Hortonworks Data Platform

Installing HDP Using Apache Ambari

(Jan 14, 2013)

docs.hortonworks.com

Hortonworks Data Platform : Installing HDP 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

1. Getting Ready to Install 1
1.1. Understand the Basics 1
1.2. Meet Minimum System Requirements 2
1.2.1. Hardware Recommendations 2
1.2.2. Operating Systems Requirements 2
1.2.3. Browser Requirements 3
1.2.4. Software Requirements 3
1.2.5. Database Requirements 4
1.3. Decide on Deployment Type 4
1.4. Collect Information 4
1.5. Prepare the Environment 4
1.5.1. Check Existing Installs 5
1.5.2. Set Up Password-less SSH 5
1.5.3. Enable NTP on the Cluster 6
1.5.4. Check DNS 6
1.5.5. Disable SELinux 6
1.5.6. Disable iptables 7
1.6. Optional: Configure the Local Repositories 7
2. Running the Installer 9
2.1. Set Up the Bits 9
2.1.1. RHEL/CentOS 5.x 9
2.1.2. RHEL/CentOS 6.x 10
2.1.3. SLES 11 10
2.2. Set Up the Server 10
2.2.1. Setup Options 11
2.3. Start the Ambari Server 11
3. Installing, Configuring, and Deploying the Cluster 12
3.1. Log into Apache Ambari 12
3.2. Welcome 12
3.3. Install Options 12
3.4. Confirm Hosts 13
3.5. Choose Services 13
3.6. Assign Masters 14
3.7. Assign Slaves and Clients 14
3.8. Customize Services 14
3.8.1. HDFS 15
3.8.2. MapReduce 17
3.8.3. Hive/HCat 20
3.8.4. WebHCat 21
3.8.5. HBase 22
3.8.6. ZooKeeper 23
3.8.7. Oozie 24
3.8.8. Nagios 25
3.8.9. Misc 25
3.8.10. Recommended Memory Configurations for the MapReduce Service 26
3.9. Review 26
3.10. Install, Start and Test 27

3.11. Summary 27
4. Troubleshooting Ambari Deployments 28
4.1. Getting the Logs 28
4.2. Quick Checks 28
4.3. Specific Issues 28
4.3.1. Problem: Browser crashed before Install Wizard completed 29
4.3.2. Problem: Install Wizard reports that the cluster install has failed 29
4.3.3. Problem: “Unable to create new native thread” exceptions in HDFS DataNode logs or those of any system daemon 30
4.3.4. Problem: The “yum install ambari-server” Command Fails 30
4.3.5. Problem: HDFS Smoke Test Fails 30
4.3.6. Problem: The HCatalog Daemon Metastore Smoke Test Fails 31
4.3.7. Problem: MySQL and Nagios fail to install on RightScale CentOS 5 images on EC2 31
4.3.8. Trouble starting Ambari on system reboot 32
5. Appendix: Installing Ambari Agents Manually 33
5.1. RHEL/CentOS v. 5.x and 6.x 33
5.2. SLES 33

List of Tables

2.1. Download the repo 9
3.1. HDFS Settings:NameNode 15
3.2. HDFS Settings:SNameNode 15
3.3. HDFS Settings:DataNodes 15
3.4. HDFS Settings:General 15
3.5. HDFS Settings:Advanced 16
3.6. MapReduce Settings: JobTracker 17
3.7. MapReduce Settings: TaskTracker 18
3.8. MapReduce Settings: General 18
3.9. MapReduce Settings: Advanced 18
3.10. Hive/HCat Settings: Hive Metastore 20
3.11. Hive/HCat Settings: Advanced Settings 20
3.12. WebHCat Settings: Advanced Settings 21
3.13. HBase Settings: HBase Master 22
3.14. HBase Settings: RegionServer 22
3.15. HBase Settings: General 22
3.16. HBase Settings: Advanced 23
3.17. ZooKeeper Settings: ZooKeeper Server 23
3.18. ZooKeeper Settings: Advanced 23
3.19. Oozie Settings: Oozie Server 24
3.20. Oozie Settings: Advanced 24
3.21. Nagios Settings:General 25
3.22. Misc Settings:Users/Groups 25

1. Getting Ready to Install

This section describes the information and materials you need to get ready to install the Hortonworks Data Platform (HDP) 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.

- **Core Hadoop**: The basic components of Apache Hadoop.
  - **Hadoop Distributed File System (HDFS)**: A special purpose file system that is designed to work the MapReduce engine. It provides high-throughput access to data in a highly distributed environment.
  - **MapReduce**: A framework for performing high volume distributed data processing using the MapReduce programming paradigm.
- **Essential Hadoop**: : A set of Apache components designed to ease working with Core Hadoop.
  - **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-like queries using HiveQL, the tool's native language, that can be compiled into sequences of MapReduce programs.
  - **Apache HCatalog**: A metadata abstraction layer that insulates users and scripts from how and where data is physically stored.
  - **WebHCat**: A component that provides a set of REST-like 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.
- **HDP 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.

- **Apache Sqoop**: A component that provides a mechanism for moving data between HDP and external structured data stores. Can be integrated with Oozie workflows.
- **Apache Flume**: A log aggregator. This component must be installed manually. 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 the components from the other layers based on your needs. For more information on the structure of the HDP, see Understanding Hadoop Ecosystem.

