Hortonworks Data Platform: Ambari Upgrade Guide
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.

Except where otherwise noted, this document is licensed under Creative Commons Attribution ShareAlike 4.0 License. http://creativecommons.org/licenses/by-sa/4.0/legalcode
# Table of Contents

1. Upgrading Ambari and HDP .................................................................................................................. 1
2. Getting Ready to Upgrade Ambari and HDP .......................................................................................... 2
3. Upgrading Ambari .................................................................................................................................. 4
   3.1. Preparing to Upgrade ......................................................................................................................... 4
   3.2. Upgrade Ambari .................................................................................................................................... 5
   3.3. Post-Upgrade Tasks ............................................................................................................................ 10
      3.3.1. Migrate to Ambari Alerts and Metrics ......................................................................................... 10
      3.3.2. Adjusting for Kerberos ................................................................................................................. 13
      3.3.3. Upgrade Ambari Metrics .............................................................................................................. 14
4. Upgrading HDP ....................................................................................................................................... 15
   4.1. Prerequisites ....................................................................................................................................... 17
      4.1.1. Rolling Upgrade Prerequisites ........................................................................................................ 17
   4.2. Prepare to Upgrade .............................................................................................................................. 19
   4.3. Register and Install Target Version .................................................................................................... 20
      4.3.1. Register Target Version ................................................................................................................ 21
      4.3.2. Install Target Version .................................................................................................................. 22
   4.4. Perform the Upgrade ............................................................................................................................ 22
      4.4.1. Perform Rolling Upgrade ............................................................................................................... 23
      4.4.2. Perform Express Upgrade ............................................................................................................ 25
List of Tables

4.1. HDP Upgrade Options ......................................................... 16
1. Upgrading Ambari and HDP

Ambari and the HDP Stack being managed by Ambari can be upgraded independently.

This guide provides information on:

- Getting Ready to Upgrade Ambari and HDP [2]
- Upgrading Ambari [4]
- Upgrading HDP [15]

**Important**

Ambari 2.2 does not support managing an HDP 2.0 cluster. If you are running HDP 2.0, in order to use Ambari 2.2 you must first upgrade to HDP 2.1 or higher using either Ambari 1.7 or 2.0 prior to upgrading to Ambari 2.2. Once completed, upgrade your current Ambari to Ambari 2.2.
2. Getting Ready to Upgrade Ambari and HDP

When preparing to upgrade Ambari and the HDP Cluster, we strongly recommend you review this checklist of items to confirm your cluster operation is healthy. Attempting to upgrade a cluster that is operating in an unhealthy state can produce unexpected results.

**Important**

If planning to upgrade Ambari and upgrade to a new HDP minor version (for example: moving from HDP 2.2 to HDP 2.3), upgrade Ambari to the latest version before upgrading the cluster. Refer to the Stack Compatibility Matrix to see which HDP versions each Ambari version supports.

- Ensure all services in the cluster are running.
- Run each Service Check (found under the Service Actions menu) and confirm they execute successfully.
- Clear all alerts, or understand why they are being generated. Remediate as necessary.
- Confirm start and stop for all services are executing successfully.
- Time service start and stops. The time to start and stop services is a big contributor to overall upgrade time so having this information handy is useful.
- Download the software packages prior to the upgrade. Place them in a local repository and/or consider using a storage proxy since multi-gigabyte downloads will be required on all nodes in the cluster. Refer to Using a Local Repository for more information.
- Ensure point-in-time backups are taken of all DBs supporting the clusters. This includes (among others) Ambari, Hive Metastore, Ranger and Oozie.

**Note**

If you are upgrading to Ambari 2.2, you must make sure that the Ranger db root password is NOT set to blank before performing any Stack upgrade. Ambari 2.1x requires that the Ranger db root password has a value. If you upgrade Ambari to 2.2 and upgrade HDP without setting a value for the Ranger db root password, Ranger Admin will fail to start after the upgrade.

To prepare Ranger for upgrades, set the password for the Ranger DB root user to a non-blank value. Then, set the Ranger db root password field in Ambari Web to match this value. Finally, restart Ranger Admin using Ambari Web.

You should also review the data in the xa_access_audit table in the Ranger Audit Database. Truncate historical data to reduce upgrading time, especially if you have a lot of data in xa_access_audit table.

For Ambari Upgrades
• This (Ambari 2.2) Upgrade Guide will help you upgrade your existing Ambari install to version 2.2. If you are upgrading to another Ambari version, please be sure to use the Ambari Upgrade Guide for that version.

• Be sure to review the Known Issues and Behavioral Changes for this Ambari release in the Ambari 2.2.0 Release Notes.

For HDP Cluster Upgrades

• Ensure sufficient disk space on /usr/hdp/<version> (roughly 3GB for each additional HDP release).

• If you plan to add new services available with HDP to your cluster, the new services might include new service accounts. Any operational procedures required to support these new service accounts should be performed prior to the upgrade. The accounts will typically be required on all nodes in the cluster.

• If your cluster includes Storm, document any running Storm topologies.
3. Upgrading Ambari

Ambari and the HDP cluster being managed by Ambari can be upgraded independently. This section describes the process to upgrade Ambari. You are strongly encouraged to read completely through this entire document before starting the upgrade process, to that you understand the interdependencies and order of the steps. It is highly recommended you validate these steps in a test environment to adjust + account for any special configurations for your cluster.

- Preparing to Upgrade [4]
- Upgrade Ambari [5]
- Post-Upgrade Tasks [10]

The high-level process for upgrading Ambari is as follows:

### 3.1. Preparing to Upgrade

- Be sure to review the Release Notes for this Ambari release for Known Issues and Behavioral Changes.

- You must have root, administrative, or root-equivalent authorization on the Ambari server host and all servers in the cluster.

- You must backup the Ambari Server database.

- You must make a safe copy of the Ambari Server configuration file found at `/etc/ambari-server/conf/ambari.properties`.

- If you are using Ambari with JDK 1.6, you must first move to JDK 1.7 before attempting to upgrade to Ambari 2.2. Refer the the Ambari Reference Guide for the Ambari version you are currently running for instructions on Changing the JDK.

- If you are managing Ganglia and Nagios from Ambari:
  - Support for managing Ganglia and Nagios from Ambari has been removed.
  - Record the location of the Nagios server before you begin the upgrade process.
  - Record the location of the Ganglia server before you begin the upgrade process.
  - You must stop the Nagios and Ganglia services from Ambari Web.
• After upgrading Ambari, you must remove Nagios and Ganglia from your cluster and replace with Ambari Alerts and Metrics. For more information, see Migrate to Ambari Alerts and Metrics in Ambari.

