



**Cloudera Professional Services**

HDP 2.4.2 to HDP 2.6.5 RDA Cluster Upgrade Document for Standard Chartered Bank

**Nabeel Moidu, Solutions Architect**

**Version 1.0, 1 January 2020**

|  |  |  |  |
| --- | --- | --- | --- |
| **Revision History** | | | |
| Version | Author | Description | Date |
| 1.0 | Nabeel Moidu | Initial Version | 15/01/2020 |

TABLE OF CONTENTS

[1. Introduction 1](#_Toc30247592)

[1.1. Overview of the Upgrade Process 1](#_Toc30247593)

[1.1.1. Express Upgrade Steps Explained 1](#_Toc30247594)

[2. Detailed Upgrade Steps 1](#_Toc30247595)

[2.1. Decommission any unused nodes 1](#_Toc30247596)

[2.2. Pre-upgrade Verification 1](#_Toc30247597)

[2.2.1. Verify Your Environment 1](#_Toc30247598)

[2.2.2. Verify the Status of the HDP Cluster 1](#_Toc30247599)

[2.3. Upgrading HDP 1](#_Toc30247600)

[2.3.1. Preparing to Upgrade HDP 1](#_Toc30247601)

[2.3.1.1. Create HDFS Checkpoint 1](#_Toc30247602)

[2.3.1.2. Preparing for Yarn state recovery 1](#_Toc30247603)

[2.3.1.3. Preparing for Upgrade to Kafka 1.0 1](#_Toc30247604)

[2.3.1.4. Preparing for MySQL version related configuration changes 1](#_Toc30247605)

[2.3.1.5. Register and Install HDP 2.6.5 Packages 1](#_Toc30247606)

[2.3.2. Upgrade HDP from 2.4.2.0 to HDP 2.6.5.212 1](#_Toc30247607)

[2.3.2.1. Modifications to the Environment 1](#_Toc30247608)

[3. Post-Upgrade, Pre-Finalize Tests 1](#_Toc30247609)

[3.1. HDFS 1](#_Toc30247610)

[3.2. Yarn MapReduce 1](#_Toc30247611)

[3.3. HBase 1](#_Toc30247612)

[3.4. Phoenix 1](#_Toc30247613)

[3.5. Application Checks 1](#_Toc30247614)

[4. Post Finalize Tasks 1](#_Toc30247615)

[4.1. General Post-upgrade Tasks 1](#_Toc30247616)

[4.2. Kafka Post-upgrade Tasks 1](#_Toc30247617)

[4.3. Ranger Post-upgrade Tasks 1](#_Toc30247618)

[4.4. HBase Post-upgrade Tasks 1](#_Toc30247619)

[4.5. 1](#_Toc30247620)

[4.6. Phoenix Hot-Fix Installation (only if issue is observed) 1](#_Toc30247621)

[Appendix 1 - QA Cluster HDP Upgrade Report from Ambari 1](#_Toc30247622)

[Appendix 2 – Troubleshooting 1](#_Toc30247623)

# Introduction

The Retail Data Analytics Division within Standard Chartered Bank approached Cloudera Inc. with a request to provide technical resources to plan and execute an HDP cluster upgrade for three environments: Performance Testing (PT), Disaster Recovery (DR) and Production (Prod) clusters. The goal of the current engagement is to ensure a smooth version transition for the Production platform with minimal side-effects.

The latest version of HDP product series that does not involve breaking changes to existing 2.4.2 version deployed in RDA is HD 2.6.5. A hotfix containing earlier bugfixes provided to SCB HaaS cluster was delivered to RDA upon special request by PS to avoid known pain points in the upgraded version. The target version thus set for the cluster upgrade is HDP 2.6.5.212-8.

The document’s scope is Cloudera software, the steps needed for application & data changes and moves are contained in a separate SCB AIG document.

## Overview of the Upgrade Process

HDP cluster upgrades generally consist of two major steps:

1. Upgrading the Ambari version
2. Upgrading the HDP version

This document assumes an earlier upgrade activity of Ambari software from 2.2.2.2 to 2.6.2.2 has been completed as per the document provided prior to this activity.

HDP version upgrade supports two upgrade modes:

* Express Upgrade and
* Rolling Upgrade.

Express upgrade requires a full stop of the cluster while all nodes and services are upgraded at the same time. The configuration changes are done at once and from the outside world the cluster changes over night. This is a faster approach and generally recommended but requires a maintenance downtime on all services directly dependent on the Hadoop cluster.

Rolling upgrade, on the other hand, does the software version change in a service-by-service, node-by-node fashion. The worker nodes are gradually migrated to a newer version and the cluster services are available during the whole process. But the whole process takes a significantly longer time and is more complicated to execute. Not all cluster services fully support rolling upgrades. The process is also more sensitive to errors and the chance of unforeseen complications is higher.

For simplification of the process, lessening of risk and faster completion of the upgrade, we choose to go ahead with the Express Upgrade option.

### Express Upgrade Steps Explained

The upgrade plan assumes the following:

* Java upgrade from the current version to openjdk 1.8.0-212 has been completed prior to this upgrade
* MySQL upgrade to 5.7 from current version 5.1 will be handled during the upgrade window but prior to upgrading the HDP software.
* All jobs in the cluster will be terminated prior to upgrade.
* All unused services can be removed from cluster

# Detailed Upgrade Steps

This section contains a detailed list of tasks and steps to do the Ambari and HDP.

