Hortonworks Data Platform

Ambari Upgrade Guide

(Apr 1, 2014)

docs.hortonworks.com

Hortonworks Data Platform: Ambari Upgrade Guide

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 Ambari Server to 1.5.0

This procedure upgrades Ambari Server from version 1.2.5 and above to 1.5.0. If your current Ambari Server version is 1.2.4 or below, you must upgrade the Ambari Server version to 1.2.5 before upgrading to version 1.5.0. Upgrading the Ambari Server version does not change the underlying Hadoop Stack.

Note

You must know the location of the Nagios server for Step 9. Use the **Services View**-> **Summary** panel to locate the host on which it is running.

1. Stop the Nagios and Ganglia services. In **Ambari Web**:

   a. Browse to **Services** and select each service.

   b. Use **Service Actions** to stop the service.

2. Stop the Ambari Server and all Ambari Agents. From the Ambari Server host:

   ```
   ambari-server stop
   ```

   From each Ambari Agent host:

   ```
   ambari-agent stop
   ```

3. Get the new Ambari bits. Use `wget` to fetch the repository file and replace the old repository file with the new repository file on every host.

   - Fetch the new repository file:

     For RHEL/CentOS 5/Oracle Linux 5

     ```
     wget http://public-repo-1.hortonworks.com/ambari/centos5/1.x/updates/1.5.1/ambari.repo
     ```

     For RHEL/CentOS 6/Oracle Linux 6

     ```
     wget http://public-repo-1.hortonworks.com/ambari/centos6/1.x/updates/1.5.0/ambari.repo
     ```

     For SLES 11

     ```
     wget http://public-repo-1.hortonworks.com/ambari/suse11/1.x/updates/1.5.1/ambari.repo
     ```

     

     ### Important

     Check your current directory before you download the new repository file to make sure that there are no previous versions of the file. If you do not, and a previous version exists, the new download will be saved with a numeric extension such as `ambari.repo.1`. Make sure that the version you copy is the new version.

   - Replace the old repository file with the new repository file.

For RHEL/CentOS 5/Oracle Linux 5

```
cp ambari.repo /etc/yum.repos.d/ambari.repo
```

For RHEL/CentOS 6/Oracle Linux 6

```
cp ambari.repo /etc/yum.repos.d/ambari.repo
```

For SLES 11

```
cp ambari.repo /etc/zypp/repos.d/ambari.repo
```

Note

If your cluster does not have access to the Internet, set up a local repository with this data before you continue. See Configure the Local Repositories for more information.

4. Upgrade Ambari Server.

Note

Ambari Server no longer automatically turns `iptables` off. Check your installation setup to make sure that you are not relying on this function. After you have upgraded the server you must either disable iptables manually or make sure that you have all the appropriate ports available.

From the Ambari Server host:

- RHEL/CentOS/Oracle Linux

```
yum clean all
yum upgrade ambari-server ambari-log4j
```

- SLES

```
zypper clean
zypper up ambari-server ambari-log4j
```

5. Check for upgrade success:

- As the process runs, the console should produce output similar, although not identical, to this:

```
Setting up Upgrade Process
Resolving Dependencies
--> Running transaction check
---> Package ambari-server.x86_64 0:1.2.2.3-1 will be updated
---> Package ambari-server.x86_64 0:1.2.2.4-1 will be updated ...
---> Package ambari-server.x86_64 0:1.2.2.5-1 will be an update ...
```

After the process is complete, check each host to make sure the new 1.5.0 files have been installed:

```
rpm -qa | grep ambari
```

- If the upgrade fails, the console should produce output similar to this:

```
Setting up Upgrade Process
No Packages marked for Update
```

6. Check to see if you have a folder named `/etc/ambari-server/conf.save`. If you do, rename it back:

```
mv /etc/ambari-server/conf.save /etc/ambari-server/conf
```

7. Upgrade the Ambari Server schema. From the Ambari Server host:

```
ambari-server upgrade
```

8. Upgrade the Ambari Agent on all hosts. From each Ambari Agent host:

- RHEL/CentOS/Oracle Linux

```
yum upgrade ambari-agent ambari-log4j
```

- SLES

```
zypper up ambari-agent ambari-log4j
```

Note

If you get a warning that begins "There are some running programs that use files deleted by recent upgrade," you may ignore it.

9. Check to see if you have a file named `/etc/ambari-agent/conf.save` on each Agent host. If you do, rename it back. On each Agent host:

```
mv /etc/ambari-agent/conf.save /etc/ambari-agent/conf
```

10. Upgrade the Nagios add-ons package. On the Nagios host:

- RHEL/CentOS/Oracle Linux

```
yum upgrade hdp_mon_nagios_addons
```

- SLES

```
zypper up hdp_mon_nagios_addons
```

11. Start the Server and the Agents on all hosts. From the Server host:

```
ambari-server start
```

From each Agent host:

```
ambari-agent start
```

12. Open **Ambari Web**. Point your browser to `http://{your.ambari.server}:8080`

Important

Refresh your browser so that it loads the new version of the code. Hold the Shift key down while clicking the refresh button on the browser. If you have problems, clear your browser cache manually and restart Ambari Server.

Use the Admin name and password you have set up to log in.

13. Start the Nagios and Ganglia services. In **Ambari Web**:

    a. Browse to **Services** and select each service.

    b. Use **Service Actions** to start the service.

14. If you have customized logging properties, you will see refresh indicators next to each service name after upgrading to Ambari 1.5.0.

Note

Restarting a service pushes the configuration properties displayed in Custom log 4j Properties to each host running components for that service.

To preserve any custom logging properties after upgrading, for each service:

a. Replace default logging properties with your custom logging properties, using Service Configs -> Custom log 4j Properties.

b. Restart all components in any services for which you have customized logging properties.

For more information about logging properties, see Customizing Your Logging Properties.

2. Upgrading the HDP Stack to 1.3.3

The stack is the coordinated set of Hadoop components that you have installed. If you have a current instance of the 1.2.0/1.2.1 stack that was installed and managed by Ambari that you want to upgrade to the current 1.3.3 version of the stack and to also upgrade to the 1.2.5 version of Ambari Server and Agents, use the following instructions. This insures that the upgraded stack can still be managed by Ambari.

If you are upgrading from the 1.3.0 stack to the 1.3.3 stack, use Section 5: Upgrading the Stack (from 1.3.0 to 1.3.3), not Section 4: Upgrading the Stack (from 1.2.* to 1.3.3).

Note

If you have already upgraded to Ambari Server 1.2.5 and just want to upgrade the HDP stack, you can skip Sections 3.4 and 3.5.

2.1. Preparing for the Upgrade

Use the following steps to prepare your system for the upgrade.

1. If you are upgrading Ambari as well as the stack, you must know the location of the Nagios and Ganglia servers for that process. Use the **Services**->**Nagios/Ganglia**-> **Summary** panel to locate the hosts on which they are running.
2. Use the **Services** view on the **Ambari Web** UI to stop all services, including MapReduce and all clients, running on HDFS. Do **not** stop HDFS yet.
3. Create the following logs and other files.

   Because the upgrade to 1.3.3 stack includes a version upgrade of HDFS, creating these logs allows you to check the integrity of the file system post upgrade. While this is not absolutely necessary, doing so is strongly encouraged.

   a. Run `fsck` with the following flags and send the results to a log. The resulting file contains a complete block map of the file system. You use this log later to confirm the upgrade.

   ```
   su $HDFS_USER
   hadoop fsck / -files -blocks -locations > /tmp/dfs-old-fsck-1.log
   ```

   where *$HDFS_USER* is the HDFS Service user (by default, `hdfs`).

   b. Capture the complete namespace of the filesystem. (The following command does a recursive listing of the root file system.)