1.2. Meet Minimum System Requirements

To run the Hortonworks Data Platform, your system must meet minimum requirements.

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

1.2.1. Hardware Recommendations

Although there is no single hardware requirement for installing HDP, there are some basic guidelines. You can see sample setups here: 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)
- CentOS v5.x or 6.x (64-bit)
- SUSE Linux Enterprise Server (SLES) 11, SP1 (64-bit)

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.

For example, if you are using RHEL 6 your hosts must be able to access the "Red Hat Enterprise Linux Server 6 Optional (RPMs)" repo. If this repo is disabled, the installation is unable to access the rubygems package, which is necessary for Ambari to operate.

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
  - 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)
  - Firefox latest stable release
  - Google Chrome latest stable release

1.2.4. Software Requirements

On each of your hosts:

- yum
- rpm
- scp
- curl
- wget
- pdsh

1.2.5. Database Requirements

Hive or HCatalog requires a MySQL database for its use. You can choose to use a current instance or let the Ambari install wizard create one for you.

1.3. Decide on Deployment Type

While it is possible to deploy all of HDP 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 HDP installation, you need to collect the following information:

- The fully qualified domain name (FQDN) for each host in your system, and which component(s) you wish 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 wish to use as mount points for storing:
  - NameNode data
  - DataNodes data
  - MapReduce data
  - ZooKeeper data, if you install ZooKeeper
  - Various log, pid, and db files, depending on your install type
- The hostname (for an existing instance), database name, username, and password for the MySQL instance, if you install Hive/HCatalog.

Note

If you are using an existing instance, the user you create for HDP's use must be granted all privileges.

1.5. Prepare the Environment

To deploy your HDP 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

1.5.1. Check Existing Installs

Ambari automatically installs the correct versions of the files that are necessary for Ambari and HDP 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 v5	RHEL/CentOS v6	SLES 11
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	• 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 3.2.3-2.el5 • nagios-plugins 1.4.15-2.el5 • nagios-common 2.12-10.el5	• nagios 3.2.3-2.el6 • nagios-plugins1.4.9-1	• nagios 3.2.3-2.1 • nagios-plugins 1.4.9-1 • nagios-www 3.2.3-2.1
Ganglia Collector[c]	• ganglia-gmetad 3.2.0-99 • rrdtool 1.4.5-1.el5	• ganglia-gmetad 3.2.0-99 • rrdtool 1.4.5-1.el6	• ganglia-gmetad 3.2.0-99 • rrdtool 1.4.5-4.5.1
Ganglia Monitor[d]	ganglia-gmond 3.2.0-99	ganglia-gmond 3.2.0-99	ganglia-gmond 3.2.0-99

[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 Collector 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. Depending on your version of SSH, you may need to set permissions on your .ssh directory (to 700) and the authorized_keys file in that directory (to 640).

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

3. Add the SSH Public Key to the authorized_keys file.

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

4. 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)?
```

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

1.5.3. Enable NTP on the Cluster

The clocks of all the nodes in your cluster 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.

Note

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.

1.5.5. Disable SELinux

SELinux must be disabled for Ambari to function. To temporarily disable SELinux, run the following command on each host in your cluster:

```
setenforce 0
```

Permanently disabling SELinux so that on system reboot it does not restart is strongly recommended. To do this, edit the SELinux config and set SELINUX to disabled. On each host:

```
vi /etc/selinux/config
```

```
# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#       enforcing - SELinux security policy is enforced.
#       permissive - SELinux prints warnings instead of enforcing.
#       disabled - SELinux is fully disabled.
SELINUX=disabled
# SELINUXTYPE= type of policy in use. Possible values are:
#       targeted - Only targeted network daemons are protected.
#       strict - Full SELinux protection.
SELINUXTYPE=targeted
```

1.5.6. Disable iptables

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

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

1. Using the instructions in the Firewalls document, set up the local mirror repositories as needed for HDP, HDP Utils and EPEL.

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 1.2.0 repos directory.

   ```
   cd HDPLocal/1.2.0/repos
   ```

   c. Edit the repo info file:

   ```
   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 and HDP-EPEL repositories, you would update stanzas to look something like this:

```
  <os type="centos6">
     <repo>
        <baseurl>http://{your.hosted.local.repository}/HDP-1.2.0/repos/
centos6</baseurl>
        <repoid>HDP-1.2.0</repoid>
       <reponame>HDP</reponame>
     </repo>
     <repo>
       <baseurl>http://{your.hosted.local.repository}/HDP-1.2.0/repos/
centos6</baseurl>
       <repoid>HDP-epel</repoid>
       <reponame>HDP-epel</reponame>
       <mirrorslist><![CDATA[http://mirrors.fedoraproject.org/mirrorlist?
repo=epel-6&arch=$basearch]]></mirrorslist>
     </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 the Hortonworks Data Platform. Ambari fetches the software packages from remote repositories over the Internet.

2.1. Set Up the Bits

1. Log into the machine which is to serve as the Ambari Server as `root`. This is the main installation host.

2. Download the HDP RPM Package or Repository File from the Hortonworks public repo.

Table 2.1. Download the repo

Platform	RPM Package	Repository File
RHEL and CentOS 5	rpm -Uvh http://public-repo-1.hortonworks.com/AMBARI-1.x/repos/centos5/AMBARI-1.x-1.el5.noarch.rpm	wget http://public-repo-1.hortonworks.com/AMBARI-1.x/repos/centos5/ambari.repo cp ambari.repo /etc/yum.repos.d
RHEL and CentOS 6	rpm -Uvh http://public-repo-1.hortonworks.com/AMBARI-1.x/repos/centos6/AMBARI-1.x-1.el6.noarch.rpm	wget http://public-repo-1.hortonworks.com/AMBARI-1.x/repos/centos6/ambari.repo cp ambari.repo /etc/yum.repos.d
SLES 11	rpm -Uvh http://public-repo-1.hortonworks.com/AMBARI-1.x/repos/suse11/AMBARI-1.x-1.noarch.rpm	wget http://public-repo-1.hortonworks.com/AMBARI-1.x/repos/suse11/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 install based on your base platform.

2.1.1. RHEL/CentOS 5.x

1. Install the epel repository:

```
yum install epel-release
```

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

```
yum repolist
```

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

```
repo id                 repo name
AMBARI-1.x              Ambari 1.x
HDP-UTILS-1.1.0.15      Hortonworks Data Platform Utils Version - HDP-UT
epel                    Extra Packages for Enterprise Linux 6 - x86_64
```

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

```
yum install ambari-server
```

2.1.2. RHEL/CentOS 6.x

1. Install the epel repository:

```
yum install epel-release
```

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

```
yum repolist
```

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

```
repo id                 repo name
AMBARI-1.x              Ambari 1.x
HDP-UTILS-1.1.0.15      Hortonworks Data Platform Utils Version - HDP-UT
epel                    Extra Packages for Enterprise Linux 6 - x86_64
```

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

```
yum install ambari-server
```

2.1.3. SLES 11

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-1.1.0.15   | Hortonworks Data Platform Utils Version
                             - HDP-UTILS-1.1.0.15
```

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.

1. Run the Ambari Server setup:

```
ambari-server setup
```

If you have *not* disabled SELinux, you may get a warning. Enter y to continue. If you have *not* temporarily disabled iptables, the setup will do it for you.

2. PostgreSQL is configured by the process. When you are prompted to enter Advanced Database Configuration, enter `n` to set up the default username and password: `ambari-server/bigdata`. To use your own username and password, enter `y`.

3. 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.

4. 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 –verbose	Prints verbose info and warning messages to the console during Setup.

2.3. 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 HDP.

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 wish.

3.2. Welcome

The first step creates the cluster name.

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

3.3. 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. Use the **Choose File** button in the **Host Connectivity Information** section to find the private key file that matches the public key you installed earlier on all your hosts.
3. If you do not wish to have Ambari automatically install the Ambari Agent on all your hosts using SSH, you have the option of doing this work manually. Uncheck **Provide your**

SSH Private Keyand **OK** out of the **Warning**. See Appendix: Installing Ambari Agents Manually for more information.

Note

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

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.

3.4. Confirm Hosts

This screen allows you to make sure that Ambari has located the correct hosts for your cluster.

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

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

3.5. Choose Services

Hortonworks Data Platform is made up of a number of components. 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**.

3.6. 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 RAM indicated.

1. If you wish to change locations, click the dropdown list next to the service in the left column and select the appropriate host.

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

3.7. Assign Slaves and Clients

The Ambari install wizard attempts to assign the slave components (DataNodes, TaskTrackers, 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.

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

3.8. Customize Services

The **Customize Services** screen presents you with a set of tabs that let you manage configuration settings for HDP 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 nine groups of configuration properties. The install wizard has set reasonable defaults for all properties except for one in the Hive/HCat and two in the Nagios tabs. Those three are the only ones you *must* set yourself:

HDFS

MapReduce

Hive/HCat

WebHCat

HBase

ZooKeeper

Oozie

Nagios

Misc

3.8.1. HDFS

This tab covers HDFS settings. Here you can set properties for the NameNode, Secondary NameMode, DataNodes, and some general and advanced properties. Click the name of the group to expand and collapse the display.

Table 3.1. HDFS Settings:NameNode

Name	Notes
NameNode host	This value is prepopulated based on your choices on previous screens
NameNode directories	NameNode directories for HDFS to store the file system image.
NameNode Java heap size	Initial and maximum Java heap size for NameNode (Java options -Xms and -Xmx)
NameNode new generation size	Default size of Java new generation for NameNode (Java option -XX:NewSize)

Table 3.2. HDFS Settings:SNameNode

Name	Notes
SNameNode host	This value is prepopulated based on your choices on previous screens
Secondary NameNode Checkpoint Directory	Directory on the local filesystem where the Secondary NameNode should store the temporary images to merge

Table 3.3. HDFS Settings:DataNodes

Name	Notes
DataNode hosts	The hostnames of the hosts on which this group's DataNodes run.
DataNode directories	The directories where HDFS should store the data blocks for this group.
DataNode maximum Java heap size	Maximum Java heap size for DataNode (Java option -Xmx)
DataNode volumes failure toleration	The number of volumes that are allowed to fail before a DataNode stops offering services.

Table 3.4. HDFS Settings:General

Name	Notes
WebHDFS enabled	Check to enable WebHDFS
Hadoop maximum Java heap size	Maximum Java heap size for daemons such as Balancer (Java option -Xmx) [a]
Reserved space for HDFS	Space in GB per volume reserved for HDFS
HDFS Maximum Checkpoint Delay	Maximum delay between two consecutive checkpoints for HDFS in seconds

Name	Notes
HDFS Maximum Edit Log Size for Checkpointing	Maximum size of the edits log file that forces an urgent checkpoint even if the maximum checkpoint delay is not reached

[a]The default value for this property is 1 GB. This value may need to be reduced for a VM-based installation. On the other hand, for significant work using Hive Server, 2GB is a more realistic value.

