

# Hortonworks Data Platform

Apache Ambari Administration

(August 30, 2016)

## Hortonworks Data Platform: Apache Ambari Administration

Copyright © 2012-2016 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. Administering Ambari: Overview .....	1
1. Terminology .....	1
2. Logging in to Ambari .....	2
3. About the Ambari Administration Interface .....	2
4. Changing the Default Admin Password .....	3
2. Create and Manage a Cluster and Cluster Roles .....	4
1. Creating a Cluster .....	4
2. Managing Cluster Roles .....	4
2.1. Cluster Roles .....	4
2.2. Role Comparison Chart .....	5
2.3. Modifying User and Group Cluster Roles .....	7
3. Launching the Cluster Dashboard .....	8
4. Renaming a Cluster .....	8
5. Managing Versions .....	9
6. Registering Remote Clusters .....	10
3. Managing Users and Groups .....	12
1. Users and Groups Overview .....	12
1.1. Local and LDAP User and Group Types .....	12
1.2. Ambari Admin Privileges .....	13
2. Creating a Local User .....	13
3. Setting User Status .....	13
4. Setting the Ambari Admin Flag .....	14
5. Changing the Password for a Local User .....	14
6. Deleting a Local User .....	14
7. Creating a Local Group .....	15
8. Managing Group Membership .....	15
8.1. Adding a User to a Group .....	15
8.2. Modifying Group Membership .....	15
9. Deleting a Local Group .....	16
4. Managing Views .....	17
1. Terminology .....	17
2. Basic Concepts .....	18
2.1. Ambari Views Versions and Instances .....	19
2.2. Deploying a View .....	19
3. Creating View Instances .....	19
4. Creating View URLs .....	20
5. Setting View Permissions .....	20
6. Configuring View Instances .....	21
7. Migrating from One View Instance to Another .....	22
8. Additional Information .....	23

# 1. Administering Ambari: Overview

Apache Ambari is a system to help you provision, manage and monitor Hadoop clusters. This guide is intended for Cluster Operators and System Administrators responsible for installing and maintaining Ambari and the Hadoop clusters managed by Ambari.

Installing Ambari creates a default user with "Admin Admin" privilege, with the following username/password: `admin/admin`. This *Ambari-level Administrator* user has full control over all aspects of Ambari. This includes the ability to fully control all clusters managed by the Ambari instance as well as managing users, clusters and views.

When you sign into Ambari as Ambari Admin, you can:

- [Create and Manage a Cluster and Cluster Roles](#)
- [Register and Manage Stack Versions](#)
- [Manage Users and Groups](#)
- [Manage Views](#)

For specific information about provisioning an HDP cluster, see [Install, Configure, and Deploy an HDP Cluster](#).

## 1. Terminology

The following basic terms help describe the key concepts associated with Ambari Administration.

Term	Definition
Ambari Admin	Specific privilege granted to a user that enables the user to administer Ambari. The default user <code>admin</code> created by Ambari is flagged as an "Ambari Admin". Users with the Ambari Admin privilege can grant, or revoke this privilege on other users.
Account	User name, password and privileges.
Cluster	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 (and synchronized) with an external LDAP (if configured).
Permissions	Represents the permission that can be granted to a principal (user or group) on a particular view. See <a href="#">Managing Views</a> for more information.
Principal	User or group that can be authenticated by Ambari.
Privilege	Represents the mapping of a principal to a permission or role and a resource. For example: the user <code>joe.operator</code> is granted the role of Cluster Operator on the cluster <code>DevCluster</code> .
Resource	Represents 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	Represents the role that is assigned to a principal (user or group) on a particular cluster. See <a href="#">Managing Cluster Roles</a> for more information.
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 (and synchronized) with an external LDAP (if configured).

Term	Definition
Version	Represents a Stack version, which includes a set of repositories to install that version on a cluster. For more information about Stack versions, see <a href="#">Managing Versions</a> .
View	Defines a user interface component that is available to Ambari.