   ```
   su $HDFS_USER
   hadoop dfs -lsr / > /tmp/dfs-old-lsr-1.log
   ```

   where *$HDFS_USER* is the HDFS Service user (by default, `hdfs`).

   c. Create a list of all the DataNodes in the cluster.

   ```
   su $HDFS_USER
   hadoop dfsadmin -report > /tmp/dfs-old-report-1.log
   ```

where *$HDFS_USER* is the HDFS Service user (by default, `hdfs`).

d. Optional: copy all or unrecoverable only data stored in HDFS to a local file system or to a backup instance of HDFS.

e. Optional: create the logs again and check to make sure the results are identical.

4. Save the namespace. You must be the HDFS service user to do this and you must put the cluster in Safe Mode.

```
hadoop dfsadmin -safemode enter
hadoop dfsadmin -saveNamespace
```

5. Copy the following checkpoint files into a backup directory. You can find the directory by using the **Services View** in the UI. Select the **HDFS** service, the **Configs** tab, in the Namenode section, look up the property **NameNode Directories**. It will be on your NameNode host.

   - `dfs.name.dir/edits` // depending on your system, may not exist
   - `dfs.name.dir/image/fsimage`

6. Stop HDFS. Make sure all services in the cluster are completely stopped.

7. If you are upgrading Hive, back up the Hive database.

8. Stop Ambari Server. On the Server host:

```
ambari-server stop
```

9. Stop Ambari Agents. On each host:

```
ambari-agent stop
```

2.2. Setting Up the Ambari Repository

This process prepares the updated repository.

1. Check to see if you have a `conf.save` directory for Ambari server and agents. If you do, move them to a back-up location:

```
mv /etc/ambari-server/conf.save/ /etc/ambari-server/conf.save.bak
```

```
mv /etc/ambari-agent/conf.save/ /etc/ambari-agent/conf.save.bak
```

2. Get the new Ambari bits. Use `wget` to fetch the repository file and replace the old repo file with the new repo file on every host.

Important

Check your current directory before you download the new repo file to make sure that there are no previous versions of the file. If you do not, and a previous version exists, the new download is saved with a numeric extension such as `ambari.repo.1`. Make sure that the version you copy is the new version.

- For RHEL/CentOS/Oracle Linux 5

```
wget http://public-repo-1.hortonworks.com/ambari/centos5/1.x/updates/1.5.
0/ambari.repo
cp ambari.repo /etc/yum.repos.d/ambari.repo
```

- For RHEL/CentOS/Oracle Linux 6

```
wget http://public-repo-1.hortonworks.com/ambari/centos6/1.x/updates/1.5.
0/ambari.repo
cp ambari.repo /etc/yum.repos.d/ambari.repo
```

- For SLES 11

```
wget http://public-repo-1.hortonworks.com/ambari/suse11/1.x/updates/1.5.0/
ambari.repo
cp ambari.repo /etc/zypp/repos.d/ambari.repo
```

Note

If your cluster does not have access to the Internet, you need to set up a local repository with this data before you continue. See Configure the Local Repositories for more information.

2.3. Upgrading Ambari

This process upgrades Ambari Server, Ambari Agents, Ganglia, and Nagios.

1. Upgrade Ambari Server. From the Ambari Server host:

   - RHEL/CentOS/Oracle Linux

   ```
   yum clean all
   yum upgrade ambari-server ambari-log4j
   ```

   - SLES

   ```
   zypper clean
   zypper up ambari-server ambari-log4j
   ```

2. Check for upgrade success:

   - As the process runs, the console should produce output similar, although not identical, to this:

   ```
   Setting up Upgrade Process
   Resolving Dependencies
   --> Running transaction check
   ---> Package ambari-agent.x86_64 0:1.2.2.3-1 will be updated
   ---> Package ambari-agent.x86_64 0:1.2.2.4-1 will be updated ...
   ---> Package ambari-agent.x86_64 0:1.2.2.5-1 will be an update ...
   ```

   After the process is complete, check each host to make sure the new 1.2.5 files have been installed:

   ```
   rpm -qa | grep ambari
   ```

- If the upgrade fails, the console should produce output similar to this:

```
Setting up Upgrade Process
No Packages marked for Update
```

3. Check to see if you have a folder named `/etc/ambari-server/conf.save`. If you do, rename it back:

```
mv /etc/ambari-server/conf.save /etc/ambari-server/conf
```

4. Upgrade the Ambari Server schema. From the Ambari Server host:

```
ambari-server upgrade
```

5. Upgrade the Ambari Agent on all hosts. From each Ambari Agent host:

- RHEL/CentOS/Oracle Linux

```
yum upgrade ambari-agent ambari-log4j
```

- SLES

```
zypper up ambari-agent ambari-log4j
```

Note

If you get a warning that begins "There are some running programs that use files deleted by recent upgrade," you can ignore it.

6. Check to see if you have a folder named `/etc/ambari-agent/conf.save` on each Agent host. If you do, rename it back. On each Agent host:

```
mv /etc/ambari-agent/conf.save /etc/ambari-agent/conf
```

7. Upgrade Ganglia and Nagios:

- Upgrade Ganglia Server. From the Ganglia Server host:

  - RHEL/CentOS/Oracle Linux

```
yum upgrade ganglia-gmond ganglia-gmetad libganglia
yum erase gweb hdp_mon_ganglia_addons
yum install ganglia-web
```

  - SLES

```
zypper up ganglia-gmond ganglia-gmetad libganglia
zypper remove gweb hdp_mon_ganglia_addons
zypper install ganglia-web
```

- Upgrade Ganglia Monitor. From every host that has Ganglia Monitor installed:

  - RHEL/CentOS/Oracle Linux

```
yum upgrade ganglia-gmond libganglia
```

  - SLES

```
zypper up ganglia-gmond libganglia
```

- Upgrade Nagios. From the Nagios Server host:
  - RHEL/CentOS/Oracle Linux

```
yum upgrade nagios
yum upgrade hdp_mon_nagios_addons
yum erase nagios-plugins-1.4.15
yum install nagios-plugins-1.4.9
```

The 1.4.9 version of the plugin may already be installed. In this case, the second step is a no-op.

  - SLES

```
zypper up nagios
zypper up hdp_mon_nagios_addons
zypper remove nagios-plugins-1.4.15
zypper install nagios-plugins-1.4.9
```

The 1.4.9 version of the plugin may already be installed. In this case, the second step is a no-op.

2.4. Upgrading the Stack (from 1.2.* to 1.3.3)

1. Update the stack version in the Server database, depending on if you are using a local repository:

```
ambari-server upgradestack HDP-1.3.3
```

2. Upgrade the HDP repository on all hosts and replace the old repo file with the new file:

Important

The file you download is named `hdp.repo`. To function properly in the system, it must be named `HDP.repo`. Once you have completed the "mv" of the new repo file to the repos.d folder, make sure there is no file named `hdp.repo` anywhere in your repos.d folder.

- For RHEL/CentOS/Oracle Linux 5

```
wget  http://public-repo-1.hortonworks.com/HDP/centos5/1.x/updates/1.3.3.
0/hdp.repo
mv hdp.repo /etc/yum.repos.d/HDP.repo
```

- For RHEL/CentOS/Oracle Linux 6

```
wget http://public-repo-1.hortonworks.com/HDP/centos6/1.x/updates/1.3.3.0/
hdp.repo
mv hdp.repo /etc/yum.repos.d/HDP.repo
```

- For SLES 11

```
wget  http://public-repo-1.hortonworks.com/HDP/suse11/1.x/updates/1.3.3.0/
hdp.repo
```

```
mv hdp.repo /etc/zypp/repos.d/HDP.repo
```

3. Upgrade the stack on all Agent hosts. Skip any components your installation does not use:

   - For RHEL/CentOS/Oracle Linux

     a. Upgrade the following components:

     ```
     yum upgrade "collectd*" "epel-release*" "gccxml*" "pig*" "hadoop*"
      "sqoop*" "zookeeper*" "hbase*" "hive*" "hcatalog*" "webhcat-tar*"
     hdp_mon_nagios_addons
     ```

     b. Check to see that the components in that list are upgraded.

