

Hortonworks Data Platform

Ambari Troubleshooting Guide

(May 9, 2016)

Hortonworks Data Platform : Ambari Troubleshooting Guide

Copyright © 2012-2016 Hortonworks, Inc. All rights reserved.

The Hortonworks Data Platform, powered by Apache Hadoop, is a massively scalable and 100% open source platform for storing, processing and analyzing large volumes of data. It is designed to deal with data from many sources and formats in a very quick, easy and cost-effective manner. The Hortonworks Data Platform consists of the essential set of Apache Hadoop projects including MapReduce, Hadoop Distributed File System (HDFS), HCatalog, Pig, Hive, HBase, Zookeeper and Ambari. Hortonworks is the major contributor of code and patches to many of these projects. These projects have been integrated and tested as part of the Hortonworks Data Platform release process and installation and configuration tools have also been included.

Unlike other providers of platforms built using Apache Hadoop, Hortonworks contributes 100% of our code back to the Apache Software Foundation. The Hortonworks Data Platform is Apache-licensed and completely open source. We sell only expert technical support, [training](#) and partner-enablement services. All of our technology is, and will remain free and open source. Please visit the [Hortonworks Data Platform](#) page for more information on Hortonworks technology. For more information on Hortonworks services, please visit either the [Support](#) or [Training](#) page. Feel free to [Contact Us](#) directly to discuss your specific needs.

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Table of Contents

1. Troubleshooting Ambari Deployments	1
1.1. Reviewing Ambari Log Files	1
1.2. Resolving Ambari Install and Setup Problems	2
1.2.1. Problem: Browser crashed before Install Wizard completes	2
1.2.2. Problem: Install Wizard reports that the cluster install has failed	2
1.2.3. Problem: Ambari Agents May Fail to Register with Ambari Server.	3
1.2.4. Problem: The “yum install ambari-server” Command Fails	4
1.2.5. Problem: HDFS Smoke Test Fails	4
1.2.6. Problem: yum Fails on Free Disk Space Check	4
1.2.7. Problem: Ambari Agents fail to register during the Confirm Host step	5
1.3. Resolving Cluster Deployment Problems	5
1.3.1. Problem: Trouble Starting Ambari on System Reboot	5
1.3.2. Problem: Metrics and Host information display incorrectly in Ambari Web	6
1.3.3. Problem: On SUSE 11 Ambari Agent crashes within the first 24 hours	6
1.3.4. Problem: Attempting to Start HBase REST server causes either REST server or Ambari Web to fail	6
1.3.5. Problem: Multiple Ambari Agent processes are running, causing re-register	6
1.3.6. Problem: Ambari stops MySQL database during deployment, causing Ambari Server to crash.	7
1.3.7. Problem: Cluster Install Fails with Groupmod Error	7
1.3.8. Problem: Host registration fails during Agent bootstrap on SLES due to timeout.	7
1.3.9. Problem: Host Check Fails if Transparent Huge Pages (THP) is not disabled.	8
1.3.10. Problem: DataNode Fails to Install on RHEL/CentOS 7.	8
1.3.11. Problem: When running Ambari Server as non-root, kadmin couldn't open log file.	9
1.3.12. Problem: Adding client-only services does not automatically install component dependencies.	9
1.3.13. Problem: Automatic Agent Registration with SSH fails for a non-root configuration.	9
1.4. Resolving Cluster Upgrade Problems	9
1.4.1. Problem: Versions tab does not show in Ambari Web.	9
1.4.2. Problem: YARN Service Checks Fail and ResourceManager fails to start	10
1.4.3. Problem: After HDP 2.2 -> 2.3 Manual Upgrade, Ambari alerts service flags Kafka Broker down.	10
1.4.4. Problem: Ranger Admin UI does not function after upgrading HDP from 2.2 to 2.3	10
1.4.5. Problem: Rolling or Express Upgrade fails while stopping HBase RegionServers	11
1.5. Resolving General Problems	12
1.5.1. Problem: When installing HDP 2.3.0 or 2.3.2, YARN ATS fails to start.	12