## 2. Logging in to Ambari

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

1. Enter the following URL in a web browser:

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

where `<your.ambari.server>` is the hostname for your Ambari server machine and 8080 is the default HTTP port.

2. Enter the user account credentials for the default administrative user automatically created during install:

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

3. The [Ambari Administration web page](#) displays. From this page you can [Manage Users and Groups](#), [Manage Views](#), [Manage Versions](#), and [Create a Cluster](#).

## 3. About the Ambari Administration Interface

When you log in to the Ambari Administration interface, a landing page displays links to the operations available. Plus, the operations are available from the left menu for clusters, views, users, and groups.

The screenshot displays the Ambari Administration web interface. At the top, there is a header with the Ambari logo and the text 'Ambari', along with a user profile icon labeled 'admin'. On the left side, there is a navigation menu with three main sections: 'Clusters' (containing 'No clusters' and a 'Versions' link), 'Views' (containing a 'Views' link), and 'User + Group Management' (containing 'Users' and 'Groups' links). The main content area is titled 'Welcome to Apache Ambari' and includes the subtitle 'Provision a cluster, manage who can access the cluster, and customize views for Ambari users.' Below this, there are three primary action cards: 1. 'Create a Cluster' with the instruction 'Use the Install Wizard to select services and configure your cluster' and a 'Launch Install Wizard' button. 2. 'Manage Users + Groups' with the instruction 'Manage the users and groups that can access Ambari' and buttons for 'Users' and 'Groups'. 3. 'Deploy Views' with the instruction 'Create view instances and grant permissions' and a 'Views' button.

- Clusters displays a link to a cluster (if created) and links to manage access permissions for that cluster. See [Creating a Cluster](#) for more information.

- User and Group Management provides the ability create and edit users and groups. See [Managing Users and Groups](#) for more information.
- Views lets you to create and edit instances of deployed Views and manage access permissions for those instances. See [Managing Views](#) for more information.
- Versions provides the ability to manage the Stack versions that are available for the clusters. See [Managing Versions](#) for more information.

## 4. Changing the Default Admin Password

During install and setup, the Cluster Installer wizard automatically creates a default user with "Ambari Admin" privilege. You can change the password for this user (or other Local users in the system) from the Ambari Administration interface. You can change the password for the default `admin` user to create a unique administrator credential for your system.

To change the password for the default `admin` account:

1. Browse to the Users section.
2. Select the `admin` user.
3. Click the Change Password button.
4. Enter the current `admin` password and the new password twice.
5. Click OK to save the new password.

## 2. Create and Manage a Cluster and Cluster Roles

An Ambari-level Administrator user has full control over all aspects of Ambari, including the ability to create and manage a cluster. This section describes the abilities to create and manage clusters:

- [Creating a Cluster](#)
- [Managing Cluster Roles](#)
- [Launching the Cluster Dashboard](#)
- [Renaming a Cluster](#)
- [Managing Versions](#)
- [Registering Remote Clusters](#)

### 1. Creating a Cluster

To create a cluster, from the Ambari Administration interface:

1. Click **Install Cluster**. This displays the Cluster Install Wizard dialog.
2. Follow the steps in the wizard to install your cluster.

For more information about prerequisites and system requirements, see [Getting Ready](#) in **Hortonworks Data Platform Apache Ambari Automated Installation**.

### 2. Managing Cluster Roles

Ambari-level administrators can create different levels of cluster-level administrators to manage certain administrative operations that need to be performed on a cluster. This effectively spreads out the responsibilities of managing a cluster while not handing over total control of the Ambari management facility.

- [Cluster Roles](#)
- [Role Comparison Chart](#)
- [Modifying User and Group Cluster Roles](#)

#### 2.1. Cluster Roles

To enhance the granularity of permissions that may be granted to Ambari users, the following new, cluster-level roles are available:



### Note

In Ambari 2.2 and earlier, the only roles available were Operator and Read-only. Those roles are renamed to **Cluster Administrator** and **Cluster User** respectively.

#### Cluster User

Users assigned to the Cluster User role are able to view information about the cluster and its services, including configurations, service status and health alerts. Effectively, this is a “read-only” user.

