Upgrade 3

Ambari Managed HDF Upgrade

Date of Publish: 2019-03-15

https://docs.hortonworks.com/

Contents

Pre-upgrade tasks

Upgrade paths

Before you begin the upgrade process, it is important to understand your upgrade paths.

HDF upgrade paths

- HDF 3.3.x
- HDF 3.2.x

If you are running an earlier HDF version, upgrade to at lease HDF 3.1.0, and then proceed to the HDF 3.3.0 upgrade.

HDP and Ambari versions

- Supported HDP versions – 3.0.x, 3.1.0
- Supported Ambari versions – 2.7.0 - 2.7.3

Warning: We strongly discourage downgrading from Kafka 2.1.0 to earlier versions. Though Ambari allows you to downgrade to earlier versions of Kafka after you upgrade to 2.1.0, consider the following before you do so:

- If you upgrade both the broker and consumer to 2.1.0, a downgrade might cause data loss.
- If you upgrade the broker to 2.1.0 and the consumer remains at 2.0.0, then you can downgrade the broker to 2.0.0 without any issue.

Stop HDF Services

If you are upgrading HDF services installed on an HDP cluster, you must stop HDF services before you upgrade Ambari, HDP, and HDF.

About this task

- This step is necessary whether or not you are upgrading Ambari.
- This step is not necessary if you are upgrading an HDF-only cluster.

Procedure

1. Identify the HDF services you have running. They are:
   - NiFi
   - NiFi Registry
   - Schema Registry
   - Streaming Analytics Manager
2. Stop the service from Ambari Web, by browsing to **Services** > hdf-service-name and select **Stop** from the **Service Actions** menu. Then select **Turn on Maintenance Mode**.

   Replace hdf-service-name with each of the HDF services you have running.

Stop Ambari-Dependent Services

If you are upgrading Ambari, you must stop Ambari Metrics, Log Search, and the Amber Server and Agents before you upgrade.

Procedure

1. 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.
2. 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.
3. Stop the Ambari Server. On the host running Ambari Server:

   ```
   ambari-server stop
   ```
4. Stop all Ambari Agents. On each host in your cluster running an Ambari Agent:

   ```
   ambari-agent stop
   ```

Set Kafka Properties

Before you begin the upgrade process, you need to set the following properties in Kafka (kafka-broker) using Ambari:

About this task

- inter.broker.protocol.version
- log.message.format.version

Procedure

1. Check whether you have set the value of the inter.broker.protocol.version property in Ambari.
   a) If yes, then change the value to your current Kafka version.

      You do not need to change the value, if it is already set to the current Kafka version you are using.
   b) If no, then open /var/log/kafka/server.log, locate the inter.broker.protocol.version property at the end of the log file, and find you current Kafka version. Then go to Ambari and add the inter.broker.protocol.version property with the value you found.
2. Check whether you have already set the value of the log.message.format.version property as previous or current Kafka version in Ambari.
   a) If yes, then do not change anything.
   b) If no, then open /var/log/kafka/server.log, locate the log.message.format.version property at the end of the log file, and find the Kafka version. Then go to Ambari and add the log.message.format.version property with the value you found.

Note: You must use the first two digit of the Kafka version as the value of the above properties. For example, use 2.0, 1.1, or 1.0.

Note: For more information on Kafka 2.1.0 upgrade scenario, see the Kafka 2.1.0 Upgrade Documentation.

Upgrading Only the HDF Management Pack

If you are in an environment that does not require an Ambari upgrade, the first step in upgrading HDF is to upgrade the HDF Management Pack. Follow the available steps for the HDF Management Pack upgrade for an HDF-only cluster, or for HDF Services running on an HDP cluster.

Note:

If you are performing this HDF Management Pack-only upgrade, you may skip the steps for the Ambari and Management Pack upgrade. Both are not necessary

Upgrade the Management Pack for an HDF-only Cluster

To safely upgrade the HDF Management Pack on an HDF-only cluster, you should stop Ambari Server, back up the files and the database, upgrade the Management Pack, and restart Ambari Server.

