Hortonworks DataFlow

Ambari Managed HDF Upgrade

(November 9, 2017)

docs.cloudera.com

Hortonworks DataFlow: Ambari Managed HDF Upgrade

Table of Contents

List of Tables

1. Upgrading Ambari and Your Hortonworks Stack

This document describes how to upgrade HDF amd Ambari.

If you are upgrading an HDF cluster, you should perform the following steps.

1. Preparing to Upgrade
2. Upgrading your Management Pack
3. Upgrading Ambari
4. Upgrading HDF

If you are upgrading HDF services on an HDP cluster, you should perform the following steps:

1. Preparing to Upgrade
2. Upgrading Ambari
3. Upgrading HDP
4. Upgrading HDF Services

2. Preparing to Upgrade

When preparing to upgrade, we strongly recommend you review this pre-upgrade checklist of items to confirm your cluster operation is healthy. Attempting to upgrade a cluster that is operating in an unhealthy state may produce unexpected results.

2.1. General Upgrade Checklist

- 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.
- If you are using a local repository, 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 is required on all nodes in the cluster.
- Ensure point-in-time backups are taken of all DBs supporting the clusters. This includes Ambari and Ranger databases.

2.2. Ambari Upgrade Checklist

- This (Ambari 2.6.0) *Upgrade Guide* helps you upgrade your existing Ambari install to version 2.6.0. If you are upgrading to another Ambari version, use the *Ambari Upgrade Guide* for that version.
- Be sure to review the Known Issues and Behavioral Changes for this Ambari release in the *Release Notes*.

More Information

- Ambari 2.6.0 Release Notes

2.3. HDF Upgrade Checklist

- If you plan to add new services to your cluster, the new services may include new service accounts. You should perform any operational procedures required to support these new service accounts prior to performing your upgrade. The services accounts are typically required on all nodes in your cluster.
- If your cluster includes Storm, document any running Storm topologies.

3. Upgrading the Management Pack

About This Task

A management pack bundles service definitions, stack definitions, and stack add-on service definitions so they do not need to be included with the Ambari core functionality and can be updated in between major releases. Upgrade the management pack to ensure that you have the latest versions of the available Apache components.

Context

If you are upgrading to the latest version of Ambari, follow steps in the *Upgrading Ambari* chapter of this guide. If you are already running the latest version of Ambari, use the following steps to upgrade your management pack.

Steps

1. Back up your Ambari resources folder:

```
cp -r /var/lib/ambari-server/resources /var/lib/ambari-server/resources.
backup
```

2. Upgrade the HDF managemet pack with the command appropriate for your operating system:

   - **RHEL/CentOS/Oracle Linux 6:**

```
ambari-server upgrade-mpack \
--mpack=http://public-repo-1.hortonworks.com/HDF/centos6/3.x/updates/3.0.
2.0/tars/hdf_ambari_mp/hdf-ambari-mpack-3.0.2.0-76.tar.gz \
--verbose
```

   - **RHEL/CentOS/Oracle Linux 7:**

```
ambari-server upgrade-mpack \
--mpack=http://public-repo-1.hortonworks.com/HDF/centos7/3.x/updates/3.0.
2.0/tars/hdf_ambari_mp/hdf-ambari-mpack-3.0.2.0-76.tar.gz \
--verbose
```

   - **SUSE Linux Enterprise Server (SLES) v12 SP1**

```
ambari-server upgrade-mpack \
--mpack=http://public-repo-1.hortonworks.com/HDF/sles12/3.x/updates/3.0.2.
0/tars/hdf_ambari_mp/hdf-ambari-mpack-3.0.2.0-76.tar.gz \
--verbose
```

   - **Debian 7:**

```
ambari-server upgrade-mpack \
--mpack=http://public-repo-1.hortonworks.com/HDF/debian7/3.x/updates/3.0.
2.0/tars/hdf_ambari_mp/hdf-ambari-mpack-3.0.2.0-76.tar.gz \
--verbose
```

   - **Ubuntu 14:**

```
ambari-server upgrade-mpack \
```

```
--mpack=http://public-repo-1.hortonworks.com/HDF/ubuntu14/3.x/updates/3.0.
2.0/tars/hdf_ambari_mp/hdf-ambari-mpack-3.0.2.0-76.tar.gz \
--verbose
```

