

Hortonworks Data Platform

Apache Ambari Administration

(October 30, 2017)

Hortonworks Data Platform: Apache Ambari Administration

Copyright © 2012-2017 Hortonworks, Inc. Some rights reserved.

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

Unlike other providers of platforms built using Apache Hadoop, Hortonworks contributes 100% of our code back to the Apache Software Foundation. The Hortonworks Data Platform is Apache-licensed and completely open source. We sell only expert technical support, [training](#) and partner-enablement services. All of our technology is, and will remain free and open source.

Please visit the [Hortonworks Data Platform](#) page for more information on Hortonworks technology. For more information on Hortonworks services, please visit either the [Support](#) or [Training](#) page. Feel free to [Contact Us](#) directly to discuss your specific needs.



Except where otherwise noted, this document is licensed under
Creative Commons Attribution ShareAlike 4.0 License.
<http://creativecommons.org/licenses/by-sa/4.0/legalcode>

Table of Contents

1. Ambari Administration Overview	1
1.1. Terminology	1
1.2. Viewing the Ambari Admin Page	2
2. Administering Ambari	4
2.1. Log In to Ambari	4
2.2. Change the Default Admin Password	4
2.3. Create a Cluster	5
2.4. Manage Cluster Roles	5
2.4.1. Understanding Cluster Roles	5
2.4.2. Modify User and Group Cluster Roles	8
2.5. Open the Cluster Dashboard	9
2.6. Rename a Cluster	9
2.7. Managing Versions	9
2.7.1. Register New Version	10
2.7.2. Update Version Repository Base URLs	10
2.7.3. De-Register a Version	11
2.8. Register Remote Clusters	11
3. Managing Users and Groups	12
3.1. Understanding Users and Groups	12
3.1.1. Local and LDAP User and Group Types	12
3.1.2. Ambari Administrator Privileges	13
3.2. Create a Local User	13
3.3. Set User Status	13
3.4. Set the Ambari Admin Flag	14
3.5. Change the Password for a Local User	14
3.6. Delete a Local User	14
3.7. Create a Local Group	15
3.8. Managing Group Membership	15
3.8.1. Add a User to a Group	15
3.8.2. Modify Group Membership	16
3.9. Delete a Local Group	16
3.10. Enable User Home Directory Creation	16
4. Installing Ambari Agents Manually	18
4.1. Download the Ambari Repo	18
4.2. Install the Ambari Agents Manually	23
5. Customizing HDP Services	25
5.1. Defining Service Users and Groups for a HDP 2.x Stack	25
5.2. Setting Properties That Depend on Service Usernames/Groups	26
6. Using Custom and Private Hostnames	28
6.1. Configure a Custom Host Name	28
6.2. Configure a Public Host Name	29
6.2.1. Public Hostname Limitations	30
6.2.2. Checking if Public Hostnames Are Correctly Configured	30
7. Changing Host Names	31
8. Moving the Ambari Server	33
8.1. Back up Current Data	33
8.2. Update all Agents	33
8.3. Install the New Ambari Server	34

9. Moving the ZooKeeper Server	36
10. Configuring LZO Compression	38
10.1. Configure core-site.xml for LZO	38
10.2. Running Compression with Hive Queries	38
10.2.1. Create LZO Files	39
10.2.2. Write Custom Java to Create LZO Files	39
11. Using Existing Databases	40
11.1. Using Existing Databases - Ambari	40
11.1.1. Using Ambari with Oracle	41
11.1.2. Using Ambari with MySQL/MariaDB	42
11.1.3. Using Ambari with PostgreSQL	43
11.1.4. Troubleshooting Existing Databases with Ambari	44
11.2. Using New and Existing Databases - Hive	45
11.2.1. Using Hive with Oracle	45
11.2.2. Using Hive with MySQL/MariaDB	46
11.2.3. Using Hive with PostgreSQL	47
11.2.4. Troubleshooting Existing Databases with Hive	48
11.3. Using Existing Databases - Oozie	49
11.3.1. Using Oozie with Oracle	50
11.3.2. Using Oozie with MySQL/MariaDB	50
11.3.3. Using Oozie with PostgreSQL	51
11.3.4. Troubleshooting Existing Databases with Oozie	52
12. Setting up a local repository	54
12.1. Case study for setting up a local repository	54
13. Setting up Ambari to use an Internet Proxy Server	57
14. Configuring Network Port Numbers	59
14.1. Default Network Port Numbers - Ambari	59
14.2. Optional: Changing the Default Ambari Server Port	60
14.3. Optional: Changing the Ambari Server-Agent Port	60
15. Change the JDK Version	62
16. Using Ambari Blueprints	64
17. Tuning Ambari Performance	65
17.1. Purging Ambari Server Database History	66
18. Customizing Ambari Log + PID Directories	68
18.1. Customizing Ambari Server Log + PID Directories	68
18.2. Customizing Ambari Agent Log + PID Directories	69
19. Configuring Include File Management for HDFS and YARN	70
19.1. Enable Include File Management for HDFS	70
19.2. Enable Include File Management for Yarn	71
19.3. Disable Include File Management for HDFS	71
19.4. Disable Include File Management for Yarn	71

List of Tables

11.1. Hive Security Authorization Settings 49

1. Ambari Administration Overview

Apache Ambari enables you to provision, manage, and monitor Hadoop clusters. You should use this guide if you are responsible for installing and maintaining Ambari and the Hadoop clusters managed by Ambari.

Installing Ambari creates the default user `admin/admin`. This *Ambari-level Administrator* user has full control over all aspects of Ambari, including all clusters managed by the Ambari instance, as well as the ability to manage users, groups, and clusters.

When you log in to Ambari as Ambari Admin, you can perform the following tasks:

- [Administering Ambari \[4\]](#)
- [Managing Versions \[9\]](#)
- [Managing Users and Groups \[12\]](#)

More Information

[Install, Configure, and Deploy an HDP Cluster.](#)

1.1. Terminology

Familiarity with the following basic terms can help you to understand the key concepts associated with Ambari administration:

Ambari Admin	Specific privileges granted to a user that enables that user to administer Ambari. Users with the Ambari Admin privilege can grant this privilege to other users, or revoke it from them.
account	User name, password, and privileges.
cluster	An installation of a Hadoop cluster, based on a particular stack, that is managed by Ambari.
group	Unique group of users in Ambari.
group type	Local and LDAP. Local groups are maintained in the Ambari database. LDAP groups are imported to (and synchronized with) an external LDAP, if one is configured.
permissions	The permissions granted to a principal user or group for a particular view.
principal	User or group that can be authenticated by Ambari.
privilege	The mapping of a principal to a permission or role and a resource. For example, the user joe.operator is granted the role of Cluster Operator on the cluster DevCluster .
resource	The resource available and managed in Ambari. Ambari supports two types of resources: cluster and view. An Ambari Admin assigns permissions for a resource for users and groups.

role	The role that is assigned to a principal (user or group) on a particular cluster.
user	Unique user in Ambari.
user type	Local and LDAP. Local users are maintained in the Ambari database and authentication is performed against the Ambari database. LDAP users are imported to (and synchronized with) an external LDAP, if one is configured.
version	Stack version, which includes a set of repositories to install that version on a cluster.
view	A user interface component that is available to Ambari.

More Information

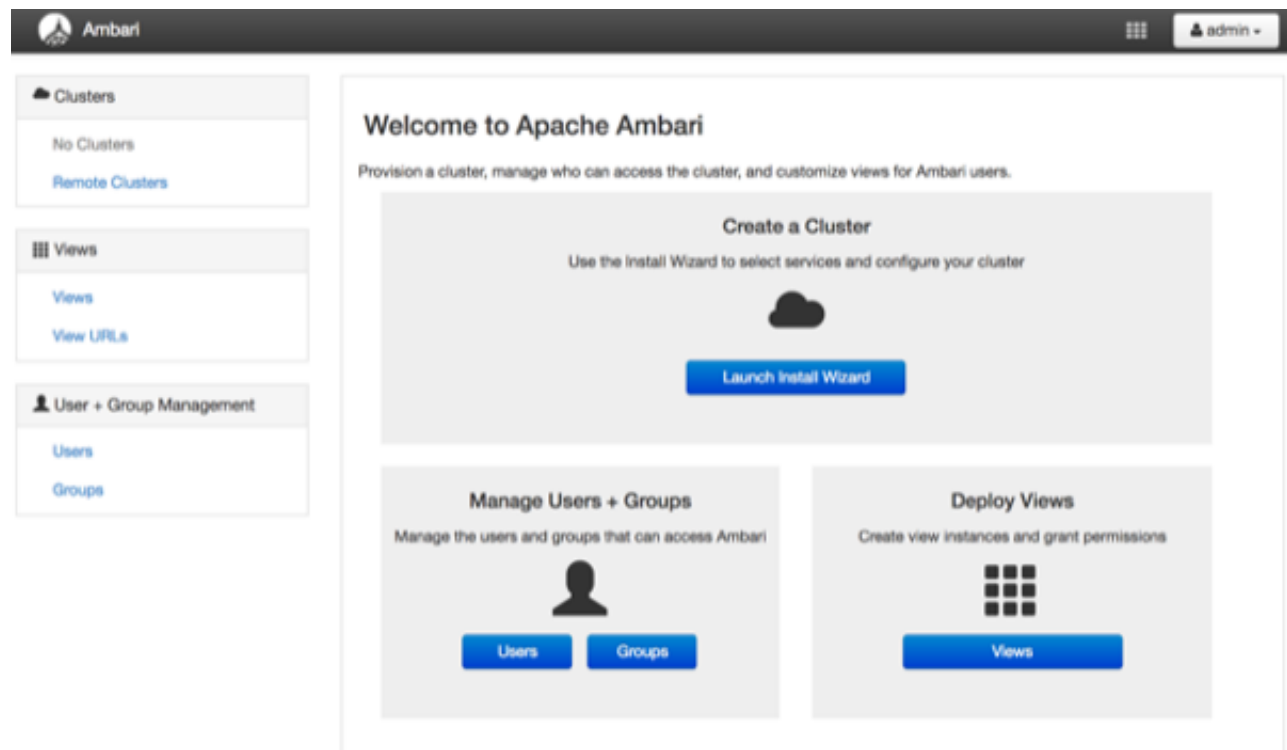
[Administering Ambari Views](#)

[Managing Cluster Roles](#)

[Managing Versions](#)

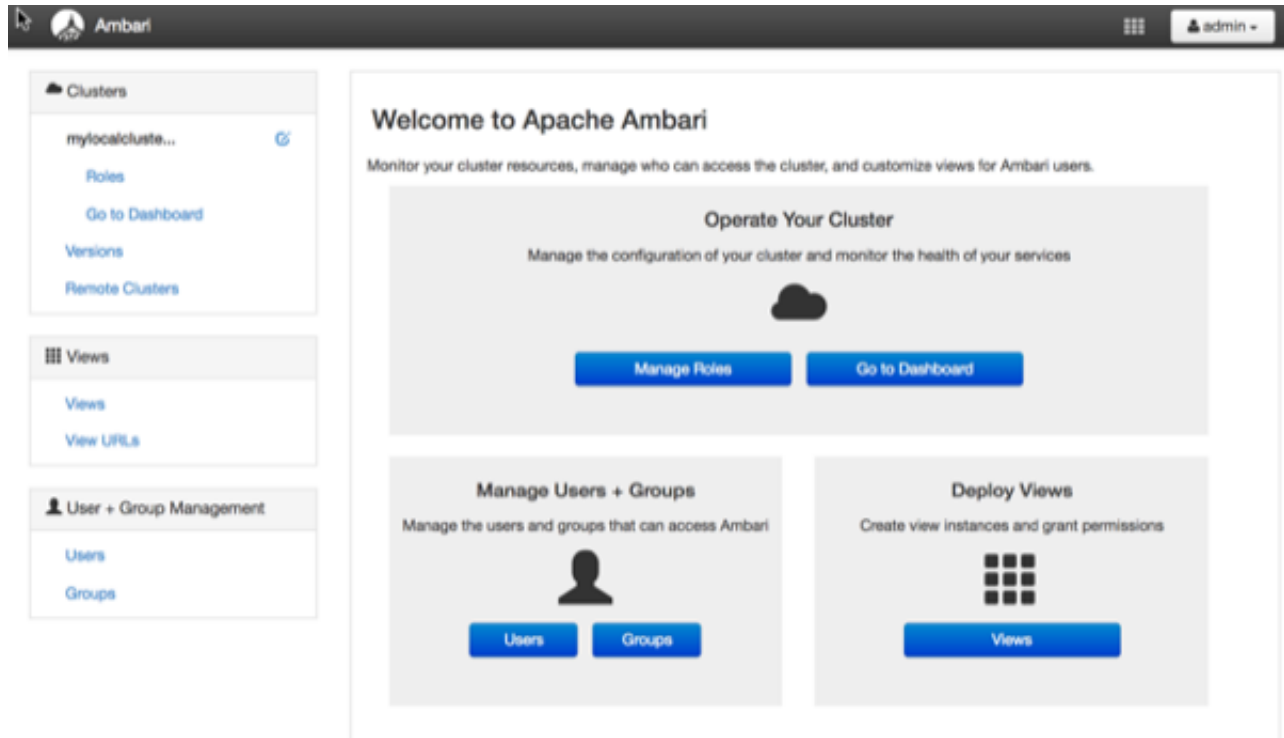
1.2. Viewing the Ambari Admin Page

When you log in to Ambari, the Ambari Admin page displays links to the operations that you can perform. These operations are also available from the left navigation pane. If you have not created a cluster, you can launch the Apache Ambari Install Wizard to create one:



- **Clusters** displays the current cluster name (if created), a link to rename your cluster, and a link to manage remote clusters.
- **Views** enables you to create and edit instances of deployed views and to manage access permissions for those instances.
- **User + Group Management** enables you to create and edit users and groups.