#### Service Operator

Users assigned to the Service Operator role have control over service lifecycles, such as starting and stopping services, 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.



### Note

Ambari-level 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, as well as fully control all clusters managed by the Ambari instance.

## 2.2. Role Comparison Chart

	Cluster User	Service Operator	Service Administrator	Cluster Operator	Cluster Administrator	Ambari Administrator
<b>Service-level Authorizations</b>						
View metrics						
View status information						
View configurations						

	Cluster User	Service Operator	Service Administrator	Cluster Operator	Cluster Administrator	Ambari Administrator
Compare configurations						
View service alerts						
Start/Stop/Restart Service						
Decommission/recommission						
Run service checks						
Turn on/off maintenance mode						
Perform service-specific tasks						
Modify configurations						
Manage configuration groups						
Move to another host						
Enable HA						
Enable/disable service alerts						
Add Service to cluster						
<b>Host-level Authorizations</b>						
View metrics						
View status information						
View configuration						
Turn on/off maintenance mode						
Install components						
Add/Delete hosts						
<b>Cluster-level Authorizations</b>						

	Cluster User	Service Operator	Service Administrator	Cluster Operator	Cluster Administrator	Ambari Administrator
View metrics						
View status information						
View configuration						
View stack version details						
View alerts						
Enable/disable alerts						
Enable/disable Kerberos						
Upgrade/downgrade stack						
<b>Ambari-level Authorizations</b>						
Create new clusters						
Set service users and groups						
Rename clusters						
Manage users						
Manage groups						
Manage Ambari Views						
Assign permissions/roles						
Manage stack versions						
Edit stack repository URLs						

### 2.3. Modifying User and Group Cluster Roles

To modifying cluster roles:

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

- The Roles can be displayed in list view or block view. Use the icons to switch between the views.

	<p><b>Block View</b> displays the list of Cluster Roles and shows the assignment of users and groups to each role. You can add or modify the user and group assignment from this view.</p>
	<p><b>List View</b> displays the list of users and groups with cluster role assignments. You can modify the user and group assignment from this view.</p>

- Using the Block View, you can 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/group name and selecting the name.
- Using the List View, you can edit user and group assignments by finding the user or group, and adjusting the role using the dropdown.
- Click the check mark to save the current, displayed assignment.

## 3. Launching the Cluster Dashboard

After you have created a cluster, a link to **Go to Dashboard** is available to open the cluster operating dashboard. From the operating dashboard, you can manage and monitor the cluster services (to manage service lifecycle, change configurations, review alerts, etc ). For more information on using the cluster dashboard and on managing and monitoring your cluster services, see [Viewing the Ambari Dashboards](#) in the **Hortonworks Data Platform Apache Ambari User Guide**.

## 4. Renaming a Cluster

After you have created a cluster, you can rename the cluster using the Rename Cluster function.

To rename a cluster:

- In Clusters, click the **Rename Cluster** icon, next to the cluster name.

The cluster name becomes write-able.

- Enter alphanumeric characters as a cluster name.
- Click the check mark.
- Confirm.



### Important

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.



## Important

By changing the name of the cluster, the Ambari REST API resource for the cluster also changes. You must adjust any API calls you make to use this new name.

## 5. Managing Versions

This section describes the steps to manage versions of the cluster Stack software that are registered in Ambari. After you perform a cluster install, Ambari will automatically register the version of the Stack software. The Versions table will list the Stack, Name, Version and Cluster that is running that version.

- [Register New Version](#)
- [Upgrade Version Repository Base URLs](#)
- [Deregister Version](#)

### Register New Version

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 Public Repository requires Internet connectivity. Using a Local Repository requires you have configured the software in a repository available in your network. To use the public 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. Refer to [Using a Local Repository](#) for more information in setting-up a local repository in your environment.
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**.

