Apply the partition algorithm.

| Apply the algorithm for system pool and usage.

| • Apply the failure group selection rule.

| Failure groups are created based on the rack location of the node.

| • Define the Rack mapping file.

| Nodes can be defined to belong to racks.

| • Partition the function matrix.

| The reason why one disk is divided into two partitions is so that one partition is used for the ext3 or ext4 to store the map or reduce intermediate data, while the other partition is used as a data disk in the IBM Spectrum Scale file system. Also, only data disks can be partitioned. Metadata disks cannot be partitioned.

| • A policy file is required when a standard IBM Spectrum Scale NSD file is used.

| A policy file, gpfs_fs.pol, must be created and placed into the /var/lib/ambari-server/resources directory. Add the policy filename, gpfs_fs.pol, into the **GPFS policy file** field in the Standard Config tab.

| 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.

| **Note:** Deploying HDP over an existing IBM Spectrum Scale FPO cluster through Ambari, requires to either store the Yarn’s intermediate data into the IBM Spectrum Scale file system, or use idle disks formatted as a local file system. It is recommended to use the idle disks formatted as a local file system. See “Deploy HDP or IBM Spectrum Scale service on pre-existing IBM Spectrum Scale file system” on page 177 for more information.

| Hadoop Storage Tiering

| When using Hadoop Storage Tiering configuration, the jobs running on the Hadoop cluster with native HDFS can read and write the data from IBM Spectrum Scale in real time.

| There would be only one copy of the data. See Hadoop Storage Tiering for more information.

| For information on viewfs, see “Open Source Apache viewfs support” on page 76.

| Limited Hadoop nodes as IBM Spectrum Scale nodes

| For any deployment model, you do not have to put all the Hadoop cluster nodes as GPFS nodes.

- 1. On all the Hadoop hosts that are either a namenode or a datanode in the native HDP cluster, assign a GPFS node so that the Transparency namenodes and datanodes are able to do a RPC in IBM Spectrum Scale.
- 2. If you need to configure additional Transparency datanodes other than the native datanodes, assign a GPFS Node on them as well.
- 3. You could also have GPFS Nodes that do not use Transparency service. For example, if you want to use a host GPFS protocol node, assign GPFS Node on that host.

Configuring multiple file system mount point access

Currently, the IBM Spectrum Scale service GUI can support only up to two file systems.

During the IBM Spectrum Scale Ambari deployment, the following fields are required for setting up the multiple file system access:

| Fields | Description |
|------------------------------|--|
| gpfs.storage.type | Type of Storage. Comma-delimited string.

The first value will be treated as the primary file system and the values after that will be treated as the secondary file systems.

Only the following combination of file system values is supported:

gpfs.storage.type=local,remote

gpfs.storage.type=remote,remote |
| gpfs.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. |

Note:

- 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 “Configure remote mount access” on page 152.
- Follow the instructions based on the type of deployment model that you have:
- Add remote mount file systems access to existing HDP and an FPO file system that was deployed by IBM Spectrum Scale Ambari service.
- Prerequisites:

- Deployed HDP.
- Deployed FPO file system via IBM Spectrum Scale service through Ambari. The Ambari server requires to be on the GPFS master node.
- Pre-existing remote mount file system.

Use the `gpfs.storage.type=local,remote` configuration setting.

On the Ambari server node on the local FPO file system:

- Stop All services.

On the Ambari UI, click **Actions > Stop All**¹ to stop all the services.

- On the owning Spectrum Scale cluster, run the `/usr/lpp/mmfs/bin/mmlsmount 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 `/remoteefs`.

Setting of the fields would be as follows:

```
gpfs.storage.type=local,remote
gpfs.mnt.dir=/localfs,/remoteefs
gpfs.replica.enforced=dfs,dfs
gpfs.data.dir=myDataDir OR gpfs.data.dir=
GPFS FileSystem Name=localfs,remoteefs
```

- Restart Spectrum Scale service.
- Restart any service with **Restart Required** icon.
- Click **Ambari > Actions > Start All** to start all the services.

2. Add remote mount file systems access to existing HDP and an IBM Spectrum Scale FPO file system that was deployed manually.

Prerequisites:

- An FPO file system that is manually created.
- Deployed HDP on the manually created FPO file system. The Ambari server requires to be on the GPFS master node.
- Pre-existing remote mount file system.

Use the `gpfs.storage.type=local,remote` configuration setting.

On the Ambari server node on the local FPO file system, perform the following:

- Stop All services.

On the Ambari UI, click **Actions > Stop All**¹ to stop all the services.

- Start 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 `/remoteefs`.

Setting of the fields would be as follows:

```
gpfs.storage.type=local,remote
gpfs.mnt.dir=/localfs,/remoteefs
gpfs.replica.enforced=dfs,dfs
gpfs.data.dir=myDataDir OR gpfs.data.dir=
GPFS FileSystem Name=localgpfs,remotegpfs
```

- Click **Ambari > Actions > Start All** to start all the services.

3. Add remote mount file systems access to existing HDP and a manually created IBM Spectrum Scale cluster.

Create the FPO file system onto the local IBM Spectrum Scale cluster.

Prerequisites:

- A manual IBM Spectrum Scale cluster is created.
- No FPO file system was created.
- Deployed HDP onto the manual IBM Spectrum Scale cluster. The Ambari server requires to be on the GPFS master node.
- Pre-existing remote mount file system.

Use **gpfs.storage.type=local,remote** configuration setting.

On the Ambari server node on the local cluster:

- Stop All services.

On the Ambari UI, click **Actions > Stop All**¹ to stop all the services.

- Start 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 `/remoteefs`.

- 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 `/remoteefs`.

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.

4. Add only the remote mount file systems access to existing HDP and a manually created IBM Spectrum Scale cluster.

Prerequisites:

- A manual IBM Spectrum Scale cluster is created.
- Deployed HDP onto the manual IBM Spectrum Scale cluster. The Ambari server node requires to be on the GPFS master node.
- Pre-existing remote mount file systems.

Use **gpfs.storage.type=remote,remote** configuration setting.

On the Ambari server node, on the local cluster:

- Stop All services.
On the Ambari UI, click **Actions > Stop All**¹ to stop all the services.
 - Start 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
```

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

¹For FPO cluster, do not run STOP ALL from the Ambari GUI. Refer to the Limitations > General sections on how to properly stop IBM Spectrum Scale.

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
```

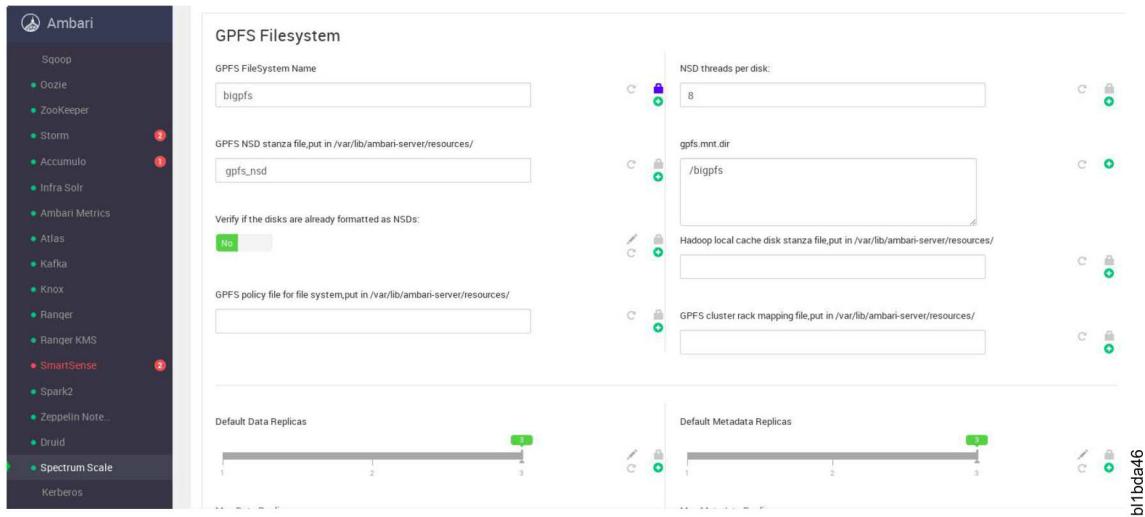


Figure 21. AMBARI RACK MAPPING

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 23. 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. |

| *Table 23. IBM Spectrum Scale partitioning function matrix (continued)*

| 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 |
|--|--|---|--|
| #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**

| **Enabling Ranger**

| This section provides instructions to enable Ranger.

- | Ranger can be configured before or after IBM Spectrum Scale service is deployed. The HDFS Transparency does not need to be in an unintegrated state.

| **Ranger Procedure:**

- | This topic lists the steps to install Ranger

- | Follow these steps to enable Ranger:

- | • Configuring MySQL for Ranger
- | • Installing Ranger through Ambari
- | • Enabling Ranger HDFS plugin
- | • Logging into Ranger UI

- | *Configuring MySQL for Ranger:*

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

- | 1. Create a non-root user to create the Ranger databases. In this example, the username *rangerdba* with password *rangerdba* is used.

- a. Log in as the root user to the DB host node. Ensure that the DB is running. This is the node that has MySQL installed, which is usually the Hive server node. Use the following commands to create the *rangerdba* user, and grant the user adequate privileges:

```
CREATE USER 'rangerdba'@'localhost' IDENTIFIED BY 'rangerdba';

GRANT ALL PRIVILEGES ON *.* TO 'rangerdba'@'localhost';

CREATE USER 'rangerdba'@'%' IDENTIFIED BY 'rangerdba';

GRANT ALL PRIVILEGES ON *.* TO 'rangerdba'@'%';

GRANT ALL PRIVILEGES ON *.* TO 'rangerdba'@'localhost' WITH GRANT OPTION;

GRANT ALL PRIVILEGES ON *.* TO 'rangerdba'@'%' WITH GRANT OPTION;
```

FLUSH PRIVILEGES;

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

- b. Reconnect to the database as user *rangerdba* by using the following command:

```
mysql -u rangerdba -prangerdba
```

- After testing the *rangerdba* login, use the **exit** command to exit MySQL.

- | 2. Check MySQL Java connector

- a. Run the following command to confirm that the mysql-connector-java.jar file is in the Java share directory. This command must be run on the Ambari server node.

```
ls /usr/share/java/mysql-connector-java.jar
```

Note: If the /usr/share/java/mysql-connector-java.jar is not found, install the mysql-connector-java package on the ambari-server node

```
$ yum install mysql-connector-java
```

- b. Use the following command to set the jdbc/driver/path based on the location of the MySQL JDBC driver .jar file. This command must be run on the Ambari server node.

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

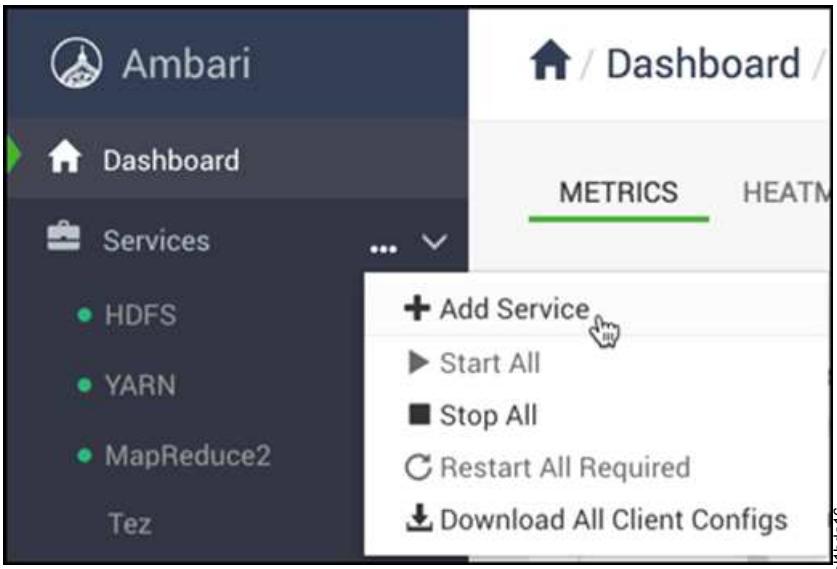
For example:

```
ambari-server setup --jdbc-db=mysql --jdbc-driver=/usr/share/java/mysql-connector-java.jar
```

| *Installing Ranger through Ambari:*

| This topic lists the steps to install Ranger through Ambari.

- | 1. Log in to Ambari UI.
| 2. Add the Ranger service. Click **Ambari dashboard > Actions > Add Service**.



- | 3. On the Choose Services page, select **Ranger** and **Ranger KMS**.

| | | |
|--|-------------------|---|
| <input checked="" type="checkbox"/> Ranger | 1.0.0.3.0 | Comprehensive security for Hadoop |
| <input checked="" type="checkbox"/> Ranger KMS | 1.0.0.3.0 | Key Management Server |
| <input type="checkbox"/> SmartSense | 1.5.0.2.7.0.0-897 | SmartSense - Hortonworks SmartSense Tool (HST) helps quickly gather configuration, metrics, logs from common HDP services that aids to quickly troubleshoot support cases and receive cluster-specific recommendations. |
| <input type="checkbox"/> Spark2 | 2.3.0 | Apache Spark 2.3 is a fast and general engine for large-scale data processing. |
| <input type="checkbox"/> Zeppelin Notebook | 0.8.0 | A web-based notebook that enables interactive data analytics. It enables you to make beautiful data-driven, interactive and collaborative documents with SQL, Scala and more. |
| <input type="checkbox"/> Druid | 0.12.1 | A fast column-oriented distributed data store. |
| <input type="checkbox"/> Spectrum Scale | 4.2.x | High-performance, scalable storage manages yottabytes of unstructured data (formerly known as General Parallel File System, or GPFS) |
| <input type="checkbox"/> Superset | 0.23.0 | Superset is a data exploration platform designed to be visual, intuitive and interactive. This service is Technical Preview . |

NEXT →

b1fbda17

- | 4. Customize the services. In the Ranger Admin dashboard, configure the following:
| • Under “DB Flavor”, select MYSQL.
| • For the Ranger DB host, the host name must be the location of MYSQL.

- For Ranger DB username, set the value to *rangeradmin*.
- For Ranger DB password, set the value to *rangeradmin*.

DB FLAVOR

Ranger DB host

Driver class name for a JDBC Ranger database

Ranger DB name

Ranger DB password

Ranger DB username

JDBC connect string for a Ranger database

b11bda8

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

Database Administrator (DBA) username

Database Administrator (DBA) password

JDBC connect string for root user

TEST CONNECTION

Connection OK

b11bda19

- In ADVANCED tab set password for:
 - Ranger Usersync user's password
 - Ranger Tagsync user's password
 - Ranger KMS keyadmin user's password

Ranger Admin username for Ambari

Ranger Admin user's password for Ambari

Ranger Usersync user's password

Ranger Tagsync user's password

Ranger KMS keyadmin user's password

b11bda20

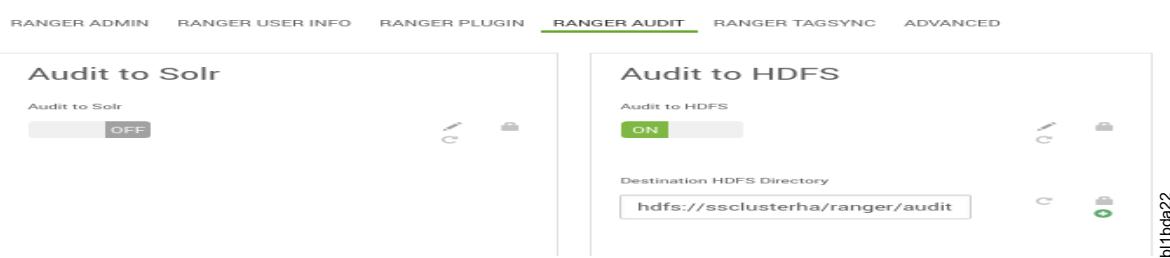
- Provide password for 'Advanced ranger-env' tab for 'Ranger Admin user's password'

Advanced ranger-env

Ranger Admin user's password

b11bda21

- In the Ranger Audit tab, ensure that the Audit to Solr option is disabled.



5. Customize the services, In the 'RANGER KMS'
 - a. Under "DB Flavor", select MYSQL.
 - b. For the 'Ranger KMS DB host', the hostname must be the location of MYSQL.
 - c. For Ranger KMS DB username, set the value to rangerkms.
 - d. For Ranger KMS DB password, set the value to rangerkms.

The screenshot shows the 'Ranger KMS DB' configuration page. It includes fields for 'DB FLAVOR' (set to 'MYSQL'), 'Ranger KMS DB host' (set to 'c902f10x06.gpfs.net'), 'SQL connector jar' (set to '{{driver_curl_target}}'), 'Driver class name for a JDBC Ranger KMS database' (set to 'com.mysql.jdbc.Driver'), 'Ranger KMS DB name' (set to 'rangerkms'), 'JDBC connect string' (set to 'jdbc:mysql://c902f10x06.gpfs.net:3306/rai'), 'Ranger KMS DB username' (set to 'rangerkms'), and 'Ranger KMS DB password' (both fields are masked). A note in a yellow box states: 'To use MySQL with Ranger_kms, you must download the https://dev.mysql.com/downloads/connector/j/ from MySQL. Once downloaded to the Ambari Server host, run: ambari-server setup --jdbc-db=mysql --jdbc-driver=/path/to/mysql/com.mysql.jdbc.Driver'.

- e. For 'Database Administrator (DBA) password', set the root user password.

- f. For 'KMS master key password' set new password.

Ranger KMS Root DB

Database Administrator (DBA) username: root

Database Administrator (DBA) password:

KMS Master Secret Password

KMS master key password:

b1bda24

- g. Deploy and complete the installation.

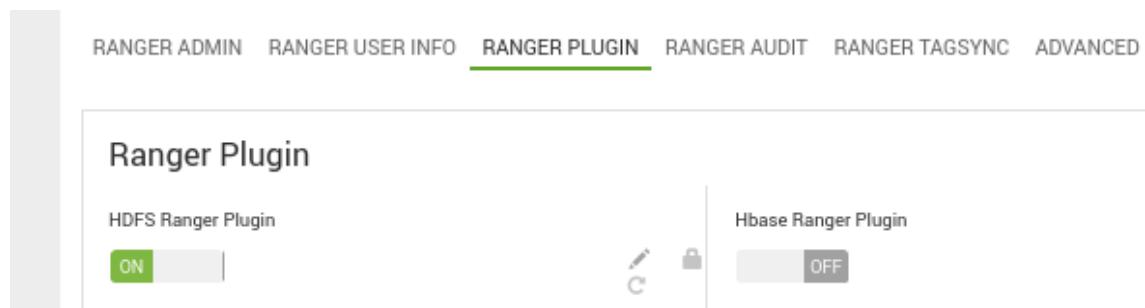
Assign the Ranger server to be on the same node as the HDFS Transparency namenode for better performance.

- Select **Next > Next > Deploy**.

Enabling Ranger HDFS plug-in:

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

- From the dashboard, click **Configs tab > Advanced tab > Advanced ranger-hdfs-plugin-properties**, check the box for Enable Ranger for HDFS.



b1bda25

- Save the configuration. The Restart required message is displayed at the top of the page. Click **Restart**, and select **Restart All Affected** to restart the HDFS service, and load the new configuration.

Note: After each step we need to save the config and **Restart All Affected** for the services requesting for it and load the new configuration.

After the HDFS restarts, the Ranger plug-in for HDFS is enabled.

Logging into Ranger UI:

This topic provides instructions to log in to the Ranger UI.

- | To log into the Ranger UI, log onto: <http://<gateway>:6080> using the following username and password:
- | User ID/Password: admin/admin



| **Disabling Ranger**

- | If you do not want to use Ranger any more, do the following:
 - | From the dashboard, click **Configs tab > Advanced tab > Advanced ranger-hdfs-plugin-properties** and uncheck the Disable Ranger box for HDFS.
 - | To increase performance, you could also disable Ranger in HDFS Transparency by executing the following steps:
 1. Log in to the Ambari GUI.
 2. Select the **IBM Spectrum Scale service > Configs** and set the value of **gpfs.ranger.enabled** to *false*.
 3. Save the configuration.
 4. Restart IBM Spectrum Scale service, then restart HDFS to sync this value to all the nodes.

| **Kerberos**

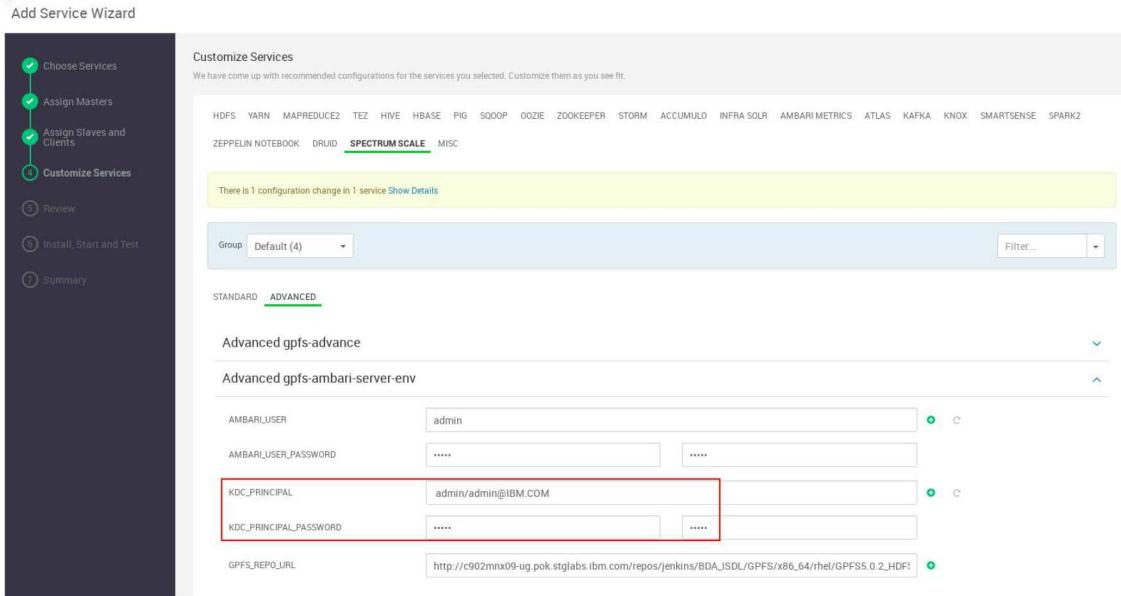
| **Enabling Kerberos**

- | Only MIT KDC is supported for IBM Spectrum Scale service through Ambari.
- | If you are using Kerberos that is not MIT KDC:
 1. Disable the Kerberos
 2. Install IBM Spectrum Scale service
 3. Enable Kerberos.
- | **Note:**
 - | If Kerberos is not disabled, then the IBM Spectrum Scale service can hang.
 - | DataNode reports exceptions after Kerberos is enabled on RHEL7.5. See 2nd generation HDFS Protocol troubleshooting - P19.

| **Enabling Kerberos when Spectrum Scale service is not integrated:**

- | IBM Spectrum Scale service is not integrated into Ambari.

- Follow “Setting up KDC server and enabling Kerberos” on page 196 to enable Kerberos. This is before deploying IBM Spectrum Scale service in “Install Mpack package” on page 152 and “Deploy the IBM Spectrum Scale service” on page 153. Once the Kerberos is enabled, the KDC information must be set during the deployment of the IBM Spectrum Scale Customizing Services panel.
- 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:



b11bda47

After all the required fields are set for the customized services panel, review all the fields in Customizing Services before clicking **NEXT**.

Note: The Admin principal and Admin password are the same as the corresponding KDC_PRINCIPAL and KDC_PRINCIPAL_PASSWORD values.

KDC_PRINCIPAL=Admin principal
KDC_PRINCIPAL_PASSWORD=Admin password

The KDC admin principal and KDC admin principal password are generated when the KDC server is set up.

Admin session expiration error

Missing KDC administrator credentials. Please enter admin principal and password.

Admin principal
admin/admin@IBM.COM

Admin password

Save Admin Credentials ?

CANCEL **SAVE** b1bda48

3. Continue 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” on page 196 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.

Ambari / Services / Spectrum Scale / Configs

SUMMARY CONFIGS METRICS

Version: 4

Config Group: Default (4)

STANDARD ADVANCED

Advanced gpfs-advance

Advanced gpfs-ambari-server-env

AMBARI_USER: admin

AMBARI_USER_PASSWORD: *****

KDC_PRINCIPAL: root/admin@IBM.COM

KDC_PRINCIPAL_PASSWORD: *****

GPFS_REPO_URL: http://c902mnx09-ug.pok.stglabs.ibm.com/repos/jenkins/BDA_ISDL/GPFS/x86_64/hel/GPFS5.0.1_HDFS

Advanced gpfs-env

- | 3. Restart all the services. Click **Ambari panel > Service Actions > Stop All and Start All**¹.
 - | ¹For FPO cluster, do not run STOP ALL from the Ambari GUI. Refer to the Limitations > General sections on how to properly stop IBM Spectrum Scale.
- | **Setting up KDC server and enabling Kerberos:**
 - | This topic provides steps to set up KDC server and enable Kerberos.
 - | 1. To set up the Key Distribution Center (KDC) server:
 - | • For HDP, follow the Install a new MIT KDC documentation.
 - | **Note:** If the KDC server is already implemented, skip this step.
 - | 2. On the Ambari GUI, click **Admin > Kerberos**, and follow the GUI panel guide to enable the Kerberos service.
- | **Kinit on the Namenodes:**
 - | This topic describes kinit on the Namenodes. These commands are run internally during HDFS service start up when the IBM Spectrum Scale service is integrated.
 - | On the Namenodes, run: # kinit -kt /etc/security/keytabs/nn.service.keytab nn/
| NN_HOSTNAME@REALM_NAME
| where,
 - | • NN_HOSTNAME is the Namenode host name (FQDN)
 - | • REALM_NAME is the KDC Realm
 - | • nn is the Kerberos Namenode naming convention created during Kerberos setup.
 - | For example: kinit -kt /etc/security/keytabs/nn.service.keytab nn/c902f05x01.gpfs.net@IBM.COM
 - | **Note:** If in a non-root environment, this command is internally run with sudo privilege.
 - | If HA, run the command on both the Namenodes.
- | **Kinit on the Datanodes:**
 - | This topic describes kinit on the Datanodes.
 - | On the Datanodes, run: # kinit -kt /etc/security/keytabs/dn.service.keytab dn/
| DN_HOSTNAME@REALM_NAME.
| where,
 - | • DN_HOSTNAME is the Datanode host name (FQDN)
 - | • REALM_NAME is the KDC Realm
 - | • dn is the Kerberos Datanode naming convention created during Kerberos setup
 - | For example: kinit -kt /etc/security/keytabs/dn.service.keytab dn/c902f05x01.gpfs.net@IBM.COM
 - | If in a non-root environment, ensure that you run this command with sudo privilege.
- | **Issues in Kerberos enabled environment:**
 - | This section lists the issues in the kerberos enabled environment and their workarounds.
- | **Bad local directories in Yarn**
 - | If Yarn shows an alert for bad local directories when IBM Spectrum Scale is integrated, and if the Yarn service check failed then Yarn does not have the correct permission to access the local mounted

| directories created by IBM Spectrum Scale. Click **Ambari > Yarn > Configs > Advanced > Node Manager**, and review the `yarn.nodemanager.local-dirs` for the local directory values.

| **Workaround**

| Fix the local directory permissions on all nodes to have `yarn:hadoop` user ID and group ID permissions.
| Restart all services, or go back to the previous step and continue with the process.

| For example,

```
| # Local directories under /opt/mapred  
| /dev/sdf1 on /opt/mapred/local1 type ext4 (rw,relatime,data=ordered)  
| /dev/sdg1 on /opt/mapred/local2 type ext4 (rw,relatime,data=ordered)  
| /dev/sdh1 on /opt/mapred/local3 type ext4 (rw,relatime,data=ordered)  
  
| # Check the directories under /opt/mapred  
| In /opt/mapred directory:  
| drwxrwxrwx 6 root root 4096 Mar  8 23:19 local3  
| drwxrwxrwx 6 root root 4096 Mar  8 23:19 local2  
|     drwxrwxrwx 6 root root 4096 Mar  8 23:19 local1  
  
| # Workaround:  
| # Change permission from root:root to yarn:hadoop for all the local* directories under /opt/mapred  
| # for all the nodes.  
  
| Under /opt/mapred directory:  
| chown yarn:hadoop local*  
  
| drwxrwxrwx 6 yarn hadoop 4096 Mar  8 23:19 local3  
| drwxrwxrwx 6 yarn hadoop 4096 Mar  8 23:19 local2  
|     drwxrwxrwx 6 yarn hadoop 4096 Mar  8 23:19 local1  
  
| # Restart all services (Or go back to your previous step and continue with the process).
```

| **Nodemanager failure due to device busy**

| Nodemanager fails to start due to local directory error:
| `OSError: [Errno 16] Device or resource busy: '/opt/mapred/local1'`

| To fix this issue:

- | 1. Go to **Yarn > Configs > Search for `yarn.nodemanager.local-dirs`**.
- | 2. Check the values for the Yarn local directories.
- | 3. The correct local directory values must contain the Yarn directory in the local directory path. For example:
| `yarn.nodemanager.local-dirs="/opt/mapred/local1/yarn,/opt/mapred/local2/yarn,/opt/mapred/local3/yarn"`
- | 4. If the `<local-dir>/yarn` is not specified in `yarn.nodemanager.local-dirs`, add the path, and save the configuration.

| **Journal nodes not installed in the native HDFS HA**

| If you are unintegrating from a Kerberos-enabled Namenode HA mode environment to native HDFS, you may sometimes find JournalNodes components missing in HDFS. In such cases, manually install the journal nodes.

| **Disabling Kerberos**

| This topic lists the steps to disable kerberos.

| **Note:** While enabling or disabling Kerberos, you do not need to stop the IBM Spectrum Scale service.

- | To disable Kerberos from Ambari:
 - | 1. Go to **Ambari GUI > Admin > Kerberos > Disable Kerberos.**

| **Short-circuit read (SSR)**

- | In HDFS, read requests go through the DataNode. When the client requests the DataNode to read a file, the DataNode reads that file off the disk, and sends the data to the client over a TCP socket. The short-circuit read (SSR) obtains the file descriptor from the DataNode, allowing the client to read the file directly.

