Hortonworks Data Platform

New Book

(Jul 21, 2015)

Hortonworks Data Platform: New Book

Copyright © 2012-2015 Hortonworks, Inc. Some rights reserved.

The Hortonworks Data Platform, powered by Apache Hadoop, is a massively scalable and 100% open source platform for storing, processing and analyzing large volumes of data. It is designed to deal with data from many sources and formats in a very quick, easy and cost-effective manner. The Hortonworks Data Platform consists of the essential set of Apache Hadoop projects including MapReduce, Hadoop Distributed File System (HDFS), HCatalog, Pig, Hive, HBase, Zookeeper and Ambari. Hortonworks is the major contributor of code and patches to many of these projects. These projects have been integrated and tested as part of the Hortonworks Data Platform release process and installation and configuration tools have also been included.

Unlike other providers of platforms built using Apache Hadoop, Hortonworks contributes 100% of our code back to the Apache Software Foundation. The Hortonworks Data Platform is Apache-licensed and completely open source. We sell only expert technical support, training and partner-enablement services. All of our technology is, and will remain free and open source.

Please visit the Hortonworks Data Platform page for more information on Hortonworks technology. For more information on Hortonworks services, please visit either the Support or Training page. Feel free to Contact Us directly to discuss your specific needs.

Table of Contents

List of Tables

1. Upgrading HDP Manually

This chapter describes how to upgrade a Hortonworks Data Platform for Windows cluster from HDP 2.2 to HDP 2.3. This is an "in-place" upgrade, where your data and metadata do not need to be moved during the upgrade process. However, services must be stopped and reinstalled.

The upgrade instructions in this section apply only to HDP clusters that are not configured for high availability.

All of the instructions in this section are executed at the MS-DOS command prompt.

- Before You Begin

- Back Up Critical HDFS Metadata

- Back Up Your Configuration Files

- Stop Running HDP Services

- Uninstall HDP on All Nodes

- Updating the HDP Cluster Properties File

- Installing HDP 2.3 and Maintaining Your Prior Data

- Updating Hive Metastore Database Table Structures

- Upgrade HDFS Metadata

- Upgrade HBase

- Upgrade Oozie

- Upgrade Tez

- Adding Node Label Support for YARN Applications

- Configure Ranger Security

- Using Apache Slider

- Starting HDP Services

- Setting up HDP

- Validating Your Data

- Verify that HDP Services are Working

- Finalizing Your Upgrade

- Troubleshooting

1.1. Before You Begin

To prepare for upgrade, gather the following information:

- **Cluster Properties**: Save a copy of your existing cluster properties file (for example, `c:\hdp\cluster.properties`) as a starting point for configuring new HDP components.

- **Files required for uninstalling HDP**: Confirm that the uninstallation packages are available on each node in the cluster. The uninstallation package is in `c:\HadoopInstallFiles`. Without this package, the Uninstaller cannot remove HDP from the node.

- **HDP data directory**: Identify where user data and metadata are being stored by HDFS and MapReduce. These directories are retained during the upgrade process. The data directory is defined in the cluster properties file; for example:

  ```
  #Data directory HDP_DATA_DIR=c:\hdp_data
  ```

- **New HDP components:** Identify the hosts for new service components (such as Falcon, Storm, Knox, and Ranger), and ensure that the hosts meet all the prerequisites for the new components.

- **Hadoop user password:** You must run some of the upgrade and configuration steps with the `hadoop` user, so you must know the password. If you do not know the password for the hadoop user, then reset the password to a known password and continue. To change the password, enter:

  ```
  net user hadoop NewPassword
  ```

1.2. Back Up Critical HDFS Metadata

Back up the following critical data before attempting an upgrade.

1. On the node that hosts the NameNode, open the Hadoop Command Line shortcut (or open a command window in the Hadoop directory). As the `hadoop` user, go to the HDFS home directory:

   ```
   runas /user:hadoop "cmd /K cd %HDFS_DATA_DIR%"
   ```

2. Run the `fsck` command to fix any file system errors.

   ```
   hdfs fsck / -files -blocks -locations > dfs-old-fsck-1.log
   ```

   The console output is printed to the `dfs-old-fsck-1.log` file.