After you create a cluster, the Ambari Admin page displays links to additional operations you can perform to manage that cluster.



- **Versions** enables you to manage the stack versions that are available for the clusters.
- **Roles** enables you to add users and groups to roles having different permissions on the cluster.
- **Go To Dashboard** enables you to manage and monitor your Hadoop cluster.

More Information

[Create a Cluster \[5\]](#)

[Register Remote Clusters \[11\]](#)

[Administering Views](#)

[Managing Users and Groups \[12\]](#)

[Managing Versions](#)

2. Administering Ambari

An Ambari-level Administrator user has full control over all aspects of Ambari, including the abilities to create and manage a cluster. This chapter describes tasks you perform as a cluster administrator.

- [Log In to Ambari \[4\]](#)
- [Change the Default Admin Password \[4\]](#)
- [Create a Cluster \[5\]](#)
- [Manage Cluster Roles \[5\]](#)
- [Open the Cluster Dashboard \[9\]](#)
- [Rename a Cluster \[9\]](#)
- [Managing Versions \[9\]](#)
- [Register Remote Clusters \[11\]](#)

2.1. Log In to Ambari

After installing Ambari, you can log in to Ambari as follows:

Steps

1. Access Ambari

```
http://<your.ambari.server>:8080
```

<your.ambari.server> is the name of your Ambari server and 8080 is the default HTTP port.

2. Enter your credentials:

```
username/password = admin/admin
```

The Ambari Admin page displays.

More Information

[Viewing the Ambari Admin Page \[2\]](#)

2.2. Change the Default Admin Password

Using the Ambari Admin page, you can change the password for the default `admin` user to create a unique administrator credential for your system:

Steps

1. In **User + Group Management**, click **Users**.
2. Select the `admin` user.

3. Click **Change Password**.
4. Enter the current `admin` password and then your new password, twice.
5. Click **OK**.

2.3. Create a Cluster

After you have successfully installed Ambari, you can then create a cluster by using the Cluster Install wizard:

Steps

1. In the Ambari Admin page, click **Install Cluster**.
2. Complete the wizard pages.

More Information

[Install, Configure, and Deploy an HDP Cluster](#)

2.4. Manage Cluster Roles

Ambari-level administrators can assign users and groups different roles with permissions to perform operations at Ambari-, Cluster-, Service-, Host-, and User- (view-only) levels. This effectively distributes the responsibilities of managing a cluster while not relinquishing total control of the Ambari management facility.

More Information

[Understanding Cluster Roles \[5\]](#)

[Role Comparison Chart](#)

[Modify User and Group Cluster Roles \[8\]](#)

2.4.1. Understanding Cluster Roles

In Ambari 2.2 and earlier, the only roles available were Operator and Read-only. To enhance the granularity of permissions that can be granted to Ambari users, the following new, cluster-level roles are available:

Cluster User

Users assigned to the Cluster User role can view information about the cluster and its services, including configurations, service status, and health alerts. In Ambari 2.2 and earlier, this user was referred to as the Read-only user. Effectively, the cluster user is a view-only user.

Service Operator

Users assigned to the Service Operator role have control over service life cycles, such as starting and stopping services, performing service checks, and performing service-specific tasks such as rebalancing HDFS and refreshing the YARN Capacity Scheduler.

Service Administrator

Users assigned to the Service Administrator role have the same permissions as users assigned to the Service Operator role but have the added ability to configure services. This includes the ability to manage configuration groups, move service masters, and enable HA.

Cluster Operator

Users assigned to the Cluster Operator role have the same permissions as users assigned to the Service Administrator role but have the added ability to perform host-level tasks such as adding and removing hosts and components.

Cluster Administrator

Users assigned to the Cluster Administrators role have control over the relevant cluster, its hosts, and services. In Ambari 2.2 and earlier, this user was referred to as the Operator user.

Ambari Administrator

Ambari Administrator users have full control over all aspects of Ambari. This includes the ability to create clusters, change cluster names, register new versions of cluster software, and fully control all clusters managed by the Ambari instance.

The following chart compares these cluster roles:

	Cluster User	Service Operator	Service Administrator	Cluster Operator	Cluster Administrator	Ambari Administrator
Service-Level Authorizations						
View metrics						
View status information						
View configurations						
Compare configurations						
View service alerts						
Start, stop, or restart service						
Decommission or recommission						
Run service checks						
Turn maintenance mode on or off						
Perform service-specific tasks						
Modify configurations						

	Cluster User	Service Operator	Service Administrator	Cluster Operator	Cluster Administrator	Ambari Administrator
Manage configuration groups						
Move to another host						
Enable HA						
Enable or disable service alerts						
Add service to cluster						
Host-Level Authorizations						
View metrics						
View status information						
View configuration						
Turn maintenance mode on or off						
Install components						
Add or delete hosts						
Cluster-Level Authorizations						
View metrics						
View status information						
View configuration						
View stack version details						
View alerts						
Enable or disable alerts						
Enable or disable Kerberos						
Upgrade or downgrade stack						
Ambari-level Authorizations						

	Cluster User	Service Operator	Service Administrator	Cluster Operator	Cluster Administrator	Ambari Administrator
Create new clusters						
Set service users and groups						
Rename clusters						
Manage users						
Manage groups						
Manage Ambari Views						
Assign permission and roles						
Manage stack versions						
Edit stack repository URLs						

2.4.2. Modify User and Group Cluster Roles


If you want to modify cluster roles:

Steps

1. In the Clusters section, under the cluster name, click **Roles** to display the current user and group role assignments for that cluster.


2.



Using the block view (), edit the user and group assignment to each role for the cluster by clicking in the space, typing the first character in an existing user or group name, and selecting the name.

3.



Using the list view (), edit user and group assignments by finding the user or group and adjusting the role by using the drop-down menu.

4.



Click the check mark () to save your changes.

2.5. Open the Cluster Dashboard

After you create a cluster, you use the **Go to Dashboard** link to open the cluster operations dashboard. From this dashboard, you can manage and monitor cluster services, including managing the service life cycle, changing configurations, reviewing alerts, and so on.

More Information

[Viewing the Ambari Dashboards](#)

2.6. Rename a Cluster

After you create a cluster, you can rename it by using the Rename Cluster feature:

Steps

1. In Clusters, click the **Rename Cluster** icon, next to the cluster name that you want to change:
2. Enter alphanumeric characters to rename your cluster.
3. Click the check mark.
4. Confirm.
5. Restart Ambari server and the Ambari agents.

After renaming the cluster, alert checks must be re-queued on the agents. Therefore, you must restart Ambari Server and the Ambari Agents for the change to take effect.

6. Adjust any API calls you make to use the new name.

Changing the name of the cluster changes the name of the Ambari REST API resource for the cluster

2.7. Managing Versions

This section describes how to manage versions of the cluster stack software that are registered in Ambari, including registering a new version, upgrading version-repository base URLs, and de-registering a version. After you install a cluster, Ambari automatically registers the version of the stack software. The Versions table lists the stack, name, version, and cluster that is running that version.

- [Register New Version \[10\]](#)
- [Update Version Repository Base URLs \[10\]](#)
- [De-Register a Version \[11\]](#)

More Information

[HDP Repositories](#)

[Using a Local Repository](#)

[Using a Local RedHat Satellite Spacewalk Repository](#)

2.7.1. Register New Version

Steps

1. Browse to **Versions**.
2. Proceed to register a new version by clicking + **Register Version**.
3. Select the software version and method of delivery for your cluster.
 - **Choose HDP Stack.** The available HDP versions are shown in TABs. When you select a TAB, Ambari attempts to discover what specific version of that HDP Stack is available. That list is shown in a DROPDOWN. For that specific version, the available Services are displayed, with their Versions shown in the TABLE.
 - **Choose HDP Version.** If Ambari has access to the Internet, the specific Versions will be listed as options in the DROPDOWN. If you have a Version Definition File for a version that is not listed, you can click **Add Version...** and upload the VDF file. In addition, a **Default Version Definition** is also included in the list if you do not have Internet access or are not sure which specific version to install. If you choose the **Default Version Definition**, you must enter a "two-digit Version Number" in the **Name** input field.
 - **Choose Repository Delivery Method.** Using a Private Repository requires Internet connectivity. Using a Local Repository requires you have configured the software in a repository available in your network. To use the private software repositories, see the list of available HDP Repositories for each OS. Or, if you are using a local repository, enter the Base URLs for the local repository you have created.
4. Review **Advanced Options**.
 - **Skip Repository Base URL validation (Advanced):** Ambari will attempt to connect to the repository Base URLs and validate that you have entered a validate repository. If not, an error will be shown that you must correct before proceeding. This option will skip the Base URL validation.
 - **Use RedHat Satellite/Spacewalk:** This option will only be enabled when you plan to use a Local Repository. When you choose this option for the software repositories, **you are responsible for configuring the repository channel in Satellite/Spacewalk.** Please refer to the Red Hat Satellite/Spacewalk documentation for more information. Once configured, it is very important that ensure the repositories you confirm for the selected **stack version** are available on the hosts in the cluster. Ambari will not distribute or use .repo files and will rely on Satellite/Spacewalk as having the repositories configured with the correct stack version.
5. Click **Save**.

2.7.2. Update Version Repository Base URLs

Steps

1. Browse to **Versions**.
2. Click the version you want to modify.
3. Modify the base URLs for the repositories. To use the private software repositories, see the list of available HDP repositories for each OS. Or, if you are using a local repository, enter the Base URLs for the local repository you have created.
4. Click **Save**.
5. Click **Confirm Change**.

You **must** confirm the change since you are about to change repository Base URLs that are already in use. Please confirm that you intend to make this change and that the new Base URLs point to the same exact stack version and build.

2.7.3. De-Register a Version

Steps

1. Browse to **Versions**.
2. Click the version you want to de-register.
Only versions that are not installed in a cluster can be de-registered.
3. Click **Deregister Version** and then confirm.

2.8. Register Remote Clusters

You might work with clusters that are managed by Ambari but are not local to your Ambari server. These clusters are considered remote with respect to your local Ambari server. They are managed by a remote Ambari server. If you

plan to run a standalone server to host views, including accessing clusters managed by a different Ambari server, you can register those clusters from the standalone Ambari server as remote clusters. After you register these remote clusters, you can use them to configure view instances.

Steps:

1. Browse to **Remote Clusters** and click **Register Remote Cluster**.
2. Enter a name for the remote cluster, the Ambari cluster URL, and a cluster user name and associated password.
3. Click **Save**.

The remote cluster is now available for configuring View instances.

More Information

[Configuring View Instances](#)

3. Managing Users and Groups

As an Ambari administrator, you can create and manage users and groups available to Ambari. You can also import user and group information into Ambari from external LDAP systems. This section describes the specific tasks you perform when managing users and groups in Ambari:

- [Local and LDAP User and Group Types \[12\]](#)
- [Ambari Administrator Privileges \[13\]](#)
- [Create a Local User \[13\]](#)
- [Set User Status \[13\]](#)
- [Set the Ambari Admin Flag \[14\]](#)
- [Change the Password for a Local User \[14\]](#)
- [Delete a Local User \[14\]](#)
- [Create a Local Group \[15\]](#)
- [Managing Group Membership \[15\]](#)
- [Delete a Local Group \[16\]](#)
- [Enable User Home Directory Creation \[16\]](#)

3.1. Understanding Users and Groups

Ambari supports two types of users and groups: local and LDAP. The following topics describe how you use the Ambari Admin page to manage these users and groups.

- [Local and LDAP User and Group Types \[12\]](#)
- [Ambari Administrator Privileges \[13\]](#)

3.1.1. Local and LDAP User and Group Types

Local users are stored in and authenticate against the Ambari database. LDAP users have basic account information stored in the Ambari database. Unlike local users, LDAP users authenticate against an external LDAP system.

