Hortonworks Data Platform

Installing Hadoop 2 Using Apache Ambari

(Aug 30, 2013)

docs.hortonworks.com

Hortonworks Data Platform : Installing Hadoop 2 Using Apache Ambari

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.

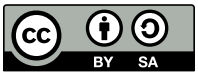

Table of Contents

List of Tables

1. Getting Ready to Install

This section describes the information and materials you need to get ready to install Hadoop 2 using the Apache Ambari Install Wizard. **Apache Ambari** provides an end-to-end management and monitoring application for Apache Hadoop. With Ambari, you can deploy and operate a complete Hadoop stack using a graphical user interface (GUI), manage configuration changes, monitor services, and create alerts for all the nodes in your cluster from a central point.

1.1. Understand the Basics

The Hortonworks Data Platform consists of three layers of components. A coordinated and tested set of these components is sometimes referred to as the Stack.

- **Core Hadoop 2**: The basic components of Apache Hadoop.
  - **Hadoop Distributed File System (HDFS)** : A special purpose file system designed to work to provides high-throughput access to data in a highly distributed environment.
  - **YARN**: A resource negotiator for managing high volume distributed data processing. Previously part of the first version of MapReduce.
  - **MapReduce 2 (MR2)** : A set of client libraries for computation using the MapReduce programming paradigm and a History Server for logging job and task information. Previously part of the first version of MapReduce.
- **Essential Hadoop**: A set of Apache components designed to ease working with Core Hadoop 2.
  - **Apache Pig** A platform for creating higher level data flow programs that can be compiled into sequences of MapReduce programs, using Pig Latin, the platform's native language.
  - **Apache Hive**: A tool for creating higher level SQL queries using HiveQL, the tool's native language, that can be compiled into sequences of MapReduce programs. Included with **Apache HCatalog**.
  - **Apache HCatalog**: A metadata abstraction layer that insulates users and scripts from how and where data is physically stored. Now part of **Apache Hive**. Includes **WebHCat**, which provides a set of REST APIs for HCatalog and related Hadoop components. Originally named **Templeton**.
  - **Apache HBase**: A distributed, column-oriented database that provides the ability to access and manipulate data randomly in the context of the large blocks that make up HDFS.
  - **Apache ZooKeeper**: A centralized tool for providing services to highly distributed systems. ZooKeeper is necessary for HBase installations.
- **Hadoop Support**: A set of components that allow you to monitor your Hadoop installation and to connect Hadoop with your larger compute environment.

- **Apache Oozie**: A server based workflow engine optimized for running workflows that execute Hadoop jobs.

  Running the current Oozie examples requires some reconfiguration from the standard Ambari installation. See Using HDP for Workflow and Scheduling (Oozie)

- **Apache Sqoop**: A component that provides a mechanism for moving data between Hadoop and external structured data stores. Can be integrated with Oozie workflows.

- **Apache Flume**: A log aggregator. This component must be installed manually. It is not supported in the context of Ambari at this time.

  See Installing and Configuring Flume for more information.

- **Ganglia**: An Open Source tool for monitoring high-performance computing systems.

- **Nagios**: An Open Source tool for monitoring systems, services, and networks.

You must always install HDFS, but you can select components from the other layers based on your needs.

1.2. Meet Minimum System Requirements

To run Hadoop, your system must meet minimum requirements.

- Hardware Recommendations
- Operating Systems Requirements
- Browser Requirements
- Software Requirements
- Database Requirements

1.2.1. Hardware Recommendations

There is no single hardware requirement set for installing Hadoop.

For more information on the parameters that may affect your installation, see Hardware Recommendations For Apache Hadoop.

1.2.2. Operating Systems Requirements

The following operating systems are supported:

- Red Hat Enterprise Linux (RHEL) v5.x or 6.x (64-bit) [6.x only for Beta]
- CentOS v5.x or 6.x (64-bit) [6.x only for Beta]
- Oracle Linux v5.x or 6.x [6.x only for Beta]

- SUSE Linux Enterprise Server (SLES) 11, SP1 (64-bit) [not supported for Beta]

Important

The installer pulls many packages from the base OS repos. If you do not have a complete set of base OS repos available to all your machines at the time of installation you may run into issues.

If you encounter problems with base OS repos being unavailable, please contact your system administrator to arrange for these additional repos to be proxied or mirrored. For more information see Optional: Configure the Local Repositories

1.2.3. Browser Requirements

The Ambari Install Wizard runs as a browser-based Web app. You must have a machine capable of running a graphical browser to use this tool. The supported browsers are:

- Windows (Vista, 7)
  - Internet Explorer 9.0 and higher (for Vista + Windows 7)
  - Firefox latest stable release
  - Safari latest stable release
  - Google Chrome latest stable release
- Mac OS X (10.6 or later)
  - Firefox latest stable release
  - Safari latest stable release
  - Google Chrome latest stable release
- Linux (RHEL, CentOS, SLES, Oracle Linux)
  - Firefox latest stable release
  - Google Chrome latest stable release

1.2.4. Software Requirements

On each of your hosts:

- yum
- rpm
- scp
- curl

- wget

Important

The Python version shipped with SUSE 11, 2.6.0-8.12.2, has a critical bug that may cause the Ambari Agent to fail with 24 hours. If you are installing on SUSE 11, please update all your hosts to Python version 2.6.8-0.15.1.

1.2.5. Database Requirements

Hive/HCatalog, Oozie, and Ambari all require their own internal databases.

- Hive/HCatalog: By default uses an Ambari-installed MySQL 5.x instance. With appropriate preparation, you can also use an existing MySQL 5.x or Oracle 11g r2 instance. See Using Non-Default Databasesfor more information on using existing instances.
- Oozie: By default uses an Ambari-installed Derby instance. With appropriate preparation, you can also use an existing MySQL 5.x or Oracle 11g r2 instance. See Using Non-Default Databases for more information on using existing instances.
- Ambari: By default uses an Ambari-installed PostgreSQL 8.x instance. With appropriate preparation, you can also use an existing Oracle 11g r2 instance. See Using Non-Default Databases for more information on using existing instances.

1.3. Decide on Deployment Type

While it is possible to deploy all of Hadoop on a single host, this is appropriate only for initial evaluation. In general you should use at least three hosts: one master host and two slaves.

1.4. Collect Information

To deploy your Hadoop installation, you need to collect the following information:

- The fully qualified domain name (FQDN) for each host in your system, and which components you want to set up on which host. The Ambari install wizard *does not* support using IP addresses. You can use `hostname -f` to check for the FQDN if you do not know it.
- The base directories you want to use as mount points for storing:
  - NameNode data
  - DataNodes data
  - Secondary NameNode data
  - Oozie data
  - YARN data
  - ZooKeeper data, if you install ZooKeeper

- Various log, pid, and db files, depending on your install type

1.5. Prepare the Environment

To deploy your Hadoop instance, you need to prepare your deploy environment:

- Check Existing Installs
- Set up Password-less SSH
- Enable NTP on the Cluster
- Check DNS
- Disable SELinux
- Disable iptables
- Disable PackageKit
- Check umask Value

1.5.1. Check Existing Installs

Ambari automatically installs the correct versions of the files that are necessary for Ambari and Hadoop to run. Versions other than the ones that Ambari uses can cause problems in running the installer, so remove any existing installs that do not match the following lists.

	RHEL/CentOS/Oracle Linux v5 **[Not supported for Beta]**	**RHEL/CentOS/Oracle Linux v6**	**SLES 11** **[Not supported for Beta]**
Ambari Server	• libffi 3.0.5-1.el5 • python26 2.6.8-2.el5 • python26-libs 2.6.8-2.el5 • postgresql 8.4.13-1.el6_3 • postgresql-libs 8.4.13-1.el6_3 • postgresql-server 8.4.13-1.el6_3	• postgresql 8.4.13-1.el6_3 • postgresql-libs 8.4.13-1.el6_3 • postgresql-server 8.4.13-1.el6_3	• libpq5 9.1.5-0.2.1 • postgresql 8.3.20-0.4.1 • postgresql-init 9.1-0.6.10.1 • postgresql-server 8.3.20-0.4.1
Ambari Agent [a]	• libffi 3.0.5-1.el5 • python26 2.6.8-2.el5 • python26-libs 2.6.8-2.el5	None	None
Nagios Server[b]	• nagios 3.5.0-99 • nagios-devel 3.5.0-99 • nagios-www 3.5.0-99 • nagios-plugins 1.4.9-1	• nagios 3.5.0-99 • nagios-devel 3.5.0-99 • nagios-www 3.5.0-99 • nagios-plugins 1.4.9-1	• nagios 3.5.0-99 • nagios-devel 3.5.0-99 • nagios-www 3.5.0-99 • nagios-plugins 1.4.9-1

	RHEL/CentOS/Oracle Linux v5 [Not supported for Beta]	RHEL/CentOS/Oracle Linux v6	SLES 11 [Not supported for Beta]
Ganglia Server[c]	• ganglia-gmetad 3.5.0-99 • ganglia-devel 3.5.0-99 • libganglia 3.5.0-99 • ganglia-web 3.5.7-99 • rrdtool 1.4.5-1.el5	• ganglia-gmetad 3.5.0-99 • ganglia-devel 3.5.0-99 • libganglia 3.5.0-99 • ganglia-web 3.5.7-99 • rrdtool 1.4.5-1.el6	• ganglia-gmetad 3.5.0-99 • ganglia-devel 3.5.0-99 • libganglia 3.5.0-99 • ganglia-web 3.5.7-99 • rrdtool 1.4.5-4.5.1
Ganglia Monitor[d]	• ganglia-gmond 3.5.0-99 • libganglia 3.5.0-99	• ganglia-gmond 3.5.0-99 • libganglia 3.5.0-99	• ganglia-gmond 3.5.0-99 • libganglia 3.5.0-99

[a]Installed on each host in your cluster. Communicates with the Ambari Server to execute commands.