Table 3.5. HDFS Settings:Advanced

Name	Notes
Hadoop Log Dir Prefix	The parent directory for Hadoop log files. The HDFS log directory will be ${hadoop_log_dir_prefix}/${hdfs_user} and the MapReduce log directory will be ${hadoop_log_dir_prefix}/${mapred_user}
Hadoop PID Dir Prefix	The parent directory in which the PID files for Hadoop processes will be created. The HDFS PID directory will be ${hadoop_pid_dir_prefix}/${hdfs_user} and the MapReduce PID directory will be ${hadoop_pid_dir_prefix}/${mapred_user}
Exclude hosts	Names a file that contains a list of hosts that are not permitted to connect to the NameNode. This file will be placed inside the Hadoop conf directory.
Include hosts	Names a file that contains a list of hosts that are permitted to connect to the NameNode. This file will be placed inside the Hadoop conf directory.
Block replication	Default block replication
dfs.block.local-path-access.user	The user who is allowed to perform short-circuit reads
dfs.datanode.socket.write.timeout	DFS client write socket timeout
dfs.replication.max	Maximal block replication
dfs.heartbeat.interval	DataNode heartbeat interval in seconds
dfs.safemode.threshold.pct	The percentage of blocks that should satisfy the minimal replication requirement set by dfs.replication.min. Values less than or equal to 0 mean not to start in safe mode. Values greater than 1 make safe mode permanent.
dfs.balance.bandwidthPerSec	The maximum amount of bandwidth that each DataNode can utilize for balancing purposes in terms of the number of bytes per second
dfs.block.size	Default block size for new files
dfs.datanode.ipc.address	The DataNode IPC server address and port. If the port is 0 the server starts on a free port.
dfs.blockreport.initialDelay	Delay in seconds for first block report
dfs.datanode.du.pct	The percentage of real available space to use when calculating remaining space
dfs.namenode.handler.count	The number of server threads for the NameNode
dfs.datanode.max.xcievers	PRIVATE CONFIG VARIABLE
dfs.umaskmode	The octal umask to be used in creating files and directories
dfs.web.ugi	The user account used by the web interface. Syntax: USERNAME, GROUP1, GROUP2 ....
dfs.permissions	If `true`, enable permissions checking in HDFS. If `false`, permission checking is turned off, but all other behavior stays unchanged. Switching from one value to the other does not change the mode, owner, or group of files or directories.
dfs.permissions.supergroup	The name of the group of superusers
ipc.server.max.response.size	

Name	Notes
dfs.block.access.token.enable	If `true` access tokens are required to access DataNodes. If `false` access tokens are not checked.
dfs.secondary.https.port	The https port where the SecondaryNameNode binds
dfs.https.port	The https port where the NameNode binds
dfs.access.time.precision	The access time for HDFS file is precise to this value. The default value is 1 hour. A value of 0 disables access times for HDFS.
dfs.cluster.administrators	ACL for all who can view the default servlets in HDFS
ipc.server.read.threadpool.size	
io.file.buffer.size	The size of bugger for use in sequence files. The size of this bugger should probably be a multiple of hardware page size (4096 on Intel x86). This value determines how much data is buffered during read and write operations.
io.serializations	
io.compression.codec.lzo.class	The implementation class for the LZO codec.
fs.trash.interval	Number of minutes between trash checkpoints. If zero, the trash feature is disabled.
ipc.client.idlethreshold	The threshold number of connections after which connections are inspected for idleness
ipc.client.connection.maxidletime	Maximum time after which the client brings down the connection to the server
ipc.client.connect.max.retries	Maximum number of retries for IPC connections
webinterface.private.actions	If true, the native web interfaces for JT and NN may contain actions, such as kill job, delete file, etc. that should not be exposed to the public. Enable this option if these interfaces are reachable only by appropriately authorized users.
Custom Hadoop Configs	Use this text box to enter values for core-site.xml properties not exposed by the UI. Enter in "key=value" format, with a newline as a delimiter between pairs.
Custom HDFS Configs	Use this text box to enter values for hdfs-site.xml properties not exposed by the UI. Enter in "key=value" format, with a newline as a delimiter between pairs.

3.8.2. MapReduce

This tab covers MapReduce settings. Here you can set properties for the JobTracker and TaskTrackers, as well as some general and advanced properties. Click the name of the group to expand and collapse the display

Table 3.6. MapReduce Settings: JobTracker

Name	Notes
JobTracker host	This value is prepopulated based on your choices on previous screens
JobTracker new generation size	Default size of Java new generation size for JobTracker (Java option -XX:NewSize)
JobTracker maximum new generation size	Maximum size of Java new generation for JobTracker (Java option -XX:MaxNewSize)
JobTracker maximum Java heap size	Maximum Java heap size for JobTracker in MB (Java option -Xmx)

Table 3.7. MapReduce Settings: TaskTracker

Name	Notes
TaskTracker hosts	This value is prepopulated based on your choices on previous screens
MapReduce local directories	Directories for MapReduce to store intermediate data files
Number of Map slots per node	Number of slots that Map tasks that run simultaneously can occupy on a TaskTracker
Number of Reduce slots per node	Number of slots that Reduce tasks that run simultaneously can occupy on a TaskTracker.
Java options for MapReduce tasks	Java options for the TaskTracker child processes