**Note:** It is advisable to have password-less ssh enabled on all cluster nodes for the duration of the upgrade maintenance window so that the necessary commands to be run on all nodes can be executed in parallel. This significantly reduces the scope for human error during the upgrade. This can allow use of pre-written scripts combining steps that can be executed without dependence on any other activity steps and speed up the execution of the process.

## Decommission any unused nodes

If any node in the cluster is currently not being used, remove the nodes from the HDP cluster. They can be added back later if required once the upgrade is completed. This can avoid unnecessary failures in service checks etc during the upgrade process.

The steps to decommission a node is as below:

* For **each** node to be decommissioned: Ambari Web UI > Hosts > click on host.
  + Stop all the services. Host Actions > Stop All Components
  + Stop the ambari-agent. From the command line: sudo service ambari-agent stop
  + Delete the host: Host Actions > Delete Host
* Cleanup the HDP/Ambari files from the decommissioned hosts:

# remove HDP packages

yum remove `yum list installed | grep "@HDP-2.4" | awk '{ print $1 }'`

# remove HDP-Utils

yum remove `yum list installed | grep "@HDP-UTILS-" | awk '{ print $1 }'`

# clean up any remaining directories but leave the users

python /usr/lib/python2.6/site-packages/ambari\_agent/HostCleanup.py -s -k users

# finally remove the ambari-agent

yum remove `yum list installed | grep -I "@Ambari-2.6.2.2" | awk '{ print $1 }'`

# and delete ambari leftovers

rm -rf /etc/ambari-\*

rm -rf /var/lib/ambari-\*

rm -rf /usr/share/hdp/\*

rm -rf /etc/hive2

rm -rf /var/lib/hive2

rm -rf /etc/tez-hive2

rm -rf /usr/lib/ambari-\*

rm -rf /var/log/ambari-\*

rm -rf /var/log/hive2

rm -rf /var/log/hst

## Pre-upgrade Verification

The goal of the steps listed here is to make sure that the cluster environment is ready for an upgrade. Make sure that all steps are executed and any irregularities fixed!

### Verify Your Environment

* Make sure that installed OS and JDK versions are supported by both new and old HDP releases[[1]](#footnote-1). On **each host** execute:

cat /etc/redhat-release

and check the java.home property in the /etc/ambari-server/conf/ambari.properties file.

**Note:** The Java version to be used for the upgrade is openjdk 1.8.0\_212, which is supported by HDP 2.6.5

* Make sure that OS package repositories are properly setup and available on all cluster nodes for the target versions of HDP 2.6.5.

**Note:** We used yum repolist | egrep -i "2.6.5|utils" to check for the HDP 2.6.5 and HDP Utils

HDP-2.6

HDP-UTILS 64

We also checked that the proper packages and versions are really available inside the repo with:

yum --disablerepo="\*" --enablerepo="HDP-2.6" list available

* Make sure that the metastore DB is supported[[2]](#footnote-2).

**Note:** RDA is using MySQL 5.1, which is expected to be upgraded to MySQL 5.7 either prior to or during the upgrade window, but prior to platform upgrade. MySQL 5.7 is supported by HDP 2.6.5.

* Validate that there are no broken symlinks to the configurations and that you are using the precise HDP version, from which you intend to upgrade. Be sure to check in the /etc/hadoop/conf directory (ls -l /etc/hadoop/conf).
* **B**e sure to have enough space on **/usr/hdp** (10 GB and above), using below can find required space availability on all cluster nodes

for i in `awk ' {if($NF ~ "HKLPASRAD"){print $NF}}' /etc/hosts`; do echo -n $i" "; ssh $i " df -h /usr/hdp 2>/dev/null |tail -1" ;done

* **B**e sure also to have enough space on **/tmp** , using below can find required space availability on all cluster nodes

for i in `awk ' {if($NF ~ "HKLPASRAD"){print $NF}}' /etc/hosts`; do echo -n $i" "; ssh $i " df -h /usr/hdp 2>/dev/null |tail -1" ;done

* Clear the logs (/var/log), especially if the logs are on a mount that is low on space and shared with the operating system
* Identify dead nodes and exclude or decommission them.
* Remove any hosts in an abnormal state from the cluster. You can add them again after the upgrade.
* Identify any special \*-env.sh setting such as special bootstrap libraries. Document them in case you need to reapply them after the upgrade.
* Identify corrupt blocks and either repair or make note of them (hdfs fsck).
* Validate that you have access to each of the backing DB metastores used by HDP. These include, but are not limited to Ambari, Hive, Ranger. Each component has an upgrade process, some of which are fairly automatic. Ensure the connected user to these databases have sufficient privileges to make schema changes as required by the upgrade.
* Confirm that Ambari recognizes all the components that are installed in the cluster. If not, redeploy the component by using Ambari.
* Ambari will not proceed with the upgrade if any components are in “maintenance” mode. Take time now to get those services up, healthy and out of maintenance mode.
* Storm, Knox services on the cluster are unused and can be removed from the cluster. Remove unwanted services prior to starting the upgrade. This can save time spent in upgrade installing newer versions of the packages and also running service checks pre and post upgrade on these components.

### Verify the Status of the HDP Cluster

* (Optional) Clean up old snapshots on HDFS to reclaim as much space before the upgrade as you can.
* Run HDFS Balancer to ensure that data distribution is done.
* Confirm that the current state of HDFS NameNode does not indicate that there is a previous upgrade still in progress.
* Confirm that all the services are running and indicate a healthy state.
* Confirm that there are no open Ambari alerts.
* Take note of any configuration differences between nodes.
* Confirm that the databases for Apache Oozie, Ambari, Hive Metastore and Ranger have been backed up.
* Run each Service Check (found under the Service Actions menu) and confirm they execute successfully.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a cell phone

Description automatically generated

* (Optional) 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.