[b]The host that runs the Nagios server

[c]The host that runs the Ganglia Server

[d]Installed on each host in the cluster. Sends metrics data to the Ganglia Collector.

1.5.2. Set Up Password-less SSH

To have Ambari Server automatically install Ambari Agents in all your cluster hosts, you must set up password-less SSH connections between the main installation (Ambari Server) host and all other machines. The Ambari Server host acts as the client and uses the key-pair to access the other hosts in the cluster to install the Ambari Agent.

Note

You can choose to install the Agents on each cluster host manually. In this case you do not need to setup SSH. See Appendix: Installing Ambari Agents Manually for more information.

1. Generate public and private SSH keys on the Ambari Server host

```
ssh-keygen
```

2. Copy the SSH Public Key (id_rsa.pub) to the root account on your target hosts.

```
.ssh/id_rsa
.ssh/id_rsa.pub
```

3. Add the SSH Public Key to the authorized_keys file on your target hosts.

```
cat id_rsa.pub >> authorized_keys
```

4. Depending on your version of SSH, you may need to set permissions on the .ssh directory (to 700) and the authorized_keys file in that directory (to 600) on the target hosts.

```
chmod 700 ~/.ssh
chmod 600 ~/.ssh /authorized_keys
```

5. From the Ambari Server, make sure you can connect to each host in the cluster using SSH.

```
ssh root@{remote.target.host}
```

You may see this warning. This happens on your first connection and is normal.

```
Are you sure you want to continue connecting (yes/no)?
```

6. Retain a copy of the SSH Private Key on the machine from which you will run the web-based Ambari Install Wizard.

Note

It is possible to use a non-root SSH account, if that account can execute `sudo` without entering a password.

1.5.3. Enable NTP on the Cluster and on the Browser Host

The clocks of all the nodes in your cluster and the machine that runs the browser through which you access Ambari Web must be able to synchronize with each other.

1.5.4. Check DNS

All hosts in your system must be configured for DNS and Reverse DNS.

If you are unable to configure DNS and Reverse DNS, you must edit the hosts file on every host in your cluster to contain the address of each of your hosts and to set the Fully Qualified Domain Name hostmane of each of those hosts. The following instructions cover basic hostname network setup for generic Linux hosts. Different versions and flavors of Linux might require slightly different commands. Please refer to your specific operating system documentation for the specific details for your system.

1.5.4.1. Edit the Host File

1. Using a text editor, open the hosts file on every host in your cluster. For example:

```
vi /etc/hosts
```

2. Add a line for each host in your cluster. The line should consist of the IP address and the FQDN. For example:

```
1.2.3.4 fully.qualified.domain.name
```

Note

Do **not** remove the following two lines from your host file, or various programs that require network functionality may fail.

```
127.0.0.1 localhost.localdomain localhost
::1 localhost6.localdomain6 localhost6
```

1.5.4.2. Set the Hostname

1. Use the "hostname" command to set the hostname on each host in your cluster. For example:

```
hostname fully.qualified.domain.name
```

2. Confirm that the hostname is set by running the following command:

```
hostname -f
```

This should return the name you just set.

1.5.4.3. Edit the Network Configuration File

1. Using a text editor, open the network configuration file on every host. This file is used to set the desired network configuration for each host. For example:

```
vi /etc/sysconfig/network
```

2. Modify the HOSTNAME property to set the fully.qualified.domain.name.

```
NETWORKING=yes
NETWORKING_IPV6=yes
HOSTNAME=fully.qualified.domain.name
```

1.5.5. Disable SELinux

SELinux must be temporarily disabled for the Ambari setup to function. Run the following command on each host in your cluster:

```
setenforce 0
```

1.5.6. Disable iptables

```
chkconfig iptables off
/etc/init.d/iptables stop
```

You can restart iptables after setup is complete.

1.5.7. Disable PackageKit

On the RHEL/CentOS installation host, if PackageKit is installed, open `/etc/yum/pluginconf.d/refresh-packagekit.conf` with a text editor and make this change:

```
enabled=0
```

Note

PackageKit is not enabled by default on SLES. Unless you have specifically enabled it, you do not need to do this step.

1.5.8. Check umask Value

Make sure umask is set to 022.

1.6. Optional: Configure the Local Repositories

If your cluster does **not** have access to the Internet, or you are creating a large cluster and you want to conserve bandwidth, you need to provide access to the bits using an alternative method.

1. Set up the local mirror repositories as needed for HDP and HDP Utils.

   For more information on your options, see Deploying HDP In Production Data Centers with Firewalls.

2. From the Ambari Server host, fetch the Ambari repository file or RPM package as described in Set Up the Bits. You need a connection to the Internet for this step.

   If you do not have a connection to the Internet for this machine, you should follow the instructions in Deploying HDP In Production Data Centers with Firewalls and be sure to perform the optional steps for setting up the Ambari local repository.

3. Configure Ambari Server so that it knows to connect to the mirrored repositories during installation.

   a. On Ambari Server, browse to the stacks definitions directory

   ```
   cd /var/lib/ambari-server/resources/stacks
   ```

   There are two stack definitions in this directory: HDP and HDPLocal. The HDP definition points to the publicly hosted HDP software packages. You must modify the HDPLocal definition to point to the local repositories you have set up.

   b. Browse to the stack HDPLocal 2.0.5 repos directory, for the 2.0.5 version of the stack..

   ```
   cd HDPLocal/2.0.5/repos
   ```

   c. Use a text editor to edit the repo info file. For example:

   ```
   vi repoinfo.xml
   ```

   d. You must update the <baseurl> value to point to your local repositories for each operating system that your cluster includes. So, for example, if your system includes hosts running CentOS 6, to point to the HDP 2.0.5 repositories, you would update stanzas to look something like this:

   ```
     <os type="centos6">
        <repo>
           <baseurl>http://{your.hosted.local.repository}/HDP-2.0.5/repos/
centos6</baseurl>
           <repoid>HDP-2.0.5</repoid>
          <reponame>HDP</reponame>
        </repo>
     </os>
   ```

   The appropriate relative path depends on how you have set up your local repos.

   

   Important

   If you have mixed operating systems in your cluster (for example, CentOS 6 and RHEL 6), you must configure the repositories and have properly edited <os type> stanzas for both OSes - centos6 and redhat6. If you do not, some hosts in your cluster will not be able to retrieve the software packages for their operating system.

   e. Save this file.

f. If you have not already installed the JDK on all hosts, download jdk-6u31-linux-x64.bin to `/var/lib/ambari-server/resources`.

g. If you have already installed the JDK on all hosts, you **must** use the option -j flag when running Ambari Server setup.

```
ambari-server setup –j /my/jdk/home
```

You must also provide the appropriate JDK path when running the Ambari install wizard. See Installing, Configuring and Deploying the Cluster: Install Options

2. Running the Installer

This section describes the process for installing Apache Ambari and preparing to deploy Hadoop.

2.1. Set Up the Bits

1. Log into the machine that is to serve as the Ambari Server as `root`. You may login and `sudo` as `su` if this is what your environment requires. This machine is the main installation host.

2. Download the the Ambari repository file and copy it to your repos.d.

Table 2.1. Download the repo

Platform	Access
RHEL, CentOS, and Oracle Linux 5 Not supported for Beta	wget http://private-repo-1.hortonworks.com/ambari-beta/centos5/1.x/beta/ambari.repo cp ambari.repo /etc/yum.repos.d
RHEL, CentOS and Oracle Linux 6	wget http://private-repo-1.hortonworks.com/ambari-beta/centos6/1.x/beta/ambari.repo cp ambari.repo /etc/yum.repos.d
SLES 11 Not supported for Beta	wget http://private-repo-1.hortonworks.com/ambari-beta/suse11/1.x/beta/ambari.repo cp ambari.repo /etc/zypp/repos.d

Note

If your cluster does not have access to the Internet, or you are creating a large cluster and you want to conserve bandwidth, you need to provide access to the bits using an alternative method. For more information, see Optional: Configure the Local Repositories section.

When you have the software, continue your installation based on your base platform.