#### Update Version Repository Base URLs

1. Browse to **Versions**. You will see a list of versions registered in Ambari, with one marked as "Current" for the cluster.
2. Click on the version you wish to modify.
3. Modify the Base URLs for the repositories. To use the public 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. Refer to [Using a Local Repository](#) for more information in setting-up a local repository in your environment.
4. Click **Save**.
5. 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.
6. Click **Confirm Change**.

#### Deregister Version

1. Browse to **Versions**. You will see a list of versions registered in Ambari, with one marked as "Current" for the cluster.
2. Click on the version you wish to modify. Only versions that are not installed in a cluster can be deregistered.
3. You will see a **Deregister Version** button. Click and confirm.

## 6. Registering Remote Clusters

You might have clusters that are being managed by Ambari but are not "local" to your Ambari Server (meaning: those clusters are "remote" and are being managed by a "remote" Ambari Server).

In cases where you plan to run an Ambari Server Standalone for hosting views, and you plan to access clusters that are being managed by a different Ambari Server, you can register those clusters from the standalone Ambari server as Remote Clusters. Once you have a Remote Cluster registered, you can use that Remote Cluster for configuring view instances.

To register a remote cluster:

1. Browse to **Remote Clusters** and click **Register Remote Cluster**.
2. Enter a name for this cluster, the Ambari Cluster URL, a Cluster User and associated password.
3. Click **Save**.
4. The **Remote Cluster** is now available to be used when [Configuring View Instances](#).

## 3. Managing Users and Groups

An "Ambari Admin" can create and manage users and groups available to Ambari. An Ambari Admin 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 Types](#)
- [Ambari Admin Privileges](#)
- [Creating a Local User](#)
- [Setting User Status](#)
- [Setting the Ambari Admin Flag](#)
- [Changing the Password for a Local User](#)
- [Deleting a Local User](#)
- [Creating a Local Group](#)
- [Managing Group Membership](#)
- [Deleting a Local Group](#)

### 1. Users and Groups Overview

Ambari supports two types of users and groups: Local and LDAP. The following topics describe how Ambari Administration supports managing Local and LDAP users and groups.

- [Local and LDAP User and Group Types](#)
- [Ambari Admin Privileges](#)

#### 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. For more information about running `ambari-server setup-ldap`, see [Configure Ambari to use LDAP Server](#). A new Ambari user or group, created either locally or by synchronizing against LDAP, is granted no privileges by default. You, as an Ambari Admin, must explicitly grant each user permissions to access clusters or views.

## 1.2. Ambari Admin Privileges

As an Ambari Admin, you 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 Admin for Local and LDAP Ambari users.

### Ambari Administrator Privileges for Ambari Local and LDAP Users

Administrator User 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 / Inactive	Available	Available

## 2. Creating a Local User

To create a local user:

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



### Note

All user names are converted to lowercase.

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

## 3. Setting User Status

User status indicates whether the user is active and should be allowed to log into 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.

To set user Status:

1. On the Ambari Administration interface, browse to Users.
2. Click the user name of the user to modify.
3. Click the Status control to toggle between Active or Inactive.
4. Choose OK to confirm the change. The change is saved immediately.

## 4. Setting the Ambari Admin Flag

You can elevate one or more users to have Ambari administrative privileges, by setting the Ambari Admin flag. You must be logged in as an account that is an Ambari Admin to set or remove the Ambari Admin flag.

To set the Ambari Admin Flag:

1. Browse to the Users section.
2. Click the user name you wish to modify.
3. Click on the Ambari Admin control.
4. Switch Yes to set, or No to remove the Admin flag.



### Important

To prevent you from accidentally locking yourself out of the Ambari Administration user interface, Ambari prevents setting the Ambari Admin flag for your own Ambari Admin account to No.

## 5. Changing the Password for a Local User

An Ambari Administrator can change local user passwords. LDAP passwords are not managed by Ambari since LDAP users authenticate to external LDAP. Therefore, LDAP user passwords cannot be changed from Ambari.

To change the password for a local user:

1. Browse to the user.
2. Click `Change password`.
3. Enter YOUR administrator password to confirm that you have privileges required to change a local user password.
4. Enter a password, then confirm that password.
5. Click `Save`.