     ```
     yum list installed | grep HDP-$old-stack-version-number
     ```

     None of the components from that list should appear in the returned list.

     c. Upgrade Oozie, if you are using Oozie:

     ```
     rpm -e --nopostun oozie-$old_version_number
     yum install oozie
     ```

     You can get the value of $old_version_number from the output of the previous step.

     d. Upgrade Oozie Client:

     ```
     yum upgrade oozie-client
     ```

     e. Upgrade ZooKeeper:

     i. Check to see if ZooKeeper needs upgrading.

     ```
     yum list installed | grep zookeeper
     ```

     If the displayed version number is **not** 3.4.5.1.3.2.0, you need to upgrade.

     ii. Because HBase depends on ZooKeeper, deleting the current version of ZooKeeper automatically deletes the current version of HBase. It must be re-installed. Check to see if HBase is currently installed.

     ```
     yum list installed | grep hbase
     ```

     iii. Delete the current version of ZooKeeper.

     ```
     yum erase zookeeper
     ```

     iv. Install ZooKeeper

     ```
     yum install zookeeper
     ```

     v. If you need to, re-install HBase.

     ```
     yum install hbase
     ```

     vi. Check to see if all components have been upgraded:

```
yum list installed | grep HDP-$old-stack-version-number
```

The only non-upgraded component you may see in this list is `extjs`, which does not need to be upgraded.

- For SLES

  a. Upgrade the following components:

  ```
  zypper up collectd epel-release* gccxml* pig* hadoop* sqoop* hive*
   hcatalog* webhcat-tar* hdp_mon_nagios_addons*
  yast --update hadoop hcatalog hive
  ```

  b. Upgrade ZooKeeper and HBase:

  ```
  zypper update zookeeper-3.4.5.1.3.2.0
  zypper remove zookeeper
  zypper se -s zookeeper
  ```

  You should see ZooKeeper v3.4.5.1.3.2.0 in the output.

  Install ZooKeeper v3.4.5.1.3.2.0:

  ```
  zypper install zookeeper-3.4.5.1.3.2.0
  ```

  This command also uninstalls HBase. Now use the following commands to install HBase:

  ```
  zypper install hbase-0.94.6.1.3.2.0
  zypper update hbase
  ```

  c. Upgrade Oozie:

  ```
  rpm -e --nopostun oozie-$old_version_number
  zypper update oozie-3.3.2.1.3.2.0
  zypper remove oozie
  zypper se -s oozie
  ```

  You should see Oozie v3.3.2.1.3.2.0 in the output.

  Install Oozie v3.3.2.1.3.2.0:

  ```
  zypper install oozie-3.3.2.1.3.2.0
  ```

4. Start the Ambari Server. On the Server host:

   ```
   ambari-server start
   ```

5. Start each Ambari Agent. On all Agent hosts:

   ```
   ambari-agent start
   ```

6. Because the file system version has now changed you must start the NameNode manually. On the NameNode host:

   ```
   sudo su -l $HDFS_USER -c "/usr/lib/hadoop/bin/hadoop-daemon.sh start
    namenode -upgrade"
   ```

   Depending on the size of your system, this step may take up to 10 minutes.

7. Track the status of the upgrade:

```
hadoop dfsadmin -upgradeProgress status
```

Continue tracking until you see:

```
Upgrade for version -44 has been completed.
Upgrade is not finalized.
```

Note

You finalize the upgrade later. DO NOT run the balancer before finalizing an upgrade. No block deletion occurs until you finalize the upgrade. Running the balancer before finalizing an upgrade may duplicate data blocks, and increase disk usage.

8. Open the Ambari Web GUI. If you have continued to run the Ambari Web GUI, do a hard reset on your browser. Use **Services View** to start the HDFS service. This starts the SecondaryNameNode and the DataNodes.

9. After the DataNodes are started, HDFS exits safemode. To monitor the status:

```
hadoop dfsadmin -safemode get
```

Depending on the size of your system, this may take up to 10 minutes or so. When HDFS exits safemode, this is displayed as a response to the command:

```
Safe mode is OFF
```

10. Make sure that the HDFS upgrade was successful. Go through steps 2 and 3 in Preparing for the Upgrade to create new versions of the logs and reports. Substitute "new" for "`old`" in the file names as necessary

11. Compare the old and new versions of the following:

- `dfs-old-fsck-1.log` versus `dfs-new-fsck-1.log`.

  The files should be identical unless the `hadoop fsck` reporting format has changed in the new version.

- `dfs-old-lsr-1.log` versus `dfs-new-lsr-1.log`.

  The files should be identical unless the the format of `hadoop fs -lsr` reporting or the data structures have changed in the new version.

- `dfs-old-report-1.log` versus `fs-new-report-1.log`

  Make sure all DataNodes previously belonging to the cluster are up and running.

12. Use the Ambari Web **Services** view-> Services Navigation->**Start All** to start services back up.

13. The upgrade is now fully functional but not yet finalized. Using the `finalize` command removes the previous version of the NameNode and DataNode's storage directories.

Important

After the upgrade is finalized, the system cannot be rolled back. Usually this step is not taken until a thorough testing of the upgrade has been performed.

The upgrade must be finalized before another upgrade can be performed.

Note

Directories used by Hadoop 1 services set in /etc/hadoop/conf/taskcontroller.cfg are not automatically deleted after upgrade. Administrators can choose to delete these directories after the upgrade.

To finalize the upgrade:

```
sudo su -l $HDFS_USER -c "hadoop dfsadmin -finalizeUpgrade"
```

where *$HDFS_USER* is the HDFS Service user (by default, `hdfs`).

2.5. Upgrading the Stack (from 1.3.0 to 1.3.3)

1. Update the stack version in the Server database, depending on if you are using a local repository:

```
ambari-server upgradestack HDP-1.3.3
```

2. Upgrade the HDP repository on all hosts and replace the old repo file with the new file:

Important

The file you download is named `hdp.repo`. To function properly in the system, it must be named HDP`.repo`. After you have completed the "mv" of the new repo file to the repos.d folder, make sure there is no file named `hdp.repo` anywhere in your repos.d folder.

- For RHEL/CentOS/Oracle Linux 5

```
wget  http://public-repo-1.hortonworks.com/HDP/centos5/1.x/updates/1.3.3.0/hdp.repo
mv hdp.repo /etc/yum.repos.d/HDP.repo
```

- For RHEL/CentOS/Oracle Linux 6

```
wget http://public-repo-1.hortonworks.com/HDP/centos6/1.x/updates/1.3.3.0/hdp.repo
mv hdp.repo /etc/yum.repos.d/HDP.repo
```

- For SLES 11

```
wget  http://public-repo-1.hortonworks.com/HDP/suse11/1.x/updates/1.3.3.0/hdp.repo

mv hdp.repo /etc/zypp/repos.d/HDP.repo
```

3. Upgrade the stack on all Agent hosts. Skip any components your installation does not use:

   - For RHEL/CentOS/Oracle Linux

     a. Upgrade the following components:

     ```
     yum upgrade "collectd*" "epel-release*" "gccxml*" "pig*" "hadoop*"
      "sqoop*" "zookeeper*" "hbase*" "hive*" "hcatalog*" "webhcat-tar*"
     "oozie*" hdp_mon_nagios_addons
     ```

     b. Check to see if those components have been upgraded:

     ```
     yum list installed | grep HDP-$old-stack-version-number
     ```

     The only non-upgraded component you may see in this list is `extjs`, which does not need to be upgraded.