2.1.1. RHEL/CentOS/Oracle Linux 5.x [Not supported for Beta]

Confirm the repository is configured by checking the repo list

```
yum repolist
```

You should see the Ambari and HDP utilities repositories in the list

```
  repo id                 repo name
| AMBARI-1.x              | Ambari 1.x
| HDP-UTILS-1.1.0.16      | Hortonworks Data Platform Utils
```

1. Install the Ambari bits using yum. This also installs PostgreSQL:

```
yum install ambari-server
```

2.1.2. RHEL/CentOS/Oracle Linux 6.x

1. Confirm the repository is configured by checking the repo list

```
yum repolist
```

You should see the Ambari and HDP utilities repositories in the list

```
  repo id                 repo name
| AMBARI-1.x            | Ambari 1.x
| HDP-UTILS-1.1.0.16    | Hortonworks Data Platform Utils
```

2. Install the Ambari bits using yum. This also installs PostgreSQL:

```
yum install ambari-server
```

2.1.3. SLES 11 [Not supported for Beta]

1. Confirm the downloaded repository is configured by checking the repo list:

```
zypper repos
```

You should see the Ambari and HDP utilities in the list:

```
# | Alias                 | Name
1 | AMBARI.dev-1.x        | Ambari 1.x
2 | HDP-UTILS-1.1.0.16    | Hortonworks Data Platform Utils
```

2. Install the Ambari bits using zypper. This also installs PostgreSQL:

```
zypper install ambari-server
```

2.2. Set Up the Server

The Ambari Server manages the install process. Run the Ambari Server setup and follow the prompts:

```
ambari-server setup
```

1. If you have *not* temporarily disabled SELinux, you may get a warning. Enter `y` to continue.

2. By default, Ambari Server runs under `root`. If you want to create a different user to run the Ambari Server instead, or to assign a previously created user, select `y` at **Customize user account for ambari-server daemon** and give the prompt the username you want to use.

3. If you have not temporarily disabled iptables, Setup does it for you. You should re-enable this after the installation is complete.

4. Agree to the Oracle JDK license when asked. You must accept this license to be able to download the necessary JDK from Oracle. The JDK is installed during the deploy phase.

Note

If you already have a local copy of the Oracle JDK v 1.6 update 31 64-bit binaries accessible from the install host, you can skip this and the next step. See Setup Options for more information. You can set the appropriate path during the Installing, Configuring and Deploying the Cluster: Install Options section of the install wizard.

5. At **Enter advanced database configuration**:

   - To use the default PostgreSQL database, named `ambari`, with the default username and password (`ambari/bigdata`), enter `n`.
   - To use an existing Oracle 11g r2 instance or to select your own database name, username and password for either database, enter `y`.
     - Select the database you want to use and provide any information required by the prompts, including hostname, port, Service Name or SID, username, and password.

Important

If you are using an existing Oracle instance, you must have prepared the instance using the steps detailed in Using Non-Default Databases **before** running the installer.

6. Setup completes.

2.2.1. Setup Options

The following table describes options frequently used for Ambari Server setup.

Option	Description
-j –java-home	Specifies the JAVA_HOME path to use on the Ambari Server and all hosts in the cluster. Use this option when you are using local repositories. For more information, see Optional: Configure the Local Repositories. This path must be valid on all hosts and you must also specify this path when performing your cluster install. See Installing, Configuring and Deploying the Cluster: Install Options for more information. For example: `ambari-server setup –j /usr/java/default` By default when you do not specify this option, Setup automatically downloads the JDK binary to `/var/lib/ambari-server/resources` and installs the JDK to `/usr/jdk64`.
-s –silent	Setup runs silently. Accepts all default prompt values.
-v	Prints verbose info and warning messages to the console during Setup.

Option	Description
–verbose	

2.3. Optional: Set Up LDAP or Active Directory Authentication

By default Ambari uses an internal database as the user store for authentication and authorization. If you want to add LDAP or Active Directory (AD) external authentication in addition for Ambari Web, you need to collect the following information and run a special setup command. Ambari Server must not be running when you execute this command. An LDAP client must be installed on the Ambari Server host.

Important

Ambari Server should not be running when you do this: either make the edits before you start Ambari Server the first time or bring the server down to make the edits.

1. The following table details the properties and values you need to know to set up LDAP authentication.

Note

If you are going to set `bindAnonymously` to false (the default), you need to make sure you have an LDAP Manager name and password set up. If you are going to use SSL, you need to make sure you have already set up your certificate and keys.

Table 2.2. Ambari Server LDAP Properties

Property	Values	Description
authentication.ldap.primaryUrl	server:port	The hostname and port for the LDAP or AD server. Example: my.ldap.server:389
authentication.ldap.secondaryUrl	server:port	The hostname and port for the secondary LDAP or AD server. Example: my.secondary.ldap.server:389 This is an optional value.
authentication.ldap.useSSL	true or false	If true, use SSL when connecting to the LDAP or AD server.
authentication.ldap. usernameAttribute	[LDAP attribute]	The attribute for username Example: uid
authentication.ldap.baseDn	[Distinguished Name]	The root Distinguished Name to search in the directory for users. Example: ou=people,dc=hadoop,dc=apache, dc=org

Property	Values	Description
authentication.ldap. bindAnonymously	true or false	If true, bind to the LDAP or AD server anonymously
authentication.ldap.managerDn	[Full Distinguished Name]	If Bind anonymous is set to false, the Distinguished Name ("DN") for the manager. Example: uid=hdfs,ou=people,dc=hadoop, dc=apache,dc=org
authentication.ldap. managerPassword	[password]	If Bind anonymous is set to false, the password for the manager

2. Run the LDAP setup command and answer the prompts with the information you collected above:

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

   a. At the **Primary URL*** prompt, enter the server URL and port you collected above. Prompts marked with an asterisk are required values.

   b. At the **Secondary URL** prompt, enter the secondary server URL and port. This is optional value.

   c. At the **Use SSL*** prompt, enter your selection.

   d. At the **User name attribute*** prompt, enter your selection. The default value is `uid`.

   e. At the **Base DN*** prompt, enter your selection.

   f. At the **Bind anonymously*** prompt, enter your selection.

   g. At the **Manager DN*** prompt, enter your selection if you have have set `bind.Anonymously` to false.

   h. At the **Enter the Manager Password*** , enter the password for your LDAP manager.

   i. Review your settings and if they are correct, select y.

   LDAP setup is complete.

Initially the users you have enabled all have Ambari User privileges. Ambari Users can read metrics, view service status and configuration, and browse job information. For these new users to be able to start or stop services, modify configurations, and run smoke tests, they need to be Admins. To make this change, use Ambari Web **Admin** -> **Users** -> **Edit**.

2.4. Optional: Set Up HTTPS for Ambari Web

If you want to limit access to the Ambari Web GUI to HTTPS connections, you need to provide a certificate. While it is possible to use a self-signed certificate for initial trials, they are not suitable for production environments. Once your certificate is in place, you must run a special setup command.

Important

Ambari Server should not be running when you do this. Either make these changes before you start Ambari the first time, or bring the server down before running the setup command.

1. Log into the Ambari Server host.

2. Locate your certificate. If you want to create a temporary self-signed certificate, use this as an example:

```
openssl genrsa -out $wserver.key 2048
openssl req -new -key $wserver.key -out $wserver.csr
openssl x509 -req -days 365 -in $wserver.csr -signkey $wserver.key -out
 $wserver.crt
```

   Where `$wserver` is the Ambari Server hostname.

Important

The certificate you use must be PEM-encoded, not DER-encoded. If you attempt to use a DER-encoded certificate, you see this error:

```
unable to load certificate
140109766494024:error:0906D06C:PEM routines:PEM_read_bio:no start
 line:pem_lib.c
:698:Expecting: TRUSTED CERTIFICATE
```

You can convert a DER-encoded certificate to a a PEM-encoded certificate using the following command:

```
openssl x509 -in cert.crt -inform der -outform pem -out cert.pem
```

where `cert.crt` is the DER-encoded certificate and `cert.pem` is the resulting PEM-encoded certificate.

3. Run the special setup command and answer the prompts

```
ambari-server setup-https
```

   a. Respond y to **Do you want to configure HTTPS**?

   b. Select the port you want to use for SSL. Default is 8443.

   c. Provide the path to your certificate and your private key.

   d. Provide the password for the private key.

   HTTPS setup for Ambari Web is complete.

2.5. Optional: Set Up HTTPS for Ganglia and Nagios

By default Ganglia and Nagios communicate with the Ambari Server using HTTP. If you want them to use HTTPS instead, use the following instructions.

Important

The servers should not be running when you do this: either make the edits before you start Ambari Server the first time or bring the servers down to make the edits.

2.5.1. Set Up Ganglia

Use the following instructions to set up HTTPS for Ganglia.

1. Set up the truststore for Ambari Server.
   a. Log into the Ambari Server host.
   b. Set the truststore path and password in `ambari.properties`. With a text editor, open:

      ```
      /etc/ambari-server/conf/ambari.properties
      ```

   c. Add the following lines.

      ```
      ssl.trustStore.path = $path-to-truststore
      ssl.trustStore.password = $password-for-truststore
      ```

2. Set up the Ganglia server.
   a. Log into the Ganglia server host.
   b. Create a self-signed certificate on the Ganglia server host. For example:

      ```
      openssl genrsa -out $gserver.key 2048
      openssl req -new -key $gserver.key -out $gserver.csr
      openssl x509 -req -days 365 -in $gserver.csr -signkey $gserver.key -out
       $gserver.crt
      ```

      Where `$gserver` is the Ganglia server hostname.

   c. Install SSL on the Ganglia server host.