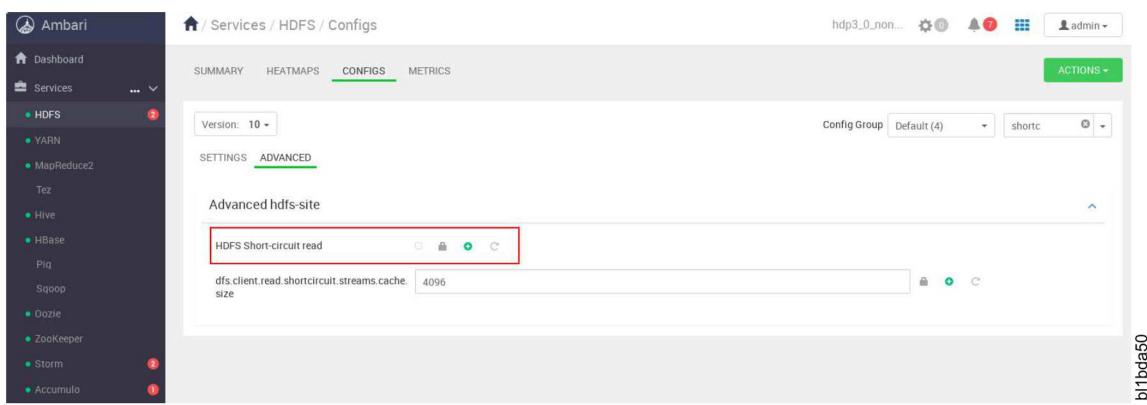
- | This is possible only in cases where the client is co-located with the data, and is used in the FPO mode.
- | The short-circuit reads provide a substantial performance boost to many applications.

- | **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**.



| **Disabling short circuit write**

- | This section describes how to disable short circuit write.

- | **Note:** By default, the short circuit write is enabled only if the short circuit read is enabled.

- | 1. Go to **Ambari GUI > Spectrum Scale > Custom gpfs-site**, add the `gpfs.short-circuit-write.enabled=false` property, and save the configuration.
- | 2. Restart IBM Spectrum Scale service.
- | 3. Restart HDFS service.
- | 4. Restart any services that are down.

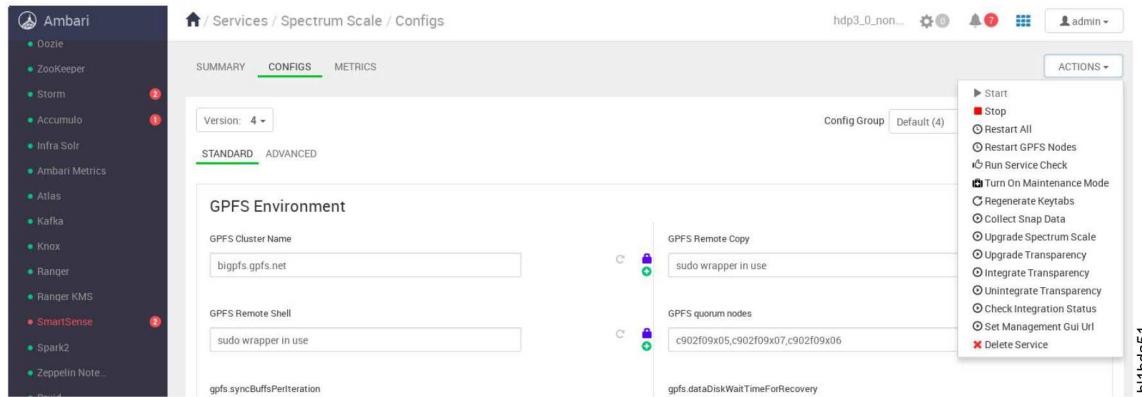
- | **Note:** If `gpfs.short-circuit-write.enabled` is *disabled*, there will be a lot of traffic over the local network lo adapter when you run a teragen job.

IBM Spectrum Scale service management

- 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.

Actions dropdown list

- To go to the Service Actions dropdown list, click **Spectrum Scale > Actions**.

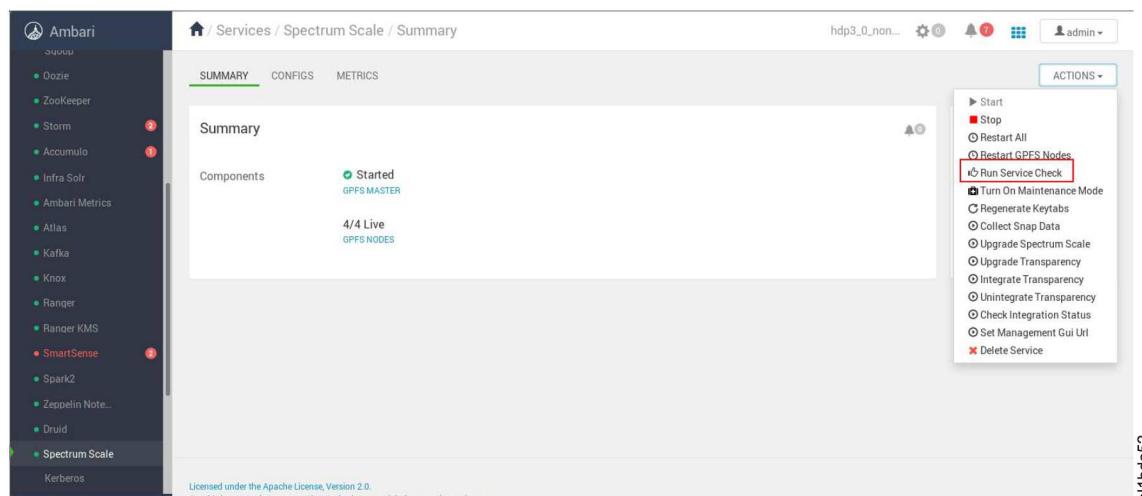


- Note:** Do not use the **Delete Service** action from the dropdown **Actions** menu. If you wish to get rid of the Spectrum Scale service, follow the procedure in “Uninstalling IBM Spectrum Scale Mpack and service” on page 167.

- When the IBM Spectrum Scale Mpack and service is deleted, the IBM Spectrum Scale file system and packages are preserved as is. For an FPO cluster created through Ambari, the mounted local disks `/opt/mapred/local*` and entries in `/etc/fstab` are preserved as is.

Running the service check

- To check the status and stability of the service, run a service check on the IBM Spectrum Scale dashboard by clicking **Run Service Check** in the Actions dropdown menu.



- Review the service check output logs for any issues.
- To manually check the HDFS Transparency Namenodes and Datanodes state, run the following command:

```

| /usr/lpp/mmfs/bin/mmhadoopctl connector getstate
| $ /usr/lpp/mmfs/bin/mmhadoopctl connector getstate
| c902f05x01.gpfs.net: namenode running as process 4749.
| c902f05x01.gpfs.net: datanode running as process 10214.
| c902f05x02.gpfs.net: datanode running as process 4767.
| c902f05x03.gpfs.net: datanode running as process 8204.

```

Stop all without stopping IBM Spectrum Scale service

To prevent IBM Spectrum Scale service from being stopped when you click **ACTION > STOP ALL**, place the IBM Spectrum Scale service into maintenance mode.

Click Ambari GUI > Spectrum Scale service > Actions > Turn on Maintenance Mode.

This prevents any Ambari actions from occurring on the service that is in maintenance mode.

To get out of Maintenance Mode, click Ambari GUI > Spectrum Scale service > Actions > Turn off Maintenance Mode.

Note: For FPO cluster, do not run STOP ALL from the Ambari GUI. Refer to the Limitations > General sections on how to properly stop IBM Spectrum Scale.

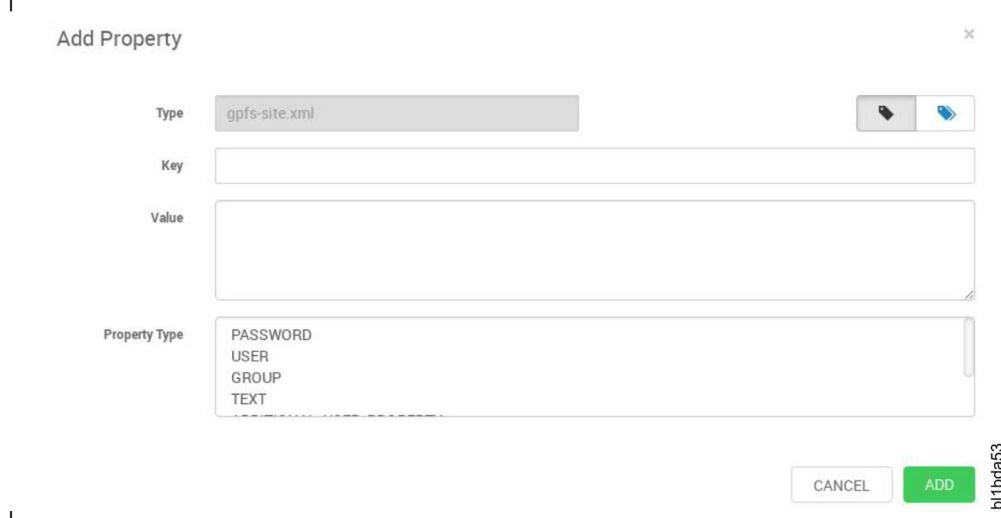
Modifying IBM Spectrum Scale service configurations

The IBM Spectrum Scale service has standard and advanced configuration panels.

Click Ambari GUI > Spectrum Scale > Configs tab.

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.



| In Ambari, if any configuration in the gpfs-site is changed in the Spectrum Scale dashboard, a Stop All¹ service followed by a Start All service is required. Check your environment to ensure that the changes made are in effect.

| ¹For FPO cluster, do not run STOP ALL from the Ambari GUI. Refer to the Limitations > General sections on how to properly stop IBM Spectrum Scale.

| **Note:** You must plan a cluster maintenance window and prepare for cluster downtime when restarting the IBM Spectrum Scale service and the HDFS service. Ensure that no I/O activities are active on the IBM Spectrum Scale file system before shutting down IBM Spectrum Scale. If the I/O activities are active, IBM Spectrum Scale fails to shut down as the kernel extension cannot be unloaded.

| **GPFS yum repo directory:**

| If the IBM Spectrum Scale yum repo directory is changed, you need to update the GPFS_REPO_URL in Ambari for the upgrade process to know where the packages are located.

| To update the GPFS_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/5.0.1/gpfs_rpms`

- | 3. Save the GPFS_REPO_URL configuration.

- | 4. The GPFS_REPO_URL becomes effective during the Upgrading IBM Spectrum Scale and Upgrading HDFS Transparency process.

| **HDFS and IBM Spectrum Scale restart order**

| When the IBM Spectrum Scale service is integrated, HDFS Transparency Namenodes and Datanodes are managed as HDFS Namenode and Datanode components respectively in the Ambari HDFS service.

| When configuration is changed in IBM Spectrum ScaleTM, the following restart order should be followed:

- | 1. Stop the HDFS service first.
- | 2. Restart the IBM Spectrum Scale service.
- | 3. Restart the HDFS service.

Integrating HDFS transparency

You must plan a cluster maintenance window, and prepare for the cluster down time when integrating the HDFS Transparency with the native HDFS. After each integration, you must run the **ambari-server restart** on the Ambari server node. Ensure that all the services are stopped.

- | To integrate the HDFS Transparency with the native HDFS:
 1. On the dashboard, click **Services > Stop All**¹ to stop all services. Verify that all services are stopped.
| If not, stop the services.
 - 1For FPO cluster, do not run STOP ALL from the Ambari GUI. Refer to the Limitations > General sections on how to properly stop IBM Spectrum Scale.
 2. Click **Spectrum Scale > Actions > Integrate Transparency**.

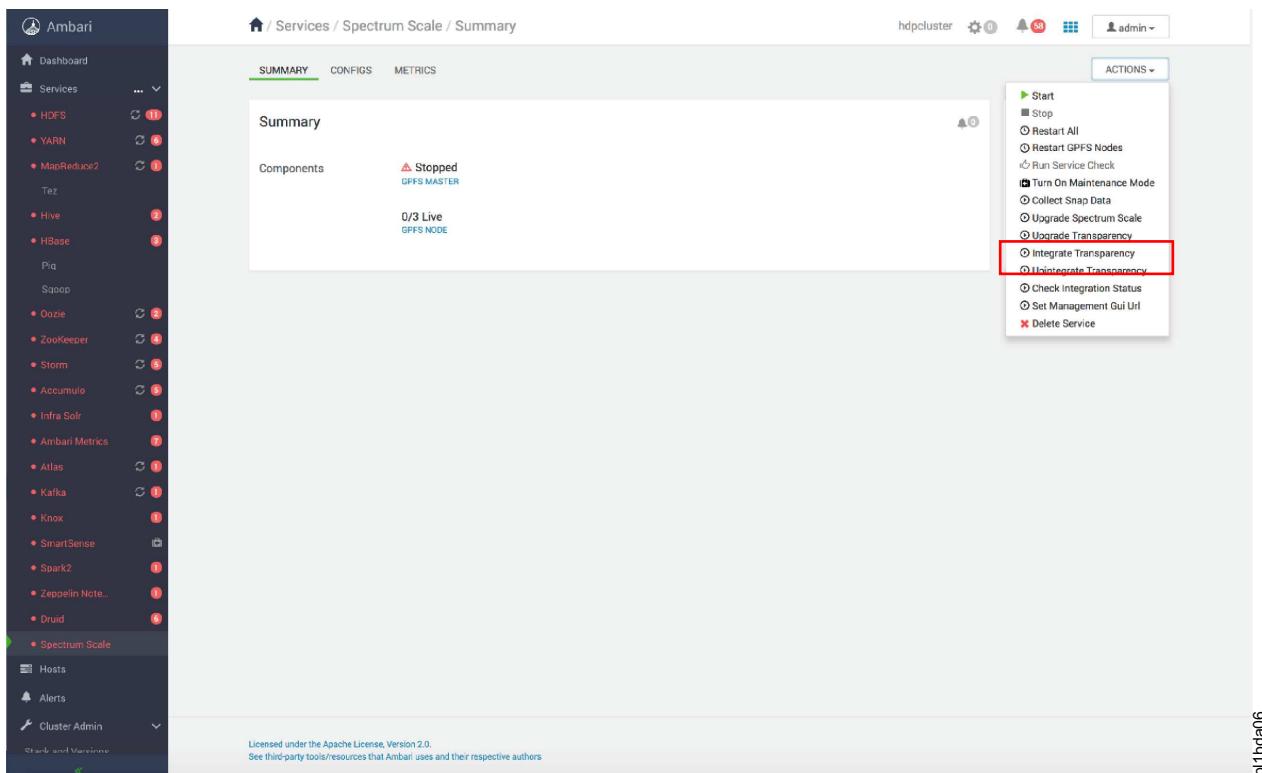


Figure 22. IBM SPECTRUM SCALE INTEGRATE TRANSPARENCY

- | 3. On the Ambari server node, run the **ambari-server restart** command to restart the Ambari server.
- | 4. Log back in to the Ambari GUI.
- | 5. Start all the services from Ambari GUI. The Hadoop cluster starts using IBM Spectrum Scale and the HDFS Transparency. The HDFS dashboard displays the Namenode and Datanode status of the HDFS Transparency.
- | On the HDFS dashboard, check the NameNode and DataNodes status.

bl1bda06

| Note: JournalNodes are not used when IBM Spectrum Scale service is integrated.

The screenshot shows the HDFS Summary page in the Ambari interface. At the top, there are tabs for SUMMARY, HEATMAPS, CONFIGS, and METRICS. The SUMMARY tab is selected. Below the tabs, the page is titled "Summary".
Components
- **Started**:
 - **NAMENODE**: Status is Started.
- **NAMENODE UPTIME**: 23m 18s.
- **Memory Usage**: 9.1% (92.4 MB / 1011.3 MB) for NAMENODE HEAP.
- **DATANODES**: 3/3 Started.
- **JOURNALNODES**: 0/0 Live.
- **NFSGATEWAYS**: 0/0 Started.
DATANODES STATUS
- **Live**: 3.
- **Dead**: 0.
- **Decommissioning**: 0.

bl1bda07

| Command verification

| To verify that the HDFS Transparency is available, use the following command to check the connector state:

```
| # Ensure all node GPFS state are active  
| /usr/lpp/mmfs/bin/mmgetstate -a  
| # Ensure all Namenode and Datanodes are running.  
| /usr/lpp/mmfs/hadoop/sbin/mmhadoopctl connector getstate
```

| For more information on how to verify the HDFS transparency integration state, see “Verifying Transparency integration state” on page 205

| Cluster environment

| After the IBM Spectrum Scale service is deployed, IBM Spectrum Scale HDFS Transparency is used instead of HDFS. HDFS Transparency inherits the native HDFS configuration and adds the additional changes for the HDFS Transparency to function correctly.

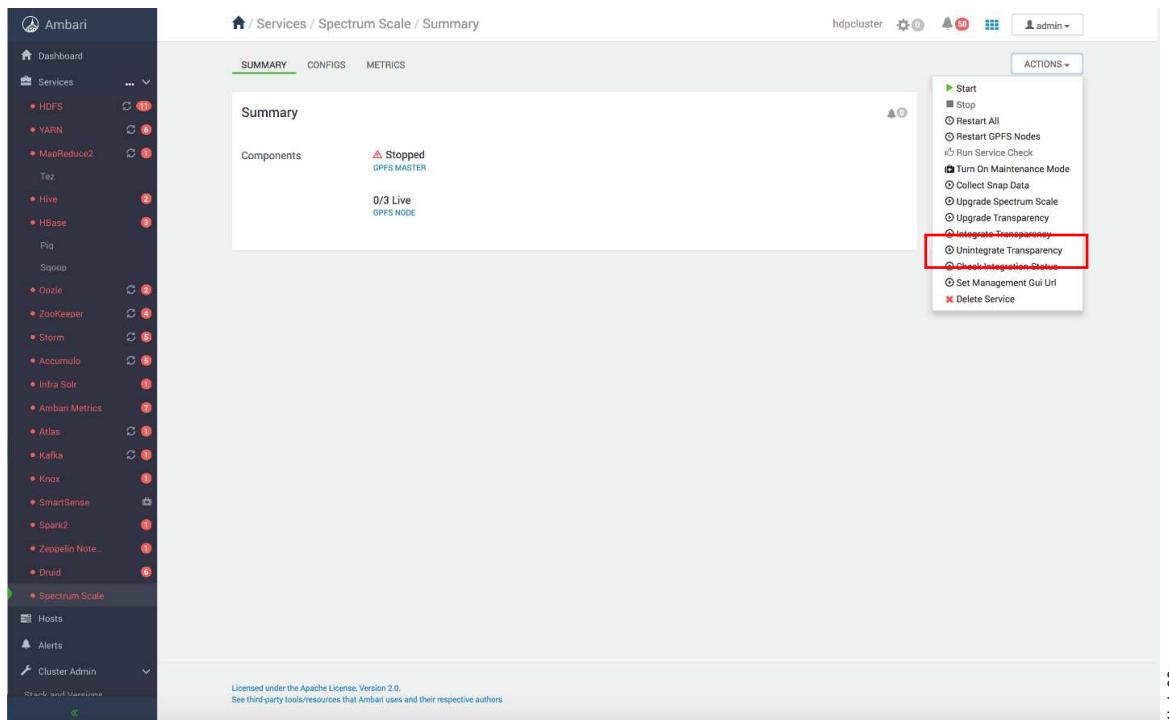
| After IBM Spectrum Scale is deployed, a new HDFS configuration set V2 is created, and is visible in the **HDFS Service Dashboard > CONFIG HISTORY**.

| Unintegrating HDFS Transparency

| You must plan a cluster maintenance window, and prepare for the cluster downtime while unintegrating the HDFS Transparency back to native HDFS. After each unintegration, you need to run the **ambari-server restart** on the Ambari server node. Ensure that all the services are stopped.

| 1. On the dashboard, click **Actions > Stop All**¹ to stop all services.

- ¹For FPO cluster, do not run STOP ALL from the Ambari GUI. Refer to the Limitations > General sections on how to properly stop IBM Spectrum Scale.
2. If Kerberos is enabled, set the **KDC_PRINCIPAL** and **KDC_PRINCIPAL_PASSWORD** values in the **IBM Spectrum Scale services > Configs > Advanced**. Save the configuration changes.
 3. Click **Spectrum Scale > Actions > Unintegrate Transparency**.



b1bda08

Figure 23. IBM SPECTRUM SCALE UNINTEGRATE 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 services from the Ambari GUI. The Hadoop cluster starts using native HDFS. The IBM Spectrum Scale service is not removed from the Ambari panel, and will be displayed in GREEN. IBM Spectrum Scale will function, but the HDFS Transparency will not function.
- Note:** When unintegrated back to native HDFS, the HDFS configuration used remains the same as the HDFS configuration used by the IBM Spectrum Scale prior to unintegration. If you must revert to the original HDFS configuration, go to the HDFS dashboard, and make the configuration changes in the Configs tab.

Command verification

- To verify that the HDFS Transparency is not available, use the following command to check the connector state:
- ```
Ensure all node GPFS state are active
/usr/lpp/mmfs/bin/mmgetstate -a

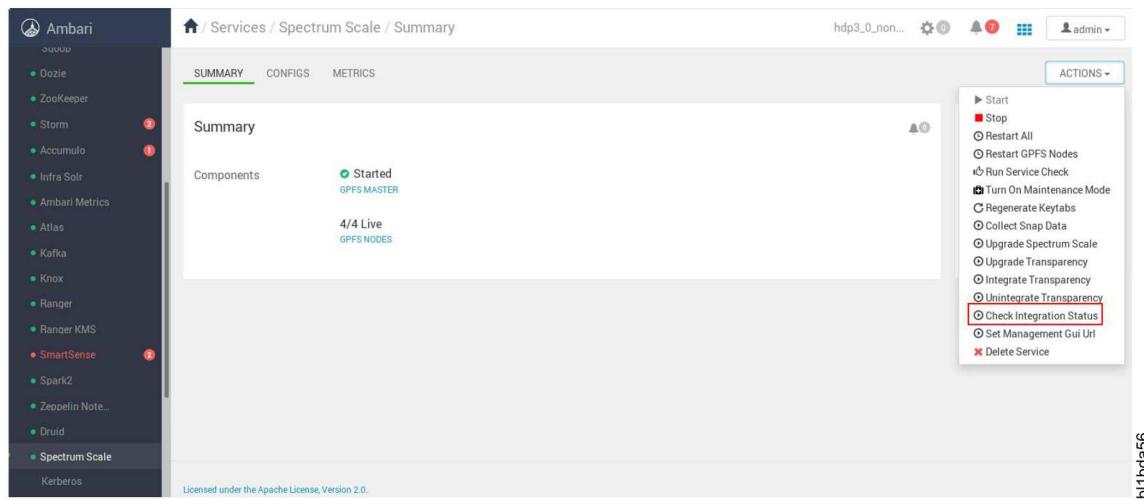
Ensure no Transparency Namenode and Datanodes are running.
/usr/lpp/mmfs/hadoop/sbin/mmhadoopctl connector getstate
```

## Cluster environment

- After using the Spectrum Scale Unintegrate Transparency function, the native HDFS will be in effect. The configuration from HDFS service before the unintegrate phase will still be in effect. The IBM Spectrum Scale configuration will not affect the native HDFS functionality. If you must revert back to the original native HDFS configuration, go to the HDFS dashboard, and select the V1 configuration version under the Configs tab.
- For information on verifying the Transparency integration state, see “Verifying Transparency integration state.”

## Verifying Transparency integration state

- To verify the HDFS Transparency integration state, click Ambari GUI > Spectrum Scale > Actions > Check Integration Status.



After the process completes, check the output log for the state information.



## Verify IBM Spectrum Scale Mpack version

- This topic describes how to verify the IBM Spectrum Scale Mpack version.
- To verify the IBM Spectrum Scale Mpack version, click Ambari GUI > Spectrum Scale > Actions > Check Integration Status.

## Ambari node management

- This section provides information to add, delete, move and set up a node in Ambari.

## Adding a host

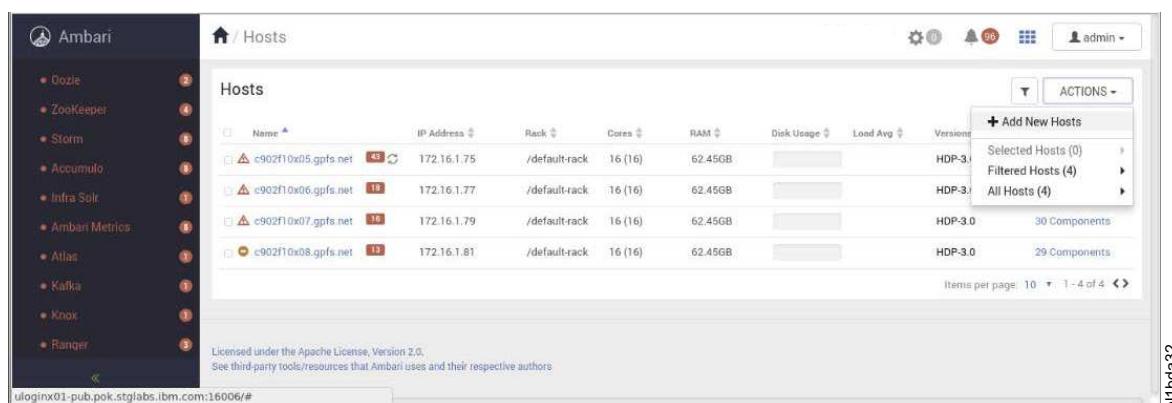
- This topic provides information to add a new node.
- See Preparing the environment section to prepare the new nodes.

### Note:

- Ensure that the IBM Spectrum Scale service is in integrated state before adding the node.
- On the new host being added, create a userid and groupid called *anonymous* with the same value as all the other GPFS nodes. For more information, see “Create the anonymous user id” on page 138 section.
- If you are adding new nodes to an existing cluster, and if the nodes being added already have IBM Spectrum Scale installed on them, ensure that the new nodes are at the same version of IBM Spectrum Scale as the existing cluster. Do not mix GPFS Nodes with different versions of IBM Spectrum Scale software in a GPFS cluster.
- If you are adding a new node to an existing cluster with inconsistent IBM Spectrum Scale versions, the new node will not install even if the failed installed node might still be displayed in the cluster list in Ambari. To delete the failed node from the cluster in Ambari, see “Deleting a host” on page 213. The new nodes can then be added to the Ambari cluster by using the Ambari web interface.
- If the Spectrum Scale cluster is configured in admin mode central mode, following steps need to be performed as prerequisite:
  - On the node to be added, execute:
    - Install all the GPFS packages.
    - Build the GPL layer using: /usr/lpp/mmfs/bin/mmbuildgpl.
  - On the admin node (usually the ambari server node) execute:
    - /usr/lpp/mmfs/bin/mmaddnode -N <FQDN-of-new-node>
    - /usr/lpp/mmfs/bin/mmchlicense server --accept -N <FQDN-of-new-node>
    - /usr/lpp/mmfs/bin/mmmount all
    - /usr/lpp/mmfs/bin/mmlsmount all

The new nodes can then be added to the Ambari cluster by using the Ambari web interface.

- On the Ambari dashboard, click **Hosts > Actions > Add New Hosts**.

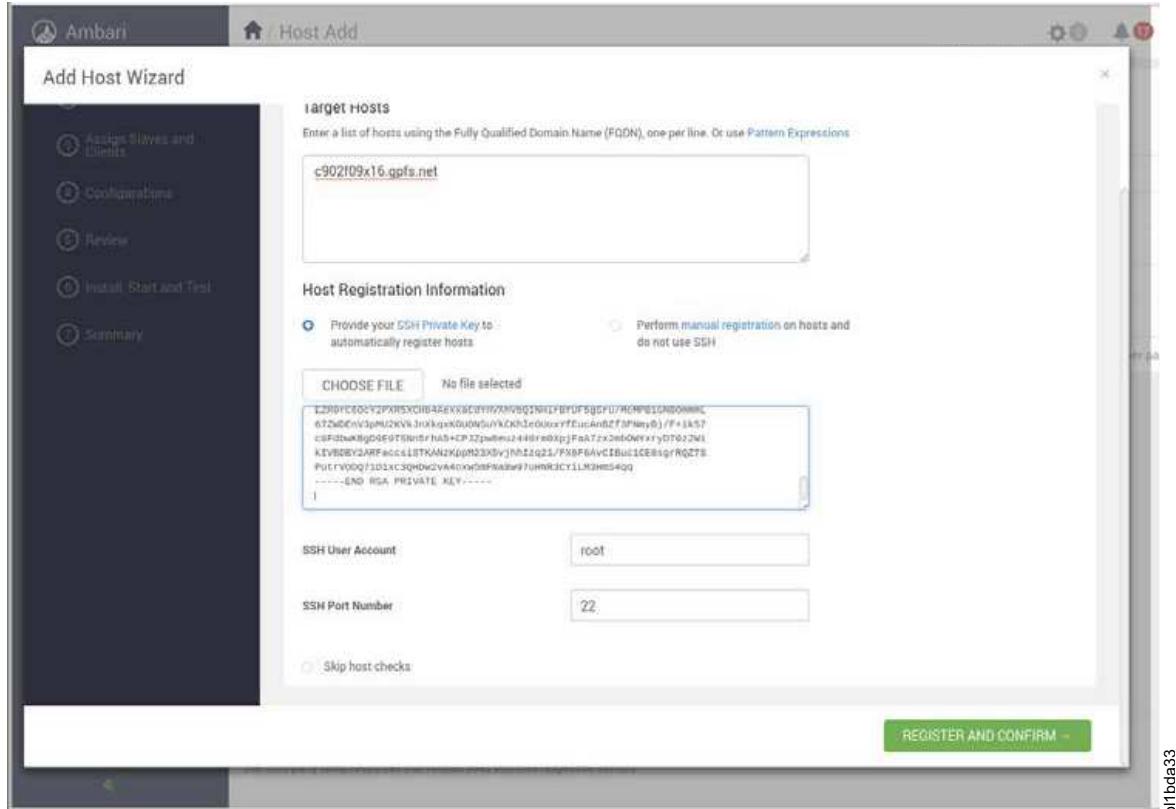


- Specify the new node information, and click **Registration and Confirm**.