3. Restart Ambari.

More Information

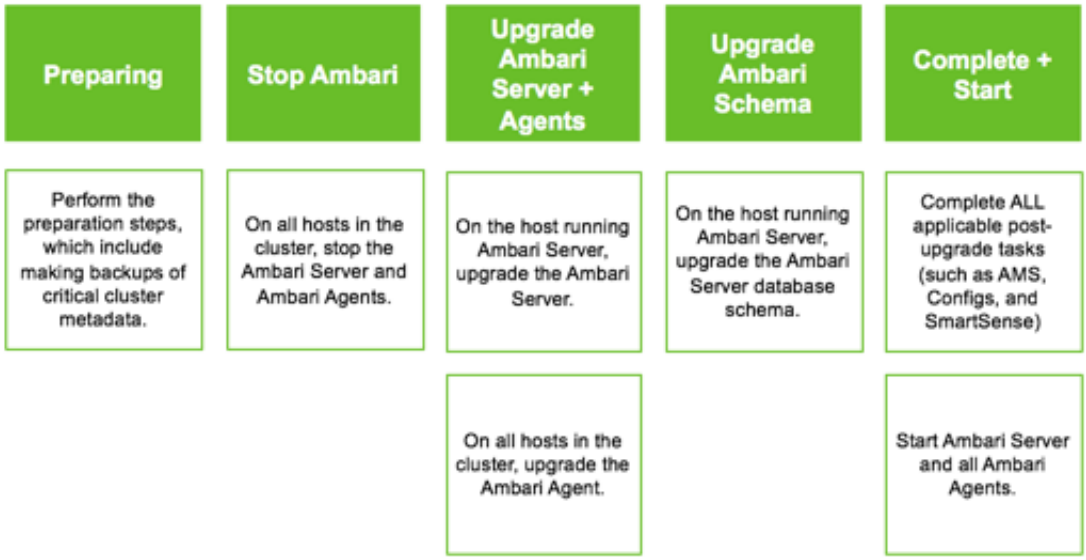
- Upgrading Ambari

4. 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 [5]
- Upgrade Ambari [6]
- *Mandatory* Post-Upgrade Tasks [11]

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



Important

Completing post-upgrade tasks is mandatory.

4.1. Preparing to Upgrade

- Be sure to review the Ambari 2.6.0.0 release notes for Known Issues and Behavioral Changes.
- You **must** have root, administrative, or root-equivalent authorization on the Ambari server host and all 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`.

- **Plan to upgrade the Ambari Metrics service:**
  - 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 System and add the Grafana component.
- After upgrading Ambari, you must also upgrade SmartSense.

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.

Next Steps

Upgrade Ambari [6]

More Information

Ambari 2.6.0.0 Release Notes

4.2. 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:**

```
wget -nv http://public-repo-1.hortonworks.com/ambari/centos6/2.x/updates/
2.6.0.0/ambari.repo -O /etc/yum.repos.d/ambari.repo
```

- **For RHEL/CentOS/Oracle Linux 7:**

```
wget -nv http://public-repo-1.hortonworks.com/ambari/centos7/2.x/updates/
2.6.0.0/ambari.repo -O /etc/yum.repos.d/ambari.repo
```

- **For SLES 11:**

```
wget -nv http://public-repo-1.hortonworks.com/ambari/suse11/2.x/updates/2.
6.0.0/ambari.repo -O /etc/zypp/repos.d/ambari.repo
```

- **For SLES 12:**

```
wget -nv http://public-repo-1.hortonworks.com/ambari/sles12/2.x/updates/2.
6.0.0/ambari.repo -O /etc/zypp/repos.d/ambari.repo
```

- **For Ubuntu 12:**

```
wget -nv http://public-repo-1.hortonworks.com/ambari/ubuntu12/2.x/updates/
2.6.0.0/ambari.list -O /etc/apt/sources.list.d/ambari.list
```

- **For Ubuntu 14:**

```
wget -nv http://public-repo-1.hortonworks.com/ambari/ubuntu14/2.x/updates/
2.6.0.0/ambari.list -O /etc/apt/sources.list.d/ambari.list
```

- **For Debian 7:**

```
wget -nv http://public-repo-1.hortonworks.com/ambari/debian7/2.x/updates/
2.6.0.0/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.

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.

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

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

```
yum clean all
```

```
yum info ambari-server
```

In the info output, visually validate that there is an available version containing "2.6"

```
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 "2.6"

```
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 "2.6"

```
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-2.5-101.noarch' to install this candidate". You will need to 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 2.6`. 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 0:2.6-111 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-2.6.*.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-2.5-101.noarch' to install this candidate". You will need to 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 2.6`. 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:

For RHEL/CentOS/Oracle Linux 6:	`rpm -qa | grep ambari-agent`
For RHEL/CentOS/Oracle Linux 7:	`rpm -qa | grep ambari-agent`
For SLES 11:	`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 Debian 7:	`dpkg -l 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
```

12. Open Ambari Web.

    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**.

    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".

14. If you have configured Ambari to authenticate against an external LDAP or Active Directory, you **must** re-run

```
ambari-server setup-ldap
```

15. 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.

16. If you are running **Ambari Metrics** service in your cluster, you **must** upgrade Ambari Metrics System and add the Grafana component.

17. If your cluster includes the SmartSense service, you **must** upgrade SmartSense along with Ambari.

18. Perform any other post-upgrade tasks, as necessary.

Important

Completing post-upgrade tasks is mandatory.

Next Steps

Post-Upgrade Tasks ***Mandatory***

4.3. *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. This will include adding the Grafana component to the system.
Adding Grafana to Ambari Metrics	Grafana is now included as a component of Ambari Metrics. 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.
Upgrading Configurations	Certain scenarios may require that you modify configurations that Ambari did not upgrade automatically.
Upgrading SmartSense	If your cluster includes the SmartSense service, you must upgrade it after upgrading Ambari.

4.3.1. 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**.

4.3.2. 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**.

4.3.3. Upgrading Ambari Metrics

Prerequisites

Upgrade to Ambari 2.5 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
   ```

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

   For RHEL/CentOS/Oracle Linux:

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

   For SLES:

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.

4.3.3.1. Adding Grafana to Ambari Metrics

As of Ambari 2.4, Grafana is included as a component of Ambari Metrics. You must add Grafana to the system and install Grafana on a host in the cluster.

Note

When using the API commands below, be sure to replace the **ambari.server** with the Ambari Server hostname, **cluster.name** with your cluster name and **host.name** with the host where you will run Grafana. This can be the same host that is running the Metrics Colllector.

1. Upgrade to Ambari 2.5 and perform needed post-upgrade checks. Make sure all services are up and healthy.
2. Add the METRICS_GRAFANA component to Ambari:

```
curl -u admin:admin -H "X-Requested-By:ambari" -i -X POST
http://ambari.server:8080/api/v1/clusters/cluster.name/services/
AMBARI_METRICS/components/METRICS_GRAFANA
```

3. Add METRICS_GRAFANA to a host in the cluster.

```
curl -u admin:admin -H "X-Requested-By:ambari" -i -X POST -d
'{"host_components":[{"HostRoles":{"component_name":"METRICS_GRAFANA"}}]}'
http://ambari.server:8080/api/v1/clusters/cluster.name/hosts?Hosts/
host_name=host.name
```

4. From **Ambari Web**, browse to **Services > Ambari Metrics** and you will see Grafana is in the **Install Pending...** state.

   You need to complete the configuration of Grafana before installing and starting.

5. To complete the configuration, click on **Services > Ambari Metrics > Configs** and enter the default Grafana Admin Password in the **General** section. Click **Save**.
6. Browse to **Hosts > host.name** (the **host.name** used in the API call where you added Grafana). You will see the Grafana component is in an **Install Pending...** state. Use the **Install Pending**... action button and select **Re-install**.

7. Once the install operation completes, select **Start** to start Grafana.

8. To access Grafana, browse to **Services > Ambari Metrics**, select **Quick Links** and then click **Grafana**.

4.3.4. Upgrading Configurations

This section describes potential cluster configuration updates that may be required.

Upgrading Kerberos krb5.conf [16]

Upgrading Log Rotation Configuration [16]

4.3.4.1. Upgrading Kerberos krb5.conf

Ambari has added support for handling more than one KDC host . Only one kadmin host is supported by the Kerberos infrastructure. This required modifications for the **krb5.conf** template. In order for Ambari to properly construct the krb5.conf configuration file, make the following configuration change if your cluster meets all of these criteria:

- Kerberos is enabled and Ambari is configured for automated setup, and
- Ambari is managing the krb5.conf, and
- You **have modified** the krb5.conf template content from the default content. If you have not modified the default content, Ambari will automatically update the template content as part of upgrade and these configuration updates do not need to be applied manually.