Local groups are stored in the Ambari database. LDAP groups have basic information stored in the Ambari database, including group membership information. Unlike local groups, LDAP groups are imported and synchronized from an external LDAP system.

To use LDAP users and groups with Ambari, you must configure Ambari to authenticate against an external LDAP system. Ambari grants no permissions by default to a new user or group, created either locally or by synchronizing against LDAP. You, as an Ambari administrator, must explicitly grant each user permissions to access clusters or views.

More Information

[Configure Ambari to use LDAP Server](#)

3.1.2. Ambari Administrator Privileges

You, as an Ambari administrator, can create new users, delete users, change user passwords, and edit user settings. You can control certain privileges for local and LDAP users. The following table lists the privileges available and those not available to the Ambari administrator for local and LDAP Ambari users.

Ambari Administrator Privileges for Ambari Local and LDAP Users

Ambari Administrator Privilege	Local User	LDAP User
Change password	Available	Not Available
Set Ambari Admin flag	Available	Available
Change group membership	Available	Not Available
Delete user	Available	Not Available
Set active or inactive status	Available	Available

3.2. Create a Local User

As an Ambari administrator, on the Ambari Admin page:

Steps

1. Browse to **Users**.
2. Click **Create Local User**.
3. Enter a unique user name.

All user names are converted to lowercase.

4. Enter a password, and then confirm that password.
5. Click **Save**.

3.3. Set User Status

User status indicates whether the user is active and should be allowed to log in to Ambari or should be inactive and denied the ability to log in. By setting the status flag as active or inactive, you can effectively disable user account access to Ambari while preserving the user account information related to permissions.

As an Ambari administrator, on the Ambari Admin page:

Steps

1. Browse to **Users**.
2. Click the name of the user to modify.

3. Click the **Status** control to toggle between Active or Inactive.
4. Choose **OK**.

The change is saved immediately.

3.4. Set the Ambari Admin Flag

You, as an Ambari administrator can grant one or more users Ambari administrator privileges by setting the Ambari Admin flag. Only an Ambari administrator can set or remove the Ambari Admin flag. Ambari prevents you from accidentally removing the flag from your own account.

As an Ambari administrator, on the Ambari Admin page:

Steps

1. Browse to **Users**.
2. Click the name of the user to modify.
3. Click the Ambari Admin control.
4. Click **Yes** to set or **No** to remove the Ambari Admin flag.

3.5. Change the Password for a Local User

An Ambari administrator can change local user passwords, but not LDAP user passwords.

As an Ambari administrator, on the Ambari Admin page:

Steps

1. Browse to **Users**.
1. Click **Change password**.
2. Enter your administrator password, to confirm that you have required privileges.
3. Enter a password, and then confirm that password.
4. Click **Save**.

3.6. Delete a Local User

Deleting a local user removes the user account from the system, including all privileges associated with the user. If you want only to disable user log in, set the user status to Inactive.

As an Ambari administrator, on the Ambari Admin page:

Steps

1. Browse to **Users**.
1. Click **Delete User**.
2. Confirm.



Note

You can reuse the name of a local user that has been deleted.

More Information

[Set User Status \[13\]](#)

3.7. Create a Local Group

As an Ambari administrator, on the Ambari Admin page:

Steps

1. Browse to **Groups**.
2. Click **Create Local Group**.
3. Enter a unique group name.
4. Click **Save**.

3.8. Managing Group Membership

You can manage membership of local groups by adding or removing users.

- [Add a User to a Group \[15\]](#)
- [Modify Group Membership \[16\]](#)

3.8.1. Add a User to a Group

As an Ambari administrator, on the Ambari Admin page:

Steps

1. Browse to **Groups**.
2. Click a name in the **Group Name** list.
3. Click the **Local Members** control to edit the member list.
4. In the empty space, type the first character in an existing user name.
5. From the list of available user names, click one.
6. Click the check mark to save the displayed members.

3.8.2. Modify Group Membership

As an Ambari administrator, on the Ambari Admin page:

Steps

1. Browse to **Groups**.
2. Click the name of the group to modify.
3. Click the **Local Members** control.
4. Click in the **Local Members** text area to modify the current membership.
5. Click the **x** next to the name of a user to remove that user.
6. To save your changes, click the check mark.

To discard your changes, click **x**.

3.9. Delete a Local Group

Deleting a local group also removes associated group membership information, including privileges.

As an Ambari administrator, on the Ambari Admin page:

Steps

1. Browse to the group.
2. Click **Delete Group**.
3. Confirm.

3.10. Enable User Home Directory Creation

A common requirement to initialize user accounts to run Hadoop components is the existence of a unique, `/user/<username>` HDFS home directory. You can enable automated creation of a `/user/<username>` HDFS home directory for each user that you create. Home directory creation occurs for users created either manually using the Ambari Admin page, or through LDAP synchronization.

To enable automated user home directory creation, perform the following steps on your Ambari Server host:

Steps.

1. Edit the `ambari-properties` file using a command line editor (`vi`, in this example).

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

2. Add the following property:

```
ambari.post.user.creation.hook.enabled=true.
```

3. Add the script path to the ambari properties file:

```
ambari.post.user.creation.hook=/var/lib/ambari-server/resources/  
scripts/post-user-creation-hook.sh
```

4. Restart Ambari server.

```
ambari-server restart
```



Important

In a Kerberized environment, you must modify the kinit file path in the default user creation hook script.

```
/var/lib/ambari-server/resources/scripts/post-user-creation-hook.sh
```

After enabling the post-user creation script, Ambari executes the script whenever a user is created and logs a message each time the script is invoked. If the script has a non-zero exit code, an ERROR is logged, otherwise an INFO-level message that includes the script path and parameters is logged.

4. Installing Ambari Agents Manually

In cases where you do not have SSH for Ambari to automatically install the Agents or you want to pre-install the Agents, you can perform a manual agent setup.

Steps

1. [Download the Ambari Repo \[18\]](#)
2. [Install the Ambari Agents Manually \[23\]](#)

4.1. Download the Ambari Repo

Select the OS family running on your installation host.

RHEL/CentOS/Oracle Linux 7

On a server host that has Internet access, use a command line editor to:

Steps

1. Log in to your host as `root`.
2. Download the Ambari repository file to a directory on your installation host.

```
wget -nv https://archive.cloudera.com/p/ambari/2.x/2.6.0.0/centos7/ambari.repo -O /etc/yum.repos.d/ambari.repo
```



Important

Do not modify the `ambari.repo` file name. This file is expected to be available on the Ambari Server host during Agent registration.

3. Confirm that the repository is configured by checking the repo list.

```
yum repolist
```

You should see values similar to the following for Ambari repositories in the list.

```
repo id                repo name
status
ambari-2.6.0.0-267     ambari Version - ambari-2.6.0.0-267      12
epel/x86_64            Extra Packages for Enterprise Linux 7 - x86_64
11,387
ol7_UEKR4/x86_64      Latest Unbreakable Enterprise Kernel Release 4
for Oracle Linux 7Server (x86_64)      295
ol7_latest/x86_64     Oracle Linux 7Server Latest (x86_64)
18,642
puppetlabs-deps/x86_64 Puppet Labs Dependencies El 7 - x86_64
17
puppetlabs-products/x86_64 Puppet Labs Products El 7 - x86_64
225
repolist: 30,578
```

Version values vary, depending on the installation.

4. Proceed to [Install the Ambari Agents manually](#).



Note

Accept the warning about trusting the Cloudera GPG Key. That key will be automatically downloaded and used to validate packages from Cloudera. You will see the following message:

```
Importing GPG key 0x07513CAD: Userid: "Jenkins (HDP
Builds) <jenkin@hortonworks.com>" From : https://
archive.cloudera.com/p/ambari/2.x/2.6.0.0/centos7/RPM-GPG-
KEY/RPM-GPG-KEY-Jenkins
```

RHEL/CentOS/Oracle Linux 6

On a server host that has Internet access, use a command line editor to:

Steps

1. Log in to your host as `root`.
2. Download the Ambari repository file to a directory on your installation host.

```
wget -nv https://archive.cloudera.com/p/ambari/2.x/2.6.0.0/centos6/ambari.
repo -O /etc/yum.repos.d/ambari.repo
```



Important

Do not modify the `ambari.repo` file name. This file is expected to be available on the Ambari Server host during Agent registration.

3. Confirm that the repository is configured by checking the repo list.

```
yum repolist
```

You should see values similar to the following for Ambari repositories in the list.

repo id	repo name	status
ambari-2.6.0.0-267	ambari Version - ambari-2.6.0.0-267	12
base	CentOS-6 - Base	6,696
extras	CentOS-6 - Extras	64
updates	CentOS-6 - Updates	974
repolist: 7,746		

Version values vary, depending on the installation.

4. Proceed to [Install the Ambari Agents manually](#).



Note

Accept the warning about trusting the Cloudera GPG Key. That key will be automatically downloaded and used to validate packages from Cloudera. You will see the following message:

```
Importing GPG key 0x07513CAD: Userid: "Jenkins (HDP
Builds) <jenkin@hortonworks.com>" From : https://
archive.cloudera.com/p/ambari/2.x/2.6.0.0/centos6/RPM-GPG-
KEY/RPM-GPG-KEY-Jenkins
```


SLES 11

On a server host that has Internet access, use a command line editor to:

Steps

1. Log in to your host as `root`.
2. Download the Ambari repository file to a directory on your installation host.

```
wget -nv https://archive.cloudera.com/p/ambari/2.x/2.6.0.0/suse11/ambari.repo -O /etc/zypp/repos.d/ambari.repo
```



Important

Do not modify the `ambari.repo` file name. This file is expected to be available on the Ambari Server host during Agent registration.

3. Confirm the downloaded repository is configured by checking the repo list.

```
zypper repos
```

You should see the Ambari repositories in the list.

#	Alias	Name	Enabled	Refresh
1	ambari-2.6.0.0-267	ambari Version - ambari-2.6.0.0-267	No	Yes
2	http-demeter.uni-regensburg.de-c997c8f9	SUSE-Linux-Enterprise-Software -Development-Kit-11-SP3 12.1.1-1.57	Yes	Yes
3	opensuse	OpenSuse	Yes	Yes

Version values vary, depending on the installation.

4. Proceed to [Install the Ambari Agents manually](#).

SLES 12

On a server host that has Internet access, use a command line editor to:

Steps

1. Log in to your host as `root`.
2. Download the Ambari repository file to a directory on your installation host.

```
wget -nv https://archive.cloudera.com/p/ambari/2.x/2.6.0.0/sles12/ambari.repo -O /etc/zypp/repos.d/ambari.repo
```



Important

Do not modify the `ambari.repo` file name. This file is expected to be available on the Ambari Server host during Agent registration.

3. Confirm the downloaded repository is configured by checking the repo list.

```
zypper repos
```

You should see the Ambari repositories in the list.

#	Alias	Name	Enabled	Refresh
1	ambari-2.6.0.0-267	ambari Version - ambari-2.6.0.0-267	Yes	No
2	http-demeter.uni -regensburg.de-c997c8f9	SUSE-Linux-Enterprise-Software -Development-Kit-12-SP1 12.1.1-1.57	Yes	Yes
3	opensuse	OpenSuse	Yes	Yes

Version values vary, depending on the installation.

4. Proceed to [Install the Ambari Agents manually](#).

Ubuntu 14

On a server host that has Internet access, use a command line editor to:

Steps

1. Log in to your host as `root`.
2. Download the Ambari repository file to a directory on your installation host.

```
wget -O /etc/apt/sources.list.d/ambari.list https://archive.cloudera.com/p/  
ambari/2.x/2.6.0.0/ubuntu14/ambari.list
```

```
apt-key adv --recv-keys --keyserver keyserver.ubuntu.com B9733A7A07513CAD
```

```
apt-get update
```



Important

Do not modify the `ambari.list` file name. This file is expected to be available on the Ambari Server host during Agent registration.

3. Confirm that Ambari packages downloaded successfully by checking the package name list.

```
apt-cache showpkg ambari-server
```

```
apt-cache showpkg ambari-agent
```

```
apt-cache showpkg ambari-metrics-assembly
```

You should see the Ambari packages in the list.

1. Proceed to [Install the Ambari Agents manually](#).

Ubuntu 16

On a server host that has Internet access, use a command line editor to:

Steps

1. Log in to your host as `root`.
2. Download the Ambari repository file to a directory on your installation host.

```
wget -O /etc/apt/sources.list.d/ambari.list https://archive.cloudera.com/p/ambari/2.x/2.6.0.0/ubuntu16/ambari.list
```

```
apt-key adv --recv-keys --keyserver keyserver.ubuntu.com B9733A7A07513CAD
```

```
apt-get update
```



Important

Do not modify the `ambari.list` file name. This file is expected to be available on the Ambari Server host during Agent registration.