1.5.2. Problem: After upgrading to Ambari 2.2, you receive File Does Not Exist alerts.	12
1.5.3. During Enable Kerberos, the Check Kerberos operation fails.	13
1.5.4. Problem: Hive developers may encounter an exception error message during Hive Service Check	13
1.5.5. Problem: API calls for PUT, POST, DELETE respond with a "400 - Bad Request"	13
1.5.6. Problem: Ambari is checking disk full on non-local disks; causing a high number of auto-mounted home directories	14
1.5.7. Problem: Ambari Web shows Storm summary values as N/A in a Kerberized cluster	14
1.5.8. Problem: kadmin running Ambari Server as non-root, cannot open log file.	14
1.5.9. Problem: After changing NameNode RPC port, Ambari shows both NameNodes as standby.	15

1. Troubleshooting Ambari Deployments

The first step in troubleshooting any problem in an Ambari-deploying Hadoop cluster is [Reviewing the Ambari Log Files](#).

Find a recommended solution to a troubleshooting problem in one of the following sections:

- [Resolving Ambari Install and Setup Problems](#)
- [Resolving Cluster Install and Configuration Problems](#)
- [Resolving Cluster Upgrade Problems](#)
- [Resolving General Problems](#)

1.1. Reviewing Ambari Log Files

Find files that log activity on an Ambari host in the following locations:

- Ambari Server logs on the Ambari Server host:

```
/var/log/ambari-server/ambari-server.log
```

- Ambari Agent logs on any host with an Ambari Agent:

```
/var/log/ambari-agent/ambari-agent.log
```

- Ambari Agent task logs on any host with an Ambari Agent:

```
/var/lib/ambari-agent/data/
```

This location contains logs for all tasks executed on an Ambari Agent host. Each log name includes:

- command-N.json - the command file corresponding to a specific task.
- output-N.txt - the output from the command execution.
- errors-N.txt - error messages.



Note

You can configure the logging level for `ambari-server.log` by modifying `/etc/ambari-server/conf/log4j.properties` on the Ambari Server host. For the Ambari Agents, you can set the `loglevel` in `/etc/ambari-agent/conf/ambari-agent.ini` on each host running an Ambari Agent. In general, you should also consider setting log-rotate policies for your systems. Refer to your operating system documentation and <http://linuxconfig.org/logrotate-8-manual-page> for more information on setting up log-rotate in your environment.

1.2. Resolving Ambari Install and Setup Problems

Try the recommended solution for each of the following problems.

1.2.1. Problem: Browser crashed before Install Wizard completes

Your browser crashes or you accidentally close your browser before the Install Wizard completes.

1.2.1.1. Solution

The response to a browser closure depends on where you are in the process:

- The browser closes before you press the `Deploy` button.

Re-launch the same browser and continue the install process. Using a different browser forces you to re-start the entire process.

- The browser closes after you press `Deploy`, while or after the `Install`, `Start`, and `Test` screen opens.

Re-launch the same browser and continue the process, or log in again, using a different browser. When the `Install`, `Start`, and `Test` displays, proceed.

1.2.2. Problem: Install Wizard reports that the cluster install has failed

The `Install`, `Start`, and `Test` screen reports that the cluster install has failed.

1.2.2.1. Solution

The response to a report of install failure depends on the cause of the failure:

- The failure is due to intermittent network connection errors during software package installs.

Use the `Retry` button on the `Install`, `Start`, and `Test` screen.

- The failure is due to misconfiguration or other setup errors.

1. Use the left navigation bar to go back to the appropriate screen. For example, `Customize Services`.
2. Make your changes.
3. Continue in the normal way.

- The failure occurs during the start/test sequence.

1. Click **Next and Complete**, then proceed to the **Monitoring Dashboard**.
 2. Use the **Services View** to make your changes.
 3. Re-start the service using **Service Actions**.
- The failure is due to something else.
 1. Open an SSH connection to the Ambari Server host.
 2. Clear the database. At the command line, type:

```
ambari-server reset
```
 3. Clear your browser cache.
 4. Re-run the Install Wizard.