### 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.

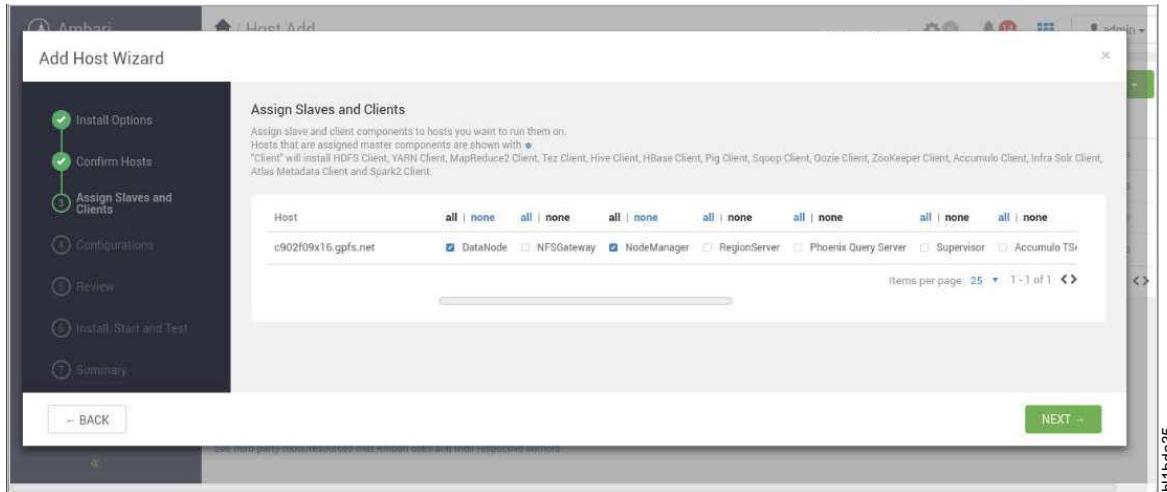


b1bda33

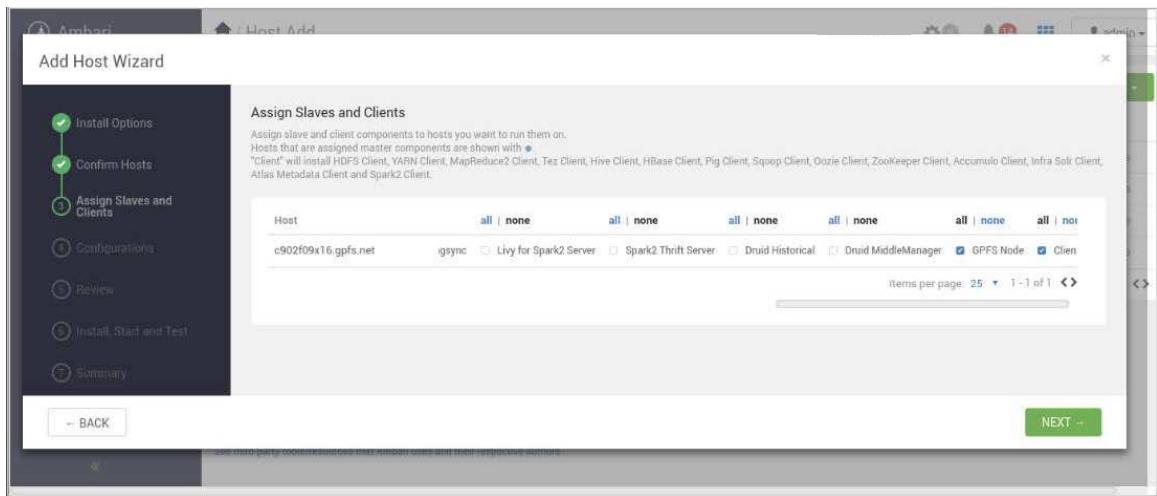
- Select the services that you want to install on the new node.

**Note:**

- If HDFS Transparency Datanode is needed on a host, select Datanode, NodeManager, and GPFS Node components for that host.
- If you just want the IBM Spectrum Scale client and not the Transparency components on a host, just select the GPFS Node component.

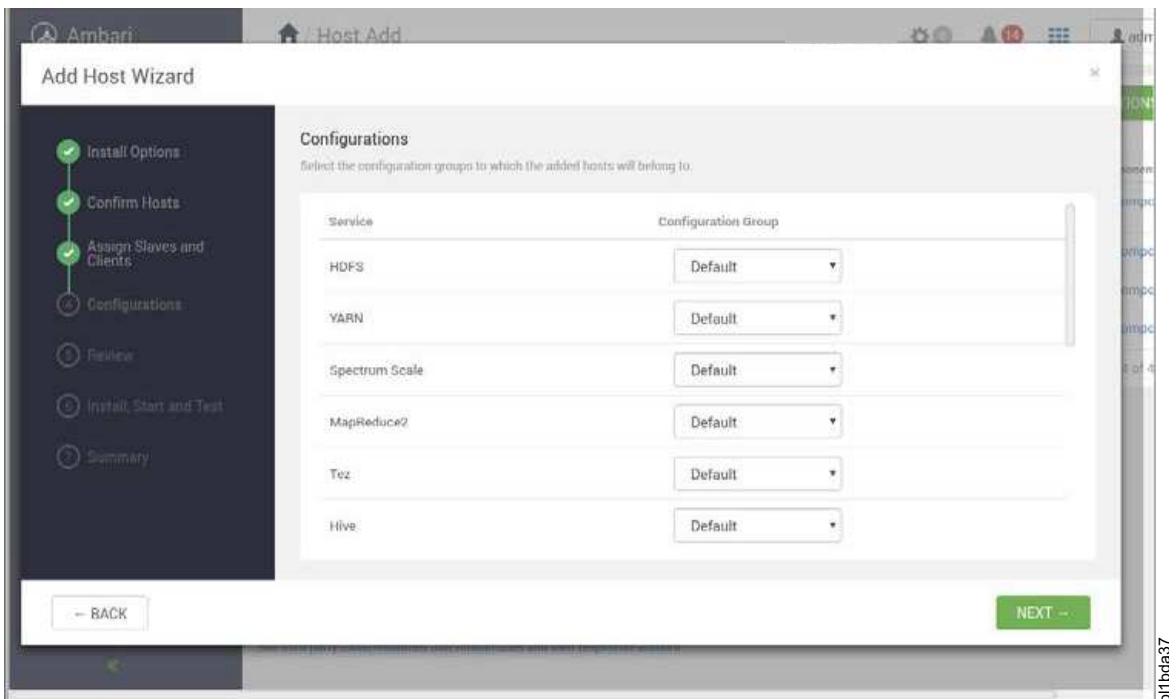


b1bda35



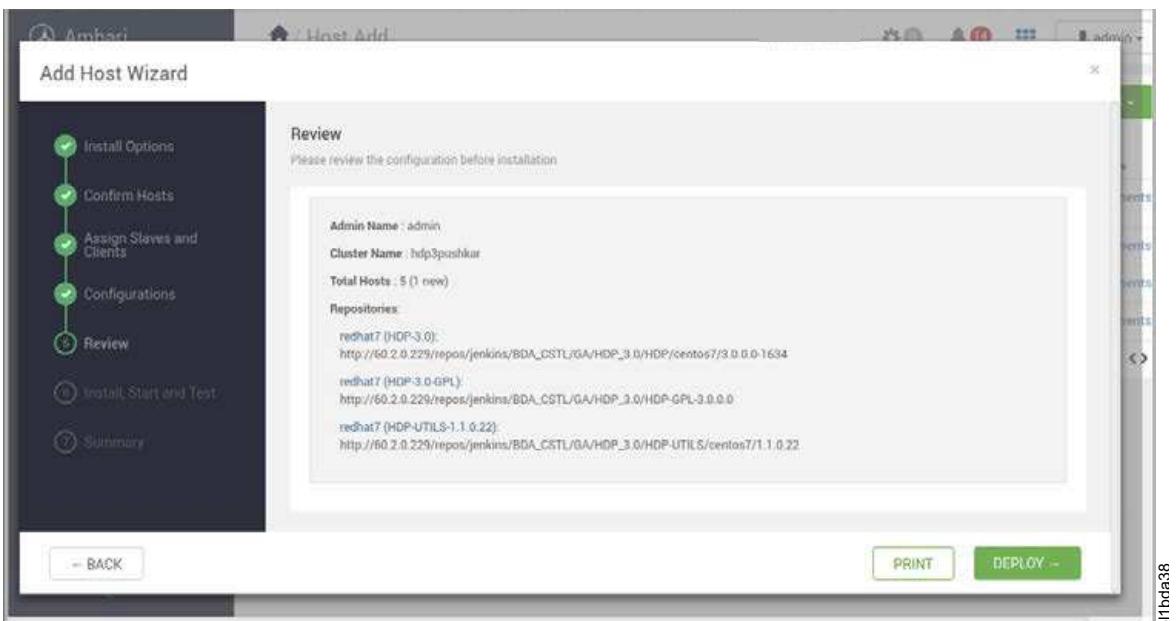
b1bda36

4. If several configuration groups are created, select one of them for the new node.



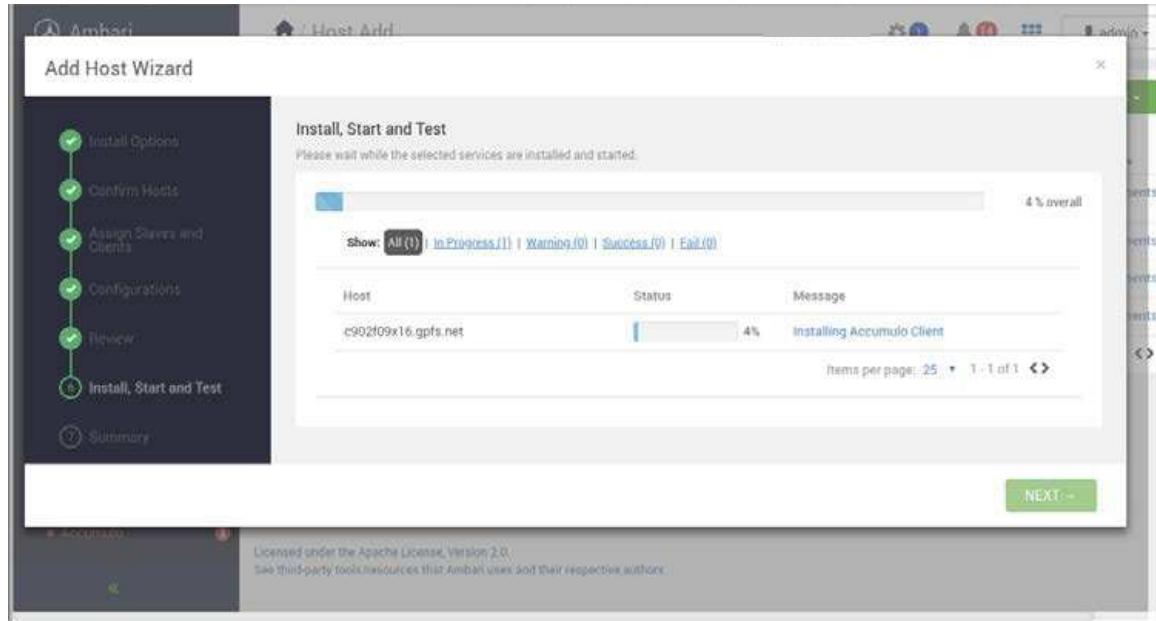
b1fda37

5. Review the information and start the deployment by clicking **Deploy**.



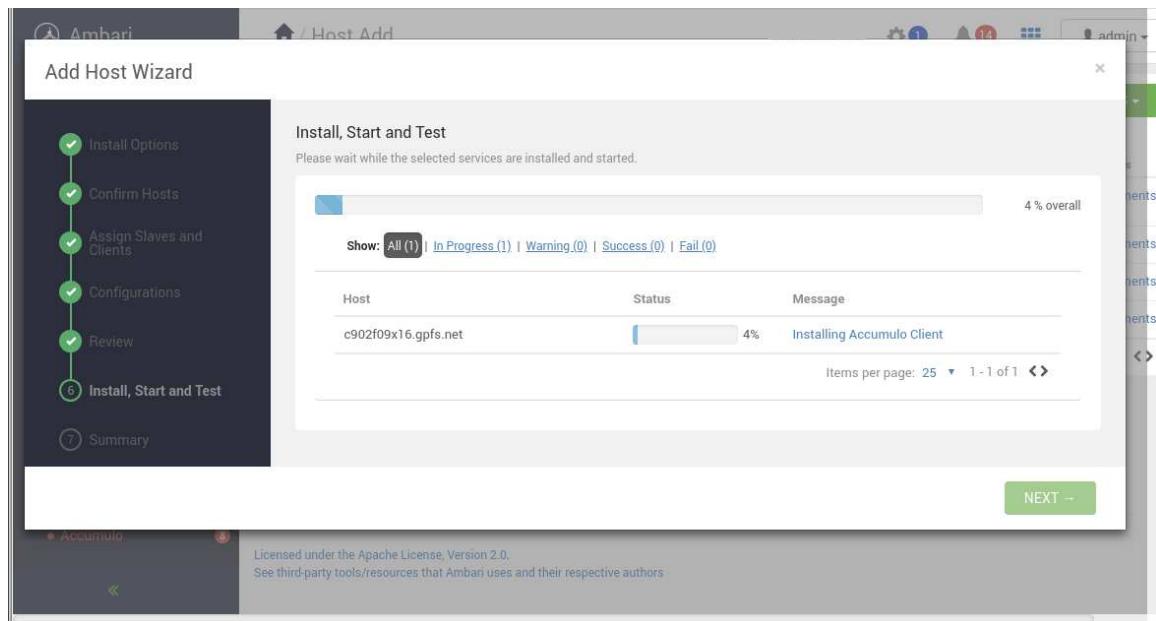
b1fda38

6. Install, Start and Test panel.

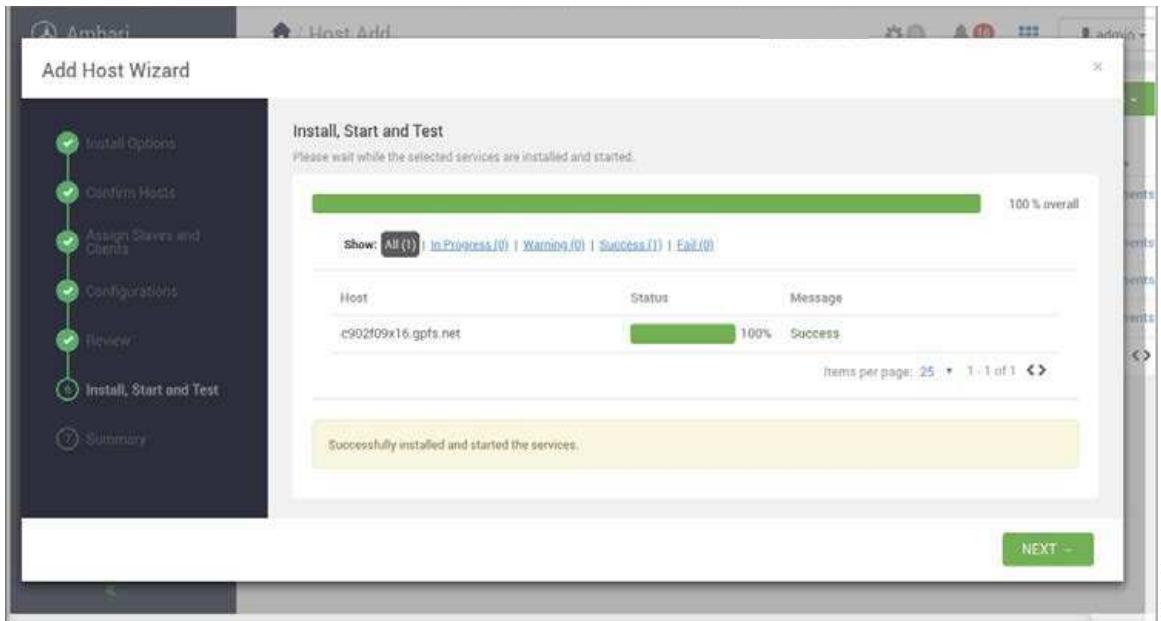


b1bda39

- After the Install, Start and Test wizard finished, click **Complete**.



b1bda40



b11bda41

8. A new node is added to the Ambari cluster.

From Hosts dashboard, the new node is added to the host list.

Name	IP Address	Rack	Cores	RAM	Disk Usage	Load Avg	Versions
c902f09x16.gpfs.net	172.16.1.65	/default-rack	16 (16)	62.45GB	0.15	HDP-3.0	
c903f11x05.gpfs.net	172.16.1.75	/default-rack	16 (16)	62.45GB	0.98	HDP-3.0	
c902f11x06.gpfs.net	172.16.1.77	/default-rack	16 (16)	62.45GB	0.43	HDP-3.0	
c902f11x07.gpfs.net	172.16.1.79	/default-rack	16 (16)	62.45GB	0.52	HDP-3.0	
c902f11x08.gpfs.net	172.16.1.81	/default-rack	16 (16)	62.45GB	0.41	HDP-3.0	

b11bda42

9. For any service with restart required icon, go to the service dashboard, select **Restart** > **Restart All Affected**.

**Note:** Ambari does not create NSDs on the new nodes. To create IBM Spectrum Scale NSDs and add NSDs to the file system, follow the steps in the ADD section in Deploying a big data solution using IBM Spectrum Scale.

10. Restart HDFS service in Ambari.

Check the cluster information

**Note:** In case of FPO, Ambari does not create NSDs on the new nodes. To create IBM Spectrum Scale NSDs and add NSDs to the filesystem, follow 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/mmlsnsd

File system Disk name NSD servers

bigpfs gpfs1nsd c902f05x01(gpfs.net)
bigpfs gpfs2nsd c902f05x02(gpfs.net)
bigpfs gpfs3nsd c902f05x03(gpfs.net)
bigpfs gpfs4nsd c902f05x04(gpfs.net)
bigpfs gpfs5nsd c902f05x03(gpfs.net)
bigpfs gpfs6nsd c902f05x02(gpfs.net)
bigpfs gpfs7nsd c902f05x01(gpfs.net)
bigpfs gpfs8nsd c902f05x04(gpfs.net)
bigpfs gpfs9nsd c902f05x02(gpfs.net)
bigpfs gpfs10nsd c902f05x03(gpfs.net)
bigpfs gpfs11nsd c902f05x04(gpfs.net)
bigpfs gpfs12nsd c902f05x01(gpfs.net)
bigpfs gpfs13nsd c902f05x02(gpfs.net)
bigpfs gpfs14nsd c902f05x03(gpfs.net)
bigpfs gpfs15nsd c902f05x04(gpfs.net)
bigpfs gpfs16nsd c902f05x01(gpfs.net)

[root@c902f05x01 ~]#
[root@c902f05x05 ~]# mount | grep bigpfs
bigpfs on /bigpfs type gpfs (rw,relatime)
[root@c902f05x05 ~]#
[root@c902f05x01 ~]# /usr/lpp/mmfs/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.

1. Stop all the components on the node to be deleted. For example: c902f09x16.gpfs.net
2. From Ambari dashboard, click **Hosts tab**, and select *the host that must be removed* and then click **Host Actions > Stop All Components**.

The screenshot shows the Ambari interface for managing hosts. On the left, the 'Services' sidebar lists various components like HDFS, YARN, MapReduce2, Tez, Hive, HBase, Pig, and Sqoop. The main area displays the 'Components' table for the host 'c902f09x16.gpfs.net'. The table includes columns for Status, Name, Type, and Action. Several components are shown as 'Slave' type. On the right, there's a 'Host Metrics' section with tabs for CPU Usage, Load, and Memory Usage, all showing 'No Data Available'. A 'HOST ACTIONS' dropdown menu is open, with the 'Stop All Components' option highlighted in red.

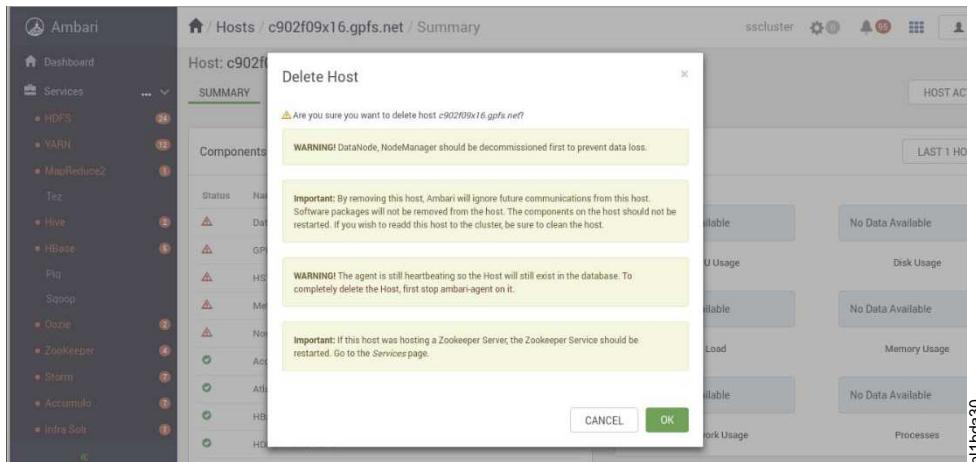
3. Stop the Ambari agent on the host to be deleted.

```
[root@c902f09x16 ~]
ambari-agent stop
Verifying Python version compatibility...
Using python /usr/bin/python2
Found ambari-agent PID: 22182
Stopping ambari-agent
Removing PID file at
/var/run/ambari-agent/ambari-agent.pid
ambari-agent successfully stopped
[root@c902f09x16 ~]#
```

4. To delete the host, click **Host Actions > Delete Hosts**.

The screenshot shows the Ambari interface for managing hosts. The host 'c902f09x16.gpfs.net' is selected. The 'HOST ACTIONS' dropdown menu is open, with the 'Delete Host' option highlighted in red.

The system displays a Warning message.



5. Click **OK**. The node is deleted from the Hosts list.

Name	IP Address	Rack	Cores	RAM	Disk Usage	Load Avg	Versions	Components
c902f10x13.gpfs.net	172.16.1.91	/default-rack	16 (16)	62.45GB	No Data Available	No Data Available	HDP-3.0	58 Components
c902f10x14.gpfs.net	172.16.1.93	/default-rack	16 (16)	62.45GB	No Data Available	No Data Available	HDP-3.0	36 Components
c902f10x15.gpfs.net	172.16.1.95	/default-rack	16 (16)	62.45GB	No Data Available	No Data Available	HDP-3.0	30 Components

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:

- From the dashboard, select **HDFS > Actions > Restart All**.
- After the HDFS service restarts, the deleted host is removed from the HDFS Transparency. The DataNodes status is 4/4 started, and the DataNodes Status is 4 live.

**Note:** This does not remove the Ambari packages and the IBM Spectrum Scale™ packages and NSD disks. It does not remove the node from the GPFS cluster either. Follow the IBM Spectrum Scale documentation on removing disks and packages from the environment.

```
[root@c902f10x13 ~]# /usr/lpp/mmfs/bin/mmgetstate -a
```

Node number	Node name	GPFS state
1	c902f10x13	active
2	c902f10x14	active
3	c902f10x15	active
4	c902f09x16	down

```
[root@c902f10x13 ~]#
```

```
[root@c902f10x13 ~]# /usr/lpp/mmfs/hadoop/sbin/mmhadoopctl connector getstate
c902f10x13(gpfs.net: namenode running as process 3413.
c902f10x14(gpfs.net: namenode running as process 5237.
c902f10x15(gpfs.net: datanode running as process 24456.
c902f10x13(gpfs.net: datanode running as process 15439.
c902f10x14(gpfs.net: datanode running as process 17884.
```

```
[root@c902f10x13 ~]#
```

## | Moving a namenode

| IBM Spectrum Scale HDFS Transparency Namenode is stateless, and does not maintain the FSImage-like information. The **move Namenode** option is not supported by the Ambari HDFS GUI when HDFS Transparency is integrated with the installed management pack version 4.1-X and later.

### | Note:

- | • The move Namenode script can only be run as a root.
- | • The move Namenode script can be executed in a Kerberized environment when the 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 when the IBM Spectrum Scale service is integrated.

| *Instructions for HA cluster:*

| This topic describes the steps to manually move a Namenode when HDFS Transparency is in integrate state.

- | 1. From the dashboard, select **Actions > Stop All**<sup>1</sup>.

| <sup>1</sup>For FPO cluster, do not run STOP ALL from the Ambari GUI. Refer to the Limitations > General sections on how to properly stop IBM Spectrum Scale.

- | 2. On the Ambari server host, run the following command:

```
| python /var/lib/ambari-server/resources/mpacks/SpectrumScaleExtension-MPack-
| <version>/extensions/SpectrumScaleExtension/<version>/
| services/GPFS/package/files/COMMON/MoveNameNodeTransparency.py
```

```
| Follow the command prompts and type the required input.
| $ python /var/lib/ambari-server/resources/mpacks/SpectrumScaleExtension-MPack-2.7.0.0/extensions/
| SpectrumScaleExtension/2.7.0.0/services/GPFS/package/files/COMMON/MoveNameNodeTransparency.py
| Enter the Ambari Server User:(Default User admin):
| Enter the Password for Ambari Server.
| Password:
| Retype password:
| SSL Enabled (True/False) (Default False):
| Enter the Ambari Server Port.(Default 8080)
| Enter the Fully Qualified HostName of the Source NameNode which has to be Removed:- c902f09x02.gpfs.net
| Enter the Fully Qualified HostName of the Destination NameNode has to be Added: c902f09x03.gpfs.net
```

| **Note:**

- SSL Enabled means Ambari HTTPS.
- The 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**<sup>1</sup>.

| <sup>1</sup>For FPO cluster, do not run STOP ALL from the Ambari GUI. Refer to the Limitations > General sections on how to properly stop IBM Spectrum Scale.

2. On the Ambari server host, run the following command:

```
python /var/lib/ambari-server/resources/mpacks/SpectrumScaleExtension-MPack-<version>/extensions/SpectrumScaleExtension/<version>/services/GPFS/package/files/COMMON/MoveNameNodeTransparency.py
```

| Follow the command prompts and type the required input.

```
$ python /var/lib/ambari-server/resources/mpacks/SpectrumScaleExtension-MPack-2.7.0.0/extensions/SpectrumScaleExtension/2.7.0.0/services/GPFS/package/files/COMMON/MoveNameNodeTransparency.py
```

| Enter the Ambari Server User:(Default User admin ):

| Enter the Password for Ambari Server.

| Password:

| Retype password:

| SSL Enabled (True/False) (Default False):

| Enter the Ambari Server Port.(Default 8080)

| Enter the Fully Qualified HostName of the Source NameNode which has to be Removed:c902f09x02.gpfs.net

| Enter the Fully Qualified HostName of the Destination NameNode has to be Added:c902f09x03.gpfs.net

| **Note:**

- SSL Enabled means Ambari HTTPS.
- The destination node must be one of the HDFS Transparency node.

3. From the dashboard, select **Actions > Start All**.

4. Moving the Namenode process is now completed. Verify that the Active Namenode and the Standby Namenode are correct.

| **Revert to native HDFS after move Namenode:**

| The **move Namenode** is executed when IBM Spectrum Scale is integrated using HDFS Transparency.

| However, you can later choose to use the native HDFS instead by unintegrating HDFS Transparency.

- | In that case, you must follow these steps, to ensure that the native HDFS have the correct Namenode setting.
  - | 1. Follow the steps 1-3 of “Unintegrating HDFS Transparency” on page 203 section to revert to native HDFS mode.
  - | 2. Do not start all the services after unintegrating.
  - | 3. Ensure **ambari-server restart** was run on the Ambari server.
  - | 4. If you have a Namenode HA enabled environment, then follow the HA steps listed below. Else, follow the non-HA environment steps.

| **If HA is enabled, perform the following steps, for example:**

- | Namenode being moved: c902f09x02
- | Execute the Move Namenode service during the HDFS Transparency Integration to the new Namenode c902f09x04
- | Namenode not moved: c902f09x03
  - | 1. Start the Zookeeper Server from the Ambari GUI.
  - | 2. Start the Namenode that was not moved (c902f09x03) from the Hosts dashboard, clicking the **Namenode that was not moved > Summary tab > Components > NameNode / HDFS (Active or Standby) > Start**. This will start only the Namenode. Do not start any other services or hosts.
  - | 3. Format the ZKFC on the Namenode that was not moved (c902f09x03) by running the following command:
 

```
| sudo su hdfs -l -c 'hdfs zkfc -formatZK'
```
  - | 4. On the new Namenode (c902f09x04), run the following command:
 

```
| sudo su hdfs -l -c 'hdfs namenode -bootstrapStandby'
```
  - | 5. Start All services.
  - | From the dashboard, select **Actions > Start All**. The Hadoop cluster will now use native HDFS.

| **If HA is not enabled, perform the following steps, for example:**

- | 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:
    - chown -R hdfs:hadoop /hadoop/hdfs/namenode**
    - mkdir -p /var/lib/hdfs/namenode/formatted**
  - | 3. Start All services.
  - | From the dashboard, select **Actions > Start All**. The Hadoop cluster will now use native HDFS.

| **Moving the Ambari server**

- | This section describes how to move the Ambari server onto a new host.

