

Hortonworks Data Platform

Ambari Troubleshooting Guide

(May 23, 2014)

Hortonworks Data Platform : Ambari Troubleshooting Guide

Copyright © 2012-2014 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 Installer Problems	1
1.2.1. Problem: Browser crashed before Install Wizard completes	1
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	3
1.2.5. Problem: HDFS Smoke Test Fails	3
1.2.6. Problem: yum Fails on Free Disk Space Check	4
1.2.7. Problem: The HCatalog Daemon Metastore Smoke Test Fails	4
1.2.8. Problem: A service with a customized service user is not appearing properly in Ambari Web	5
1.3. Resolving Cluster Deployment Problems	5
1.3.1. Problem: YARN ATS Component May Fail to Start When Installing HDP 2.1.3 with Ambari	5
1.3.2. Problem: Trouble Starting Ambari on System Reboot	5
1.3.3. Problem: Metrics and Host information display incorrectly in Ambari Web	6
1.3.4. Problem: On SUSE 11 Ambari Agent crashes within the first 24 hours	6
1.3.5. Problem: Attempting to Start HBase REST server causes either REST server or Ambari Web to fail	6
1.3.6. Problem: Multiple Ambari Agent processes are running, causing re- register	6
1.3.7. Problem: Some graphs do not show a complete hour of data until the cluster has been running for an hour	7
1.3.8. Problem: Ambari stops MySQL database during deployment, causing Ambari Server to crash.	7
1.3.9. Problem: Service Fails with Unknown Host Exception	7
1.4. Resolving General Problems	8
1.4.1. Problem: Hive developers may encounter an exception error message during Hive Service Check	8
1.4.2. Problem: API calls for PUT, POST, DELETE respond with a "400 - Bad Request"	8
1.4.3. Problem: Nagios alerts for CPU utilization show as UNKNOWN	9

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 Installer Problems](#)
- [Resolving Cluster Deployment 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

`<your.Ambari.server.host>/var/log/ambari-server/ambari-server.log`

- Ambari Agent logs

`<your.Ambari.agent.host>/var/log/ambari-agent/ambari-agent.log`

- Ambari Action logs

`<your.Ambari.agent.host>/var/lib/ambari-agent/data/`

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

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

1.2. Resolving Ambari Installer 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 prior to hitting 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 the **Deploy** button has launched the **Install, Start, and Test** screen.

Re-launch the same browser and continue the process or use a different browser and re-login. You are returned to the **Install, Start, and Test** screen.

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 nav 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** and 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 (like RHEL/CentOS/Oracle Linux 6.x), the default value of `nproc` is lower than the value required if you are deploying the HBase service. To change this value:

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 by running the following command:

```
yum upgrade openssl
```

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

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

4. Restart Ambari Agent(s) and Click “Retry Failed” on the Wizard.

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. Check the [Minimum System Requirements](#) and 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:

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

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: The HCatalog Daemon Metastore Smoke Test Fails

If the HCatalog smoke test fails, this is displayed in your console:

```
Metastore startup failed, see /var/log/hcatalog/hcat.err
```

1.2.7.1. Solution:

1. Log into the HCatalog node in your cluster.
2. Open /var/log/hcatalog/hcat.err or /var/log/hive/hive.log (one of the two will exist depending on the installation) with a text editor.
3. In the file, see if there is a MySQL Unknown Host Exception like this:

```
at java.lang.reflect.Method.invoke (Method.java:597)
at org.apache.hadoop.util.Runjar.main (runjar.java:156)
Caused by: java.net.UnknownHostException:mysql.host.com
at java.net.InetAddress.getAllByName (InetAddress.java:1157)
```

This exception can be thrown if you are using a previously existing MySQL instance and you have incorrectly identified the host name during the installation process. When you do the reinstall, make sure this name is correct.