Table 3.8. MapReduce Settings: General

Name	Notes
MapReduce Capacity Scheduler	The scheduler to use for scheduling MapReduce jobs
Cluster's Map slot size (virtual memory)	The virtual memory size of a single Map slot in the MapReduce framework. Use -1 for no limit
Cluster's Reduce slot size (virtual memory)	The virtual memory size of a single Reduce slot in the MapReduce framework. Use -1 for no limit
Upper limit on virtual memory for single Map task	Upper limit on virtual memory for single Map task. Use -1 for no limit.
Upper limit on virtual memory for single Reduce task	Upper limit on virtual memory for single Reduce task. Use -1 for no limit.
Default virtual memory for a job's map-task	Virtual memory for single Map task. Use -1 for no limit.
Default virtual memory for a job's reduce-task	Virtual memory for single Reduce task. Use -1 for no limit.
Map-side sort buffer memory	The total amount of Map-side buffer memory to use while sorting files (Expert-only configuration)
Limit on buffer	Percentage of sort buffer used for record collection (Expert-only configuration)
Job log retention (hours)	The maximum time, in hours, for which the user-logs are to be retained after the job completion.
Maximum number tasks for a Job	Maximum number of tasks for a single Job. Use -1 for no limit.
LZO compression	Check to enable LZO compression in addition to Snappy
Snappy compression	Check to enable Snappy compression
Enable Job Diagnostics	Check to enable tools for tracing the path and troubleshooting the performance of MapReduce jobs

Table 3.9. MapReduce Settings: Advanced

Name	Notes
MapReduce system directories	MapReduce system directories
io.sort.factor	
mapred.tasktracker.tasks.sleeptime-before-sigkill	Normally this is the amount of time before killing processes, and the recommended default is 5.000 seconds, a value of 5000 here. In this case it is used solely to blast tasks before killing them, and killing them very quickly (.25 second) to guarantee that we do not leave VMs around for later jobs
mapred.job.tracker.handler.count	The number of server threads for the JobTracker. Roughly 4% of the number of TaskTracker nodes.
mapreduce.cluster.administrators	ACL for MapReduce administrators by group

Name	Notes
mapred.reduce.parallel.copies	
tasktracker.http.threads	
mapred.map.tasks.speculative.execution	If `true`, then multiple instances of some map tasks may be executed in parallel
mapred.reduce.tasks.speculative.execution	If `true`, then multiple instances of some reduce tasks may be executed in parallel
mapred.reduce.slowstart.completed.maps	
mapred.inmem.merge.threshold	The threshold, in terms of the number of files, for triggering the in-memory merge process. When the threshold is hit, we initiate the merge and spill to disk. A value of less than or equal to 0 means no threshold is set and ramfs's memory consumption triggers the merge.
mapred.job.shuffle.merge.percent	The threshold, expressed as a percentage of the total memory allocated to storing in-memory map outputs (defined in `mapred.job.shuffle.input.buffer.percent`), for triggering the in-memory merge process.
mapred.job.shuffle.input.buffer.percent	The percentage of memory to be allocated from the maximum heap size for storing map outputs during the shuffle.
mapred.output.compression.type	If the job outputs are to be compressed as SequenceFiles, how should they be compressed? Acceptable values are: NONE, RECORD, or BLOCK.
mapred.jobtracker.completeuserjobs.maximum	
mapred.jobtracker.restart.recover	A value of `true` enables job recovery on restart; `false` starts afresh
mapred.job.reduce.input.buffer.percent	The percentage of memory relative to the maximum heap size. When the shuffle is concluded, any remaining map outputs in memory must consume less than this threshold before the reduce can begin.
mapreduce.reduce.input.limit	The limit on the input size of the reduce. If the estimated input size of the reduce is greater than this value, job is failed. A value of -1 means that no limit is set.
mapred.task.timeout	The number of milliseconds before a task will be terminated if it neither reads an input, writes an output, or updates its status string.
jetty.connector	
mapred.child.root.logger	
mapred.max.tracker.blacklists	If a node is reported blacklisted by this number of successful jobs within the timeout window, it will be graylisted.
mapred.healthChecker.interval	
mapred.healthChecker.script.timeout	
mapred.job.tracker.persist.jobstatus.active	Indicates if persistency of job status is active or not
mapred.job.tracker.persist.jobstatus.hours	The number of hours job status information is persisted in DFS. Job status information is available after it drops off the memory queue and between JobTracker restarts. A value of zero means that job status information is not persisted at all.
mapred.jobtracker.retirejob.check	
mapred.jobtracker.retirejob.interval	
mapred.job.tracker.history.completed.location	
mapreduce.fileoutputcommitter.marksuccessfuljobs	

Name	Notes
mapred.job.reuse.jvm.num.tasks	The number of tasks to run per JVM. A value if -1 indicates no limit.
hadoop.job.history.user.location	
mapreduce.jobtracker.staging.root.dir	The path prefix for the staging directories. The next level is always the user's name. It is a path in the default file system.
mapreduce.tasktracker.group	The group that the TaskTracker controller uses for accessing the controller. The mapred user *must* be a member and users should *not* be members.
mapreduce.jobtracker.split.metainfo.maxsize	If the size of the split metainfo file is larger than this value, the JobTacker will fail the job during initialization.
mapred.jobtracker.blacklist.fault-timeout-window	Sliding window in minutes
mapred.jobtracker.blacklist.fault-bucket-width.	Bucket size in minutes.
mapred.queue.names	Comma separated list of queues configured for this jobtracker
Custom MapReduce Configs	Use this text box to enter values for mapred-site.xml properties not exposed by the UI. Enter in "key=value" format, with a newline as a delimiter between pairs.