- | Moving the Ambari server is supported only if the current Ambari server host and the IBM Spectrum Scale service are active and functional.
- | The IBM Spectrum Scale master component is tightly integrated with the Ambari server, therefore the Moving the Ambari Server cannot be run when the IBM Spectrum Scale is in integrate state.

| Plan a cluster maintenance window and prepare for cluster downtime.

| **Note:**

- | • The new host has to be a GPFS node.
- | • Requires the `SpectrumScale_UpgradeIntegrationPackage` script which is packaged with the Mpack package. This script is used to remove the Scale Mpack and service, and reinstall it to a different location. The software stack is not upgraded. Ignore the \*STARTING WITH SPECTRUM SCALE EXPRESS UPGRADE POST STEPS and STARTING WITH SPECTRUM SCALE EXPRESS UPGRADE PRE STEPS outputs from the script.
- | 1. Log 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 > Actions > Stop**.
- | 5. On the Ambari server node as root, run the `SpectrumScale_UpgradeIntegrationPackage` script with the `--preEU` option.

| The `--preEU` option saves the existing IBM Spectrum Scale service information into JSON files in the local directory where the script was run. It also removes the IBM Spectrum Scale service from the Ambari cluster. This does not affect the IBM Spectrum Scale file system.

| Before proceeding, review the following questions and have the information ready for your environment. If Kerberos is enabled, more inputs are required.

```
$ cd /root/GPFS_Ambari
$./SpectrumScale_UpgradeIntegrationPackage --preEU
Are you sure you want to upgrade the GPFS Ambari integration package (Y/N)? (Default Y):
STARTING WITH PRE EXPRESS UPGRADE STEPS

Enter the Ambari server username:
Enter the password for the Ambari server.
SSL Enabled (True/False) (Default False):
Enter the Ambari server Port. (Default 8080):
...
Note: If Kerberos is enabled, then the KDC principal and password information are required.
Kerberos is Enabled. Proceeding with Configuration
Enter kdc principal:
Enter kdc password:
6. Run the MPack uninstaller script to remove the existing MPack.
$./SpectrumScaleMPackUninstaller.py
7. Move the Ambari server to the new host 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 step 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:
| ...
| ****
| Upgrade of the Spectrum Scale Service completed successfully.
| From the Ambari GUI, check the IBM Spectrum Scale installation progress through the background operations panel.
| ****
| ****
| IMPORTANT: You need to ensure that the HDFS Transparency package, gpfs.hdfs-protocol-3.0.X, is updated in the Spectrum
| Scale repository. Then follow the "Upgrade Transparency" service action in the Spectrum Scale service UI panel to propagate
| the package to all the GPFS Nodes.
| After that is completed, invoke the "Start All" services in Ambari.
| ****

```

## 12. Start all services by clicking **Ambari > Actions > Start All**.

Restart all components by using the restart icon.

### Note:

- If the 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.

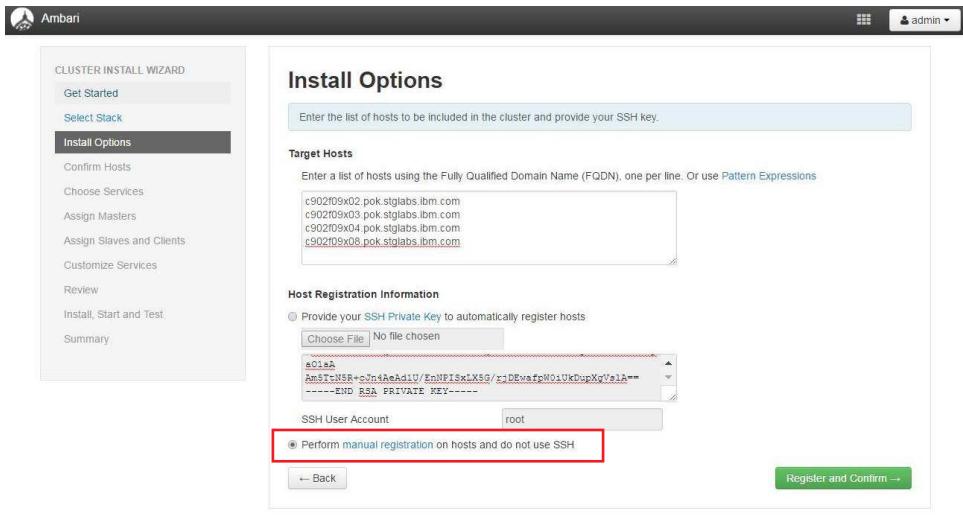
1. On Ambari dashboard, select **Hosts > Choose host > Components > Add** (Choose GPFS Node component)
2. Log back into Ambari.
3. From the dashboard, select **HDFS > Actions > Restart All**.

## **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” on page 137 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.
  - |     **Note:** Ensure that the UID and GID numbers of this ID are consistent across all the nodes.
  - | 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 HDP example uses gpfadm as the non-root id for the Ambari server as well for the Ambari agents and the Spectrum Scale cluster user. The user ID and group ID numbers of this user must be same on all the hosts.
- | For example:
  - | As root: ssh gpfadm@<ambari-agent-host> must work without a password.
  - | As gpfadm: ssh gpfadm@<ambari-agent-host> must work without a password.
- | 3. Set up an Ambari cluster as the non-root user.
  - | 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. HDFS Transparency requires certain password-less SSH configurations to be set up. Before adding the IBM Spectrum Scale service, follow the instructions for password-less SSH on your Namenode hosts as per "Password-less ssh access" on page 18.
- | **Note:** You can set **gpfss.ssh.user** to *gpfadm* in the IBM Spectrum Scale service UI panel.

| 5. 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.

| 6. 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
```

### HDP sudoers

| The non-root userid is gpfsadm:gpfsadm for the Ambari Server, Ambari agents and the Spectrum Scale cluster user.

| Example of /etc/sudoers file added entries in HDP environment:

```
Ambari Commands
gpfsadm 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 = gpfsadm
Defaults !env_reset,env_delete-=PATH
Defaults: ambari-server !requiretty
```

```
Ambari Agent non root configuration
```

```
Ambari Customizable Users
gpfsadm ALL=(ALL) NOPASSWD:SETENV: /bin/su hdfs *,/bin/su ambari-qa *,
/bin/su ranger *,/bin/su zookeeper *,/bin/su knox *,/bin/su falcon *,
/bin/su ams *, /bin/su flume *,/bin/su hbase *,/bin/su spark *,/bin/su accumulo *,
/bin/su hive *,/bin/su hcat *,/bin/su kafka *,/bin/su mapred *,/bin/su oozie *,
/bin/su sqoop *,/bin/su storm *,/bin/su tez *,/bin/su atlas *,/bin/su yarn *,
/bin/su kms *,/bin/su activity_analyzer *,/bin/su livy *,/bin/su zeppelin *,
/bin/su infra-solr *,/bin/su logsearch *
```

```
Ambari: Core System Commands
```

```
gpfsadm 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 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: Hadoop and Configuration Commands
```

```
gpfsadm 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
```

```
gpfsadm ALL=(ALL) NOPASSWD:SETENV: /usr/sbin/groupadd, /usr/sbin/groupmod,
/usr/sbin/useradd, /usr/sbin/usermod
```

```

Ambari: Knox Commands
gpfsadm 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
gpfsadm 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
gpfsadm 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 = gpfsadm
Defaults !env_reset,env_delete-=PATH
Defaults: gpfsadm !requiretty

#GPFS cluster non-root added
Preserve GPFS environment variables:
Defaults env_keep += "MMODE environmentType GPFS_rshPath GPFS_rcpPath
mmScriptTraceGPFSCMDPOR-TRANGE GPFS_CIM_MSG_FORMAT"

Allow members of the gpfs group to run all commands but only selected
commands without a password:
%gpfsadm 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:%gpfsadm !requiretty

```

7. 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.

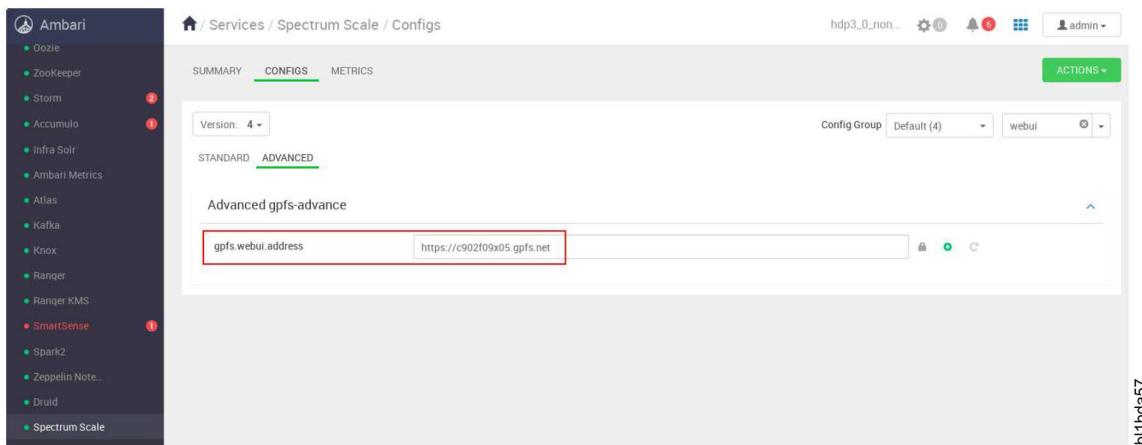
8. Perform the steps from Adding the IBM Spectrum Scale service to Ambari. This requires restarting Ambari as root. Exceptions occurs as non-root user.

## IBM Spectrum Scale management GUI

- The IBM Spectrum Scale management GUI must be manually installed and accessed.
- Installation instructions for IBM Spectrum Scale management GUI are available on IBM Knowledge Center here: Manually installing IBM Spectrum Scale management GUI.
- The IBM Spectrum Scale management GUI can be accessed as a quick link URL within the IBM Spectrum Scale Ambari service.
- If you are running IBM Spectrum Scale 4.2.0 or later, the rpms required to install the GUI are included in Standard and Advanced Editions for Linux on x86 and Power (Big Endian or Little Endian). The GUI requires RHEL 7.

### Procedure

- 1. Deploy IBM Spectrum Scale Management GUI.
- 2. Set the **gpfs.webui.address** field in the IBM Spectrum Scale Service configuration advanced panel.  
For example, [https://<ambari\\_server\\_fully\\_qualified\\_hostname>/gui](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.



h11hda57

- 3. Restart the IBM Spectrum Scale service
- 4. Sync the configuration.
- Click **Ambari GUI** > **IBM Spectrum Scale service** > **Actions** > **Set Management GUI URL**.
- 5. Restart Ambari server.

## IBM Spectrum Scale versus Native HDFS

When IBM Spectrum Scale service is added, the native HDFS is no longer used. The Hadoop application interacts with HDFS transparency similar to their interactions with the native HDFS.

The application can access HDFS by using Hadoop file system APIs and Distributed File System APIs. The application can have its own cluster that is larger than the HDFS protocol cluster. However, all the nodes within the application cluster must be able to connect to all the nodes in the HDFS protocol cluster by RPC.

**Note:** The Secondary NameNode and Journal nodes in native HDFS are not needed for HDFS Transparency because of the following reasons:

- The HDFS Transparency namenode is stateless.
- Metadata are distributed.
- The namenode does not maintain the FSImage-like or EditLog information.

## Functional limitations

This topic lists the functional limitations.

### General

- The maximum number of Extended Attributes (EA) is limited by IBM Spectrum Scale. The total size of the EA key and value must be less than a metadata block size in IBM Spectrum Scale.
- The EA operation on snapshots is not supported.
- Raw namespace is not implemented because it is not used internally.
- If **gpfs.replica.enforced** is configured as gpfs, the Hadoop shell command **hadoop dfs -setrep** does not take effect. Also, **hadoop dfs -setrep -w** stops functioning and does not exit.

- HDFS Transparency namenode does not provide *safemode* because it is stateless.
  - HDFS Transparency namenode does not need the second namenode like native HDFS because it is stateless.
  - Maximal replica for IBM Spectrum Scale is 3 from code. However, the maximal replica for your file system might be less than 3. You can check this by running `/usr/lpp/mmfs/bin/mmlsfs <fs-name> -R`.
  - Spectrum Scale has no ACL entry number limit. The maximal entry number is limited by Int32.
  - **SendPacketDownStreamAvgInfo** and **SlowPeersReport** from `http://<namenode/datanode:port>/jmx` are not supported.
  - GPFS file data replication factor on ESS requires to be set to 1, and `dfs.replica` should be set to 1.
  - HDFS supported interface for `hdfs xxx` is `hdfs dfs xxx`. Other interface from `hdfs xxx` is considered native HDFS specific, that is not used by the HDFS Transparency.
- These are some examples of what is not supported:
- `fsck`
  - `dfsadmin`
  - - `safemode`
  - Native HDFS caching (`cachefileadmin`)
  - Namenode format not needed to run (`namenode -format`)
- Distcp over snapshot is not supported
  - For HDFS Transparency version 3.0.x, the environment variables above can be exported, except for `HADOOP_COMMON_LIB_NATIVE_DIR`.
- This is because HDFS Transparency uses its own native .so library.
- For HDFS Transparency version 3.0.x:
- If you did not export `HADOOP_CONF_DIR`, then HDFS Transparency will read all the configuration files under `/var/mmfs/hadoop/etc/hadoop` such as the `gpfs-site.xml` file and the `hadoop-env.sh` file.
  - If you export `HADOOP_CONF_DIR`, then HDFS Transparency will read all the configuration files under `$HADOOP_CONF_DIR`. Since `gpfs-site.xml` is required for HDFS Transparency, it will only read the `gpfs-site.xml` file from the `/var/mmfs/hadoop/etc/hadoop` directory.

#### For HDP

- The “+” is not supported when using `hftp://namenode:50070`.

### Functional differences

- This topic lists the functional differences.
- ACLs is limited to 32 in native HDFS but not in IBM Spectrum Scale.
  - File name length is limited in native HDFS while IBM Spectrum Scale uses maximal 255 utf-8 chars.
  - The `hdfs fsck` is not supported in HDFS Transparency. Instead, use the IBM Spectrum Scale `mmfsck` command. If your file system is mounted, run `/usr/lpp/mmfs/bin/mmfsck -o -y`. If your file system is not mounted, run `/usr/lpp/mmfs/bin/mmfsck -y`.

### Configuration that differs from native HDFS in IBM Spectrum Scale

- This topic lists the differences between native HDFS and IBM Spectrum Scale.

*Table 24. 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.

| *Table 24. NATIVE HDFS AND IBM SPECTRUM SCALE DIFFERENCES (continued)*

<b>Property name</b>	<b>Value</b>	<b>New definition or limitation</b>
dfs.namenode.acls.enabled	True/false	For native HDFS, the namenode manages all metadata, including the ACL information. HDFS can use this to turn the ACL checking on or off. However, for IBM Spectrum Scale, the HDFS protocol does not hold the metadata. When on, the ACL is set and stored in the IBM Spectrum Scale file system. If the administrator turns it off later, the ACL entries that are set and stored in IBM Spectrum Scale take effect. This will be improved in the next release.
dfs.blocksize	Long digital	Must be a multiple of the IBM Spectrum Scale file system block size ( <code>mmlsfs -B</code> ). The maximal value is $1024 * \text{file-system-data-block-size}$ ( <code>mmlsfs -B</code> ).
gpfs.data.dir	String	A user in Hadoop must have full access to this directory. If this configuration is omitted, a user in Hadoop must have full access to <code>gpfs.mount.dir</code> .
dfs.namenode.fs-limits.max-xattrs-per-inode	INT	Does not apply to the HDFS protocol.
dfs.namenode.fs-limits.max-xattr-size	INT	Does not apply to the HDFS protocol.
dfs.namenode.fs-limits.max-component-length	INT	Does not apply to HDFS Transparency. The file name length is controlled by IBM Spectrum Scale. Refer to 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.

---

## | **Limitations**

### | **Limitations and information**

| Known information, limitations and workarounds for IBM Spectrum Scale and HDFS Transparency integration are stated in this section.

#### | **General**

- The IBM Spectrum Scale service does not support the rolling upgrade of IBM Spectrum Scale and Transparency from the Ambari GUI.

- The rolling upgrade of Hortonworks HDP cluster is not supported if the IBM Spectrum Scale service is still integrated.
- The minimum recommended version for IBM Spectrum Scale is 4.1 and above. HDFS Transparency is not dependent on the version of IBM Spectrum Scale.
- Manual Kerberos setup requires Kerberos setting in Ambari to be disabled before deploying IBM Spectrum Scale mpact. If IBM Spectrum Scale service is already installed, the HDFS Transparency requires to be unintegrated before enabling Kerberos in Ambari.
- Federation Support
 

Federation is supported for open source Apache Hadoop stack. The HDFS Transparency connector supports two or more IBM Spectrum Scale file systems to act as one uniform file system for Hadoop applications. For more information, see “Hadoop Storage Tiering” on page 48.
- The latest JDK supported version for Ambari is 1.8.0.77.
- Ambari is required to be restarted as root in a non-root environment, to avoid exceptions.
- All configuration changes must be made through the Ambari GUI, and not manually set into the HDFS configuration files or into the HDFS Transparency configuration files. This is to ensure that the configuration changes are propagated properly.
- In your existing cluster, if the HDFS settings in the HDFS Transparency configuration files were manually changed (For example: settings in core-site, hdfs-site, or log4j.properties in /var/mmfs/hadoop/etc/hadoop) and these changes were not implemented in the existing native HDFS configuration files, during the deployment of Ambari IOP or HDP and Spectrum Scale service, the HDFS Transparency configuration is replaced by the Ambari UI HDFS configurations. Therefore, save changes that are set for the HDFS Transparency configuration files so that these values can later be applied through the Ambari GUI.
- For FPO systems, ensure that you follow the proper steps to stop/start IBM Spectrum Scale. Otherwise, restarting the IBM Spectrum Scale NSD might not be possible after NSDs go down and auto recovery fails. This can occur when doing **STOP ALL/START ALL** from Ambari which stops IBM Spectrum Scale without properly handling the NSDs in FPO mode for Mpack 2.4.2.6 and earlier and for Mpack 2.7.0.0. See P10: IBM Spectrum Scale NSD are not able to be recovered in FPO clusters (Stop/Start of Scale service via Ambari GUI) for more information.

## 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 you need to deploy the IOP or HDP over an existing IBM Spectrum Scale FPO cluster, either store the Yarn’s intermediate data into the IBM Spectrum Scale file system, or use idle disks formatted as a local file system. It is recommended to use the latter method. If a new IBM Spectrum Scale cluster is created through the Ambari deployment, all the Yarn’s NodeManager nodes should be FPO nodes with the same number of disks for each node specified in the NSD stanza.
- If you are deploying Ambari HDP on top of an existing IBM Spectrum Scale and HDFS Transparency cluster:
  - Perform a backup of the existing HDFS and HDFS Transparency configuration before proceeding to deploy Ambari IOP or HDP, or deploy the IBM Spectrum Scale service with Ambari on a system that has HDFS Transparency installed on it.

- Ensure that the HDFS configuration provided through the Ambari UI is consistent with the existing HDFS configuration.
  - The existing HDFS Namenode and Datanode values must match the Ambari HDFS UI Namenode and Datanode values. Otherwise, the existing HDFS configuration will be overwritten by the default Ambari UI HDFS parameters after the Add Service Wizard completes.
  - The HDFS Datanodes being assigned in the **Assign Slaves and Clients** page in Ambari must contain the existing HDFS Transparency Datanodes. If the host did not have HDFS Datanode and GPFS Node set in Ambari, data on that host is not accessible, and cluster might be under replicated. If the node was not configured as an HDFS Datanode and GPFS node during the **Assign Slaves and Clients**, the host can add those components through the HOSTS component panel to resolve those issues. For more information, see “Adding GPFS node component” on page 219.
  - The HDFS Namenodes specified in the Ambari GUI during configuration must match the existing HDFS Transparency Namenodes.
  - Verify that the host names that are used are the data network addresses that IBM Spectrum Scale uses for its cluster setup. Otherwise in an existing or shared file system, the IBM Spectrum Scale service fails during installation because of a wrong host name.
  - While deploying HDP over an existing IBM Spectrum Scale file system, the IBM Spectrum Scale cluster must be started, and the file system must be mounted on all the nodes before starting the Ambari deployment.
- When deploying the Ambari IOP or HDP cluster, ensure there are no mount points in the cluster. Otherwise, the Ambari will take the shared mount point directory as the directory for the open source services. This will cause the different nodes to write to the same directory.
- Ensure that all the hosts for the IBM Spectrum Scale cluster contain the same domain name while creating the cluster through Ambari.
- IBM Spectrum Scale service requires that all the Namenodes and Datanodes are GPFS nodes.
- The IBM Spectrum Scale Ambari management pack uses the manual installation method and not the IBM Spectrum Scale installation toolkit.
- If installing a new FPO cluster through Ambari, Ambari creates the IBM Spectrum Scale with the recommended settings for FPO, and builds the GPFS portability layer on each node.
- It is recommended to assign HDFS Transparency Namenode running over GPFS node with metadata disks.
- It is recommended to assign Yarn ResourceManager node to be running HDFS Transparency NameNode.

## Configuration

- After adding and removing nodes from Ambari, some aspects of the IBM Spectrum Scale configuration, such as page pool, as seen by running the `mmlsconfig` command, are not refreshed until after the next restart of the IBM Spectrum Scale Ambari service. However, this does not impact the functionality.
- Short circuit is disabled when IBM Spectrum Scale service is installed. For information on how to enable or disable Short Circuit, see Short Circuit Read Configuration.

## Ambari GUI

- If any GPFS node other than the GPFS Master is stopped, the IBM Spectrum Scale panel does not display any alert.
- The NFS gateway is displayed on the HDFS dashboard but is not used by HDFS Transparency. NFS gateway is not supported. Use IBM Spectrum Scale protocol for better scaling if your application requires NFS interface.
- The **IBM Spectrum Scale Service UI Panel > Actions > Collect\_Snap\_Data** does not work if you configure an optional argument file (`/var/lib/ambari-server/resources/gpfs.snap.args`).

- For IBM Spectrum Scale GUI quick link, it is required to initialize the IBM Spectrum Scale management GUI before accessing through Ambari quick links. See IBM Spectrum Scale management GUI.

## Node management

- Ambari adds nodes and installs the IBM Spectrum Scale software on the existing IBM Spectrum Scale cluster, but does not create or add NSDs to the existing file system.
- Adding a node in Ambari fails if the node to be added does not have the same IBM Spectrum Scale version or the same HDFS Transparency version as the version currently installed on the Ambari IBM Spectrum Scale HDFS Transparency cluster. Ensure that the node to be added is at the same IBM Spectrum Scale level as the existing cluster.
- Decommissioning a Datanode is not supported when IBM Spectrum Scale is integrated.
- Moving a Namenode from the Ambari HDFS UI when HDFS Transparency is integrated is not supported. To manually move the Namenode, see Moving a Namenode.
- New key value pairs added to the IBM Spectrum Scale Ambari management pack GUI Advance configuration **Custom Add Property** panel are not effective in the IBM Spectrum Scale file system. Therefore, any values not seen in the Standard or Advanced configuration panel will need to be set manually on the command line using the IBM Spectrum Scale `/usr/lpp/mmfs/bin/mmchconfig` command.

## IBM Spectrum Scale

- Ensure that bi-directional password-less SSH is set up between all GPFS Nodes.
- The Hadoop services IDs and groups are required to have the same values across the cluster. Any user name needs a user ID in the OS or active directory service when writing to the file system. This is required for IBM Spectrum Scale.
  - **If you are using LDAP/AD:** Create the IDs and groups on the LDAP server, and ensure that all nodes can authenticate the users.
  - **If you are using local IDs:** The IDs must be the same on all nodes with the same ID and group values across the nodes.
- IBM Spectrum Scale only supports installation through a local repository.
- The management pack does not support IBM Spectrum Scale protocol and Transparent Cloud Tiering (TCT) packages.
- Ensure that in an HDFS Transparency environment, the IBM Spectrum Scale file system is set to permit any supported POSIX ACL types. Issue `mm1sfs <Device> -k` to ensure the `-k` value is set to *all*.

## HDP

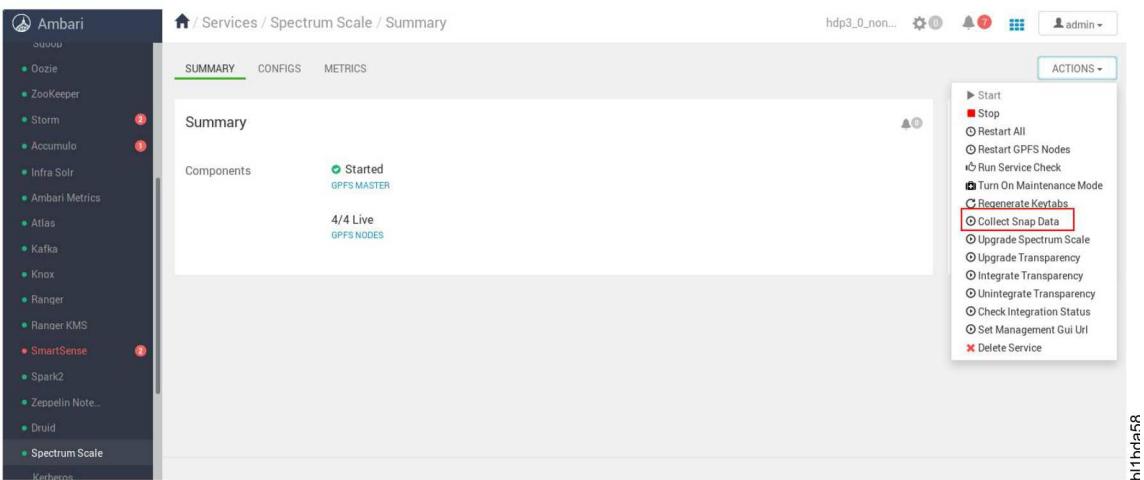
- The Manage JournalNodes is shown in **HDFS > Actions** submenu. This function should not be used when IBM Spectrum Scale service is deployed.
- The `+` is not supported when using `hftp://namenode:50070`.

---

## Problem determination

### Snap data collection

- You can collect the IBM Spectrum Scale snap data from the Ambari GUI. The command is run by the IBM Spectrum Scale Master, and the snap data is saved to `/var/log/ambari.gpfs.snap.<timestamp>` on the IBM Spectrum Scale Master node.



| By default, the IBM Spectrum Scale Master runs the following command:

```
| /usr/lpp/mmfs/bin/gpfs.snap -d /var/log/ambari.gpfs.snap.<timestamp> -N <all nodes>
| --check-space --timeout 600
```

| Where **<all nodes>** is the list of nodes in the IBM Spectrum Scale cluster and in the Ambari cluster. The external nodes in a shared cluster, such as ESS servers, are not included.

#### | Note:

- | • If your cluster has IBM Spectrum Scale file system version 4.2.2.0 and later, **gpfs.snap** will include the **--hadoop** option.
- | • If you run the Collect Snap Data through Ambari GUI, the Ambari logs will be captured into a tar package under the `/var/log` directory. The base **gpfs.snap --hadoop option** command does not capture the Ambari logs. The Ambari logs are only captured by clicking, **IBM Spectrum Scale Service > Actions > Collect Snap Data** in the Ambari GUI.

| You can also override the default behavior of this snap command by providing the arguments to the **gpfs.snap** command in the file `/var/lib/ambari-server/resources/gpfs.snap.args`. This works only if you are running the **gpfs.snap** on the command line. For example, if you wanted to write the snap data to a different location, collect the snap data from all nodes in the cluster, and increase the timeout. You can provide a **gpfs.snap.args file** option similar to that in the following example:

```
| # cat /var/lib/ambari-server/resources/gpfs.snap.args
| -d /root/gpfs.snap.out -a --timeout 1200
```

| The Ambari snap data tar package is stored as `/var/log/ambari.mpack.snap.<TIMESTAMP>.tar.gz` on the IBM Spectrum Scale Master node.

| The Ambari snap captures the following information:

- | 1. From all Ambari client:
  - `/var/log/hadoop/root/*`
  - `/var/lib/ambari-agent/data/`
  - `/var/log/ambari-agent/ambari-agent.log`
- | 2. From Ambari-server:
  - `/var/log/ambari-server/ambari-server.log`
  - `/var/run/ambari-server/stack-recommendations/`

*Figure 24. AMBARI COLLECT SNAP DATA*

## General

| Note: For known HDFS Transparency issues, see 2nd generation HDFS Protocol troubleshooting wiki.

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 HDP** over existing IBM Spectrum Scale file system.

- ## I 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 HDP and HDP-UTILS repository over the Ambari server and agents, according to your specification in the Ambari GUI in **Select Stack** section.

If you interrupt the Ambari deployment, clean the files before starting up Ambari the next time, especially when you specify a different IBM Spectrum Scale, HDP, or HDP-UTILS yum URL.

- | 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:

| For HortonWorks HDP 3.0, HDFS Transparency 3.0 has removed the configuration **gpfs.supergroup**  
| defined in /var/mmfs/hadoop/etc/hadoop/gpfs-site.xml.

| By default, the groups from the configuration **dfs.permissions.superusergroup** in  
| `/var/mmfs/hadoop/etc/hadoop/hdfs-site.xml` and the group root are super groups.

| 6. 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`

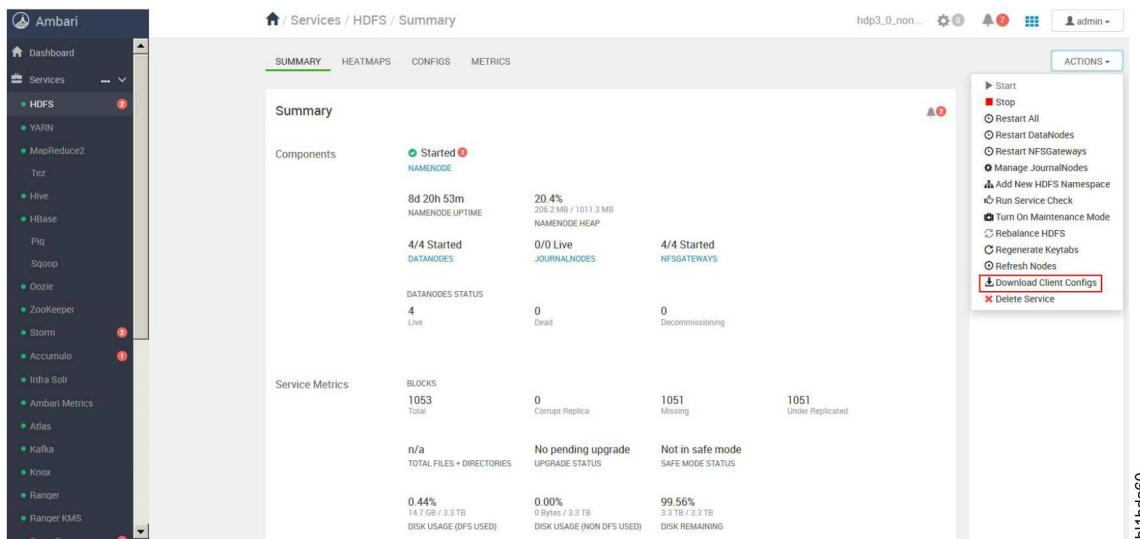
| The size of the thread pool can be increased to match the number of CPUs on the node where the  
| Ambari Server is running.

| 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
```

| 7. HDFS Download Client Configs does not contain HDFS Transparency configuration.

| **Solution:** In the HDFS dashboard, go to **Service Actions > Download Client Configs**, the tar  
| configuration downloaded does not contain the HDFS Transparency information.



| The workaround is to tar up the HDFS Transparency directory.

| Run the following command on a HDFS Transparency host to tar up the HDFS Transparency  
| directory into `/tmp`:

```
cd /var/mmfs/hadoop/etc/
tar -cvf /tmp/hdfs.transparency.hadoop/etc.tar hadoop
```

| 8. HDFS checkpoint confirmation warning message from **Actions > Stop All<sup>1</sup>** 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.