3. Capture the complete namespace directory tree of the file system:

   ```
   hdfs dfs -ls -R / > dfs-old-lsr-1.log
   ```

4. Create a list of DataNodes in the cluster:

   ```
   hdfs dfsadmin -report > dfs-old-report-1.log
   ```

5. Capture output from the `fsck` command:

   ```
   hdfs fsck / -blocks -locations -files > fsck-old-report-1.log
   ```

   Verify that there are no missing or corrupted files/replicas in the `fsck` command output.

6. Save the HDFS namespace:

a. Place the NameNode in safe mode, to keep HDFS from accepting any new writes:

```
hdfs dfsadmin -safemode enter
```

b. Save the namespace.

```
hdfs dfsadmin -saveNamespace
```

> **Warning**
>
> From this point on, HDFS should not accept any new writes. Stay in safe mode!

c. Finalize the namespace:

```
hdfs namenode -finalize
```

d. On the machine that hosts the NameNode, copy the following checkpoint directories into a backup directory:

```
%HDFS_DATA_DIR%\hdfs\nn\edits\current
%HDFS_DATA_DIR%\hdfs\nn\edits\image
%%HDFS_DATA_DIR%\hdfs\nn\edits\previous.checkpoint
```

1.3. Back Up Your Configuration Files

Copy customized configuration files from the home directories of each of the components to a backup directory. Configuration files are in the Component Home directory, including:

- %HADOOP_CONF_DIR%

- %HBASE_CONF_DIR%

- %HCAT_HOME%\etc

- %HIVE_CONF_DIR%

- %OOZIE_HOME%\conf

- %ZOOKEEPER_HOME%\conf

- %TEZ_HOME%\conf

1.4. Stop Running HDP Services

Stop all running services. From any node in the cluster, enter the following commands to stop the remote services and then stop the local services:

```
%HADOOP_NODE_INSTALL_ROOT%\stop_remote_hdp_services.cmd
%HADOOP_NODE_INSTALL_ROOT%\stop_local_hdp_services.cmd
```

1.5. Uninstall HDP on All Nodes

At each cluster node, uninstall HDP for Windows using either of the following methods:

- **From the Command line**: If you have the HDP 2.3.0 MSI installer
  (`hdp-2.3.0.0.winpkg.msi`), you can uninstall HDP from the command prompt. At
  each node in the cluster, open the command prompt as an administrator and enter:

```
msiexec /lv hdp_uninstall.log /qb /x c:\MSI_INSTALLER\hdp-2.3.0.0-GA.winpkg.
msi HDP_DIR=c:\hdp DESTROY_DATA=no
```

- **From the Control Panel:** Open Programs and Features, right-click on `Horton Works Data Platform for Windows` and select `Uninstall`.

  The uninstall option keeps existing data in place, maintaining the data directories for
  HDFS and MapReduce.

1.6. Updating the HDP Cluster Properties File

To keep the same metadata and user data when upgrading, use the directory settings and
database settings in the cluster properties file of your current version to form the basis of
your new cluster properties file.

You should also ensure that your cluster properties file for HDP 2.3 is updated with all the
relevent parameters. A sample cluster properties file is provided below, and for complete
information on manually creating a cluster properties file see Defining Cluster Properties in
Installing HDP on Windows.