   - For SLES

     a. Upgrade the following components:

     ```
     zypper up collectd gccxml* pig* hadoop* sqoop* hive* hcatalog* webhcat-
     tar* zookeeper* oozie* hbase* hdp_mon_nagios_addons*
     yast --update hadoop hcatalog hive
     ```

4. Start the Ambari Server. On the Server host:

   ```
   ambari-server start
   ```

5. Start each Ambari Agent. On all Agent hosts:

   ```
   ambari-agent start
   ```

6. Because the file system version has now changed you must start the NameNode manually. On the NameNode host:

   ```
   sudo su -l $HDFS_USER -c "/usr/lib/hadoop/bin/hadoop-daemon.sh start
    namenode -upgrade"
   ```

   Depending on the size of your system, this step may take up to 10 minutes.

7. Track the status of the upgrade:

   ```
   hadoop dfsadmin -upgradeProgress status
   ```

   Continue tracking until you see:

   ```
   Upgrade for version -44 has been completed.
   Upgrade is not finalized.
   ```

   

   Note

   You finalize the upgrade later.

8. Open the Ambari Web GUI. If you have continued to run the Ambari Web GUI, do a hard reset on your browser. Use **Services View** to start the HDFS service. This starts the SecondaryNameNode and the DataNodes.

9. After the DataNodes are started, HDFS exits safemode. To monitor the status:

```
hadoop dfsadmin -safemode get
```

Depending on the size of your system, this may take up to 10 minutes or so. When HDFS exits safemode, this is displayed as a response to the command:

```
Safe mode is OFF
```

10. Make sure that the HDFS upgrade succeeded. Go through steps 2 and 3 in Preparing for the Upgrade to create new versions of the logs and reports. Substitute "new" for "old" in the file names as necessary

11. Compare the old and new versions of the following files:

- `dfs-old-fsck-1.log` versus `dfs-new-fsck-1.log`.

  The files should be identical unless the `hadoop fsck` reporting format has changed in the new version.

- `dfs-old-lsr-1.log` versus `dfs-new-lsr-1.log`.

  The files should be identical unless the the format of `hadoop fs -lsr` reporting or the data structures have changed in the new version.

- `dfs-old-report-1.log` versus `fs-new-report-1.log`

  Make sure all DataNodes previously belonging to the cluster are up and running.

12. Use the Ambari Web **Services** view-> Services Navigation->**Start All** to start services back up.

13. The upgrade is now fully functional but not yet finalized. Using the `finalize` comand removes the previous version of the NameNode and DataNode storage directories.

Important

After the upgrade is finalized, the system cannot be rolled back. Do not take this step until you perform thorough testing of the upgrade.

The upgrade must be finalized, however, before another upgrade can be performed.

Note

Directories used by Hadoop 1 services set in /etc/hadoop/conf/taskcontroller.cfg are not automatically deleted after upgrade. Administrators can choose to delete these directories after the upgrade.

To finalize the upgrade:

```
sudo su -l $HDFS_USER -c "hadoop dfsadmin -finalizeUpgrade"
```

where *$HDFS_USER* is the HDFS Service user (by default, `hdfs`).

3. Upgrading the HDP Stack from 1.3.2 or later to 2.0

The stack is the coordinated set of Hadoop components that you have installed. Use the following instructions to 1) upgrade a current, Ambari-installed and managed instance of a 1.3.2 or later version stack to a new, 2.0 version of the stack and 2) upgrade Ambari Server and Agents to 1.5.0 version. This procedure causes the upgraded stack to be managed by Ambari.

Note

If your stack has Kerberos Security turned on, you should turn it off before performing the upgrade. On Ambari Web->Admin view->Security-> click **Disable Security**. You can re-enable Security after performing the upgrade.

If you are upgrading from any other 1.x version of the stack, you must upgrade to 1.3.2 or later before you can upgrade to 2.0. See Upgrading the HDP Stack to 1.3.3 for more information. Upgrades from previous 2.x versions are not supported.

Note

If you have already upgraded to Ambari Server 1.5.0 and just want to upgrade the HDP stack, you can skip Section 2 and Section 3.

3.1. Preparing for the Upgrade

Use the following steps to prepare your system for the upgrade.

1. If you are upgrading Ambari as well as the stack, you must know the location of the Nagios servers for that process. Use the **Services**->**Nagios**-> **Summary** panel to locate the hosts on which they are running.

2. IF the Oozie service is installed in your cluster, list all current jobs.

```
oozie jobs -oozie http://localhost:11000/oozie -len 100 -filter status=
RUNNING
```

3. Stop all jobs in a RUNNING or SUSPENDED state on your Oozie server host. For example:

```
oozie job -oozie {your.oozie.server.host}:11000/oozie -kill {oozie.job.id}
```

4. Use the **Services** view on the **Ambari Web** UI to stop all services, including all clients, running on HDFS. Do **not** stop HDFS yet.

5. Finalize any prior upgrade, if you have not done so already.

```
su $HDFSUSER
hadoop namenode -finalize
```

6. Create the following logs and other files.

Because the upgrade to 2.0 includes a version upgrade of HDFS, creating these logs allows you to check the integrity of the file system post upgrade.

a. Run `fsck` with the following flags and send the results to a log. The resulting file contains a complete block map of the file system. You use this log later to confirm the upgrade.

```
su $HDFS_USER
hadoop fsck / -files -blocks -locations > /tmp/dfs-old-fsck-1.log
```

where *$HDFS_USER* is the HDFS Service user (by default, `hdfs`).

b. Capture the complete namespace of the filesystem. (The following command does a recursive listing of the root file system.)

```
su $HDFS_USER
hadoop dfs -lsr / > /tmp/dfs-old-lsr-1.log
```

where *$HDFS_USER* is the HDFS Service user (by default, `hdfs`).

c. Create a list of all the DataNodes in the cluster.

```
su $HDFS_USER
hadoop dfsadmin -report > /tmp/dfs-old-report-1.log
```

where *$HDFS_USER* is the HDFS Service user (by default, `hdfs`).

d. Optional: copy all or unrecoverable only data stored in HDFS to a local file system or to a backup instance of HDFS.

e. Optional: create the logs again and check to make sure the results are identical.

7. Save the namespace. You must be the HDFS service user to do this and you must put the cluster in Safe Mode.

Important

This is a **critical** step. If you do not do this step before you do the upgrade, the NameNode will **not** start afterwards.

```
su $HDFS_USER
hadoop dfsadmin -safemode enter
hadoop dfsadmin -saveNamespace
```

8. Copy the following checkpoint files into a backup directory. You can find the directory by using the **Services View** in the UI. Select the **HDFS** service, the **Configs** tab, in the Namenode section, look up the property **NameNode Directories**. It will be on your NameNode host.

- `dfs.name.dir/edits`
- `dfs.name.dir/image/fsimage`
- `dfs.name.dir/current/fsimage`

9. On the JobTracker host, copy `/etc/hadoop/conf` to a backup directory.

Note

If you have deployed a custom version of `capacity-scheduler.xml` and `mapred-queue-acls.xml`, after the upgrade you will need to use Ambari Web to edit the default Capacity Scheduler. Select Services view ->**YARN**->**Configs**->**Scheduler**->**Capacity Scheduler**.

Important

Fair Scheduler is not supported for use with HDP 2.x.

10. Store the layoutVersion for the NameNode. Make a copy of the file at *`$dfs.name.dir`*`/current/VERSION` where *`$dfs.name.dir`* is the value of the config parameter `NameNode directories`. This file will be used later to verify that the layout version is upgraded.