The screenshot shows the Ambari dashboard with the HDFS service selected. A context menu is open, and the 'Stop All' option is highlighted. The main panel displays HDFS metrics such as NameNode uptime (8d 1h 50m), DataNodes status (4/4 Started), and Service Metrics (BLOCKS: 1048 Total, 0 Corrupt Replica, 1045 Missing, 1045 Under Replicated).

b11bda61

Therefore, when doing **Ambari dashboard > Actions > Stop All**<sup>1</sup>, Ambari will generate a confirmation box to ask user to do an HDFS checkpoint using the **hdfs dfsadmin -safemode** commands. This is not needed when HDFS Transparency is integrated, and this step can be skipped. Click on next to skip this step.

## Confirmation

The last HDFS checkpoint is older than 12 hours. Make sure that you have taken a checkpoint before proceeding. Otherwise, the NameNode(s) can take a very long time to start up.

1. Login to the NameNode host **c902f09x07.gpfs.net**.
2. Put the NameNode in Safe Mode (read-only mode):

```
sudo su hdfs -l -c 'hdfs dfsadmin -safemode enter'
```

3. Once in Safe Mode, create a Checkpoint:

```
sudo su hdfs -l -c 'hdfs dfsadmin -saveNamespace'
```

CANCEL

NEXT

b11bda62

9. 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.

10. Kerberos authentication error during Unintegrate Transparency action

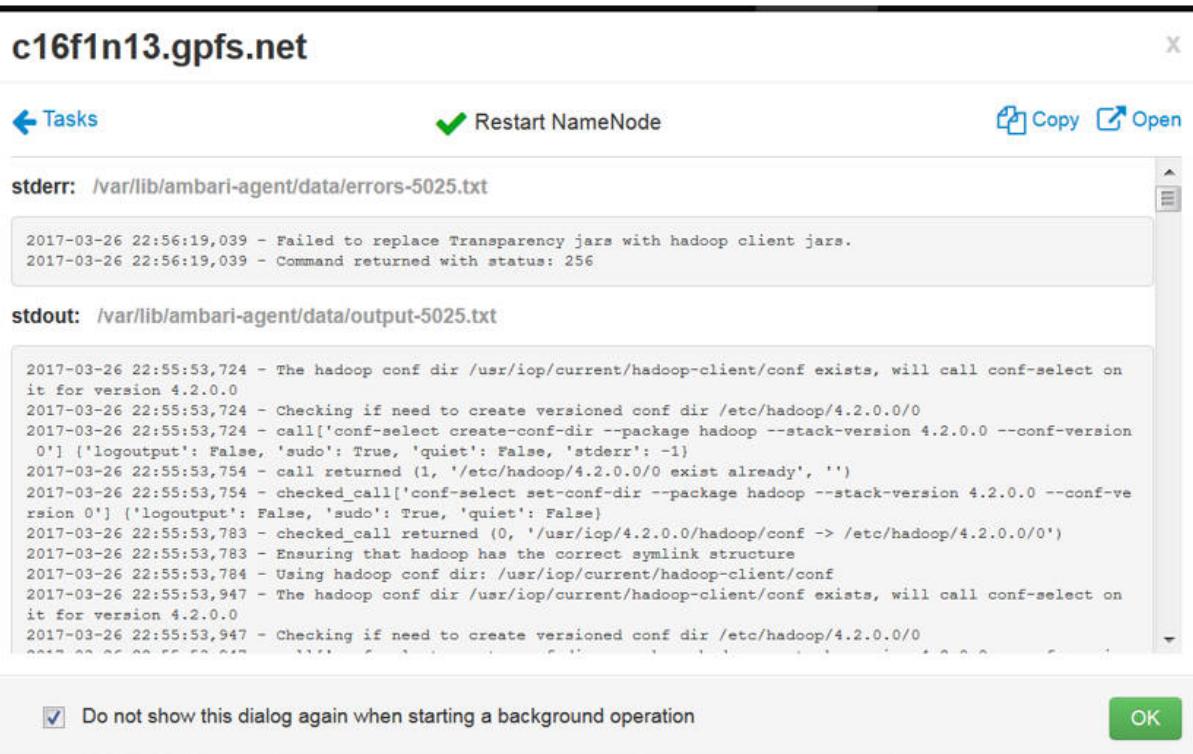
**ERROR:** Kerberos Authentication Not done Successfully. Exiting Unintegration.

Enter Correct Credentials of Kerberos KDC Server in Spectrum Scale Configuration.

**Solution:**

If error occurs in a Kerberos environment, check to ensure that the KDC\_PRINCIPAL and KDC\_PRINCIPAL\_PASSWORD values in **Spectrum Scale services > Configs > Advanced tab** have the correct values. Save the configuration changes.

- | 11. Namenodes and Datanodes failed with the error Fail to replace Transparency jars with hadoop client jars when short-circuit is enabled.



**Solution:** Install the Java OpenJDK development tool-kit package, java-<version>-openjdk-devel, on all the Transparency nodes. Ensure that the version is compatible with your existing JDK version. See HDFS Transparency package.

- | 12. ssh rejects additional ssh connections which causes the HDFS Transparency syncconf connection to be rejected.

| **Solution:** If the **ssh maxstartups** 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 **maxstartups** value.

```
maxstartups 1024:30:1024
```

| Restart the ssh daemon:

```
service sshd restart
```

- | 13. Not able to view Solr audits in Ranger.

| **Solution:** To resolve this issue:

- | a. Remove the solr ranger audit write lock file if it exists as root or as the owner of the file.

```
$ ls /bigpfs/apps/solr/data/ranger_audits/core_node1/data/index/write.lock
$ rm /bigpfs/apps/solr/data/ranger_audits/core_node1/data/index/write.lock
```

- | b. Restart HDFS and Solr.

| Click **Ambari GUI > HDFS > Actions > Restart All**

- | Click Ambari GUI > Solr > Actions > Restart All
- | 14. On restarting the service that failed due to network port being in use, the Namenode is still up after  
| doing a STOP ALL<sup>1</sup> 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.
- | 15. UID/GID failed with illegal value Illegal value: USER = xxxx > MAX = 8388607  
| **Solution:** If you have installed Ranger, and you need to leverage Ranger capabilities, then you need  
| to make the UID/GID less than 8388607.  
| If you do not need Ranger, follow these steps to disable Ranger from HDFS Transparency:  
| a. On the Ambari GUI, click IBM Spectrum Scale > Configs and set the **Add gpfs.ranger.enabled**  
| to *false*.  
| b. Save the configuration.  
| c. Restart IBM Spectrum Scale.  
| d. Restart HDFS.
- | 16. 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.
- | 17. 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.  
| For FPO cluster, do not run STOP ALL from the Ambari GUI. Refer to the Limitations > General  
| sections on how to properly stop IBM Spectrum Scale.
- | 18. IBM Spectrum Scale service cannot be deployed in a non-root environment.  
| **Solution:**  
| If the deployment of IBM Spectrum Scale service in an 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.
- | 19. User permission denied when Ranger is disabled.  
| If Kerberos is enabled and Ranger is disabled, the user gets the permission denied errors when  
| accessing the file system for HDFS Transparency 3.0.0 and earlier.  
| **Solution:**  
| Check the Kerberos principal mapping **hadoop.security.auth\_to\_local** field in the  
| /var/mmfs/hadoop/etc/hadoop/core-site.xml or in Ambari under HDFS Config to ensure that the  
| Namenode and Datanode are mapped to root instead of HDFS. For example, change

FROM:  
RULE:[2:\$1@\$0] (dn@COMPANY.DIV.COM)s/.\*/hdfs/  
RULE:[2:\$1@\$0] (nn@COMPANY.DIV.COM)s/.\*/hdfs/

TO:  
RULE:[2:\$1@\$0] (dn@COMPANY.DIV.COM)s/.\*/root/  
RULE:[2:\$1@\$0] (nn@COMPANY.DIV.COM)s/.\*/root/

Restart the HDFS service in Ambari or HDFS Transparency by using the following command:

/usr/lpp/mmfs/bin/mmhadoopctl connector stop; /usr/lpp/mmfs/bin/mmhadoopctl connector start

## 20. 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:

- Restart each node's Ambari agent by issuing the following command:

ambari-agent restart

- Restart HDFS service from Ambari.

## 21. 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.

## 22. 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.

## 23. **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.

## 24. Enable the autostart of services when IBM Spectrum Scale is integrated.

### Solution:

- In Ambari GUI, go to **Admin > Service Auto Start Configuration** and enable autostart.

b. Enable autoload and automount on the IBM Spectrum Scale cluster (on the HDP cluster side).

c. If ESS is being used, enable autoload on the ESS cluster.

For more information, see IBM Spectrum Scale `mmchfs fsname -A yes` for automount and `mmchconfig autoload=yes` commands.

25. 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:**

- 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.
- 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.

<sup>1</sup>For FPO cluster, do not run STOP ALL from the Ambari GUI. Refer to the Limitations > General sections on how to properly stop IBM Spectrum Scale.

## Service fails to start

- 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 -efaf | grep zkfc
```

If the zkfc process is up, kill the zkfc process from the Namenode host by running the `kill -9` command on the pid.

- Once the zkfc process is not running in any Namenode host, go into the HDFS service dashboard and do a **Start the HDFS service**.

- In non-root Ambari environment, IBM Spectrum Scale fail to start due to NFS mount point not being accessible by root.

**Solution:** For example, the /usrhome/am\_agent is a NFS mount point with permission set to 700. The following error is seen:

```
2017-04-04 15:42:49,901 - ===== Check for changes to the configuration. =====
2017-04-04 15:42:49,901 - Updating remote.copy needs service reboot.
2017-04-04 15:42:49,901 - Values don't match for gpfs.remote.copy. running_config[gpfs.remote.copy]:
 sudo wrapper in use; gpfs_config[gpfs.remote.copy]: /usr/bin/scp
2017-04-04 15:42:49,902 - Updating remote.shell needs service reboot.
2017-04-04 15:42:49,902 - Values don't match for gpfs.remote.shell. running_config[gpfs.remote.shell]:
 /usr/lpp/mmfs/bin/sshwrap; gpfs_config[gpfs.remote.shell]: /usr/bin/ssh
2017-04-04 15:42:49,902 - Shutdown all gpfs clients.
2017-04-04 15:42:49,902 - Run command: sudo /usr/lpp/mmfs/bin/mmshutdown -N k-001,k-002,k-003,k-004
```

```
| 2017-04-04 15:44:03,608 - Status: 0, Output:
| Tue 4 Apr 15:42:50 CEST 2017: mmshutdown: Starting force unmount of GPFS file systems
| k-003.gpfs.net: mmremote: Invalid current working directory detected: /usrhome/am_agent
| To resolve this issue: Change the permissions of the home directory of the GPFS non-root user to at
| least 711.
| Example: For the /usrhome/am_agent directory, set the directory with at least a 711 permission set or
| rwx--x-x.
| Where, 7= rwx for the user itself, 1= x for the group, 1= x for others; x will allow users to cd into the
| home directory.
| This is because the IBM Spectrum Scale command does a cd into the home directory of the user.
| Therefore, the permission should be set to at least 711.
```

### 3. Accumulo Tserver failed to start.

**Solution:** Accumulo Tserver might go down. Ensure that the block size is set to the IBM Spectrum Scale file system value.

- In **Accumulo > Configs > Custom accumulo-site**, set the *tserver.wal.blocksize* to <GPFS File system block size of the data pool>.

For example, *tserver.wal.blocksize* = 2097152.

```
[root@c902f05x04 ~]# mm1sfs /dev/bigpfs -B
flag value description
```

```

B 262144 Block size (system pool)
2097152 Block size (other pools)
[root@c902f05x04 ~]#
```

- Restart Accumulo service.
- Run Accumulo service check.

From **Ambari GUI > Accumulo > Service Actions > Run Service Check**.

For additional failures, see What to do when the Accumulo service start or service check fails?.

### 4. Hive fails to start, with the default HDP and Ambari recommendations

**Solution:**

There is bug in HDP3.0 - <https://hortonworks.jira.com/projects/SPEC/issues/SPEC-18> which causes Accumulo to use the same port number as HiveServer2 leading to port binding conflict. As a workaround, use the following configuration:

- Put accumulo and hiveserver2 on different hosts or
- Use non-default port for either of them.

### 5. Kafka service fails to start if Kafka is added after Spectrum Scale.

**Solution:**

Check the Kafka configuration log directory to see if the Kafka log directory contains the Spectrum Scale shared mount point (/<gpfs mount point>/kafka-logs). Remove the share mount point from the directory list and restart the Kafka service. For more information, see “Adding Services” on page 146.

### 6. HBase service fails to start.

**Solution:**

In Spectrum Scale integrated, Hbase Master fails to start or goes down. This could be because of stale znodes in Zookeeper created for Hbase.

To clean znodes of Hbase, do the following

- Login to zookeeper by executing the following command:

```
/usr/hdp/current/zookeeper-server/bin/zkCli.sh -server <any zookeeper hostname>
```

- b. rmr /hbase-unsecure or rmr /hbase-secure (depending on kerberos enabled/disabled).

### 7. Start All services fails because **zkfc** fails to start. Therefore, putting the Namenodes into standby mode.

The sequence of steps when this error occurs is to enable Namenode HA in native HDFS then integrate IBM Spectrum Scale service to use HDFS Transparency and later enable Kerberos. During the period when Kerberos is enabling, **zkfc** fails to start. This leads to both Namenodes being in the standby mode. Therefore, HDFS cannot be used. As a result, the **Start All services** fails.

**Solution:**

Restart HDFS or restart all services from Ambari.

8. Namenode and Datanodes failed to start with mapreduce.tar.gz error.

For Mpack version 2.7.0.0, the Namenode and Datanode might fail to start with the following error message when the data directory (`gpfs.data.dir`) is specified through the Ambari Spectrum Scale UI: Failed to replace mapreduce.tar.gz with Transparency jars.

**Solution:**

Follow these steps to set the mapreduce.tar.gz into a proper directory for the Namenode/Datanode to start: Check if the `/<gpfs.mnt.dir>/<gpfs.data.dir>/hdp/apps/<hdp-version>/mapreduce/` directory exists. If not, create the directory by running the following command:

```
mkdir -p /<gpfs.mnt.dir>/<gpfs.data.dir>/hdp/apps/<hdp-version>/mapreduce/
```

Copy the mapreduce.tar.gz from HDP to the Scale directory by running the following command:

```
cp /usr/hdp/<hdp-version>/hadoop/mapreduce.tar.gz
/<gpfs.mnt.dir>/<gpfs.data.dir>/hdp/apps/<hdp-version>/mapreduce/mapreduce.tar.gz
```

where, `<gpfs.mnt.dir>` is the Spectrum Scale mount point `<gpfs.data.dir>` is the Spectrum Scale data directory `<hdp-version>` is the HDP version and can be obtained by running `hdp-select versions`. For example,

```
mkdir -p /bigpfs/datadir1/hdp/apps/2.6.5.0-292/mapreduce/
cp /usr/hdp/2.6.5.0-292/hadoop/mapreduce.tar.gz
/bigpfs/datadir1/hdp/apps/2.6.5.0-292/mapreduce/mapreduce.tar.gz
```

Restart the failed HDFS components.

9. The zookeeper failover controller (ZKFC) fails during the **Start All** operation after integrating IBM Spectrum Scale service with Namenode High Availability for the first time.

There is a timing issue during the formatting of the zookeeper directory which is shared by both ZKFC in HA mode on which ZKFC should be started first.

**Solution:**

Rerun the **Start All** operation to get the services back up.

10. The zkfc fails to start when Kerberos is enabled.

The zkfc might fail to start with Can't set priority for process error if IBM Spectrum Scale is first added to an HA enabled HDP cluster before adding Kerberos. The `hdfs_jaas.conf` file might not be generated during the kerberos enablement action.

**Solution:**

- a. Create the `hdfs_jaas.conf` file in the `/etc/hadoop/conf/secure` directory, on both the Namenodes.

For example,

```
cat /etc/hadoop/conf/secure/hdfs_jaas.conf
```

```
Client {
 com.sun.security.auth.module.Krb5LoginModule required
 useKeyTab=true
 storeKey=true
 useTicketCache=false
 keyTab="/etc/security/keytabs/nn.service.keytab"
 principal="nn/c902f09x13.gpfs.net@IBM.COM";
};
```

**Note:** Ensure that you change the keyTab and principal values based on your environment.

- b. If `/etc/hosts` is used for hostname resolution instead of DNS in your environment, use the FQDN hostname in `/etc/hosts`.

- Ensure that the output from the command **hostname** matches the following:
  - 1) Hostname specified in the Ambari wizard.
  - 2) IP/hostname used for DNS.
- Check the same for all the hosts in the cluster and restart HDFS.

## 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.nodemanager.local-dirs** and **yarn.nodemanager.log-dirs** are writable for the user yarn. If this is not the case, add write permission to the **yarn.nodemanager.local-dirs** and **yarn.nodemanager.log-dirs** directories for the user yarn.

2. What to do when the Accumulo service start or service check fails?

**Solution:**

**Note:**

- Ensure that the HDFS and Zookeeper services are running before you proceed.
- If it is non root environment, run the commands in the workaround steps after logging in with non root user only.
- If GPFS is unintegrated, remove the **tserver.wal.blocksize** entry from Accumulo. From Ambari, go to **Accumulo > Configs > Custom accumulo-site** and remove the **tserver.wal.blocksize** value and save the configuration.
- If GPFS is integrated, follow the workaround for **tserver.wal.blocksize** as mentioned in the FAQ Accumulo Tserver failed to start.

If the problem still exists, contact IBM service.

3. Atlas service check fails.

**Solution:**

- a. Restart the Ambari Infra service.
- b. Restart the Hbase service.
- c. Re-run the Atlas service check.

4. Falcon service check fails.

**Solution:**

This is a known issue with HDP. For information on resolving this issue, go to Falcon Web UI is inaccessible(HTTP 503 error) and Ambari Service Check for Falcon fails: "ERROR: Unable to initialize Falcon Client object".

5. What to do when the Hive service check fails with the following error:

```
Templeton Smoke Test (ddl cmd): Failed. : {"error":"Unable to access program: /usr/hdp/${hdp.version}/hive/bin/hcat"}http_code <401>
```

**Solution:**

HDP is not able to properly parse the  **\${hdp.version}** value. To set the HDP version, execute the following steps:

- | a. Get the HDP version for your environment by running the **/usr/bin/hdp-select versions** command on any Ambari node.
- | b. In the Ambari GUI, click **HIVE > Configs**. Find the **templeton.hcat** field under the **Advanced webhcat-site**.
- | c. Replace the  **\${hdp.version}** in the **templeton.hcat** field with the hardcoded **hdp.version** value found in step a.  
|     For example, if the value of **hdp.version** is 2.6.5.0-292, set the **templeton.hcat** value from **/usr/hdp/\${hdp.version}/hive/bin/hcat** to **/usr/hdp/2.6.5.0-292/hive/bin/hcat**.
- | d. Restart the HIVE service components.
- | e. Re-run the HIVE service check.

---

# Chapter 5. Open Source Apache Hadoop

## Apache Hadoop 3.0.x Support

Apache Hadoop 3.0.x is supported with HDFS Transparency 3.0.0. When you use Apache Hadoop, the configuration files of HDFS Transparency are located under /var/mmfs/hadoop/etc/hadoop. By default, the logs of HDFS Transparency are located under /var/log/transparency/.

If you want to run Apache Hadoop with HDFS Transparency 3.0, execute the following steps:

1. Set **ulimit nofile** to 64K on all the nodes.
2. Set up the ntp server to synchronize the time on all nodes.
3. Root password-less access from NameNodes to all other DataNodes.  
For more details, refer “Setting password-less ssh access for root” on page 141.
4. Install HDFS Transparency 3.0.0-x (gpfs.hdfs-protocol-3.0.0-x.<arch>.rpm) on all HDFS Transparency nodes
5. **ssh** to TransparencyNode1.
6. Update the /var/mmfs/hadoop/etc/hadoop/core-site.xml with your NameNode hostname.

```
<configuration>
 <property>
 <name>fs.defaultFS</name>
 <value>hdfs://c8f2n04:8020</value>
 </property>
</configuration>
```

7. Update the /var/mmfs/hadoop/etc/hadoop/hdfs-site.xml according to your configuration:

Configuration	Default	Recommendation	Comment
<b>dfs.replication</b>	1	1 or 2 or 3	Check your file system by <b>mm1sfs &lt;fs-name&gt; -r</b> and update this configuration according to the value from <b>mm1sfs</b> .
<b>dfs.blocksize</b>	N/A	134217728 or 268435456 or 536870912	Usually, 128MB (134217728) is used and 512MB (536870912) might be used for IBM ESS file system.
<b>dfs.client.read.shortcircuit</b>	false	true	Refer “Short-circuit read (SSR)” on page 198

For other configurations, take the default value.

8. Update the /var/mmfs/hadoop/etc/hadoop/gpfs-site.xml for the gpfs.mnt.dir, gpfs.data.dir and gpfs.storage.type configurations.
9. Update the /var/mmfs/hadoop/etc/hadoop/hadoop-env.sh and change **export JAVA\_HOME=** into **export JAVA\_HOME=<your-real-JDK8-Home-Dir>**.
10. Update the /var/mmfs/hadoop/etc/hadoop/workers to add DataNodes. One DataNode hostname per line.
11. Synchronize all these changes into other DataNodes by executing the following command:  
`/usr/lpp/mmfs/bin/mmhadoopctl connector syncconf /var/mmfs/hadoop/etc/hadoop`
12. Start HDFS Transparency by executing **mmhadoopctl**:  
`/usr/lpp/mmfs/bin/mmhadoopctl connector start`
13. Check the service status of HDFS Transparency by executing **mmhadoopctl**:

```
/usr/lpp/mmfs/bin/mmhadoopctl connector getstate
```

**Note:** If HDFS Transparency is not up on some nodes, login to those nodes and check the logs located under /var/log/transparency. If you do not get any errors, HDFS Transparency should be up by now.

If you want to configure the Yarn, execute the following steps:

- a. Download Apache Hadoop 3.0.x from Apache Hadoop website.
- b. Unzip the packages to /opt/Hadoop-3.0.x on HadoopNode1.
- c. Login to HadoopNode1.
- d. Copy the **hadoop-env.sh**, **hdfs-site.xml**, and workers from /var/mmfs/hadoop/etc/hadoop on HDFS Transparency node to HadoopNode1:/opt/hadoop-3.0.x/etc/hadoop/.
- e. Copy /usr/lpp/mmfs/hadoop/template/mapred-site.xml.template and /usr/lpp/mmfs/hadoop/template/yarn-site.xml.template from HDFS Transparency node into HadoopNode1:/opt/hadoop-3.0.x/etc/hadoop as mapred-site.xml and yarn-site.xml.
- f. Update /opt/hadoop-3.0.x/etc/hadoop/mapred-site.xml with the correct path location for **yarn.app.mapreduce.am.env**, **mapreduce.map.env**, and **mapreduce.reduce.env** configurations.

For example, change the value from HADOOP\_MAPRED\_HOME=/opt/hadoop-3.0.2 to HADOOP\_MAPRED\_HOME=/opt/hadoop-3.0.x

**Note:** /opt/hadoop-3.0.x is the real location for Hadoop.

- g. Update /opt/hadoop-3.0.x/etc/hadoop/yarn-site.xml. Especially configuring the correct hostname for **yarn.resourcemanager.hostname**.
- h. Synchronize /opt/hadoop-3.0.x from HadoopNode1 to all other Hadoop nodes and keep the same location for all hosts.
- i. On the Resource Manager node, run the following command to start the Yarn service:

```
#cd /opt/hadoop-3.0.x/sbin/
#export YARN_NODEMANAGER_USER=root
#export YARN_RESOURCEMANAGER_USER=root
#/start-yarn.sh
```

**Note:** By default, the logs for Yarn service will be under /opt/hadoop-3.0.x/logs. If you plan to start Yarn services with other user name, you could change the user root in the above command to your target user name.

- j. Run the following command to submit word count jobs:

```
#/opt/hadoop-3.0.x/bin/hadoop dfs -put /etc/passwd /passwd
#/opt/hadoop-3.0.x/bin/hadoop
jar /opt/hadoop-3.0.x/share/hadoop/mapreduce/hadoop-mapreduce-examples-3.0.2.jar
wordcount /passwd /results
```

The Yarn service works well if the word count job executed successfully.

Refer Chapter 3, “IBM Spectrum Scale Hadoop performance tuning guide,” on page 109.

---

# **Chapter 6. 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 Mpact and HDFS Transparency. The IBM Spectrum Scale file system is independent of the versioning for the Mpact and HDFS Transparency. Hortonworks has been certified to work with IBM Spectrum Scale 4.2.3 and later.

*Table 25. Hadoop distribution support matrix*

IBM Spectrum Scale Ambari management pack (Mpack)	HDFS Transparency	Hadoop distribution	Ambari version	RHEL x86_64	RHEL ppc64le
Mpack 2.4.2.7	HDFS Transparency 2.7.3-4	HDP 2.6.5 HDP 2.6.4	Ambari 2.6.2.0 Ambari 2.6.1.0	7.2/7.3/7.4/7.5	7.2/7.3/7.4/7.5
Mpack 2.4.2.6	HDFS Transparency 2.7.3-1+	HDP 2.6.5	Ambari 2.6.2.0	7.2/7.3/7.4/7.5	7.2/7.3/7.4/7.5
Mpack 2.4.2.5	HDFS Transparency 2.7.3-1+	HDP 2.6.5	Ambari 2.6.2.0	7.2/7.3/7.4/7.5	7.2/7.3/7.4/7.5
Mpack 2.4.2.4	HDFS Transparency 2.7.3-1+	HDP 2.6.4	Ambari 2.6.1.0	7.2/7.3/7.4/7.5	7.2/7.3/7.4/7.5
Mpack 2.4.2.3	HDFS Transparency 2.7.3-1+	HDP 2.6.3	Ambari 2.6.0.0	7.2/7.3/7.4/7.5	7.2/7.3/7.4/7.5
Mpack 2.4.2.2	HDFS Transparency 2.7.3-1+	HDP 2.6.2	Ambari 2.5.2.0	7.2/7.3/7.4/7.5	7.2/7.3/7.4/7.5
Mpack 2.4.2.1	HDFS Transparency 2.7.3-1+	HDP 2.6.2	Ambari 2.5.2.0	7.2/7.3/7.4/7.5	7.2/7.3/7.4/7.5
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	7.2/7.3/7.4/7.5	7.2/7.3/7.4/7.5
Mpack 2.4.2.1	HDFS Transparency 2.7.3-1+	BI 4.2.5 <sup>1</sup>	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. Mpack 2.4.2.1 is the last supported release for BI 4.2.5.
2. 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 <module> from resource_management.libraries.functions.get_lzo_packagesimport get_lzo_packages
ImportError: No module named get_lzo_packages
```

- Preserving Kerberos token delegation during Namenode failover with HDP is supported when Mpack 2.4.2.7 and HDFS Transparency 2.7.3-4 are applied together.

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 26. 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.

*Table 26. Preferred Simple Format and Standard Format (continued)*

Preferred Simple Format	Standard Format
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 /var/lib/ambari-server/resources/ on the Ambari server.	
Only /dev/sdX and /dev/dx-X 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/sdg
DISK compute002.private.dns.zone:/dev/sdb,/dev/sdc,/dev/sdd,/dev/sde,/dev/sdf,/dev/sdg
DISK compute003.private.dns.zone:/dev/sdb,/dev/sdc,/dev/sdd,/dev/sde,/dev/sdf,/dev/sdg
DISK compute005.private.dns.zone:/dev/sdb,/dev/sdc,/dev/sdd,/dev/sde,/dev/sdf,/dev/sdg
DISK compute006.private.dns.zone:/dev/sdb,/dev/sdc,/dev/sdd,/dev/sde,/dev/sdf,/dev/sdg
DISK compute007.private.dns.zone:/dev/sdb,/dev/sdc,/dev/sdd,/dev/sde,/dev/sdf,/dev/sdg
```

If you want to select disks such as SSD drives for metadata , add the label -meta to those disks.

In a simple NSD file, add the label meta for the disks that you want to use as metadata disks, as shown in the following example. If -meta is used, the Partition algorithm is ignored.