1.6.1. Sample Cluster Properties File

The following snapshot illustrates a sample cluster properties file:

```
A Typical Hadoop Cluster.
 #Log directory
 HDP_LOG_DIR=d:\hadoop\logs

 #Data directory
 HDP_DATA_DIR=d:\hadoop\data
 HDFS_NAMENODE_DATA_DIR=d:\hadoop\data\hdfs\nn,c:\hdpdata,d:\hdpdatann
 HDFS_DATANODE_DATA_DIR=d:\hadoop\data\hdfs\dn,c:\hdpdata,d:\hdpdatadn

 #Hosts
 NAMENODE_HOST=onprem-ranger1
 SECONDARY_NAMENODE_HOST=onprem-ranger1
 HIVE_SERVER_HOST=onprem-ranger1
 OOZIE_SERVER_HOST=onprem-ranger1
 WEBHCAT_HOST=onprem-ranger1
 FLUME_HOSTS=onprem-ranger1
 HBASE_MASTER=onprem-ranger1
 HBASE_REGIONSERVERS=onprem-ranger2
 SLAVE_HOSTS=onprem-ranger2
 ZOOKEEPER_HOSTS=onprem-ranger1
 KNOX_HOST=onprem-ranger2
 STORM_SUPERVISORS=onprem-ranger2
 STORM_NIMBUS=onprem-ranger1
```

```
SPARK_JOB_SERVER=onprem-ranger1
SPARK_HIVE_METASTORE=metastore
IS_SLIDER=

#Database host
DB_FLAVOR=mssql
DB_PORT=9433
DB_HOSTNAME=singlehcatms7.cloudapp.net

#Hive properties
HIVE_DB_NAME=onpremranger1hive
HIVE_DB_USERNAME=hive
HIVE_DB_PASSWORD=hive
HIVE_DR=YES

#Oozie properties
OOZIE_DB_NAME=onpremranger1oozie
OOZIE_DB_USERNAME=oozie
OOZIE_DB_PASSWORD=oozie

#ASV/HDFS properties
DEFAULT_FS=HDFS
RESOURCEMANAGER_HOST=onprem-ranger1
IS_TEZ=yes
ENABLE_LZO=yes
RANGER_HOST=onprem-ranger1
RANGER_ADMIN_DB_HOST=localhost
RANGER_ADMIN_DB_PORT=3306
RANGER_ADMIN_DB_ROOT_PASSWORD=hcattest
RANGER_ADMIN_DB_DBNAME= xasecure
RANGER_ADMIN_DB_USERNAME= xaadmin
RANGER_ADMIN_DB_PASSWORD=admin
RANGER_AUDIT_DB_HOST=localhost
RANGER_AUDIT_DB_PORT=3306
RANGER_AUDIT_DB_ROOT_PASSWORD=hcattest
RANGER_EXTERNAL_URL=http://localhost:6080
RANGER_AUDIT_DB_DBNAME= xasecure
RANGER_AUDIT_DB_USERNAME= xalogger
RANGER_AUDIT_DB_PASSWORD=xalogger
RANGER_AUTHENTICATION_METHOD=LDAP
RANGER_LDAP_URL=ldap://71.127.43.33:389
RANGER_LDAP_USERDNPATTERN=uid={0},ou=users,dc=xasecure,dc=net
RANGER_LDAP_GROUPSEARCHBASE=ou=groups,dc=xasecure,dc=net
RANGER_LDAP_GROUPSEARCHFILTER=(member=uid={0},ou=users,dc=xasecure,dc=net)
RANGER_LDAP_GROUPROLEATTRIBUTE=cn
RANGER_POLICY_ADMIN_URL=http://localhost:6080
RANGER_HDFS_REPO=hadoopdev
RANGER_HIVE_REPO=hivedev
RANGER_HBASE_REPO=hbasedev
RANGER_KNOX_REPO=knoxdev
RANGER_STORM_REPO=stormdev
RANGER_SYNC_INTERVAL=360
RANGER_SYNC_LDAP_URL=ldap://10.0.0.4:389
RANGER_SYNC_LDAP_BIND_DN=cn=Administrator,cn=users,dc=hwqe,dc=net
RANGER_SYNC_LDAP_BIND_PASSWORD=Horton!#%works
RANGER_SYNC_LDAP_USER_SEARCH_SCOPE=sub
RANGER_SYNC_LDAP_USER_OBJECT_CLASS=person
RANGER_SYNC_LDAP_USER_NAME_ATTRIBUTE=cn
RANGER_SYNC_LDAP_USER_GROUP_NAME_ATTRIBUTE=memberof,ismemberof
RANGER_SYNC_LDAP_USERNAME_CASE_CONVERSION=lower
```

```
RANGER_SYNC_LDAP_GROUPNAME_CASE_CONVERSION=lower
AUTHSERVICEHOSTNAME=localhost
AUTHSERVICEPORT=5151
RANGER_SYNC_LDAP_USER_SEARCH_BASE=cn=users,dc=hwqe,dc=net
POLICYMGR_HTTP_ENABLED=true
REMOTELOGINENABLED=true
SYNCSOURCE=LDAP
```