• If you are on Ambari 1.7 or earlier:
  • If your cluster has Kerberos enabled, you must disable Kerberos prior to performing the upgrade. Be sure to review Adjusting for Kerberos for more information on the additional steps required pre- and post-upgrade.

• If you are on Ambari 2.0 or later:
  • Record the location of the Metrics Collector component before you begin the upgrade process.
  • You must stop the Ambari Metrics service from Ambari Web.
  • After upgrading Ambari, you must also Upgrade Ambari Metrics service.
  • Proceed to Upgrade to Ambari 2.2.
  • After upgrading Ambari, you must also Upgrade Hortonworks SmartSense.

Note
If your current Ambari version is 1.6.1 or below, you must upgrade the Ambari Server version to 1.7 before upgrading to version 2.2.

Note
During Ambari upgrade, the existing /var/lib/ambari-server/ambari-env.sh file is overwritten and a backup copy of ambari-env.sh (with extension .rpmsave) is created. If you have manually modified ambari-env.sh (for example, to change Ambari Server heap), you will need to re-apply your changes to the new file.

3.2. Upgrade Ambari

1. If you are running Ambari Metrics service in your cluster, stop the service. From Ambari Web, browse to Services > Ambari Metrics and select Stop from the Service Actions menu.

2. Stop the Ambari Server. On the host running Ambari Server:
   
   ambari-server stop

3. Stop all Ambari Agents. On each host in your cluster running an Ambari Agent:
   
   ambari-agent stop

4. Fetch the new Ambari repo and replace the old repository file with the new repository file on all hosts in your cluster.
Important

Check your current directory before you download the new repository file to make sure that there are no previous versions of the `ambari.repo` 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.

Select the repository appropriate for your environment from the following list:

- **For RHEL/CentOS/Oracle Linux 6:**
  
  ```bash
  updates/2.2.0.0/ambari.repo -O /etc/yum.repos.d/ambari.repo
  ```

- **For RHEL/CentOS/Oracle Linux 7:**
  
  ```bash
  wget -nv http://dev.hortonworks.com.s3.amazonaws.com/ambari/centos7/2.x/
  updates/2.2.0.0/ambari.repo -O /etc/yum.repos.d/ambari.repo
  ```

- **For SLES 11:**
  
  ```bash
  updates/2.2.0.0/ambari.repo -O /etc/zypp/repos.d/ambari.repo
  ```

- **For Ubuntu 12:**
  
  ```bash
  wget -nv http://dev.hortonworks.com.s3.amazonaws.com/ambari/ubuntu12/2.x/
  updates/2.2.0.0/ambari.list -O /etc/apt/sources.list.d/ambari.list
  ```

- **For Ubuntu 14:**
  
  ```bash
  wget -nv http://dev.hortonworks.com.s3.amazonaws.com/ambari/ubuntu14/2.x/
  updates/2.2.0.0/ambari.list -O /etc/apt/sources.list.d/ambari.list
  ```

- **For Debian 7:**
  
  ```bash
  wget -nv http://dev.hortonworks.com.s3.amazonaws.com/ambari/debian7/2.x/
  updates/2.2.0.0/ambari.list -O /etc/apt/sources.list.d/ambari.list
  ```

**Note**

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

**Note**

Ambari Server does not automatically turn off `iptables`. Check that your installation setup does not depend on `iptables` being disabled. After upgrading the server, you must either disable `iptables` manually or make sure that you have appropriate ports available on all cluster hosts. For more information about ports, see Configuring Network Port Numbers.

5. Upgrade Ambari Server. On the host running Ambari Server:
- For RHEL/CentOS/Oracle Linux:

```bash
yum clean all
yum info ambari-server
```

In the info output, visually validate that there is an available version containing "2.2.0"

```bash
yum upgrade ambari-server
```

- For SLES:

```bash
zypper clean
zypper info ambari-server
```

In the info output, visually validate that there is an available version containing "2.2.0"

```bash
zypper up ambari-server
```

- For Ubuntu/Debian:

```bash
apt-get clean all
apt-get update
apt-cache show ambari-server | grep Version
```

In the info output, visually validate that there is an available version containing "2.2.0"

```bash
apt-get install ambari-server
```

**Important**

When performing upgrade on SLES, you will see a message "There is an update candidate for 'ambari-server', but it is from different vendor. Use 'zypper install ambari-server-2.2.0-101.noarch' to install this candidate". You will need to use yast to update the package, as follows:

a. From the command line run: > yast.

```bash
> yast
```

You will see command line UI for YaST program.

b. Choose Software > Software Management, then click the Enter button.

c. In the Search Phrase field, enter ambari-server, then click the Enter button.

d. On the right side you will see the search result `ambari-server 2.2.0`. Click Actions, choose Update, then click the Enter button.

e. Go to Accept, and click enter.

6. Check for upgrade success by noting progress during the Ambari Server installation process you started in Step 5.
• As the process runs, the console displays output similar, although not identical, to the following:

| Setting up Upgrade Process                      |
| Resolving Dependencies                          |
| --> Running transaction check                   |

• If the upgrade fails, the console displays output similar to the following:

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

• A successful upgrade displays output similar to the following:

| Updated:                                        |
| ambari-server.noarch 0:2.2.0-111               |
| Complete!                                       |

**Note**

Confirm there is only one ambari-server*.jar file in /usr/lib/ambari-server. If there is more than one JAR file with name ambari-server*.jar, move all JARs except ambari-server-2.2.0.*.jar to /tmp before proceeding with upgrade.

7. Upgrade all Ambari Agents. On each host in your cluster running an Ambari Agent:

• For RHEL/CentOS/Oracle Linux:

  ```
  yum upgrade ambari-agent
  ```

• For SLES:

  ```
  zypper up ambari-agent
  ```

**Note**

Ignore the warning that begins with "There are some running programs that use files deleted by recent upgrade".

**Important**

When performing upgrade on SLES, you will see a message "There is an update candidate for 'ambari-agent', but it is from different vendor. Use 'zypper install ambari-agent-2.2.0-101.noarch' to install this candidate". You will need to use yast to update the package, as follows:

a. From the command line run: > yast

  ```
  > yast
  ```

  You will see command line UI for YaST program.

b. Choose Software > Software Management, then click the Enter button.
c. In the Search Phrase field, enter ambari-agent, then click the Enter button.

d. On the right side you will see the search result ambari-agent 2.2.0. Click Actions, choose Update, then click the Enter button.

e. Go to Accept, and click enter.