1.2.3. Problem: Ambari Agents May Fail to Register with Ambari Server.

When deploying HDP using Ambari 1.4.x or later on RHEL CentOS 6.5, click the “Failed” link on the Confirm Hosts page in the Cluster Install wizard to display the Agent logs. The following log entry indicates the SSL connection between the Agent and Server failed during registration:

```
INFO 2014-04-02 04:25:22,669 NetUtil.py:55 - Failed to
connect to https://{ambari-server}:8440/cert/ca due to
[Errno 1] _ssl.c:492: error:100AE081:elliptic curve
routines:EC_GROUP_new_by_curve_name:unknown group
```

For more detailed information about this OpenSSL issue, see https://bugzilla.redhat.com/show_bug.cgi?id=1025598

1.2.3.1. Solution:

In certain recent Linux distributions, such as RHEL/Centos/Oracle Linux 6.x, the default value of `nproc` is lower than the value required to deploy the HBase service successfully. If you are deploying HBase, change the value of `nproc`:

1. Check the OpenSSL library version installed on your host(s):

```
rpm -qa | grep openssl openssl-1.0.1e-15.el6.x86_64
```

2. If the output reads `openssl-1.0.1e-15.x86_64 (1.0.1 build 15)`, you must upgrade the OpenSSL library. To upgrade the OpenSSL library, run the following command:

```
yum upgrade openssl
```

3. Verify you have the newer version of OpenSSL (1.0.1 build 16):

```
rpm -qa | grep openssl-1.0.1e-16.el6.x86_64
```

4. Restart Ambari Agent(s) and click Retry -> Failed in the wizard user interface.

1.2.4. Problem: The “yum install ambari-server” Command Fails

You are unable to get the initial install command to run.

1.2.4.1. Solution:

You may have incompatible versions of some software components in your environment. See Meet Minimum System Requirements in Installing HDP Using Ambari for more information, then make any necessary changes.

1.2.5. Problem: HDFS Smoke Test Fails

If your DataNodes are incorrectly configured, the smoke tests fail and you get this error message in the DataNode logs:

```
DisallowedDataNodeException  
org.apache.hadoop.hdfs.server.protocol.  
DisallowedDatanodeException
```

1.2.5.1. Solution:

1. Make sure that reverse DNS look-up is properly configured for all nodes in your cluster.
2. Make sure you have the correct FQDNs when specifying the hosts for your cluster. Do not use IP addresses - they are not supported.
3. Restart the installation process.

1.2.6. Problem: yum Fails on Free Disk Space Check

If you boot your Hadoop DataNodes with/as a ramdisk, you must disable the free space check for yum before doing the install. If you do not disable the free space check, yum will fail with the following error:

```
Fail: Execution of '/usr/bin/yum -d 0 -e 0 -y install unzip'  
returned 1. Error Downloading Packages: unzip-6.0-1.el6.x86_64:  
Insufficient space in download directory /var/cache/yum/x86_64/6/  
base/packages * free 0 * needed 149 k
```

1.2.6.1. Solution:

To disable free space check, update the DataNode image with a directive in `/etc/yum.conf`:

```
diskspacecheck=0
```


1.2.7. Problem: Ambari Agents fail to register during the Confirm Host step

Ambari Agents fail to register during the Confirm Host step of deploying a new cluster when OpenJDK 1.7.0 is being used by the Ambari Server

1.2.7.1. Solution: Disable all cipher suites that use ECC on the Ambari Server.

OpenJDK 1.7 has compatibility issues with SSL/TLS cipher suites that use Elliptical Curve Cryptography (ECC). Because of these compatibility issues, all cipher suites that use ECC need to be disabled on the Ambari Server. To do so, edit the `/etc/ambari-server/conf/ambari.properties` file and add the following line to the end of the file:

```
security.server.disabled.ciphers=
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384|TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384|
TLS_ECDH_ECDSA_WITH_AES_256_CBC_SHA384|TLS_ECDH_RSA_WITH_AES_256_CBC_SHA384|
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA|TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA|
TLS_ECDH_ECDSA_WITH_AES_256_CBC_SHA|TLS_ECDH_RSA_WITH_AES_256_CBC_SHA|
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256|TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256|
TLS_ECDH_ECDSA_WITH_AES_128_CBC_SHA256|TLS_ECDH_RSA_WITH_AES_128_CBC_SHA256|
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA|TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA|
TLS_ECDH_ECDSA_WITH_AES_128_CBC_SHA|TLS_ECDH_RSA_WITH_AES_128_CBC_SHA|
TLS_ECDHE_ECDSA_WITH_3DES_EDE_CBC_SHA|TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA|
TLS_ECDH_ECDSA_WITH_3DES_EDE_CBC_SHA|TLS_ECDH_RSA_WITH_3DES_EDE_CBC_SHA|
TLS_ECDH_anon_WITH_AES_256_CBC_SHA|TLS_ECDH_anon_WITH_AES_128_CBC_SHA|
TLS_ECDH_anon_WITH_3DES_EDE_CBC_SHA|TLS_ECDHE_ECDSA_WITH_NULL_SHA|
TLS_ECDHE_RSA_WITH_NULL_SHA|TLS_ECDH_ECDSA_WITH_NULL_SHA|
TLS_ECDH_RSA_WITH_NULL_SHA|TLS_ECDH_anon_WITH_NULL_SHA
```