**PT:**

**STOP: 3-4 minutes**

**START: 18-19 minutes**

**DR:**

**STOP: 2-3 minutes**

**START: 30 minutes**

## Upgrading HDP

### Preparing to Upgrade HDP

* Make sure Ambari and the cluster are healthy and operating normally.
* Make sure that all services are started.
* Make sure that no service is in Maintenance Mode.
* All Ambari Agents must be communicating and heartbeating to Ambari Server. Any hosts that are not heartbeating must be in Maintenance Mode.
* 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.
* Make sure that all service checks are passing. Run **Service Actions > Run Service Check** on all services (and remediate if necessary) prior to attempting an HDP upgrade.
* Take a blueprint of the cluster configuration

curl -H "X-Requested-By: ambari" -u admin http:// 10.21.216.72:8080/api/v1/clusters/SCBHaaSPROD?format=blueprint -o /opt/HDP2.4\_\_backup/pre\_upgrade\_blueprint.json

* Make sure that upgraded Ambari and all metastore databases have been backed up.
* Upgrade libtirpc-devel package on all nodes in cluster to avoid error below :

A screenshot of a social media post

Description automatically generated

wget http://10.23.139.15/libtirpc-devel-0.2.1-15.el6.x86\_64.rpm

sudo rpm -ivh libtirpc-devel-0.2.1-15.el6.x86\_64.rpm

A screenshot of a cell phone

Description automatically generated

#### Create HDFS Checkpoint

* Perform the following steps on the Active NameNode host. You can locate the Active NameNode from **Ambari Web > Services > HDFS** in the Summary area.
* Check the NameNode directory to ensure that there is no snapshot of any prior HDFS upgrade.
  + In Ambari Web UI, browse to **Services > HDFS > Configs**, and examine the dfs.namenode.name.dir in the NameNode Directories property.

**Note:** This should be /hadoop/hdfs/namenode

* + Make sure that only a /current directory and no /previous directory exists in the previously found directory (dfs.namenode.name.dir) on the Active NameNode host.
* Check HDFS filesystem and fix any eventual errors. It’s a good idea to save output of these commands and any fixes applied.

hdfs fsck /

hdfs fsck / -list-corruptfileblocks

hdfs dfsadmin -report

* Save the namespace. As the HDFS user you must put the cluster in Safe Mode (we’re explicitly setting the NameNode to make sure the save is stored on it).:

su -l hdfs

hdfs dfsadmin -safemode enter

hdfs dfsadmin -fs hdfs://<active\_namenode>:8020 -saveNamespace

A close up of a screen

Description automatically generated

* Copy the checkpoint files located in /hadoop/hdfs/namenode/current (${dfs.namenode.name.dir}/current) into a backup directory (execute on the Active NameNode host).

cp -R /hadoop/hdfs/namenode/current /opt/HDP-2.4\_backup/namespace-backup

* As the HDFS user, take the NameNode out of Safe Mode:

hdfs dfsadmin -safemode leave

* Finalize any prior HDFS upgrade, if you have not done so already. Run the following as the HDFS user:

hdfs dfsadmin -finalizeUpgrade

A close up of a logo

Description automatically generated

#### Preparing for Yarn state recovery

**1 YARN** **Timeline Service State Recovery**

YARN state recovery should be enabled. Check the **Services > YARN > Configs > Advanced** property yarn.timeline-service.recovery.enabled is set to true.

A screenshot of a cell phone

Description automatically generated

**2 ResourceManager Work Preserving Recovery**

YARN work preserving recovery should be enabled. Check the **Services > YARN > Configs > Advanced** property yarn.resourcemanager.work-preserving-recovery.enabled is set to true

A screenshot of a cell phone

Description automatically generated

**3 MapReduce2** - **MapReduce Distributed Cache**

MapReduce should reference Hadoop libraries from the distributed cache in HDFS.

**JobHistory State Recovery**

JobHistory state recovery should be enabled. Check the following:

**Services > MapReduce > Configs > Advanced** property **mapreduce.jobhistory.recovery.enable** is set to true. Also set values for **mapreduce.jobhistory.recovery.store.class**  and **mapreduce.jobhistory.recovery.store.leveldb.path**

A screenshot of a cell phone

Description automatically generated

#### Preparing for Upgrade to Kafka 1.0

This step prepares the Kafka service for version upgrade from Kafka 0.x to Kafka 1.0. HDP 2.4.2 is shipped with Kafka 0.9.0.

* Stop Kafka Service. **Ambari Web UI > Services > Kafka** and select **Stop** from the Service Actions menu.
* In **Custom kafka-broker** section in Ambari Configs, click **Add Property** and create inter.broker.protocol.version and log.message.format.version with the following values and click **Save**.

Key: inter.broker.protocol.version

Value: 0.9.0

Key: log.message.format.version

Value: 0.9.0

* Due to “Additional Upgrade Guidance” in the Kafka upgrade documentation[[3]](#footnote-3), we will explicitly enable the unclean leader election. In **Custom kafka-broker** section in Ambari Configs, click **Add Property** and unclean.leader.election.enable with the following value and click **Save**.