- **For Ubuntu/Debian:**

  ```
  apt-get update
  apt-get install ambari-agent
  ```

8. After the upgrade process completes, check each host to make sure the new files have been installed:

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

9. Upgrade Ambari Server database schema. On the host running Ambari Server:

  ```
  ambari-server upgrade
  ```

10. Start the Ambari Server. On the host running Ambari Server:

    ```
    ambari-server start
    ```

11. Start all Ambari Agents. On each host in your cluster running an Ambari Agent:

    ```
    ambari-agent start
    ```


    Point your browser to http://<your.ambari.server>:8080

    where <your.ambari.server> is the name of your ambari server host. For example, c6401.ambari.apache.org.

    **Important**

    Refresh your browser so that it loads the new version of the Ambari Web code. If you have problems, clear your browser cache manually, then restart Ambari Server.

13. Log in, using the Ambari administrator credentials that you have set up.

    For example, the default name/password is admin/admin.

14. You will see a Restart indicator next to each service after upgrading. Ambari upgrade has added to/adjusted the configuration properties of your cluster based on new configuration types and properties being made available for each service with this release of Ambari. Review these changes by comparing the previous configuration with the latest version created by "ambari-upgrade".

15. If you have configured Ambari to authenticate against an external LDAP or Active Directory, you **must** re-run "ambari-server setup-ldap". For more information, see Set Up LDAP or Active Directory Authentication.
16. If you have configured your cluster for Hive or Oozie with an external database (Oracle, MySQL or PostgreSQL), you must re-run “ambari-server setup –jdbc-db and –jdbc-driver” to get the JDBC driver JAR file in place. For more information, see Using Non-Default Databases - Hive and Using Non-Default Databases - Oozie.

17. If your cluster includes Ganglia and Nagios:

You must remove Nagios and Ganglia from your cluster and replace with Ambari Alerts and Metrics. For more information, see Migrate to Ambari Alerts and Metrics.

18. If you have upgraded from Ambari 1.7 or earlier:

- If your cluster was configured for Kerberos, you must adjust your cluster for Kerberos. For more information, see Adjusting for Kerberos.

- If your cluster is running HDP 2.2 Stack, you must get the cluster hosts to advertise the "current version". This is done by restarting a master or slave component (such as a DataNode) on each host to have the host. For example, in Ambari Web, navigate to the Hosts page, select any Host that has the DataNode component and restart that DataNode component on that single host.

19. If you are running Ambari Metrics service in your cluster, Upgrade Ambari Metrics service.

20. If your cluster includes the SmartSense service, you must upgrade it along with Ambari, Upgrade SmartSense Service.

3.3. Post-Upgrade Tasks

Depending on the configuration of your cluster and the Ambari version you are starting with, review the following post-upgrade tasks.

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Migrate to Ambari Alerts and Metrics [10]</td>
<td>If your cluster includes legacy Nagios and Ganglia services, they must be removed and replaced with Ambari Alerts and Metrics.</td>
</tr>
<tr>
<td>Adjusting for Kerberos [13]</td>
<td>If you have upgraded from Ambari 1.7 or earlier and your cluster was configured for Kerberos, you must adjust Ambari.</td>
</tr>
<tr>
<td>Upgrade Ambari Metrics [14]</td>
<td>If your cluster includes the Ambari Metrics System (&quot;AMS&quot;) service, you must upgrade the system along with Ambari.</td>
</tr>
<tr>
<td>Upgrading SmartSense</td>
<td>If your cluster includes the SmartSense service, you must upgrade it along with Ambari.</td>
</tr>
</tbody>
</table>

3.3.1. Migrate to Ambari Alerts and Metrics

Starting with Ambari 2.0, Ambari includes built-in systems for alerting and metrics collection. Therefore, when upgrading to Ambari 2.2, the legacy Nagios and Ganglia services must be removed and replaced with the new systems. You only need to perform these steps and migrate to Ambari Alerts and Metrics if you are running Nagios and Ganglia.

Important

We highly recommended that you perform and validate this procedure in a test environment prior to attempting the Ambari upgrade on production systems.
3.3.1.1. Moving from Nagios to Ambari Alerts

After upgrading to Ambari 2.2, the Nagios service will be removed from the cluster. The Nagios server and packages will remain on the existing installed host but Nagios itself is removed from Ambari management.

**Important**

Nagios used the operating system sendmail utility to dispatch email alerts on changes. With Ambari Alerts, the email dispatch is handled from the Ambari Server via Javamail. Therefore, you must provide SMTP information to Ambari for sending email alerts. Have this information ready. You will use it after the Ambari 2.2 upgrade to get Ambari Alert email notifications configured in the new Ambari Alerts system.

The Ambari Alerts system is configured automatically to replace Nagios but you must:

1. Configure email notifications in Ambari to handle dispatch of alerts. Browse to Ambari Web "Alerts".
2. In the Actions menu, select "Manage Notifications".
3. Click to "Create a new Notification". Enter information about the SMTP host, port to and from email addresses and select the Alerts to receive notifications.
4. Click "Save".

**Note**

(Optional) Remove the Nagios packages (nagios, nagios-www) from the Nagios host.

For more information Ambari Alerts, see Managing Alerts in the Ambari User’s Guide.

3.3.1.2. Moving from Ganglia to Ambari Metrics

After upgrading to Ambari 2.2, the Ganglia service stays intact in the cluster. You must perform the following steps to remove Ganglia from the cluster and to move to the new Ambari Metrics system.

**Important**

- If you are using HDP 2.2 Stack, Storm metrics will not work with Ambari Metrics until you are upgraded to HDP 2.2.4 or later.
- It is recommended that, after moving to Ambari Metrics restart Storm to pick up the new metrics sink classes.
- Do not add the Ambari Metrics service to your cluster until you have removed Ganglia using the steps below.

2. Using the Ambari REST API, remove the Ganglia service by executing the following:

```
curl -u admin.user:admin.password -H 'X-Requested-By:ambari' -X DELETE 'http://ambari.server:8080/api/v1/clusters/cluster.name/services/GANGLIA'
```

where `admin.user` and `admin.password` are credentials for an Ambari Administrator, `ambari.server` is the Ambari Server host and `cluster.name` is the name of your cluster.

3. Refresh Ambari Web and make sure that Ganglia service is no longer visible.

4. In the Actions menu on the left beneath the list of Services, use the "Add Service" wizard to add Ambari Metrics to the cluster.