3. Confirm that Ambari packages downloaded successfully by checking the package name list.

```
apt-cache showpkg ambari-server
```

```
apt-cache showpkg ambari-agent
```

```
apt-cache showpkg ambari-metrics-assembly
```

You should see the Ambari packages in the list.

4. Proceed to [Install the Ambari Agents manually](#).

Debian 7

On a server host that has Internet access, use a command line editor to:

Steps

1. Log in to your host as `root`.
2. Download the Ambari repository file to a directory on your installation host.

```
wget -O /etc/apt/sources.list.d/ambari.list https://archive.cloudera.com/p/ambari/2.x/2.6.0.0/debian7/ambari.list
```

```
apt-key adv --recv-keys --keyserver keyserver.ubuntu.com B9733A7A07513CAD
```

```
apt-get update
```



Important

Do not modify the `ambari.list` file name. This file is expected to be available on the Ambari Server host during Agent registration.

3. Confirm that Ambari packages downloaded successfully by checking the package name list.

```
apt-cache showpkg ambari-server
```

```
apt-cache showpkg ambari-agent
```

```
apt-cache showpkg ambari-metrics-assembly
```

You should see the Ambari packages in the list.

4. Proceed to [Install the Ambari Agents manually](#).

4.2. Install the Ambari Agents Manually

Use the instructions specific to the OS family running on your agent hosts.

RHEL/CentOS/Oracle Linux

Steps

1. Install the Ambari Agent on every host in your cluster.

```
yum install ambari-agent
```

2. Using a text editor, configure the Ambari Agent by editing the `ambari-agent.ini` file as shown in the following 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 on every host in your cluster.

```
ambari-agent start
```

The agent registers with the Server on start.

SLES

Steps

1. Install the Ambari Agent on every host in your cluster.

```
zypper install ambari-agent
```

2. Configure the Ambari Agent by editing the `ambari-agent.ini` file as shown in the following 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 on every host in your cluster.

```
ambari-agent start
```

The agent registers with the Server on start.

Debian/Ubuntu

Steps

1. Install the Ambari Agent on every host in your cluster.

```
apt-get install ambari-agent
```

2. Configure the Ambari Agent by editing the `ambari-agent.ini` file as shown in the following 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 on every host in your cluster.

```
ambari-agent start
```

The agent registers with the Server on start.

5. Customizing HDP Services

- [Defining Service Users and Groups for a HDP 2.x Stack \[25\]](#)
- [Setting Properties That Depend on Service Usernames/Groups \[26\]](#)

5.1. Defining Service Users and Groups for a HDP 2.x Stack

The individual services in Hadoop run under the ownership of their respective Unix accounts. These accounts are known as service users. These service users belong to a special Unix group. "Smoke Test" is a service user dedicated specifically for running smoke tests on components during installation using the `Services` View of the Ambari Web GUI. You can also run service checks as the "Smoke Test" user on-demand after installation. You can customize any of these users and groups using the `Misc` tab during the `Customize Services` installation step.



Note

Use the `Skip Group Modifications` option to not modify the Linux groups in the cluster. Choosing this option is typically required if your environment manages groups using LDAP and not on the local Linux machines.

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 .

Service Users

Service*	Component	Default User Account
Accumulo	Accumulo Tracer, Accumulo Monitor, Accumulo GC, Accumulo Master	accumulo (HDP 2.2 or later)
Ambari Metrics	Metrics Collector, Metrics Monitor	ams
Ambari Infra	Infra Solr Instance	infra-solr
Atlas	Atlas Metadata Server	atlas (HDP 2.3 or later)
Falcon	Falcon Server	falcon
Flume	Flume Agents	flume
HBase	MasterServer RegionServer	hbase
HDFS	NameNode SecondaryNameNode DataNode	hdfs
Hive	Hive Metastore, HiveServer2	hive
HUE	HUE	hue

Service*	Component	Default User Account
Kafka	Kafka Broker	kafka
Knox	Knox Gateway	knox
Mahout	Mahout clients	mahout (HDP 2.2 or later)
MapReduce2	HistoryServer	mapred
Oozie	Oozie Server	oozie
PostgreSQL	PostgreSQL (with Ambari Server)	postgres (Created as part of installing the default PostgreSQL database with Ambari Server. If you are not using the Ambari PostgreSQL database, this user is not needed.)
Ranger	Ranger Admin, Ranger Usersync	ranger (HDP 2.2 or later)
Ranger KMS	Ranger KMS Server	kms (HDP 2.3 or later)
Slider	Slider clients	slider
SmartSense	HST Server, HST Agent, Activity Analyzer, Activity Explorer	<same as ambari agent>
Spark	Lively Servers	livy
Spark	Spark History Server	spark (HDP 2.2 or later)
Sqoop	Sqoop	sqoop
Storm	Masters (Nimbus, DRPC Server, Storm REST API, Server, Storm UI Server) Slaves (Supervisors, Logviewers)	storm
Tez	Tez clients	tez
WebHCat	WebHCat Server	hcat
YARN	NodeManager ResourceManager	yarn
Zeppelin Notebook	Zeppelin Notebook	zeppelin
ZooKeeper	ZooKeeper	zookeeper

*For all components, 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 UI. The default user account for the smoke test user is ambari-qa.

Service Groups

Service	Components	Default Group Account
All	All	hadoop
Atlas	Atlas Metadata Server	atlas
Knox	Knox Gateway	knox
Ranger	Ranger Admin, Ranger Usersync	ranger
Ranger KMS	Ranger KMS Server	kms
Spark	Spark History Server	spark

5.2. Setting Properties That Depend on Service Usernames/Groups

Some properties must be set to match specific service user names or service groups. If you have set up non-default, customized service user names for the HDFS or HBase service or the Hadoop group name, you must edit the following properties, using **Services > Service.Name > Configs > Advanced**:

HDFS Settings: Advanced

- `dfs.permissions.superusergroup` 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".

MapReduce Settings: Advanced

- `mapreduce.cluster.administrators` A single space followed by the Hadoop group name.

6. Using Custom and Private Hostnames

The Ambari Server relies on the host names of Ambari Agent to communicate to operators which hosts are members of the cluster.

Ambari Agents can be configured to register with the Ambari Server using custom hostnames and public hostnames. Ambari uses the hostname of the agent to name and refer to that host in the Ambari web UI, for example in the **Hosts** list. The public hostname, if configured, is used as an alias for the host when referenced in configuration and is used in **Quick Links** URLs.

To determine whether to use a custom hostname or public hostname, consider the following scenarios:

Scenario	Configuration
If you have a host with the host name <i>revo1.hortonworks.local</i> , but you want it to show up in Ambari web UI as <i>c1r1.hortonworks.local</i> , you should configure a custom hostname.	Configure a Custom Host Name
If you have a host with the host name <i>revo1.hortonworks.local</i> and want to use a DNS CNAME of <i>nn1.hortonworks.local</i> to be used for Quick Links and as a configuration alias, you should configure a public hostname.	Configure a Public Host Name

6.1. Configure a Custom Host Name

It is recommended to not customize the hostname of an Ambari Agent after components have already been deployed to that host. Customizing the hostname for the Ambari Agent will result in a new agent being registered with the custom hostname, not the hostname of an existing agent being updated.

If you have not yet deployed components to the Ambari Agent, use the following steps to configure a custom hostname:

1. Edit the contents of the `/var/lib/ambari-agent/hostname.sh` script to return the hostname that you want the Ambari Agent to register with. Make sure that you **chmod** the script so it is executable by the user running the Ambari Agent. For most installations, 0755 is the sufficient permission to use.
2. As an example, the following script could be used to have the Ambari Agent register with the 'c1r1.hortonworks.local' hostname.

```
#!/bin/sh
echo `c1r1.hortonworks.local`
```

3. To configure the Ambari Agent to use this script, edit the `/etc/ambari-agent/conf/ambari-agent.ini` using a text editor.

4. Add the following property to the `[agent]` section:

```
hostname_script=/var/lib/ambari-agent/hostname.sh
```

5. In this case, Ambari Agent will use the `/var/lib/ambari-agent/hostname.sh` script to determine the hostname that it will use to register with the Ambari Server.
6. Restart the agent to ensure that it registers with the custom hostname:

```
ambari-agent restart
```

6.2. Configure a Public Host Name

It is common in large cluster deployments to use DNS aliases for specific hosts, so that configuration files mentioning those hosts do not need to be changed when the services on that host are moved to another physical machine. For example, if you have multiple deployed applications that write to HDFS, using a DNS alias instead of a physical hostname to refer to the HDFS NameNode allows you to move the NameNode to other physical machines without having to change those deployed application's HDFS client configuration.

In Ambari, individual hosts can be configured to use a public hostname when referencing individual hosts in configuration files or in **Quick Links**. For example, if you have a physical host with a FQDN of `rev01.hortonworks.local`, and you have a DNS CNAME that also points to that host using `nn1.hortonworks.local`, it is possible to configure Ambari to use `nn1.hortonworks.local` for **Quick Links** associated with that host, and whenever `nn1.hortonworks.local` is used in configuration, Ambari will understand that it is associated with the `rev01.hortonworks.local` host. That way if you need to move the NameNode to `rev04.hortonworks.local`, you can configure that new host to use `nn1.hortonworks.local` for its public hostname without having to make client configuration changes.



Note

You still have to modify the specific configuration properties that reference `rev01.hortonworks.local` and update them with the alias which you have chosen to use, `nn1.hortonworks.local` in this case, in order for this feature to work as expected.

To get started, follow the steps below for each host that you would like to configure with a public hostname:

1. Edit the contents of the `/var/lib/ambari-agent/public_hostname.sh` script to return the public hostname that you want the Ambari Agent to be configured with. Make sure that you **chmod** the script so it is executable by the user running the Ambari Agent. For most installations, 0755 is the sufficient permission to use.
2. As an example, the following script could be used to configure the Ambari Agent to use 'nn1.hortonworks.local' as the public hostname.

```
#!/bin/sh
echo `nn1.hortonworks.local`
```

3. To configure the Ambari Agent to use this script, edit the `/etc/ambari-agent/conf/ambari-agent.ini` using a text editor.
4. Add the following property to the `[agent]` section:

```
public_hostname_script=/var/lib/ambari-agent/public_hostname.sh
```

5. In this case, Ambari Agent will use the `/var/lib/ambari-agent/public_hostname.sh` script to determine the hostname that it will use as the public hostname in the Ambari Server.
6. Restart the agent for these changes to take effect.

```
ambari-agent restart
```

6.2.1. Public Hostname Limitations

This capability is currently available on the host level, and not the host-component level; that is, if you have a host which has a NameNode and ResourceManager on it, that host's public hostname will be used in the configuration for both the NameNode and the ResourceManager. It is not possible to specify a public hostname for just the NameNode or just the ResourceManager; instead it's only possible to specify the public hostname for the physical host that is running both components.

6.2.2. Checking if Public Hostnames Are Correctly Configured

You can use the Ambari REST API to double-check that Ambari has associated the public hostname configured for the Agent.

1. After logging in to Ambari, open a new tab and use the following URL to look at what has been configured for a specific host:

```
http://ambari.server:8080/api/v1/hosts/your.hosts.fqdn
```

2. Look for 'public_host_name' in the JSON returned from that request to ensure that the correct public hostname has been configured for the host in question.

7. Changing Host Names

Circumstances may require that you change the names of the hosts in your existing cluster. Beyond any infrastructure and environment changes you need to make, you must also change the host names that Ambari uses to manage the HDP cluster.

Prerequisites

- Make a backup of your Ambari database.
- Disable Kerberos.

Using **Ambari Web**, browse to **Admin > Kerberos** and click **Disable Kerberos**.

To change a host name in Ambari:

Steps

1. In **Ambari Web > Background Operations**, stop all pending commands and jobs.
2. Stop all services.
3. Stop `ambari-server` and `ambari-agents` on all hosts.

```
ambari-server stop
```

```
ambari-agent stop
```

4. Create `*.json` file with host names changes.

Example: `host_names_changes.json`

```
{
  "cluster1" : {
    "c6400.ambari.apache.org" : "c6410.ambari.apache.org",
    "c6401.ambari.apache.org" : "c6411.ambari.apache.org",
    ...
  }
}
```

where `cluster1` is cluster name and `"c6400.ambari.apache.org"`:
`"c6410.ambari.apache.org"` is the host names pair in the format `"current_host_name"`:
`"new_host_name"`.

5. Execute the following command on the `ambari-server` host:

```
ambari-server update-host-names host_names_changes.json
```

6. After successful end of this action, please update host names for all nodes, according to changes that you added to `*.json` file.
7. If you changed the host name for the node on which the `ambari-server` resides, then you must update that name for every `ambari-agent`.

In `/etc/ambari-agent/conf/ambari-agent.ini`, update the `"hostname"` field to the new host name for node on which the `ambari-server` resides.