11. Stop HDFS. Make sure all services in the cluster are completely stopped.

12. If you are upgrading Hive, back up the Hive database.

13. Stop Ambari Server. On the Server host:

```
ambari-server stop
```

14. Stop Ambari Agents. On each host:

```
ambari-agent stop
```

3.2. Setting Up the Ambari Repository

This process prepares the updated repository.

1. Check to see if you have a `conf.save` directory for Ambari server and agents. If you do, move them to a back-up location:

```
mv /etc/ambari-server/conf.save/ /etc/ambari-server/conf.save.bak
```

```
mv /etc/ambari-agent/conf.save/ /etc/ambari-agent/conf.save.bak
```

2. Get the new Ambari bits. Use `wget` to fetch the repository file and replace the old repo file with the new repo file on every host.

Important

Check your current directory before you download the new repo file to make sure that there are no previous versions of the file. If you do not, and a previous version exists, the new download is saved with a numeric extension such as `ambari.repo.1`. Make sure that the version you copy is the new version.

- For RHEL/CentOS/Oracle Linux 5

```
wget http://public-repo-1.hortonworks.com/ambari/centos5/1.x/updates/1.5.
0/ambari.repo
cp ambari.repo /etc/yum.repos.d/ambari.repo
```

- For RHEL/CentOS/Oracle Linux 6

```
wget http://public-repo-1.hortonworks.com/ambari/centos6/1.x/updates/1.5.0/ambari.repo
cp ambari.repo /etc/yum.repos.d/ambari.repo
```

- For SLES 11

```
wget http://public-repo-1.hortonworks.com/ambari/suse11/1.x/updates/1.5.0/ambari.repo
cp ambari.repo /etc/zypp/repos.d/ambari.repo
```

Note

If your cluster does not have access to the Internet, you need to set up a local repository with this data before you continue. See Configure the Local Repositories for more information.

3.3. Upgrading to Ambari 1.5.0

This process upgrades Ambari Server, Ambari Agents, Ganglia, and Nagios.

Note

Ambari Server no longer automatically turns `iptables` off. Check your installation setup to make sure that you are not relying on this function. After you have upgraded the server you must either disable iptables manually or make sure that you have all the appropriate ports available.

1. Upgrade Ambari Server. From the Ambari Server host:

   - RHEL/CentOS/Oracle Linux

   ```
   yum clean all
   yum upgrade ambari-server ambari-log4j
   ```

   - SLES

   ```
   zypper clean
   zypper up ambari-server ambari-log4j
   ```

2. Check for upgrade success:

   - As the process runs, the console should produce output similar, although not identical, to this:

   ```
   Setting up Upgrade Process
   Resolving Dependencies
   --> Running transaction check
   ---> Package ambari-agent.x86_64 0:1.2.2.3-1 will be updated
   ---> Package ambari-agent.x86_64 0:1.2.2.4-1 will be updated ...
   ---> Package ambari-agent.x86_64 0:1.2.2.5-1 will be an update ...
   ```

   After the process is complete, check each host to make sure the new 1.5.0 files have been installed:

```
rpm -qa | grep ambari
```

- If the upgrade fails, the console should produce output similar to this:

```
Setting up Upgrade Process
No Packages marked for Update
```

3. Check to see if you have a folder named `/etc/ambari-server/conf.save`. If you do, rename it back:

```
mv /etc/ambari-server/conf.save /etc/ambari-server/conf
```

4. Upgrade the Ambari Server schema. From the Ambari Server host:

```
ambari-server upgrade
```

5. Upgrade the Ambari Agent on all hosts. From each Ambari Agent host:

- RHEL/CentOS/Oracle Linux

```
yum upgrade ambari-agent ambari-log4j
```

- SLES

```
zypper up ambari-agent ambari-log4j
```

Note

If you get a warning that begins "There are some running programs that use files deleted by recent upgrade," you can ignore it.

6. Check to see if you have a folder named `/etc/ambari-agent/conf.save` on each Agent host. If you do, rename it back. On each Agent host:

```
mv /etc/ambari-agent/conf.save /etc/ambari-agent/conf
```

7. Upgrade the Nagios addons:

- RHEL/CentOS/Oracle Linux

```
yum upgrade hdp_mon_nagios_addons
```

- SLES

```
zypper up hdp_mon_nagios_addons
```

3.4. Upgrading the Stack

This stack upgrade involves removing the HDP 1.x version of MapReduce and replacing it with the HDP 2.x YARN and MapReduce2 components. This process is somewhat long and complex. To help you, a Python script is provided to automate some of the upgrade steps.

3.4.1. Prepare for the Stack Upgrade

1. Make sure that you completed the system preparation procedure; most importantly, save the namespace.

2. Stage the upgrade script:

   a. Create an "Upgrade Folder", for example `/work/upgrade_hdp_2`, on a host that can communicate with Ambari Server. The Ambari Server host would be a suitable candidate.

   b. Copy the upgrade script to the Upgrade Folder. The script is available here: `/var/lib/ambari-server/resources/scripts/UpgradeHelper_HDP2.py` on the Ambari Server host.

   c. Make sure that Python is available on the host and that the version is 2.6 or higher:

   ```
   python --version
   ```

   

   Note

   For RHEL/Centos/Oracle Linux 5, you **must** use Python 2.6.

3. Start Ambari Server only. On the Ambari Server host:

   ```
   ambari-server start
   ```

4. Back up current configuration settings and the component host mappings from MapReduce:

   a. Go to the Upgrade Folder.

   b. Execute the `backup-configs` action:

   ```
   python UpgradeHelper_HDP2.py --hostname $HOSTNAME --user $USERNAME --
   password $PASSWORD --clustername $CLUSTERNAME backup-configs
   ```

   Where

   - *$HOSTNAME* is the name of the Ambari Server host
   - *$USERNAME* is the admin user for Ambari Server
   - *$PASSWORD* is the password for the admin user
   - *$CLUSTERNAME* is the name of the cluster

   This step produces a set of files named TYPE_TAG, where TYPE is the configuration type and TAG is the tag. These files contain copies of the various configuration settings for the current (pre-upgrade) cluster. You can use these files as a reference later.

   c. Execute the `save-mr-mapping` action:

   ```
   python UpgradeHelper_HDP2.py --hostname $HOSTNAME --user $USERNAME --
   password $PASSWORD --clustername $CLUSTERNAME save-mr-mapping
   ```

   This step produces a file named `mr_mapping` that stores the host level mapping of MapReduce components such as MapReduce JobTracker/TaskTracker/Client.

5. Delete all the MapReduce server components installed on the cluster.

a. If you are not already there, go to the Upgrade Folder.

b. Execute the `delete-mr` action.

```
python UpgradeHelper_HDP2.py --hostname $HOSTNAME --user $USERNAME --
password $PASSWORD --clustername $CLUSTERNAME delete-mr
```

Optionally, execute the delete script with the -n option to view, verify, and validate API calls, if necessary.

Note

Running the delete script with the -n option exposes API calls but does not remove installed components. Use the -n option for validation purposes only.

c. The script asks you to confirm that you have executed the `save-mr-mapping` action and that you have a file named `mr_mapping` in the Upgrade Folder.

3.4.2. Upgrade the Stack

1. Stop Ambari Server. On the Ambari Server host:

```
ambari-server stop
```

2. Update the stack version in the Server database, depending on if you are using a local repository:

Important

Make sure you delete the old MapReduce version **before** you run `upgradestack`.

```
ambari-server upgradestack HDP-2.0
```

3. Upgrade the HDP repository on all hosts and replace the old repo file with the new file:

Important

The file you download is named `hdp.repo`. To function properly in the system, it must be named HDP`.repo`. Once you have completed the "mv" of the new repo file to the repos.d folder, make sure there is no file named `hdp.repo` anywhere in your repos.d folder.