1.7. Installing HDP 2.3 and Maintaining Your Prior Data

To install HDP on all your nodes while maintaining your prior data:

1. Download the HDP for Windows MSI installer.

2. Copy the installer and the new cluster properties file to all nodes of the cluster.

3. Run the following command on each node in the cluster:

```
msiexec /qb /i "c:\MSI_INSTALL\hdp-2.3.0.0.winpkg.msi"
/lv "hdp.log"
HDP_LAYOUT="c:\MSI_INSTALL\clusterproperties.txt"
HDP_DIR="c:\hdp"
HDP_USER_PASSWORD=password
DESTROY_DATA="no"
HDP="yes"
RANGER="YesorNo"
KNOX_MASTER_SECRET=password
```

> **Note**
>
> Be aware that new command line properties have been added to support optional HDP components. For more information, see Update the HDP Cluster Properties File and About the MSI Installer and HDP Public Properties.

4. Verify that you have installed HDP on all nodes of your cluster. Do NOT start any services yet.

1.8. Updating Hive Metastore Database Table Structures

Hive stores metadata in a SQL Server database. After you upgrade HDP and its components, you need to update the Hive metastore database table structures to ensure they are compatible with the latest version of Hive.

If you are upgrading from HDP 2.2, run the following scripts, in order, on your SQL Server database. You can find the scripts located in the Hive installation directory: `C:\hdp\hive-1.2.1.2.3.0.0-2543\scripts\metastore\upgrade\mssql`.

1. upgrade-0.14.0-to-1.1.0.mssql.sql

2. upgrade-1.1.0-to-1.2.0.mssql.sql

If you are upgrading from HDP 2.1, run the following scripts, in order, on your SQL Server database:

1. upgrade-0.13.0-to-0.14.0.mssql.sql

2. upgrade-0.14.0-to-1.1.0.mssql.sql

3. upgrade-1.1.0-to-1.2.0.mssql.sql

> **Note**
>
> If you use a new database name and define new users, you must add this new information into the cluster properties file that you use to upgrade.

1.9. Upgrade HDFS Metadata

To upgrade the HDFS Metadata, run the following steps on your NameNode:

1. Run the NameNode upgrade as the `hadoop` user. Note: in High Availability cluster, JournalNodes must be started before running `hdfs namenode -upgrade`.

   ```
   runas /user:hadoop "cmd /K hdfs namenode -upgrade"
   ```

2. At each DateNode, start the datanode service:

   ```
   sc start datanode
   ```

3. Leave the command prompt open until the process completes. To see the status of the upgrade, open a browser window and connect to the NameNode on port 50070 (`http://namenode-host:50070`).

   The amount of time it takes to upgrade HDFS depends upon the amount of data and number of nodes in your environment.

4. In the NameNode Administrative Interface, verify that the number of DataNodes matches the number of DataNodes in your environment.

5. Abort the command prompt using control-c to end the NameNode upgrade process.

6. Start the NameNode service:

   ```
   sc start namenode
   ```

7. Open a browser, connect to the NameNode on port 50070 (`http://namenode-host:50070`), and verify that SafeMode is off.

1.10. Upgrade HBase

To upgrade HBase, run the following commands as the `hadoop` user at both the HBase Master and the RegionServers hosts:

1. Start the ZooKeeper service:

```
sc start zkServer
```

2. Upgrade HBase by running the following command:

```
%HBASE_HOME%\bin\hbase.cmd upgrade -execute
```

If the command completes successfully, you should see a completed Znode upgrade with no errors.