Procedure

1. Stop the Ambari Server.

```
ambari-server stop
```

2. Backup the Ambari Server files and database.

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

3. Upgrade the HDF Management Pack.

```
ambari-server upgrade-mpack --mpack=<hdf-mpack-url-or-path-location> --
verbose
```

4. Start the Ambari Server.

```
ambari-server start
```

5. In Ambari UI, navigate to **Manage Versions** > **Admin** > **Versions** and create a new version for upgrade. Use the VDF newest release to install new versions.

Upgrade the Management Pack for HDF Services on an HDP Cluster

Procedure

1. Stop NiFi, NiFi Registry, Schema Registry and SAM.
2. Stop the Ambari Server and Ambari Agents.

```
ambari-server stop
ambari-agent stop
```

3. Backup the Ambari Server files and database.

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

4. Download the HDF Management Pack for the version to which you want to upgrade. You can find the Management Pack download locations in the *HDF Release Notes*.
5. Uninstall the existing Management Pack and install the new one.

```
ambari-server uninstall-mpack --mpack-name=hdf-ambari-mpack --verbose
ambari-server install-mpack --mpack=hdf-ambari-mpack-<version>-<build-
number>.tar.gz --verbose
```

6. Start the Ambari Server and Agents.

```
ambari-server start
ambari-agent start
```

7. In the Ambari Console go to **Manage Versions -> Admin -> Versions -> HDP-3.1.0.0 -> HDF-3.4**, and update the HDF repository URL according to the HDF repository information found in the *HDF Release Notes* for the HDF version to which you want to upgrade.
8. On each Ambari Agent, update the base URL in the yum repo file by changing the base URL under [HDF-3.4-repo-1] to the public URL found in the *HDF Release Notes*.

```
vi /etc/yum.repos.d/ambari-hdp-1.repo
```

Upgrade Ambari and the HDF Management Pack

If you are an environment that requires an Ambari upgrade in addition to the HDF upgrade, the first step in upgrading HDF is to upgrade to the latest version of Ambari and to upgrade the HDF management pack.

Note:

If you are performing the Ambari plus HDF Management Pack upgrade, you may skip the steps for the Management Pack-only upgrade. Both are not necessary

Preparing to Upgrade Ambari

- Be sure to review the *Ambari Release Notes* for the Ambari version to which you are upgrading, 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.

The following table lists recommended
( ),
and unsupported (X) upgrade paths.

From / To	Ambari 2.7.x	Ambari 2.6.x
Ambari 2.6.x	✅	X
Ambari 2.5x	X	✅
Ambari 2.4x	X	✅

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.

Get the Ambari Repository

The first step in upgrading Ambari is obtaining the public repositories.

Before you begin

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 is saved with a numeric extension, such as ambari.repo.1. Make sure that the version you copy is the new version.

Procedure

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

Select the repository appropriate for your environment:

- For RHEL/CentOS/Oracle Linux 7:

```
wget -nv http://public-repo-1.hortonworks.com/ambari/centos7/2.x/
updates/2.7.3.0/ambari.repo -O /etc/yum.repos.d/ambari.repo
```

- For Amazon Linux 2:

```
wget -nv http://public-repo-1.hortonworks.com/ambari/amazonlinux2/2.x/
updates/2.7.3.0/ambari.repo -O /etc/yum.repos.d/ambari.repo
```

- For SLES 12:

```
wget -nv http://public-repo-1.hortonworks.com/ambari/sles12/2.x/
updates/2.7.3.0/ambari.repo -O /etc/zypp/repos.d/ambari.repo
```

- For Ubuntu 14:

```
wget -nv http://public-repo-1.hortonworks.com/ambari/ubuntu14/2.x/
updates/2.7.3.0/ambari.list -O /etc/apt/sources.list.d/ambari.list
```