- For RHEL/CentOS/Oracle Linux 5

```
wget  http://public-repo-1.hortonworks.com/HDP/centos5/2.x/updates/2.0.10.
0/hdp.repo
mv hdp.repo /etc/yum.repos.d/HDP.repo
```

- For RHEL/CentOS/Oracle Linux 6

```
wget http://public-repo-1.hortonworks.com/HDP/centos6/2.x/updates/2.0.10.
0/hdp.repo
```

```
mv hdp.repo /etc/yum.repos.d/HDP.repo
```

- For SLES 11

```
wget  http://public-repo-1.hortonworks.com/HDP/suse11/2.x/updates/2.0.10.
0/hdp.repo
mv hdp.repo /etc/zypp/repos.d/HDP.repo
```

4. Back up the files in following directories on the Oozie server host and make sure that all files, including *site.xml files are copied.

```
mkdir oozie-conf-bak
cp -R /etc/oozie/conf/* oozie-conf-bak
```

5. Remove the old oozie directories on all Oozie server and client hosts

   - rm -rf /etc/oozie/conf
   - rm -rf /usr/lib/oozie/
   - rm -rf /var/lib/oozie/

6. Upgrade the stack on all Agent hosts.

Note

For each host, identify the HDP components installed on each host. Use Ambari Web, as described here, to view components on each host in your cluster. Based on the HDP components installed, tailor the following upgrade commands for each host to upgrade only components residing on that host. For example, if you know that a host has **no** HBase service or client packages installed, then you can adapt the command to **not** include HBase, as follows:

```
yum upgrade "collectd*" "gccxml*" "pig*" "hadoop*" "sqoop*"
 "zookeeper*" "hive*"
```

   - For RHEL/CentOS/Oracle Linux

     a. Remove remaining MapReduce components on all hosts:

     ```
     yum erase hadoop-pipes hadoop-sbin hadoop-native
     ```

     b. Upgrade the following components:

     ```
     yum upgrade "collectd*" "gccxml*" "pig*" "hadoop*" "sqoop*"
     "zookeeper*" "hbase*" "hive*" hdp_mon_nagios_addons

     yum install webhcat-tar-hive webhcat-tar-pig

     yum install hive*

     yum install oozie oozie-client

     rpm -e --nodeps bigtop-jsvc

     yum install bigtop-jsvc
     ```

     c. Check to see that the components in that list are upgraded.

```
yum list installed | grep HDP-$old-stack-version-number
```

None of the components from that list should appear in the returned list.

- For SLES

  a. Remove remaining MapReduce components on all hosts:

  ```
  zypper remove hadoop-pipes hadoop-sbin hadoop-native
  ```

  b. Upgrade the following components:

  ```
  zypper up "collectd*" "epel-release*" "gccxml*" "pig*" "hadoop*"
   "sqoop*" "zookeeper*" "hbase*" "hive*" "hcatalog*" "webhcat-tar*"
   "oozie*" hdp_mon_nagios_addons
  ```

  c. To verify that components were upgraded, execute:

  ```
  rpm -qa | grep hadoop, rpm -qa | grep hive and rpm -qa | grep hcatalog
  ```

  d. If components were not upgraded, upgrade them as follows:

  ```
  yast --update hadoop hcatalog hive
  ```

3.4.3. Add YARN/MR2 and Update Configurations

1. Start the Ambari Server. On the Server host:

   ```
   ambari-server start
   ```

2. Start each Ambari Agent. On all Agent hosts:

   ```
   ambari-agent start
   ```

3. After the Server and all Agents are running, log into Ambari Web. Do a hard refresh on your browser to make sure you are displaying the updated GUI. Make sure all hosts are healthy and all services are in a Stopped state.

4. Add YARN and MapReduce2 services:

   a. If you are not already there, go to the Upgrade Folder.

   b. Execute the `add-yarn-mr2` action:

   ```
   python UpgradeHelper_HDP2.py --hostname $HOSTNAME --user $USERNAME --
   password $PASSWORD --clustername $CLUSTERNAME add-yarn-mr2
   ```

   If desired, you can use the -n option to see the API calls as they are being made so that you can verify them.

5. Update the respective configurations:

   a. If you are not already there, go to the Upgrade Folder.

   b. Execute the `update-configs` action:

   ```
   python UpgradeHelper_HDP2.py --hostname $HOSTNAME --user $USERNAME --
   password $PASSWORD --clustername $CLUSTERNAME update-configs
   ```

6. Update individual configuration settings as needed. On the Ambari Server, use `/var/lib/ambari-server/resources/scripts/configs.sh` to inspect and update the configuration properties.

Note

configs.sh creates temporary files. We recommended that you run configs.sh as root or as a user having write permission on the local folder.

a. Get configuration details:

```
configs.sh get $HOSTNAME $CLUSTERNAME $CONFIGURATION-TYPE

For example:
configs.sh get localhost myclustername global
```

b. Evaluate each property value returned and modify as needed:

```
configs.sh set $HOSTNAME $CLUSTERNAME $CONFIGURATION-TYPE "property name"
 "new value"

For example:
configs.sh set localhost myclustername global yarn_log_dir_prefix “/apps/
logs/yarn”
```

c. Remove properties that are not needed:

```
configs.sh delete $HOSTNAME $CLUSTERNAME $CONFIGURATION-TYPE "property
 name"

For example:
configs.sh delete localhost myclustername global dfs.client-write-packet-
size
```

Table 3.1. Key Properties to Check

Configuration Type	Property	Description
global	yarn_log_dir_prefix	The location for the YARN logs
global	yarn_pid_dir_prefix	The location for the YARN pid files
global	yarn_user	The YARN user
-	-	-
yarn-site	yarn.nodemanager.local-dirs	The location for container logs
yarn-site	yarn.nodemanager.log-dirs	The directories for localized files

Note

Make any necessary modifications **before** starting the services.

d. Install the YARN and MapReduce2 services:

i. If you are not already there, go to the Upgrade Folder.

ii. Execute the `install-yarn-mr2` action:

```
python UpgradeHelper_HDP2.py --hostname $HOSTNAME --user $USERNAME --
password $PASSWORD --clustername $CLUSTERNAME install-yarn-mr2
```

Note

This is a two step process. You can use the Ambari Web GUI to monitor the progress. Both steps must be **complete** before you continue to the next step.

3.4.4. Complete the Stack Upgrade

1. Because the file system version has now changed you must start the NameNode manually. On the NameNode host:

   ```
   su -l $HDFS_USER -c "export HADOOP_LIBEXEC_DIR=/usr/lib/hadoop/libexec && /
   usr/lib/hadoop/sbin/hadoop-daemon.sh start namenode -upgrade"
   ```

   To check if the Upgrade is in progress, check that the "`\previous`" directory has been created in \NameNode and \JournalNode directories. The "`\previous`" directory contains a snapshot of the data before upgrade.

2. Prepare the NameNode to work with Ambari:

   a. Open the Ambari Web GUI. If it has been open throughout the process, do a hard reset on your browser to force a reload.

   b. On the Services view, click **HDFS** to open the HDFS service.

   c. Click **View Host** to open the NameNode host details page.

   d. Use the dropdown menu to stop the NameNode.

   e. On the Services view, restart the HDFS service. Make sure it passes the ServiceCheck. It is now under Ambari's control.

3. After the DataNodes are started, HDFS exits safemode. To monitor the status, run the following command:

   ```
   sudo su -l $HDFS_USER -c "hdfs dfsadmin -safemode get”
   ```

   Depending on the size of your system, a response may not display for up to 10 minutes. When HDFS exits safemode, the following message displays:

   ```
   Safe mode is OFF
   ```

4. Make sure that the HDFS upgrade was successful. Go through steps 2 and 3 in Preparing for the Upgrade to create new versions of the logs and reports. Substitute "`new`" for "`old`" in the file names as necessary.