5. This will install an Ambari Metrics Collector into the cluster, and an Ambari Metrics Monitor on each host.

6. Pay careful attention to following service configuration for Ambari Metrics.

By default, Ambari Metrics uses the local filesystem as the default storage backend. This is **embedded mode** and the rootdir for HBase is set to a local filesystem path by default when using Ambari Metrics in **embedded mode**. If you want to run Ambari Metrics in **distributed mode** (i.e. use HDFS instead of local filesystem for storage), set this rootdir value to an HDFS dir. For example: `hdfs://namenode.example.org:8020/apps/ams/metrics`. See Tuning Ambari Metrics for more information.

<table>
<thead>
<tr>
<th>Section</th>
<th>Property</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Advanced ams-hbase-site</td>
<td>hbase.rootdir</td>
<td>file:///var/lib/ambari-metrics-collector/hbase</td>
</tr>
</tbody>
</table>

Note

(Optional) Remove the Ganglia packages (ganglia-gmetad and ganglia-gmond) from the hosts.

Important

If you are managing a HDP 2.2 cluster that includes Kafka, you must adjust the Kafka configuration to send metrics to the Ambari Metrics system.

From Ambari Web, browse to Services > Kafka > Configs and edit the kafka-env template found under Advanced kafka-env to include the following:

```bash
# Add kafka sink to classpath and related dependencies
if [ -e "/usr/lib/ambari-metrics-kafka-sink/ambari-metrics-kafka-sink.jar" ];
then
    export CLASSPATH=$CLASSPATH:/usr/lib/ambari-metrics-kafka-sink/ambari-metrics-kafka-sink.jar
```

7. For the cluster services to start sending metrics to Ambari Metrics, restart all services. For example, restart HDFS, YARN, HBase, Flume, Storm and Kafka.
export CLASSPATH=$CLASSPATH:/usr/lib/ambari-metrics-kafka-sink/lib/*
fi

### 3.3.2. Adjusting for Kerberos

**Important**

You only need to perform these Kerberos steps if you are running Ambari 1.7 or earlier and your cluster is Kerberos enabled. You do not need to perform these steps if you are running Ambari 2.0 or later.

If you are upgrading to Ambari 2.2 from an Ambari-managed cluster that is already Kerberos-enabled, because of new Kerberos features introduced in Ambari, you need perform the following steps:

1. Review the procedure for Configuring Ambari and Hadoop for Kerberos in the Ambari Security Guide.

2. Have your Kerberos environment information readily available, including your KDC Admin account credentials.

3. Take note of current Kerberos security settings for your cluster.
   
   a. Browse to Services > HDFS > Configs.

   b. Record the core-site auth-to-local property value.

4. Disable Kerberos before performing the Ambari upgrade. In Ambari Web, browse to Admin > Kerberos, and click Disable Kerberos. You can re-enable Kerberos after performing the Ambari upgrade.

5. Upgrade Ambari according to the steps in Upgrading to Ambari 2.2.

6. Ensure your cluster and the Services are healthy.

7. Browse to Admin > Kerberos and you’ll notice Ambari thinks that Kerberos is not enabled. Run the Enable Kerberos Wizard, following the instructions in the Ambari Security Guide. Be sure to pay close attention to the principal names in the Configure Identities step in the wizard to confirm principals names match what you expect for your environment.

8. Ensure your cluster and the Services are healthy.

9. Verify the Kerberos security settings for your cluster are correct.

   - Browse to Services > HDFS > Configs.

   - Check the core-site auth-to-local property value.

   - Adjust as necessary, based on the pre-upgrade value recorded in Step 3.
3.3.3. Upgrade Ambari Metrics

1. Upgrade to Ambari 2.2 and perform needed post-upgrade checks. Make sure all services are up and healthy.

2. Make sure Ambari Metrics service is stopped. From Ambari Web, browse to Services > Ambari Metrics and select Stop from the Service Actions menu.

3. On every host in your cluster running a Metrics Monitor, run the following commands:

   For RHEL/CentOS/Oracle Linux:
   
   ```
   yum clean all
   yum upgrade ambari-metrics-monitor ambari-metrics-hadoop-sink
   ```

   For SLES:
   
   ```
   zypper clean
   zypper up ambari-metrics-monitor ambari-metrics-hadoop-sink
   ```

   For Ubuntu/Debian:
   
   ```
   apt-get clean all
   apt-get update
   apt-get install ambari-metrics-assembly
   ```

4. Execute the following command on all hosts running the Metrics Collector:

   For RHEL/CentOS/Oracle Linux:
   
   ```
   yum upgrade ambari-metrics-collector
   ```

   For SLES:
   
   ```
   zypper up ambari-metrics-collector
   ```

   For Ubuntu/Debian:
   
   ```
   apt-get install ambari-metrics-assembly
   ```

5. Start Ambari Metrics Service. From Ambari Web, browse to Services > Ambari Metrics and select Start from the Service Actions menu.

6. Updated Ambari Metrics Sink jars will be installed on all hosts. You must restart each service to pick up the latest sink implementations.

   (For example: HDFS, YARN, Kafka, HBase, Flume, Storm)
4. Upgrading HDP

- Prerequisites [17]
- Prepare to Upgrade [19]
- Register and Install Target Version [20]
- Perform the Upgrade [22]

There are different HDP upgrade options based on your current HDP and the target HDP. This section describes the different upgrade options, their prerequisites, and the overall process. You are strongly encouraged to read completely through this entire document before starting the upgrade process, to that you understand the interdependencies and order of the steps. It is highly recommended you validate these steps in a test environment to adjust and account for any special configurations for your cluster.

The high-level process for performing an HDP upgrade is as follows:

Ambari will guide you through the steps required to upgrade HDP. Make sure Ambari and the cluster are healthy, operating normally and all service checks are passing. See Preparing to Upgrade Ambari and HDP for more information on performing the Ambari upgrade.

Note

Be sure to review the available HDP upgrade scenarios below. It is strongly recommended that you first upgrade to Ambari 2.2 before upgrading HDP unless otherwise noted. After upgrading Ambari, be sure the cluster is operating normally and service checks are passing prior to attempting an HDP upgrade.
There are two methods for upgrading HDP with Ambari: **Rolling Upgrade** and **Express Upgrade**.

- A **Rolling Upgrade** orchestrates the HDP upgrade in an order that is meant to preserve cluster operation and minimize service impact during upgrade. This process has more stringent prerequisites (particularly regarding cluster high availability configuration) and can take longer to complete than an Express Upgrade.

- An **Express Upgrade** orchestrates the HDP upgrade in an order that will incur cluster downtime but with less stringent prerequisites.

