

# Hortonworks Data Platform

## Apache Ambari Minor Upgrade

(September 21, 2018)

## Hortonworks Data Platform: Apache Ambari Minor Upgrade

Copyright © 2012-2018 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 Ambari .....	4
3.2. Upgrade Ambari .....	5
3.3. <i>Mandatory</i> Post-Upgrade Tasks .....	10
3.3.1. Upgrading Ambari Infra .....	11
3.3.2. Upgrading Ambari Log Search .....	12
3.3.3. Upgrading Ambari Metrics .....	13
3.4. Upgrading SmartSense .....	14
4. Upgrading HDP .....	15
4.1. Prerequisites .....	16
4.2. Prepare to Upgrade .....	17
4.2.1. Checkpoint HDFS .....	18
4.3. Register and Install Target Version .....	19
4.3.1. Performing a Patch Upgrade .....	21
4.3.2. Limitation: Patch Upgrades in Mixed-OS Environments .....	23
4.3.3. Reverting a Patch Upgrade .....	23
4.4. Perform the Upgrade .....	25
4.4.1. Perform Rolling Upgrade .....	25
4.4.2. Perform Express Upgrade .....	27
4.5. Upgrade Troubleshooting .....	29

# 1. Upgrading Ambari and HDP

Ambari and the stack managed by Ambari can be upgraded independently. Use this guide to upgrade an Ambari-2.7.0 managed, HDP-3.0.0 cluster to an Ambari-2.7.1-managed, HDP-3.0.1 cluster.



## Important

Ambari 2.7.1 **only supports fully managing a HDP 3.0.1** cluster.

If you need to upgrade from Ambari-2.6.x / HDP-3.0.0, use the instructions in the Ambari Major Upgrade Guide.

### Next Steps

[Getting Ready to Upgrade Ambari and HDP \[2\]](#)

### More Information

[Ambari Major Upgrade Guide](#)

## 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

Always ensure that you are using the most recent version of Ambari to perform your upgrade.

- 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.
- Ensure point-in-time backups are taken of all databases that support the cluster. This includes (among others) Ambari, Hive, Ranger, Druid, Superset, and Oozie.

-

### For Large Clusters

In a large cluster, NameNode startup processes can take a long time. NameNode startup time depends not only on host properties, but also on data volume and network parameters. To ensure that the Ambari requests to start the NameNode do not timeout during an upgrade, you should configure the Ambari NameNode restart timeout parameter, `upgrade.parameter.nn-restart.timeout` in `/etc/ambari-server/conf/ambari.properties` on the Ambari Server host. You may need to add the restart timeout parameter and value to the Ambari server host, following a default installation. For a large cluster, you should add ten percent to the usual time (in seconds) required to restart your NameNode. Although no standard way to determine an appropriate value exists, you may use the following guidance:

For example, record the time (seconds) required to restart the active NameNode for your current Ambari server version. If restarting takes 10 minutes, (600 seconds), then add

```
upgrade.parameter.nn-restart.timeout=660
```

to the `/etc/ambari-server/conf/ambari.properties` file on the Ambari Server host.

After adding or resetting the Ambari NameNode restart parameter, restart your Ambari server before starting the HDP upgrade.

```
ambari-server restart
```

-

### For Ambari Upgrades

- Be sure to review the Known Issues and Behavioral Changes for this Ambari-2.7.1 release.

-

### For HDP Cluster Upgrades

- Ensure sufficient disk space on `/usr/hdp/<version>` (roughly 3GB for each additional HDP release).
- If your cluster includes Storm, document any running Storm topologies, as they will need to be stopped during the upgrade process.

### Next Steps

[Upgrading Ambari \[4\]](#)

### More Information

[Managing Admin Credentials](#)

[Hortonworks Support Matrix](#)

[Using a Local Repository](#)

[Ambari 2.7.1 Release Notes](#)

[Preparing to Upgrade Ambari \[4\]](#)

## 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 Ambari \[4\]](#)
- [Upgrade Ambari \[5\]](#)
- The high-level process for upgrading Ambari is as follows:



### Next Steps

[Preparing to Upgrade Ambari \[4\]](#)

### More Information

[Hortonworks Support Matrix](#)

### 3.1. Preparing to Upgrade Ambari

- Be sure to review the Ambari 2.7.1.0 release notes for Known Issues and Behavioral Changes.
- You **must** have root, administrative, or root-equivalent authorization on the Ambari Server host and all Ambari Agent hosts 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 your cluster is SSO-enabled, do not stop Knox before upgrading Ambari.



## Note

This guide specifically describes upgrading an Ambari-2.7.0-managed HDP-3.0.0 cluster to Ambari-2.7.1/HDP-3.0.0. If you need to upgrade from Ambari-2.6.x / HDP-2.6.x, use the instructions in the Ambari Major Upgrade Guide.

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.

### Next Steps

[Upgrade Ambari \[5\]](#)

### More Information

[Ambari Major Upgrade Guide](#)

[Upgrade SmartSense](#)

[Ambari 2.7.1.0 Release Notes](#)

## 3.2. Upgrade Ambari

1. Turn off Auto Restart from Ambari Web by browsing to **Admin > Service Auto Start**. Set **Auto-Start Services** to **Disabled**. Click **Save**.