Once this has been done, restart the Ambari Server using `ambari-server restart` and retry registering Agents.

1.3. Resolving Cluster Deployment Problems

Try the recommended solution for each of the following problems.

1.3.1. Problem: Trouble Starting Ambari on System Reboot

If you reboot your cluster, you must restart the Ambari Server and all the Ambari Agents manually.

1.3.1.1. Solution:

Log in to each machine in your cluster separately:

1. On the Ambari Server host machine:

```
ambari-server start
```

2. On each host in your cluster:

```
ambari-agent start
```

1.3.2. Problem: Metrics and Host information display incorrectly in Ambari Web

Charts appear incorrectly or not at all despite Host health status is displayed incorrectly.

1.3.2.1. Solution:

All the hosts in your cluster and the machine from which you browse to Ambari Web must be in sync with each other. The easiest way to assure this is to enable NTP.

1.3.3. Problem: On SUSE 11 Ambari Agent crashes within the first 24 hours

SUSE 11 ships with Python version 2.6.0-8.12.2 which contains a known defect that causes this crash.

1.3.3.1. Solution:

Upgrade to Python version 2.6.8-0.15.1.

1.3.4. Problem: Attempting to Start HBase REST server causes either REST server or Ambari Web to fail

As an option you can start the HBase REST server manually after the install process is complete. It can be started on any host that has the HBase Master or the Region Server installed. If you install the REST server on the same host as the Ambari server, the http ports will conflict.

1.3.4.1. Solution

In starting the REST server, use the `-p` option to set a custom port. Use the following command to start the REST server. `/usr/lib/hbase/bin/hbase-daemon.sh start rest -p <custom_port_number>`

1.3.5. Problem: Multiple Ambari Agent processes are running, causing re-register

On a cluster host `ps aux | grep ambari-agent` shows more than one agent process running. This causes Ambari Server to get incorrect ids from the host and forces Agent to restart and re-register.

1.3.5.1. Solution

On the affected host, kill the processes and restart.

1. Kill the Agent processes and remove the Agent PID files found here: `/var/run/ambari-agent/ambari-agent.pid`.
2. Restart the Agent process:

```
ambari-agent start
```

1.3.6. Problem: Ambari stops MySQL database during deployment, causing Ambari Server to crash.

The Hive Service uses MySQL Server by default. If you choose MySQL server as the database on the Ambari Server host as the managed server for Hive, Ambari stops this database during deployment and crashes.

1.3.6.1. Solution

If you plan to use the default MySQL Server setup for Hive and use MySQL Server for Ambari - make sure that the two MySQL Server instances are different.

If you plan to use the same MySQL Server for Hive and Ambari - make sure to choose the existing database option for Hive.

1.3.7. Problem: Cluster Install Fails with Groupmod Error

The cluster fails to install with an error related to running `groupmod`. This can occur in environments where groups are managed in LDAP, and not on local Linux machines. You may see an error message similar to the following one:

```
Fail: Execution of 'groupmod hadoop' returned 10. groupmod: group 'hadoop' does not exist in /etc/group
```

1.3.7.1. Solution

When installing the cluster using the Cluster Installer Wizard, at the **Customize Services** step, select the **Misc** tab and choose the **Skip group modifications** during install option.