If you meet all of the above criteria, you must update the **krb5.conf** template content found in **Services > Kerberos > Advanced**:

Original Template Entry	Updated Template Entry
admin_server = {{admin_server_host|default(kdc_host, True)}}	admin_server = {{admin_server_host|default(kdc_host_list[0]|trim(), True)}}
kdc = {{kdc_host}}	{% for kdc_host in kdc_host_list %} kdc = {{kdc_host|trim()}} {%- endfor -%}

4.3.4.2. Upgrading Log Rotation Configuration

Ambari 2.5.0 provides s a simplified log rotation configuration. These changes will be made automatically during your next stack upgrade, but are not automatically made during the Ambari upgrade. After upgrading Ambari from version 2.x to 2.5.0, if you want to utilize

the simplified log rotation configuration, you must update configurations for all services in your cluster, using the following steps:

Steps

1. **ZooKeeper**

   a. In **Ambari Web**, browse to **ZooKeeper > Configs**.

   b. Scroll down to **Custom zookeeper-log4j**.

   c. In **Custom zookeeper-log4j**, click **Add Property**.

   d. In **Add Property**, type the following properties and values:

      zookeeper_log_max_backup_size=10

      zookeeper_log_number_of_backup_files=10

      For example:

   e. Click **Add**.

   f. Browse to **Advanced zookeeper-log4j**.

   g. In **Advanced zookeeper-log4j** *content section*, find and replace the following properties and values:

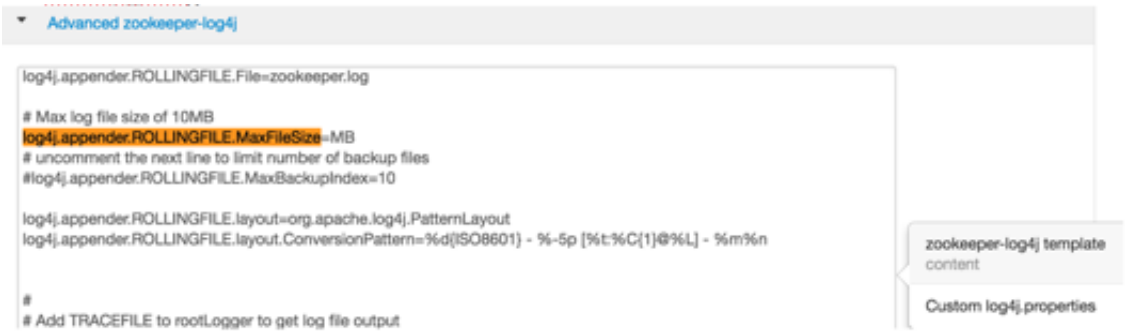
      Find: log4j.appender.ROLLINGFILE.MaxFileSize=<value>

      Replace:
      log4j.appender.ROLLINGFILE.MaxFileSize={{zookeeper_log_number_of_backup_files}}MB

      Find: #log4j.appender.ROLLINGFILE.MaxBackupIndex=<value>MB

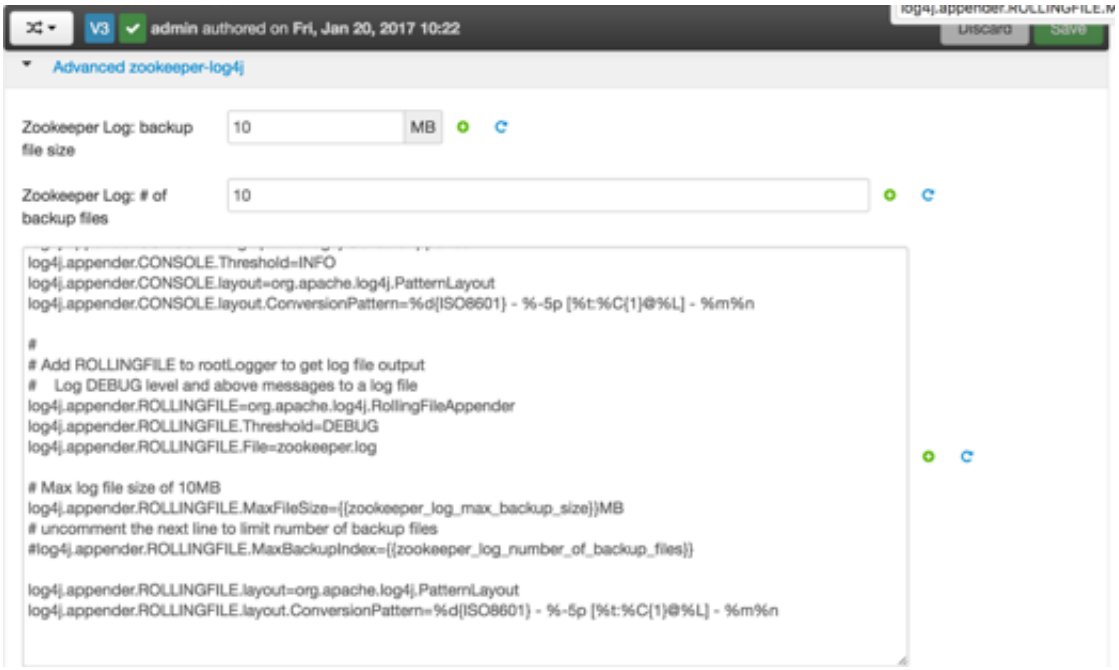
      Replace:
      #log4j.appender.ROLLINGFILE.MaxBackupIndex={{zookeeper_log_number_of_backup_files}}

      For example:



h. In **Configs**, click **Save**.