1.11. Upgrade Oozie

To upgrade Oozie, run the following commands as the `hadoop` user:

1. Run `ooziedb.cmd`:

```
runas /user:hadoop "%OOZIE_HOME%\bin\ooziedb.cmd upgrade -run"
```

2. Replace your configuration: At each server and client, copy `oozie\conf` from the backup to the `oozie\conf` directory.

3. Replace the contents of `/user/hdfs/share` in HDFS. At the Oozie server host:

a. Back up the `/user/hdfs/share` folder in HDFS, and then delete it. If you have any custom files in this folder, back them up separately. Add them back after the share folder is updated.

```
mkdir c:\tmp\oozie_tmp
runas /user:hadoop "cmd /c hdfs dfs -copyToLocal /user/hdfs/share c:\tmp\
oozie_tmp\oozie_share_backup"
runas /user:hadoop "cmd /c hdfs dfs -rm -r /user/hdfs/share"
```

b. Add the latest share libraries.

```
runas /user:hadoop "cmd /c hdfs dfs -copyFromLocal
%OOZIE_HOME%\share / user/hdfs/.
```

c. Start the Oozie service:

```
sc start oozie
```

1.12. Upgrade Tez

If your cluster properties file specifies `IS_TEZ=yes` (use Tez for Hive), perform the following steps using the Hadoop user `hadoop`:

1. Open the command prompt with the `hadoop` account:

```
runas /user:hadoop cmd
```

2. Make a Tez application directory in HDFS:

```
%HADOOP_HOME%\bin\hdfs dfs -mkdir /apps/tez
```

3. Allow all users read and write access:

```
%HADOOP_HOME%\bin\hdfs dfs -chmod -R 755 /apps/tez
```

4. Change the owner of the file to `hadoop`:

```
%HADOOP_HOME%\bin\hdfs dfs -chown -R hadoop:users /apps/tez
```

5. Copy the Tez home directory on the local machine, into the HDFS `/apps/tez` directory:

```
%HADOOP_HOME%\bin\hdfs dfs -put %TEZ_HOME%\* /apps/tez
```

6. Remove the Tez configuration directory from the HDFS Tez application directory:

```
%HADOOP_HOME%\bin\hdfs dfs -rm -r -skipTrash /apps/tez/conf
```

7. Ensure that the following properties are set in the `%HIVE_HOME%\conf\hive-site.xml` file:

Table 1.1. Property value information

Property	Default Value	Description
hive.auto.convert.join. noconditionaltask	true	Specifies whether Hive optimizes converting common JOIN statements into MAPJOIN statements. JOIN statements are converted if this property is enabled and the sum of size for n-1 of the tables/ partitions for an n-way join is smaller than the size specified with the hive.auto.convert.join. noconditionaltask.size property.
hive.auto.convert.join. noconditionaltask.size	10000000 (10 MB)	Specifies the size used to calculate whether Hive converts a JOIN statement into a MAPJOIN statement. The configuration property is ignored unless hive.auto.convert.join. noconditionaltask is enabled.
hive.optimize. reducededuplication. min.reducer	4	Specifies the minimum reducer parallelism threshold to meet before merging two MapReduce jobs. However, combining a mapreduce job with parallelism 100 with a mapreduce job with parallelism 1 may negatively impact query performance even with the reduced number of jobs. The optimization is disabled if the number of reducers is less than the specified value.
hive.tez.container.size	-1	By default, Tez uses the java options from map tasks. Use this property to override that value. Assigned value must match value specified for mapreduce.map.child.java.opts.
hive.tez.java.opts	n/a	Set to the same value as `mapreduce.map.java.opts`.

Adjust the settings above to your environment where appropriate; `hive-default.xml.template` contains examples of the properties.

8. To verify that the installation process succeeded, run smoke tests for Tez and Hive.

1.13. Adding Node Label Support for YARN Applications

Node labels can be used to restrict YARN applications so that they run only on cluster nodes that have a specified node label. Node Labels are supported on Windows. To enable node label support, make the following changes. See the Linux Node Labels documentation for more information. If you do not plan to use node labels, none of these changes are needed.