1.3.8. Problem: Host registration fails during Agent bootstrap on SLES due to timeout.

When using SLES and performing host registration using SSH, the Agent bootstrap may fail due to timeout when running the `setupAgent.py` script. The host on which the timeout occurs will show the following process hanging:

```
c6401.ambari.apache.org:/etc/
# ps -ef | grep zypper
root 18318 18317 5 03:15 pts/1 00:00:00 zypper -q search -s --
match-exact ambari-agent
```

1.3.8.1. Solution

1. If you have a repository registered that is prompting to accept keys, via user interaction, you may see the hang and timeout. In this case, run `zypper refresh` and confirm all repository keys are accepted for the `zypper` command to work without user interaction.

2. Another alternative is to perform manual Agent setup and not use SSH for host registration. This option does not require that Ambari call zypper without user interaction.

1.3.9. Problem: Host Check Fails if Transparent Huge Pages (THP) is not disabled.

When installing Ambari on RHEL/CentOS 6 using the Cluster Installer Wizard at the Host Checks step, one or more host checks may fail if you have not disabled Transparent Huge Pages on all hosts.

Host Checks will warn you when a failure occurs.

1.3.9.1. Solution

Disable THP. On all hosts,

1. Add the following command to your `/etc/rc.local` file:

```
if test -f /sys/kernel/mm/transparent_hugepage/enabled; then
echo never > /sys/kernel/mm/transparent_hugepage/enabled
fi if test -f /sys/kernel/mm/transparent_hugepage/defrag; then
echo never > /sys/kernel/mm/transparent_hugepage/defrag
fi
```

2. To confirm, reboot the host then run the following command:

```
$ cat /sys/kernel/mm/transparent_hugepage/enabled always madvise
[never]
```

1.3.10. Problem: DataNode Fails to Install on RHEL/CentOS 7.

During cluster install, DataNode fails to install with the following error:

```
resource_management.core.exceptions. Fail: Execution of '/usr/bin/
yum -d 0 -e 0 -y install snappy-devel' returned 1. Error: Package:
snappy-devel-1.0.5-1.el6.x86_64 (HDP-UTILS-1.1.0.20) Requires:
snappy(x86-64) = 1.0.5-1.el6 Installed: snappy-1.1.0-3.el7.x86_64
(@anaconda/7.1) snappy(x86-64) = 1.1.0-3.el7 Available:
snappy-1.0.5-1.el6.x86_64 (HDP-UTILS-1.1.0.20) snappy(x86-64) =
1.0.5-1.el6
```

1.3.10.1. Solution:

Hadoop requires the snappy-devel package that is a lower version than what is on the machine already. Run the following on the host and retry.

```
yum remove snappy

yum install snappy-devel
```

1.3.11. Problem: When running Ambari Server as non-root, kadmin couldn't open log file.

When running Ambari Server as non-root, when enabling Kerberos, if kadmin fails to authenticate, you will see the following error in ambari-server.log if Ambari cannot access the kadmind.log.

```
STDERR: Couldn't open log file /var/log/kadmind.log: Permission
denied kadmin: GSS-API (or Kerberos) error while initializing
kadmin interface
```

1.3.11.1. Solution:

To avoid this error, be sure the kadmind.log file has 644 permissions.

1.3.12. Problem: Adding client-only services does not automatically install component dependencies.

When adding client-only services to a cluster (using Add Service), Ambari does not automatically install dependent client components with the newly added clients.

1.3.12.1. Solution:

On hosts where client components need to be installed, browse to Hosts and to the Host Details page. Click + Add and select the client components to install on that host.

1.3.13. Problem: Automatic Agent Registration with SSH fails for a non-root configuration.

When using an Agent non-root configuration, if you attempt to register hosts automatically using SSH, the Agent registration will fail.

1.3.13.1. Solution:

The option to automatically register hosts with SSH is not supported when using a Agent non-root configuration. You must manually register the Agents.

1.4. Resolving Cluster Upgrade Problems

Try the recommended solution for each of the following problems.

1.4.1. Problem: Versions tab does not show in Ambari Web.

After performing an upgrade from HDP 2.1 and restarting Ambari Server and the Agents, if you browse to Admin > Stack and Versions in Ambari Web, the Versions tab does not display.