      ```
      yum install mod_ssl
      ```

   d. Edit the SSL configuration file on the Ganglia server host.
      i. Using a text editor, open:

         ```
         /etc/httpd/conf.d/ssl.conf
         ```

      ii. Add lines setting the certificate and key file names to the files you created above [17]. For example:

```
SSLCertificateFile    = $gserver.crt
SSLCertificateKeyFile = $gserver.key
```

iii. Restart the `httpd` service on the Ganglia server host.

```
service httpd restart
```

3. Set up and restart the Ambari Server.

   a. Log into the Ambari Server.

   b. Add the cert to the truststore on the Ambari Server host.

   ```
   ambari-server setup-ganglia-https
   ```

   The setup script uses the truststore path and password you added to `ambari.properties` above. [17]

   c. Restart the Server

   ```
   ambari-server restart
   ```

2.5.2. Set Up Nagios

Use the following instructions to set up HTTPS for Nagios.

1. If you have not set up the truststore for Ambari Server previously.

   a. Log into the Ambari Server host.

   b. Set the truststore path and password in `ambari.properties`. With a text editor, open:

   ```
   /etc/ambari-server/conf/ambari.properties
   ```

   c. Add the following lines.

   ```
   ssl.trustStore.path = $path-to-truststore
   ssl.trustStore.password = $password-for-truststore
   ```

2. Set up the Nagios server.

   a. Log into the Nagios server host.

   b. Create a self-signed certificate on the Nagios server host. For example:

   ```
   openssl genrsa -out $nserver.key 2048
   openssl req -new -key $nserver.key -out $nserver.csr
   openssl x509 -req -days 365 -in $nserver.csr -signkey $nserver.key -out
    $nserver.crt
   ```

   Where `$nserver` is the Nagios server hostname.

   c. Install SSL on the Nagios server host.

   ```
   yum install mod_ssl
   ```

d. Edit the SSL configuration file on the Nagios server host.

i. Using a text editor, open:

```
/etc/httpd/conf.d/ssl.conf
```

ii. Add lines setting the certificate and key file names to the files you created above [18]. For example:

```
SSLCertificateFile    = $nserver.crt
SSLCertificateKeyFile = $nserver.key
```

iii. Restart the `httpd` service on the Nagios server host.

```
service httpd restart
```

3. Set up and restart the Ambari Server.

a. Log into the Ambari Server.

b. Add the cert to the truststore on the Ambari Server host.

```
ambari-server setup-nagios-https
```

The setup script uses the truststore path and password you added to `ambari.properties` above. [18]

c. Restart the Server

```
ambari-server restart
```

2.6. Optional: Encrypt Database and LDAP Passwords

By default the passwords for Ambari's database and for access to the LDAP server are stored in a plain text configuration file. To have those passwords encrypted, you need to run a special setup command.

Important

Ambari Server should not be running when you do this: either make the edits before you start Ambari Server the first time or bring the server down to make the edits.

1. Run the special setup command:

```
ambari-server encrypt-passwords
```

2. Provide a master key for encrypting the passwords. You are prompted to enter the key twice for accuracy.

Important

If your passwords are encrypted, you need access to the master key to start Ambari Server.

3. You have three options for maintaining the master key:
   - At the **Persist** prompt, select y. This stores the key in a file on the server.
   - Create an environment variable AMBARI_SECURITY_MASTER_KEY and set it to the key.
   - Provide the key manually at the prompt on server startup.

2.6.1. Reset Encryption

There may be situations in which you want to:

- Remove encryption entirely
- Change the current master key, either because the key has been forgotten or because you want to change the current key as a part of a security routine.

Important

Ambari Server should not be running when you do this.

2.6.1.1. Remove Encryption Entirely

To reset Ambari database and LDAP passwords to a completely unencrypted state:

1. On the Ambari host, open `/etc/ambari-server/conf/ambari.properties` with a text editor and set this property

   ```
   security.passwords.encryption.enabled=false
   ```

2. Delete `/var/lib/ambari-server/keys/credentials.jceks`
3. Delete `/var/lib/ambari-server/keys/master`
4. You must now reset the database password and, if necessary, the LDAP password. Run `ambari-server setup` and `ambari-server setup-ldap` again.

2.6.1.2. Change the Current Master Key

To change the master key:

- **If** you know the current master key or if the current master key has been persisted:
  1. Re-run the encryption setup command.

     ```
     ambari-server encrypt-passwords
     ```

2. Enter the current master key when prompted if necessary (if it is not persisted or set as an environment variable).
3. At the **Do you want to reset Master Key** prompt, enter **yes**.
4. At the prompt, enter the new master key.

- **If** you do **not** know the current master key:

1. Remove encryption entirely, as described here.
2. Re-run `ambari-server encrypt-passwords` as described here.
3. Restart the Ambari Server.

```
ambari-server restart
```

2.7. Optional: Set Up Two-Way SSL Between Ambari Server and Ambari Agents

Two-way SSL provides a way to encrypt communication between Ambari Server and Ambari Agents. By default Ambari ships with Two-way SSL disabled. To enable Two-way SSL:

Important

Ambari Server should not be running when you do this: either make the edits before you start Ambari Server the first time or bring the server down to make the edits.

1. On the Ambari Server host, open `/etc/ambari-server/conf/ambari.properties` with a text editor.
2. Add the following property:

```
security.server.two_way_ssl = true
```

The Agent certificates are downloaded automatically during Agent Registration.

2.8. Optional: Change the Ambari Server Port

By default Ambari uses port 8080 for access to Ambari Web and the REST API. If you want to change the port number, you need to edit the Ambari properties file.

Important

Ambari Server should not be running when you do this: either make the edits before you start Ambari Server the first time or bring the server down to make the edits.

1. On the Ambari Server host, open `/etc/ambari-server/conf/ambari.properties` with a text editor.

2. Add the client API port property and set it to your desired port value:

```
client.api.port=<port_number>
```

3. Start (or re-start) the Ambari Server. You can now access Ambari Web via the newly configured port:

```
http://{your.ambari.server}:<port_number>
```

2.9. Start the Ambari Server

- To start the Ambari Server:

```
ambari-server start
```

- To check the Ambari Server processes:

```
ps -ef | grep Ambari
```

- To stop the Ambari Server:

```
ambari-server stop
```

3. Installing, Configuring, and Deploying the Cluster

This section describes using the Ambari install wizard in your browser to complete your installation, configuration and deployment of Hadoop.

3.1. Log into Apache Ambari

Once you have started the Ambari service, you can access the Ambari Install Wizard through your browser.

1. Point your browser to `http://{main.install.hostname}:8080`.
2. Log in to the Ambari Server using the default username/password: admin/admin. You can change this later to whatever you want.

3.2. Welcome

The first step creates the cluster name.

1. At the **Welcome** page, type a name for the cluster you want to create in the text box. No whitespaces or special characters can be used in the name.
2. Click the **Next** button.

3.3. Select Stack

The Service Stack (or simply the Stack) is a coordinated and tested set of Hadoop components. Use the radio button to select the Stack version you want to install.

3.4. Install Options

In order to build up the cluster, the install wizard needs to know general information about how you want to set up your cluster. You need to supply the FQDN of each of your hosts. The wizard also needs to access the private key file you created in Set Up Password-less SSH It uses these to locate all the hosts in the system and to access and interact with them securely.

1. Use the **Target Hosts** text box to enter your list of host names, one per line. You can use ranges inside brackets to indicate larger sets of hosts. For example, for host01.domain through host10.domain use `host[01-10].domain`

   

   Note

   If you are deploying on EC2, use the **internal Private DNS** hostnames.

2. If you want to let Ambari automatically install the Ambari Agent on all your hosts using SSH, select **Provide your SSH Private Key** and either use the **Choose File** button in the **Host Registration Information** section to find the private key file that matches the

public key you installed earlier on all your hosts or cut and paste the key into the text box manually.

Note

If you are using IE 9, the **Choose File** button may not appear. Use the text box to cut and paste your private key manually.

Fill in the username for the SSH key you have selected. If you do not want to use `root`, you must provide the username for an account that can execute `sudo` without entering a password.

3. If you do not want Ambari to automatically install the Ambari Agents, select **Perform manual registration**. See Appendix: Installing Ambari Agents Manually for more information.
4. Advanced Options
   - If you want to use a local software repository (for example, if your installation does not have access to the Internet), check **Use a Local Software Repository**. For more information on using a local repository see Optional: Configure the Local Repositories
   - If you want to use an existing JDK rather than installing a fresh copy in the default location, check **Path to 64-bit JDK JAVA_HOME** and put the path in the text box. **Note**: this path must be valid on **all** the hosts in your cluster.
5. Click the **Register and Confirm** button to continue.

3.5. Confirm Hosts

This screen lets you confirm that Ambari has located the correct hosts for your cluster and to check those hosts to make sure they have the correct directories, packages, and processes to continue the install.

If any hosts were selected in error, you can remove them by selecting the appropriate checkboxes and clicking the grey **Remove Selected** button. To remove a single host, click the small white **Remove** button in the Action column.

At the bottom of the screen, you may notice a yellow box that indicates some warnings were encountered during the check process. For example, your host may have already had a copy of `wget` or `curl`. Click **Click here to see the warnings** to see a list of what was checked and what caused the warning.

Optionally, the administrator can use the host cleanup script provided by Ambari, which walks the user through the clearing up of any encountered warnings.

When you are satisfied with the list of hosts, click **Next**.

3.6. Choose Services

Hortonworks Data Platform is made up of a number of services. You must at a minimum install HDFS, but you can decide which of the other services you want to install. See Understand the Basics for more information on your options.