The following table describes which upgrade option is available based on the current and target HDP version combinations. It is important to note that the HDP Stack version is based on the following format: `major.minor.maintenance.patch-build#`. A "minor" release is a release that increments the second-digit. A "maintenance" release is one that increments the third-digit. This terminology is used in the following table to describe the different upgrade paths. For example: if you want to upgrade from HDP 2.2 to HDP 2.3, that is a "Minor Upgrade". If you want to upgrade from HDP 2.2.x to HDP 2.2.y, that is a "Maintenance" Upgrade.

### Table 4.1. HDP Upgrade Options

<table>
<thead>
<tr>
<th>Upgrade Path</th>
<th>Current HDP*</th>
<th>Target HDP*</th>
<th>Rolling Upgrade</th>
<th>Express Upgrade</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maintenance</td>
<td>HDP 2.3.x.y</td>
<td>HDP 2.3.x.y</td>
<td>✅</td>
<td>✅</td>
</tr>
<tr>
<td>Maintenance</td>
<td>HDP 2.2.x.y</td>
<td>HDP 2.2.x.y</td>
<td>✅</td>
<td></td>
</tr>
<tr>
<td>Minor **</td>
<td>HDP 2.2</td>
<td>HDP 2.3</td>
<td>✅</td>
<td></td>
</tr>
<tr>
<td>Minor **</td>
<td>HDP 2.1</td>
<td>HDP 2.3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Minor</td>
<td>HDP 2.1</td>
<td>HDP 2.2</td>
<td></td>
<td>Contact Hortonworks Support for information on performing a HDP 2.1 to HDP 2.2 upgrade.</td>
</tr>
<tr>
<td>Minor</td>
<td>HDP 2.0</td>
<td>HDP 2.2</td>
<td></td>
<td>Ambari 2.2 does not support HDP 2.0 and you must upgrade to HDP 2.2 using either Ambari 1.7 or 2.0 before going to Ambari 2.2.</td>
</tr>
</tbody>
</table>

* The instructions in this document sometimes refer to an HDP "version" using the HDP #.#.#.x.y convention. For example, you will see the HDP 2.3.x.y convention to refer to any HDP 2.3 "maintenance" release. Therefore, to use a specific HDP 2.3 maintenance release, be sure to replace 2.3.x.y in the following instructions with the appropriate maintenance version, such as 2.3.0.0 for the HDP 2.3 GA release, or 2.3.2.0 for an HDP 2.3 maintenance release.

** A Minor upgrade can be performed from any current maintenance release to any target maintenance release in that minor release. For example, you can go from HDP 2.2.x.y to HDP 2.3.x.y as part of a Minor upgrade.

*** An Express Upgrade from HDP 2.1 -> 2.3 has no option to "Downgrade" since the bits for HDP 2.1 will have to be removed midway through the process.
4.1. Prerequisites

To perform an HDP upgrade using Ambari, your cluster must meet the following prerequisites. These prerequisites are required regardless of Rolling or Express upgrade because these checks are important for Ambari to know the cluster is in a healthy operating mode and can be successfully managed from Ambari. However, there are additional prerequisites for a Rolling Upgrade that require your cluster to be configured a certain way to support the rolling upgrade orchestration process.

<table>
<thead>
<tr>
<th>Check</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HDP Version</td>
<td>All hosts must have the target HDP version installed. See the Register and Install Target Version section for more information.</td>
</tr>
<tr>
<td>Disk Space</td>
<td>Be sure to have adequate space on /usr/hdp for the target HDP version. Each complete install of an HDP version will occupy about 2.5 GB of disk space.</td>
</tr>
<tr>
<td>Ambari Agent Heartbeats</td>
<td>All Ambari Agents must be communicating and heartbeating to Ambari Server. Any hosts that are not heartbeating must be in Maintenance Mode.</td>
</tr>
<tr>
<td>Host Maintenance Mode</td>
<td>The following two scenarios are checked:</td>
</tr>
<tr>
<td></td>
<td>• Any hosts in Maintenance Mode must not be hosting any Service Master Components.</td>
</tr>
<tr>
<td></td>
<td>• Any host in Maintenance Mode that is not hosting Master Components is allowed but you will receive a warning. You can proceed with your upgrade but these hosts will not be upgraded and before you can finalize the upgrade, you must delete the hosts from the cluster.</td>
</tr>
<tr>
<td>Service Maintenance Mode</td>
<td>No Services can be in Maintenance Mode.</td>
</tr>
<tr>
<td>Services Started</td>
<td>All Services must be started.</td>
</tr>
<tr>
<td>Service Checks</td>
<td>All Service Checks must pass. Be sure to run Service Actions &gt; Run Service Check on all services (and remediate if necessary) prior to attempting an HDP upgrade.</td>
</tr>
</tbody>
</table>

4.1.1. Rolling Upgrade Prerequisites

To perform a Rolling Upgrade, your cluster must meet the following prerequisites. If you do not meet these upgrade prerequisites, you can consider an Express Upgrade.

HDFS

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NameNode HA</td>
<td>NameNode HA must be enabled and working properly. See the Ambari User's Guide for more information, Configuring NameNode High Availability.</td>
</tr>
<tr>
<td>NameNode Truncate</td>
<td>In HDP 2.2, truncate must be disabled and this prerequisite is REQUIRED. There is an hdfs-site configuration dfs.allow.truncate, This configuration property is set to true if truncate is enabled. If the property is not set or is false, truncate is disabled. In 2.3 onwards, truncate is always enabled, which is acceptable. Therefore, this check is not applicable and will not be performed with HDP 2.3 maintenance upgrades.</td>
</tr>
</tbody>
</table>

YARN
<table>
<thead>
<tr>
<th>Requirement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Timeline Service State Recovery</strong></td>
<td>YARN state recovery should be enabled. Check the Services &gt; YARN &gt; Configs &gt; Advanced property yarn.timeline-service.recovery.enabled is set to true.</td>
</tr>
<tr>
<td><strong>ResourceManager Work Preserving Recovery</strong></td>
<td>YARN work preserving recovery should be enabled. Check the Services &gt; YARN &gt; Configs &gt; Advanced property yarn.resourcemanager.work-preserving-recovery.enabled is set to true.</td>
</tr>
<tr>
<td><strong>ResourceManager HA</strong></td>
<td>ResourceManager HA should be enabled to prevent a disruption in service during the upgrade. See the Ambari User's Guide for more information on Configuring ResourceManager High Availability. This prerequisite is OPTIONAL.</td>
</tr>
</tbody>
</table>