For example:



i. Restart **ZooKeeper**, as prompted.

2. **Kafka**

a. In **Ambari Web**, browse to **Kafka > Configs**.

b. Scroll down to **Custom Kafka-log4j**.

c. In **Custom Kafka-log4j**, click **Add Property**.

d. In **Add Property**, type the following properties and values:

kafka_log_maxfilesize=256

kafka_log_maxbackupindex=20

controller_log_maxfilesize=256

controller_log_maxbackupindex=20

e. Click **Add**.

f. Browse to **Advanced kafka-log4j**.

g. In **Advanced kafka-log4j** *content section*, find and replace the following properties and values:

Find: log4j.appender.kafkaAppender=org.apache.log4j.DailyRollingFileAppender

Add: log4j.appender.kafkaAppender.MaxFileSize = {{kafka_log_maxfilesize}}MB

Add: log4j.appender.kafkaAppender.MaxBackupIndex = {{kafka_log_maxbackupindex}}MB

Find: log4j.appender.controllerAppender=org.apache.log4j.DailyRollingFileAppender

Add: log4j.appender.controllerAppender.MaxFileSize = {{controller_log_maxfilesize}}MB

Add: log4j.appender.controllerAppender.MaxBackupIndex = {{controller_log_maxbackupindex}}

h. In **Configs**, click **Save**.

i. Restart **Kafka**, as prompted.

3. **Ranger**

a. In **Ambari Web**, browse to **Ranger > Configs > Advanced**.

b. Scroll down to **Custom admin-log4j**.

c. In **Custom admin-log4j**, click **Add Property**.

d. In **Add Property**, type the following properties and values:

ranger_xa_log_maxfilesize=256

ranger_xa_log_maxbackupindex=20

e. Click **Add**.

f. Browse to **Advanced admin-log4j**.

g. In **Advanced admin-log4j** *content section*, find and replace the following properties and values:

Find: log4j.appender.xa_log_appender=org.apache.log4j.DailyRollingFileAppender

Add: log4j.appender.xa_log_appender.MaxFileSize={{ranger_xa_log_maxfilesize}}MB

Add:
log4j.appender.xa_log_appender.MaxBackupIndex={{ranger_xa_log_maxbackupindex}}

h. Scroll down to **Custom usersync-log4j**.

i. In **Custom usersync-log4j**, click **Add Property**.

j. In **Add Property**, type the following properties and values:

ranger_usersync_log_maxfilesize=256

ranger_usersync_log_number_of_backup_files=20

k. Click **Add**.

l. Browse to **Advanced usersync-log4j**.

m. In **Advanced usersync-log4j** *content section*, find and replace the following properties and values:

Find: log4j.appender.logFile=org.apache.log4j.DailyRollingFileAppender

Add: log4j.appender.logFile.MaxFileSize = {{ranger_usersync_log_maxfilesize}}MB

Add: log4j.appender.logFile.MaxBackupIndex =
{{ranger_usersync_log_number_of_backup_files}}

n. Scroll down to **Custom tagsync-log4j**.

o. In **Custom tagsync-log4j**, click **Add Property**.

p. In **Add Property**, type the following properties and values:

ranger_tagsync_log_maxfilesize=256

ranger_tagsync_log_number_of_backup_files=20

q. Click **Add**.

r. Browse to **Advanced tagsync-log4j**.

s. In **Advanced tagsync-log4j** *content section*, find and replace the following properties and values:

Find: log4j.appender.logFile=org.apache.log4j.DailyRollingFileAppender

Add: log4j.appender.logFile.MaxFileSize = {{ranger_tagsync_log_maxfilesize}}MB

Add: log4j.appender.logFile.MaxBackupIndex =
{{ranger_tagsync_log_number_of_backup_files}}

t. In **Configs**, click **Save**.

u. Restart **Ranger**, as prompted.