- For Ubuntu 16:

```
wget -nv http://public-repo-1.hortonworks.com/ambari/ubuntu16/2.x/
updates/2.7.3.0/ambari.list -O /etc/apt/sources.list.d/ambari.list
```

- For Debian 9:

```
wget -nv http://public-repo-1.hortonworks.com/ambari/debian9/2.x/
updates/2.7.3.0/ambari.list -O /etc/apt/sources.list.d/ambari.list
```

- For RHEL/CENTOS/Oracle Linux 7 running on IBM Power Systems:

```
wget -nv http://public-repo-1.hortonworks.com/ambari/centos7-ppc/2.x/
updates/2.7.3.0/ambari.repo -O /etc/yum.repos.d/ambari.repo
```

Note:

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

Note:

Ambari Server does not automatically turn off iptables. Check that your installation setup does not depend on iptables being disabled. After upgrading the server, you must either disable iptables manually or make sure that you have appropriate ports available on all cluster hosts. For information on disabling your iptables, see *Configuring iptables* in the *Apache Ambari Installation*.

Related Information

Using a Local Repository

Configuring iptables

Upgrade Ambari Server

You can perform the Ambari Server upgrade on the host running Ambari Server. The upgrade process consists of running the upgrade commands appropriate for your operating system, and verifying the successful upgrade.

Procedure

1. 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 "2.7"

     ```
     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
     ```

     Important:

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

     a. Display the command line UI for YaST, by entering:

        ```
        > yast
        ```

     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. Click **Actions**, choose **Update**, then click the **Enter** button.
     e. Go to **Accept**, and click **enter**.

   - 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
     ```

2. Check for upgrade success by noting progress during the Ambari Server installation process.

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:<version>-<build> Complete!
```

Upgrade the Ambari Agents

Perform the Ambari Agent upgrade on each host running an Ambari Agent. The Agent upgrade process consists of running the upgrade commands appropriate for your operating system and verifying the successful upgrade.

Procedure

1. 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-<version>-<build>.noarch' to install this candidate". You will need to use yast to update the package, as follows:

     a. Display the command line UI for YaST by entering:

        ```
        > yast
        ```

     b. Choose **Software** > **Software Management**, then click **Enter**.
     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. 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
     ```

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

   For RHEL/CentOS/Oracle Linux 7:

   ```
   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 9:

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

Upgrade Ambari Metrics

If you have Ambari Metrics installed, you have some manual steps to perform as part of the upgrade. Upgrading Ambari Metrics involves ensuring that it is in Maintenance Mode, upgrading the Metrics Collector, and upgrading the Ambari Metrics Grafana component.

Procedure

1. 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
   ```

2. 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
   ```

3. 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
   ```

   Note: DO NOT START the Ambari Metrics System service. It will be started automatically during the HDP upgrade process.

Upgrade SmartSense

If you have previously installed SmartSense as part of your Ambari-managed cluster, you must perform some manual steps to upgrade. A SmartSense upgrade includes ensuring that SmartSense is in Maintenance Mode, upgrading binaries on the HST server and agents on every node in your cluster, and upgrading the Ambari service and view.

Procedure

1. Upgrade binaries on the HST server and all HST agents on every node in the cluster, assuming that the Ambari repository is configured on all nodes in the cluster:

   - RHEL or CentOS

     ```
     yum clean all
     yum info smartsense-hst
     ```

     In the info output, visually validate that there is an available version containing "1.5.x”:

     ```
     yum upgrade smartsense-hst
     ```

   - SLES

     ```
     zypper clean
     zypper info smartsense-hst
     ```

     In the info output, visually validate that there is an available version containing "1.5.x":

     ```
     zypper up smartsense-hst
     ```

   - Ubuntu or Debian

     ```
     apt-get clean all
     apt-get update
     apt-cache show smartsense-hst | grep Version
     ```

     In the info output, visually validate that there is an available version containing "1.5.x":

     ```
     apt-get install smartsense-hst
     ```

2. Upgrade Ambari service and Ambari view by running the hst upgrade-ambari-servicecommand as the root user from the machine running the Ambari server. You can run the command in the interactive or non-interactive mode:

   Interactive mode example:

   ```
   # hst upgrade-ambari-service
   Please enter Ambari Server hostname (ambari-server.hortonworks.local):
   Please enter Ambari Server port (8080):
   Please enter Ambari admin user id (admin):
   Please enter password for admin:

   Un-installing old view ...
   Installing new view ...
   Removing deprecated alerts ...
   Updating SmartSense configurations in Ambari ...

   SmartSense service upgrade completed!
   NOTE: It is required to restart Ambari Server for changes to reflect.
    Please restart ambari using 'ambari-server restart'
   ```

   Non-interactive mode example:

   ```
   # hst upgrade-ambari-service -u admin -p 8080 -H ambari-
   server.hortonworks.local -P MySecurePassword123
   Un-installing old view ...
   Installing new view ...
   Removing deprecated alerts ...
   Updating SmartSense configurations in Ambari ...
   SmartSense service upgrade completed!
   NOTE: It is required to restart Ambari Server for changes to reflect.
    Please restart ambari using 'ambari-server restart'
   ```

3. Restart the Ambari server:

   ```
   # ambari-server restart
   ```

4. If you have HST Gateway installed, you need to also upgrade your HST Gateway:
   a) If the HST Gateway is installed on the same node as HST Server or HST Agent, then the HST Gateway will get upgraded along with them.
   b) If the HST Gateway is a standalone node outside of the cluster, perform upgrade steps described in *Upgrade SmartSense Gateway* in the SmartSense documentation.

What to do next

It is very important to make sure you DO NOT START the SmartSense service. It will be started automatically during the Hortonworks stack upgrade process.

Related Information

Upgrade SmartSense gateway

Backup and Upgrade Ambari Infra

The Ambari Infra Solr instance is used to index data for Ranger, and Log Search. The version of Solr used by Ambari Infra in Ambari 2.6 is Solr 5. The version of Solr used by the Ambari Infra in Ambari 2.7 is Solr 7. When moving from Solr 5 to Solr 7 indexed data needs to be backed up from Solr 5, migrated, and restored into Solr 7 as there are on disk format changes, and collection-specific schema changes. The Ambari Infra Solr components must also be upgraded. Fortunately scripts are available to do both, and are explained below.

This process will be broken up into four steps:

Generate Migration Config	The migration utility requires some basic information about your cluster and this step will generate a configuration file that captures that information.
Back up Ambari Infra Solr Data	This process will backup all indexed data either to a node-local disk, shared disk (NFS mount), or HDFS filesystem.
Remove existing collections & Upgrade Binaries	This step will remove the Solr 5 collections, upgrade Ambari Infra to Solr 7, and create the new collections with the upgraded schema required by HDP 3.0 services. This step will also upgrade LogSearch binaries if they are installed.
Migrate & Restore	This step will migrate the backed up data to the new format required by Solr 7 and restore the data into the new collections. This step will be completed after the HDP 3.0 Upgrade has been completed in the Post-upgrade Steps section of the upgrade guide

Generate Migration Config

The utility used in this process is included in the ambari-infra-solr-client package. This package must be upgraded before the utility can be run. To do this:

1. SSH into a host that has a Infra Solr Instance installed on it. You can locate this host by going to the Ambari Web UI and clicking Hosts. Click on the Filter icon and type Infra Solr Instance: All to find each host that has an Infra Solr Instance installed on it.
2. Upgrade the ambari-infra-solr-client package.