8. Start ambari-server and ambari-agents on all hosts.

```
ambari-server start
```

```
ambari-agent start
```

9. If you have NameNode HA enabled, after starting the ZooKeeper service, you must:

- a. Start all ZooKeeper components.
- b. Execute the following command on one of the NameNode hosts:

```
hdfs zkfc -formatZK -force
```

10. Start all services, using Ambari Web.

For each, browse to **Services > service_name > Service Actions > Start**.

11. If you disabled Kerberos before starting this procedure, you must enable Kerberos security by working through either the automated or manual setup procedure. If you enable Kerberos with the manual option, you **must** be sure to generate and deploy new keytabs that contain the new host names.

More Information

[Enable Kerberos Security](#)

8. Moving the Ambari Server

To transfer an Ambari Server that uses the default, embedded, PostgreSQL database from one host to a new host:

Steps

1. [Back up current data](#) - from the original Ambari Server database.
2. [Update all Agents](#) - to point to the new Ambari Server.
3. [Install the New Ambari Server](#) - on the new host and populate databases with information from the original Server.

If your Ambari Server is using one of the non-default databases, such as MySQL, Oracle, or an existing PostgreSQL instance, you must use backup, restore, and stop/start procedures specific to that database type.

8.1. Back up Current Data

Steps

1. On the Ambari Server host, 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.db.username} -f ambari.sql
```

```
Password: {ambari.db.password}
```

where the following:

Variable	Description	Default
ambari.db.username	The database username.	ambari
ambari.db.password	The database password.	bigdata

4. Create a backup of the Ambari Server meta info.

```
ambari-server backup
```

8.2. Update all Agents

1. On each agent host, stop the agent.

```
ambari-agent stop
```

2. Remove old agent certificates (if any exist).

```
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.ambari.server.fqdn}
```

```
url_port=8440
```

```
secured_url_port=8441
```

8.3. Install the New Ambari Server

1. Install the new Ambari Server on the new host.

```
yum install ambari-server
```

2. Run setup the Ambari Server and setup similar to how the original Ambari Server is configured.

```
ambari-server setup
```

3. Restart the PostgreSQL instance.

```
service postgresql restart
```

4. Open the PostgreSQL interactive terminal.

```
su - postgres
```

```
psql
```

5. Using the interactive terminal, drop the "ambari" database created by the new ambari setup and install.

```
drop database ambari;
```

6. Check to make sure the databases have been dropped. The "ambari" databases should not be listed.

```
\l
```

7. Create new "ambari" database to hold the transferred data.

```
create database ambari;
```

8. Exit the PostgreSQL interactive terminal.

```
\q
```

9. Copy the saved data (`/tmp/dbdumps/ambari.sql`) from [Back up Current Data](#) to the new Ambari Server host.

10 Load the saved data into the new database.

```
psql -d ambari -f /tmp/dbdumps/ambari.sql
```

11 Start the new Server.

```
ambari-server start
```

12 On each Agent host, start the Ambari Agent.

```
ambari-agent start
```

13 Open Ambari Web. Point your browser to:

```
<new.Ambari.Server>:8080
```

The new Ambari Server is ready to use.

9. Moving the ZooKeeper Server

To move the ZooKeeper server to a new host:

Steps

1. In **Ambari Web** > **Service Actions**, stop the ZooKeeper server.
2. In **Ambari Web** > **Hosts**, click the host on which you want to install the new ZooKeeper server.
3. On the **Summary** page of the new ZooKeeper host, click **Add > ZooKeeper Server**
4. Update the following properties on the new ZooKeeper server (use the existing ZooKeeper server settings as a reference).
 - ha.zookeeper.quorum
 - hbase.zookeeper.quorum
 - templeton.zookeeper.hosts
 - yarn.resourcemanager.zk-address
 - hive.zookeeper.quorum
 - hive.cluster.delegation.token.store.zookeeper.connectString
5. In **Ambari Web** > **Hosts**, click the original ZooKeeper server host.
6. In **ZooKeeper** > **Service Actions** > **Delete Service** to delete the original ZooKeeper server.
7. Save the HDFS namespace.
8. Restart the new ZooKeeper server and the Hive service.



Note

In Ambari 2.4.0.0, adding or removing ZooKeeper servers requires manually editing the following Atlas properties. Select **Atlas** > **Configs** > **Advanced**, then select **Advanced application-properties** and edit the following properties to reflect the new ZooKeeper server settings:

- atlas.graph.index.search.solr.zookeeper-url

Example format:

```
host1:2181/infra-solr,host2:2181/infra-solr,host3:2181/infra-solr
```

- atlas.kafka.zookeeper.connect

Example format:

```
host1:2181,host2:2181,host3:2181
```

- atlas.audit.hbase.zookeeper.quorum

Example format:

```
host1,host2,host3
```

After updating these properties (to refresh the configuration files), restart Atlas and the following services that contain Atlas hooks :

- Hive
- Storm
- Falcon
- Sqoop

Next Steps

Review and confirm all recommended configuration changes.

More Information

[Review and Confirm Configuration Changes](#)

10. Configuring LZO Compression

LZO is a lossless data compression library that favors speed over compression ratio. Ambari does not install nor enable LZO compression by default. To enable LZO compression in your HDP cluster, you must configure `core-site.xml` for LZO.

Optionally, you can implement LZO to optimize Hive queries in your cluster for speed.

More Information

[Configure core-site.xml for LZO \[38\]](#)

[Running Compression with Hive Queries \[38\]](#)

10.1. Configure core-site.xml for LZO

To enable LZO compression:

Steps

1. Browse to **Ambari Web > Services > HDFS > Configs**, then expand **Advanced core-site**.
 2. Find the `io.compression.codecs` property key.
 3. Append to the `io.compression.codecs` property key, the following value:
`com.hadoop.compression.lzo.LzoCodec`
 4. Add a description of the config modification, then click **Save**.
 5. Expand the **Custom core-site.xml** section.
 6. Click **Add Property**.
 7. Add to Custom `core-site.xml` the following property key and value
- | | |
|-----------------------|--------------------------------------------------|
| Property Key | <code>io.compression.codec.lzo.class</code> |
| Property Value | <code>com.hadoop.compression.lzo.LzoCodec</code> |
8. Click **Save**.
 9. Add a description of the config modification, then click **Save**.
 10. Restart the HDFS, MapReduce2 and YARN services.

If performing a Restart or a Restart All does not start the required package install, you may need to stop, then start the HDFS service to install the necessary LZO packages. Restart is only available for a service in the Running or Started state.

10.2. Running Compression with Hive Queries

Running Compression with Hive Queries requires creating LZO files. To create LZO files, use one of the following procedures:

- [Create LZO Files \[39\]](#)
- [Write Custom Java to Create LZO Files \[39\]](#)

10.2.1. Create LZO Files

1. Create LZO files as the output of the Hive query.
2. Use `lzo` command utility or your custom Java to generate `lzo.index` for the `.lzo` files.

Hive Query Parameters

Prefix the query string with these parameters:

```
SET mapreduce.output.fileoutputformat.compress.codec=com.hadoop.compression.lzo.LzoCodec
SET hive.exec.compress.output=true
SET mapreduce.output.fileoutputformat.compress=true
```

For example:

```
hive -e "SET mapreduce.output.fileoutputformat.compress.codec=com.hadoop.compression.lzo.LzoCodec;SET hive.exec.compress.output=true;SET mapreduce.output.fileoutputformat.compress=true;"
```

10.2.2. Write Custom Java to Create LZO Files

1. Create text files as the output of the Hive query.
2. Write custom Java code to
 - convert Hive query generated text files to `.lzo` files
 - generate `lzo.index` files for the `.lzo` files

Hive Query Parameters

Prefix the query string with these parameters:

```
SET hive.exec.compress.output=false SET mapreduce.output.fileoutputformat.compress=false
```

For example:

```
hive -e "SET hive.exec.compress.output=false;SET mapreduce.output.fileoutputformat.compress=false;<query-string>"
```

11. Using Existing Databases

Ambari installs the PostgreSQL, MySQL, and Derby databases for use with Ambari, Hive, Oozie, and Ranger respectively, as default options. You may instead use a new, or an existing, non-default database instance with these components. To prepare Ambari to connect to a database for Hive or Oozie, you must download and set up database connectors before you set up the Ambari Server by running `ambari-server setup`.

- [Using Existing Databases - Ambari \[40\]](#)
- [Using New and Existing Databases - Hive \[45\]](#)
- [Using Existing Databases - Oozie \[49\]](#)
- [Configuring a Database Instance for Ranger](#)



Important

Using the **Microsoft SQL Server** or **SQL Anywhere** database options are not supported.

Using MySQL requires the default, InnoDB engine for MySQL 5.6.

More Information

[Hortonworks Support Matrix](#)

11.1. Using Existing Databases - Ambari

Before using Ambari with an existing database, other than the embedded PostgreSQL database instance that Ambari Server uses by default, perform the instructions in one of these database-specific topics:

- [Using Ambari with Oracle \[41\]](#)
- [Using Ambari with MySQL/MariaDB \[42\]](#)
- [Using Ambari with PostgreSQL \[43\]](#)
- [Troubleshooting Existing Databases with Ambari \[44\]](#)



Important

Using the **Microsoft SQL Server** or **SQL Anywhere** database options are not supported.



Important

For High Availability (HA) purposes, it is **required** that the relational database used with Ambari is also made highly available following best practices for the given database type.

11.1.1. Using Ambari with Oracle

To set up Oracle for use with Ambari:

Steps

1. On the Ambari Server host, install the appropriate JDBC . jar file.
 - a. Download the Oracle JDBC (OJDBC) driver from <http://www.oracle.com/technetwork/database/features/jdbc/index-091264.html>.

- b. For **Oracle Database 11g**: select

```
Oracle Database 11g Release 2 drivers > ojdbc6.jar
```

- c. For **Oracle Database 12c**: select

```
Oracle Database 12c Release 1 drivers > ojdbc7.jar
```

- d. Copy the .jar file to the Java share directory. For example:

```
cp ojdbc7.jar /usr/share/java/
```

- e. Make sure the .jar file has the appropriate permissions. For example:

```
chmod 644 /usr/share/java/ojdbc7.jar
```

2. Create a user for Ambari and grant that user appropriate permissions.

For example, using the Oracle database admin utility, run the following commands:

```
# sqlplus sys/root as sysdba
CREATE USER <AMBARIUSER> IDENTIFIED BY <AMBARIPASSWORD> default tablespace
"USERS" temporary tablespace "TEMP";
GRANT unlimited tablespace to <AMBARIUSER>;
GRANT create session to <AMBARIUSER>;
GRANT create TABLE to <AMBARIUSER>;
GRANT create SEQUENCE to <AMBARIUSER>;
QUIT;
```

Where <AMBARIUSER> is the Ambari user name and <AMBARIPASSWORD> is the Ambari user password.

3. Load the Ambari Server database schema.
 - a. You must pre-load the Ambari database schema into your Oracle database using the schema script.

```
sqlplus <AMBARIUSER>/<AMBARIPASSWORD> < Ambari-DDL-Oracle-CREATE.sql
```

- b. Find the Ambari-DDL-Oracle-CREATE.sql file in the /var/lib/ambari-server/resources/ directory of the Ambari Server host after you have installed Ambari Server.

4. When setting up the Ambari Server, select **Advanced Database Configuration** > **Option [2] Oracle** and respond to the prompts using the username/password credentials you created in step 2.

11.1.2. Using Ambari with MySQL/MariaDB

To set up MySQL/MariaDB for use with Ambari:

Steps

1. On the Ambari Server host:

- a. [Download the MySQL Connector/JDBC driver from MySQL](#).

- b. On the Ambari Server host run:

```
ambari-server setup --jdbc-db=mysql --jdbc-driver=/path/to/mysql/mysql-connector-java.jar
```

- c. Confirm that `.jar` is in the Java share directory.

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

- d. Make sure the `.jar` file has the appropriate permissions - 644.

2. Create a user for Ambari and grant it permissions.

- For example, using the MySQL database admin utility:

```
# mysql -u root -p
CREATE USER '<AMBARIUSER>'@'%' IDENTIFIED BY '<AMBARIPASSWORD>';
GRANT ALL PRIVILEGES ON *.* TO '<AMBARIUSER>'@'%' ;
CREATE USER '<AMBARIUSER>'@'localhost' IDENTIFIED BY '<AMBARIPASSWORD>';
GRANT ALL PRIVILEGES ON *.* TO '<AMBARIUSER>'@'localhost' ;
CREATE USER '<AMBARIUSER>'@'<AMBARISERVERFQDN>' IDENTIFIED BY
'<AMBARIPASSWORD>';
GRANT ALL PRIVILEGES ON *.* TO '<AMBARIUSER>'@'<AMBARISERVERFQDN>' ;
FLUSH PRIVILEGES;
```

- Where `<AMBARIUSER>` is the Ambari user name, `<AMBARIPASSWORD>` is the Ambari user password and `<AMBARISERVERFQDN>` is the Fully Qualified Domain Name of the Ambari Server host.

3. Load the Ambari Server database schema.

- You must pre-load the Ambari database schema into your MySQL/MariaDB database using the schema script. Run the script in the same location where you find the `Ambari-DDL-MySQL-CREATE.sql` file. You should find the `Ambari-DDL-MySQL-CREATE.sql` file in the `/var/lib/ambari-server/resources/` directory of the Ambari Server host, after you have installed Ambari Server.