**Note:**

Key: unclean.leader.election.enable

Value: false

* Do a Kafka rolling restart: **Ambari Web UI > Services > Kafka** and select **Rolling restart** from the Service Actions menu.
* Then change the setting back to true. After this do another Rolling restart of the Kafka service.

#### Preparing for MySQL version related configuration changes

This step prepares the MySQL service for version upgrade from MySQL 5.1 to 5.7

* Convert all HDP related table engines from MyISAM to InnoDB
* Edit my.cnf file to add skip\_ssl

[mysqld]

skip\_ssl

#### Register and Install HDP 2.6.5 Packages

* Log in to Ambari Web UI and browse to **Admin > Stack and Versions.**
* Click the **Versions** tab. You should see HDP 2.4.2.0 as the currently running version (marked as **Current)**.
* Click **Manage Versions**. Proceed to register a new version by clicking **+ Register Version**.
* Click the HDP-2.6 main version branch and then from the drop down list select HDP-2.6.5.0.
* Choose the **Use Local Repository** option.

A screenshot of a cell phone

Description automatically generated

* Click **Save** and return to the dashboard by clicking on **Go To Dashboard**.
* Browse again to **Admin > Stack and Versions**. Click the **Versions** tab. You should see HDP 2.4.2.0 as the currently running version (marked as **Current)** and HDP 2.6.5.0 with an **Install** button.

A screenshot of a computer

Description automatically generated

* On the HDP 2.6.5.0, click on the **Install** button click **OK** to confirm. The installation process starts and distributes the binaries across the cluster.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

* When the install process completes, the **Upgrade** button replaces the **Install** button.

A screenshot of a computer

Description automatically generated

* There is a change starting with HDP 2.6.3 in how the Ranger backing database is organized and some double entries may be possible in earlier versions. We check with the following query and saw that there is no need to do the Ranger DB deduplication process from the following link:  
  <https://docs.hortonworks.com/HDPDocuments/Ambari-2.6.2.2/bk_ambari-upgrade/content/upgrade_remove_duplicate_ranger_entries.html>

**Note:** We DR Ranger DB. All entries showed a count of 1.

select group\_name, count(1) duplicateCount from x\_group group by group\_name order by 2 desc;

### Upgrade HDP from 2.4.2.0 to HDP 2.6.5.212

* Disable Auto Start of services. Navigate to **Admin > Service Auto Start**. Turn the toggle switch off to **Disabled** and hit **Save**.