1.4.1.1. Solution:

Give all the Agent hosts in the cluster a chance connect to Ambari Server by waiting for Ambari to show the Agent heartbeats as green and then refresh your browser.

1.4.2. Problem: YARN Service Checks Fail and ResourceManager fails to start

When upgrading from HDP 2.2 -> 2.3, if your cluster contains `yarn.scheduler.capacity.root.accessible-node-labels.default.capacity` and `yarn.scheduler.capacity.root.accessible-node-labels.default.maximum-capacity` properties, these values must be valid before upgrading to HDP 2.3. During HDP upgrade, on ResourceManager start, if these values are invalid, you will get the following error:

```
Illegal capacity of 0.0 for children of queue root for label=defaultUpdate
YARN Configuration Properties for HDP 2.3
```

1.4.2.1. Solution:

From Ambari Web, browse to **Services > YARN > Configs**. On the Advanced tab, delete the following properties from capacity-scheduler:

```
yarn.scheduler.capacity.root.accessible-node-labels.default.capacity
```

```
yarn.scheduler.capacity.root.accessible-node-labels.default.maximum-capacity
```

Also, be sure these properties are valid (or not included) in your Blueprints when you create clusters.

1.4.3. Problem: After HDP 2.2 -> 2.3 Manual Upgrade, Ambari alerts service flags Kafka Broker down.

After performing a manual upgrade from HDP 2.2 - 2.3, Ambari display alerts for Kafka Brokers.

1.4.3.1. Solution:

During upgrade, Ambari adds the `listeners=PLAINTEXT://localhost:6667` to `/etc/kafka/conf/server.properties` this cause Kafka to listen on localhost at port 6667 and Ambari alerts service will not be able to reach Kafka broker and flags it as service down. users can run following command. It is not until Ambari set-current is run (per the upgrade instructions) that this value is corrected.

1.4.4. Problem: Ranger Admin UI does not function after upgrading HDP from 2.2 to 2.3

Ranger upgrade patches may fail to complete, during an upgrade of the HDP Stack from 2.2 to 2.3, causing the Ranger Admin UI to not function correctly after the upgrade.

1.4.4.1. Solution: Run the DB and Java patch scripts manually, then Retry Upgrading Ranger.

1. On the Rolling Upgrade dialog, stdout tab, review the Ambari log files to determine which patch caused a timeout. For example, the following image shows a timeout during SQL patch execution.

Rolling Upgrade to HDP-2.3.0.0

Upgrade Paused

```

2015-10-15 08:36:55,152 [I] ----- Check ranger user connection -----
2015-10-15 08:36:55,152 [I] Checking connection..
2015-10-15 08:36:55,772 [I] Checking connection passed.
2015-10-15 08:36:55,773 [I] ----- Check audit table exists -----
2015-10-15 08:36:56,452 [I] Table xa_access_audit already exists in database 'ranger_audit'
2015-10-15 08:36:56,452 [I] -----Granting privileges TO 'rangerlogger' on 'ranger_audit'-----
2015-10-15 08:36:57,057 [I] Granting privileges to 'rangerlogger' done on 'ranger_audit'
2015-10-15 08:36:57,057 [I] -----Granting privileges TO 'rangerlogger' on 'ranger_audit'-----
2015-10-15 08:36:57,709 [I] Granting privileges to 'rangerlogger' done on 'ranger_audit'
2015-10-15 08:36:57,770 [I] -----Granting privileges TO 'rangerlogger' on 'ranger_audit'-----
2015-10-15 08:36:58,413 [I] Granting privileges to 'rangerlogger' done on 'ranger_audit'
2015-10-15 08:36:58,414 [I] ----- Applying Audit DB patches -----
2015-10-15 08:36:58,414 [I] ----- Checking XA_ACCESS_AUDIT table to apply audit db patches -----
2015-10-15 08:36:59,077 [I] Table xa_access_audit already exists in database 'ranger_audit'
2015-10-15 08:36:59,078 [I] Executing patch on ranger_audit from file: @11-auditcolumnssize.sql
2015-10-15 08:45:39,146 [I] @11-auditcolumnssize.sql patch applied
2015-10-15 08:45:39,846 [I] Patch version updated
2015-10-15 08:45:39,846 [I] ----- Checking XA_ACCESS_AUDIT table to apply audit db patches -----
2015-10-15 08:45:40,513 [I] Table xa_access_audit already exists in database 'ranger_audit'
2015-10-15 08:45:40,514 [I] Executing patch on ranger_audit from file: @15-auditlogaggregation.sql

```

Based on your review, determine whether DB or Java patch scripts (or both) have failed to complete.