3.8.3. Hive/HCat

This tab covers Hive and HCatalog settings. Here you can set properties for the Hive Metastore and database and related options. Click the name of the group to expand and collapse the display.

Table 3.10. Hive/HCat Settings: Hive Metastore

Name	Notes
Hive Metastore host	The host that has been assigned to run the Hive Metastore
Hive Database	Check **New MySQL Database** to have Ambari create one for you or **Existing MySQL Database** to use an existing instance.
Database Type	**MySQL** is pre-populated
Database host	The FQDN of the host that has been assigned to run the database
Database name	The name for the database. Can be any legal name.
Database user	The username used to connect to the database
Database password	The password for accessing the Hive/HCatalog Metastore. This is a **required** property. You must type it in twice.

Table 3.11. Hive/HCat Settings: Advanced Settings

Name	Notes
Hive PID dir	Directory for Hive process PID files
HCat log dir	Directory for HCatalog log files
HCat PID dir	Directory for HCatalog process PID files
hive.metastore.local	Whether to connect to remove a metastore server or open a new metastore server in the Hive Client JVM
javax.jdo.option. ConnectionDriverName	Driver class name for a JDBC metastore
hive.metastore.warehouse.dir	The location of the default database for the warehouse
hive.metastore.cache.pinobjtypes	Comma separated list of metastore object types that should be pinned in the cache

Name	Notes
hive.semantic.analyzer.factory.impl	Which Semantic Analyzer Factory implementation class is used by CLI
hadoop.clientside.fs.operations	If FS operations are owned by the client
hive.metastore.client.socket.timeout	Metastore client socket timeout in seconds
hive.metastore.execute.setugi	In unsecure mode, setting this property to `true` causes the metastore to execute DFS operations using the client's reported user and group permissions. **Note** : this property must be set on both the client and server sides. This is a best effort property. If client is set to true and server is set to false, the client setting is ignored.
hive.security.authorization.enabled	Whether hive client authorization is enabled
hive.security.authorization.manager	The class name of the hive client authorization manager. A user defined authorization class should implement the `org.apache.hadoop.hive.ql.security.authorization.HiveAuthorizationProvider` interface
hive.server2.enable.doAs	
hive.hdfs.impl.disable.cache	
Custom Hive Configs	Use this text box to enter values for hive-site.xml properties not exposed by the UI. Enter in "key=value" format, with a newline as a delimiter between pairs.

3.8.4. WebHCat

This tab covers Hive/HCatalog settings for the MySQL instance. Here you can set some advanced properties for the WebHCat interface. Click the name of the group to expand and collapse the display.

Table 3.12. WebHCat Settings: Advanced Settings

Name	Notes
templeton.port	HTTP port for the main server
templeton.hadoop.conf.dir	Path to the Hadoop configuration
templeton.jar	Path to the Templeton .jar file
templeton.libjars	Jar files to add to the classpath
templeton.hadoop	Path to the Hadoop executable
templeton.pig.archive	Path to the Pig archive
templeton.pig.path	Path to the Pig executable
templeton.hcat	Path to the HCatalog executable
templeton.hive.archive	Path to the Hive archive
templeton.hive.path	Path to the Hive executable
templeton.storage.class	The class to use for storage
templeton.override.enabled	`True` to enable the override path in templeton.override.jars
templeton.streaming.jar	HDFS path to the Hadoop streaming jar file
templeton.exec.timeout	Time out for the WebHCat API
Custom WebHCat Configs	Use this text box to enter values for the webhcat-site.xml properties not exposed by the UI. Enter in "key=value" format, with a newline as a delimiter between pairs.

3.8.5. HBase

This tab covers HBase settings. Here you can set properties for the HBase Master and RegionServer, as well as some general and advanced properties. Click the name of the group to expand and collapse the display.

Table 3.13. HBase Settings: HBase Master

Name	Notes
HBase Master host	This value is prepopulated based on your choices on previous screens
HBase Master Maximum Java heap size	Maximum Java heap size for HBase master (Java option -Xmx)

Table 3.14. HBase Settings: RegionServer

Name	Notes
RegionServer hosts	This value is prepopulated based on your choices on previous screens
HBase Region Servers maximum Java heap size	Maximum Java heap size for HBase Region Servers (Java option -Xmx) **Important**: For more information on sizing, see Recommended Memory Configurations for the MapReduce Service for recommended sizing.
HBase Region Server Handler	Count of RPC Listener instances spun up on RegionServers
HBase Region Major Compaction	The time between major compactions of all HStoreFiles in a region. Set to 0 to disable automated major compactions.
HBase Region Block Multiplier	Block updates if memstore reaches "`Multiplier * HBase Region Memstore Flush Size`" bytes. Useful preventing runaway memstore size during spikes in update traffic
HBase Region Memstore Flush Size	Memstore will be flushed to disk if size of the memstore exceeds this number of bytes.