```
yum clean all
```

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

3. If you are using a custom username for running Infra Solr, for example a username that is not ‘infra-solr’ additional scripts need to be downloaded. To do this, again only if you are using a custom username for Infra Solr, perform the following steps:

   a.
   ```
   wget --no-check-certificate -O
   /usr/lib/ambari-infra-solr-client/migrationConfigGenerator.py
   https://raw.githubusercontent.com/apache/ambari/trunk/ambari-infra/
   ambari-infra-
   solr-client/src/main/python/migrationConfigGenerator.py
   ```

   b.
   ```
   chmod +x /usr/lib/ambari-infra-solr-client/migrationConfigGenerator.py
   ```

   c.
   ```
   wget --no-check-certificate -O /usr/lib/ambari-infra-solr-client/
   migrationHelper.py
   https://raw.githubusercontent.com/apache/ambari/trunk/ambari-infra/
   ambari-infra-
   solr-client/src/main/python/migrationHelper.py
   ```

   d.
   ```
   chmod +x /usr/lib/ambari-infra-solr-client/migrationHelper.py
   ```

4. You can now proceed to configuring and running the migration tool from the same host.

   Run the following commands as root, or with a user that has sudo access:

Export the variable that will hold the full path and filename of the configuration file.

```
export CONFIG_INI_LOCATION=ambari_solr_migration.ini
```

Ensure the script generates cleanly and there are no yellow warning texts visible. If so, review the yellow warnings.

5. Run the migrationConfigGenerator.py script, located in the /usr/lib/ambari-infra-solr-client/ directory, with the following parameters:

--ini-file $CONFIG_INI_LOCATION	This is the previously exported environmental variable that holds the path and filename of the configuration file that will be generated.
--host ambari.hortonworks.local	This should be the hostname of the Ambari Server.
--port 8080	This is the port of the Ambari Server. If the Ambari Server is configured to use HTTPS, please use the HTTPS port and add the -s parameter to configure HTTPS as the communication protocol.
--cluster cl1	This is the name of the cluster that is being managed by Ambari. To find the name of your cluster, look in the upper right and corner of the Ambari Web UI, just to the left of the background operations and alerts.
--username admin	This is the name of a user that is an “Ambari Admin” .
--password admin	This is the password of the aforementioned user.
--backup-base-path=/my/path	This is the location where the backed up data will be stored. Data will be backed up to this local directory path on each host that is running an Infra Solr instance in the cluster. So, if you have 3 Infra Solr server instances and you use --backup-base-path=/home/solr/backup, this directory will be created on all 3 hosts and the data for that host will be backed up to this path. If you are using a shared file system that is mounted on each Infra Solr instance in the cluster, please use the --shared-drive parameter instead of --backup-base-path. The value of this parameter should be the path to the mounted drive that will be used for the backup. When this option is chosen, a directory will be created in this path for each Ambari Infra Solr instance with the backed up data. For example, if you had an NFS mount /export/solr on each host, you would use --shared-drive=/exports/solr. Only use this option if this path exists and is shared amongst all hosts that are running the Ambari Infra Solr.
--java-home /usr/jdk64/jdk1.8.0_112	This should point to a valid Java 1.8 JDK that is available at the same path on each host in the cluster that is running an Ambari Infra Solr instance.

If the Ranger Audit collection is being stored in HDFS, please add the following parameter, --ranger-hdfs-base-path	The value of this parameter should be set to the path in HDFS where the Solr collection for the Ranger Audit data has been configured to store its data. Example: --ranger-hdfs-base-path=/user/infra-solr

Example Invocations:

If using HTTPS for the Ambari Server:

```
/usr/bin/python /usr/lib/ambari-infra-solr-client/
migrationConfigGenerator.py
              --ini-file $CONFIG_INI_LOCATION --host
 c7401.ambari.apache.org --port 8443 -s
              --cluster cl1 --username admin --password admin --backup-
base-path=/my/path
              --java-home /usr/jdk64/jdk1.8.0_112
```

If using HTTP for the Ambari Server:

```
/usr/bin/python /usr/lib/ambari-infra-solr-client/
migrationConfigGenerator.py
              --ini-file $CONFIG_INI_LOCATION --host
 c7401.ambari.apache.org --port 8080 --cluster
              cl1 --username admin --password admin --backup-base-path=/
my/path --java-home
              /usr/jdk64/jdk1.8.0_112
```

Back up Ambari Infra Solr Data

Once the configuration file has been generated, it's recommended to review the ini file created by the process. There is a configuration section for each collection that was detected. If, for whatever reason, you do not want to backup a specific collection you can set enabled = false and the collection will not be backed up. Ensure that enabled = true is set for all of the collections you do wish to back up. Only the Atlas, and Ranger collections will be backed up. Log Search will not be backed up.

To execute the backup, run the following command from the same host on which you generated the configuration file:

```
# /usr/lib/ambari-infra-solr-client/ambariSolrMigration.sh --ini-file
 $CONFIG_INI_LOCATION
       --mode backup | tee backup_output.txt
```

During this process, the script will generate Ambari tasks that are visible in the Background Operations dialog in the Ambari Server.

Once the process has completed, please retain the output of the script for your records. This output will be helpful when debugging any issues that may occur during the migration process, and the output contains information regarding the number of documents and size of each backed up collection.

Remove Existing Collections & Upgrade Binaries

Once the data base been backed up, the old collections need to be deleted, and the Ambari Infra Solr, and Log Search (if installed) components need to be upgraded. To do all of that, run the following script:

```
# /usr/lib/ambari-infra-solr-client/ambariSolrMigration.sh --ini-file
 $CONFIG_INI_LOCATION
       --mode delete | tee delete_output.txt
```

During this process, the script will generate Ambari tasks that are visible in the Background Operations dialog in the Ambari Server.

Once the process has completed, please retain the output of the script for your records. This output will be helpful when debugging any issues that may occur during the migration process.

Upgrade Ambari Log Search

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

Before you begin

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

Procedure

1. 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
   ```

2. 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
   ```

Upgrade the HDF Management Pack

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 components.

Before you begin

Get the HDF Management Pack location and build number from the *HDF Release Notes*.

Procedure

1. Back up your Ambari resources folder:

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

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

   RHEL/CentOS/Oracle Linux 7:

```
ambari-server upgrade-mpack \
--mpack=http://public-repo-1.hortonworks.com/HDF/centos7/3.x/updates/
<version>/tars/hdf_ambari_mp/hdf-ambari-mpack-<version>-<build-
number>.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/
<version>/tars/hdf_ambari_mp/hdf-ambari-mpack-<version>-<build-
number>.tar.gz \
--verbose
```

   Debian 9:

```
ambari-server upgrade-mpack \
--mpack=http://public-repo-1.hortonworks.com/HDF/debian9/3.x/updates/
<version>/tars/hdf_ambari_mp/hdf-ambari-mpack-<version>-<build-
number>.tar.gz \
--verbose
```

   Ubuntu 14:

```
ambari-server upgrade-mpack \
--mpack=http://public-repo-1.hortonworks.com/HDF/ubuntu14/3.x/updates/
<version>/tars/hdf_ambari_mp/hdf-ambari-mpack-<version>-<build-
number>.tar.gz \
--verbose
```

   Ubuntu 16:

```
ambari-server upgrade-mpack \
--mpack=http://public-repo-1.hortonworks.com/HDF/ubuntu16/3.x/updates/
<version>/tars/hdf_ambari_mp/hdf-ambari-mpack-<version>-<build-
number>.tar.gz \
--verbose
```

Related Information

HDF Release Notes

Upgrade the Ambari Database Schema

Procedure

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

```
ambari-server upgrade
```

2. 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-2.7.1.0.jar to /tmp before proceeding with upgrade.
3. Start the Ambari Server. On the host running Ambari Server:

```
ambari-server start
```

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