**MapReduce2**

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MapReduce Distributed Cache</strong></td>
<td>MapReduce should reference Hadoop libraries from the distributed cache in HDFS. Refer to the YARN Resource Management guide for more information.</td>
</tr>
<tr>
<td><strong>JobHistory State Recovery</strong></td>
<td>JobHistory state recovery should be enabled. Check the following: Services &gt; MapReduce &gt; Configs &gt; Advanced property mapreduce.jobhistory.recovery.enable is set to true. Once enabled, dependent properties for mapreduce.jobhistory.recovery.store.class and mapreduce.jobhistory.recovery.store.leveldb.path should also be set.</td>
</tr>
<tr>
<td><strong>Encrypted Shuffle</strong></td>
<td>If encrypted shuffle has been enabled, the ssl-client.xml file must be copied to /etc/hadoop/conf/secure directory on each node in the cluster.</td>
</tr>
</tbody>
</table>

**Tez**

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Tez Distributed Cache</strong></td>
<td>Tez should reference Hadoop libraries from the distributed cache in HDFS. Check the Services &gt; Tez &gt; Configs configuration and confirm tez.lib.uris is set with the path in HDFS for the tez.tar.gz file.</td>
</tr>
</tbody>
</table>

**Hive**

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Hive Dynamic Service Discovery</strong></td>
<td>HiveServer2 dynamic service discovery should be configured for Rolling Upgrade.</td>
</tr>
<tr>
<td><strong>HiveServer2 Port</strong></td>
<td>During the upgrade, Ambari will switch the HiveServer2 port from 10000 to 10010 (or 10011 if using HTTP transport mode).</td>
</tr>
<tr>
<td><strong>Hive Client Retry</strong></td>
<td>Hive client retry properties must be configured. Review the Services &gt; Hive &gt; Configs configuration and confirm hive.metastore.failure.retries and hive.metastore.client.connect.retry.delay are specified.</td>
</tr>
<tr>
<td><strong>Multiple Hive Metastore</strong></td>
<td>Multiple Hive Metastore instances are recommended for Rolling Upgrade. This ensures that there is at least one Hive Metastore running during the upgrade process. Additional Hive Metastore instances can be added through Ambari. This prerequisite is OPTIONAL.</td>
</tr>
</tbody>
</table>

**Oozie**

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Oozie Client Retry</strong></td>
<td>Oozie client retry properties must be configured. Review the Services &gt; Oozie &gt; Configs &gt; oozie-env configuration and confirm</td>
</tr>
<tr>
<td>Requirement</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>-------------</td>
</tr>
<tr>
<td>&quot;export OOZIE_CLIENT_OPTS=&quot;${OOZIE_CLIENT_OPTS} -Doozie.connection.retry.count=&lt;number of retries&gt;&quot;</td>
<td>is specified.</td>
</tr>
</tbody>
</table>

### 4.2. Prepare to Upgrade

Review [Preparing to Upgrade Ambari and HDP](#). It is also strongly recommended that you perform backups of your databases before beginning upgrade.

- Ambari database
- Hive Metastore database
- Oozie Server database
- Ranger Admin database
- Ranger Audit database

#### Checkpoint HDFS

1. Perform the following steps on the NameNode host. If you are configured for NameNode HA, perform the following steps on the Active NameNode. You can locate the Active NameNode from [Ambari Web > Services > HDFS](#) in the Summary area.

2. Check the NameNode directory to ensure that there is no snapshot of any prior HDFS upgrade. Specifically, using Ambari Web, browse to [Services > HDFS > Configs](#), and examine the `dfs.namenode.name.dir` in the NameNode Directories property. Make sure that only a "/current" directory and no "/previous" directory exists on the NameNode host.

3. Create the following log and other files. Creating these logs allows you to check the integrity of the file system after the Stack upgrade.

   As the HDFS user, "su -l <HDFS_USER>" run the following (where <HDFS_USER> is the HDFS Service user, for example, hdfs):

   - 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.

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

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

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

   - **Optional:** Capture the complete namespace of the file system. The following command does a recursive listing of the root file system:

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

   - **Optional:** Copy all unrecoverable data stored in HDFS to a local file system or to a backup instance of HDFS.
4. Save the namespace. As the HDFS user, "su -l <HDFS_USER> ", you must put the cluster in Safe Mode.

```bash
hdfs dfsadmin -safemode enter
hdfs dfsadmin -saveNamespace
```

**Note**

In a highly-available NameNode configuration, the command ```hdfs dfsadmin -saveNamespace``` sets a checkpoint in the first NameNode specified in the configuration, in ```dfs.ha.namenodes.[nameserviceID]```.

You can also use the ```dfsadmin -fs``` option to specify which NameNode to connect. For example, to force a checkpoint in NameNode2:

```bash
hdfs dfsadmin -fs hdfs://namenode2-hostname:namenode2-port -saveNamespace
```

5. Copy the checkpoint files located in ```${dfs.namenode.name.dir}/current``` into a backup directory.

**Note**

In a highly-available NameNode configuration, the location of the checkpoint depends on where the ```saveNamespace``` command is sent, as defined in the preceding step.

6. Store the layoutVersion for the NameNode located at

```bash
${dfs.namenode.name.dir}/current/VERSION
```

into a backup directory where ```${dfs.namenode.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.

7. As the HDFS user, "su -l <HDFS_USER> ", take the NameNode out of Safe Mode.

```bash
hdfs dfsadmin -safemode leave
```

8. Finalize any prior HDFS upgrade, if you have not done so already. As the HDFS user, " su -l <HDFS_USER> ", run the following:

```bash
hdfs dfsadmin -finalizeUpgrade
```

Proceed to [Register and Install Target Version](#).

### 4.3. Register and Install Target Version

This section describes the steps to register the software repositories with Ambari for the new target version (i.e. the version you will be upgrading to) and how to install the software on all host prior to performing the upgrade.
4.3.1. Register Target Version

1. Log in to Ambari.

2. Browse to Admin > Stack and Versions.

3. Click on the Versions tab. You will see the version currently running, marked as "Current".

   **Note**
   
   The full version is dependent on the HDP version you are actually running. For example, if you are currently running the HDP 2.2.0.0 release, you would see something like HDP-2.2.0.0-2557 as the full version number.

4. Click Manage Versions.

5. Proceed to register a new version by clicking + Register Version.

6. Enter the "name" for the version you are registering. The name is made up of two parts; the "HDP-major.minor" version, and the "VersionNumber".
   
   a. Select the "HDP-major.minor" version from the dropdown.
   
   b. Enter a "two-digit Version Number" in the input field.

   For example, if you are registering version "HDP-2.3.0", select HDP and enter 2.0.