1. Open the command prompt using the `hadoop` account:

```
runas /user:hadoop cmd
```

2. Create a top-level YARN application directory in HDFS:

```
%HADOOP_HOME%\bin\hdfs dfs -mkdir /system/yarn/node-labels
```

3. Make sure permissions are set for write access from the `hadoop` account (rwx for all the directories in the path).

4. Change the owner of the file to `hadoop`:

```
%HADOOP_HOME%\bin\hdfs dfs -chown -R hadoop:users /apps/tez
```

5. Add the following property values to `yarn-site.xml`:

Table 1.2. Node label configuration

Property	Value
yarn.node-labels.manager-class	org.apache.hadoop.yarn.server.resourcemanager. nodelabels.RMNodeLabelsManager
yarn.node-labels.fs-store.root-dir	/system/yarn/node-labels
yarn.node-labels.fs-store.retry-policy-spec	2000,500

6. To verify that the installation process succeeded, run smoke tests for Tez and Hive as described in Validating the Installation.

1.14. Configure Ranger Security

Apache Ranger delivers a comprehensive approach to security for a Hadoop cluster. It provides central security policy administration across the core enterprise security requirements of authorization, accounting, and data protection. For more information about requirements and settings, see Minimum System Requirements and Defining Cluster Properties.

The Ranger Policy Manager and Ranger UserSync are installed in only one host (specified in the HDP Setup Ranger Host parameter); the Ranger plug-ins for corresponding components are installed wherever those components are installed.

Make sure that the MySQL database used by Ranger is set up to connect from any host in the cluster. For multiple hosts, set up the Ranger MySQL database to connect from any host in the cluster using the root password. To do this, enter:

```
grant all privileges on *.* to 'root'@'%' identified by
RootPasswordHere' ;flush privileges;
```

Note

In HDP v2.2, MySQL is the only database supported for use with Ranger.

1.15. Using Apache Slider

Apache Slider lets you deploy distributed applications across a Hadoop cluster. On the Windows platform, Slider application packages are included in the Windows Installer msi for HBase and Storm. (Accumulo is available on Linux, but is not currently supported on Windows). Until more information is available, please see the Linux document Running Applications on YARN Using Slider. .

1.16. Starting HDP Services

The following steps describe how to start local and remote services. For more information, see Managing HDP on Windows.

1. Start local services on the Master Nodes:

   ```
   %HADOOP_NODE%\start_local_hdp_services.cmd
   ```

   Wait for the Master Node services to start up before continuing.

2. At any Master Node, start all slave node services:

   ```
   %HADOOP_NODE%\start_remote_hdp_services.cmd
   ```

3. At the Knox Gateway:

   ```
   %HADOOP_NODE%\start_local_hdp_services.cmd
   ```

4. If the Apache Hadoop REST service is disabled, run the following command to enable it:

   ```
   sc config name= rest start= demand
   ```

5. If REST HDP services are not already started, start them with the following command:

   ```
   sc start rest
   ```

6. Smoke test your installation as described in Validating the Installation.

1.17. Setting up HDP

After starting the local and remote services, run the following commands to set up HDP:

1. Open a command prompt using the `hadoop` user in the Hadoop home directory:

```
runas /user:hadoop "cmd /K cd %HDFS_HOME%\bin"
```