5. Compare the old and new versions of the following:

   - `dfs-old-fsck-1.log` versus `dfs-new-fsck-1.log`.

     The files should be identical unless the `hadoop fsck` reporting format has changed in the new version.

- `dfs-old-lsr-1.log` versus `dfs-new-lsr-1.log`.

  The files should be identical unless the the format of `hadoop fs -lsr` reporting or the data structures have changed in the new version.

- `dfs-old-report-1.log` versus `fs-new-report-1.log`

  Make sure all DataNodes previously belonging to the cluster are up and running.

6. Use the Ambari Web Services view to start YARN.
7. Use the Ambari Web Services view to start MapReduce2.
8. Upgrade HBase:
   a. Make sure that all HBase components - RegionServers and HBase Master - are stopped.
   b. Use the Ambari Web Services view, start the ZooKeeper service. Wait until the ZK service is up and running.
   c. On the HBase Master host, make these configuration changes:
      i. In `HBASE_CONFDIR/hbase-site.xml`, set the property `dfs.client.read.shortcircuit` to `false`.
      ii. In the configuration file, find the value of the `hbase.tmp.dir` property and make sure that the directory exists and is readable and writeable for the HBase service user and group.

```
chown -R $HBASE_USER:$HADOOP_GROUP $HBASE.TMP.DIR
```

      iii. Go to the Upgrade Folder and check in the saved global configuration file named `global_<$TAG>` for the value of the property `hbase_pid_dir` and `hbase_log_dir`. Make sure that the directories are readable and writeable for the HBase service user and group.

```
chown -R $HBASE_USER:$HADOOP_GROUP $hbase_pid_dir
chown -R $HBASE_USER:$HADOOP_GROUP $hbase_log_dir
```

      Do this on **every** host where a RegionServer is installed as well as on the HBase Master host.

      iv. Check for HFiles in V1 format. HBase 0.96.0 discontinues support for HFileV1. Before the actual upgrade, run the following command to check if there are HFiles in V1 format:

```
hbase upgrade -check
```

      HFileV1 was a common format prior to HBase 0.94. You may see output similar to:

```
Tables Processed:

hdfs://localhost:41020/myHBase/.META.
hdfs://localhost:41020/myHBase/usertable
hdfs://localhost:41020/myHBase/TestTable
```

```
hdfs://localhost:41020/myHBase/t

Count of HFileV1: 2
HFileV1:
hdfs://localhost:41020/myHBase/usertable/
fa02dac1f38d03577bd0f7e666f12812/family/249450144068442524
hdfs://localhost:41020/myHBase/usertable/
ecdd3eaee2d2fcf8184ac025555bb2af/family/249450144068442512

Count of corrupted files: 1
Corrupted Files:
hdfs://localhost:41020/myHBase/usertable/
fa02dac1f38d03577bd0f7e666f12812/family/1
Count of Regions with HFileV1: 2
Regions to Major Compact:
hdfs://localhost:41020/myHBase/usertable/
fa02dac1f38d03577bd0f7e666f12812
hdfs://localhost:41020/myHBase/usertable/
ecdd3eaee2d2fcf8184ac025555bb2af
```

When you run the upgrade check, if "Count of HFileV1" returns any files, start the hbase shell to use major compaction for regions that have HFileV1 format. For example in the sample output above, you must compact the `fa02dac1f38d03577bd0f7e666f12812` and `ecdd3eaee2d2fcf8184ac025555bb2af` regions.

v. Upgrade HBase. You must be the HBase service user.

```
sudo su -l $HBASE_USER -c "hbase upgrade -execute"
```

Make sure that the output contains the string "Successfully completed Znode upgrade".

vi. Use the Services view to start the HBase service. Make sure that Service Check passes.

9. Upgrade Oozie:

a. On the Services view, make sure YARN and MapReduce2 are running.

b. Make sure that the Oozie service is stopped.

c. Upgrade Oozie. You must be the Oozie service user. On the Oozie host:

```
sudo su -l $OOZIE_USER -c"/usr/lib/oozie/bin/ooziedb.sh upgrade -run"
```

Make sure that the output contains the string "Oozie DB has been upgrade to Oozie version *'OOZIE Build Version*"'.

d. Prepare the WAR file:

Note

The Oozie server must be **not** running for this step. If you get the message "ERROR: Stop Oozie first", it means the script still thinks it's running. Check, and if needed, remove the process id (pid) file indicated in the output.

```
/usr/lib/oozie/bin/oozie-setup.sh prepare-war
```

Make sure that the output contains the string "New Oozie WAR file with added".

e. Modify the following configuration properties in `oozie-site.xml`. On the Ambari Server, use `/var/lib/ambari-server/resources/scripts/configs.sh` to inspect and update the configuration properties as described here [25].

Table 3.2. Properties to Modify

Action	Property Name	Property Value
Add	oozie.service.URIHandlerService.uri.handlers	`org.apache.oozie.dependenc`
Add	oozie.service.coord.push.check.requeue.interval	`30000`
Add	oozie.services.ext	`org.apache.oozie.service.P HCatAccessorService`
Add/Modify	oozie.service.SchemaService.wf.ext.schemas	`shell-action-0.1.xsd,email ssh-action-0.1.xsd,distcp- sla-0.2.xsd` [a]

[a] Use this list if you have not modified the default Ambari values. If you have added custom schemas, make sure they exist after the modification. The schemas being added here are shell-action-0.2.xsd, oozie-sla-0.1.xsd, and oozie-sla-0.2.xsd. You can add these to your existing list.

f. Replace the content of `/user/oozie/share` in HDFS. On the Oozie server host:

i. Extract the Oozie sharelib into a `tmp` folder.

```
mkdir -p /tmp/oozie_tmp
cp /usr/lib/oozie/oozie-sharelib.tar.gz /tmp/oozie_tmp
cd /tmp/oozie_tmp
tar xzvf oozie-sharelib.tar.gz
```

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

```
su -l $HDFS_USR -c "$hdfs dfs -copyToLocal /user/oozie/share /tmp/
oozie_tmp/oozie_share_backup"
su -l $HDFS_USR -c "$hdfs dfs -rm -r /user/oozie/share"
```

iii. Add the latest share libs that you extracted in step 1. After you have added the files, modify ownership and acl.

```
su -l $HDFS_USR -c "hdfs dfs -copyFromLocal /tmp/oozie_tmp/share /user/
oozie/."
su -l $HDFS_USR -c "hdfs dfs -chown -R oozie:hadoop /user/oozie"
su -l $HDFS_USR -c "hdfs dfs -chmod -R 755 /user/oozie"
```

g. Use the Services view to start the Oozie service. Make sure that ServiceCheck passes for Oozie.

10. Update WebHcat.

a. Modify the `webhcat-site` config type.

On the Ambari server, use `/var/lib/ambari-server/resources/scripts/configs.sh` to modify configuration properties in templeton.storage.class:

```
configs.sh set $HOSTNAME $CLUSTERNAME $CONFIGURATION-TYPE $PROPERTY-NAME
 $PROPERTY-VALUE
For example: configs.sh set <yourhostname> <yourclustername> webhcat-
site "templeton.storage.class" "org.apache.hive.hcatalog.templeton.tool.
ZooKeeperStorage"
```

b. Update the Pig and Hive tar bundles, by updating the following files:

- /apps/webhcat/pig.tar.gz
- /apps/webhcat/hive.tar.gz

Note

You will find these files on a host where webhcat is installed.