7. Select one or more OS families and enter the respective Base URLs for each repository.

   To use the public software repositories, see the list of available HDP Repositories for each OS. Or, if you are using a local repository, enter the Base URLs for the local repository you have created. Refer to Using a Local Repository for more information in setting-up a local repository in your environment.

8. Click Save.

9. Ambari will attempt to validate the Base URLs by checking that the Base URLs entered are valid, reachable and resolve as software repositories.

10. Click Go To Dashboard, and browse back to Admin > Stack and Versions > Versions.

   You will see the version currently running, marked "Current", and the version you just registered HDP-2.3.2.0 which is showing an Install Packages button.

11. Proceed to Install Target Version.
4.3.2. Install Target Version

1. Log in to Ambari.
2. Browse to Admin > Stack and Versions.
3. Click on the Versions tab.

4. Click Install Packages and click OK to confirm.

5. The Install version operation will start. This will install the target version on all hosts in the cluster. You can monitor the progress of the install by clicking the “Installing” link.

6. Once the install is complete, the Install Packages button will be replaced with the option to Perform Upgrade.

7. Proceed to Perform the Upgrade.

4.4. Perform the Upgrade

1. Log in to Ambari.
2. Browse to Admin > Stack and Versions.
3. Click on the Versions tab. The registered and installed target HDP version should be visible.

4. Click Perform Upgrade on the target version.

5. Based on your current HDP version and the target HDP version, Ambari will perform a set of prerequisite checks to determine if you can perform a Rolling Upgrade or an Express Upgrade. A dialog will be presented with the options available.

6. Select your upgrade method: Rolling Upgrade or Express Upgrade. There are also advanced options available.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Skip all Service Check failures</td>
<td>Ambari will automatically skip any Service Check failures and complete the task without requiring user intervention to continue. After all the Service Checks have run in a task, you will be presented with summary of the failures and an option to continue the upgrade or pause.</td>
</tr>
<tr>
<td>Skip all Slave Component failures</td>
<td>Ambari will automatically skip any Slave Component failures and complete the task of upgrading Slave components without requiring user intervention to continue. After all Slave Components have been</td>
</tr>
<tr>
<td>Option</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Option upgraded, you will be presented with a summary of the failures and an option to continue the upgrade or pause.</td>
<td></td>
</tr>
</tbody>
</table>

7. Click **Proceed**. Based on your option choice, proceed with the upgrade:

- If you selected Rolling Upgrade, proceed to **Perform Rolling Upgrade**.
- If you selected Express Upgrade, proceed to **Perform Express Upgrade**.

### 4.4.1. Perform Rolling Upgrade

1. **Ambari** will check that your cluster meets the **Rolling Upgrade Prerequisites**. A dialog will be presented with the results:

   a. If any prerequisites are not met but are required, the result will be shown with an error. You will not be allowed to proceed with the upgrade. Make the appropriate corrections and return to Perform Upgrade again.

   b. If any prerequisites are not met but are optional, the result will be shown with a warning. You will be allowed to Proceed with the upgrade.

   c. A list of configuration changes (if any) will be displayed.

2. Once the prerequisite checks are complete, the upgrade will start. The time it takes to perform the upgrade depends on many factors. As part of the upgrade process, each component in the cluster is restarted in a serial fashion as such, the stop/start time is a big contributor to the overall time.

3. The upgrade process involves a set of stages. This table lists the high-level stages and if the process requires any action by you during normal operation.

    **Note**
    
    If any stage fails, the upgrade will halt and prompt for action.

<table>
<thead>
<tr>
<th>Stage</th>
<th>Description</th>
<th>Action Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prepare Backups</td>
<td>This is a confirmation step used to confirm that you have taken proper backups before proceeding.</td>
<td>You must acknowledge the prompt for database backups.</td>
</tr>
<tr>
<td>ZooKeeper</td>
<td>All ZooKeeper servers are upgraded and restarted.</td>
<td>None</td>
</tr>
<tr>
<td>Ranger</td>
<td>Ranger Admin and UserSync servers are upgraded and restarted.</td>
<td>None. If Ranger Admin does not function after the upgrade completes, run the upgrade scripts manually, then Retry Upgrading Ranger, as described <a href="#">here</a>.</td>
</tr>
<tr>
<td>Core Masters</td>
<td>This stage upgrades the master components of core services. This includes JournalNodes &amp; NameNodes (HDFS), HistoryServer (MapReduce2), ResourceManager &amp; ATS (YARN) and HBase Masters (HBase).</td>
<td>None</td>
</tr>
<tr>
<td>All Service Checks</td>
<td>All Service Checks are performed against the cluster.</td>
<td>If any service check fails, you will be prompted to Ignore and Continue, Downgrade or Retry. If you selected the &quot;Skip all Service Check failures&quot; option, you will only be prompted at the completion of all of the Service Checks.</td>
</tr>
<tr>
<td>Stage</td>
<td>Description</td>
<td>Action Required</td>
</tr>
<tr>
<td>-----------------------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>--------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Core Slaves</td>
<td>This stage upgrades the slave components of core services. This includes DataNodes (HDFS), RegionServers (HBase) and NodeManagers (YARN). This is done in two batches: 20% of the slaves first, then the remaining slaves.</td>
<td>After the first 20% batch is complete, you will be prompted to Verify the cluster is operating correctly.</td>
</tr>
<tr>
<td>All Service Checks</td>
<td>All Service Checks are performed against the cluster.</td>
<td>If any service check fails, you will be prompted to Ignore and Continue, Downgrade or Retry. If you selected the “Skip all Service Check failures” option, you will only be prompted at the completion of all of the Service Checks.</td>
</tr>
<tr>
<td>Hive</td>
<td>This stage upgrades the Hive Metastore, HiveServer2 and WebHCat components and the Hive clients.</td>
<td>Ambari will switch the HiveServer2 port from 10000 to 10010 (or, 10011 if using HTTP transport mode). You will be prompted to confirm prior to the switch.</td>
</tr>
<tr>
<td>Spark</td>
<td>The Spark Job History Server and clients are upgraded.</td>
<td>None</td>
</tr>
<tr>
<td>Oozie</td>
<td>The Oozie Server and clients are upgraded.</td>
<td>None</td>
</tr>
<tr>
<td>Falcon</td>
<td>The Falcon Server and clients are upgraded.</td>
<td>None</td>
</tr>
<tr>
<td>Client Components</td>
<td>All remaining clients are upgraded.</td>
<td>None</td>
</tr>
<tr>
<td>All Service Checks</td>
<td>All Service Checks are performed against the cluster.</td>
<td>If any service check fails, you will be prompted to Ignore and Continue, Downgrade or Retry. If you selected the “Skip all Service Check failures” option, you will only be prompted at the completion of all of the Service Checks.</td>
</tr>
<tr>
<td>Kafka</td>
<td>The Kafka Brokers are upgraded.</td>
<td>None</td>
</tr>
<tr>
<td>Knox</td>
<td>The Knox Gateways are upgraded.</td>
<td>None</td>
</tr>
<tr>
<td>Storm</td>
<td>The Nimbus, REST API, DRPC Server and UI Server components are upgraded, along with the Supervisors.</td>
<td>None</td>
</tr>
<tr>
<td>Slider</td>
<td>The Slider clients are upgraded.</td>
<td>None</td>
</tr>
<tr>
<td>Flume</td>
<td>The Flume agents are upgraded.</td>
<td>None</td>
</tr>
<tr>
<td>Finalize Upgrade Pre-Check</td>
<td>Checks if any hosts were not upgraded, either because the host was in Maintenance Mode, or one or more components on the host failed to upgrade (and were skipped).</td>
<td>Click on the list that displays &quot;# hosts&quot; for details on the hosts (and their components) that are not upgraded. You can Pause Upgrade, delete the hosts and return to finalize.</td>
</tr>
<tr>
<td>Finalize</td>
<td>The component upgrades are complete. You are presented the option to Finalize, which when selected, completes the upgrade process + saves the cluster state.</td>
<td>Prompted to Finalize, Finalize Later or Downgrade.</td>
</tr>
</tbody>
</table>