```
ambari-agent start
```

5. 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.

Upgrading an HDF Cluster

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 1: Ambari-managed HDF Upgrade Prerequisites

Prerequisite	Description
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.

Prerequisite	Description
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.

Registering Your Target Version

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

Procedure

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 register, specify whether it will be a public or private repository, and select your operating system.
6. Click **Save**.

Results

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

Installing Your Target Version

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

Procedure

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**.

Results

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.

Upgrade HDF

Upgrading HDF installs your target software version onto each node in your cluster. You can perform either an express or a rolling upgrade.

About this task

Rolling Upgrade is not supported by NiFi. When you select the Rolling Upgrade option for the HDF stack, NiFi stops all services, and performs an Express Upgrade. Express Upgrades stops the NiFi Service completely, and restarts it with the new version installed.

Before you begin

Turn off maintenance mode for LogSearch and Ambari Metrics if required action appears when attempting to upgrade stack.

Procedure

1. From **Admin | Stack and Versions | Versions**, click **Upgrade**.
2. In the **Upgrade Options** pop-up window, click either **Express Upgrade** or **Rolling 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 Services have been upgraded Checks, 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, you are 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**.

Results

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

Start Ambari LogSearch and Metrics

After you have upgraded your HDF cluster, you should ensure that Ambari LogSearch and Metrics are both running.

Procedure

1. From the Ambari UI, verify whether Ambari LogSearch and Metrics are running.
2. If they are not running, manually start them before proceeding.

Upgrading HDF services on an HDP cluster

Upgrade HDP

You can upgrade to HDP 3.0.x and HDP 3.1.0. Upgrading your HDP cluster involves registering your new HDP version, updating base URLs, installing the new HDP packages, and then performing either an express or rolling upgrade.

Before you begin

You have obtained the necessary HDP, HDP UTILs, and HDF base URLs from the *HDF Release Notes*.

Procedure

1. In Ambari, go to **Manage Versions** and register your new HDP version.
2. Select the HDP 3.0.x version to which you want to upgrade, and update the base URLs for HDP, HDF, and HDP UTILs.
3. Install the new packages from the registered version of HDP.
4. Perform an upgrade to your latest version of HDP as described in the HDP upgrade documentation.

Related Information
Upgrading HDP
HDF Release Notes

Upgrade HDF services

Ensure that you complete these steps on each node that contains an HDF service.

Before you begin
Ensure that you have backed up your SAM and Schema Registry databases.

Procedure

1. Confirm HDF components installed along with version (e.g. 3.2.0.0-520).

   For example:

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

hdf-select.noarch                 3.2.0.0-520.el6           @HDF-3.2

nifi_3_2_0_0_520.x86_64           1.7.0.3.2.0.0-520.el6     @HDF-3.2

                                  0.7.0.3.2.0.0-520.el6     @HDF-3.2

registry_3_2_0_0_520.noarch       0.5.3.3.2.0.0-520.el6     @HDF-3.2

storm__3_0_2_0_76.x86_64          1.2.1.3.2.0.0-520.el6     @HDF-3.2

streamline_3_0_2_0_76.x86_64      0.6.0.3.2.0.0-520.el6     @HDF-3.2

zookeeper_3_0_2_0_76.noarch       3.4.6.3.2.0.0-520.el6     @HDF-3.2
```

2. Display the current version associated with each component.

```
[root@host registry]# hdf-select status | grep 3.2.0.0-520

nifi - 3.2.0.0-520
registry - 3.2.0.0-520
storm-client - 3.2.0.0-520
storm-nimbus - 3.2.0.0-520
storm-supervisor - 3.2.0.0-520
streamline - 3.2.0.0-520
zookeeper-client - 3.2.0.0-520
zookeeper-server - 3.2.0.0-520
```

3. Install binaries for the HDF services to which you want to upgrade.

   See the *HDF Release Notes* for the repository information, including service version and build number.

```
[root@host ~]# yum install -y <service>_<version>_<build-number>*
```

   For example:

```
[root@host ~]# yum install -y nifi_3_4_0_0_155*
```

Note: Only run the yum install command on Ambari Agent hosts where NiFi is already installed.