## 6. Deleting a Local User

Deleting a local user removes the user account from the system, including all privileges associated with the user. You can reuse the name of a local user that has been deleted. To delete a local user:

1. Browse to the User.
2. Click `Delete User`.
3. Confirm.



### Note

If you want to disable user log in, [set the user Status](#) to Inactive.

## 7. Creating a Local Group

To create a local group:

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

## 8. Managing Group Membership

You can manage group membership of Local groups by adding or removing users from groups.

- [Adding a User to a Group](#)
- [Modifying Group Membership](#)

### 8.1. Adding a User to a Group

To add a user to group:

1. Browse to Groups.
2. Click a name in the Group Name list.
3. Choose 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, choose a user name.
6. Click the check mark to save the current, displayed members as group members.

### 8.2. Modifying Group Membership

To modify Local group membership:

1. In the Ambari Administration interface, browse to Groups.
2. Click the name of the Group to modify.
3. Choose the `Local Members` control to edit the member list.
4. Click in the Local Members text area to modify the current membership.

5. Click the **X** to remove a user.
6. To save your changes, click the checkmark. To discard your changes, click the **X**.

## 9. Deleting a Local Group

Deleting a local group removes all privileges associated with the group. To delete a local group:

1. Browse to the Group.
2. Click **Delete Group**.
3. Confirm. The group is deleted and the associated group membership information is removed.

## 4. Managing Views

The Ambari Views Framework offers a systematic way to plug in UI capabilities to surface custom visualization, management and monitoring features in Ambari Web. The development and use of Views allows you to extend and customize Ambari Web to meet your specific needs.

A View extends Ambari to let third parties plug in new resource types along with APIs, providers, and UIs to support them. A View is deployed into the Ambari Server and Ambari Admins can create View instances and set the privileges on access to users and groups.

The following sections cover the basics of Views and how to deploy and manage View instances in Ambari:

- [Terminology](#)
- [Basic Concepts](#)
- [Deploying Views](#)
- [Creating View Instances](#)
- [Creating View URLs](#)
- [Setting View Permissions](#)
- [Configuring View Instances](#)
- [Additional Information](#)



### Note

To learn more about using Views in a Standalone Ambari Server, and about Views that are included with Ambari by default, refer to [Hortonworks Data Platform Apache Ambari Views](#).

## 1. Terminology

The following are Views terms and concepts you should be familiar with:

Term	Description
Views Framework	The core framework that is used to develop a View. This is very similar to a Java Web App.
View Definition	Describes the View resources and core View properties such as name, version and any necessary configuration properties. On deployment, the View definition is read by Ambari.
View Package	Packages the View client and server assets (and dependencies) into a bundle that is ready to deploy into Ambari.
View Deployment	Deploying a View into Ambari. This makes the View available to Ambari Admins for creating instances.
View Name	Unique identifier for a View. A View can have one or more versions of a View. The name is defined in the View Definition (created by the View Developer) that is built into the View Package.
View Version	Specific version of a View. Multiple versions of a View (uniquely identified by View name) can be deployed into Ambari.

Term	Description
View Instance	Instantiation of a specific View version. Instances are created and configured by Ambari Admins and must have a unique View instance name.
View Instance Name	Unique identifier of a specific instance of View.
Framework Services	View context, instance data, configuration properties and events are available from the Views Framework.

## 2. Basic Concepts

Views are basically Web applications that can be “plugged into” Ambari. Just like a typical web application, a View can include server-side resources and client-side assets. Server-side resources, which are written in Java, can integrate with external systems (such as cluster services) and expose REST end-points that are used by the view. Client-side assets, such as HTML/JavaScript/CSS, provide the UI for the view that is rendered in the Ambari Web interface.

Ambari Views Framework Ambari exposes the Views Framework as the basis for View development. The Framework provides the following:

- Method for describing and packaging a View
- Method for deploying a View
- Framework services for a View to integrate with Ambari
- Method for managing View versions, instances, and permissions

The Views Framework is separate from Views themselves. The Framework is a core feature of Ambari and Views build on that Framework. Although Ambari does include some Views out-of-the-box, the feature of Ambari is the Framework to enable the development, deployment and creation of views.