4. **Ranger-KMS**

   a. In **Ambari Web**, browse to **Ranger-KMS > Configs > Advanced**.

   b. Scroll down to **Custom kms-log4j**.

   c. In **Custom kms-log4j**, click **Add Property**.

   d. In **Add Property**, type the following properties and values:

      ranger_kms_log_maxfilesize=256

      ranger_kms_log_maxbackupindex=20

      ranger_kms_audit_log_maxfilesize=256

      ranger_kms_audit_log_maxbackupindex=20

   e. Click **Add**.

   f. Browse to **Advanced kms-log4j** .

   g. In **Advanced kms-log4j** *content section*, find and replace the following properties and values:

      Find: log4j.appender.kms=org.apache.log4j.DailyRollingFileAppender

      Add: log4j.appender.kms.MaxFileSize = {{ranger_kms_log_maxfilesize}}MB

      Add: log4j.appender.kms.MaxBackupIndex = {{ranger_kms_log_maxbackupindex}}

      Find: log4j.appender.kms-audit=org.apache.log4j.DailyRollingFileAppender

      Add: log4j.appender.kms-audit.MaxFileSize={{ranger_kms_audit_log_maxfilesize}}MB

      Add: log4j.appender.kms-audit.MaxBackupIndex = {{ranger_kms_audit_log_maxbackupindex}}

   h. In **Configs**, click **Save**.

   i. Restart **Ranger-KMS**.

5. **Storm**

   a. In **Ambari Web**, browse to **Storm > Configs**.

   b. Scroll down to **Custom cluster-log4j property**.

   c. In **Custom cluster-log4j property**, click **Add Property**.

   d. In **Add Property**, type the following properties and values:

storm_a1_maxfilesize=100

storm_a1_maxbackupindex=9

e. Click **Add**.

f. Browse to **Advanced storm-cluster-log4j** .

g. In **Advanced storm-cluster-log4j** *content section*, find and replace the following properties and values:

Find: In RollingFile="A1"<SizeBasedTriggeringPolicy size="<value>MB"/>

Replace: <SizeBasedTriggeringPolicy size="{{storm_a1_maxfilesize}}MB"/>

Find: In RollingFile="A1"<DefaultRolloverStrategy max="<value>"/>

Replace: <DefaultRolloverStrategy max="{{storm_a1_maxbackupindex}}"/>

h. Scroll down to **Custom worker-log4j property**.

i. In **Custom worker-log4j property**, click **Add Property**.

j. In **Add Property**, type the following properties and values:

storm_wrkr_a1_maxfilesize=100

storm_wrkr_a1_maxbackupindex=9

storm_wrkr_out_maxfilesize=100

storm_wrkr_out_maxbackupindex=4

storm_wrkr_err_maxfilesize=100

storm_wrkr_err_maxbackupindex=4

k. Click **Add**.

l. Browse to **Advanced storm-worker-log4j** .

m. In **Advanced storm-worker-log4j** *content section*, find and replace the following properties and values:

Find: In RollingFile="A1"<SizeBasedTriggeringPolicy size="<value> MB"/>

Replace: <SizeBasedTriggeringPolicy size="{{storm_wrkr_a1_maxfilesize}} MB"/>

Find: In RollingFile="A1"<DefaultRolloverStrategy max="<value>"/>

Replace: <DefaultRolloverStrategy max="{{storm_wrkr_a1_maxbackupindex}}"/>

Find: In RollingFile="STDOUT"<SizeBasedTriggeringPolicy size="<value>" MB/>

Replace: <SizeBasedTriggeringPolicy size="{{storm_wrkr_out_maxfilesize}} MB"/>

Find: In RollingFile="STDOUT"<DefaultRolloverStrategy max="<value>"/>

Replace: <DefaultRolloverStrategy max="{{storm_wrkr_out_maxbackupindex}}"/>

Find: In RollingFile="STDERR"<SizeBasedTriggeringPolicy size="<value>" MB/>

Replace: <SizeBasedTriggeringPolicy size="{{storm_wrkr_err_maxfilesize}} MB"/>

Find: In RollingFile="STDOUT"<DefaultRolloverStrategy max="<value>"/>

Replace: <DefaultRolloverStrategy max="{{storm_wrkr_err_maxbackupindex}}"/>

n. In **Configs**, click **Save**.

o. Restart **Storm**, as prompted.

4.3.5. Upgrading SmartSense

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

More Information

Upgrading SmartSense

Next Steps

Restart services.