4. In the file, see if there is an `ERROR Failed initializing database` entry like this:

```
11/12/29 20:52:04 ERROR DataNucleus.Plugin: Bundle
org.eclipse.jdt.core required
11/12/29 20:52:04 ERROR DataStore.Schema: Failed initialising
database
```

This exception can be thrown if you are using a previously existing MySQL instance and you have incorrectly identified the username/password during the installation process. It can also occur when the user you specify does not have adequate privileges on the database. When you do the reinstall, make sure this username/password is correct and that the user has adequate privilege.

5. Restart the installation process.

1.2.8. Problem: A service with a customized service user is not appearing properly in Ambari Web

You are unable to monitor or manage a service in Ambari Web when you have created a customized service user name with a hyphen, for example, `hdfs-user`.

1.2.8.1. Solution

Hyphenated service user names are not supported. You must re-run the Ambari Install Wizard and create a different name.

1.3. Resolving Cluster Deployment Problems

Try the recommended solution for each of the following problems:.

1.3.1. Problem: YARN ATS Component May Fail to Start When Installing HDP 2.1.3 with Ambari

If you install HDP 2.1.3 using Ambari, you must change the following YARN configuration property.

1.3.1.1. Solution:

Browse to **Ambari Web > Services > YARN > Configs** and modify the following property:

```
yarn.timeline-service.store-class=org.apache.hadoop.yarn.server.timeline.
LevelDbTimelineStore
```

1.3.2. Problem: Trouble Starting Ambari on System Reboot

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

1.3.2.1. 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
```

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

Charts appear incorrectly or not at all despite being available in the native Ganglia interface or Host health status is displayed incorrectly.

1.3.3.1. 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.

1.3.4. 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.

1.3.4.1. Solution:

Upgrade to Python version 2.6.8-0.15.1 .

1.3.5. 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.

1.3.5.1. 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>
```

1.3.6. 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.

1.3.6.1. 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
```

1.3.7. Problem: Some graphs do not show a complete hour of data until the cluster has been running for an hour

When a cluster is first started, some graphs, like **Services View -> HDFS** and **Services View -> MapReduce**, do not plot a complete hour of data, instead showing data only for the length of time the service has been running. Other graphs display the run of a complete hour.

1.3.7.1. Solution

Let the cluster run. After an hour all graphs will show a complete hour of data.

1.3.8. 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.

1.3.8.1. 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.

1.3.9. Problem: Service Fails with Unknown Host Exception

JVM `networkaddress.cache.negative.ttl` default setting of 10 (never cache) may result in DNS failure. Long, or multiple queries running on the JVM may fail. Occurs in Java 6,7, and 8.

1.3.9.1. Solution

Appropriate values for `networkaddress.cache.negative.ttl` depend on various system factors, including network traffic, cluster size, and resource availability. You can set Java VM options in an Ambari-installed cluster using the following procedure:

1. Edit the template for `hadoop-env.sh` file. Ambari deploys the template file on your cluster in the following location:

```
/var/lib/ambari-server/resources/stacks/{stackName}/  
{stackVersion}/hooks/before-START/templates/hadoop-env.sh.j2
```

where `{stackName}` and `{stackVersion}` refer to your specific stack name and version.

2. Change the following line in the template to add options to all Hadoop processes, then save the file.

```
export HADOOP_OPTS="-Djava.net.preferIPv4Stack=true  
${HADOOP_OPTS}"
```

3. Restart Ambari server.

```
ambari-server restart
```

4. Restart affected services, using the Ambari Web UI.

1.4. Resolving General Problems

1.4.1. 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
```

1.4.1.1. 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;
```

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

Removing a registered host not added to a cluster. `curl` command and REST API calls require a header element.

1.4.2.1. 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
```

1.4.3. Problem: Nagios alerts for CPU utilization show as UNKNOWN

Alerts for Master components such as "NameNode CPU Utilization" and "ResourceManager CPU Utilization" show as "UNKNOWN".

1.4.3.1. Solution

Nagios uses Simple Network Management Protocol (SNMP) to check CPU utilization. Be sure SNMP is started and port 161 is accessible between the Nagios Server and the hosts running the Master components.

```
service snmpd start
```