From Ambari Web, browse to **Admin > Service Auto Start**. Set **Auto-Start Services** to **Disabled**. Click **Save**.

2. If you are running SmartSense in your cluster, stop the service.

From **Ambari Web**, browse to **Services > SmartSense** and select **Stop** from the **Service Actions** menu. Then, select **Turn on Maintenance Mode** from the **Service Actions** menu.

3. If you are running Ambari Metrics in your cluster, stop the service and put it in Maintenance Mode.

From **Ambari Web**, browse to **Services > Ambari Metrics** and select **Stop** from the **Service Actions** menu. Then, select **Turn on Maintenance Mode** from the **Service Actions** menu.

4. If you are running Log Search in your cluster, stop the service.

From **Ambari Web**, browse to **Services > Log Search** and select **Stop** from the **Service Actions** menu. Then, select **Turn on Maintenance Mode** from the **Service Actions** menu.

5. Stop the Ambari Server. On **the host** running Ambari Server:

```
ambari-server stop
```

6. Stop all Ambari Agents. On **each host** in your cluster running an Ambari Agent:

```
ambari-agent stop
```



7. 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 7:**

```
wget -nv https://username:password@archive.cloudera.com/p/ambari/2.x/2.7.1.0/centos7/ambari.repo -O /etc/yum.repos.d/ambari.repo
```

- **For SLES 12:**

```
wget -nv https://username:password@archive.cloudera.com/p/ambari/2.x/2.7.1.0/sles12/ambari.repo -O /etc/zypp/repos.d/ambari.repo
```

- **For Ubuntu 14:**

```
wget -nv https://username:password@archive.cloudera.com/p/ambari/2.x/2.7.1.0/ubuntu14/ambari.list -O /etc/apt/sources.list.d/ambari.list
```

- **For Ubuntu 16:**

```
wget -nv https://username:password@archive.cloudera.com/p/ambari/2.x/2.7.1.0/ubuntu16/ambari.list -O /etc/apt/sources.list.d/ambari.list
```

8. Upgrade Ambari Server. On the host running Ambari Server:

- **For RHEL/CentOS/Oracle/Amazon Linux:**

```
yum clean all
```

```
yum info ambari-server
```

In the info output, visually validate that there is an available version containing "2.7"

```
yum upgrade ambari-server
```

- **For SLES:**

```
zypper clean
```

```
zypper info ambari-server
```

In the info output, visually validate that there is an available version containing "2.7"

```
zypper up ambari-server
```

- **For Ubuntu/Debian:**

```
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 "2.7"

```
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-2.7.1-750.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-server**, then click the **Enter** button.
- d. On the right side you will see the search result **ambari-server 2.7.1**. Click **Actions**, choose **Update**, then click the **Enter** button.
- e. Go to **Accept**, and click **enter**.

9. Check for upgrade success by noting progress during the Ambari Server installation process you started in Step 8.

- 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 0:2.7.1 Complete!
```



### Important

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-2.7.1*.jar` to `/tmp` before proceeding with upgrade.

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

- **For RHEL/CentOS/Oracle/Amazon 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-2.7.1-750.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 2.7.1`. 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
```

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

**For RHEL/CentOS/Oracle Linux 7, Amazon Linux 2:**

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

**For SLES 12:**

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

**For Ubuntu 14:**

```
dpkg -l ambari-agent
```

**For Ubuntu 16:**

```
dpkg -l ambari-agent
```

**For Ubuntu 18:** `dpkg -l ambari-agent`

**For Debian 9:** `dpkg -l ambari-agent`

12. Upgrade Ambari Server database schema. On **the host** running Ambari Server:

```
ambari-server upgrade
```

When the Ambari Server database schema has been upgraded, you should see command output like this:

```
Ambari Server 'upgrade' completed successfully
```

13. Start the Ambari Server. On **the host** running Ambari Server:

```
ambari-server start
```

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

```
ambari-agent start
```

15. Open Ambari Web.

Point your browser to the Ambari Web UI:

- When Ambari Server is configured for HTTPS:

```
https://<your.ambari.server>:8443
```

- When Ambari Server is configured for HTTP:

```
http://<your.ambari.server>:8080
```

where <your.ambari.server> is the name of your ambari server host. For example, c7401.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.

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

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

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".



### Important

DO NOT RESTART these services unless future steps in the upgrade guide prompt you to do so. Manually restarting these services may significantly disrupt your upgrade. Ambari will restart each service automatically during the HDP upgrade.



### Note

Even though the installer prompts you to sync ldap, doing so is not required.



### Important

Ambari Metrics System, and SmartSense must be upgraded **after** first registering the HDP 3.0.x version and installing HDP 3.0.x packages. Please do not attempt to start Ambari Metrics or SmartSense until the HDP 3.0.x upgrade has completed.