```
$ cat /var/lib/ambari-server/resources/gpfs_nsd
```

```
DISK compute001.private.dns.zone:/dev/sdb-meta,/dev/sdc,/dev/sdd
DISK compute002.private.dns.zone:/dev/sdb-meta,/dev/sdc,/dev/sdd
DISK compute003.private.dns.zone:/dev/sdb-meta,/dev/sdc,/dev/sdd
DISK compute005.private.dns.zone:/dev/sdb-meta,/dev/sdc,/dev/sdd
DISK compute006.private.dns.zone:/dev/sdb,/dev/sdc,/dev/sdd
DISK compute007.private.dns.zone:/dev/sdb,/dev/sdc,/dev/sdd
```

In the simple NSD file, /dev/sdb from compute001, compute002, compute003, and compute005 are specified as meta disks in the IBM Spectrum Scale file system.

The partition algorithm is ignored if the nodes listed in the simple NSD file do not match the set of nodes that will be used for the NodeManager service. If nodes that are not NodeManagers are in the NSD file or nodes that will be NodeManagers are not in the NSD file, no partitioning will be done.

## Standard NSD file

This section describes a standard NSD file with an example.

The following is an example of a Standard IBM Spectrum Scale NSD File

```
%pool: pool=system blockSize=256K layoutMap=cluster allowWriteAffinity=no
%pool: pool=datapool blockSize=2M layoutMap=cluster allowWriteAffinity=yes writeAffinityDepth=1
blockGroupFactor=256
```

```

gpfstest9
%nsd: nsd=node9_meta_sdb device=/dev/sdb servers=gpfstest9 usage=metadataOnly failureGroup=101 pool=system
%nsd: nsd=node9_meta_sdc device=/dev/sdc servers=gpfstest9 usage=metadataOnly failureGroup=101 pool=system

%nsd: nsd=node9_data_sde2 device=/dev/sde2 servers=gpfstest9 usage=dataOnly failureGroup=1,0,1 pool=datapool
%nsd: nsd=node9_data_sdf2 device=/dev/sdf2 servers=gpfstest9 usage=dataOnly failureGroup=1,0,1 pool=datapool

gpfstest10
%nsd: nsd=node10_meta_sdb device=/dev/sdb servers=gpfstest10 usage=metadataOnly failureGroup=201 pool=system
%nsd: nsd=node10_meta_sdc device=/dev/sdc servers=gpfstest10 usage=metadataOnly failureGroup=201 pool=system

%nsd: nsd=node10_data_sde2 device=/dev/sde2 servers=gpfstest10 usage=dataOnly failureGroup=2,0,1 pool=datapool
%nsd: nsd=node10_data_sdf2 device=/dev/sdf2 servers=gpfstest10 usage=dataOnly failureGroup=2,0,1 pool=datapool

gpfstest11
%nsd: nsd=node11_meta_sdb device=/dev/sdb servers=gpfstest11 usage=metadataOnly failureGroup=301 pool=system
%nsd: nsd=node11_meta_sdc device=/dev/sdc servers=gpfstest11 usage=metadataOnly failureGroup=301 pool=system

%nsd: nsd=node11_data_sde2 device=/dev/sde2 servers=gpfstest11 usage=dataOnly failureGroup=3,0,1 pool=datapool
%nsd: nsd=node11_data_sdf2 device=/dev/sdf2 servers=gpfstest11 usage=dataOnly failureGroup=3,0,1 pool=datapool

```

**Note:** Starting with IBM Spectrum Scale version 5.0.0, the default block size is 4M.

Type the /var/lib/ambari-server/resources/gpfs\_nsd filename in the NSD stanza field.

Because of the limitations of the Ambari framework, the NSD file must be copied to the Ambari server in the /var/lib/ambari-server/resources/ directory. Ensure that the correct file name is specified on the IBM Spectrum Scale Customize Services page.

If you are using a standard NSD stanza file and only one datapool is defined, you can either specify the policy file or let it be done by IBM Spectrum Scale. However, if you have more than one data pool, you should specify a policy to define the location of the data in the data pool. If there is no policy specified, by default the data will be stored to the first data pool only.

## Policy File

This section describes a policy file with an example.

The bigpfs.pol is an example of a policy file.

```
RULE 'default' SET POOL 'datapool'
```

---

## Installation

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

This installation section will describe how to install ESS Remote mount with all Hadoop nodes as Spectrum Scale nodes as the 1st deployment model as described in the Chapter 1, “Hadoop Scale Storage Architecture,” on page 1 section.

Before starting the software deployment, follow the Planning section to setup the environment and download the software.

**Note:**

- This chapter describes how to add IBM Spectrum Scale service as a root user. If you plan to restrict root access, review the “Restricting root access” on page 327 section.
- To install the IBM Spectrum Scale service, an existing HDFS cluster is required. This can be created by installing the HDP stack with native HDFS.

For other installation scenarios, see “Configuration” on page 167 section.

## 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 the *Product overview* section in the *IBM Spectrum Scale: Concepts, Planning, and Installation Guide*.

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 trial 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 Mpack 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 <OLDDUID> -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. If you had already downloaded and installed an Mpack in a specific directory before, then download and unzip the new management pack into a different directory than the currently installed version.

**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 go 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/java-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
$ mmcrlfileset 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

```

- 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/*

```

- For each node, link a local directory to its corresponding node directory named with its host name:

```

for host in <your host name list>
do
 echo "$host"
 mmdsh -N $host "ln -s /bigpfs/hadoop/local/$host /hadoop/yarn/local"
 mmdsh -N $host "chown -R yarn:hadoop /hadoop/yarn/local"

```

```

// If additional user created directories are configured under Yarn in the shared storage, then ensure to
// create the corresponding user created directories in the local host and link them to the share storage
// directories.

```

```

// For example: yarn.nodemanager.log-dirs is set to /hadoop/yarn/log
// mmdsh -N $host "mkdir -p /hadoop/yarn/local/log"
// mmdsh -N $host "chown -R yarn:hadoop /hadoop/yarn/log"
done

```

**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.

- 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.

- Stop all services:

Log into Ambari. Click **Actions > Stop All**.

- 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.
- If you are using a simple NSD stanza file to create a new FPO cluster, the disk partitioning under Ambari would happen only if the following conditions are met:
  - 1) The NSD stanza requires all GPFS nodes to be specified in the NSD stanza file.
  - 2) Each of those nodes should have the same number of disks specified in the stanza file.
  - 3) Number of host entries in the stanza file/NSD servers should be equal to the number of Node managers. This requires all hosts running GPFS node to be set up as a Node manager too. If you do not want Hadoop jobs to run on a specific GPFS node host (For example, on the Ambari server host), you could remove the Node manager component from that host after deploying the IBM Spectrum scale service.

For more details on disk partitioning, see the following:

- Disk-partitioning algorithm.
- Partitioning function matrix in automatic deployment.
- 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](http://c902mnx09-ug.pok.stglabs.ibm.com/repos/GPFS/4.2.2.3/gpfs_rpms)

This directory should contain the HDFS Transparency rpm.

- 5) From Ambari Mpack version 2.4.2.2 a new GUI field **Verify if the disks are already formatted as NSDs:** is added. A value of *Yes* specifies that the NSDs are to be created only if each disk has not been formatted as a NSD by a previous invocation of the **mmcrnsd** command. The default value is *Yes*. A value of *No* specifies that the disks are to be created irrespective of their previous state.

**Important:** Setting this field to *No* would result in data stored over those NSDs being erased during the service deployment process. Therefore, set this field to *No* only if it is intended.

- 6) **gpfs.supergroup** configuration: If you plan to install BigSQL, see Setting IBM Spectrum Scale configuration for BigSQL to set the correct **gpfs.supergroup** value.

**Note:** Follow the deployment model section based on your environment.

For New FPO, go to step 7.

For Remote cluster mount, go to step 8.

For Multiple file system, go to step 9.

For Pre-existing file system, go to step 10.

- 7) 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.pol`, must be created and placed into the `/var/lib/ambari-server/resources` directory. Add the policy filename, `gpfs_fs.pol`, into the **GPFS policy file** field in the Standard Config tab.

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.

- 8) 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.
- 9) 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.
- 10) 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/sd1,/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/sd1,/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/sd1,/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/sd1,/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/sd1,/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.

11) 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.

12) 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. Check the `mmlsfs -r` replication value and update the HDFS config panel **dfs.replication** field in Ambari if the value does not match the `mmlsfs` command output. For more information, see *mmlsfs* command in *IBM Spectrum Scale: Command and Programming Reference*.

**Note:** For IBM ESS file system, if the file system replication is set to 1, the HDFS **dfs.replication** field should be set as 1 in the HDFS config panel in Ambari to ensure that the services do not get misleading error message for not having enough space. For example: spark2 thrift server down due to do not have enough space error.

- b. 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.
- c. 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**.

- d. 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.
  - e. 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**.

7. 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.

1. 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>
```

2. One or more file systems from the ESS are already mounted onto the local IBM Spectrum Scale cluster. See the *Mounting a remote GPFS file system* topic in the *IBM Spectrum Scale: Administration Guide* topic.

**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 **mmremotecluster show all** command.

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

```
[root@mn01 ~]# /usr/lpp/mmfs/bin/mmremoteefs 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*.
4. 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

gpfs.storage.type:	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:

<b>gpfs.storage.type</b>	<b>GPFS Filesystem</b>	<b>gpfs.mnt.dir</b>
remote	GPFS FileSystem Name essfs	<del>essfs</del>
GPFS NSD stanza file put in /var/lib/ambari-server/resources/		
<b>gpfs.replica.enforced</b>	GPFS policy file for file system put in /var/lib/ambari-server/resources/	
dfs		

b1adv256

- 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.

- 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 18 and “Cluster and file system information configuration” on page 24 sections.

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 as root.

- 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 18 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
<b>gpfs.storage.type</b>	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: <code>gpfs.storage.type=local,remote</code> <code>gpfs.storage.type=remote,remote</code>

Fields	Description
<b>gpfs.mnt.dir</b>	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.
<b>gpfs.replica.enforced</b>	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.
<b>gpfs.data.dir</b>	Only one value must be specified. Null is a valid value. The data directory is created only for the primary file system.
<b>GPFS FileSystem Name</b>	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/1pp/mmfs/bin/mmlsmount 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 /remoteefs.

Setting of the fields would be as follows:

```
gpfs.storage.type=local,remote
gpfs.mnt.dir=/localfs,/remoteefs
gpfs.replica.enforced=dfs,dfs
gpfs.data.dir=myDataDir OR gpfs.data.dir=
GPFS FileSystem Name=localfs,remoteefs
```

- 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 `/remoteefs`.

- 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 `/remoteefs`.

Setting of the fields would be as follows:

```
gpfs.storage.type=local,remote
gpfs.mnt.dir=/localfs,/remoteefs
gpfs.replica.enforced=dfs,dfs
gpfs.data.dir=myDataDir OR gpfs.data.dir=
GPFS FileSystem Name=localfs,remoteefs
```

**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 /remoteefs1 and the secondary file system mount point is /remoteefs2.

Setting of the fields would be as follows:

```
gpfs.storage.type=remote,remote
gpfs.mnt.dir=/remoteefs1,/remoteefs2
gpfs.replica.enforced=dfs,dfs
gpfs.data.dir=myDataDir OR gpfs.data.dir=
GPFS FileSystem Name=remoteefs1,remoteefs2
```

- 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.AHSProxy: 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.

**All Applications**

ID	User	Name	Type	Queue	StartTime	FinishTime	State	FinalStatus	Progress	Tracking URL	Blacklisted Nodes
application_1496256368436_0001	ambari-qa	word count	MAPREDUCE	default	Wed May 31 14:51:31 -0400 2017	2017-05-31 14:51:52 -0400	FINISHED	SUCCEEDED	N/A	<a href="#">History</a>	N/A

**Application application\_1496256368436\_0001**

**Application Overview**

User: ambari-qa  
 Name: word count  
 Application Type: MAPREDUCE  
 Application Tags:  
 YarnApplicationState: FINISHED  
 Queue: default  
 FinalStatus reported by AM: SUCCEEDED  
 Started: Wed May 31 14:51:31 -0400 2017  
 Elapsed: 21sec  
 Tracking URL: [History](#)  
 Diagnostics:  
 Application Node Label expression: <Not set>  
 AM container Node Label expression: <DEFAULT\_PARTITION>

**Application Metrics**

Total Resource Preempted: <memory:0, vCores:0>  
 Total Number of Non-AM Containers Preempted: 0  
 Total Number of AM Containers Preempted: 0  
 Resource Preempted from Current Attempt: <memory:0, vCores:0>  
 Number of Non-AM Containers Preempted from Current Attempt: 0  
 Aggregate Resource Allocation: 369771 MB-seconds, 36 vcore-seconds

Attempt ID	Started	Node	Logs	Blacklisted Nodes
appattempt_1496256368436_0001_000001	Wed May 31 14:51:31 -0400 2017	<a href="http://c902f05x02.gpfs.net:8042">http://c902f05x02.gpfs.net:8042</a>	Logs	N/A

- 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.

- 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.

- 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 Mpak and license bin executables.

The SpectrumScaleMPackUninstaller.py script verifies the settings before it uninstalls the Mpak 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 **bigsq1\_user\_password** must be set to *bigsq1*.
- 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 *bigsq1-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.

## Upgrading IBM Spectrum Scale service MPack

This section describes the IBM Spectrum Scale MPack upgrade process.

You must plan a cluster maintenance window and prepare for cluster downtime when you upgrade the IBM Spectrum Scale MPack.

### Note:

- You must perform the Mpack upgrade only if the target Mpack version is supported on your HDP level. Ensure that you check the support matrix and verify whether the Mpack version is supported with your HDP level.
- To see the default configuration modifications under Spectrum Scale Mpacks, refer to the Big data and analytics section under the Big Data and Analytics - summary of changes section.
- The cluster must be at management pack version 2.4.2.0 or later.

- 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.
- Ensure that the anonymous user id is created and have the same uid/gid in your cluster before upgrading. From Mpack 2.4.2.6, having an anonymous user id is mandatory. For more information, see the “Create the anonymous user id” on page 138 topic.

### Procedure

- As the root user, download a management pack at a higher PTF version, than the version of IBM Spectrum Scale service installed on your system, onto a directory on the Ambari server node. For information on downloading the management packs, see the “IBM Spectrum Scale service” on page 253 topic.

**Note:** The downloaded management pack should be stored and unzipped in a directory different than the currently installed version of the Mpack.

In this example, the downloaded management pack has been downloaded in the /root/GPFS\_Ambari/upgrade\_Mpack directory. The management pack contains the upgrade script to upgrade the MPack.

For example, if the currently installed Mpack is at 2.4.2.0 version then plan to upgrade to Mpack 2.4.2.1 version.

The **SpectrumScale\_UpgradeIntegrationPackage** script used for the upgrade and migration is run from the /root/GPFS\_Ambari/upgrade\_Mpack directory.

Ensure that the current Mpack installable package resides on a separate directory on the Ambari server node. This example uses the /root/GPFS\_Ambari/currently\_installed\_Mpack directory.

The **SpectrumScaleMPackUninstaller.py** script used as part of this procedure would have to be run from the /root/GPFS\_Ambari/currently\_installed\_Mpack directory.

- Log in to Ambari.
- Stop all the services. Click **Ambari > Actions > Stop All**<sup>1</sup>.

<sup>1</sup>For FPO cluster, do not run STOP ALL from the Ambari GUI. Refer to the Limitations > General sections on how to properly stop IBM Spectrum Scale.

- 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.

- If the IBM Spectrum Scale service is not already stopped, stop the IBM Spectrum Scale service by clicking **Ambari > Spectrum Scale > Service Actions > Stop**.
- As the root user on the Ambari server node, from the /root/GPFS\_Ambari/upgrade\_Mpack directory, run the **SpectrumScale\_UpgradeIntegrationPackage** script with the **--preEU** option.

The **--preEU** option saves the existing IBM Spectrum Scale service information into JSON files in the local directory where the script was run. It also removes the IBM Spectrum Scale service from the Ambari cluster so that the BI cluster can be properly migrated. This does not affect the IBM Spectrum Scale file system.

Before you proceed, review the following questions for the upgrade script and have the information for your environment handy. If Kerberos is enabled, more inputs are required.

```
$ cd /root/GPFS_Ambari/upgrade_Mpack
$./SpectrumScale_UpgradeIntegrationPackage --preEU
Are you sure you want to upgrade the GPFS Ambari integration package (Y/N)? (Default Y):

STARTING WITH SPECTRUM SCALE EXPRESS UPGRADE PRE STEPS

Enter the Ambari server User:(Default admin):
Enter the password for the Ambari server.
Password:
Retype password:
```

```

SSL Enabled (True/False) (Default False):
Enter the Ambari server Port. (Default 8080):
...
Note: If Kerberos is enabled, then the KDC principal and password information are required.
Kerberos is Enabled. Proceeding with Configuration
Enter kdc principal:
Enter kdc password:
...
| 7. As a root user on the Ambari server, run the MPack uninstaller script,
| SpectrumScaleMPackUninstaller.py, from the currently installed Mpack directory to remove the
| existing MPack link in Ambari.
| $ cd /root/GPFS_Ambari/currently_installed_Mpack
| $./SpectrumScaleMPackUninstaller.py
| INFO: ***Starting the MPack Uninstaller***

| Enter Ambari Server Port Number. If it is not entered, the uninstaller will take default port 8080:
| INFO: Taking default port 8080 as Ambari Server Port Number.
| Enter Ambari Server IP Address : 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. 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.

| 9. On the Ambari server node as root, from the /root/GPFS\_Ambari/upgrade\_Mpack directory, run the **SpectrumScale\_UpgradeIntegrationPackage** script with the **--postEU** option in the directory where the **--preEU** step was run and where the JSON configurations were stored.

Before you proceed, for the **--postEU** option, review the following questions and have the information for your environment handy. If Kerberos is enabled, more inputs are required.

```

$ cd /root/GPFS_Ambari/upgrade_Mpack
$./SpectrumScale_UpgradeIntegrationPackage --postEU
Are you sure you want to upgrade the GPFS Ambari integration package (Y/N)? (Default Y):

STARTING WITH SPECTRUM SCALE EXPRESS UPGRADE POST STEPS

Starting Post Express Upgrade Steps. Enter Credentials
Enter the Ambari server User:(Default admin):
Enter the password for the Ambari server.
Password:
Retype password:
SSL Enabled (True/False) (Default False):
Enter the Ambari server Port. (Default 8080):
....
Accept License
Do you agree to the above license terms? [yes or no]
yes

```

```

Installing...
Enter Ambari Server Port Number. If it is not entered, the installer will take default port 8080 :
INFO: Taking default port 8080 as Ambari Server Port Number.
Enter Ambari Server IP Address :
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.

```

**Note:** If the Mpack requires a corresponding HDFS Transparency update version, ensure that the process in the “Upgrading HDFS Transparency” topic is done before doing a **Start All** in the next step.

#### 10. Start all the services.

Click **Ambari > Actions > Start All**.

Restart all the components by using the restart icon.

**Note:**

- If the **Start All** fails, try starting each of the services individually. Ensure that the manual starting services in the Ambari order is executed first. For more information, see the “Manually starting services in Ambari” on page 289 topic.
- If the IBM Spectrum Scale service is restarted by using the restart icon, the HDFS service also needs to be restarted.
- The NameNode Last Checkpoint alert can be ignored and can be disabled.
- If the HBase master failed to start with `FileNotFoundException` 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.

- 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.

- 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**<sup>1</sup> 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**<sup>1</sup>. 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.

<sup>1</sup>For FPO cluster, do not run STOP ALL from the Ambari GUI. Refer to the Limitations > General sections on how to properly stop IBM Spectrum Scale.

- From the dashboard, click **Spectrum Scale > Service Actions > Upgrade Transparency**.

The screenshot shows the Ambari interface for a cluster named 'mycluster'. The top navigation bar includes 'Dashboard', 'Services', 'Hosts', 'Alerts', 'Admin', and a user dropdown. On the left, a sidebar lists various services with their current status (e.g., HDFS, YARN, MapReduce-e2, Hive, HBase, Pig, SQuoop, Oozie, ZooKeeper, Flume, Ambari Metrics, Kafka, Knox, Spark2, Spectrum Scale, Säder, Solr, SystemML, Titan). The 'Spectrum Scale' service is selected. The main content area has tabs for 'Summary' and 'Configs'. Under 'Summary', there's a 'Metrics' section with four cards: 'Filesystem Utilization' (n/a), 'Inode Utilization' (n/a), 'Active Quorum Nodes' (n/a/n/a), and 'Active NSD Nodes' (n/a/n/a). To the right, a 'Service Actions' dropdown menu is open, listing options like Start, Stop, Restart All, etc. The 'Upgrade Transparency' option is highlighted with a red box.

- 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:

- a. Remove the existing HDFS Transparency package by running the following command:

```
$ yum erase gpfs.hdfs-protocol<Old version>
$ yum clean all
```

- b. yum install the new package.

```
$ yum install gpfs.hdfs-protocol-3.0.0.0.<OS>.rpm
```

6. On the dashboard, click **Actions > Start All**.

7. 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<sup>1</sup>** to stop all services.

<sup>1</sup>For FPO cluster, do not run STOP ALL from the Ambari GUI. Refer to the Limitations > General sections on how to properly stop IBM Spectrum Scale.

- From the dashboard, select **Spectrum Scale > Service Actions > Upgrade Spectrum Scale.**

The screenshot shows the Ambari dashboard for a cluster named 'nampatra'. The left sidebar lists various services with their status and alert counts. The 'Spectrum Scale' service is selected, highlighted in dark grey. The main panel shows a 'Summary' section with 'GPFS Master' status (Stopped, No alerts) and 'GPFS Node' count (0/4). Below it is a 'Metrics' section with four cards: 'Filesystem Utilization' (n/a), 'Inode Utilization' (n/a), 'Active Quorum Nodes' (n/a/n/a), and 'Active NSD Nodes' (n/a/n/a). To the right, a 'Service Actions' dropdown menu is open, listing options like Start, Stop, Restart All, etc. The 'Upgrade Spectrum Scale' option is highlighted with a red border.

- Verify that the selected IBM Spectrum Scale PTF packages are installed on the nodes.

```
$ xdsh c902f05[x01-x04] "rpm -qa | grep gpfs"
```

**Note:** Check that the gpfs version installed on the nodes are the updated ones.

- From the dashboard, select **Actions > Start All**

**Note:** IBM Spectrum Scale starts with the latest PTF packages. Verify that HDFS Transparency Namenode and Datanodes are functioning.

```
$ /usr/lpp/mmfs/hadoop/sbin/mmhadoopctl connector getstate
c902f05x01(gpfs.net: namenode running as process 18150.
c902f05x01(gpfs.net: datanode running as process 22958.
c902f05x04(gpfs.net: datanode running as process 15560.
c902f05x03(gpfs.net: datanode running as process 17275.
c902f05x02(gpfs.net: datanode running as process 26416.
```

## Migrating IOP to HDP

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:**

- Mpack version 2.4.2.1.1 replaces version 2.4.2.1.
- To see the default configuration modifications under Spectrum Scale Mpacs, refer to Big Data and Analytics - summary of changes section.
- The cluster must be at management pack version 2.4.2.0 or later.
- 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.
- Ensure that an anonymous user id is created and has the same uid/gid in your cluster before upgrading.

**Procedure**

1. As the root user, download a management pack onto a directory on the Ambari server node. Ensure that the management pack is at a higher PTF version than the version of IBM Spectrum Scale service installed on your system. For information on downloading the management packs, see the “IBM Spectrum Scale service” on page 253 topic.

**Note:** The downloaded management pack should be stored and unzipped in a different directory than the currently installed version of the Mpack.

In this example, the downloaded management pack has been downloaded in the /root/GPFS\_Ambari/upgrade\_Mpack directory. The management pack contains the upgrade script to upgrade the MPack.

For example, if the currently installed Mpack is at 2.4.2.0 version, plan to upgrade to Mpack 2.4.2.1 version.

The `SpectrumScale_UpgradeIntegrationPackage` script used for upgrade and migration is run from the /root/GPFS\_Ambari/upgrade\_Mpack directory.

Ensure that the current Mpack installable package resides on a separate directory on the Ambari server node. This example uses the /root/GPFS\_Ambari/currently\_installed\_Mpack directory.

The `SpectrumScaleMPackUninstaller.py` script used as part of this procedure would have to be run from the /root/GPFS\_Ambari/currently\_installed\_Mpack directory.

2. Log in to Ambari.
3. Stop 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.
4. Stop all the services. Click **Ambari > Actions > Stop All**<sup>1</sup>.

<sup>1</sup> - For FPO cluster, do not run STOP ALL from the Ambari GUI. Refer to the General section on how to properly stop IBM Spectrum Scale.

5. After all the services have stopped, unintegrate the transparency.

Follow the steps in Unintegrating Transparency and ensure that the `ambari-server restart` command is run.

**Note:** Do not start the services.

6. If the IBM Spectrum Scale service is not already stopped, stop the IBM Spectrum Scale service by clicking **Ambari > Spectrum Scale > Service Actions > Stop**.
7. On the Ambari server node as root, from the /root/GPFS\_Ambari/upgrade\_Mpack directory, run the `SpectrumScale_UpgradeIntegrationPackage` script with the `--preEU` option.

The **--preEU** option saves the existing IBM Spectrum Scale service information into JSON files in the local directory where the script was run. It also removes the IBM Spectrum Scale service from the Ambari cluster so that the cluster can be properly migrated. This does not affect the IBM Spectrum Scale file system.

Before you proceed, review the following questions for the upgrade script and have the information for your environment handy. If Kerberos is enabled, more inputs are required.

```
$ cd /root/GPFS_Ambari/upgrade_Mpack
$./SpectrumScale_UpgradeIntegrationPackage --preEU
Are you sure you want to upgrade the GPFS Ambari integration package (Y/N)? (Default Y):

STARTING WITH SPECTRUM SCALE EXPRESS UPGRADE PRE STEPS

Enter the Ambari server User:(Default admin):
Enter the password for the Ambari server.
Password:
Retype password:
SSL Enabled (True/False) (Default False):
Enter the Ambari server Port. (Default 8080):
...
Note: If Kerberos is enabled, then the KDC principal and password information are required.
Kerberos is Enabled. Proceeding with Configuration
Enter kdc principal:
Enter kdc password:
...
```

8. If you are migrating from BI IOP 425, as root user on the Ambari server, run the Mpack uninstaller script **SpectrumScaleMPackUninstaller.py** from the currently installed Mpack directory to remove the existing Mpack link in Ambari.

```
$ cd /root/GPFS_Ambari/currently_installed_Mpack
$./SpectrumScaleMPackUninstaller.py
```

```
INFO: ***Starting the MPack Uninstaller***
```

```
Enter Ambari Server Port Number. If it is not entered, the uninstaller will take default port 8080:
INFO: Taking default port 8080 as Ambari Server Port Number.
Enter Ambari Server IP Address : 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.
```

9. Start all services. Click **Ambari > Actions > Start All**.

Wait for all the services to start. At this stage, native HDFS is used.

10. 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.

11. 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.

12. 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.

13. Add the IBM Spectrum Scale service.

On the Ambari server node as root, from the `/root/GPFS_Ambari/upgrade_Mpack` directory, run the `SpectrumScale_UpgradeIntegrationPackage` script with the `--postEU` option in the directory where the `--preEU` step was run and where the JSON configurations were stored.

Before you proceed, for the `--postEU` option, review the following questions, and have the information for your environment handy. If Kerberos is enabled, more inputs are required.

```
$ cd /root/GPFS_Ambari/upgrade_Mpack
$./SpectrumScale_UpgradeIntegrationPackage --postEU
Are you sure you want to upgrade the GPFS Ambari integration package (Y/N)? (Default Y):

STARTING WITH SPECTRUM SCALE EXPRESS UPGRADE POST STEPS

Starting Post Express Upgrade Steps. Enter Credentials
Enter the Ambari server User:(Default admin):
Enter the password for the Ambari server.
Password:
Retype password:
SSL Enabled (True/False) (Default False):
Enter the Ambari server Port. (Default 8080):
...
Accept License
Do you agree to the above license terms? [yes or no]
yes
Installing...
Enter Ambari Server Port Number. If it is not entered, the installer will take default port 8080:
INFO: Taking default port 8080 as Ambari Server Port Number.
Enter Ambari Server IP Address :
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.

```

14. 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 12.

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.

15. 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 `FileNotFoundException` error, restart HDFS and then restart the HBase master.

## 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 Setting up High Availability [HA].

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: <code>fs.defaultFS</code>	Ensure that <code>hdfs://localhost:8020</code> is used

Standard tab	Rule	Advanced tab	Rule
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**

For example:

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

### Local OS repository

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

1. Create the repository path:  
**mkdir /var/www/html/repos/<rhel\_OSlevel>**

- Synchronize the local directory with the current yum repository:  

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

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

```
reposync --gpgcheck -l --repoid=rhel-7-server-rpms --download_path=/var/www/html/repos/<rhel_OSlevel>
```
- Create the repository for this node:  

```
createrepo -v /var/www/html/repos/<rhel_OSlevel>
```
- 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.
- On all nodes in the cluster that require the repositories, create a file in /etc/yum.repos.d called `local_<rhel_OSlevel>.repo`.
- Copy this file to all nodes. The contents of this file must look like the following:  

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

- 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.

**Note:** From IBM Spectrum Scale 5.0.2, the gpfs.ext package is not available.

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](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. Click **Ambari GUI > Log Search service > Quick Links > Log Search UI**.
2. Login to **Log Search UI** and click on the top right corner button. Choose the **Configuration Editor** option.
3. In the **Configuration Editor** page, change the log path for the following components:
  - a. `hdfs_datanode`
  - b. `hdfs_namenode`
  - c. `hdfs_zkfc`

Example (existing) entry for namenode:

```

"type": "hdfs_namenode",
"rowtype": "service",
"path": "/var/log/hadoop/hdfs/hadoop-hdfs-namenode-*.*.log"

```

change to:

```

"type": "hdfs_namenode",
"rowtype": "service",
"path": "/var/log/hadoop/root/hadoop-root-namenode-*.*.log"

```

Make similar changes for datanode and zkfc.

4. Restart HDFS service.

## 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.

## Hadoop Kafka/Zookeeper and IBM Spectrum Scale Kafka/Zookeeper

IBM Spectrum Scale has new audit logging capability that uses its own libraries for Kafka and zookeeper.

To use the IBM Spectrum Scale Kafka and Zookeeper, and Hadoop Kafka and Zookeeper from HDP, you must install the Kafka and Zookeeper on different nodes for IBM Spectrum Scale and Hadoop.

In a new environment, IBM Spectrum Scale FPO is installed by the IBM Spectrum Scale Ambari Mpack. The packages for Kafka and zookeeper can be installed when HDP is installed or after the Mpack is installed. If it is required to install IBM Spectrum Scale audit logging, follow the IBM Spectrum Scale documentation to install the Kafka and Zookeeper on different nodes other than the ones installed for the HDP Hadoop cluster.

When you are installing HDP, if IBM Spectrum Scale is already installed with audit logging, check where the zookeeper and Kafka are to be installed. Ensure that the HDP zookeeper and Kafka are on nodes that are not where the IBM Spectrum Scale Kafka and zookeeper reside. If needed, change to different hosts in Ambari.