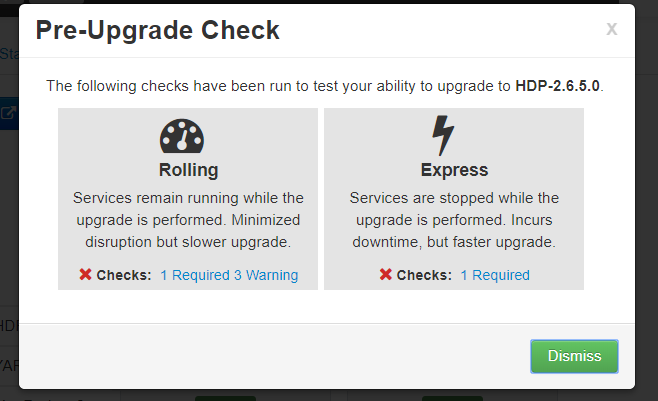
A screenshot of a social media post

Description automatically generated

A screenshot of a cell phone

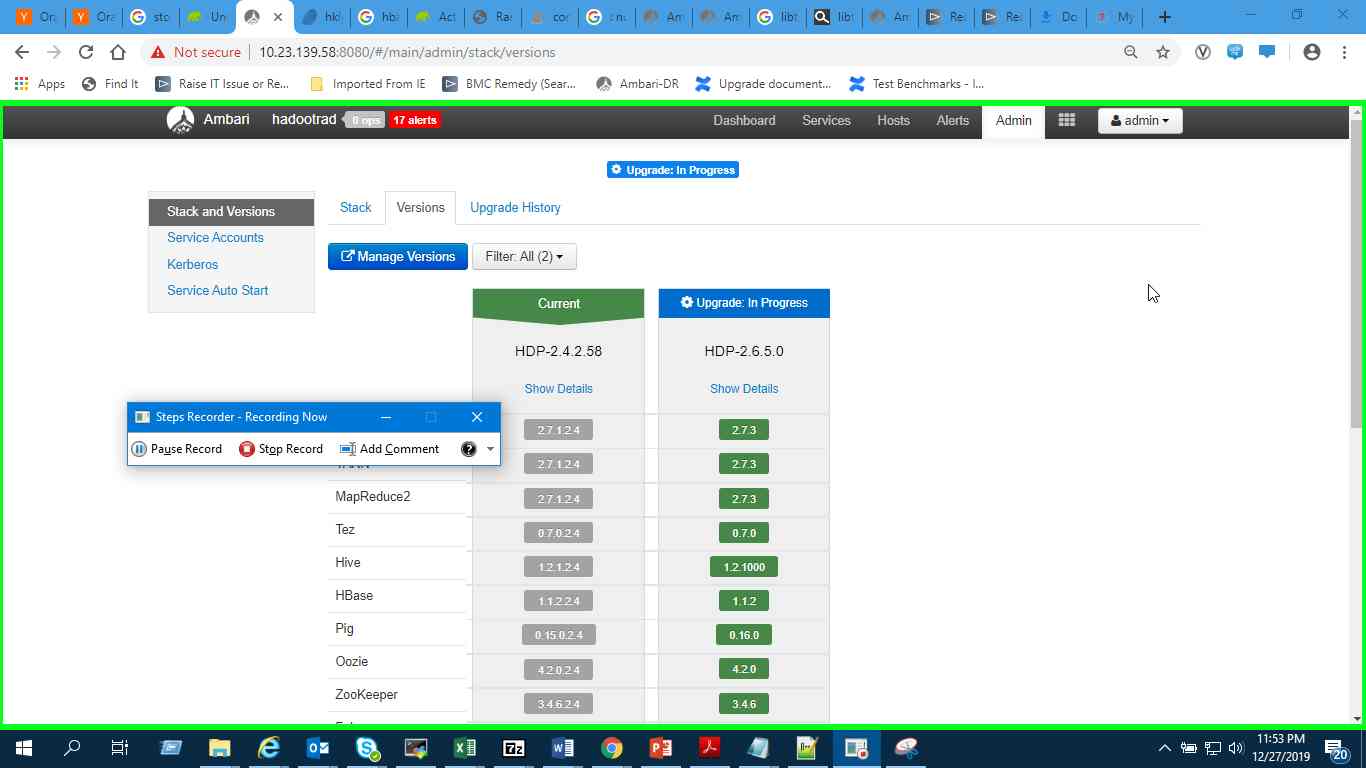
Description automatically generated

* Browse again to **Admin > Stack and Versions** in the Ambari Web UI.
* Click the **Versions** tab. You should see HDP 2.4.2.0 as the currently running version (marked as **Current)** and HDP 2.6.5.0 with a **Perform Upgrade** button.
* A number of prerequisite checks are done. Hopefully by using this guide all should be satisfied but in case there is a problem, follow the instructions from Ambari.
* Select the **Express Upgrade**. None of the advanced options should be needed. Click **Proceed**.



(example of a failed prerequisites check)

* 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.
* When the prerequisite checks successfully completes, 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 Upgrade | You should stop all YARN queues, all long-running applications on Slider, and deactivate & kill all running Storm topologies. | Perform the actions to prepare for the upgrade.  e.g. su – yarn -c “yarn applications -list | cut -f1 | grep ^application | grep -v grep | xargs -n1 yarn application -kill “ |
| Stop Components for High-Level Services | This will stop all components for High-Level Services. This includes all master components **except** 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. This step is done once before. If the upgrade is happening on a different day than the backup, please take another backup.  Name node fsimage backup and checkpointing should be done at this stage |
| Stop Components for Core Service | Stops all components with HDFS, HBase, Zookeeper and Ranger. | None |
| Update Target Stack | Updates the stack version in Ambari to the target version. There is no downgrade past this point. | 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). | None |
| All Service Checks | All Service Checks are performed against the cluster. | Any service check that fails prompts you to **Ignore and Continue**, **Downgrade** or **Retry**. If you selected the **Skip all Service Check failures** option, you will only be prompted when all Service Checks complete. |
| Restart Components | Restarts the remaining components such as Oozie, Falcon, Hive, Spark 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 **# hosts** for details on the hosts (and their components) that are not upgraded. You can **Pause Upgrade**, 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. (Click **FINALIZE LATER**) |

* Prompts as below should be acknowledged and proceeded as we have backed up all databases before the start of the upgrade:
  + *Ignore the version number in the screenshot. The rest of the screen remains the same here:*

A screenshot of a cell phone

Description automatically generated

A screenshot of a cell phone

Description automatically generated

A screenshot of a cell phone

Description automatically generated

A screenshot of a cell phone

Description automatically generated

A screenshot of a social media post

Description automatically generated

A screenshot of a cell phone

Description automatically generated

* Issues during upgrade:
  + ClassNotFoundException in hbase as below:

A screenshot of a cell phone

Description automatically generated

Resolution: Change the parameter below after pausing the upgrade:

A screenshot of a cell phone

Description automatically generated

A screenshot of a cell phone

Description automatically generated

* + OutOfMemory error in HBase:

A screenshot of a cell phone

Description automatically generated

Resolution: Change the parameter below after pausing the upgrade:

A screenshot of a cell phone

Description automatically generated

A close up of a logo

Description automatically generated

A picture containing screenshot

Description automatically generated

* Falcon errors:

A close up of a screen

Description automatically generated

Resolution:

* + Remove org.apache.falcon.metadata.MetadataMappingService{{atlas\_application\_class\_addition}} from falcon-startup.proerties

* Coprocessor launch error in HBase :

A screenshot of a cell phone

Description automatically generated

Resolution: Set the property below in in HBase config

A screenshot of a cell phone

Description automatically generated

A picture containing screenshot

Description automatically generated

* Test Hive service check was still failing but we manually executed the failing script and it succeeded. Therefore we ignored the error and proceeded with the upgrade.
* On Prod we had the same problem as above. It seems that HCat smoke tests are slow when executed from edge nodes. QA has one HCat client on a master node, Prod doesn’t have any, hence the test is always failing due to python script running for longer than 300 seconds, which is the running time timeout value. We will increase the timeout value according to KB article[[4]](#footnote-4) (it’s about Oozie but same setting applies to Hive):

To increase the timeout, find and update the timeout set in the metainfo.xml file located in Edit /var/lib/ambari-server/resources/common-services/HIVE/your\_version\_number/metainfo.xml,

Example:

<commandScript>

<script>scripts/service\_check.py</script>

<scriptType>PYTHON</scriptType>

<timeout>300</timeout>

</commandScript>

Change the timeout to a larger value

<commandScript>

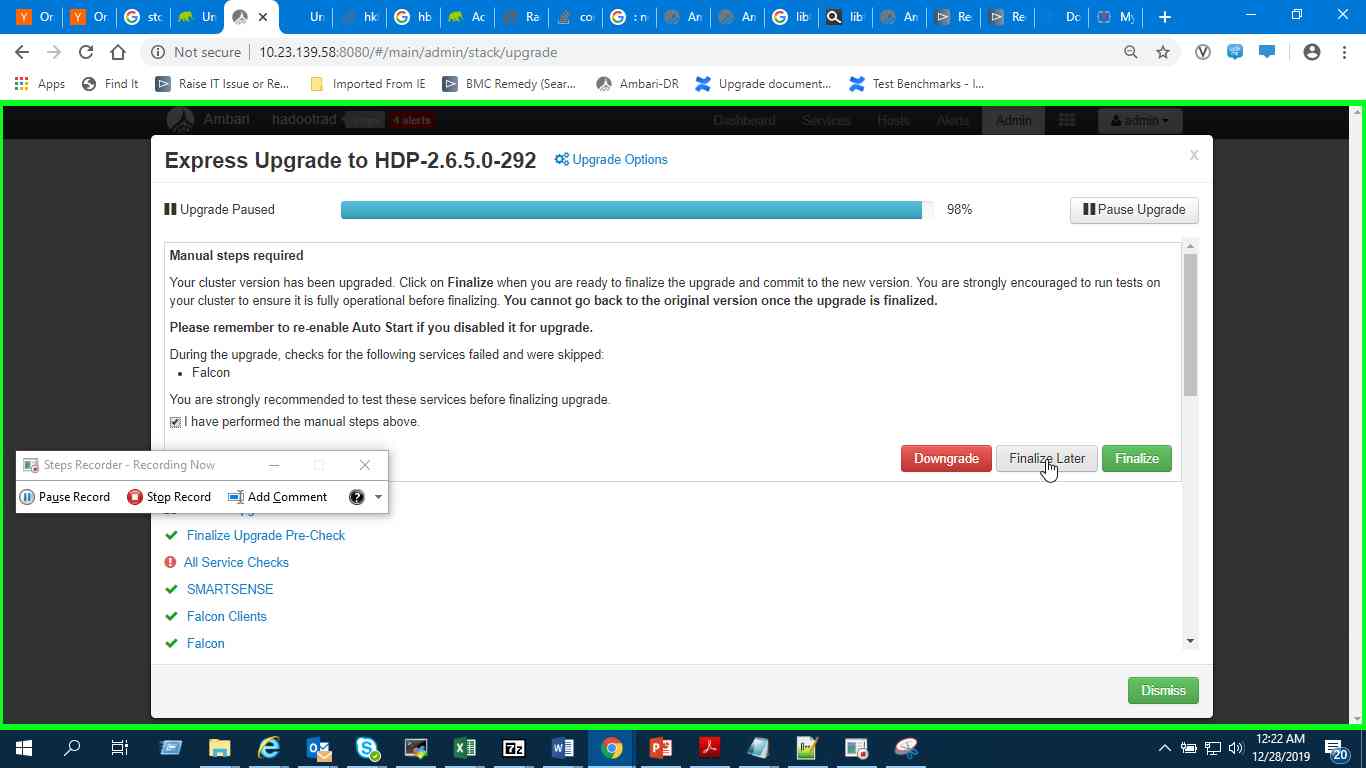
<script>scripts/service\_check.py</script>