The development and delivery of a View follows this process flow:

- Develop the View (similar to how you would build a Web application)
- Package the View (similar to a WAR)
- Deploy the View into Ambari (using the Ambari Administration interface)
- Create and configure instances of the View (performed by Ambari Admins)

Considering the above, it is important to understand the different personas involved. The following table describes the three personas:

Persona	Description
View Developer	Person who builds the front-end and back-end of a View and uses the Framework services available during development. The Developer created the View, resulting in a View Package that is delivered to an Ambari Admin.
Ambari Admin	Ambari user that has Ambari Admin privilege and uses the Views Management section of the Ambari Administration interface to create and managing instances of Views. Ambari Admin also deploys the View Packages delivered by the View Developer.
View User	Ambari user that has access to one or more Views in Ambari Web. Basically, this is the end user.



## Important

This document covers the tasks related to an Ambari Admin using and making Views available to users in their Ambari deployment. This document does not cover View development and packaging. See [Additional Information](#) for more information on where to obtain information about developing Views.

## 2.1. Ambari Views Versions and Instances

After Views are developed, views are identified by unique a view name. Each View can have one or more View versions. Each View name + version combination is deployed as a single View package. Once a View package is deployed, the Ambari Admin can create View instances, where each instance is identified by a unique View instance name. The Ambari Admin can then set access permissions for each View instance.

## 2.2. Deploying a View

Deploying a View involves obtaining the View Package and making the View available to the Ambari Server. Each View deployed has a unique name. Multiple versions of a View can be deployed at the same time. You can configure multiple versions of a View for your users, depending on their roles, and deploy these versions at the same time.

1. Obtain the View package. For example, `files-0.1.0.jar`.

2. On the Ambari Server host, browse to the views directory.

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

3. Copy the View package into place.

4. The View is extracted and registered with Ambari. Reload the Ambari Administration interface to show the newly deployed view. The View is now available to create instances.



## Note

`/var/lib/ambari-server/resources/views` is the default directory into which Views are deployed. You can change the default location by editing the `views.dir` property in `ambari.properties`.

For more information about building Views, see the [Apache Ambari Wiki page](#).

## 3. Creating View Instances

To create a View instance:

1. Browse to a View and expand.

2. Click the "Create Instance" button.

3. Provide the following information:

Item	Required	Description
View Version	Yes	Select the version of the View to instantiate.
Instance Name	Yes	Must be unique for a given View.
Display Label	Yes	Readable display name used for the View instance when shown in Ambari Web.
Description	Yes	Readable description used for the View instance when shown in Ambari Web.
Visible	No	Designates whether the View is visible or not visible to the end-user in Ambari web. Use this property to temporarily hide a view in Ambari Web from users.
Settings	Maybe	Depends on the View. If the View has a certain set of Settings that can be customized, they will appear in this section. If a Setting is required, you are prompted to provide the required information.
Cluster Configuration	Maybe	Depends on the View. If the View has a set of Configuration properties that can be derived from a cluster configuration, they will appear in this section.

If Ambari has a cluster configured that will work with the View instance, then the choice of **Local Cluster** will be available. If you have [registered one or more Remote Clusters](#), then the choice of **Remote Cluster** will also be available. If neither Local or Remote clusters are available, you will have to enter the **Custom** configuration manually. Refer to [Configuring View Instances](#) for more information on the configuration options.

## 4. Creating View URLs

The URL to access a View instance is based on the view name, version and instance name. In some cases, you want to provide a convenient “short url” to reference the view instance. This can be done using View URLs.

1. After you have created your view instance, browse to the **View URLs** section.
2. Click **Create New URL**.
3. Enter the URL name, select the View, select the Instance and enter the Short URL.
4. Click **Save**.

## 5. Setting View Permissions

After a view instance has been created, an Ambari Admin can set which users and groups can access the view by setting the Use permission. By default, after view instance creation, no permissions are set on a view.