2. Make, and then set, ownership and permissions for the following directories:

```
hdfs dfs -mkdir -p /mapred/history/done /mapred/history/
done_intermediate
hdfs dfs -chmod -R 1777 /mapred/history/done_intermediate
hdfs dfs -chmod 770 /mapred/history/done
hdfs dfs -chown -R hadoop:hadoopUsers /mapred
hdfs dfs -chmod 755 /mapred/mapred/history
hdfs dfs -mkdir /tmp
hdfs dfs -chmod 777 /tmp
hdfs dfs -mkdir /app-logs
hdfs dfs -chown hadoop:hadoopUsers /app-logs
hdfs dfs -chmod 1777 /app-logs hdfs dfs -mkdir -p /yarn /yarn/generic-
history/
hdfs dfs -chmod -R 700 /yarn
hdfs dfs -chown -R hadoop:hadoop /yarn
hdfs dfs -mkdir -p /apps/hbase/data
hdfs dfs -chown hadoop:hadoop /apps/hbase/data
hdfs dfs -chown hadoop:hadoop /apps/hbase
hdfs dfs -mkdir -p /user/hbase
hdfs dfs -chown hadoop:hadoop /user/hbase
hdfs dfs -mkdir -p /user/hive /hive/warehouse
hdfs dfs -chown hadoop:hadoop /user/hive
hdfs dfs -chmod -R 755 /user/hive
hdfs dfs -chown -R hadoop:users /hive/warehouse
hdfs dfs -chmod -R 777 /hive/warehouse
hdfs dfs -mkdir -p /user/hcat
hdfs dfs -chown -R hadoop:hadoop /user/hcat
hdfs dfs -mkdir -p /apps/webhcat
hdfs dfs -chown -R hadoop:users/apps/webhcat
hdfs dfs -chmod -R 755 /apps/webhcat
```

3. When using Tez for Hive, you must also Upgrade Tez.

1.18. Validating Your Data

Verify that your data is intact by comparing the HDFS data directory tree with your previous directory tree.

1. Run the following command as the `hadoop` user, from the Hadoop home directory:

```
runas /user:hadoop "cmd /K cd %HDFS_HOME%\bin"
```

2. Run an `lsr` report on your upgraded system. Enter the following at the Hadoop command line:

```
hdfs fs -ls R / > fs-new-lsr-1.log
```

3. Compare the directory listing to the older HDP directories. All old directories, files and timestamps should match. There will also be some new entries in the HDP directory listing:

   • `/apps/hbase` is used by HBase

- /mapred/system/jobtracker should have a new timestamp

4. Run an fsck report on your upgraded system. From the Hadoop command line:

```
hdfs fsck / -blocks -locations -files > fsck-new-report-1.log
```

5. To check the validity of your current HDFS data, compare this fsck report to the report generated before the upgrade.

1.19. Verify that HDP Services are Working

To verify that the HDP services work as expected, run the provided smoke tests as the hadoop user. (Alternately, create a smoke test user in HDFS and run the test as that user. See this Guide's Appendix.)

You can also verify HDP 2.0 services by running the following Desktop Shortcuts as the hadoop user or smoketest user:

- Hadoop Name Node status

- HBase Master status

- Hadoop YARN status

1.20. Finalizing Your Upgrade

When you are satisfied that HDP is successfully functioning with the data from your HDP 2.2 cluster, finalize the upgrade.

> **Warning**
>
> Once you have finalized the upgrade, you cannot revert.

1. Open a command prompt with the hadoop user in the HDFS Home directory:

```
runas /user:hadoop "cmd /K cd %HDFS_HOME%\bin"
```

2. Finalize the upgrade:

```
hdfs dfsadmin -finalizeUpgrade
```

1.21. Troubleshooting

To validate the installation, run the smoke test suite as described in Validating the Installation.

If the HBase RegionServer and Master do not start, check to see if a parenthesis in the Path variable caused a problem during setup of the services. To fix this, run the following commands from an Administrator command prompt:

```
%HBASE_HOME%\bin\hbase.cmd --service master start >
%HBASE_HOME%\bin\master. xml
%HBASE_HOME%\bin\hbase.cmd --service regionserver start >
%HBASE_HOME%\bin\ regionserver.xml
%HBASE_HOME%\bin\hbase.cmd --service rest >
%HBASE_HOME%\bin\rest.xml
%HBASE_HOME%\bin\hbase.cmd --service thrift >
%HBASE_HOME%\bin\thrift.xml
```

Restart the HBase services.

If Flume is not starting, it could be because flumeservice.xml is missing. To fix this, navigate to `%FLUME_HOME%\bin` and locate the `flumeagent.xml` file. If the file does not exist, locate `flumeservice.xml` file and rename it to `flumeagent.xml`. After the file is renamed, go to Windows Services and restart the Flume agent service.

To help diagnose installation issues, see log file information in File Locations, Logging, and Common HDFS Commands.