#### Next Steps

[Upgrading HDP \[15\]](#)

#### More Information

[Configuring Ambari Authentication for LDAP/AD](#)

[Using a new or existing database with Hive](#)

[Using an existing database with Oozie](#)

[Configuring a Database Instance for Ranger](#)

[Install Databases for HDF services](#)

## 3.3. Mandatory Post-Upgrade Tasks

Depending on the configuration of your cluster and your current Ambari version, you must upgrade any of the following features in your cluster, as described in the following topics:

#### [Upgrading Ambari Infra](#)

If your cluster includes Ambari Infra service, you must upgrade it along with Ambari.

#### [Upgrading Ambari Log Search](#)

If your cluster includes Ambari Log Search service, you must upgrade it along with Ambari.

#### [Upgrading Ambari Metrics](#)

If your cluster includes the Ambari Metrics System (AMS) service, you must upgrade the system along with Ambari.

#### [Upgrading SmartSense](#)

If your cluster includes the SmartSense service, you must upgrade it after upgrading Ambari.

## Next Steps

Restart services, only after you complete all applicable, post-upgrade tasks.

### 3.3.1. Upgrading Ambari Infra

If you have Ambari Solr installed, you must upgrade Ambari Infra after upgrading Ambari.

#### Steps

1. Make sure Ambari Infra services are stopped. From **Ambari Web**, browse to **Services > Ambari Infra** and select **Stop** from the **Service Actions** menu.
2. On every host in your cluster with an Infra Solr Client installed, run the following commands:

#### For RHEL/CentOS/Oracle Linux:

```
yum clean all
```

```
yum upgrade ambari-infra-solr-client
```

#### For SLES:

```
zypper clean
```

```
zypper up ambari-infra-solr-client
```

#### For Ubuntu/Debian:

```
apt-get clean all
```

```
apt-get update
```

```
apt-get install ambari-infra-solr-client
```

3. Execute the following command on all hosts running an Ambari Infra Solr Instance:

#### For RHEL/CentOS/Oracle Linux:

```
yum upgrade ambari-infra-solr
```

#### For SLES:

```
zypper up ambari-infra-solr
```

#### For Ubuntu/Debian:

```
apt-get install ambari-infra-solr
```

4. Start the Ambari Infra services.

From **Ambari Web**, browse to **Services > Ambari Infra** select **Service Actions** then choose **Start**.

#### More Information

[Ambari Infra](#)

## 3.3.2. Upgrading Ambari Log Search

If you have Ambari Log Search installed, you must upgrade Ambari Log Search after upgrading Ambari.

### Prerequisites

Before starting this upgrade, ensure the Ambari Infra components have been upgraded.

### Steps

1. Make sure Ambari Log Search service is stopped. From **Ambari Web**, browse to **Services > Log Search** and select **Stop** from the **Service Actions** menu.
2. On every host in your cluster running a Log Feeder, run the following commands:

#### For RHEL/CentOS/Oracle Linux:

```
yum clean all
```

```
yum upgrade ambari-logsearch-logfeeder
```

#### For SLES:

```
zypper clean
```

```
zypper up ambari-logsearch-logfeeder
```

#### For Ubuntu/Debian:

```
apt-get clean all
```

```
apt-get update
```

```
apt-get install ambari-logsearch-logfeeder
```

3. Execute the following command on all hosts running the Log Search Server:

#### For RHEL/CentOS/Oracle Linux:

```
yum upgrade ambari-logsearch-portal
```

#### For SLES:

```
zypper up ambari-logsearch-portal
```

#### For Ubuntu/Debian:

```
apt-get install ambari-logsearch-portal
```

4. Start Log Search Service.

From **Ambari Web**, browse to **Services > Log Search** select **Service Actions** then choose **Start**.

### More Information

[Upgrading Ambari Infra](#)

### 3.3.3. Upgrading Ambari Metrics

#### Prerequisites

Upgrade to Ambari 2.6 and perform needed post-upgrade checks. Make sure all services are up and healthy.

#### Steps

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

3. 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 clean all
```

```
apt-get update
```

```
apt-get install ambari-metrics-collector
```

4. Execute the following command on the host running the Grafana component:

#### For RHEL/CentOS/Oracle Linux:

```
yum upgrade ambari-metrics-grafana
```

#### For SLES:



```
zypper up ambari-metrics-grafana
```

**For Ubuntu/Debian:**

```
apt-get clean all
```

```
apt-get update
```

```
apt-get install ambari-metrics-grafana
```

**5. Start Ambari Metrics Service.**

From **Ambari Web**, browse to **Services > Ambari Metrics** select **Service Actions** then choose **Start**.

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

Please wait to restart all services until after you have completed all applicable post-upgrade tasks, for example: HDFS, YARN, Kafka, HBase, Flume, Storm.

**Next Steps**

- Restart services, only after you complete all applicable, post-upgrade tasks.

**Note**

New Ambari Metrics Sinks will not be activated until all services are restarted.

- If you are upgrading from Ambari 2.2.1 or earlier, and your Ambari Metrics service does not contain Grafana, proceed to add Grafana to Ambari Metrics.

## 3.4. Upgrading SmartSense

If your cluster includes the SmartSense service, you must upgrade it after upgrading Ambari.

**More Information**

[Upgrading SmartSense](#)

**Next Steps**

Restart services.

## 4. Upgrading HDP

- [Prerequisites \[16\]](#)
- [Prepare to Upgrade \[17\]](#)
- [Register and Install Target Version \[19\]](#)

This topic describes available 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.



### Note

Be sure to review the available HDP upgrade scenarios below. It is **strongly recommended** that you **first upgrade to Ambari 2.7** 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.

Ambari 2.7.x does not support IOP-HDP migration. Customers migrating from IOP should use Ambari 2.6.1 to migrate IOP to HDP 2.6.4, then upgrade Ambari to 2.7 and use it to upgrade HDP 2.6.4 to HDP 3.0.

Ambari-2.7.1 supports both rolling and express methods for upgrading HDP-3.0 to HDP-3.0.1.

An **Express Upgrade** orchestrates the HDP upgrade in an order that will incur cluster downtime.

### Next Steps

[Prerequisites \[16\]](#)

### More Information

[Preparing to Upgrade Ambari and HDP](#)

## 4.1. Prerequisites

To perform an HDP upgrade using Ambari, your cluster must meet the following prerequisites. Meeting these prerequisites is essential for Ambari to know the cluster is in a healthy operating mode and can successfully manage the upgrade process.

<b>Disk Space</b>	Be sure to have adequate space on <code>/usr/hdp</code> for the target HDP version. Each complete install of an HDP version will occupy about 8 GB of disk space.
<b>Ambari Agent Heartbeats</b>	All Ambari Agents must be communicating and heartbeating to Ambari Server. Any hosts that are not heartbeating must be in Maintenance Mode.
<b>Hive Upgrade</b>	<p>The upgrade process does not back up the Hive MetaStore, nor does it compact ACID tables. Before upgrading Hive, you must:</p> <ul style="list-style-type: none"><li>• Manually make a manual backup of your Hive metastore database before upgrading.</li><li>• If you have ACID tables in your Hive metastore, enable ACID operations using Ambari Web or set Hive configuration properties to enable ACID.</li></ul>
<b>Host Maintenance Mode</b>	<p>The following two scenarios are checked:</p> <ul style="list-style-type: none"><li>• Any hosts in Maintenance Mode must not be hosting any Service Master Components.</li><li>• 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 <b>before</b> you can finalize the upgrade, you must delete the hosts from the cluster.</li></ul>
<b>Service Maintenance Mode</b>	No Services can be in Maintenance Mode, except for Ambari Metrics System, SmartSense, and Log Search.
<b>Services Started</b>	All Services must be started, except for Ambari Metrics System, SmartSense, and Log Search.
<b>Service Checks</b>	All Service Checks must pass. Be sure to run <b>Service Actions &gt; Run Service Check</b> on all services (and remediate if necessary) prior to attempting an HDP upgrade.
<b>KDC Admin Credentials</b>	The Ambari Server will add new components as part of the HDP 2.6 to HDP 3.0 upgrade and needs to be configured to enable password encryption and saving the KDC admin credentials so necessary principals can be created. For steps on how to do this, see

[Encrypt Database and LDAP Passwords in Ambari and Managing Admin Credentials.](#)

**KDC Admin Host FQDN**

Ensure that the KDC Admin host is set to a fully qualified domain name (FQDN) and not an IP address. You can check this setting by going to **Services > Kerberos > Configs > KDC Hosts**.

**KDC kadmin principal**

The MIT KDC integration in Ambari 2.7 has been improved to connect more securely with the Kerberos Administration server (kadmind). This increased security expects a Kerberos admin service principal to be present with a specifically formatted principal name. The format expected is `kadmin/fully.qualified.kdc.hostname@REALM`. This expectation is a behavior change from previous versions of Ambari, where having such a Kerberos admin service principal was not required.

This principal is present by default in most MIT KDC installations, but some customers have reported that this principal does not exist in their KDC. Due to this, it's recommended that before upgrading a kerberized cluster to Ambari 2.7, you ensure this principal exists in your KDC database.

**All Prior Upgrades are Finalized**

Any prior upgrade started with Ambari must be finalized.

**Apache Zeppelin Notebooks and Configuration File Storage**

In releases of Zeppelin earlier than HDP-2.6.3, notebooks and configuration files were stored on the local disk of the Zeppelin server. In HDP-2.6.3+, the default Zeppelin storage is HDFS.

When upgrading to HDP-2.6.3+ from versions earlier than HDP-2.6.3, perform the steps described in [Enabling HDFS Storage for Zeppelin Notebooks and Configuration in HDP-2.6.3+](#).

**Next Steps**

[Prepare to Upgrade \[17\]](#)

**More Information**

[Register and Install Target Version \[19\]](#)

[Remove Service](#)

## 4.2. Prepare to Upgrade

Make sure that you have reviewed and completed all [prerequisites](#) described in previous chapters.

It is **strongly** recommended that you perform backups of all your databases before beginning the upgrade.

### Next Steps

[Checkpoint HDFS \[18\]](#)

## 4.2.1. 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-lsr-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.