5. Upgrading to HDP 2.6.3

If you already have HDP 2.6.0 installed, upgrading your cluster to HDP 2.6.3 means:

- Keeping the same configuration files you used for HDP 2.6.0.
- Keeping the same data and metadata in the same location you used for HDP 2.6.0
- Installing any new components (added for the first time in HDP 2.6.0) side-by-side with existing components

5.1. Before you begin

- Ensure that you know which HDP components you need to upgrade at your installation.
- Decide whether you are going to upgrade using a local repository or a remote repository.
- If you are using the Falcon service, install the Berkeley DB prior to performing an upgrade.

  See the Prerequisite to Installing or Upgrading Falcon in the Data Movement and Integration guide.

5.2. Upgrade options

- If you are upgrading your cluster manually, use the Non-Ambari Upgrade Guide.
- If you are upgrading your cluster through Ambari, use the Ambari Upgrade Guide

More information:

- Upgrading HDP
- Register and Install HDP Version
- Obtain the HDP repos

6. Upgrading HDF

HDF offers an Ambari-managed Express Upgrade. An Express Upgrade orchestrates the HDF upgrade in an order that incurs some cluster downtime, but has less stringent prerequisites.

6.1. Prerequisites

To perform an HDF upgrade using Ambari, your cluster must meet the following prerequisites. These prerequisites are required because they allow Ambari to know whether the cluster is in a healthy operating mode and can be successfully managed from Ambari.

Table 6.1. Ambari-managed HDF Express Upgrade Prerequisites

Disk Space	Be sure to have adequate space on `/usr/hdf` for the target HDF installation.
Ambari Agent Heartbeats	All Ambari Agents must be communicating and heartbeating to Ambari Server. Any hosts that are not heartbeating must be in Maintenance Mode.
Host Maintenance Mode	The following two scenarios are checked: • Any hosts in Maintenance Mode must not be hosting any Service Master Components. • 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.
Service Maintenance Mode	No Services can be in Maintenance Mode.
Services Started	All Services must be started.
Service Checks	All Service Checks must pass. Be sure to run **Service Actions > Run Service Check** on all services (and remediate if necessary) prior to attempting an HDF upgrade.

6.2. Registering Your Target Version

About This Task

Registering your target version makes Ambari aware of the Hortonworks stack to which you want to upgrade, provides the public repo location, and specifies your public or private repo delivery preference.

Steps

1. Click the **Admin** tab, and then click **Stack and Versions**.
2. Click the **Versions** tab.
3. Click the **Manage Versions** button.
4. Click the + **Register Version** button.

5. Select the target version you want to regist, specify whether it will be a public or private repo, and select your operating system.

6. Click **Save**.

Result

From the **Versions** tab, you now see your target HDF version registered, but not yet installed.

6.3. Installing Your Target Version

About This Task

Installing your target version downloads the public repositories containing software packages for your target version onto each node in your cluster.

Steps

1. From the **Versions** tab, identify the target version you just registered, and click the **Install on ...** button.

2. Click **OK** to confirm.

3. You can monitor the progress of the install by clicking **Installing**.

Result

When the installation completes, you are able to see both your current and target HDF versions from **Admin | Stack and Versions | Versions**. Your target version has an active **Upgrade** button.

6.4. Upgrading HDF

About This Task

Upgrading HDF installs your target software version onto each node in your cluster. Note that the Express Upgrade is the only option available to HDF 3.0.0.

Steps

1. From **Admin | Stack and Versions | Versions**, click **Upgrade**.

2. In the **Upgrade Options** pop-up window, click **Express Upgrade**, and specify if you would like customized upgrade failure tolerance. If you select:

   - **Skip all Service Check failures** – Ambari skips any Service Check failures and completes the upgrade without requiring user intervention to continue. After all the Service Checks have run in a task, you are presented with summary of the failures and an option to continue the upgrade or pause.

   - **Skip all Slave Component failures** – Ambari skips any Slave Component failures and completes the Slave components upgrade without requiring user intervention to

continue. After all Slave Components have been upgraded, youare presented with a summary of the failures and an option to continue the upgrade or pause.

3. Click **Proceed**.

4. Once the upgrade completes, again confirm that you have performed the required manual steps and click **Finalize**.

Result

From **Admin | Stack and Versions | Versions**, you are now able to see only the HDF version to which you upgraded.

6.5. Restarting NiFi Certificate Authority

After you have upgraded to your target HDF version, you will need to restart the NiFi Certificate Authority (CA).

1. From **Services | NiFi | Configs**, click **Restart**.

2. Click **Confirm Restart All**.

7. Upgrading HDF Services

About This Task

If you are upgrading HDF Services in an Ambari-managed HDP cluster, some manual upgrade steps are required on each host where and HDF service is running.