To set permissions on a view:

1. Browse to a view and expand. For example, browse to the Slider or Jobs view.
2. Click on the view instance you want to modify.
3. In the Permissions section, click the Users or Groups control.
4. Modify the user and group lists as appropriate.

5. Click the check mark to save changes.



### Note

The Framework provides a way for view developers to specify custom permissions, beyond just the default Use permission. If custom permissions are specified, they will show up in the Ambari Administration interface and the Ambari Admin can set users and groups on these permissions. See [Additional Information](#) for more information on developing with the Views framework.



### Note

View permissions can also be inherited from Cluster roles. If you are using a Local Cluster for view configuration, you can optionally choose to provide view Use permission based on cluster roles.

## 6. Configuring View Instances

Besides providing the basic **Details** when [creating a view instance](#), configuring a view instance to talk to a cluster is handled in the Cluster Configuration section.



### Note

This section describes the view instance configuration options at a high-level. For specifics on configuring a specific view, refer to [Hortonworks Data Platform Apache Ambari Views](#) for details.

There are three options available for handling Cluster Configuration: Local Cluster, Remote Cluster, or Custom.

Option	Description
Local Cluster	If you are configuring this view instance in an Ambari Server that is also managing a cluster, you will be given an option to select that <b>Local Cluster</b> for view configuration. When you select this <b>Local Cluster</b> option, Ambari will automatically determine the cluster configuration properties needed for the view instance. If your Ambari Server is not managing a cluster, then you must use either the <b>Remote Cluster</b> or the <b>Custom</b> configuration option.
Remote Cluster	<p>If you plan to configure a view to work with a cluster that is remote from this Ambari Server AND that cluster is being managed by Ambari, you can use the option to configure the view based on the Remote Cluster. You must first <a href="#">register the Remote Cluster</a>, which enables the Remote Cluster option. When you select the <b>Remote Cluster</b> option, Ambari will automatically determine the cluster configuration properties needed for the view instance.</p> <div style="display: flex; align-items: center;"> <div> <p><b>Important</b></p> <p>Be sure the Remote Cluster includes the Required Services needed for the view, as specified in <a href="#">Hortonworks Data Platform Apache Ambari Views</a> for the view you are configuring.</p> </div> </div>
Custom	If your cluster is remote from the Ambari Server running the view, and that cluster is not being managed by Ambari, the Custom option exposes the various configuration properties that are needed to configure the view to work with the cluster.

### How to Choose the Cluster Configuration Option

The choice of configuration option depends on whether you have a Local cluster, or Remote cluster that is being managed (or not being managed) by Ambari. Use the

following table to help determine what options will be available for view configuration depending on the scenario.

Scenario	Option to Choose
Cluster is managed by a local Ambari Server (also) running the view	Local Cluster
Cluster is managed by Ambari AND Cluster is remote from the standalone Ambari Server running the view	Remote Cluster
Cluster is remote from the standalone Ambari Server running the view AND Cluster is not managed by Ambari.	Custom

## 7. Migrating from One View Instance to Another

If you have more than one instance of the same Ambari View, you can migrate view data (for example, entity data, instance data and View USE permissions) from one instance to another. This is useful when a new view version is released and you want to migrate the view data from a view instance for the previous version to a view instance on the newer version.



### Note

Migrating view data between instances is only supported for Hive, Pig and Tez views.

To migrate from view `INSTANCEA` (the source view instance) to view `INSTANCEB` (the target view instance), an Ambari Administrator can run the following command:

```
curl -v -u admin:admin -X PUT -H "X-Requested-By:1"
http://AMBARI_SERVER_HOST:8080/api/v1/views/VIEW_NAME/
versions/TARGET_VIEW_VERSION/instances/INSTANCEB/migrate/
SOURCE_VIEW_VERSION/INSTANCEA
```

where:

- `AMBARI_SERVER_HOST` is the Ambari Server hostname or IP address
- `VIEW_NAME` is the name of the view
- `TARGET_VIEW_VERSION` is the version of the target view
- `SOURCE_VIEW_VERSION` is the version of the source view