2. Log in to the Ranger Admin host.
3. On the Ranger Admin host, in `/usr/hdp/2.3.x.y-z/ranger-admin/`, run the following commands:

For DB patch failure:

```
python db_setup.py
```

For JAVA patch failure:

```
python db_setup.py -javapatch
```

4. Confirm that all patches complete successfully.
5. In Ambari Web, in the Rolling Upgrade dialog at the Ranger Paused step, click Retry.

1.4.5. Problem: Rolling or Express Upgrade fails while stopping HBase RegionServers

When upgrading to HDP 2.3.2 on Ubuntu or Debian, and you are using custom service accounts, the upgrade (Rolling or Express) fails while stopping HBase RegionServers.

1.4.5.1. Solution:

If you perform an upgrade (Rolling or Express) to HDP 2.3.2, the HBase RegionServers will fail when stopping if you have custom service accounts configured. This occurs because when the HDP 2.3.2 packages are installed, the ownership of the pid directory for HBase is changed. To correct this situation, on the host, change the pid directory ownership for your custom service accounts. For example: if your custom service account for HBase is "cstm-hbase", change the ownership as described below and proceed with the upgrade.

```
chown cstm-hbase:hbase -R /var/run/hbase/
```

1.5. Resolving General Problems

1.5.1. Problem: When installing HDP 2.3.0 or 2.3.2, YARN ATS fails to start.

If you install an HDP cluster using HDP 2.3.0 or HDP 2.3.2, the YARN ATS server will fail to start with the following error in the yarn log:

```
2015-12-09 22:56:41,816 FATAL
applicationhistoryservice.ApplicationHistoryServer
(ApplicationHistoryServer.java:launchAppHistoryServer (161)) - Error starting
ApplicationHistoryServer java.lang.RuntimeException:
java.lang.RuntimeException : java.lang.ClassNotFoundException : Class
org.apache.hadoop.yarn.server.timeline.EntityGroupFSTimelineStore not found at
org.apache.hadoop.conf.Configuration.getClass (Configuration.java:2227)
```

1.5.1.1. Solution:

Update the YARN configuration to use the LevelDB store:

1. In **Ambari Web**, browse to **Services > YARN > Configs**.
2. Filter for the `yarn.timeline-service.store-class` property and set to `org.apache.hadoop.yarn.server.timeline.LeveldbTimelineStore` value.
3. Save the configuration change and restart YARN.

1.5.2. Problem: After upgrading to Ambari 2.2, you receive File Does Not Exist alerts.

After upgrading to Ambari 2.2, you receive alerts for "DataNode Unmounted Data Dir" that the `/var/lib/ambari-agent/data/datanode/dfs_data_dir_mount.hist` file does not exist. The `hadoop-env/dfs.datanode.data.dir.mount.file` configuration property is no longer customizable from Ambari. The original default value of `/etc/hadoop/conf/dfs_data_dir_mount.hist` is now `/var/lib/ambari-agent/data/datanode/dfs_data_dir_mount.hist`, which is not customizable. On Ambari Agent upgrade, Ambari will automatically move the file from `/etc/hadoop/conf/dfs_data_dir_mount.hist` to `/var/lib/ambari-agent/data/datanode/dfs_data_dir_mount.hist`. If you have not modified this configuration property, no action is required.

1.5.2.1. Solution:

If you had previously modified the `hadoop-env/dfs.datanode.data.dir.mount.file` value to a custom location, after upgrading to Ambari 2.2, you must restart your DataNodes for the file to be written to be the new location.

1.5.3. During Enable Kerberos, the Check Kerberos operation fails.

When enabling Kerberos using the wizard, the Check Kerberos operation fails. In `/var/log/ambari-server/ambari-server.log`, you see a message: `02:45:44,490 WARN [qtp567239306-238] MITKerberosOperationHandler:384 - Failed to execute kadmin:`

1.5.3.1. Solution 1:

Check that NTP is running and confirm your hosts and the KDC times are in sync. A time skew as little as 5 minutes can cause Kerberos authentication to fail.