```
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:

```
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

`[$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.

```
hdfs dfsadmin -safemode leave
```

### Next Steps

[Register and Install Target Version \[19\]](#)

## 4.3. Register and Install Target Version

Before you use Ambari to perform the stack upgrade, you must register the software repositories for the new target version with Ambari and then install the software on all hosts in the cluster.

### Register Target Version

#### Steps

1. Log in to Ambari.

2. Browse to **Cluster Admin > Stack and Versions**.
3. Click the **Versions** tab. You see the version currently running, marked as **Current**.



### Note

The full version depends on the HDP version you are actually running. For example, if you are currently running the HDP 2.6.5.0 release, you would see something like **HDP-2.6.5.0-292** as the full version number.

4. Click **Manage Versions**.
5. Proceed to register a new version by clicking **Register Version**.
6. Select the software version and method of delivery for your cluster.

- a. **Choose HDP Stack.**

Available HDP minor versions display on tabs. When you click a tab, Ambari displays available maintenance versions for that HDP Stack on a drop-down list. When you click a specific maintenance version, a list of available Services and their version displays in a table.

- b. **Choose HDP Version.**

If your Ambari host has internet access, available maintenance versions display as options in a drop-down list. If you have a Version Definition File (VDF) for a version that is not listed, you can click **Add Version...** and upload the VDF file. In addition, a **Default Version Definition** is also included in the list if you do not have Internet access or are not sure which specific version to install. If you choose the **Default Version Definition**, you must enter a "two-digit Version Number" in the **Name** input field.

- c. **Choose Repository Delivery Method.**

- Use private Repository

Using a private repository requires internet connectivity. To use the private software repositories, see the list of available HDP Repositories for each OS.

- Use Local Repository

Using a local repository requires that you have configured the software in a repository available in your network. If you are using a local repository, enter the Base URLs for your local repository.

7. Click **Save**.
8. Click **Dashboard**.

## Install Target Version

### Steps

1. Browse to **Cluster Admin > Stack and Versions**.

2. Click the **Versions** tab.
3. On a registered target version, click **Install Packages** and click **OK** to confirm.

The Install version operation starts. This installs the target version on all hosts in the cluster. You can monitor the progress of the install by clicking the **Installing** link.

When the installation completes, the **Upgrade** button replaces the **Install Packages** button.

### More Information

[Managing Versions](#)

[Using a Local Repository](#)

## 4.3.1. Performing a Patch Upgrade

A patch release contains a unique software fix that affects one or more, but not all, services in your cluster. A four-digit release number identifies each patch release. Performing a patch upgrade applies changes only to those services in your cluster affected by the patch release.



### Note

Applying a patch upgrade is supported only for clusters running HDP-2.6.4 and above.

### Prerequisites

Before performing a patch upgrade, you must obtain from Hortonworks Customer Support, the specific VDF file associated with the patch release.

### Steps

1. Register the patch release as a target version, using the Add Version option, just as you would during any upgrade.

Patch versions include a "bug" icon to indicate that the patch includes a bug fix.



	Stack	Versions	Upgrade History
	<a href="#">Manage Versions</a>		Filter: All (2) ▾
		Current	Current ▾
		HDP-2.6.4.0	⚙ HDP-2.6.4.1-23
		<a href="#">Show Details</a>	<a href="#">Show Details</a>
HDFS		2.7.3	2.7.3
YARN		2.7.3	2.7.3
MapReduce2		2.7.3	2.7.3
Tez		0.7.0	0.7.0
HBase		1.1.2	1.1.2
ZooKeeper		3.4.6	3.4.6
Ambari Infra		0.1.0	0.1.0
Ambari Metrics		0.1.0	0.1.0
Atlas		0.8.0	0.8.0
Kafka		0.10.1	0.10.1
Ranger		0.7.0	0.7.0
SmartSense		1.4.3.2.6.1.0-134	1.4.3.2.6.1.0-134
Kerberos		1.10.3-10	1.10.3-10

2. Install the patch release as the current version, just as you would during any upgrade.
3. Restart affected services.

When you perform a Patch Express Upgrade of a service, a dependent service may stop and require a manual restart. For example, a patch Express Upgrade of HDFS could stop HBase. In such cases, you must manually re-start HBase using the Ambari Web UI > Service Actions menu.

### More Information

[Limitation: Patch Upgrades in Mixed-OS Environments \[23\]](#)

[Register and Install Target Version \[19\]](#)

[Perform Express Upgrade \[27\]](#)

[Performing Service Actions](#)

[Reverting a Patch Upgrade \[23\]](#)

## 4.3.2. Limitation: Patch Upgrades in Mixed-OS Environments

### Summary

Applying a patch upgrade to a cluster with more than one version of an operating system, (mixed OS) may require that you manually rebuild tarballs to include forward-compatible native libraries and upload them to a location accessible by both operating systems.