1. Select **all** to preselect all items or **minimum** to preselect only HDFS.
2. Use the checkboxes to unselect (if you have used **all)** or select (if you have used **minimum**) to arrive at your desired list of components.

Note

If you want to use Ambari for monitoring your cluster, make sure you select **Nagios** and **Ganglia.** If you do not select them, you get a warning popup when you finish this section. If you are using other monitoring tools, you can ignore the warning.

3. When you have made your selections, click **Next**.

3.7. Assign Masters

The Ambari install wizard attempts to assign the master nodes for various services you have selected to appropriate hosts in your cluster. The right column shows the current service assignments by host, with the hostname and its number of CPU cores and amount of RAM indicated.

1. To change locations, click the dropdown list next to the service in the left column and select the appropriate host.
2. To remove a ZooKeeper instance, click the green minus icon next to the host address you want to remove.
3. When you are satisfied with the assignments, click the **Next** button.

3.8. Assign Slaves and Clients

The Ambari install wizard attempts to assign the slave components (DataNodes, NodeManagers, and RegionServers) to appropriate hosts in your cluster. It also attempts to select hosts for installing the appropriate set of clients.

1. Use **all** or **none** to select all of the hosts in the column or none of the hosts, respectively.

   If a host has a red asterisk next to it, that host is also running one or more master components. Hover your mouse over the asterisk to see which master components are on that host.

2. Fine-tune your selections by using the checkboxes next to specific hosts.

Note

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 attempt to start it on the same host as the Ambari server, however, you need to start it with the -p option, as its default port is 8080 and that conflicts with the Ambari Web default port.

```
/usr/lib/hbase/bin/hbase-daemon.sh start rest -p
 <custom_port_number>
```

3. When you are satisfied with your assignments, click the **Next** button.

3.9. Customize Services

The **Customize Services** screen presents you with a set of tabs that let you manage configuration settings for Hadoop components. The wizard attempts to set reasonable defaults for each of the options here, but you can use this set of tabs to tweak those settings. and you are strongly encouraged to do so, as your requirements may be slightly different. Pay particular attention to the directories suggested by the installer.

Hover your mouse over each of the properties to see a brief description of what it does. The number of tabs you see is based on the type of installation you have decided to do. In a complete installation there are eight groups of configuration properties and other related options, such as database settings for Hive/HCat and Oozie, and admin name/password and alert email for Nagios.

The install wizard sets reasonable defaults for all properties except for those related to databases in the Hive/HCat tab and the Oozie tab, and two related properties in the Nagios tab. These four are marked in red and are the only ones you *must* set yourself.

Note

If you decide to use an existing database instance for Hive/HCatalog or for Oozie, you must have completed the preparations described in Using Non-Default Databases **prior** to running the install wizard.

Click the name of the group in each tab to expand and collapse the display.

3.9.1. Service Users and Groups

The individual services in Hadoop are each run under the ownership of a corresponding Unix account. These accounts are known as service users. These service users belong to a special Unix group. In addition there is a special service user for running smoke tests on components during installation and on-demand using the Management Header in the **Services** View of the Ambari Web GUI. Any of these users and groups can be customized using the **Misc** tab of the **Customize Services** step.

If you choose to customize names, Ambari checks to see if these custom accounts already exist. If they do not exist, Ambari creates them. The default accounts are always created during installation whether or not custom accounts are specified. These default accounts are not used and can be removed post-install.

Note

All new service user accounts, and any existing user accounts used as service users, must have a UID >= 1000.

Table 3.1. Service Users

Service	Component	Default User Account
HDFS	NameNode SecondaryNameNode	hdfs

Service	Component	Default User Account
	DataNode	
MapReduce2	HistoryServer	mapred
YARN	NodeManager ResourceManager	yarn
Hive	Hive Metastore HiveServer2	hive
HCat	HCatalog Server	hcat
WebHCat	WebHCat Server	hcat
Oozie	Oozie Server	oozie
HBase	MasterServer RegionServer	hbase
ZooKeeper	ZooKeeper	zookeeper
Ganglia	Ganglia Server Ganglia Collectors	nobody
Nagios	Nagios Server	nagios[a]
Smoke Test[b]	All	ambari-qa

[a]If you plan to use an existing user account named “nagios”, that “nagios” account must either be in a group named “nagios” or you must customize the Nagios Group.

[b]The Smoke Test user performs smoke tests against cluster services as part of the install process. It also can perform these on-demand from the Ambari Web GUI.

Table 3.2. Service Group

Service	Components	Default Group Account
All	All	hadoop
Nagios	Nagios Server	nagios
Ganglia	Ganglia Server Ganglia Collector	nobody

3.9.2. Properties That Depend on Service Usernames/ Groups

Some properties must be set to match specific service usernames or service groups. If you have set up non-default, customized service usernames for the HDFS or HBase service or the Hadoop group name, you must edit the following properties:

Table 3.3. HDFS Settings: Advanced

Property Name	Value
dfs.permissions.supergroup	The same as the HDFS username. The default is "hdfs"
dfs.cluster.administrators	A single space followed by the HDFS username.
dfs.block.local-path-access.user	The HBase username. The default is "hbase".

Table 3.4. MapReduce Settings: Advanced

Property Name	Value
mapreduce.tasktracker.group	The Hadoop group name. The default is "hadoop".

Property Name	Value
mapreduce.cluster.administrators	A single space followed by the Hadoop group name.

3.10. Review

The assignments you have made are displayed. Check to make sure everything is correct. If you need to make changes, use the left navigation bar to return to the appropriate screen.

To print your information for later reference, click **Print.**

When you are satisfied with your choices, click the **Deploy** button.

3.11. Install, Start and Test

The progress of the install is shown on the screen. Each component is installed and started and a simple test is run on the component. You are given an overall status on the process in the progress bar at the top of the screen and a host by host status in the main section.

To see specific information on what tasks have been completed per host, click the link in the **Message** column for the appropriate host. In the **Tasks** pop-up, click the individual task to see the related log files. You can select filter conditions by using the **Show** dropdown list. To see a larger version of the log contents, click the **Open** icon or to copy the contents to the clipboard, use the **Copy** icon.

Depending on which components you are installing, the entire process may take 40 or more minutes. Please be patient.

When **Successfully installed and started the services** appears, click **Next**.

3.12. Summary

The Summary page gives you a summary of the accomplished tasks. Click **Complete**. You are taken to the Ambari Web GUI.

4. Troubleshooting Ambari Deployments

The following information can help you troubleshoot issues you may run into with your Ambari-based installation.

4.1. Getting the Logs

The first thing to do if you run into trouble is to find the logs. Ambari Server logs are found at `/var/log/ambari-server/ambari-server.log` Ambari Agent logs are found at `/var/log/ambari-agent/ambari-agent.log` .

4.2. Quick Checks

- Make sure all the appropriate services are running. If you have access to Ambari Web, use the **Services View** to check the status of each component. If you do not have access to Manage Services, you must start and stop the services manually.
- If the first HDFS `put` command fails to replicate the block, the clocks in the nodes may not be synchronized. Make sure that Network Time Protocol (NTP) is enabled for your cluster.
- If HBase does not start, check if its slaves are running on 64-bit JVMs. Ambari requires that all hosts must run on 64-bit machines.
- Make sure umask is set to 0022.
- Make sure the HCatalog host can access the MySQL server. From a shell try:

  ```
  mysqld -h $FQDN_for_MySQL_server -u $FQDN_for_HCatalog_Server -p
  ```

  You will need to provide the password you set up for Hive/HCatalog during the installation process.
- Make sure MySQL is running. By default, MySQL server does not start automatically on reboot.

  To set auto-start on boot, from a shell, type:

  ```
  chkconfig --level 35 mysql on
  ```

  To then start the service manually from a shell, type:

  ```
  service mysqld start
  ```

4.3. Specific Issues

The following are common issues you might encounter.

4.3.1. Problem: Browser crashed before Install Wizard completed

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

4.3.1.1. 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.

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

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

4.3.2.1. 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 the **Mangement Header**.

- 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 the browser's cache.
  4. Re-run the entire Install Wizard.

4.3.3. Problem: "Unable to create new native thread" exceptions in HDFS DataNode logs or those of any system daemon

If your nproc limit is incorrectly configured, the smoke tests fail and you see an error similar to this in the DataNode logs:

```
INFO org.apache.hadoop.hdfs.DFSClient: Exception
increateBlockOutputStream java.io.EOFException
INFO org.apache.hadoop.hdfs.DFSClient: Abandoning block
blk_-6935524980745310745_139190
```

4.3.3.1. 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. Using a text editor, open `/etc/security/limits.d/90-nproc.conf` and change the `nproc` limit to approximately 32000. For more information, see ulimit and nproc recommendations for HBase servers.
2. Restart the HBase server.

4.3.4. Problem: The "yum install ambari-server" Command Fails

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

4.3.4.1. Solution:

You may have incompatible versions of some software components in your environment. Check the list in Check Existing Installs and make any necessary changes. Also make sure you are running a Supported Operating System

4.3.5. 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
```

4.3.5.1. 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.

4.3.6. 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
```

4.3.6.1. 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
   atjava.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 hostname 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.