For example, to update a *.tar.gz file:

i. Move the file to a local directory.

```
su -l $HCAT_USR -c "hadoop --config /etc/hadoop/conf fs -copyToLocal /
apps/webhcat/*.tar.gz ${local_backup_dir}"
```

ii. Remove the old file.

```
su -l $HCAT_USR -c "hadoop --config /etc/hadoop/conf fs -rm /apps/
webhcat/*.tar.gz"
```

iii. Copy the new file.

```
su -l $HCAT_USR -c "hadoop --config /etc/hadoop/conf fs -
copyFromLocal /usr/share/HDP-webhcat/*.tar.gz /apps/webhcat"
```

c. Update /app/webhcat/hadoop-streaming.jar file.

i. Move the file to a local directory.

```
su -l $HCAT_USR -c "hadoop --config /etc/hadoop/conf fs -copyToLocal /
apps/webhcat/hadoop-streaming*.jar ${local_backup_dir}"
```

ii. Remove the old file.

```
su -l $HCAT_USR -c "hadoop --config /etc/hadoop/conf fs -rm /apps/
webhcat/hadoop-streaming*.jar"
```

iii. Copy the new hadoop-streaming.jar file.

```
su -l $HCAT_USR -c "hadoop --config /etc/hadoop/conf fs -
copyFromLocal /user/lib/hadoop-mapreduce/hadoop-streaming*.jar /apps/
webhcat"
```

11. Make sure Ganglia no longer attempts to monitor JobTracker.

a. Make sure Ganglia is stopped.

b. Log into the host where JobTracker was installed (and where ResourceManager is installed after the upgrade).

c. Backup the folder `/etc/ganglia/hdp/HDPJobTracker` .

d. Remove the folder `/etc/ganglia/hdp/HDPJobTracker`.

e. Remove the folder `$ganglia_runtime_dir/HDPJobTracker`.

Note

For the value of `$ganglia_runtime_dir`, in the Upgrade Folder, check the saved global configuration file `global_<$TAG>`.

12. Use the Services view to start the remaining services back up.

13. The upgrade is now fully functional but not yet finalized. Using the `finalize` command removes the previous version of the NameNode and DataNode storage directories.

Important

After the upgrade is finalized, the system cannot be rolled back. Usually this step is not taken until a thorough testing of the upgrade has been performed.

The upgrade must be finalized before another upgrade can be performed.

Note

Directories used by Hadoop 1 services set in /etc/hadoop/conf/taskcontroller.cfg are not automatically deleted after upgrade. Administrators can choose to delete these directories after the upgrade.

To finalize the upgrade:

```
sudo su -l $HDFS_USER -c "hadoop dfsadmin -finalizeUpgrade"
```

where *$HDFS_USER* is the HDFS Service user (by default, `hdfs`).

4. Upgrading Operating Systems on an Ambari-based Hadoop Installation

Ambari requires specific versions of the files for components that it uses. There are three steps you should take to make sure that these versions continue to be available:

- Disable automatic OS updates
- Do not update any HDP components such as MySQL, Ganglia, etc.
- If you must perform an OS update, do a manual kernel update only.

5. Upgrading From Older Ambari Server versions to 1.2.5

This process upgrades Ambari Server from older, 1.x versions to version 1.2.5. It does not change the underlying Hadoop Stack. This is a twelve step manual process.

Note

You must know the location of the Nagios server for Step 9. Use the **Services View**-> **Summary** panel to locate the host on which it is running.

1. Stop the Ambari Server and all Ambari Agents. From the Ambari Server host:

   ```
   ambari-server stop
   ```

   From each Ambari Agent host:

   ```
   ambari-agent stop
   ```

2. Get the new Ambari bits. Use `wget` to fetch the repository file and replace the old repo file with the new repo file on every host.

   - Fetch the new repo file:

     For RHEL/CentOS 5/Oracle Linux 5

     ```
     wget http://public-repo-1.hortonworks.com/ambari/centos5/1.x/updates/1.2.5.17/ambari.repo
     ```

     For RHEL/CentOS 6/Oracle Linux 6

     ```
     wget http://public-repo-1.hortonworks.com/ambari/centos6/1.x/updates/1.2.5.17/ambari.repo
     ```

     For SLES 11

     ```
     wget http://public-repo-1.hortonworks.com/ambari/suse11/1.x/updates/1.2.5.17/ambari.repo
     ```

     

     ### Important

     Check your current directory before you download the new repo file to make sure that there are no previous versions of the file. If you do not, and a previous version exists, the new download will be saved with a numeric extension such as `ambari.repo.1`. Make sure that the version you copy is the new version.

   - Replace the old repo file with the new repo file.

     For RHEL/CentOS 5/Oracle Linux 5

     ```
     cp ambari.repo /etc/yum.repos.d/ambari.repo
     ```

     For RHEL/CentOS 6/Oracle Linux 6

```
cp ambari.repo /etc/yum.repos.d/ambari.repo
```

For SLES 11

```
cp ambari.repo /etc/zypp/repos.d/ambari.repo
```

Note

If your cluster does not have access to the Internet, set up a local repository with this data before you continue. See Configure the Local Repositories for more information.

3. Upgrade Ambari Server. From the Ambari Server host:

   - RHEL/CentOS/Oracle Linux

   ```
   yum clean all
   yum upgrade ambari-server-1.2.5.17 ambari-log4j-1.2.5.17
   ```

   - SLES

   ```
   zypper clean
   zypper up ambari-server-1.2.5.17 ambari-log4j-1.2.5.17
   ```

4. Check for upgrade success:

   - As the process runs, the console should produce output similar, although not identical, to this:

   ```
   Setting up Upgrade Process
   Resolving Dependencies
   --> Running transaction check
   ---> Package ambari-agent.x86_64 0:1.2.2.3-1 will be updated
   ---> Package ambari-agent.x86_64 0:1.2.2.4-1 will be updated ...
   ---> Package ambari-agent.x86_64 0:1.2.2.5-1 will be an update ...
   ```

   After the process is complete, check each host to make sure the new 1.2.4 files have been installed:

   ```
   rpm -qa | grep ambari
   ```

   - If the upgrade fails, the console should produce output similar to this:

   ```
   Setting up Upgrade Process
   No Packages marked for Update
   ```

5. Check to see if you have a folder named `/etc/ambari-server/conf.save`. If you do, rename it back:

   ```
   mv /etc/ambari-server/conf.save /etc/ambari-server/conf
   ```

6. Upgrade the Ambari Server schema. From the Ambari Server host:

   ```
   ambari-server upgrade
   ```

7. Upgrade the Ambari Agent on all hosts. From each Ambari Agent host:

   - RHEL/CentOS/Oracle Linux

```
yum upgrade ambari-agent ambari-log4j
```

- SLES

```
zypper up ambari-agent ambari-log4j
```

Note

If you get a warning that begins "There are some running programs that use files deleted by recent upgrade" you can ignore it.

8. Check to see if you have a file named `/etc/ambari-agent/conf.save` on each Agent host. If you do, rename it back. On each Agent host:

```
mv /etc/ambari-agent/conf.save /etc/ambari-agent/conf
```

9. Upgrade the Nagios and Ganglia addons package and restart. On the Nagios/Ganglia host:

   - RHEL/CentOS/Oracle Linux

```
yum upgrade hdp_mon_nagios_addons hdp_mon_ganglia_addons
service httpd restart
```

   - SLES

```
zypper up hdp_mon_nagios_addons hdp_mon_ganglia_addons
service apache2 restart
```

10. Start the Server and the Agents on all hosts. From the Server host:

```
ambari-server start
```

From each Agent host:

```
ambari-agent start
```

11. Open **Ambari Web**. Point your browser to `http://{your.ambari.server}:8080`

Important

You need to refresh your browser so that it loads the new version of the code. Hold the Shift key down while clicking the refresh button on the browser. If you have problems, clear your browser cache manually and restart Ambari Server.

Use the Admin name and password you have set up to log in.

12. Re-start the Ganglia, Nagios, and MapReduce services. In **Ambari Web**.

    a. Go to the **Services View** and select each service.

    b. Use the **Management Header** to stop and re-start each service.