### Problem Description

When you apply an HDP patch in a mixed OS environment (of the same OS family, for example RHEL 6 and RHEL 7), and if master components like Job History Server or Hive server are located on a RHEL7 machine, native libraries for RHEL7 are packaged in both the `mapreduce.tar.gz` and `tez.tar.gz` tarballs. Worker nodes running RHEL6 in the cluster will be unable to load the RHEL7 native libraries.

### Workaround

You must fetch any `mapreduce.tar.gz` and `tez.tar.gz` tarballs from a RHEL7 machine and rebuild them manually on a RHEL 6 machine. RHEL6 native libraries are forward-compatible. You must then upload the rebuilt tarballs to the correct locations in HDFS.

- Locate and extract `/usr/hdp/<version>/hadoop/mapreduce.tar.gz`
- Copy the files directory `/usr/hdp/<version>/hadoop/lib/native` into the `hadoop/lib` directory from the tarball so that `hadoop/lib/native` exists and contains all of the files and folders from the above location.
- Re-compress the tarball and upload it into HDFS at `/hdp/apps/<version>/mapreduce/mapreduce.tar.gz`
- Locate and extract `/usr/hdp/<version>/tez/lib/tez.tar.gz`
- Copy the files directory `/usr/hdp/<version>/hadoop/lib/native` into the `lib` directory from the tarball so that `lib/native` exists and contains all of the files and folders from the above location.
- Re-compress the tarball and upload it into HDFS at `/hdp/apps/<version>/tez/tez.tar.gz`

## 4.3.3. Reverting a Patch Upgrade

After installing a patch upgrade, you have the option to revert the fix. You can only revert the last installed version. If you have installed multiple patch releases, you must revert them in the reverse order you installed them.

### Steps

1. In the current patch version, choose Revert.

	Current	Current
	<div style="display: flex; justify-content: space-between;"> <span>Stack</span> <span>Versions</span> <span>Upgrade History</span> </div> <div style="display: flex; align-items: center; margin-top: 5px;"> <span style="background-color: #0070C0; color: white; padding: 2px 5px; border-radius: 3px;">Manage Versions</span> <span style="margin-left: 10px;">Filter: All (2) ▾</span> </div>	
	<div style="background-color: #0070C0; color: white; padding: 2px 5px; border-radius: 3px; display: inline-block;">Current</div> <b>HDP-2.6.4.0</b> <a href="#">Show Details</a>	<div style="background-color: #0070C0; color: white; padding: 2px 5px; border-radius: 3px; display: inline-block;">Current ▾</div> <div style="background-color: #0070C0; color: white; padding: 2px 5px; border-radius: 3px; display: inline-block; width: 100px;">Revert</div> <del>HDP-2.6.4.1-23</del> <a href="#">Show Details</a>
HDFS	<div style="background-color: #0070C0; color: white; padding: 2px 5px; border-radius: 3px;">2.7.3</div>	<div style="background-color: #808080; color: white; padding: 2px 5px; border-radius: 3px;">2.7.3</div>
YARN	<div style="background-color: #0070C0; color: white; padding: 2px 5px; border-radius: 3px;">2.7.3</div>	<div style="background-color: #808080; color: white; padding: 2px 5px; border-radius: 3px;">2.7.3</div>
MapReduce2	<div style="background-color: #0070C0; color: white; padding: 2px 5px; border-radius: 3px;">2.7.3</div>	<div style="background-color: #808080; color: white; padding: 2px 5px; border-radius: 3px;">2.7.3</div>
Tez	<div style="background-color: #0070C0; color: white; padding: 2px 5px; border-radius: 3px;">0.7.0</div>	<div style="background-color: #808080; color: white; padding: 2px 5px; border-radius: 3px;">0.7.0</div>
Hive	<div style="background-color: #0070C0; color: white; padding: 2px 5px; border-radius: 3px;">1.2.1000</div>	<div style="background-color: #808080; color: white; padding: 2px 5px; border-radius: 3px;">1.2.1000</div>
HBase	<div style="background-color: #0070C0; color: white; padding: 2px 5px; border-radius: 3px;">1.1.2</div>	<div style="background-color: #808080; color: white; padding: 2px 5px; border-radius: 3px;">1.1.2</div>
Pig	<div style="background-color: #0070C0; color: white; padding: 2px 5px; border-radius: 3px;">0.16.0</div>	<div style="background-color: #808080; color: white; padding: 2px 5px; border-radius: 3px;">0.16.0</div>
ZooKeeper	<div style="background-color: #0070C0; color: white; padding: 2px 5px; border-radius: 3px;">3.4.6</div>	<div style="background-color: #808080; color: white; padding: 2px 5px; border-radius: 3px;">3.4.6</div>
Ambari Infra	<div style="background-color: #808080; color: white; padding: 2px 5px; border-radius: 3px;">0.1.0</div>	<div style="background-color: #808080; color: white; padding: 2px 5px; border-radius: 3px;">0.1.0</div>
Ambari Metrics	<div style="background-color: #808080; color: white; padding: 2px 5px; border-radius: 3px;">0.1.0</div>	<div style="background-color: #808080; color: white; padding: 2px 5px; border-radius: 3px;">0.1.0</div>
Atlas	<div style="background-color: #0070C0; color: white; padding: 2px 5px; border-radius: 3px;">0.8.0</div>	<div style="background-color: #808080; color: white; padding: 2px 5px; border-radius: 3px;">0.8.0</div>
Kafka	<div style="background-color: #0070C0; color: white; padding: 2px 5px; border-radius: 3px;">0.10.1</div>	<div style="background-color: #808080; color: white; padding: 2px 5px; border-radius: 3px;">0.10.1</div>