```
mysql -u <AMBARIUSER> -p
CREATE DATABASE <AMBARIDATABASE>;
USE <AMBARIDATABASE>;
SOURCE Ambari-DDL-MySQL-CREATE.sql;
```

- Where <AMBARIUSER> is the Ambari user name and <AMBARIDATABASE> is the Ambari database name.
4. When setting up the Ambari Server, select **Advanced Database Configuration** > **Option [3] MySQL/MariaDB** and enter the credentials you defined in Step 2. for user name, password and database name.

11.1.3. Using Ambari with PostgreSQL

To set up PostgreSQL for use with Ambari:

Steps

1. [Download the PostgreSQL JDBC Driver from PostgreSQL.](#)
2. On the Ambari Server host run:

```
ambari-server setup --jdbc-db=postgres --jdbc-driver=/path/to/postgres/postgresql.jar
```

3. Create a user for Ambari and grant it permissions.

- Using the PostgreSQL database admin utility:

```
# sudo -u postgres psql
CREATE DATABASE <AMBARIDATABASE>;
CREATE USER <AMBARIUSER> WITH PASSWORD '<AMBARIPASSWORD>';
GRANT ALL PRIVILEGES ON DATABASE <AMBARIDATABASE> TO <AMBARIUSER>;
\connect <AMBARIDATABASE>;
CREATE SCHEMA <AMBARISCHEMA> AUTHORIZATION <AMBARIUSER>;
ALTER SCHEMA <AMBARISCHEMA> OWNER TO <AMBARIUSER>;
ALTER ROLE <AMBARIUSER> SET search_path to '<AMBARISCHEMA>', 'public';
```

- Where <AMBARIUSER> is the Ambari user name <AMBARIPASSWORD> is the Ambari user password, <AMBARIDATABASE> is the Ambari database name and <AMBARISCHEMA> is the Ambari schema name.
4. Load the Ambari Server database schema.
- You must pre-load the Ambari database schema into your PostgreSQL database using the schema script.

```
# psql -U <AMBARIUSER> -d <AMBARIDATABASE>
```



```
\connect <AMBARIDATABASE>;
```

```
\i Ambari-DDL-Postgres-CREATE.sql;
```

- Find the `Ambari-DDL-Postgres-CREATE.sql` file in the `/var/lib/ambari-server/resources/` directory of the Ambari Server host after you have installed Ambari Server.
5. When setting up the Ambari Server, select **Advanced Database Configuration** > **Option[4] PostgreSQL** and enter the credentials you defined in Step 2. for user name, password, and database name.

11.1.4. Troubleshooting Existing Databases with Ambari

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

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

Check `/var/log/ambari-server/ambari-server.log` for the following error:

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

The Oracle JDBC.jar file cannot be found.

11.1.4.1.1. Solution

Make sure the file is in the appropriate directory on the Ambari server and re-run

```
ambari-server setup
```

Review the load database procedure appropriate for your database type in [Using Existing Databases - Ambari](#).

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

Check `/var/log/ambari-server/ambari-server.log` for the following error:

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

Ambari Server cannot connect to the database.

11.1.4.2.1. Solution

Confirm that 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/ambari
```

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

Check `/var/log/ambari-server/ambari-server.log` for the following error:

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

You are using an invalid username/password.

11.1.4.3.1. Solution

Confirm the user account is set up in the database and has the correct privileges. See Step 3 above.

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

Check `/var/log/ambari-server/ambari-server.log` for the following error:

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

The schema has not been loaded.

11.1.4.4.1. Solution

Confirm you have loaded the database schema. Review the load database schema procedure appropriate for your database type.

More Information

[Using Existing Databases - Ambari \[40\]](#)

11.2. Using New and Existing Databases - Hive

Before using Hive with a new or existing database, including the embedded MySQL database instance that Ambari installs and Hive uses by default, perform the instructions in one of these database-specific topics:

- [Using Hive with Oracle \[45\]](#)
- [Using Hive with MySQL/MariaDB \[46\]](#)
- [Using Hive with PostgreSQL \[47\]](#)
- [Troubleshooting Existing Databases with Hive \[48\]](#)



Important

Using the **Microsoft SQL Server** or **SQL Anywhere** database options are not supported.

11.2.1. Using Hive with Oracle

To set up Oracle for use with Hive:

Steps

1. On the Ambari Server host, stage the appropriate JDBC driver file for later deployment.

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

b. For **Oracle Database 11g**: select

```
Oracle Database 11g Release 2 drivers > ojdbc6.jar
```

c. For **Oracle Database 12c**: select

```
Oracle Database 12c Release 1 drivers > ojdbc7.jar
```

d. Make sure the .jar file has the appropriate permissions. For example:

```
chmod 644 ojdbc7.jar
```

e. Execute the following command, adding the path to the downloaded .jar file:

```
ambari-server setup --jdbc-db=oracle --jdbc-driver=/path/to/downloaded/ojdbc7.jar
```

2. Create a user for Hive and grant it permissions.

- Using the Oracle database admin utility:

```
# sqlplus sys/root as sysdba
```

```
CREATE USER <HIVEUSER> IDENTIFIED BY <HIVEPASSWORD>;
```

```
GRANT SELECT_CATALOG_ROLE TO <HIVEUSER>;
```

```
GRANT CONNECT, RESOURCE TO <HIVEUSER>;
```

```
QUIT;
```

- Where <HIVEUSER> is the Hive user name and <HIVEPASSWORD> is the Hive user password.

11.2.2. Using Hive with MySQL/MariaDB

To set up MySQL/MariaDB for use with Hive:

Steps

1. On the Ambari Server host, stage the appropriate MySQL connector for later deployment.

a. [Download the MySQL Connector/JDBC driver from MySQL](#).

b. On the Ambari Server host run:

```
ambari-server setup --jdbc-db=mysql --jdbc-driver=/path/to/mysql/mysql-connector-java.jar
```

c. Confirm that

```
mysql-connector-java.jar
```

is in the Java share directory.

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

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

e. Execute the following command:

```
ambari-server setup --jdbc-db=mysql --jdbc-driver=/usr/share/java/mysql-connector-java.jar
```

2. Create a user for Hive and grant it permissions.

- Using the MySQL database admin utility:

```
# mysql -u root -p
CREATE USER '<HIVEUSER>'@'localhost' IDENTIFIED BY '<HIVEPASSWORD>';
GRANT ALL PRIVILEGES ON *.* TO '<HIVEUSER>'@'localhost';
CREATE USER '<HIVEUSER>'@'%' IDENTIFIED BY '<HIVEPASSWORD>';
GRANT ALL PRIVILEGES ON *.* TO '<HIVEUSER>'@'%';
CREATE USER '<HIVEUSER>'@'<HIVEMETASTOREFQDN>' IDENTIFIED BY
'<HIVEPASSWORD>';
GRANT ALL PRIVILEGES ON *.* TO '<HIVEUSER>'@'<HIVEMETASTOREFQDN>';
FLUSH PRIVILEGES;
```

- Where <HIVEUSER> is the Hive user name, <HIVEPASSWORD> is the Hive user password and <HIVEMETASTOREFQDN> is the Fully Qualified Domain Name of the Hive Metastore host.

3. Create the Hive database.

The Hive database must be created before loading the Hive database schema.

```
# mysql -u root -p
CREATE DATABASE <HIVEDATABASE>
```

Where <HIVEDATABASE> is the Hive database name.

11.2.3. Using Hive with PostgreSQL

To set up PostgreSQL for use with Hive:

Steps

1. On the Ambari Server host, stage the appropriate PostgreSQL connector for later deployment.
 - a. [Download the PostgreSQL JDBC Driver from PostgreSQL.](#)

- b. Confirm that `.jar` is in the Java share directory.

```
ls /usr/share/java/postgresql-jdbc.jar
```

- c. Change the access mode of the `.jar` file to 644.

```
chmod 644 /usr/share/java/postgresql-jdbc.jar
```

- d. Execute the following command:

```
ambari-server setup --jdbc-db=postgres --jdbc-driver=/usr/share/java/postgresql-jdbc.jar
```

2. Create a user for Hive and grant it permissions.

- Using the PostgreSQL database admin utility:

```
echo "CREATE DATABASE <HIVEDATABASE>;" | psql -U postgres
```

```
echo "CREATE USER <HIVEUSER> WITH PASSWORD '<HIVEPASSWORD>';" | psql -U postgres
```

```
echo "GRANT ALL PRIVILEGES ON DATABASE <HIVEDATABASE> TO <HIVEUSER>;" | psql -U postgres
```

- Where `<HIVEUSER>` is the Hive user name, `<HIVEPASSWORD>` is the Hive user password and `<HIVEDATABASE>` is the Hive database name.

11.2.4. Troubleshooting Existing Databases with Hive

Use these entries to help you troubleshoot any issues you might have installing Hive with existing databases.

11.2.4.1. 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.

11.2.4.1.1. Solution

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

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

Check the install log:

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

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

11.2.4.2.1. Solution

Ignore the warning, and complete the install. Check your database to confirm the Hive Metastore schema is loaded. In the Ambari Web GUI, browse to **Services > Hive**. Choose **Service Actions > Service Check** to check that the schema is correctly in place.

11.2.4.3. Problem: Hive Check Execute may fail after completing an Ambari upgrade to version 1.4.2

For secure and non-secure clusters, with Hive security authorization enabled, the Hive service check may fail. Hive security authorization may not be configured properly.

11.2.4.3.1. Solution

Two workarounds are possible. Using Ambari Web, in **HiveConfigsAdvanced**:

- Disable `hive.security.authorization`, by setting the `hive.security.authorization.enabled` value to `false`.

or

- Properly configure Hive security authorization. For example, set the following properties:

Table 11.1. Hive Security Authorization Settings

Property	Value
<code>hive.security.authorization.manager</code>	<code>org.apache.hadoop.hive.ql.security.authorization.StorageBasedAuthorizationProvider</code>
<code>hive.security.metastore.authorization.manager</code>	<code>org.apache.hadoop.hive.ql.security.authorization.StorageBasedAuthorizationProvider</code>
<code>hive.security.authenticator.manager</code>	<code>org.apache.hadoop.hive.ql.security.ProxyUserAuthenticator</code>

More Information

[Metastore Server Security](#)

[Hive Authorization](#)

11.3. Using Existing Databases - Oozie

Before using Oozie with an existing database, other than the Derby database instance that Ambari installs by default, perform the instructions in one of these database-specific topics:

- [Using Oozie with Oracle \[50\]](#)
- [Using Oozie with MySQL/MariaDB \[50\]](#)
- [Using Oozie with PostgreSQL \[51\]](#)
- [Troubleshooting Existing Databases with Oozie \[52\]](#)



Important

Using the **Microsoft SQL Server** or **SQL Anywhere** database options are not supported.

11.3.1. Using Oozie with Oracle

To set up Oracle for use with Oozie:

Steps

1. On the Ambari Server host, install the appropriate JDBC driver file.

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

b. For **Oracle Database 11g**: select

```
Oracle Database 11g Release 2 drivers > ojdbc6.jar
```

c. For **Oracle Database 12c**: select

```
Oracle Database 12c Release 1 drivers > ojdbc7.jar
```

d. Make sure the .jar file has the appropriate permissions. For example:

```
chmod 644 ojdbc7.jar
```

e. Execute the following command, adding the path to the downloaded .jar file:

```
ambari-server setup --jdbc-db=oracle --jdbc-driver=/path/to/downloaded/ojdbc7.jar
```

2. Create a user for Oozie and grant it permissions.

Using the Oracle database admin utility, run the following commands:

```
# sqlplus sys/root as sysdba  
  
CREATE USER <OOZIEUSER> IDENTIFIED BY <OOZIEPASSWORD>;  
  
GRANT ALL PRIVILEGES TO <OOZIEUSER>;  
  
GRANT CONNECT, RESOURCE TO <OOZIEUSER>;  
  
QUIT;
```

Where <OOZIEUSER> is the Oozie user name and <OOZIEPASSWORD> is the Oozie user password.

11.3.2. Using Oozie with MySQL/MariaDB

To set up MySQL/MariaDB for use with Oozie:

Steps

1. On the Ambari Server host, stage the appropriate MySQL connector for later deployment.

a. [Download the MySQL Connector/JDBC driver from MySQL.](#)

b. On the Ambari Server host run:

```
ambari-server setup --jdbc-db=mysql --jdbc-driver=/path/to/mysql/mysql-connector-java.jar
```

- c. Confirm that `mysql-connector-java.jar` is in the Java share directory.