Steps

1. Update the Base URL for HDF to the URL for HDF 3.0.2.

```
[root@host ~]# vi /etc/yum.repos.d/HDF.repo
baseurl=<os-specific-base-url>
```

   For the Base URL appropriate for your OS, see the HDF 3.0.2 Release Notes.

2. Confirm the HDF components and version you have installed.

   For example:

```
[root@host registry]# yum list installed | grep HDF

hdf-select.noarch                 3.0.0.0-453.el6           @HDF-3.0

nifi_3_0_0_0_453.x86_64           1.2.0.3.0.0.0-453.el6     @HDF-3.0

                                  0.7.0.3.0.0.0-453.el6     @HDF-3.0

registry_3_0_0_0_453.noarch       0.3.0.3.0.0.0-453.el6     @HDF-3.0

storm_3_0_0_0_453.x86_64          1.1.0.3.0.0.0-453.el6     @HDF-3.0

streamline_3_0_0_0_453.x86_64     0.5.0.3.0.0.0-453.el6     @HDF-3.0

zookeeper_3_0_0_0_453.noarch      3.4.6.3.0.0.0-453.el6     @HDF-3.0
```

3. Display the current version associated with each service.

   For example:

```
[root@host registry]# hdf-select status | grep 3.0.0.0-453

nifi - 3.0.0.0-453
registry - 3.0.0.0-453
storm-client - 3.0.0.0-453
storm-nimbus - 3.0.0.0-453
storm-supervisor - 3.0.0.0-453
streamline - 3.0.0.0-453
zookeeper-client - 3.0.0.0-453
zookeeper-server - 3.0.0.0-453
```

4. Install the binaries for each HDF service.

Note

On each Ambari Agent, do this only for the services you have already installed. For example, if only NiFi is installed, only perform the updated install for the NiFi service.

For example:

```
[root@host ~]# yum install -y nifi_3_0_2_0_76*
[root@host ~]# yum install -y registry_3_0_2_0_76*
[root@host ~]# yum install -y streamline_3_0_2_0_76*
```

5. Use `hdf-select` to ensure that you have the appropriate links to the newly installed version.

Note

Installing SAM creates Storm and ZooKeeper dependencies that you can address by falling the Warning notes during the installation.

For example:

```
[root@host ~]# hdf-select set nifi 3.0.2.0-76
[root@host ~]# hdf-select set registry 3.0.2.0-76
[root@host ~]# hdf-select set streamline 3.0.2.0-76
[root@host ~]# hdf-select set storm-nimbus 3.0.2.0-76
[root@host ~]# hdf-select set storm-supervisor 3.0.2.0-76
[root@host ~]# hdf-select set zookeeper-client 3.0.2.0-76

WARNING: Replacing link /usr/bin/zookeeper-client from /usr/hdp/current/
zookeeper-client/bin/zookeeper-client

[root@host ~]# hdf-select set zookeeper-server 3.0.2.0-76

WARNING: Replacing link /usr/bin/zookeeper-server from /usr/hdp/current/
zookeeper-server/bin/zookeeper-server
WARNING: Replacing link /usr/bin/zookeeper-server-cleanup from /usr/hdp/
current/zookeeper-server/bin/zookeeper-server-cleanup
```

6. Confirm your HDF service upgrades.

For example:

```
[root@host ~]# hdf-select status | grep 3.0.2.0-76
nifi - 3.0.2.0-76
registry - 3.0.2.0-76
storm-nimbus - 3.0.2.0-76
storm-supervisor - 3.0.2.0-76
streamline - 3.0.2.0-76
zookeeper-client - 3.0.2.0-76
zookeeper-server - 3.0.2.0-76
```

7. Execute the following operation on one host where SAM is installed:

```
[root@host ~]# export JAVA_HOME=/usr/jdk64/jdk1.8.0_112 ; source
/usr/hdf/current/streamline/conf/streamline-env.sh ;
/usr/hdf/current/streamline/bootstrap/bootstrap-storage.sh create
```

8. Execute the following operation on one host where Schema Registry is installed:

```
[root@host ~]# export JAVA_HOME=/usr/jdk64/jdk1.8.0_112 ; source
/usr/hdf/current/registry/conf/registry-env.sh ;
/usr/hdf/current/registry/bootstrap/bootstrap-storage.sh create
```

Result

You can now log onto Ambari and start your upgraded NiFi, SAM, and Schema Registry services.