4. Use hdf-select to ensure appropriate links to new installed version. Note that SAM also brings down Storm and ZooKeeper dependencies that also needs to be accounted for.

   For example,

```
[root@host ~]# hdf-select set nifi 3.4.0.0-155
[root@host ~]# hdf-select set registry 3.4.0.0-155
[root@host ~]# hdf-select set streamline 3.4.0.0-155
[root@host ~]# hdf-select set storm-nimbus 3.4.0.0-155
[root@host ~]# hdf-select set storm-supervisor 3.4.0.0-155
[root@host ~]# hdf-select set zookeeper-client 3.4.0.0-155

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

[root@host ~]# hdf-select set zookeeper-server 3.4.0.0-155

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
```

5. Confirm that hdf-select shows new version for the HDF components that were updated.

```
[root@host ~]# hdf-select status | grep 3.4.0.0-155
nifi - 3.4.0.0-155
registry - 3.4.0.0-155
storm-nimbus - 3.4.0.0-155
storm-supervisor - 3.4.0.0-155
streamline - 3.4.0.0-155
zookeeper-client - 3.4.0.0-155
zookeeper-server - 3.4.0.0-155
```

6. Log into Ambari and start NiFi, NiFi Registry, SAM and Schema Registry. Confirm all applications started and pre-existing flows, topologies, and configurations are available.

Post-Upgrade Tasks

Check the NiFi Toolkit Symlink

In some upgrade scenarios the NiFi Toolkit may not be updated. After the installation you should check the NiFi Toolkit symlink and update it if necessary.

Procedure

1. Navigate to the nodes where the NiFi, NiFi Registry, and NiFi Toolkit Certificate Authority services installed.
2. Check that the NiFi Toolkit symlink is pointing to the latest HDF version. On each node where the NiFi Toolkit is installed, run the following command:

```
ls -al /usr/hdf/current
```

3. If the NiFi Toolkit symlink is pointing to an older version, update it by entering:

```
hdf-select set nifi-toolkit [**new-version**]
```

For example:

```
hdf-select set nifi-toolkit 3.5.2.0-99
```

Update NiFi Properties

If you have upgraded to HDF 3.4.x, you must manually update one NiFi property.

Procedure

1. Navigate to the nifi.properties file.
2. Set the value of nifi.nar.library.autoload.directory to:

```
{{nifi_internal_dir}}/work/extensions
```

Review Storm Configurations

If you have configured STORM_EXT_CLASSPATH or STORM_EXT_CLASSPATH_DAEMON prior to upgrade, you must update the values to add a wild card.

Procedure

1. From the left-hand services navigation pane, select **Storm | Configs | Advance**
2. Update the STORM_EXT_CLASSPATH or STORM_EXT_CLASSPATH_DAEMON values to include a wild card.

Example

If STORM_EXT_CLASSPATH=/foo/bar/lib, update the value to STORM_EXT_CLASSPATH=/foo/bar/lib/*.

Verify Kafka Properties

After the upgrade is complete, verify the following properties in Kafka (kafka-broker) using Ambari:

About this task

- inter.broker.protocol.version
- log.message.format.version

Procedure

1. Go to Ambari and locate the inter.broker.protocol.version and log.message.format.version properties.
2. Verify the values of the properties.
   - inter.broker.protocol.version must be set to the latest Kafka version. For example, inter.broker.protocol.version = 2.1
   - log.message.format.version must be set to the version you set before upgrade. For example, log.message.format.version = 2.0 or 1.1 or 1.0.
3. If you find any performance degradation, perform proper evaluation and set the value of the log.message.format.version property to the latest Kafka version, which is 2.1, and perform rolling restart for your kafka-broker.