---

## 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 metadata and data. 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
```

**GPFS Filesystem**

GPFS FileSystem Name <input type="text" value="bigpfs"/>	GPFS FileSystem Mount Point <input type="text" value="/bigpfs"/>
GPFS NSD stanza file,putted in /var/lib/ambari-server/resources/ <input type="text"/>	Hadoop local cache disk stanza file,putted in /var/lib/ambari-server/resources/ <input type="text"/>
GPFS policy file for file system,putted in /var/lib/ambari-server/resources/ <input type="text"/>	GPFS cluster rack mapping file,putted in /var/lib/ambari-server/resources/ <input type="text"/>
<hr/>	
Default Data Replicas <input type="range" value="3"/>	Default Metadata Replicas <input type="range" value="3"/>
Max Data Replicas <input type="range" value="3"/>	Max Metadata Replicas <input type="range" value="3"/>
Percentage of Local Data on Data Disks <input type="range" value="20%"/>	

b1adm048

Figure 25. 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 27. 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.

## Ranger Requirements

1. You must have an **MySQL/Oracle/Postgres/MSSQL/SQL Anywhere Server** database instance running to be used by Ranger.
2. 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)
3. Ensure that the access for the DB Admin user is enabled in DB server from any host.
4. 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={/jdb
c/driver/path}
```

[Cancel](#) [Proceed](#)

I have met all the requirements above.

b1adm051

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

The screenshot shows the 'Ranger Admin' configuration interface for a MySQL database. It includes fields for DB FLAVOR (MySQL), Ranger DB host (c8f2n07.gpfs.net), Ranger DB name (ranger), Driver class name for a JDBC Ranger database (com.mysql.jdbc.Driver), Ranger DB username (rangeradmin), Ranger DB password (redacted), JDBC connect string (jdbc:mysql://c8f2n07.gpfs.net/ranger), and a Test Connection button.

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.

The screenshot shows the 'Ranger Admin' configuration interface for a MySQL database, similar to the previous one but with a successful test connection result. It includes fields for Database Administrator (DBA) username (rangerdba), Database Administrator (DBA) password (redacted), JDBC connect string for root user (jdbc:mysql://c8f2n07.gpfs.net), and a Test Connection button which shows 'Connection OK' with a green checkmark.

bl1adm053

- 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*.

The screenshot shows the 'Advanced' tab of the 'Ranger Admin' configuration. It displays two sections: 'Audit to Solr' (disabled) and 'Audit to HDFS' (enabled). The 'Audit to HDFS' section includes a 'Destination HDFS Directory' field containing 'hdfs://mycluster/ranger/audit'.

bl1adv241

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

Database Administrator (DBA) username  
rangerdba

Database Administrator (DBA) password  
\*\*\*\*\*

JDBC connect string for root user  
jdbc:mysql://c8f2n07.gpfs.net

Test Connection      Connection OK

bl1adm055

#### *Enabling Ranger HDFS plug-in:*

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

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

There are 2 configuration changes in 1 service [Show Details](#)

Ranger Admin    Ranger User Info    **Ranger Plugin**    Ranger Audit    Advanced

**Ranger Plugin**

**HDFS Ranger Plugin** **Hbase Ranger Plugin**

bl1adv242

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

**Dependent Configurations**

**Recommended Changes**

Based on your configuration changes, Ambari is recommending the following dependent configuration changes.  
Ambari will update all checked configuration changes to the Recommended Value. Uncheck any configuration to retain the Current Value.

Property	Service	Config Group	File Name	Current Value	Recommended Value
<input checked="" type="checkbox"/> ranger-hdfs-plugin-enabled	HDFS	Default	ranger-hdfs-plugin-provider	No	Yes
<input checked="" type="checkbox"/> dfs.namenode.inode.attributes.provider.class	HDFS	Default	hdfs-site	Property undefined	org.apache.ranger.authorization.hadoop.RangerHdfsAuthorizer

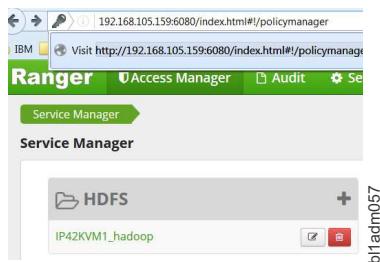
Cancel bl1adv274

- 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.

- DataNode reports exceptions after Kerberos is enabled on RHEL7.5. See 2nd generation HDFS Protocol troubleshooting - P19.

### 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:

The screenshot shows the 'Customize Services' step of the 'Add Service Wizard'. On the left, a sidebar lists steps: Choose Services, Assign Masters, Assign Slaves and Clients, **Customize Services** (which is selected), Configure Identities, Review, Install, Start and Test, and Summary. The main area is titled 'Customize Services' with a sub-header 'Advanced gpfs-advance'. It shows configuration groups: 'Advanced gpfs-ambari-server-env'. Under this group, several fields are listed: AMBARI\_USER (admin), AMBARI\_USER\_PASS WORD (\*\*\*\*), KDC\_PRINCIPAL (root/admin@IBM.COM), KDC\_PRINCIPAL\_PASS WORD (\*\*\*\*), and GPFS\_REPO\_URL (http://c902mnx09.gpfs.net/repos/GPFS/4.2.2.3/gpfs\_rpms). The KDC\_PRINCIPAL and KDC\_PRINCIPAL\_PASS WORD fields are highlighted with a red box.

bl1adv243

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

The screenshot shows the 'Add Service Wizard' interface. On the left, a sidebar lists steps: Choose Services, Assign Masters, Assign Slaves and Clients, Customize Services, Configure Identities (which is selected and highlighted in black), Review, Install, Start and Test, and Summary. The main panel is titled 'Configure Identities' with the sub-instruction 'Configure principal name and keytab location for service users and hadoop service components.' It has tabs for 'General' and 'Advanced'. Under 'General', there is a section for 'Global' configuration. The 'Admin principal' field contains 'root/admin@IBM.COM' and the 'Admin password' field contains '.....'. A checkbox for 'Save Admin Credentials' is present. Below this, other fields include 'Keytab Dir' set to '/etc/security/keytabs', 'Realm' set to 'IBM.COM', 'Spnego Principal' set to 'HTTP/\_HOST@\${realm}', and 'Spnego Keytab' set to '\${keytab\_dir}/spnego.service.keytab'. A watermark 'blhadv244' is visible on the right.

3. 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.

1. Follow the steps in "Setting up KDC server and enabling Kerberos" on page 196 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.
2. From Ambari, click **Spectrum Scale service > Configs tab > Advanced > Advanced gpfs-ambari-server-env**. Type the KDC principal values and the KDC principal password values. Save the configuration.

The screenshot shows the Ambari interface for managing configurations. On the left, a sidebar lists various services: HDFS, MapReduce2, YARN, Hive, HBase, Pig, Sqoop, Oozie, ZooKeeper, Flume, Titan, Ambari Metrics, Spectrum Scale, Kafka, Kerberos, Knox, Slider, Solr, and Spark. The 'Spectrum Scale' service is currently selected. The main panel displays a configuration group named 'V1' with one configuration entry: 'admin authored on Mon, Mar 27, 2017 15:44'. Below this, under the 'Advanced' tab, there are two sections: 'Advanced gpfs-advance' and 'Advanced gpfs-ambari-server-env'. In the 'Advanced gpfs-ambari-server-env' section, the 'KDC\_PRINCIPAL' field contains 'root/admin@IBM.COM' and the 'KDC\_PRINCIPAL\_PASS' field contains a masked password. Both of these fields are enclosed in a red rectangular box.

3. Restart all the services. Click **Ambari panel > Service Actions > Stop All and Start All**<sup>1</sup>.

<sup>1</sup>For FPO cluster, do not run STOP ALL from the Ambari GUI. Refer to the Limitations > General sections on how to properly stop IBM Spectrum Scale.

### 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**.

The screenshot shows the Ambari interface for managing HDFS configurations. The left sidebar lists various services: HDFS, MapReduce2, YARN, Hive, HBase, Pig, Sqoop, Oozie, ZooKeeper, Flume, Ambari Metrics, Spectrum Scale, Kafka, Knox, Slider, Solr, and Spark. The main content area has tabs for Summary, Heatmap (which is active), and Configs. Under Configs, it shows a group named 'HDFS Default (4)' with two versions: V2 (14 days ago) and V1 (14 days ago). The configuration 'dfs.client.read.shortcircuit.streams.cache.size' is highlighted with a red box. The configuration details show the value is 4096. There are 'Discard' and 'Save' buttons at the bottom.

**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 lo 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

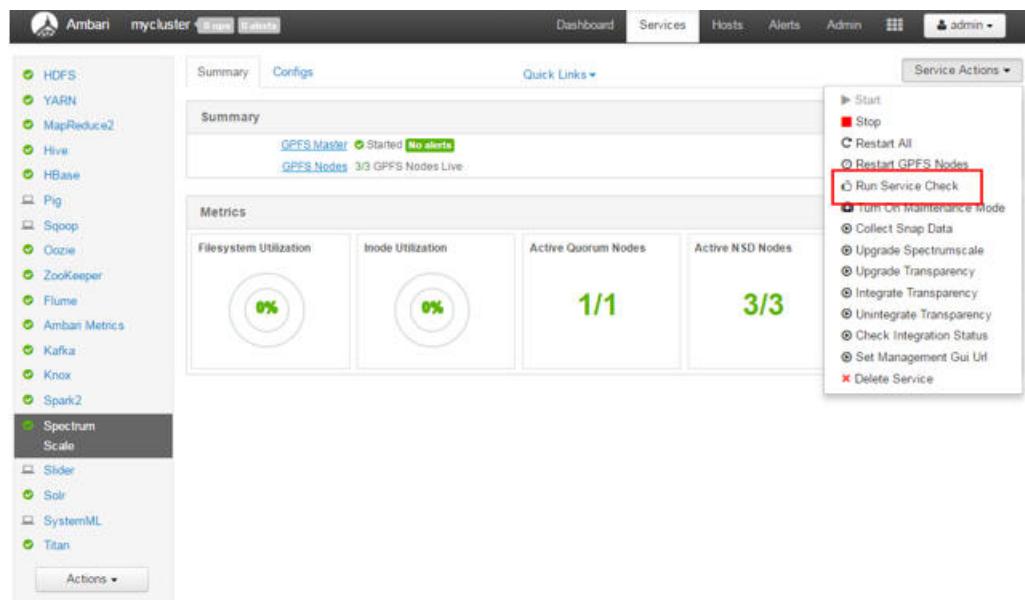
To go to the Service Actions dropdown list, click **Spectrum Scale > Service Action**.

**Note:** Do not use the **Delete Service** action from the dropdown **Actions** menu. If you wish to get rid of the Spectrum Scale service, follow the procedure in “Uninstalling IBM Spectrum Scale Mpack and service” on page 167.

When the IBM Spectrum Scale Mpack and service is deleted, the IBM Spectrum Scale file system and packages are preserved as is. For an FPO cluster created through Ambari, the mounted local disks `/opt/mapred/local*` and entries in `/etc/fstab` are preserved as is.

## Running the service check

To check the status and stability of the service, run a service check on the IBM Spectrum Scale dashboard by clicking **Run Service Check** in the Service Actions dropdown menu.

A screenshot of the Ambari dashboard for a cluster named 'mycluster'. The left sidebar shows various service icons, with 'Spectrum Scale' selected. The main area has tabs for 'Summary' and 'Configs', with 'Summary' active. It displays metrics like 'Filesystem Utilization' at 0%, 'Inode Utilization' at 0%, 'Active Quorum Nodes' at 1/1, and 'Active SSD Nodes' at 3/3. A 'Service Actions' dropdown menu is open on the right, listing options such as Start, Stop, Restart All, Restart GPFS Nodes, Run Service Check (which is highlighted with a red box), 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.

- Review the service check output logs for any issues.
- To manually check the HDFS Transparency Namenodes and Datanodes state, run the following command:

```
/usr/lpp/mmfs/bin/mmhadoopctl connector getstate
$ /usr/lpp/mmfs/bin/mmhadoopctl connector getstate
c902f05x01.gpfs.net: namenode running as process 4749.
c902f05x01.gpfs.net: datanode running as process 10214.
c902f05x02.gpfs.net: datanode running as process 4767.
c902f05x03.gpfs.net: datanode running as process 8204.
```

## Stop all without stopping IBM Spectrum Scale service

To prevent IBM Spectrum Scale service from being stopped when you click **ACTION > STOP ALL**, place the IBM Spectrum Scale service into maintenance mode.

Click **Ambari GUI > Spectrum Scale service > Actions > Turn on Maintenance Mode**.

This prevents any Ambari actions from occurring on the service that is in maintenance mode.

To get out of Maintenance Mode, click **Ambari GUI > Spectrum Scale service > Actions > Turn off Maintenance Mode**.

**Note:** For FPO cluster, do not run STOP ALL from the Ambari GUI. Refer to the Limitations > General sections on how to properly stop IBM Spectrum Scale.

## Modifying IBM Spectrum Scale service configurations

The IBM Spectrum Scale service has standard and advanced configuration panels.

Click **Ambari GUI > Spectrum Scale > Configs tab**.

The screenshot shows the 'Configs' tab of the IBM Spectrum Scale management pack. At the top, there are tabs for 'Summary' and 'Configs', with 'Service Actions' on the right. Below this, a navigation bar includes 'Group' (set to 'Default (3)'), 'Manage Config Groups', and a 'Filter...' search bar. A configuration card for 'V1' is displayed, showing it was authored by 'admin' 19 minutes ago on 'BigInsights-4.3'. Below the card, a message says 'V1 ✓ admin authored on Wed, May 17, 2017 18:43'. There are 'Discard' and 'Save' buttons. Underneath, there are 'Standard' and 'Advanced' tabs, with 'Advanced' selected. The 'GPFS Environment' section contains two fields: 'GPFS Cluster Name' (set to 'bigpfs') and 'GPFS Remote Copy' (set to '/usr/bin/scp'). A watermark 'bihadv248' is visible on the right side of the page.

## 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.

The screenshot shows the 'Add Property' dialog. It has a 'Type' field containing 'gpfs-advance.xml' and a 'Properties' field containing 'key=value (one per line)'. The properties field is currently empty. At the bottom, there are 'Cancel' and 'Add' buttons. A watermark 'bihadv248' is visible on the right.

In Ambari, if any configuration in the `gpfs-site` is changed in the Spectrum Scale dashboard, a Stop All service followed by a Start All service is required. Check your environment to ensure that the changes made are in effect.

**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/5.0.1/gpfs_rpms`
3. Save the GPFS\_REPO\_URL configuration.
4. The GPFS\_REPO\_URL becomes effective during the Upgrading IBM Spectrum Scale and Upgrading HDFS Transparency process.

## HDFS and IBM Spectrum Scale restart order

When the IBM Spectrum Scale service is integrated, HDFS Transparency Namenodes and Datanodes are managed as HDFS Namenode and Datanode components respectively in the Ambari HDFS service.

When configuration is changed in IBM Spectrum Scale™, the following restart order should be followed:

1. Stop the HDFS service first.
2. Restart the IBM Spectrum Scale service.
3. Restart the HDFS service.

## Integrating HDFS transparency

You must plan a cluster maintenance window, and prepare for the cluster down time when integrating the HDFS Transparency with the native HDFS. After each integration, you must run the **ambari-server restart** on the Ambari server node. Ensure that all the services are stopped.

**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**<sup>1</sup> to stop all services.  
<sup>1</sup>For FPO cluster, do not run STOP ALL from the Ambari GUI. Refer to the Limitations > General sections on how to properly stop IBM Spectrum Scale.
2. On the IBM Spectrum Scale dashboard, click **Service Actions** > **Integrate Transparency**.
3. Verify that all services are stopped. If not, stop the services.

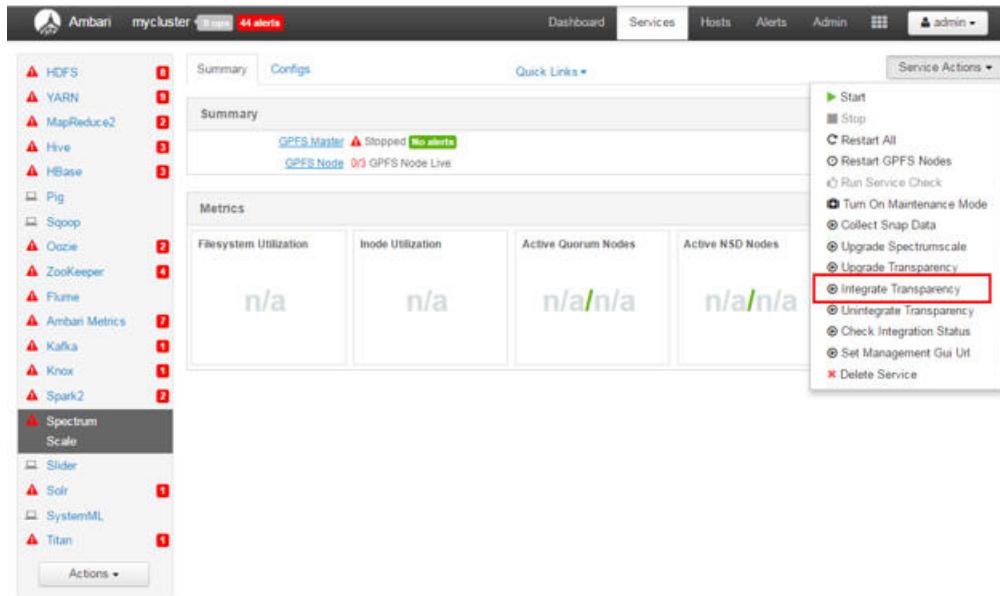
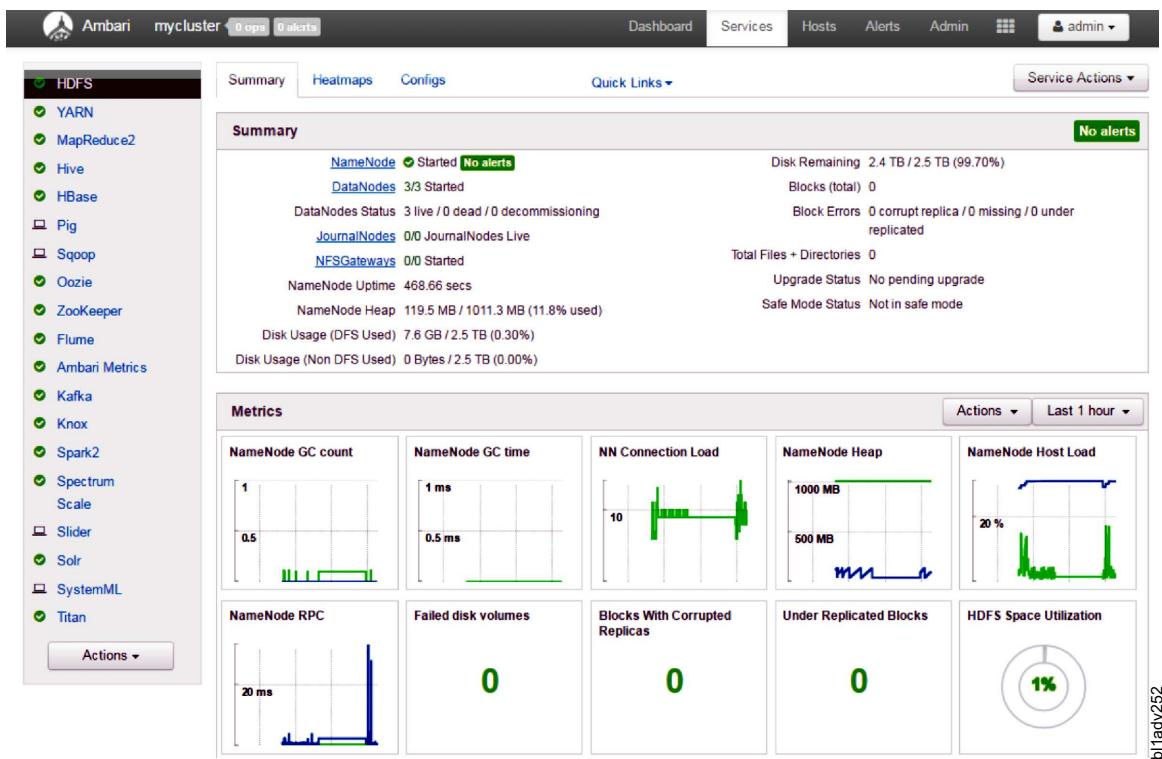


Figure 26. IBM SPECTRUM SCALE INTEGRATE TRANSPARENCY

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 DataNodes 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 318

## 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**<sup>1</sup> to stop all services.

<sup>1</sup>For FPO cluster, do not run STOP ALL from the Ambari GUI. Refer to the Limitations > General sections on how to properly stop IBM Spectrum Scale.

2. Click **Spectrum Scale > Service Actions > Unintegrate Transparency**.

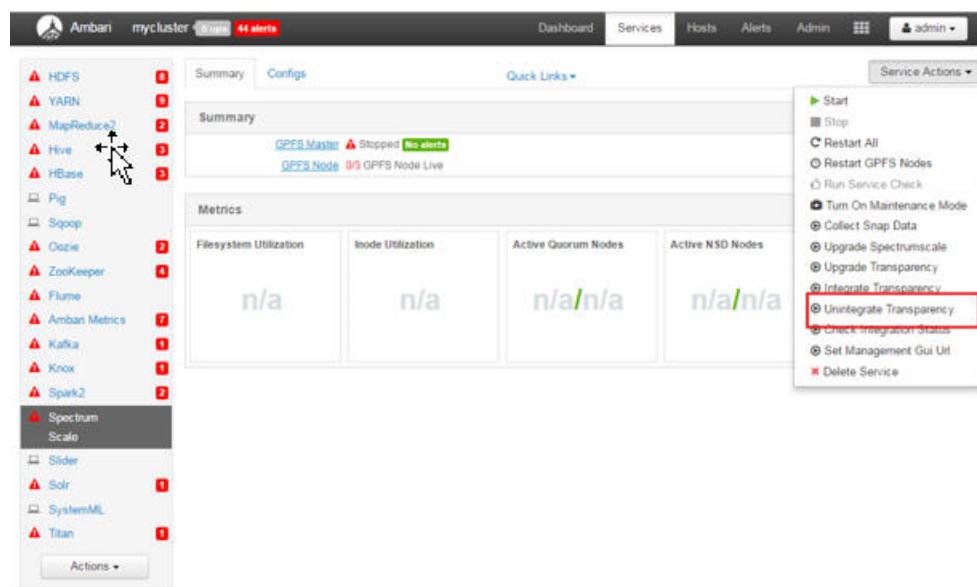


Figure 27. IBM SPECTRUM SCALE UNINTEGRATE TRANSPARENCY

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**.

The screenshot shows the Ambari interface for a cluster named 'mycluster'. On the left, there's a sidebar with various service icons. The 'Spectrum Scale' icon is highlighted. The main panel shows the 'Configs' tab for the 'GPFS Master' service. It displays metrics like Filesystem Utilization (0%), Inode Utilization (0%), Active Quorum Nodes (1/1), and Active NSD Nodes (3/3). To the right, a 'Service Actions' dropdown menu is open, listing options such as Start, Stop, Restart All, and several configuration-related actions. The 'Check Integration Status' option is specifically highlighted with a red box.

After the process completes, check the output log for the state information.

The screenshot shows a web interface with the title 'c902f05x01.gpfs.net'. Under the 'Tasks' tab, there is a green checkmark icon next to the text 'Check Integration Status GPFS Master'. Below this, there are two sections: 'stderr' and 'stdout'. The 'stderr' section contains the path '/var/lib/ambari-agent/data/errors-252.txt'. The 'stdout' section contains the path '/var/lib/ambari-agent/data/output-252.txt'. Inside the 'stdout' box, there is a message: '2017-05-31 12:02:33,255 - ===== Spectrum Scale service is INTEGRATED. ====='. Below this message is the text 'Command completed successfully!'. At the bottom of the window, there is a checkbox labeled 'Do not show this dialog again when starting a background operation' and a green 'OK' button. To the right of the 'OK' button, the text 'bl1adv254' is visible.

## 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)	quorum	
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/mmlsnsd
```

File system	Disk name	NSD servers
bigpfs	gpfs1nsd	c902f05x01(gpfs.net)
bigpfs	gpfs2nsd	c902f05x02(gpfs.net)
bigpfs	gpfs3nsd	c902f05x03(gpfs.net)
bigpfs	gpfs4nsd	c902f05x04(gpfs.net)
bigpfs	gpfs5nsd	c902f05x03(gpfs.net)
bigpfs	gpfs6nsd	c902f05x02(gpfs.net)

```

bigpfs gpfs7nsd c902f05x01(gpfs.net
bigpfs gpfs8nsd c902f05x04(gpfs.net
bigpfs gpfs9nsd c902f05x02(gpfs.net
bigpfs gpfs10nsd c902f05x03(gpfs.net
bigpfs gpfs11nsd c902f05x04(gpfs.net
bigpfs gpfs12nsd c902f05x01(gpfs.net
bigpfs gpfs13nsd c902f05x02(gpfs.net
bigpfs gpfs14nsd c902f05x03(gpfs.net
bigpfs gpfs15nsd c902f05x04(gpfs.net
bigpfs gpfs16nsd c902f05x01(gpfs.net

[root@c902f05x01 ~]#
[root@c902f05x05 ~]# mount | grep bigpfs
bigpfs on /bigpfs type gpfs (rw,relatime)
[root@c902f05x05 ~]#

[root@c902f05x01 ~]# /usr/lpp/mmfss/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.

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 DataNodes 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**.
- 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 when the IBM Spectrum Scale service is integrated.

### *Instructions for HA cluster:*

This topic describes the steps to manually move a Namenode when HDFS Transparency is in integrate state.

1. From the dashboard, select **Actions > Stop All**<sup>1</sup>.

<sup>1</sup>For FPO cluster, do not run STOP ALL from the Ambari GUI. Refer to the Limitations > General sections on how to properly stop IBM Spectrum Scale.

2. On the Ambari server host, run the following command:

```
python /var/lib/ambari-server/resources/mpacks/SpectrumScaleExtension-MPack-<version>/extensions/SpectrumScaleExtension/<version>/services/GPFS/package/files/COMMON/MoveNameNodeTransparency.py
```

Follow the command prompts and type the required input.

```
$ python /var/lib/ambari-server/resources/mpacks/SpectrumScaleExtension-MPack-2.7.0.0/extensions/SpectrumScaleExtension/2.7.0.0/services/GPFS/package/files/COMMON/MoveNameNodeTransparency.py
Enter the Ambari Server User:(Default User admin):
Enter the Password for Ambari Server.
Password:
Retype password:
SSL Enabled (True/False) (Default False):
Enter the Ambari Server Port.(Default 8080)
Enter the Fully Qualified HostName of the Source NameNode which has to be Removed:- c902f09x02.gpfs.net
Enter the Fully Qualified HostName of the Destination NameNode has to be Added: c902f09x03.gpfs.net
```

### **Note:**

- SSL Enabled means Ambari HTTPS.
  - The 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**<sup>1</sup>.

<sup>1</sup>For FPO cluster, do not run STOP ALL from the Ambari GUI. Refer to the Limitations > General sections on how to properly stop IBM Spectrum Scale.

2. On the Ambari server host, run the following command:

```
python /var/lib/ambari-server/resources/mpacks/SpectrumScaleExtension-MPack-<version>/extensions/SpectrumScaleExtension/<version>/services/GPFS/package/files/COMMON/MoveNameNodeTransparency.py
```

Follow the command prompts and type the required input.

```
$ python /var/lib/ambari-server/resources/mpacks/SpectrumScaleExtension-MPack-2.7.0.0/extensions/SpectrumScaleExtension/2.7.0.0/services/GPFS/package/files/COMMON/MoveNameNodeTransparency.py
Enter the Ambari Server User:(Default User admin):
Enter the Password for Ambari Server.
Password:
Retype password:
SSL Enabled (True/False) (Default False):
```

Enter the Ambari Server Port.(Default 8080)  
Enter the Fully Qualified HostName of the Source NameNode which has to be Removed:c902f09x02.gpfs.net  
Enter the Fully Qualified HostName of the Destination NameNode has to be Added:c902f09x03.gpfs.net

**Note:**

- SSL Enabled means Ambari HTTPS.
- The destination node must be one of the HDFS Transparency node.

3. From the dashboard, select **Actions > Start All**.
4. Moving the Namenode process is now completed. Verify that the Active Namenode and the Standby Namenode are correct.

**Revert to native HDFS after move Namenode:**

The **move Namenode** is executed when IBM Spectrum Scale is integrated using HDFS Transparency. However, you can later choose to use the native HDFS instead by unintegrating HDFS Transparency.

In that case, you must follow these steps, to ensure that the native HDFS have the correct Namenode setting.

1. Follow the steps 1-3 of Unintegrating 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**<sup>1</sup>.

<sup>1</sup>For FPO cluster, do not run STOP ALL from the Ambari GUI. Refer to the Limitations > General sections on how to properly stop IBM Spectrum Scale.