SmartSense	<div style="background-color: #808080; color: white; padding: 2px 5px; border-radius: 3px;">1.4.3.2.6.1.0-134</div>	<div style="background-color: #808080; color: white; padding: 2px 5px; border-radius: 3px;">1.4.3.2.6.1.0-134</div>
Druid	<div style="background-color: #808080; color: white; padding: 2px 5px; border-radius: 3px;">0.10.1</div>	<div style="background-color: #0070C0; color: white; padding: 2px 5px; border-radius: 3px;">0.10.1</div>
Kerberos	<div style="background-color: #808080; color: white; padding: 2px 5px; border-radius: 3px;">1.10.3-10</div>	<div style="background-color: #808080; color: white; padding: 2px 5px; border-radius: 3px;">1.10.3-10</div>
Slider	<div style="background-color: #0070C0; color: white; padding: 2px 5px; border-radius: 3px;">0.92.0</div>	<div style="background-color: #808080; color: white; padding: 2px 5px; border-radius: 3px;">0.92.0</div>

2. Restart affected services.

When you perform a Patch Express Upgrade of a service, a dependent service may stop and require a manual restart. For example, a patch Express Upgrade of HDFS could stop HBase. In such cases, you must manually re-start HBase using the Ambari Web UI > Service Actions menu.

**More Information**

[Performing Service Actions](#)

## 4.4. Perform the Upgrade

To upgrade your HDP version:

### Steps

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

The registered and installed target HDP version displays an **Upgrade** button.

4. Click **Upgrade** on the target version.

Based on your current HDP version and the target HDP version, Ambari performs a set of prerequisite checks to determine if you can perform a rolling or an express upgrade. A dialog displays the options available.

5. Select the **Express Upgrade** method. (**Only supported method for upgrading from HDP 2.6 to 3.0.1**)

Advanced options are also available.

Skip all Service Check failures	Ambari automatically skips any Service Check failures and completes the task without requiring user actions. After all the Service Checks have run in a task, you see a summary of the failures and options to continue the upgrade or pause.
---------------------------------	---

Skip all Slave Component failures	Ambari automatically skips any Slave Component failures and completes the task of upgrading Slave components without requiring user actions. After all Slave Components have been upgraded, you see a summary of the failures and options to continue the upgrade or pause.
-----------------------------------	---

6. Click **Proceed**.

### Next Steps

[Perform Express Upgrade \[27\]](#)

### More Information

[Register and Install Target Version \[19\]](#)

[Prerequisites \[16\]](#)

### 4.4.1. Perform Rolling Upgrade

1. Ambari checks that your cluster meets prerequisites. A dialog displays the results:
  - a. If any *required* prerequisites are not met, the result displays an error.

You cannot proceed with the upgrade until you make the appropriate corrections and return to Perform Upgrade again.

- b. If any *optional* prerequisites are not met, the result displays a warning.

You may proceed with the upgrade.

- c. Ambari displays a list of configuration changes that occur during the upgrade.

- When the prerequisite checks complete, the upgrade starts. The time required to perform the upgrade depends on many factors. As part of the upgrade process, each component in the cluster restarts in a serial fashion. The stop/start times contribute to the total upgrade time.
- The upgrade process includes the following stages. Some stages require that you complete an action during normal operation.

If any stage fails, the upgrade stops and prompts you for action.

Stage	Description	Action Required
Prepare Backups	This step prompts you to confirm that you have taken proper backups before proceeding.	You must acknowledge the prompt for database backups.
ZooKeeper	All ZooKeeper servers are upgraded and restarted.	None
Ranger	Ranger Admin and UserSync servers are upgraded and restarted.	None. If Ranger Admin does not function after the upgrade completes, run the upgrade scripts manually, then Retry Upgrading Ranger.
Core Masters	This stage upgrades the master components of core services. This includes JournalNodes & NameNodes (HDFS), HistoryServer (MapReduce2), ResourceManager & ATS (YARN) and HBase Masters (HBase).	None
Druid	This stage upgrades Druid components in the following order: historical, middle manager, broker, router, coordinator, overlord.	None
All Service Checks	All Service Checks are performed against the cluster.	Any service check that fail prompt you to <b>Ignore and Continue, Downgrade or Retry</b> . If you selected the <b>Skip all Service Check failures</b> option, you will only be prompted when all Service Checks complete.
Core Slaves	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.	After the first 20% batch completes, you are prompted to verify the cluster is operating correctly.
All Service Checks	All Service Checks are performed against the cluster.	Any service check that fail prompt you to <b>Ignore and Continue, Downgrade or Retry</b> . If you selected the <b>Skip all Service Check failures</b> option, you will only be prompted when all Service Checks complete.
Spark	The Spark Job History Server and clients are upgraded.	None
Oozie	The Oozie Server and clients are upgraded.	None

Stage	Description	Action Required
Client Components	All remaining clients are upgraded.	None