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

- d. Make sure the `.jar` file has the appropriate permissions - 644.

- e. Execute the following command:

```
ambari-server setup --jdbc-db=mysql --jdbc-driver=/usr/share/java/mysql-connector-java.jar
```

2. Create a user for Oozie and grant it permissions.

- Using the MySQL database admin utility:

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

- Where `<OOZIEUSER>` is the Oozie user name and `<OOZIEPASSWORD>` is the Oozie user password.

3. Create the Oozie database.

- The Oozie database must be created prior.

```
# mysql -u root -p
CREATE DATABASE <OOZIEDATABASE>
```

- Where `<OOZIEDATABASE>` is the Oozie database name.

11.3.3. Using Oozie with PostgreSQL

To set up PostgreSQL for use with Oozie:

Steps

1. On the Ambari Server host, stage the appropriate PostgreSQL connector for later deployment.

- a. [Download the PostgreSQL JDBC Driver from PostgreSQL.](#)

- b. On the Ambari Server host run:

```
ambari-server setup --jdbc-db=postgres --jdbc-driver=/path/to/postgres/postgresql.jar
```

- c. Confirm that `.jar` is in the Java share directory.

```
ls /usr/share/java/postgresql-jdbc.jar
```


- d. Change the access mode of the .jar file to 644.

```
chmod 644 /usr/share/java/postgresql-jdbc.jar
```

- e. Execute the following command:

```
ambari-server setup --jdbc-db=postgres --jdbc-driver=/usr/share/java/postgresql-jdbc.jar
```

2. Create a user for Oozie and grant it permissions.

- Using the PostgreSQL database admin utility:

```
echo "CREATE DATABASE <OOZIEDATABASE>;" | psql -U postgres
```

```
echo "CREATE USER <OOZIEUSER> WITH PASSWORD '<OOZIEPASSWORD>';" | psql -U postgres
```

```
echo "GRANT ALL PRIVILEGES ON DATABASE <OOZIEDATABASE> TO <OOZIEUSER>;" | psql -U postgres
```

- Where <OOZIEUSER> is the Oozie user name, <OOZIEPASSWORD> is the Oozie user password and <OOZIEDATABASE> is the Oozie database name.

11.3.4. Troubleshooting Existing Databases with Oozie

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

11.3.4.1. 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.

11.3.4.1.1. Solution

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

11.3.4.2. 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.

11.3.4.2.1. Solution

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 wizard 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
```

12. Setting up a local repository

You need to set up a local repository, update the version repository base URLs, and edit the repository configuration file.

As a first step, you must set up a local repository for Ambari and HDP. For more information, see [set up a local repository](#). In case the public repository (<http://public-repo-1.hortonworks.com>) is not available anymore you can get the files from the official archive: <https://archive.cloudera.com>

A case-study of setting up local repositories: [case study for setting up local repository](#).

When the local repository is created, update the version repository base urls in Ambari. For more information, see [update the version repository base urls](#).

Finally, edit the repository configuration file to use this new local repository. For more information, see [edit the repository configuration file](#).

Updating Ambari repo files

Some services depend on components that are installed from the Ambari repository. It is not updated automatically. Also, Package Manager displays an error about the unavailable repository URL when you update the package lists. Cloudera recommends you to manually update the URLs located at `/etc/yum.repos.d/ambari.repo` on all hosts (including server host).

Updating HDP repo files

When the cluster settings for the HDP repository URL is updated, repository files on hosts are not immediately regenerated. The files are re-generated when you add a new component or service. But an inaccessible repository URL causes the Package Manager to display an error about the unavailable repository URL when you update the package lists. Cloudera recommends you to manually update the URLs in the HDP repository files (for example, `/etc/yum.repos.d/ambari-hdp-1.repo`) on all agent hosts.

12.1. Case study for setting up a local repository

Review this case study to understand how to prepare a local repository for Ambari 2.6.0.0 and HDP 2.6.5 on Centos7.

```
1. yum install yum-utils createrepo -y
```

```
2. yum install httpd -y
```

```
3. //firewall configuration
```

```
4. sudo systemctl start httpd
```

```
5. sudo systemctl status httpd
```

```
6. wget -nv
```

<https://archive.cloudera.com/p/ambari/2.x/2.6.0.0/centos7/ambari.repo> -O /etc/yum.repos.d/ambari.repo

7. `wget -nv`

<https://archive.cloudera.com/p/HDP/2.x/2.6.3.0/centos7/hdp.repo> -O /etc/yum.repos.d/hdp.repo

8. `yum repolist`

```
[root@santal-localrepo centos]# yum repolist
Loaded plugins: fastestmirror
Repository HDP-UTILS-1.1.0.22 is listed more than once in the configuration
HDP-2.6.5.0 | 2.9 kB 0
HDP-2.6.5.0/primary_db | 100 kB 0
Loading mirror speeds from cached hostfile
* base: mirrors.xtom.com
* extras: mirror.hostduplex.com
* updates: mirror.sjc02.svwh.net
repo id repo name
HDP-2.6.5.0 HDP Version - HDP-2.6.5.0
HDP-3.1.4.0 HDP Version - HDP-3.1.4.0
HDP-UTILS-1.1.0.22 HDP-UTILS Version - HDP-UTILS-1.1.0.22
ambari-2.6.2.0 ambari Version - ambari-2.6.2.0
ambari-2.7.4.0 ambari Version - ambari-2.7.4.0
base/7/x86_64 CentOS-7 - Base
extras/7/x86_64 CentOS-7 - Extras
updates/7/x86_64 CentOS-7 - Updates
repolist: 11,778
```

9. `mkdir -p /var/www/html/ambari/centos7`

10. `cd /var/www/html/ambari/centos7`

11. `reposync -r ambari-2.6.0.0`

12. `mkdir -p /var/www/html/hdp/centos7`

13. `cd /var/www/html/hdp/centos7`

14. `reposync -r HDP-2.6.3.0`

15. `reposync -r HDP-UTILS-1.1.0.22`

16. `createrepo /var/www/html/ambari/centos7/ambari-2.6.0.0/`

17. `createrepo /var/www/html/hdp/centos7/HDP-2.6.3.0`

18. `createrepo /var/www/html/hdp/centos7/HDP-UTILS-1.1.0.22/`

The repositories will be available at:

- <http://<web.server>/ambari/centos7/ambari-2.6.0.0/>
- <http://<web.server>/hdp/centos7/HDP-2.6.3.0/>

13. Setting up Ambari to use an Internet Proxy Server

If you plan to use **private repositories** (i.e. available on the Internet) for installing Apache Ambari and deploying a Hadoop stack in your cluster, you must provide Ambari and the hosts in the cluster Internet access to obtain the software from those repositories. Specifically:

- **Ambari Server:** uses Internet access to validate the repositories.
- **yum** (or an equivalent package manager, depending on your operating system): performs the software installation from the repositories.

Therefore, if your environment requires use of an Internet proxy server for access, you must configure Ambari Server component and yum on all the hosts to use that proxy server.

Ambari can install software if you have no Internet access. If you have no Internet access (via a proxy server or otherwise), you can use local repositories for installing the cluster software. In that case, configuring Ambari to use a proxy server is not required. However, Ambari and the hosts in the cluster must have access to your local repositories.

To configure internet proxy settings for Ambari Server:

Steps

1. On the Ambari Server host, stop Ambari Server:

```
ambari-server stop
```

2. Add proxy settings to the following script: `/var/lib/ambari-server/ambari-env.sh`.

```
-Dhttp.proxyHost=<yourProxyHost> -Dhttp.proxyPort=<yourProxyPort>
```

3. Optionally, to prevent some host names from accessing the proxy server, define the list of excluded hosts, as follows:

```
-Dhttp.nonProxyHosts=<pipe|separated|list|of|hosts>
```

4. If your proxy server requires authentication, add the username and password, as follows:

```
-Dhttp.proxyUser=<username> -Dhttp.proxyPassword=<password>
```

5. Restart the Ambari Server to pick up this change.

To configure yum to use internet proxy settings for all hosts:

Setting up yum to use a proxy server depends a lot on your environment and operating system. The instructions below provide some guidance but we **strongly recommend** you consult with your system administrators and operating system documentation for assistance & specific instructions.

1. On each host in the cluster, specify the proxy settings in `/etc/yum.conf` by adding the following entry:

```
proxy=http://<yourProxyHost>:<yourProxyPort>
```

2. If your proxy server requires authentication, add the username and password, as follows:

```
enableProxyAuth=1
```

```
proxy_username=<username>
```

```
proxy_password=<password>
```

3. Save the yum configuration file.

It is important to highlight that defining a proxy server, username and password in `/etc/yum.conf` means *all users of yum connect to the proxy server with those details*. Please consult your system administrators and refer to your operating system documentation for more details on this configuration and possible alternatives.

CentOS / Red Hat <https://www.centos.org/docs/5/html/yum/sn-yum-proxy-server.html>

Oracle Linux https://docs.oracle.com/cd/E37670_01/E37355/html/ol_proxy_config.html

Ubuntu / Debian <https://help.ubuntu.com/community/AptGet/Howto>

More Information

[Using a Local Repository](#)

14. Configuring Network Port Numbers

This chapter lists port number assignments required to maintain communication between Ambari Server, Ambari Agents, and Ambari Web.

- [Default Network Port Numbers - Ambari \[59\]](#)
- [Optional: Changing the Default Ambari Server Port \[60\]](#)
- [Optional: Changing the Ambari Server-Agent Port \[60\]](#)

More Information

[Configuring Ports](#)

14.1. Default Network Port Numbers - Ambari

The following table lists the default ports used by Ambari Server and Ambari Agent services.

Service	Servers	Default Ports Used	Protocol	Description	Need End User Access?	Configuration Parameters
Ambari Server	Ambari Server host	8080	http	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	
Ambari Agent	All hosts running Ambari Agents	8670 You can change the Ambari Agent ping port in the Ambari Agent configuration.	tcp	Ping port used for alerts to check the health of the Ambari Agent	No	

More Information

[Optional: Changing the Default Ambari Server Port \[60\]](#)

[Configure Ambari Server for Authenticated HTTP](#)

14.2. Optional: Changing the Default Ambari Server Port

By default, Ambari Server uses port 8080 to access the Ambari Web UI and the REST API. To change the port number, you must edit the Ambari properties file.

Ambari Server should not be running when you change port numbers. Edit `ambari.properties` before you start Ambari Server the first time or stop Ambari Server before editing properties.

Steps

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. Ambari Server now accesses Ambari Web via the newly configured port:

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

14.3. Optional: Changing the Ambari Server-Agent Port

By default, Ambari Server uses port 8187 to communicate with Ambari Agents. To change the port number, you must edit the Ambari properties file.

Ambari Server should not be running when you change port numbers. Edit `ambari.properties` before you start Ambari Server the first time or stop Ambari Server before editing properties.

Steps

1. On the Ambari Server host, open `/etc/ambari-server/conf/ambari.properties` with a text editor.
2. Add the following properties and set them to your desired port values:

```
security.server.two_way_ssl.port=5222 security.server.one_way_ssl.port=5223
```

3. On every Ambari Agent host, open `/etc/ambari-agent/conf/ambari-agent.ini` with a text editor.

4. Add the following properties and set them to your desired port values:

```
url_port=5223 secured_url_port=5222
```

5. Start or re-start the Ambari Server. Ambari Server now accesses Ambari Agents via the newly configured port:

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

15. Change the JDK Version

During your initial Ambari Server Setup, you select the JDK to use or provide a path to a custom JDK already installed on your hosts. After setting up your cluster, you may change the JDK version.

The choice of JDK depends on which HDP Stack you plan to install in your cluster. See the Hortonworks Support Matrix for version compatibility information.

To change the JDK version for an existing cluster:

Steps

1. Re-run Ambari Server Setup.

```
ambari-server setup
```

2. At the prompt to change the JDK, Enter **y**.

```
Do you want to change Oracle JDK [y/n] (n)? y
```

3. At the prompt to choose a JDK, Enter **1** to change the JDK to v1.8.

```
[1] - Oracle JDK 1.8 + Java Cryptography Extension (JCE) Policy Files 8
```

```
[2] - Oracle JDK 1.7 + Java Cryptography Extension (JCE) Policy Files 7
```

```
[3] - Custom JDK
```

4. If you choose Oracle JDK 1.8 or Oracle JDK 1.7, the JDK you choose downloads and installs automatically on the Ambari Server host. This option requires that you have an internet connection. You must install this JDK on all hosts in the cluster to this same path.
5. If you choose **Custom JDK**, verify or add the custom JDK path on all hosts in the cluster. Use this option if you want to use OpenJDK or do not have an internet connection (and have pre-installed the JDK on all hosts).



Important

If you use a custom JDK, *AND if Kerberos is enabled with AES-256 encryption*, you **must** also update your JCE security policy files on the Ambari Server and all hosts in the cluster **to match the new JDK version**. If you are running Kerberos and do not update the JCE to match the JDK, you will have issues starting services.

6. After setup completes, you must restart each component for the new JDK to be used by the Hadoop services.
7. Using the Ambari Web UI, restart all services.
8. Restart Ambari Server.