Table 3.15. HBase Settings: General

Name	Notes
HBase HStore compaction threshold	When HStoreFiles in any one HStore are greater than this number, a compaction is run to rewrite all HStoreFiles files as one.
HFile block cache size	Percentage of maximum heap (-Xmx setting) to allocate to block cache used by HFile/StoreFile. You can set this to 0 to disable but this is not a recommended practice.
Maximum HStoreFile Size	If any one of a column families' HStoreFiles has grown to exceed this value, the hosting HRegion is split in two.
HBase Client Scanner Caching	Number of rows that will be fetched when calling `next` on a scanner if it is not served from (local, client) memory. Do not set this value such that the time between invocations is greater than the scanner timeout
Zookeeper timeout for HBase Session	HBase passes this to the zk quorum as suggested maximum time for a session.
HBase Client Maximum key-value Size	Specifies the combined maximum allowed size of a KeyValue instance. It should be set to a fraction of the maximum region size.

Table 3.16. HBase Settings: Advanced

Name	Notes
HBase Log Dir	Directory for HBase logs
HBase PID Dir	Directory for the PID files for HBase processes
HDFS Short-circuit read	Check to enable
HDFS shortcircuit skip checksum	Skip checksum for short-circuit read. Check to enable
HDFS append support	Check to enable
hstore blocking storefiles	If more than this number of StoreFiles in any one Store (one StoreFile is written per flush of MemStore), then updates are blocked in this HRegion until compaction is completed or until `hbase.hstore.blockingWaitTime` is exceeded.
hbase.master.lease.thread.wakefrequency	The interval between checks for the expired region server leases. The default is 15 seconds but may be reduced so that the master notices a dead RegionServer more quickly.
hbase.superuser	Comma-separated list of users who are allowed full privileges across the cluster, regardless of stored ACLs. Used only when HBase security is enabled.
hbase.regionserver.optionalcacheflushinterval	Amount of time to wait since the last time a region was flushed before invoking an optional cache flush. Default is 60,000.
Custom HBase Configs	Use this text box to enter values for hbase-site.xml properties not exposed by the UI. Enter in "key=value" format, with a newline as a delimiter between pairs.

3.8.6. ZooKeeper

This tab covers ZooKeeper settings. Here you can set properties for ZooKeeper servers as well as some advanced properties. Click the name of the group to expand and collapse the display

Table 3.17. ZooKeeper Settings: ZooKeeper Server

Name	Notes
ZooKeeper Server hosts	This value is prepopulated based on your choices on previous screens
ZooKeeper directory	Data directory for ZooKeeper.
Length of single Tick	The length of a single tick in milliseconds, which is the basic time unit used by ZooKeeper
Ticks to allow for sync at Init	Amount of time in ticks to allow followers to connect and sync to a leader
Ticks to allow for sync at Runtime	Amount of time in ticks to allow followers to connect
Port for Running ZK Server	Port for running ZK server

Table 3.18. ZooKeeper Settings: Advanced

Name	Notes
ZooKeeper Log Dir	Directory for ZooKeeper log files
ZooKeeper PID Dir	Directory for the PID files for ZooKeeper processes

3.8.7. Oozie

This screen covers Oozie settings. Here you can set properties for the Oozie server, as well as some advanced properties. Click the name of the group to expand and collapse the display.

Table 3.19. Oozie Settings: Oozie Server

Name	Notes
Oozie Server host	This value is prepopulated based on your choices on previous screens
Oozie Data Dir	Data directory in which the Oozie DB exists.

Table 3.20. Oozie Settings: Advanced

Name	Notes
Oozie Log Dir	Directory for Oozie logs
Oozie PID Dir	Directory for the PID files for Oozie processes
oozie.system.id	The Oozie system ID
oozie.systemmode	System mode for Oozie at startup
oozie.service.AuthorizationService.security.enabled	If security (username/admin role) is enabled or not. If disabled, any user can manage Oozie system and any job.
oozie.service.PurgeService.older.than	Jobs older than this value in days will be purged by the PurgeService.
oozie.service.PurgeService.purge.interval	Interval at which the PurgeService will run, given in seconds
oozie.service.CallableQueueService.queue.size	Max callable queue size
oozie.service.CallableQueueService.threads	Number of threads for executing callables.
oozie.service.CallableQueueService.callable.concurrency	Maximum concurency for a given callable type. Each command is a callable type: submit, start, run, etc. Each action type is a callable type: MapReduce, SSH, sub-workflow, etc. All commands that use action executors (action-start, action-end. etc.) use the action type as the callable type.
oozie.service.coord.normal.default.timeout	Default timeout for a coordinator action input check (in minutes) for a normal job. Set to -1 for infinite timeout.
oozie.db.schema.name	Oozie database name
oozie.service.HadoopAccessorService.jobTracker.whitelist	Whitelisted job tracker for Oozie service
oozie.authentication.type	Defines authentication for Oozie HTTP endpoint. One of `simple` | `kerberbos` | #AUTHENTICATION_HANDLER_CLASSNAME#
oozie.service.HadoopAccessorService.nameNode.whitelist	
oozie.service.WorkflowAppService.system.libpath	System library path to use for workflow applications. This path is added to workflow applications if the `oozie.use.system.libpath` property in their job properties is set to `true`.
use.system.libpath.for.mapreduce.and.pig.jobs	If `true`, submissions of MapReduce and Pig jobs automatically include the system library path. Doing so means that users do not need to specify where the Pig .jar files are because the ones that are in the system library path are used.
oozie.authentication.kerberos.name.rules	The name rules to resolve Kerberos principal names. See Hadoop's Kerberos Name for more details.