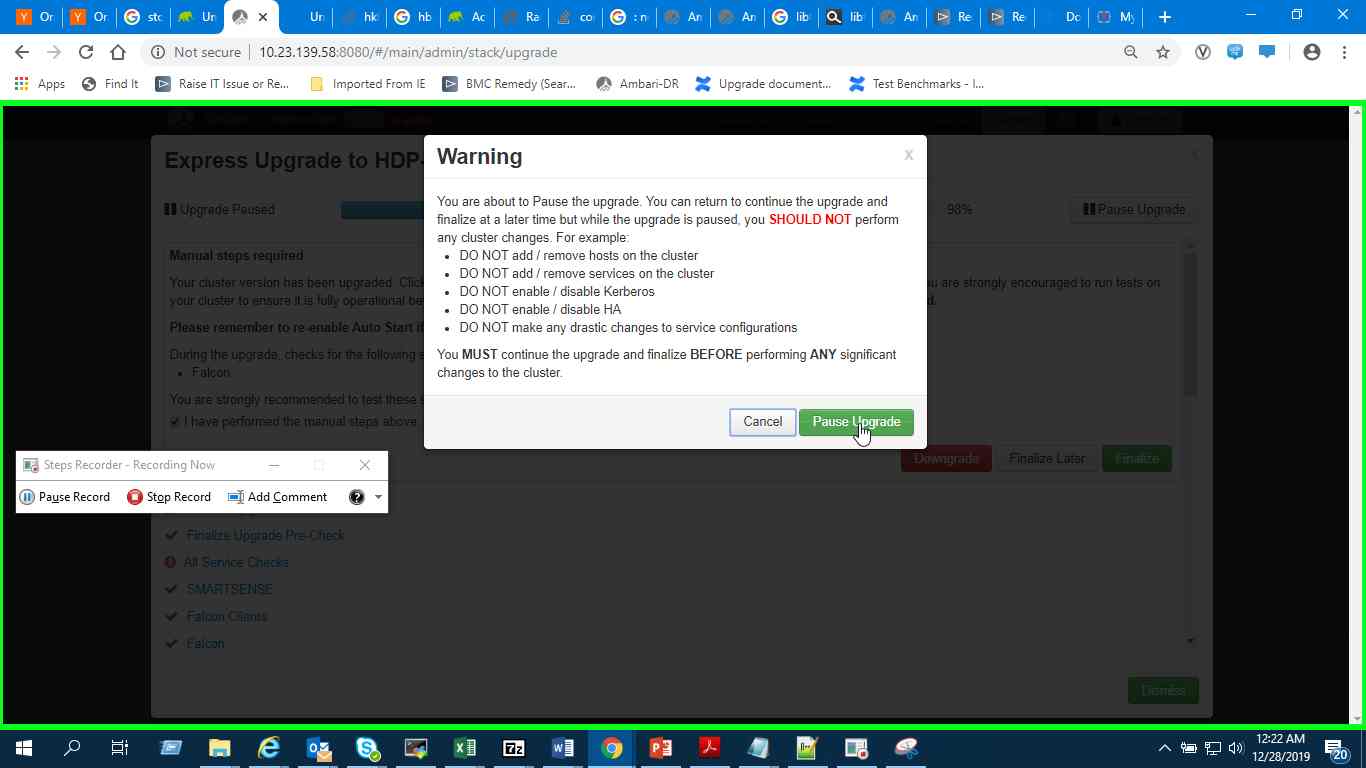
<scriptType>PYTHON</scriptType>

<timeout>600</timeout>

</commandScript>

* This didn’t completely solve the issue as the WebHCat port is not open from the edge nodes. Port 50111 needs to be open from all edge nodes to all master nodes. Until then, every Hive service check that runs from an edge node will fail. Click **Ignore and Proceed** when the Hive service check fails.
* When the Express upgrade stages complete, click the **Finalize Later**.





(*Ignore the version number in the screenshot. The screen remains the same for other parts)*

A screenshot of a social media post

Description automatically generated

**Note:** This is the ***Point of No Return***. Do not press Finalize unless you’re absolutely sure that there won’t be a need to do a downgrade.

#### Modifications to the Environment

Some settings need to be changed in the new version of HDP. These came up during testing of the new version.

* Disable Atlas Taxonomy, as it’s a tech preview feature that got removed from HDP 2.6.x. **Ambari Web UI > Atlas > Advanced Configs > Custom application-properties**:

Name: atlas.feature.taxonomy.enable

Value: false

# Post-Upgrade, Pre-Finalize Tests

At this moment we’re left with a cluster that has been upgraded to HDP 2.6.5, all the services are running but we can still rollback to the previous version. It’s the perfect occasion to do the tests on the new release. There is one limitation though, configuration changes are not allowed and we cannot add or remove hosts from the cluster.

## HDFS

* Determine if you can reach the NameNode server with your browser and use the Namenode web UI and the Utilities menu to browse the file system:

DR cluster - http://hklpasrad.global.standardchartered.com:50070

* Create the hdfs user directory in HDFS and delete it:

su - <hdfs\_user>

hdfs dfs -mkdir -p /tmp/testdirectory

* Try copying a file into HDFS and catting that file, then delete the directory:

hdfs dfs -copyFromLocal /etc/passwd /tmp/testdirectory/passwd

hdfs dfs -cat /tmp/testdirectory/passwd

hdfs dfs -rm -r -f -skipTrash /tmp/testdirectory

## Yarn MapReduce

* Run the following commands from an edge node. This will generate 3TB of data. Clean up after the run:

# 6 disks \* 17 worker nodes = 102 total disks

# generate 1TB of data

# mappers = 2\*( total disks / 3 ) = 68

# replication x3

time yarn jar \

/usr/hdp/current/hadoop-mapreduce-client/hadoop-mapreduce-examples.jar \

teragen -Ddfs.replication=3 -Dmapreduce.job.maps=68 \

10000000000 TS\_input1

# cleanup the generated data

hadoop fs -rm -r -f -skipTrash TS\_input1

## HBase

* Run the following commands. This will get the HBase status and list the tables:

hbase shell

hbase(main):001:0> list

(expect to see a list of tables)

...

hbase(main):001:0> status 'detailed'

(should return the status of HBase as all good)

...

## Phoenix

* Run the following commands. List the tables and do a select on a table:

phoenix-sqlline

0: jdbc:phoenix:> !ta

(expect to see a list of tables)

...

0: jdbc:phoenix:> select \* from **MDMS\_DAILY\_READINGS** limit 10;

(expect to see 10 rows returned)

...

## Application Checks

* Check the endpoint of one of the microservices to access data from HBase using Phoenix
* Spark jobs!

# Post Finalize Tasks

After all the tests have passed we can Finalize the upgrade. On to the Ambari Web UI > **Admin > Stack and Versions**. Go to **Versions** tab and press the **Finalize** button under the HDP 2.6.5.0 version. We have a few more tasks to finish for the upgrade process.

## General Post-upgrade Tasks

* Check the log rotation settings as Ambari 2.6.0.0 introduced simplified parameters for these[[5]](#footnote-5)
  + not needed on QA cluster
* Re-enable services Auto Start: **Ambari Web UI > Admin > Service Auto Start**. Change the **Auto-Start Services** slider to **Enabled**.

A screenshot of a social media post

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a cell phone

Description automatically generated

## Kafka Post-upgrade Tasks

After the Express upgrade is complete, we have to change the protocol values to match the newly installed Kafka 1.0 version.