```
ambari-server restart
```



Note

If, after upgrading the JDK to 1.8, you experience issues with communication between Ambari Server and Ambari Agents, refer to *Java/Python updates and Ambari Agent TLS settings* in Hortonworks Community Connection for more information.

More Information

[Hortonworks Support Matrix](#)

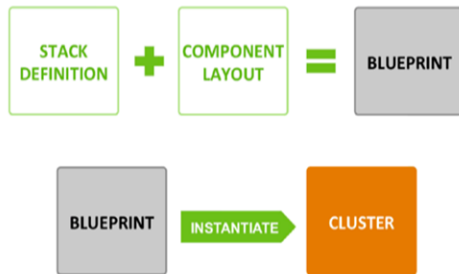
[Restart Components](#)

[Installing the JCE](#)

[Java/Python Updates and Ambari Agent TLS Settings](#)

16. Using Ambari Blueprints

Ambari Blueprints provide an API to perform cluster installations. You can build a reusable "blueprint" that defines which Stack to use, how Service Components should be laid out across a cluster and what configurations to set.



After setting up a blueprint, you can call the API to instantiate the cluster by providing the list of hosts to use. The Ambari Blueprint framework promotes reusability and facilitates automating cluster installations without UI interaction.

More Information

[Ambari Wiki - Blueprints](#)

17. Tuning Ambari Performance

For clusters larger than 100 nodes, consider the following tuning options:

- Increase available memory by adjusting heap size based on the number of cluster nodes.

Steps

1. On the Ambari Server host, edit the `ambari-env.sh` file:

```
vi /var/lib/ambari-server/ambari-env.sh
```

2. For the `AMBARI_JVM_ARGS` variable, replace the default `-Xmx2048m` with a value such as:

```
-Xmx4GB -Xmn2GB
```

based on the number of nodes in your cluster. Use the following recommendations as guidance:

# Cluster Nodes	Xmx value	Xmn value
100 - 400	4 GB	2 GB
400 - 800	4 GB	2 GB
800 - 1200	8 GB	2 GB
1200 - 1600	16 GB	2.4 GB

- Calculate the new, larger cache size, using the following relationship:

```
ecCacheSizeValue=60*<cluster_size>
```

where `<cluster_size>` is the number of nodes in the cluster.

- On the Ambari Server host, in `/etc/ambari-server/conf/ambari-properties`, add the following property and value:

```
server.ecCacheSize=<ecCacheSizeValue>
```

where `<ecCacheSizeValue>` is the value calculated previously, based on the number of nodes in the cluster.

- Add the following properties to adjust the JDBC connection pool settings:

```
server.jdbc.connection-pool.acquisition-size=5
```

```
server.jdbc.connection-pool.max-age=0
```

```
server.jdbc.connection-pool.max-idle-time=14400
```

```
server.jdbc.connection-pool.max-idle-time-excess=0
```

```
server.jdbc.connection-pool.idle-test-interval=7200
```

- If using MySQL as the Ambari database, in your MySQL configuration, increase the `wait_timeout` and `interactive_timeout` to 8 hours (28800) and max. connections from 32 to 128.



Important

It is **critical** that the Ambari configuration for `server.jdbc.connection-pool.max-idle-time` and `server.jdbc.connection-pool.idle-test-interval` must be lower than the MySQL `wait_timeout` and `interactive_timeout` set on the MySQL side. If you choose to decrease these timeout values, adjust `server.jdbc.connection-pool.max-idle-time` and `server.jdbc.connection-pool.idle-test-interval` accordingly in the Ambari configuration so that they are less than `wait_timeout` and `interactive_timeout`.

After performing one or more of these options, restart Ambari server for the option(s) to take effect.

```
ambari-server restart
```

If you are using the Ambari Metrics service, you might want to consider switching from the default embedded mode to distributed mode, as well as other tuning options.

More Information

[AMS Performance Tuning](#)

17.1. Purging Ambari Server Database History

After months of operation on larger clusters, the Ambari Server may begin to accrue a large amount of historical data in the database. This can cause UI performance degradation.

To remedy this, the following Ambari Server CLI command has been implemented to automate the removal of historical records in the Ambari Server. The `db-purge-history` command takes two arguments, the name of the cluster, and the date of the earliest record to purge.

Example Steps

In the following example we will purge history records created before August 1st, of 2017 for the cluster named 'Prod'.

1. Stop the Ambari Server by using `ambari-server stop`:

```
# ambari-server stop
Using python /usr/bin/python
Stopping ambari-server
Waiting for server stop...
Ambari Server stopped
```

2. Run the `db-purge-history` command:

```
# ambari-server db-purge-history --cluster-name Prod --from-date 2017-08-01
Using python /usr/bin/python
Purge database history...
Ambari Server configured for Embedded Postgres. Confirm you have made a
backup of the Ambari Server database [y/n] y
Ambari server is using db type Embedded Postgres. Cleanable database entries
older than 2017-08-01 will be purged. Proceed [y/n] y
Purging historical data from the database ...
Purging historical data completed. Check the ambari-server.log for details.
Ambari Server 'db-purge-history' completed successfully.
```

3. Start the Ambari Server: by using **ambari-server start**:

```
# ambari-server start
Using python /usr/bin/python
Starting ambari-server
Ambari Server running with administrator privileges.
Organizing resource files at /var/lib/ambari-server/resources...
Ambari database consistency check started...
Server PID at: /var/run/ambari-server/ambari-server.pid
Server out at: /var/log/ambari-server/ambari-server.out
Server log at: /var/log/ambari-server/ambari-server.log
Waiting for server start.....
Server started listening on 8080

DB configs consistency check: no errors and warnings were found.
Ambari Server 'start' completed successfully.
```

The `db-purge-history` command will analyze the following tables in the Ambari Server database and remove those rows that can be deleted that have a create date after the `from-date` specified when the command is run.

- AlertCurrent
- AlertNotice
- ExecutionCommand
- HostRoleCommand
- Request
- RequestOperationLevel
- RequestResourceFilter
- RoleSuccessCriteria
- Stage
- TopologyHostRequest
- TopologyHostTask
- TopologyLogicalTask

18. Customizing Ambari Log + PID Directories

Ambari Server and Agents write log activity output to .log files and use a .pid file that contains the process identification number for their running process. The log files and .pid file are found on their respective hosts in following default locations:

Ambari Server	<code>/var/log/ambari-server/ambari-server.log</code>
	<code>/var/run/ambari-server/ambari-server.pid</code>
Ambari Agent	<code>/var/log/ambari-agent/ambari-agent.log</code>
	<code>/var/run/ambari-agent/ambari-agent.pid</code>

You can configure the logging level for `ambari-server.log` by modifying `/etc/ambari-server/conf/log4j.properties` on the Ambari Server host. For the Ambari Agents, you can set the `loglevel` in `/etc/ambari-agent/conf/ambari-agent.ini` on each host running an Ambari Agent. In general, you should also consider setting log-rotate policies for your systems.

Refer to your operating system documentation for more information on setting up log-rotate in your environment.

You can also modify these locations. Use the following instructions:

- [Customizing Ambari Server Log + PID Directories \[68\]](#)
- [Customizing Ambari Agent Log + PID Directories \[69\]](#)

More Information

<http://linuxconfig.org/logrotate-8-manual-page>

18.1. Customizing Ambari Server Log + PID Directories

To modify the Ambari Server Log and PID locations:

Steps

1. On the Ambari Server host, stop the Ambari Server:

```
ambari-server stop
```

2. To modify the PID location, edit the Ambari Server properties file:

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

3. Modify the `pid.dir` property and save the file:

```
pid.dir=/var/run/ambari-server
```

4. To modify the Log location, edit the Ambari Server log4j file:

```
vi /etc/ambari-server/conf/log4j.properties
```

5. Modify the `ambari.log.dir` property and save the file:

```
ambari.log.dir=${ambari.root.dir}/var/log/ambari-server
```

6. Create the new directories and be sure to set the directory ownership and permissions to allow the Ambari Server process access.
7. Restart the Ambari Server.

```
ambari-server start
```

18.2. Customizing Ambari Agent Log + PID Directories

To modify the Ambari Agent Log and PID locations:

Steps

1. On each host running an Ambari Agent, stop the Ambari Agent:

```
ambari-agent stop
```

2. Edit the Ambari Agent properties file:

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

3. In the `[agent]` section, modify the `piddir` and `logdir` properties:

```
[agent]
```

```
logdir=/var/log/ambari-agent
```

```
piddir=/var/run/ambari-agent
```

4. Save the file.
5. Create the new directories and be sure to set the directory ownership and permissions to allow the Ambari Agent process access.
6. Restart the Ambari Agent.

```
ambari-agent start
```

19. Configuring Include File Management for HDFS and YARN

Both HDFS and YARN have the ability to control which hosts in the cluster should be included and excluded from participating in the cluster. HDFS uses the `dfs.hosts`, and `dfs.hosts.exclude` properties to control which set of datanodes are allowed to connect to the NameNode. YARN uses user-definable files configured through the `yarn.resourcemanager.nodes.include-path` and `yarn.resourcemanager.nodes.exclude-path` properties to control which nodes running the NodeManager component are allowed to communicate with the ResourceManager. When the contents of these files are modified both the HDFS NameNode and YARN ResourceManager need to be notified of these changes by invoking the `-refreshNodes` commands through `dfsadmin` for HDFS and `rmadmin` for YARN.

You can configure Ambari to manage these include files for both YARN and HDFS. This feature can be enabled just for HDFS or YARN, or enabled for both services. When enabled, Ambari will manage the associated include/exclude files and update their contents based on the state of hosts within Ambari. When the files are changed, Ambari will also call the necessary `refreshNodes` commands to update the state of the NameNode and/or ResourceManager.

The table below describes the actions Ambari will take for the following operations:

- *Add Component*: Adding a NodeManager, or DataNode
- *Delete Component*: Removing a NodeManager, or DataNode
- *Decommission Component*: Decommissioning a NodeManager, or DataNode
- *Recommission Component*: Recommissioning a NodeManager, or DataNode

Operation	Include File Actions	Exclude File Actions	Refresh Nodes Call	Triggers Master Restart Indicator
<i>Add Component</i>	Add hostname	Remove hostname	Yes	No
<i>Delete Component</i>	Remove hostname	Remove hostname	No	No
<i>Decommission Component</i>	Remove hostname (YARN only)	Add hostname	Yes	No
<i>Recommission Component</i>	Add hostname (YARN only)	Remove hostname	Yes	No

19.1. Enable Include File Management for HDFS

To enable Include File Management for HDFS:

1. In Ambari, add the `manage.include.files=true` property to the **Advanced hdfs-site** configuration section.
2. Ensure that the `dfs.hosts` property is configured in the **Custom hdfs-site** and that it is set to a valid location on the filesystem of the HDFS NameNode. Ensure that the file exists and is owned by the user being used to run the NameNode.

3. Restart services as prompted by Ambari.

Example Configuration

- In **Advanced hdfs-site**, set `manage.include.files=true`
- In **Custom hdfs-site**, set `dfs.hosts=/etc/hadoop/conf/dfs.include`

19.2. Enable Include File Management for Yarn

To enable Include File Management for Yarn:

1. In Ambari, add the `manage.include.files=true` property to the **Advanced yarn-site** configuration section.
2. Ensure that the `yarn.resourcemanager.nodes.include-path` is set to a valid location on the filesystem of the YARN Resource Manager. If the `yarn.resourcemanager.nodes.include-path` is not set, add it to the **Custom yarn-site** configuration.
3. Restart services as prompted by Ambari.

Example Configuration

- In **Advanced yarn-site**, set `manage.include.files=true`
- In **Custom yarn-site**, set `yarn.resourcemanager.nodes.include-path=/etc/hadoop/conf/yarn.include`

19.3. Disable Include File Management for HDFS

To disable Include File Management for HDFS:

1. In Ambari, set the `manage.include.files=false` property in the **Custom hdfs-site** configuration section.
2. In the same configuration section, remove the `dfs.hosts` property if configured and you no longer want HDFS to use include files for host management.
3. Restart services as prompted by Ambari.

19.4. Disable Include File Management for Yarn

To disable Include File Management for Yarn:

1. In Ambari, set the `manage.include.files=false` property in the **Custom yarn-site** configuration section.
2. In the same configuration section, remove the `yarn.resourcemanager.nodes.include-path` property if configured and you no longer want YARN to use include files for host management.

3. Restart services as prompted by Ambari.