Name	Notes
oozie.service.HadoopAccessorService. hadoop.configurations	Comma-separated list of form AUTHORITY=HADOOP_CONF_DIF, where AUTHORITY is the host/port of the Hadoop service (JobTracker, HDFS). The wildcard * configuration is used when there is no exact match for an authority. The HADOOP_CONF_DIR contains the relevant Hadoop*-site.xml files. A relative path is assumed to begin in the Oozie configuration directory. The path can also be absolute and point to Hadoop client conf/directories in the local filesystem.
oozie.service.ActionService.executor.ext.classes	
oozie.service.SchemaService.wf.ext.schemas	
oozie.service.JPAService.create.db.schema	Creates the Oozie DB. If set to `true` it creates the DB schema if it does not exist. If the DB schema exists, it is a NOP. If set to `false`, it does not create the DB schema. **Note**: if the DB schema does not exist, start up fails.
oozie.service.JPAService.jdbc.driver	The JDBC driver class
oozie.service.JPAService.jdbc.url	The JDBC URL
oozie.service.JPAService.jdbc.username	The DB username
oozie.service.JPAService.jdbc.password	The DB user password. **IMPORTANT**: If the password is empty, leave a 1 space string. The service trims the value, and if it is empty, the configuration assumes it is NULL.
oozie.service.JPAService.pool.max.active.conn	Maximum number of connections
Custom Oozie Configs	Use this text box to enter values for oozie-site.xml properties not exposed by the UI. Enter in "key=value" format, with a newline as a delimiter between pairs.

3.8.8. Nagios

This screen covers Nagios settings. Here you can set general properties for Nagios. You *must* set the password and email properties.

Table 3.21. Nagios Settings:General

Name	Notes
Nagios Admin User	Nagios Web UI Admin username [default:nagiosadmin]
Nagios Admin Password	Nagios Web UI Admin password. This is a **required** property. You must type it in twice.
Hadoop Admin email	The email address to which Nagios should send alerts. This is a **required** property.

3.8.9. Misc

This screen covers miscellaneous settings. Here you can set various general properties. Click the name of the group to expand and collapse the display.

Table 3.22. Misc Settings:Users/Groups

Name	Notes
Proxy group for Hive, WebHCat, and Oozie	The name of the proxy group: for example `users`
HDFS User	The user to run HDFS: for example `hdfs`
MapReduce User	The user to run MapReduce: for example `mapred`
HBase User	The user to run HBase : for example `hbase`
Hive User	The user to run Hive: for example `hive`

Name	Notes
HCat User	The user to run HCatalog: for example, `hcat`
WebHCat User	The user to run WebHCat: for example, `hcat`
Oozie User	The user to run Oozie : for example `oozie`
Pig User	The user to run Pig: for example `pig`
Sqoop User	The user to run Sqoop: for example `sqoop`
ZooKeeper User	The user to run ZooKeeper: for example `zookeeper`
Group	The group for the users specified above: for example `hadoop`

When you have made all your changes, click **Next**.

3.8.10. Recommended Memory Configurations for the MapReduce Service

- Make sure that there is enough memory for all the processes. Remember that system processes take around 10% of the available memory.
- For co-deploying an HBase RegionServer and MapReduce service on the same node, reduce the RegionServer's heap size (use the HBase Region Servers maximum Java heap size property to modify the RegionServer heap size).
- For co-deploying an HBase RegionServer and the MapReduce service on the same node, or for memory intensive MapReduce applications, modify the map and reduce slots as suggested in the following example:

EXAMPLE: For co-deploying an HBase RegionServer and the MapReduce service on a machine with 16GB of available memory, the following would be a recommended configuration:

2 GB: system processes

8 GB: MapReduce slots. 6 Map + 2 Reduce slots per 1 GB task

4 GB: HBase RegionServer

1 GB: TaskTracker

1 GB: DataNode

To change the number of Map and Reduce slots based on the memory requirements of your application, use the following properties:

- Number of Map slots per node [17]
- Number of Reduce slots per node [17]

3.9. Review

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

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

3.10. Install, Start and Test

The progress of your 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.

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

When **Successfully installed the cluster** appears, click **Next**.

3.11. Summary

The Summary page gives you a summary of the accomplished tasks. Click **Complete**
The Monitoring Dashboard for your cluster appears. For information on using the Administrative and Monitoring tools, please see Management and Monitoring with Apache Ambari.

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 **Manage Services** tab to check the status of each component. If you do not have access to Manage Services, you must start and stop the services manually. For information on how to do this, see Controlling HDP 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. 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 v6.x/CentOS v6.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 HDP 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. 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
```

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 v. 5.x and 6.x

1. Install the EPEL repo.

```
yum install epel-release
```

2. Install the Ambari Agent

```
yum install ambari-agent
```

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

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

[server]
hostname={your.ambari.server.hostname}
url_port=4080
secured_url_port=8443
```

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

```
ambari-agent start
```

5.2. SLES

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/ambari-agent.ini

[server]
hostname={your.ambari.server.hostname}
url_port=4080
secured_url_port=8443
```

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

```
ambari-agent start
```