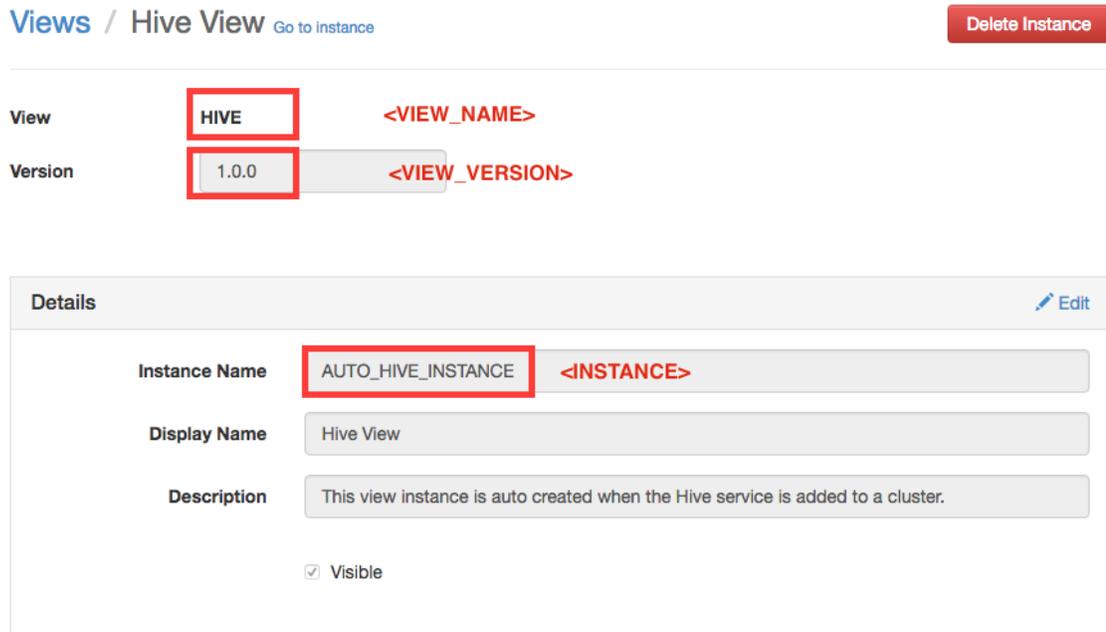
For example, if you are migrating from version 1.0.0 to 1.0.1, your `SOURCE_VIEW_VERSION` will be 1.0.0 and `TARGET_VIEW_VERSION` will be 1.0.1.

This command calls the method `/migrate` on the target instance and passes the original instance version and name.

To find out the view name, instance name, and view version:

1. In the Ambari web UI, browse to the Ambari Administration interface.
2. Click **Views**.
3. Open a specific view.

For example:



## 8. Additional Information

To learn more about developing views and the views framework itself, refer to the following resources:

Resource	Description	Link
Ambari Views Guide	Learn about the Views that are included with Ambari.	<a href="#">Hortonworks Data Platform Apache Ambari Views</a>
Views Wiki	Learn about the Views Framework and Framework services available to views developers.	<a href="https://cwiki.apache.org/confluence/display/AMBARI/Views">https://cwiki.apache.org/confluence/display/AMBARI/Views</a>
Views API	Covers the Views REST API and associated framework Java classes.	<a href="https://github.com/apache/ambari/blob/trunk/ambari-views/docs/index.md">https://github.com/apache/ambari/blob/trunk/ambari-views/docs/index.md</a>
Views Examples	Code for example views that cover different areas of the framework and framework services.	<a href="https://github.com/apache/ambari/tree/trunk/ambari-views/examples">https://github.com/apache/ambari/tree/trunk/ambari-views/examples</a>
View Contributions	Views that are being developed and contributed to the Ambari community.	<a href="https://github.com/apache/ambari/tree/trunk/contrib/views">https://github.com/apache/ambari/tree/trunk/contrib/views</a>