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 step 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)
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, 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.
*****

```

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.

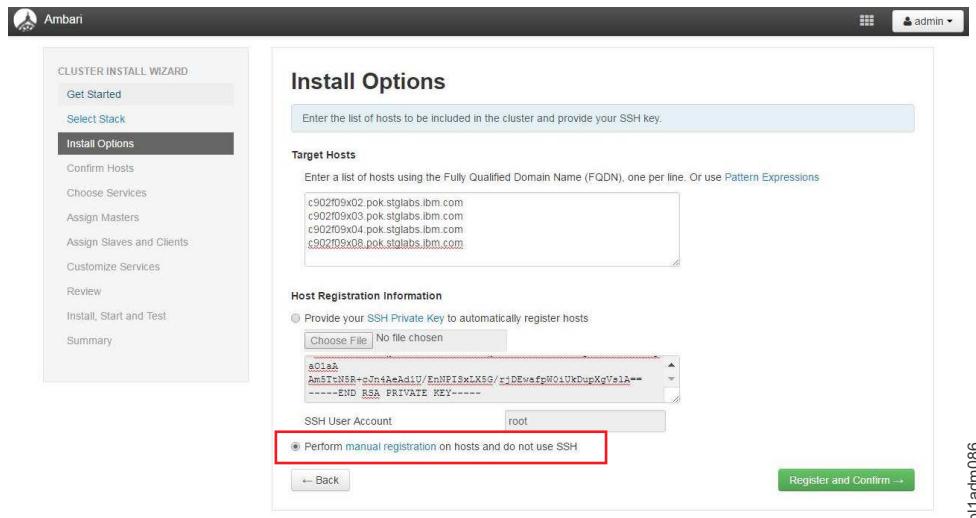
3. Set up an Ambari cluster as the non-root user.

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/klst
Cmnd_Alias BIGSQL_SERVICE_AGNT=/var/lib/ambari-agent/cache/stacks/BigInsights/*/services/
BIGSQL/package/scripts/*
Cmnd_Alias BIGSQL_SERVICE_SRVR=/var/lib/ambari-server/resources/stacks/BigInsights/*
/services/BIGSQL/package/scripts/*
Cmnd_Alias BIGSQL_DIST_EXEC=/usr/ibmpacks/current/bigsql/bigsql/bin/*,
/usr/ibmpacks/current/bigsql/bigsql/libexec/*,
/usr/ibmpacks/current/bigsql/bigsql/install/*, /usr/ibmpacks/current/IBM-DSM/ibm-datasrvrmgr/bin/*,
/usr/ibmpacks/bin/*/*
Cmnd_Alias BIGSQL_OS_CALLS=/bin/su, /usr/bin/getent, /usr/bin/id, /usr/bin/ssh, /bin/echo,
/usr/bin/scp, /bin/find, /usr/bin/du, /sbin/mkhomedir_helper, /bin/curl

am_agent ALL=(ALL) NOPASSWD:SETENV:/bin/*, /usr/bin/*, /usr/sbin/*, /usr/bin/R, /usr/bin/Rscript,
BIGSQL_SERVICE_AGNT, BIGSQL_SERVICE_SRVR, BIGSQL_DIST_EXEC, BIGSQL_OS_CALLS

Defaults exempt_group = am_agent
Defaults !env_reset,env_delete-=PATH
Defaults: am_agent !requiretty

#GPFS cluster non-root added
# Preserve GPFS environment variables:
Defaults env_keep += "MMMODE environmentType GPFS_rshPath GPFS_rcpPath mmScriptTrace
GPFSCMDPOR-TRANGE GPFS_CIM_MSG_FORMAT"

# Allow members of the gpfs group to run all commands but only selected commands without a password:
%am_agent ALL=(ALL) PASSWD: ALL, NOPASSWD: /usr/lpp/mmfs/bin/mmremote, /usr/bin/scp,
```

```

/bin/echo, /usr/lpp/mmfs/bin/mmsdrrestore

# Disable requiretty for group gpfs:
Defaults:%am_agent !requiretty

HDP sudoers

The Ambari Server user and group is ambari-server:hadoop.
The Ambari Agent user and group is am_agent:am_agent.
The Spectrum Scale cluster user and group is am_agent:am_agent.

Example of /etc/sudoers file added entries in HDP environment:

# Ambari Commands
ambari-server ALL=(ALL) NOPASSWD:SETENV: /bin/mkdir -p /etc/security/keytabs, /bin/chmod *
/etc/security/keytabs/*.keytab, /bin/chown * /etc/security/keytabs/*.keytab, /bin/chgrp *
/etc/security/keytabs/*.keytab, /bin/rm -f /etc/security/keytabs/*.keytab, /bin/cp -p -f
/var/lib/ambari-server/data/tmp/* /etc/security/keytabs/*.keytab

#Sudo Defaults - Ambari Server(In order for the agent to run its commands non-interactively,
some defaults need to be overridden)
Defaults exempt_group = ambari-server
Defaults !env_reset,env_delete-=PATH
Defaults: ambari-server !requiretty

# Ambari Agent non root configuration
# Ambari Customizable Users
am_agent ALL=(ALL) NOPASSWD:SETENV: /bin/su hdfs *,/bin/su ambari-qa *,/bin/su ranger *,
/bin/su zookeeper *,/bin/su knox *,/bin/su falcon *,/bin/su ams *, /bin/su flume *,/bin/su hbase *,
/bin/su spark *,/bin/su accumulo *,/bin/su hive *,/bin/su hcat *,/bin/su kafka *,/bin/su mapred *,
/bin/su oozie *,/bin/su sqoop *,/bin/su storm *,/bin/su tez *,/bin/su atlas *,/bin/su yarn *,
/bin/su kms *,/bin/su activity_analyzer *,/bin/su livy *,/bin/su zeppe-lin *,/bin/su infra-solr *,
/bin/su logsearch *

# Ambari: Core System Commands

am_agent ALL=(ALL) NOPASSWD:SETENV: /usr/bin/yum,/usr/bin/zypper,/usr/bin/apt-get, /bin/mkdir,
/usr/bin/test, /bin/ln, /bin/ls, /bin/chown, /bin/chmod, /bin/chgrp, /bin/cp, /usr/sbin/setenforce,
/usr/bin/test, /usr/bin/stat, /bin/mv, /bin/sed, /bin/rm, /bin/kill, /bin/readlink, /usr/bin/pgrep,
/bin/cat, /usr/bin/unzip, /bin/tar, /usr/bin/tee, /bin/touch, /usr/bin/mysql, /sbin/service mysqld *,
/usr/bin/dpkg *, /bin/rpm *, /usr/sbin/hst *, /sbin/service rpcbind *, /sbin/service portmap *,
/usr/bin/cd, /usr/lpp/mmfs/src, /usr/bin/curl, /usr/bin/make Au-toconfig, /usr/bin/make World,
/usr/bin/make InstallImages, /usr/lpp/mmfs/hadoop/sbin/mmhdoopt1,
/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 += "MMODE 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.

The screenshot shows the Ambari GUI interface for managing configurations. The top navigation bar includes tabs for 'Summary', 'Configs' (which is selected), 'Quick Links', and 'Service Actions'. Below the navigation is a search bar with the term 'quick'. The main content area displays a list of configuration groups: 'V2' and 'V1', both authored by 'admin' 21 hours ago, and 'BigInsights-4.2'. A modal window is open for 'V2', showing the configuration details. The 'Advanced' tab is selected. Under the 'Advanced gpfs-advance' section, the 'gpfs.webui.address' field is populated with the value 'https://c902f05x01.gpfs.net'. There are also fields for 'gpfs.webui.port' (set to 80) and 'gpfs.webui.username' (set to 'gpfs'). At the bottom of the modal are 'Discard' and 'Save' buttons. On the right side of the screen, there is a vertical toolbar with icons for file operations and a status indicator 'b1adv255'.

3. Restart the IBM Spectrum Scale service
4. Sync the configuration.
Click Ambari GUI > IBM Spectrum Scale service > Service Actions > Set Management GUI URL.
5. Restart Ambari server.

IBM Spectrum Scale versus Native HDFS

When IBM Spectrum Scale service is added, the native HDFS is no longer used. The Hadoop application interacts with HDFS transparency similar to their interactions with the native HDFS.

The application can access HDFS by using Hadoop file system APIs and Distributed File System APIs. The application can have its own cluster that is larger than the HDFS protocol cluster. However, all the nodes within the application cluster must be able to connect to all the nodes in the HDFS protocol cluster by RPC.

Note: The Secondary NameNode and Journal nodes in native HDFS are not needed for HDFS Transparency because of the following reasons:

- The HDFS Transparency namenode is stateless.
- Metadata are distributed.
- The namenode does not maintain the FSImage-like or EditLog information.

Functional limitations

This topic lists the functional limitations.

General

- The maximum number of Extended Attributes (EA) is limited by IBM Spectrum Scale. The total size of the EA key and value must be less than a metadata block size in IBM Spectrum Scale.
- The EA operation on snapshots is not supported.
- Raw namespace is not implemented because it is not used internally.

- If **gpfs.replica.enforced** is configured as gpfs, the Hadoop shell command **hadoop dfs -setrep** does not take effect. Also, **hadoop dfs -setrep -w** stops functioning and does not exit.
- HDFS Transparency namenode does not provide *safemode* because it is stateless.
- HDFS Transparency namenode does not need the second namenode like native HDFS because it is stateless.
- Maximal replica for 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 (`cachefileadmin`)
- 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.

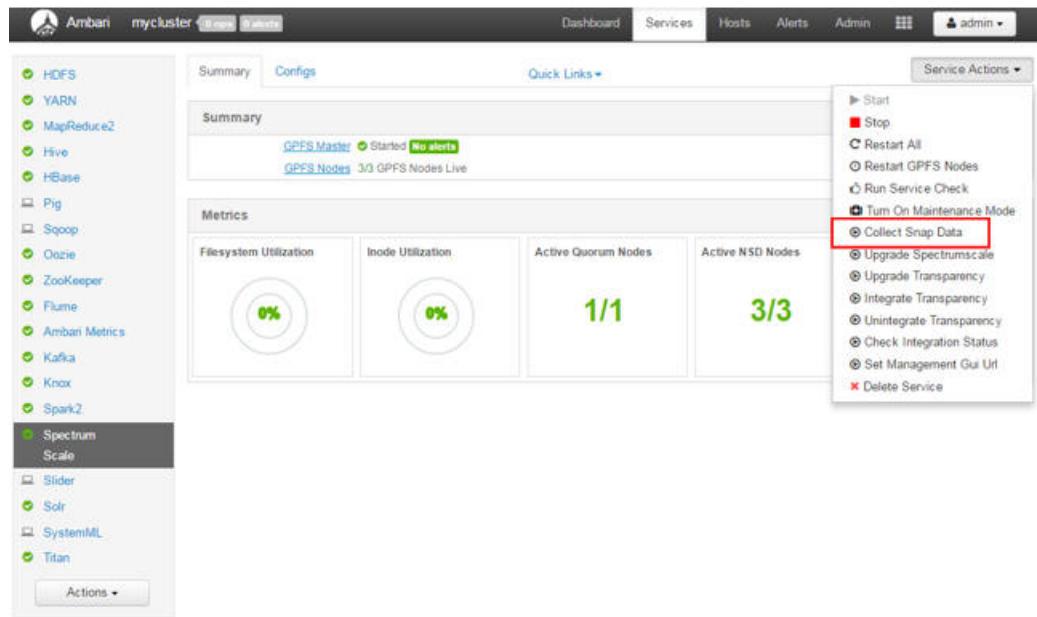
Table 28. NATIVE HDFS AND IBM SPECTRUM SCALE DIFFERENCES

| 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>mm1sfs -B</code>). The maximal value is <code>1024 * file-system-data-block-size</code> (<code>mm1sfs -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 28. 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 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.

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 `mm1sconfig` 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 `http://namenode:50070`.

FAQ

This section lists the FAQs.

For more issues, see Troubleshooting Ambari.

General

Note: For known HDFS Transparency issues, see 2nd generation HDFS Protocol troubleshooting wiki.

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`, `AMARI_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://AMARI_SERVER_HOST:8080/api/v1/clusters/CLUSTER_NAME/services/AMARI_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://AMARI_SERVER_HOST:8080/api/v1/clusters/CLUSTER_NAME/services/AMARI_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, gpfadm).

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 screenshot shows the Ambari interface for the 'mycluster' cluster. The top navigation bar includes 'Dashboard', 'Services', 'Hosts', 'Alerts', 'Admin', and a user icon for 'admin'. Below the navigation is a sidebar with service icons: MapReduce2, YARN, Hive, HBase, Pig, Sqoop, Oozie, ZooKeeper, Flume, and Ambari Metrics. The main content area is titled 'Summary' for the HDFS service. It displays various metrics: NameNode (Started), DataNodes (5/5 Started), DataNodes Status (5 live / 0 dead / 0 decommissioning), NFSGateways (0/0 Started), NameNode Uptime (17.21 hours), NameNode Heap (97.0 MB / 1011.3 MB (9.6% used)), Disk Usage (DFS Used) (3.4 GB / 2.4 TB (0.14%)), and Disk Usage (Non DFS Used) (0 Bytes / 2.4 TB (0.00%)). To the right, there's a 'Quick Links' section with links like 'Disk Usage (Remaining)', 'Blocks (total)', 'Block Errors', 'Total Files + Directories', 'Upgrade Status', and 'Safe Mode Status'. A 'Service Actions' dropdown menu is open, listing options: Start, Stop, Restart All, Run Service Check, Turn On Maintenance Mode, Rebalance HDFS, and Download Client Configs, with the last option being highlighted with a red box.

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:

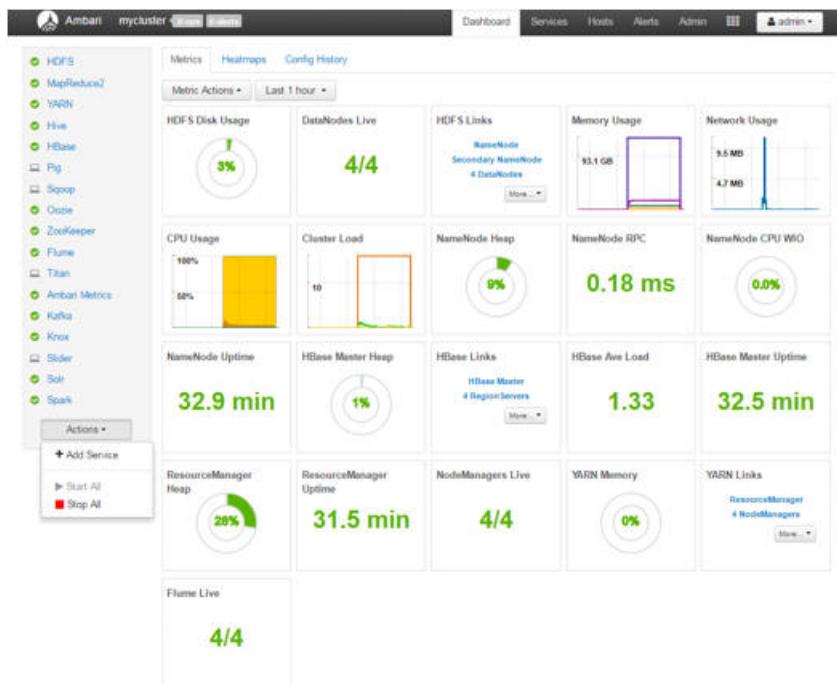
- On the HDFS dashboard, go to **Summary > JournalNodes**, and select one of the JournalNodes.

The screenshot shows the Ambari interface for the 'mycluster' cluster. The top navigation bar includes 'Dashboard', 'Services', 'Hosts', 'Alerts', and a user icon for 'admin'. Below the navigation is a sidebar with service icons: MapReduce2, YARN, Hive, HBase, Pig, Sqoop, Oozie, ZooKeeper, Flume, and Ambari Metrics. The main content area is titled 'Summary' for the JournalNodes service. It displays various metrics: NameNode (Stopped), ZKFailoverController (Stopped), DataNodes (0/3 Started), DataNodes Status (n/a), JournalNodes (0/3 JournalNodes Live), NFSGateways (0/0 Started), NameNode Uptime (Not Running), NameNode Heap (n/a / n/a (0.0% used)), Disk Usage (DFS Used) (n/a / n/a (0%)), and Disk Usage (Non DFS Used) (n/a / n/a (0%)). A red box highlights the 'JournalNodes' status line.

- 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 All¹** 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 All¹**, 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 shows the 'Alerts for HDFS' dialog box overlaid on the main management interface. The dialog lists four alerts:

- DataNode Unmounted Data Dir**: Status CRIT (1), OK (4) for 7 minutes. Description: Data dir(s) not found: /hadoop/hdfs/data.
- NameNode Host CPU Utilization**: Status OK, for 2 hours. Description: 16 CPU, load 1.8%.
- HDFS Pending Deletion Blocks**: Status OK, for 2 hours. Description: Pending Deletion Blocks [0].
- HDFS Upgrade Finalized State**: Status OK, for 2 hours. Description: HDFS cluster is not in the upgrade state.

The background interface shows various cluster statistics and metrics, including Disk Usage, Metrics, and NameNode GC.

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-version 0'] {'logoutput': False, 'sudo': True, 'quiet': False}
2017-03-26 22:55:53,783 - checked_call returned (0, '/user/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: /user/iop/current/hadoop-client/conf
2017-03-26 22:55:53,947 - The hadoop conf dir /user/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 syncconf connection to be rejected.

Solution: If the **ssh maxstartups** 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 **maxstartups** 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 ALL¹ 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 syncconf /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*.

- On the Ambari GUI, click **Spectrum Scale > Configs > Advanced > Custom gpfs-site** and set the **Add gpfs.ranger.enabled** to *false*.
- Save the configuration.
- Restart IBM Spectrum Scale.
- 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.

For FPO cluster, do not run STOP ALL from the Ambari GUI. Refer to the Limitations > General sections on how to properly stop IBM Spectrum Scale.

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.

For FPO cluster, do not run STOP ALL from the Ambari GUI. Refer to the Limitations > General sections on how to properly stop IBM Spectrum Scale.

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
------	------------	------------	------------	------------

b1adv272

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

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

b1adv273

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.e17.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 the *Building the GPFS portability layer on Linux nodes* topic in the *IBM Spectrum Scale: Concepts, Planning, and Installation Guide*.

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 autoload and automount on the IBM Spectrum Scale cluster (on the HDP cluster side).
- c. If ESS is being used, enable autoload on the ESS cluster.

For more information, see IBM Spectrum Scale **mmchfs fsname -A yes** for automount and **mmchconfig autoload=yes** commands.

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.

¹For FPO cluster, do not run STOP ALL from the Ambari GUI. Refer to the Limitations > General sections on how to properly stop IBM Spectrum Scale.

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-d; 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 -efaf | 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 **rxw--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:

- 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 the cluster is kerberos enabled, run the following command:

```
kinit -kt /etc/security/keytabs/hbase.headless.keytab hbase@<Your kerberos REALM NAME>
hbase zkcli
ls /hbase-secure/table
[ATLAS_ENTITY_AUDIT_EVENTS, hbase:meta, atlas_titan, hbase:acl, hbase:namespace]
rmr /hbase-secure/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**.

- 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:

- a. 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}}%  
{% 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:

- a. 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"
```

- b. 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:

- a. 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")
f.write("# 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
```

8. Namenode and Datanodes failed to start with mapreduce.tar.gz error.

For Mpack version 2.4.2.7 and earlier, the Namenode and Datanode might fail to start with the following error message when the data directory (gpfs.data.dir) is specified through the Ambari Spectrum Scale UI: Failed to replace mapreduce.tar.gz with Transparency jars

Solution:

Follow these steps to set the mapreduce.tar.gz into a proper directory for the Namenode/Datanode to start:

- a. Check if the /<gpfs.mnt.dir>/<gpfs.data.dir>/hdp/apps/<hdp-version>/mapreduce/ directory exists. If not, create the directory by running the following command:

```
mkdir -p /<gpfs.mnt.dir>/<gpfs.data.dir>/hdp/apps/<hdp-version>/mapreduce/
```
- b. Copy the mapreduce.tar.gz from HDP to the Scale directory by running the following command:

```
cp /usr/hdp/<hdp-version>/hadoop/mapreduce.tar.gz
/<gpfs.mnt.dir>/<gpfs.data.dir>/hdp/apps/<hdp-version>/mapreduce/mapreduce.tar.gz
```

where,

 - <gpfs.mnt.dir> is the Spectrum Scale mount point
 - <gpfs.data.dir> is the Spectrum Scale data directory
 - <hdp-version> is the HDP version and can be obtained by running **hdp-select versions**

For example,

```
mkdir -p /bigpfs/datadir1/hdp/apps/2.6.5.0-292/mapreduce/
cp /usr/hdp/2.6.5.0-292/hadoop/mapreduce.tar.gz
/bigpfs/datadir1/hdp/apps/2.6.5.0-292/mapreduce/mapreduce.tar.gz
```
- c. Restart the failed HDFS components.

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.yarn.nodemanager.log-dirs** are writable for the user yarn. If this is not the case, add write permission to the **yarn.yarn.nodemanager.local-dirs** and **yarn.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.
- If GPFS is unintegrated, remove the **tserver.wal.blocksize** entry from Accumulo. From Ambari, go to **Accumulo > Configs > Custom accumulo-site** and remove the **tserver.wal.blocksize** value and save the configuration.
- If GPFS is integrated, follow the workaround for **tserver.wal.blocksize** as mentioned in the FAQ Accumulo Tserver failed to start.

If the problem still exists, contact IBM service.

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  
>rmmr /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

fallback

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

journalized 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

Disk 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 361
accumulo support 99
add
 service 260
 services 255
adding services 146
advanced features 32
anonymous user id 138
Apache Hadoop 241
 configuration 22
Apache Ranger 87, 187, 299
Apache viewfs support 76
assign
 clients 149
 slaves 149
assign masters 149, 154
Assign Slaves and Clients 154

B

benchmarking Spark 134
benchmarks information 134
BI
 limitations 336
BigInsights 4.2.5 137, 243

C

choose services 149
Choose Services 154
cluster
 configure 148
 deploy 148
 install 148
cluster info 24
common tuning 120
complete 157
configuration
 automatic namenode HA 35
 BigSQL 295
 short-circuit read 39, 40
configuration refresh
 automatic 86
configure
 BigData and Analytics 287
 logsearch 175, 294
 multiple file system 268
 multiple file system mount point access 181
 multiple mount point access 268
 remote mount access 152, 265
configure MySQL 87, 188, 299
configure NTP 20
configure Ranger 92
configuring 82
 docker instance 44
 HDFS transparency 44
customize services 149
Customize Services 154

D

deploy
 IBM Spectrum Scale service 153
deploy HDP
 pre-existing IBM Spectrum Scale file system 177
deploy IBM Spectrum Scale
 pre-existing IBM Spectrum Scale file system 177
deploying
 dual-network 169, 288
 FPO 185, 295
deployment model 3, 4, 5, 7
DFSIO 128
difference
 HFDS transparency and native HDFS 106
 IBM Spectrum Scale 224, 334
 native HDFS 224, 334
differences
 functional 224, 334
differences from native HDFS 104
disable
 kerberos 193, 304
 short circuit write 198, 310
disk-partition 185, 295
distcp 73
dual network interfaces 16

E

elastic storage server 1
enable
 kerberos 193, 304
 Ranger HDFS plugin 91, 192, 303
enable kerberos 52
enable Kerberos
 GPFS Ambari integration version 4.2.1 194, 195, 305, 306
enable ranger 55
enable Ranger 188, 299
ESS 1
ESS storage 113

F

FAQ 340
 general 230, 340
 Hadoop storage tiering 73
 service check failures 239, 357
 service fails to start 236, 353
federate
 IBM Spectrum Scale and HDFS 77
federation 225, 337
 limitations 225, 337
file system support 101
file systems 81
fix hive schema 50
FPo 179
FPO 2, 144
 dual network interfaces 16

G

GPFS Ambari integration version 4.2.1
enable Kerberos 194, 195, 305, 306
group ids 142, 252
GUI
management 222, 331

H

hadoop
distcp support 85
introduction 109
Hadoop
configurations 21
configure nodes 20
connector 10
connector 2.4 28
connector 2.5 28
connector 2.7 28
health check 24
service health check 24
Hadoop and Software Stack Version 109
hadoop architecture 1
Hadoop connector
administer 295
Hadoop connector 2.4
removing 28
Hadoop connector 2.5
removing 28
Hadoop connector 2.7
removing 28
Hadoop distribution support 104
Hadoop Kafka 176, 295
hadoop local directories 176
hadoop nodes
hardware and software requirements 14
Hadoop nodes
configure 20
Hadoop nodes as Spectrum Scale nodes 181
Hadoop over IBM Spectrum Scale 109
hadoop performance 109
hadoop storage tiering 49
Hadoop storage tiering 180
hadoop support 9
Hadoop test case 57
hardware configuration 114
hardware requirements 14, 137, 243
hardware resource configuration 15
HBase/YCSB 131
HDFS
configuration files 22
installation 17
storage mode 21
update environment variables 22
update other configuration files 22
HDFS protocol nodes
configure 21
HDFS transparency
administer 295
application interaction 26
application interface 26
command line 26
configuration 17
DataNode 10, 38, 198, 309
DFS Client 10
docker support 43

HDFS transparency (*continued*)

FPO mode 13
FPO nodes 13
hardware and software requirements 14
high availability 32
high availability configuration 32, 35
installation 17
installation of 17
integration 202, 314
limitations 336
NameNode 10
overview 9
security 32
short-circuit read configuration 38, 39, 40, 198, 309
start services 24
stop services 24
unintegration 203, 317
update environment variables 23
upgrade 27
HDFS Transparency
hadoop service roles 15
hdp 137
HDP 2.6 137, 243
HDP 2.6.x 104
hdp 3.x 137
high availability 168, 287
hive 112
hive on tez 62
hive operations
import and export 63
hive schema
local Hadoop cluster 50
hive-mapreduce/tez 59

I

IBM BigInsights IOP 104
configuration 22
IBM Spectrum Scale 9, 10, 13, 17, 20, 21, 22, 27, 28, 32
administer 295
planning 137, 243
service installation 258
service management 199, 202, 311, 314
special configuration 99
storage mode 10
IBM Spectrum Scale Ambari management pack 259
IBM Spectrum Scale cluster 151
IBM Spectrum Scale information units ix
IBM Spectrum Scale Kafka 176, 295
IBM Spectrum Scale service 144, 253
install 150
Ambari bits 147
Apache Ranger 87, 299
HDP 146, 147, 247
IBM Spectrum Scale 146, 247, 258
Mpack package 152
Ranger 88, 188, 189, 299, 300
verify and test 272
install Ambari
restrict root access 219, 327
install IBM Spectrum Scale
restrict root access 219, 327
install options 148
install ranger 87
install, start and test 157
installation
BI IOP or HDP 255

installation (*continued*)
 ESS 146
installing
 BigInsight value-add 276
integration
 hadoop distributions 14
internal disks 112
issues
 kerberos enabled environment 196, 308

J

join
 HDFS transparency with native HDFS 79
 native HDFS with HDFS transparency 77

K

KDC server 196, 307
kerberos 193, 304
 disable 197, 309
kerberos security 66
 ranger policy 69
 ranger policy cases 69
kernel 140, 250
kinit
 datanodes 196, 308
 namenodes 196, 307
known information
 IBM Spectrum Scale and HDFS transparency
 integration 225, 337
known issues 97
known limitations
 Hadoop storage tiering 74
 HDFS transparency 84

L

license planning 11
limitation
 functional 223, 332
limitations
 IBM Spectrum Scale and HDFS transparency
 integration 225, 337
limitations from native HDFS 104
local native HDFS cluster 50
local repository 143, 170, 171, 253, 289, 290
log in
 Apache Ambari 148, 153
 Ranger UI 92, 193, 304

M

management GUI 222, 331
mapreduce 110
MapReduce cases 57
maximal map calculation 124
memory tuning 116
migration
 HDP 2.6.2 283
move
 namenode 215, 322
move namenode
 integrated state 215, 323

move Namenode
 HA cluster 215, 323
 non-HA cluster 216, 323
multiple hadoop clusters 43
multiple HDFS transparency clusters 46

N

native HDFS 100, 223, 332
 install ranger 87
Native HDFS encryption 103
network validation 141, 251
NFS/SMB 106
node
 add 206, 319
 Ambari GPFS 219, 327
 delete 213, 321
node management 206, 319
node roles 14
Nodes
 OS tuning 19
NTP 140, 250

O

OS repository 171, 289
overview
 hadoop tiering 48
 software stack 254

P

parameter checklist 168, 287
partitioning
 function matrix 186, 297
password-less
 root 141, 251
patch
 GPFS Ambari 277
performance sizing 125
physical node
 HDFS transparency cluster 43
policy file 146, 247
Post setup 157
pre-existing IBM Spectrum Scale cluster 258
problem determination 107

R

rack mapping 185, 296
ranger
 disable 96, 193, 304
 secure HDFS 94
ranger auditing 96, 304
ranger policy 68
reduce task calculation 124
remote HDFS transparency cluster 50
remove
 IBM Spectrum Scale Hadoop connector 27
removing
 Hadoop connector 2.4 27
repository
 MySQL community edition 173, 292
resource manager HA 168

restart order
 HDFS and IBM Spectrum Scale 201, 314
restrict root access 219, 327
revert
 native HDFS 217, 324
review 157
rolling upgrade 30
root ssh access
 password-less 18
rules
 failure group 185, 296
run
 service check 199, 312
run distcp
 kerberized cluster test 73
 non-kerberized cluster test 73
run hive 62
run MapReduce 57
run spark test 58

S

select version 148
SELinux 140, 250
service actions 199, 311
service configurations
 modify 200, 312
service management 199, 311
services
 stop all 200, 312
set
 IBM Spectrum Scale configuration 295
set up
 Ambari server 148
 BigInsights IOP 138, 248
 HA 167
 HDFS Transparency package 138, 248
 Hortonworks Data Platform (HDP) 138, 248
set up HA 168
setup
 BigInsights IOP 248
 HDFS transparency package 139, 249
 Hortonworks Data Platform (HDP) 139, 249
 IBM Spectrum Scale file system 139, 249
share storage 2
shared storage 113
 rack locality 97
shared storage mode
 node roles 14
short circuit write 41
simple NSD File 144, 246
single namespace 79
single viewfs namespace 77
Snap data collection 229, 335
snapshot support 105
software packages
 mirroring repository server 170, 289
software requirements 14
software stack
 installation 254
spark cases 58
Spark over Spectrum Scale 109
spark tuning 133
special tuning 120
standard NSD file 145, 246
stanza file 245
start 150

start (*continued*)
 Ambari server 148
starting services
 manually 170, 289
stop all 200, 312
storage
 additional features 8
storage architecture 1
storage models 114
summary 157
summary page 150
support matrix 137
sync HDFS transparency 23

T

Teragen 126
TeraSort 127
test 150
TPC-DS cases 64
TPC-DS test 64
TPC-H 129
troubleshooting
 Hadoop connector 335
 HDFS transparency 335
 value-add 277
tuning
 configuration 115
 hive 128
 spectrum scale FPo 116
 Spectrum scale FPO 117
 spectrum scale over shared storage or ess 119
 Yarn for ESS/Shared storage 124
Tuning over ESS/Shared Storage 116
tuning yarn 122

U

uninstall
 ambari stack 276
 IBM Spectrum Scale Mpack 167
 IBM Spectrum Scale mpack and service 274
 IBM Spectrum Scale service 167
updating
 GPFS yum repo 277
upgrade
 HDFS transparency 161, 280
 HDFS transparency cluster 29
 HDFS Transparency NameNode 30, 31
 HDP 2.6.x to HDP 3.0 164
 IBM Spectrum Scale file system 163, 282
 IBM Spectrum Scale service MPack 158, 277
upgrading
 BigInsights IOP 277
 Hadoop connector 277
 HDFS Transparency 277
user ids 142, 252

V

value-add services
 BigInsights 276
verify
 Mpack version 205
 transparency integration 205, 318
verify environment 52

verify installation 158

W

workaround

 kerberos enabled environment 196, 308

workarounds

 IBM Spectrum Scale and HDFS transparency

 integration 225, 337

workloads information 134

Y

YCSB/HBASE 132

yum repo 201, 313

Z

zero shuffle 102

Zookeeper 176, 295



Product Number: 5725-Q01
5641-GPF
5725-S28
5765-ESS

Printed in USA

SC27-9284-00