4.3.7. Problem: MySQL and Nagios fail to install on RightScale CentOS 5 images on EC2

When using a RightScale CentOS 5 AMI on Amazon EC2, in certain cases MySQL and Nagios will fail to install. The MySQL failure is due to a conflict with the pre-installed MySQL and the use of the RightScale EPEL repository (error "Could not find package mysql-server"). Nagios fails to install due to conflicts with the RightScale php-common library.

4.3.7.1. Solution:

On the machines that will host MySQL and Nagios as part of your Hadoop cluster, perform the following:

1. Remove the existing MySQL server

```
yum erase MySQL-server-community
```

2. Install MySQL server with a disabled RightScale EPEL repository

```
yum install mysql-server --disable-repo=rightscale-epel
```

3. Remove the php-common library

```
yum erase php-common-5.2.4-RightScale.x86
```

4.3.8. Problem: Trouble starting Ambari on system reboot

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

4.3.8.1. 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
```

4.3.9. 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.

4.3.9.1. 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.

4.3.10. 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 bug that causes this crash.

4.3.10.1. Solution:

Upgrade to Python version 2.6.8-0.15.1

4.3.11. 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.

4.3.11.1. 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>
```

4.3.12. 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.

4.3.12.1. 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
   ```

4.3.13. 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.

4.3.13.1. Solution

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

4.3.14. Problem: After performing a cluster install the Nagios server is not started

The Nagios server is not started after a cluster install and you are unable to manage it from Ambari Web.

4.3.14.1. Solution

1. Log into the Nagios server host.
2. Confirm that the Nagios server is not running. From a shell:

   ```
   ps -ef | grep nagios
   ```

   You should not see a Nagios process running.
3. Start the Nagios process manuallly. From a shell:

   ```
   service nagios start
   ```

4. The server starts. You should be able to see that started state reflected in Ambari Web. You can now manage (start/stop) Nagios from Ambari Web.

4.3.15. 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`.

4.3.15.1. Solution

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

4.3.16. Problem: Updated configuration changes are not pushed to client/gateway nodes

Currently configuration changes are only pushed to daemon running nodes, so any changes are not automatically pushed to client only nodes such as gateway nodes.

4.3.16.1. Solution

Copy the files to the client nodes manually.

5. Appendix: Installing Ambari Agents Manually

In some situations you may decide you do not want to have the Ambari Install Wizard install and configure the Agent software on your cluster hosts automatically. In this case you can install the software manually.

Before you begin: on every host in your cluster download the HDP repository as described in Set Up the Bits.

5.1. RHEL/CentOS/Oracle Linux 5.x [Not supported for Beta] and 6.x

1. Install the Ambari Agent

```
yum install ambari-agent
```

2. Using a text editor, Configure the Ambari Agent by editing the `ambari-agent.ini` file. For example:

```
vi  /etc/ambari-agent/conf/ambari-agent.ini

[server]
hostname={your.ambari.server.hostname}
url_port=8440
secured_url_port=8441
```

3. Start the agent. The agent registers with the Server on start.

```
ambari-agent start
```

5.2. SLES [Not supported for Beta]

1. Install the Ambari Agent

```
zypper install ambari-agent
```

2. Configure the Ambari Agent by editing the `ambari-agent.ini` file.

```
vi /etc/ambari-agent/conf/ambari-agent.ini

[server]
hostname={your.ambari.server.hostname}
url_port=8440
secured_url_port=8441
```

3. Start the agent. The agent registers with the Server on start.

```
ambari-agent start
```

6. Appendix: Using Custom Hostnames

Use the following instructions to use custom hostnames in your cluster:

1. On the **Install Options** screen, select **Perform Manual Registration** for Ambari Agents.
2. Install the Agents manually as described in Installing Ambari Agents Manually.
3. For every host, create a script (for example named `/tmp/hostname.sh`) to echo the custom name you want to use for that host. For example:

```
#!/bin/sh
echo <ambari_hostname>
```

4. With a text editor, open `/etc/ambari-agent/conf/ambari-agent.ini` on every host and add the following information:

Table 6.1. ambari-agent.ini

Section	Value
[server]	Change the hostname to the host for the Ambari Server. This is the server that the Agent registers to.
[agent]	Add this line to the agent section: `hostname_script=/tmp/hostname.sh` (or whatever you have named your script)

5. Add the hostnames to `/etc/hosts` on all nodes.

7. Appendix: Upgrading Operating Systems on an Ambari-based Hadoop Installation

Ambari requires specific versions of the files for components that it uses. There are three steps you should take to make sure that these versions continue to be available:

- Disable automatic OS updates
- Do not update any HDP components such as MySQL, Ganglia, etc.
- If you must perform an OS update, do a manual kernel update only.

8. Appendix: Configuring Ports

The tables below specify which ports must be opened for which ecosystem components to communicate with each other. Make sure the appropriate ports are opened before you install Hadoop.

8.1. HDFS Ports

The following table lists the default ports used by the various HDFS services.

Table 8.1. HDFS Ports

Service	Servers	Default Ports Used	Protocol	Description	Need End User Access?	Configuration Parameters
NameNode WebUI	Master Node hosts (NameNode and any back-up NameNodes)	50070	http	Web UI to look at current status of HDFS, explore file system	Yes (Typically admins, Dev/ Support teams)	`dfs.namdnode.http-address`
		50470	https	Secure http service		`dfs.namenode.https-address`
NameNode metadata service	Master Node hosts (NameNode and any back-up NameNodes)	8020/9000	IPC	File system metadata operations	Yes (All clients who directly need to interact with the HDFS)	Embedded in URI specified by `fs.defaultFS`
DataNode	All Slave Node hosts	50075	http	DataNode WebUI to access the status, logs etc.	Yes (Typically admins, Dev/ Support teams)	`dfs.datanode.http.address`
		50475	https	Secure http service		`dfs.datanode.https.address`
		50010		Data transfer		`dfs.datanode.address`
		0.0.0.0:8010	IPC	Metadata operations	No	`dfs.datanode.ipc.address`

Service	Servers	Default Ports Used	Protocol	Description	Need End User Access?	Configuration Parameters
Secondary NameNode	Secondary NameNode and any backup Secondary NameNode hosts	50090	http	Checkpoint for NameNode metadata	No	`dfs.namenode.secondary.http address`

8.2. MapReduce Ports

The following table lists the default port used by the History Server WebUI.

Table 8.2. MapReduce Ports

Service	Servers	Default Ports Used	Protocol	Description	Need End User Access?	Configuration Parameters
History Server WebUI		19888	http	Web UI for Job History	Yes	`mapreduce.jobhistory.webapp`

8.3. YARN Ports

The following table lists the default ports used by YARN.

Table 8.3. YARN Ports

Service	Servers	Default Ports Used	Protocol	Description
YARN Resource Manager		8025		
YARN RM Admin		8141		The address of interface
Container Manager		0.0.0.0:45454		The address of manager in the
Applications Manager		8050		The address of applications ma interface in the

8.4. Hive Ports

The following table lists the default ports used by the various Hive services.

Note

Neither of these services is used in a standard HDP installation.

Table 8.4. Hive Ports

Service	Servers	Default Ports Used	Protocol	Description	Need End User Access?	Configuration Parameters
Hive Server2	Hive Server machine (Usually a utility machine)	10000	thrift	Service for programatically (Thrift/JDBC) connecting to Hive	Yes (Clients who need to connect to Hive either programatically	ENV `Variable HIVE_PORT`

Service	Servers	Default Ports Used	Protocol	Description	Need End User Access?	Configuration Parameters
					or through UI SQL tools that use JDBC)	
Hive Metastore		9083	thrift	Service for accesing metadata about Hive tables and partitions.*	Yes (Clients that run Hive, Pig and potentially M/R jobs that use HCatalog)	`hive.metastore.uris`

* To change the metastore port, use this `hive` command: `hive --service metastore -p port_number`

8.5. HBase Ports

The following table lists the default ports used by the various HBase services.

Table 8.5. HBase Ports

Service	Servers	Default Ports Used	Protocol	Description	Need End User Access?	Configuration Parameters
HMaster	Master Node hosts (HBase Master Node and any back-up HBase Master node)	60000			Yes	`hbase.master.port`
HMaster Info Web UI	Master Node hosts (HBase master Node and back up HBase Master node if any)	60010	http	The port for the HBase-Master web UI. Set to -1 if you do not want the info server to run.	Yes	`hbase.master.info.port`
Region Server	All Slave Node hosts	60020			Yes (Typically admins, dev/support teams)	`hbase.regionserver.port`
Region Server	All Slave Node hosts	60030	http		Yes (Typically admins, dev/support teams)	`hbase.regionserver.info.por`
	All ZooKeeper Node hosts	2888		Port used by ZooKeeper peers to talk to each other.See here for more information.	No	`hbase.zookeeper.peerport`
	All ZooKeeper Node hosts	3888		Port used by ZooKeeper peers to talk to each other.See here for more information.		`hbase.zookeeper.leaderport`
		2181		Property from ZooKeeper's config `zoo.cfg`. The port at which		`hbase.zookeeper.property.cl`

Service	Servers	Default Ports Used	Protocol	Description	Need End User Access?	Configuration Parameters
				the clients will connect.		

8.6. WebHCat Port

The following table lists the default port used by the WebHCat service.

Table 8.6. WebHCat Port

Service	Servers	Default Ports Used	Protocol	Description	Need End User Access?	Configuration Parameters
WebHCat Server	Any utility machine	50111	http	Web API on top of HCatalog and other Hadoop services	Yes	`templeton.port`

8.7. Ganglia Ports

The following table lists the default ports used by the various Ganglia services.

Table 8.7. Ganglia Ports

Service	Servers	Default Ports Used	Protocol	Description	Need End User Access?	Configuration Parameters
Ganglia Server	Ganglia server host	8660/61/62/63		For metric (gmond) collectors	No	
Ganglia Monitor	All Slave Node hosts	8660		For monitoring (gmond) agents	No	
Ganglia Server	Ganglia server host	8651		For ganglia gmetad		
Ganglia Web	Ganglia server host		http[a]			

[a]See Optional: Set Up HTTPS for Ganglia for instructions on enabling HTTPS.

8.8. MySQL Port

The following table lists the default port used by the MySQL service.

Table 8.8. MySQL Port

Service	Servers	Default Ports Used	Protocol	Description	Need End User Access?	Configuration Parameters
MySQL	MySQL database server host	3306				

8.9. Ambari Ports

The following table lists the default ports used by Ambari.

Table 8.9. Ambari Web

Service	Servers	Default Ports Used	Protocol	Description	Need End User Access?	Configuration Parameters
Ambari Server	Ambari Server host	8080[a]	http[b]	Interface to Ambari Web and Ambari REST API	No	
Ambari Server	Ambari Server host	8440	https	Handshake Port for Ambari Agents to Ambari Server	No	
Ambari Server	Ambari Server host	8441	https	Registration and Heartbeat Port for Ambari Agents to Ambari Server	No	

[a]See Optional: Change the Ambari Server Port for instructions on changing the default port.

[b]See Optional: Set Up HTTPS for Ambari Web for instructions on enabling HTTPS.

8.10. Nagios Ports

The following table lists the default port used by Ambari Web.

Table 8.10. Ambari Web

Service	Servers	Default Ports Used	Protocol	Description	Need End User Access?	Configuration Parameters
Nagios Server	Nagios server host	80	http[a]	Nagios Web UI	No	

[a]See Optional: Set Up HTTPS for Nagios for instructions on enabling HTTPS.

9. Appendix: Moving the Ambari Server

Use the following instructions to transfer the Ambari Server to a new host.

Note

These steps describe moving the Ambari Server when it uses the default PostgreSQL database. If you are using a non-default database for Ambari (such as Oracle), adjust the database backup, restore and stop/start procedures to match that database.

1. Back up all current data from the original Ambari Server and MapReduce databases.
2. Update all Agents to point to the new Ambari Server.
3. Install the Server on a new host and populate databases with information from original Server.

9.1. Back up Current Data

1. Stop the original Ambari Server.