1.5.3.2. Solution 2: (on RHEL/CentOS/Oracle Linux)

Check that the Kerberos Admin principal being used has the necessary KDC ACL rights as set in `/var/kerberos/krb5kdc/kadm5.acl`.

1.5.4. Problem: Hive developers may encounter an exception error message during Hive Service Check

MySQL is the default database used by the Hive metastore. Depending on several factors, such as the version and configuration of MySQL, a Hive developer may see an exception message similar to the following one:

```
An exception was thrown while adding/validating classes) :
Specified key was too long; max key length is 767 bytes
```

1.5.4.1. Solution

Administrators can resolve this issue by altering the Hive metastore database to use the Latin1 character set, as shown in the following example: `mysql> ALTER DATABASE <metastore.database.name> character set latin1;`

1.5.5. Problem: API calls for PUT, POST, DELETE respond with a "400 - Bad Request"

When attempting to perform a REST API call, you receive a 400 error response. REST API calls require the "X-Requested-By" header.

1.5.5.1. Solution

Starting with Ambari 1.4.2, you must include the "X-Requested-By" header with the REST API calls.

For example, if using curl, include the `-H "X-Requested-By: ambari"` option.

```
curl -u admin:admin -H "X-Requested-By: ambari" -X DELETE http://  
<ambari-host>:8080/api/v1/hosts/host1
```

1.5.6. Problem: Ambari is checking disk full on non-local disks; causing a high number of auto-mounted home directories

When Ambari issues its check to detect local disk capacity and use for each Ambari Agent, it uses `df` by default instead of `df -l` to only check local disks. If using NFS auto-mounted home directories, this can lead to a high number of home directories being mounted on each host; causing shutdown delays and disk capacity check delays.

1.5.6.1. Solution:

On the Ambari Server, edit the `/etc/ambari-server/conf/ambari.properties` and add the following property to only check locally mounted devices.

```
agent.check.remote.mounts=false
```

1.5.7. Problem: Ambari Web shows Storm summary values as N/A in a Kerberized cluster

With a Kerberos-enabled cluster that includes Storm, in Ambari Web > Services > Storm, the Summary values for Slots, Tasks, Executors and Topologies show as "n/a". Ambari Server log also includes the following ERROR:

```
24 Mar 2015 13:32:41,  
288 ERROR [pool-2-thread-362]  
AppCookieManager:122 -  
SPNego authentication failed,  
cannot get hadoop.auth cookie for URL:  
http: //c6402.ambari.apache.org:8744/api/  
v1/topology/summary?field=topologies
```

1.5.7.1. Solution:

When Kerberos is enabled, Storm API requires SPNEGO authentication. Refer to the Ambari Security Guide to [Set Up Ambari for Kerberos](#) to enable Ambari to authenticate against the Storm API via SPNEGO.

1.5.8. Problem: kadmin running Ambari Server as non-root, cannot open log file.

When running Ambari Server as non-root, when enabling Kerberos, if kadmin fails to authenticate, you will see the following error in `ambari-server.log` if Ambari cannot access the `kadmind.log`.

```
STDERR: Couldn't open log file /var/log/kadmind.log: Permission denied  
kadmin: GSS-API (or Kerberos) error while initializing kadmin interface
```

1.5.8.1. Solution:

Be sure the user that Ambari Server is configured to run has permissions to write to the `kadmind.log`.

1.5.9. Problem: After changing NameNode RPC port, Ambari shows both NameNodes as standby.

If you have enabled NameNode HA and change the NameNode RPC ports (by customizing the `dfs.namenode.servicerpc-address` property), Ambari will show both NameNodes as standby.

1.5.9.1. Solution:

When modifying the NameNode RPC port (`dfs.namenode.servicerpc-address`) after enabling NameNode HA, you need to format ZKFC to make sure that the config data in ZooKeeper is refreshed. Run the following command to format ZKFC znode:

```
su - <hdfs-user> -c 'hdfs zkfc -formatZK'
```