All Service Checks	All Service Checks are performed against the cluster.	Any service check that fails prompts you to <b>Ignore and Continue</b> , <b>Downgrade</b> or <b>Retry</b> . If you selected the <b>Skip all Service Check failures</b> option, you will only be prompted when all Service Checks complete.
Kafka	The Kafka Brokers are upgraded.	None
Knox	The Knox Gateways are upgraded.	In an SSO-enabled cluster: If the cookie is lost/session is expired:client should use local login to access the Ambari and proceed further.  For example: <ambari_host:ambari_port>/#/login/local
Storm	Storm does not support rolling upgrade from a previous HDP version to HDP-3.0.1.	The rolling upgrade process prompts you to stop Storm topologies, perform the upgrade, and redeploy your topologies.
Finalize Upgrade Pre-Check	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).	Click the list that displays # hosts for details on the hosts (and their components) that are not upgraded. You can <b>Pause Upgrade</b> , delete the hosts and return to finalize.
Finalize	The component upgrades are complete. You are prompted to <b>Finalize</b> . Finalizing completes the upgrade process and saves the cluster state.	Prompted to Finalize, Finalize Later or Downgrade.

- When the rolling upgrade stages complete, may choose 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: reverses 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).

- Click **Finalize** to complete the rolling upgrade process.

## 4.4.2. Perform Express Upgrade

- Ambari checks that your cluster meets prerequisites. A dialog displays the results:
  - If any *required* prerequisites are not met, the result displays an error.  
  
You cannot proceed with the upgrade until you make the appropriate corrections and return to Perform Upgrade again.
  - If any *optional* prerequisites are not met, the result displays a warning.  
  
You may proceed with the upgrade.
  - Ambari displays a list of configuration changes that occur during the upgrade.

2. When the prerequisite checks complete, the upgrade starts. The time required to perform the upgrade depends on many factors. As part of the upgrade process, each component in the cluster restarts in a serial fashion. The stop/start times contribute to the total upgrade time.
3. The upgrade process includes the following stages. Some stages require that you complete an action during normal operation.

If any stage fails, the upgrade stops and prompts you for action.

Stage	Description	Action Required
Prepare Upgrade	You should stop all apps on YARN queues, and deactivate & kill all running Storm topologies.	Perform the actions to prepare for the upgrade.
Stop Components for High-Level Services	This will stop all components for High-Level Services. This includes all master components <b>except</b> those of HDFS, HBase, ZooKeeper and Ranger.	None
Perform Backups	This step prompts you to confirm that you have taken proper backups before proceeding.	You must acknowledge the prompt for database backups.
Stop Components for Core Service	Stops all components with HDFS, HBase, ZooKeeper and Ranger.	None
Update Target Respository	Updates the stack version in Ambari to the target version.	None
Update Service Configs	Updates (i.e. transfers or replaces) any configurations that are necessary for the upgrade.	None
Restart Components	Restarts all core components such as ZooKeeper, Ranger, HDFS, YARN, MapReduce2 and various Clients (Tez, Pig, Sqoop).	In an SSO-enabled cluster: If the cookie is lost/session is expired:client should use local login to access the Ambari and proceed further.  For example: <ambari_host:ambari_port>/#/login/local
All Service Checks	All Service Checks are performed against the cluster.	Any service check that fails prompts you to <b>Ignore and Continue</b> , <b>Downgrade</b> or <b>Retry</b> . If you selected the <b>Skip all Service Check failures</b> option, you will only be prompted when all Service Checks complete.
Restart Components	Restarts the remaining components such as Oozie, Hive, Spark2 and others.	None
Set Version on All Hosts	Sets the HDP version on all hosts to the target HDP version.	None
Finalize Upgrade Pre-Check	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).	Click the list that displays <b># hosts</b> for details on the hosts (and their components) that are not upgraded. You can <b>Pause Upgrade</b> , delete the hosts and return to finalize.
Finalize Upgrade	The component upgrades are complete. You are presented the option to Finalize, which when selected, completes the upgrade process + saves the cluster state.	Prompted to Finalize or Finalize Later or Downgrade.

4. When the upgrade stages complete, you may choose to **Finalize** the upgrade, or to **Finalize Later**. Finalizing later gives you a chance to perform more validation on the cluster.



### 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** to complete the express upgrade process.

#### More Information

[Register and Install Target Version \[19\]](#)

[Prerequisites \[16\]](#)

## 4.5. Upgrade Troubleshooting

### YARN Registry DNS instance fails to start

The YARN Registry DNS instance will fail to start if another process on the host is bound to port 53. Please ensure no other services that are binding to port 53 are on the host where the YARN Registry DNS instance is deployed.