4. Once the upgrade is complete, you have an option to **Finalize** the upgrade, to **Finalize Later** or to **Downgrade**. Finalizing later gives you a chance to perform more validation on the cluster. Downgrade moves the cluster version back to the previous version (basically: the reverse of the upgrade process stages). **Once finalized, you cannot downgrade back to the previous version.**

**Note**

If you choose to finalize later, both versions will be listed on the Stack and Versions tab with the starting version displaying as Current. It is not until you finalize that Ambari makes the target version the current version. Also,
until you finalize, you will not be able to perform operational changes to the cluster (such as move components, change configurations, etc).

5. Click **Finalize** and the upgrade process is complete.

### 4.4.2. Perform Express Upgrade

1. Ambari will check that your cluster meets the **Prerequisites**. A dialog will be presented with the results:

   a. If any prerequisites are not met but are required, the result will be shown with an error. You will not be allowed to proceed with the upgrade. Make the appropriate corrections and return to Perform Upgrade again.

   b. If any prerequisites are not met but are optional, the result will be shown with a warning. You will be allowed to Proceed with the upgrade.

   c. A list of configuration changes (if any) will be displayed.

2. Once the prerequisite checks are complete, the upgrade will start. The time it takes to perform the upgrade depends on many factors. As part of the Express upgrade process, each component in the cluster is restarted in parallel across all hosts, usually requiring less time than the serial approach used in the Rolling Upgrade.

3. The upgrade process involves a set of stages. This table lists the high-level stages and if the process requires any action by you during normal operation.

   ![Note](image)

   > If any stage fails, the upgrade will halt and prompt for action.

<table>
<thead>
<tr>
<th>Stage</th>
<th>Description</th>
<th>Action Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stop Components for High-Level Services</td>
<td>This will stop all components for High-Level Services. This includes all master components except those of HDFS, HBase, ZooKeeper and Ranger.</td>
<td>None</td>
</tr>
<tr>
<td>Perform Backups</td>
<td>This is a confirmation step used to confirm that you have taken proper backups before proceeding.</td>
<td>You must acknowledge the prompt for database backups.</td>
</tr>
<tr>
<td>Stop Components for Core Service</td>
<td>Stops all components with HDFS, HBase, ZooKeeper and Ranger.</td>
<td>None</td>
</tr>
<tr>
<td>Update Target Stack</td>
<td>Updates the stack version in Ambari to the target version.</td>
<td>None</td>
</tr>
<tr>
<td>Update Service Configs</td>
<td>Updates (i.e. transfers or replaces) any configurations that are necessary for the upgrade.</td>
<td>None</td>
</tr>
<tr>
<td>Restart Components</td>
<td>Restarts all core components such as ZooKeeper, Ranger, HDFS, YARN, MapReduce2 and various Clients (Tez, Pig, Sqoop).</td>
<td>None</td>
</tr>
<tr>
<td>All Service Checks</td>
<td>All Service Checks are performed against the cluster.</td>
<td>If any service check fails, you will be prompted to Ignore and Continue, Downgrade or Retry. If you selected the “Skip all Service Check failures” option, you will only be prompted at the completion of all of the Service Checks.</td>
</tr>
<tr>
<td>Stage</td>
<td>Description</td>
<td>Action Required</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>------------------------------------------------------------------------------------------------</td>
<td>---------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Restart Components</td>
<td>Restarts the remaining components such as Oozie, Falcon, Hive, Spark and others.</td>
<td>None</td>
</tr>
<tr>
<td>Set Version on All Hosts</td>
<td>Sets the HDP version on all hosts to the target HDP version.</td>
<td>None</td>
</tr>
<tr>
<td>Finalize Upgrade Pre-Check</td>
<td>Checks if any hosts were not upgraded, either because the host was in Maintenance Mode, or one or more components on the host failed to upgrade (and were skipped).</td>
<td>Click on the list that displays &quot;# hosts&quot; for details on the hosts (and their components) that are not upgraded. You can Pause Upgrade, delete the hosts and return to finalize.</td>
</tr>
<tr>
<td>Finalize Upgrade</td>
<td>The component upgrades are complete. You are presented the option to Finalize, which when selected, completes the upgrade process + saves the cluster state.</td>
<td>Prompted to Finalize or Finalize Later or Downgrade.</td>
</tr>
</tbody>
</table>

4. Once the upgrade is complete, you have an option to Finalize the upgrade, to Finalize Later or to Downgrade. Finalizing later gives you a chance to perform more validation on the cluster. Downgrade moves the cluster version back to the previous version (basically: the reverse of the upgrade process stages). Once finalized, you cannot downgrade back to the previous version.

**Note**

If you choose to finalize later, both versions will be listed on the Stack and Versions tab with the starting version displaying as Current. It is not until you finalize that Ambari makes the target version the current version. Also, until you finalize, you will not be able to perform operational changes to the cluster (such as move components, change configurations, etc).

5. Click Finalize and the upgrade process is complete.

**Important**

An Express Upgrade from HDP 2.1 -> 2.3 has no option to "Downgrade" since the bits for HDP 2.1 will have to be removed midway through the process.