```
ambari-server stop
```

2. Create a directory to hold the database backups.

```
cd /tmp
mkdir dbdumps
cd dbdumps/
```

3. Create the database backups.

```
pg_dump -U $AMBARI-SERVER USERNAME ambari > ambari.sql Password: $AMBARI-
SERVER PASSWORD
pg_dump -U $MAPRED USERNAME ambarirca > ambarirca.sql Password: $MAPRED
 PASSWORD
```

Where usernames and passwords reflect your particular installation. Defaults are `ambari-server/bigdata` and `mapred/mapred`.

9.2. Update Agents

1. On each Agent node, stop the Agent.

```
ambari-agent stop
```

2. Remove old Agent certificates.

```
rm /var/lib/ambari-agent/keys/*
```

3. Using a text editor, edit `/etc/ambari-agent/conf/ambari-agent.ini` to point to the new host.

```
[server]
```

```
hostname=$NEW FULLY.QUALIFIED.DOMAIN.NAME
url_port=8440
secured_url_port=8441
```

9.3. Install the New Server and Populate the Databases

1. On the new host, install the Server as described in Running the Installer, Sections 2.1 and 2.2.
2. Stop the Server so that you can copy the old database data to the new Server.

   ```
   ambari-server stop
   ```

3. Restart the PostgreSQL instance.

   ```
   service postgresql restart
   ```

4. Open the PostgreSQL interactive terminal.

   ```
   su - postgres
   psql
   ```

5. Using the interactive terminal, drop the databases created by the fresh install.

   ```
   drop database ambari;
   drop database ambarirca;
   ```

6. Check to make sure the databases have been dropped.

   ```
   /list
   ```

   The databases should not be listed.

7. Create new databases to hold the transferred data.

   ```
   create database ambari;
   create database ambarirca;
   ```

8. Exit the interactive terminal

   ```
   ^d
   ```

9. Copy the saved data from Back up Current Data to the new Server.

   ```
   cd /tmp
   scp -i <ssh-key> root@<original-server>/tmp/dbdumps/*.sql/tmp
   (Note: compress/transfer/uncompress as needed from source to dest)
   psql -d ambari -f /tmp/ambari.sql
   psql -d ambarirca -f /tmp/ambarirca.sql
   ```

10. Start the new Server.

    ```
    <exit to root>
    ambari-server start
    ```

11. On each Agent host, start the Agent.

```
ambari-agent start
```

12. Open Ambari Web. Point your compatible browser to:

```
<new_Ambari_Server>:8080
```

13. Go to **Services** -> **MapReduce** and use the Management Header to **Stop** and **Start** the MapReduce service.

14. Start other services as necessary.

The new Server is ready to use.

10. Appendix: Using Non-Default Databases

Use the following instructions to prepare a non-default database for Hive/HCatalog, Oozie, or Ambari. You **must** complete these instructions before you setup the Ambari Server by running `ambari-server setup`.

10.1. Hive/HCatalog

1. On the Hive Metastore machine, install the appropriate JDBC .jar file:
   - For **Oracle**:
     a. Download the Oracle JDBC (OJDBC) driver from http://www.oracle.com/technetwork/database/features/jdbc/index-091264.html.

        Select Oracle Database 11g Release 2 - ojdbc6.jar

     b. Copy the .jar file to the Java share directory

        ```
        cp ojdbc6.jar /usr/share/java
        ```

     c. Make sure the .jar file has the appropriate permissions - 644.
   - For **MySQL**:
     a. Install the connector.
        - RHEL/CentOS/Oracle Linux [6.x only supported for Beta]

          ```
          yum install mysql-connector-java-5.0.8-1
          ```

        - SLES [Not supported for Beta]

          ```
          zypper install mysql-connector-java-5.0.8-1
          ```

     b. Confirm that the MySQL .jar file is in the Java share directory

        ```
        ls /usr/share/java/mysql-connector-java.jar
        ```

     c. Make sure the .jar file has the appropriate permissions - 644.
2. On the Ambari Server host, install the appropriate JDBC .jar file:
   - For **Oracle**:
     a. Download the Oracle JDBC (OJDBC) driver from http://www.oracle.com/technetwork/database/features/jdbc/index-091264.html.

        Select Oracle Database 11g Release 2 - ojdbc6.jar

     b. Copy the .jar file to the Java share directory

```
cp ojdbc6.jar /var/lib/ambari-server/resources
```

c. Make sure the .jar file has the appropriate permissions - 644.

- For **MySQL**:

a. Download the mysql connector driver from the host on which you installed mysql-connector-java.

b. Copy the .jar file to the Java share directory

```
cp mysql-connector-jave.jar /var/lib/ambari-server/resources
```

c. Make sure the .jar file has the appropriate permissions - 644.

3. Create a user for Hive and grant it permissions:

- For **Oracle**, create the Hive user and grant it database permissions:

```
# sqlplus sys/root as sysdba
SQL> CREATE USER $HIVEUSER IDENTIFIED BY $HIVEPASSWORD;
SQL> GRANT SELECT_CATALOG_ROLE TO $HIVEUSER;
SQL> GRANT CONNECT, RESOURCE TO  $HIVEUSER;
SQL> QUIT;
```

Where *$HIVEUSER* is the Hive user name and *$HIVEPASSWORD* is the Hive user password.

- For **MySQL**, create the Hive user and grant it database permissions

```
# mysql -u root -p
mysql> CREATE USER '$HIVEUSER'@'%' IDENTIFIED BY '$HIVEPASSWORD';
mysql> GRANT ALL PRIVILEGES ON *.* TO '$HIVEUSER'@'%';
mysql> flush privileges;
```

Where *$HIVEUSER* is the Hive user name and *$HIVEPASSWORD* is the Hive user password.

4. For **Oracle** only: Load the Hive Metastore Schema

- The Hive Metastore database schema must be pre-loaded into your Oracle database using the schema script:

```
sqlplus $HIVEUSER/$HIVEPASSWORD < hive-schema-0.10.0.oracle.sql
```

The file `hive-schema-0.10.0.oracle.sql` is found in the `/var/lib/ambari-server/resources/` directory of the Ambari Server machine, once you have completed the Set Up the Bits step in the install process.

10.1.1. Troubleshooting Hive/HCatalog

Use these entries to help you troubleshoot any issues you might have installing Hive/HCatalog with non-default databases.

10.1.1.1. Problem: Hive Metastore Install Fails Using Oracle

Check the install log:

```
cp /usr/share/java/${jdbc_jar_name} ${target}] has failures: true
```

The Oracle JDBC .jar file cannot be found.

10.1.1.1.1. Soution

Make sure the file is in the appropriate directory on the Hive Metastore server and click **Retry**.

10.1.1.2. Problem: Install Warning when "Hive Check Execute" Fails Using Oracle

Check the install log:

```
java.sql.SQLSyntaxErrorException: ORA-01754:
a table may contain only one column of type LONG
```

The Hive Metastore schema was not properly loaded into the database.

10.1.1.2.1. Soution

Complete the install with the warning. Check your database to confirm the Hive Metastore schema is loaded. Once in the Ambari Web GIU, browse to **Services** > **Hive/HCat**. Use the Management Header to re-run the smoke test (**Maintenance** ->**Run Smoke Test**) to check that the schema is correctly in place.

10.2. Oozie

1. On the Oozie Server machine, install the appropriate JDBC .jar file:
   - For **Oracle**:
     a. Download the Oracle JDBC (OJDBC driver from http://www.oracle.com/technetwork/database/features/jdbc/index-091264.html.

        Select Oracle Database 11g Release 2 - ojdbc6.jar

     b. Copy the .jar file to the Java share directory

        ```
        cp ojdbc6.jar /usr/share/java
        ```

     c. Make sure the .jar file has the appropriate permissions - 644.
   - For **MySQL**:
     a. Install the connector.
        - RHEL/CentOS/Oracle Linux [5.x not supported in Beta]

```
yum install mysql-connector-java-5.0.8-1
```

- SLES [Not supported in Beta]

```
zypper install mysql-connector-java-5.0.8-1
```

b. Confirm that the MySQL .jar file is in the Java share directory

```
ls /usr/share/java/mysql-connector-java.jar
```

c. Make sure the .jar file has the appropriate permissions - 644.

2. On the Ambari Server host, install the appropriate JDBC .jar file:

- For **Oracle**:

a. Download the Oracle JDBC (OJDBC) driver from http://www.oracle.com/technetwork/database/features/jdbc/index-091264.html.

Select Oracle Database 11g Release 2 - ojdbc6.jar

b. Copy the .jar file to the Java share directory

```
cp ojdbc6.jar /var/lib/ambari-server/resources
```

c. Make sure the .jar file has the appropriate permissions - 644.

- For **MySQL**:

a. Download the mysql connector driver from the host on which you installed mysql-connector-java.

b. Copy the .jar file to the Java share directory

```
cp mysql-connector-jave.jar /var/lib/ambari-server/resources
```

c. Make sure the .jar file has the appropriate permissions - 644.

3. Create a user for Oozie and grant it permissions:

- For **Oracle**, create the Oozie user and grant it database permissions:

```
# sqlplus sys/root as sysdba
SQL> CREATE USER $OOZIEUSER IDENTIFIED BY $OOZIEPASSWORD;
SQL> GRANT ALL PRIVILEGES TO $OOZIEUSER;
SQL> QUIT;
```

Where *$OOZIEUSER* is the Oozie user name and *$OOZIEPASSWORD* is the Oozie user password.

- For **MySQL**

a. Create the Oozie user and grant it database permissions:

```
# mysql -u root -p
mysql> CREATE USER '$OOZIEUSER'@'%' IDENTIFIED BY '$OOZIEPASSWORD';
mysql> GRANT ALL PRIVILEGES ON *.* TO '$OOZIEUSER'@'%';
```

```
mysql> flush privileges;
```

Where *$OOZIEUSER* is the Oozie user name and *$OOZIEPASSWORD* is the Oozie user password.

b. Create the Oozie database:

```
# mysql -u root -p
mysql> CREATE DATABASE oozie;
```

10.2.1. Troubleshooting Oozie

Use these entries to help you troubleshoot any issues you might have installing Oozie with non-default databases.

10.2.1.1. Problem: Oozie Server Install Fails Using MySQL

Check the install log:

```
cp /usr/share/java/mysql-connector-java.jar
/usr/lib/oozie/libext/mysql-connector-java.jar]
has failures: true
```

The MySQL JDBC .jar file cannot be found.

10.2.1.1.1. Soution

Make sure the file is in the appropriate directory on the Oozie server and click **Retry**.

10.2.1.2. Problem: Oozie Server Install Fails Using Oracle or MySQL

Check the install log:

```
Exec[exec cd /var/tmp/oozie &&
/usr/lib/oozie/bin/ooziedb.sh create -sqlfile oozie.sql -run ]
has failures: true
```

Oozie was unable to connect to the database or was unable to successfully setup the schema for Oozie.

10.2.1.2.1. Soution

Check the database connection settings provided during the **Customize Services** step in the install wizard by browsing back to **Customize Services** -> **Oozie**. After confirming (and adjusting) your database settings, proceed forward with the install wizard.

If the Install Oozie Server continues to fail, get more information by connecting directly to the Oozie server and executing the following command as *$OOZIEUSER*:

```
su oozie
/usr/lib/oozie/bin/ooziedb.sh create -sqlfile oozie.sql -run
```

10.3. Ambari

1. On the Ambari Server machine, install the Oracle JDBC .jar file:

a. Download the Oracle JDBC (OJDBC) driver from http://www.oracle.com/technetwork/database/features/jdbc/index-091264.html.

   Select Oracle Database 11g Release 2 - ojdbc6.jar

b. Copy the .jar file to the Java share directory

```
cp ojdbc6.jar /usr/share/java
```

c. Make sure the .jar file has the appropriate permissions - 644.

2. Create the Ambari user, password, and tablespace, and grant the account database permissions:

```
# sqlplus sys/root as sysdba
SQL> create user $AMBARIUSER identified by $AMBARIPASSWORD default
                tablespace "USERS" temporary tablespace "TEMP";
SQL> grant unlimited tablespace to $AMBARIUSER;
SQL> grant create session to $AMBARIUSER;
SQL> grant create table to $AMBARIUSER;
SQL> quit;
```

   Where *$AMBARIUSER* is the Ambari user name and *$AMBARIPASSWORD* is the Ambari user password.

3. Load the Ambari Server schema:

- To set up Ambari Server to load the schema automatically:

  a. Download the Oracle Instant Client (for Linux x-86 or x86-64), Basic and the Instant Client Package - SQL*Plus, version 11.2.0.3.0, on the Ambari Server host.

     For information on the Oracle Database Instant Client, see here. To download the x86 client, see here. To download the x86-64 client, see here.

  b. Extract the zip files on your Ambari Server

```
mkdir /home/oracle
cd /home/oracle
unzip /tmp/instantclientsqlpluslinux.x6411.2.0.3.0.zip
unzip /tmp/instantclientbasiclinux.x6411.2.0.3.0.zip
```

  c. Update your PATH and LD_LIBRARY_PATH variables. For example, in BASH:

```
export PATH=/home/oracle/instantclient_11_2:${PATH}
export LD_LIBRARY_PATH=/home/oracle/instantclient_11_2:
${LD_LIBRARY_PATH}
```

- To load the schema manually, create the Ambari Server schema by running a script:

```
sqlplus $AMBARIUSER/$AMBARIPASSWORD <
/var/lib/ambari-server/resources/Ambari-DDL-Oracle-CREATE.sql
```

  The file `Ambari-DDL-Oracle-CREATE.sql` is found in the `/var/lib/ambari-server/resources/` directory of the Ambari Server machine, once you have completed the Set Up the Bits step in the install process. .

10.3.1. Troubleshooting Ambari

Use these entries to help you troubleshoot any issues you might have installing Ambari with an existing Oracle database.

10.3.1.1. Problem: Ambari Server Fails to Start: No Driver

Check `/var/log/ambari-server/ambari-server.log` for :

```
ExceptionDescription:Configurationerror.
Class[oracle.jdbc.driver.OracleDriver] not found.
```

The Oracle JDBC .jar file cannot be found.

10.3.1.1.1. Soution

Make sure the file is in the appropriate directory on the Ambari server and re-run `ambari-server setup`. See Step one above.

10.3.1.2. Problem: Ambari Server Fails to Start: No Connection

Check `/var/log/ambari-server/ambari-server.log` for :

```
The Network Adapter could not establish the connection
Error Code: 17002
```

Ambari Server cannot connect to the database.

10.3.1.2.1. Soution

Confirm the database host is reachable from the Ambari Server and is correctly configured by reading `/etc/ambari-server/conf/ambari.properties`

```
server.jdbc.url=jdbc:oracle:thin:
            @oracle.database.hostname:1521/ambaridb
server.jdbc.rca.url=jdbc:oracle:thin:
            @oracle.database.hostname:1521/ambaridb
```

10.3.1.3. Problem: Ambari Server Fails to Start: Bad Username

Check `/var/log/ambari-server/ambari-server.log` for :

```
Internal Exception: java.sql.SQLException: ORA01017:
invalid username/password; logon denied
```

You are using an invalid username/password.

10.3.1.3.1. Soution

Confirm the user account is set up in the database and has the correct privileges. See Step 2 above.

10.3.1.4. Problem: Ambari Server Fails to Start: No Schema

Check `/var/log/ambari-server/ambari-server.log` for :

```
 Internal Exception: java.sql.SQLSyntaxErrorException: ORA00942:
table or view does not exist
```

The schema has not been loaded.

10.3.1.4.1. Soution

Confirm you have loaded the database schema. See Step 3 above.