* In **Custom kafka-broker** section in Ambari Configs, edit the inter.broker.protocol.version with the following values and click **Save**.

Key: inter.broker.protocol.version

Value: 1.0

* Perform rolling restart of all Kafka Brokers: **Ambari Web UI > Services > Kafka** and select **Rolling restart** from the Service Actions menu.
* In **Custom kafka-broker** section in Ambari Configs, edit the log.message.format.version with the following values and click **Save**.

Key: log.message.format.version

Value: 1.0

* Perform rolling restart of all Kafka Brokers: **Ambari Web UI > Services > Kafka** and select **Rolling restart** from the Service Actions menu.

## Ranger Post-upgrade Tasks

* Some steps from might be needed. Check by following the documentation here: <https://docs.hortonworks.com/HDPDocuments/Ambari-2.6.2.2/bk_ambari-upgrade/content/upgrading_HDP_post_upgrade_tasks.html>  
  (Ranger delta user LDAP sync)
  + We enabled incremental user sync (delta sync) on QA

## HBase Post-upgrade Tasks

* Revert the value below after Lily upgrade has been completed :

A picture containing screenshot

Description automatically generated

## 

## Phoenix Hot-Fix Installation (only if issue is observed)

We need to Install the HDP Phoenix patch 2.6.5.69-1 as it contains the Phoenix Query Server fix for double results. Ambari could be used for installation but since this is an air gapped environment, it’s easier to just replace the jar files directly

* Download the patch or copy it from other internal systems: <http://private-repo-1.hortonworks.com/HDP/centos7/2.x/updates/2.6.5.69-1/tars/phoenix-4.7.0.2.6.5.69-1.tar.gz>
* Stop the PQS service
* Extract the tar file into a temporary directory

cd <some\_temp\_directory>

tar xzf <path\_to\_tar\_file>/phoenix-4.7.0.2.6.5.69-1.tar.gz

rm -rf bin dev examples docs phoenix-hperf

* Distribute and replace the older version of jar files on all nodes that are running PQS service

cd phoenix-4.7.0.2.6.5.69-1/

sudo cp -r /usr/hdp/current/phoenix-server /usr/hdp/current/phoenix-server.bkp17092019

sudo rm -rf /usr/hdp/current/phoenix-server/phoenix-4.7.0.2.6.5.0-292-\*.jar

ls -la /usr/hdp/current/phoenix-server/

sudo cp -r phoenix-4.7.0.2.6.5.69-1-\*.jar /usr/hdp/current/phoenix-server/

cd /usr/hdp/current/phoenix-server/

sudo rm phoenix-client.jar

sudo ln -s phoenix-4.7.0.2.6.5.69-1-client.jar phoenix-client.jar

sudo rm -rf phoenix-server.jar; sudo ln -s phoenix-4.7.0.2.6.5.69-1-server.jar phoenix-server.jar

sudo rm -rf phoenix-spark2.jar; sudo ln -s phoenix-4.7.0.2.6.5.69-1-spark2.jar phoenix-spark2.jar

sudo rm -rf phoenix-thin-client.jar; sudo ln -s phoenix-4.7.0.2.6.5.69-1-thin-client.jar phoenix-thin-client.jar

sudo rm -rf phoenix-hive.jar; sudo ln -s phoenix-4.7.0.2.6.5.69-1-hive.jar phoenix-hive.jar

cd /home/hdpadm/phoenix/phoenix-4.7.0.2.6.5.69-1

sudo rm -rf /usr/hdp/current/phoenix-server/lib/phoenix-\*.jar

sudo cp lib/phoenix-\* /usr/hdp/current/phoenix-server/lib/

* Verify that all symlinks in /usr/hdp/2.\*/phoenix/ are still correct
* Restart PQS service

# Appendix 1 – Troubleshooting

1. Use upgrade status API to get current upgrade status
   1. Invoke the following Ambari API call:  http://<ambari-server>:8080/api/v1/clusters/c1/upgrades
   2. From the output of above, identify the latest upgrade id : http://<ambari-server>:8080/api/v1/clusters/c1/upgrades/441
   3. To get information upto upgrade\_item level: http://<ambari-server>:8080/api/v1/clusters/c1/upgrades/441?fields=upgrade\_groups/upgrade\_items/UpgradeItem/status,upgrade\_groups/upgrade\_items/UpgradeItem/context,upgrade\_groups/UpgradeGroup/title
   4. To get information up to task level: http://<ambari-server>:8080/api/v1/clusters/c1/upgrades/441?fields=upgrade\_groups/upgrade\_items/tasks/Tasks/status,upgrade\_groups/upgrade\_items/tasks/Tasks/command\_detail,upgrade\_groups/upgrade\_items/tasks/Tasks/stderr

1. <https://supportmatrix.hortonworks.com/> [↑](#footnote-ref-1)
2. <https://supportmatrix.hortonworks.com/> [↑](#footnote-ref-2)
3. <https://docs.hortonworks.com/HDPDocuments/Ambari-2.6.2.2/bk_ambari-upgrade/content/prepare-for-kafka-upgrade.html> [↑](#footnote-ref-3)
4. <https://community.hortonworks.com/content/supportkb/150607/error-python-script-has-been-killed-due-to-timeout-1.html> [↑](#footnote-ref-4)
5. <https://docs.hortonworks.com/HDPDocuments/Ambari-2.6.2.2/bk_ambari-upgrade/content/upgrading_log_rotation_configuration.html> [↑](#footnote-ref-5)