

Ambari User Views: Tech Preview



Welcome to **Hortonworks Ambari User Views Technical Preview**. This Technical Preview provides early access to upcoming features, letting you test and review during the development process. These features are considered under development and are not intended for use in your production systems and are not supported by Hortonworks but your feedback is greatly appreciated. Have fun and please send feedback on the Hortonworks Community forums <http://hortonworks.com/community/forums/forum/ambari>

Table of Contents

[Introduction](#)

[System Requirements](#)

[View Package Downloads](#)

[Install Ambari and HDP](#)

[Deploy the Views](#)

[Configure the Files View](#)

[Configuring HDFS](#)

[Creating a View Instance](#)

[Configure the Pig View](#)

[Configuring HDFS](#)

[Configuring WebHCat](#)

[Creating a View Instance](#)

[Configure the Hive View](#)

[Configuring HDFS](#)

[Creating a View Instance](#)

[Configure the Capacity Scheduler View](#)

[Creating a View Instance](#)

[Configure the Tez View](#)

[Creating a View Instance](#)

[Configuring Ambari User Views for Kerberos](#)

[Create Ambari Server Proxy User Principal and Keytab](#)

[Setup Ambari Server for Kerberos with Proxy User](#)

[Configure Cluster Services for Ambari Proxy User](#)

[Configure Ambari User Views for Kerberos Authentication](#)

[Files View](#)

[Hive View](#)

[Pig View](#)

[Capacity Scheduler View](#)

[Learn More About Views](#)

[Troubleshooting](#)

[Issue: When executing a Hive query from the View:](#)

[Issue: When loading the Capacity Scheduler View](#)

[Solution: Check the view configuration and confirm the Ambari Cluster URL and credentials provided are correct for your cluster.](#)

[Appendix: Users and Permissions](#)

[Set up Ambari and HDP Users](#)

[Manage View Access Permissions](#)

Introduction

The Hortonworks **Ambari User Views Technical Preview** provides a set of views to try-out with Ambari. For background on Ambari, the Ambari Views Framework, and the current Views in this Technical Preview review the following:

Apache Ambari	Operational platform for provisioning, managing and monitoring Apache Hadoop clusters. Ambari exposes a robust set of REST APIs and a rich Web interface for cluster management.
Ambari Views framework	Initially made available as part of Ambari 1.7.0 , the Ambari Views framework offers a systematic way to plug in UI capabilities to surface custom visualization, management and monitoring features in Ambari Web.
Ambari User Views	Specific extensions of the Ambari Web interface via “user views”. The Ambari User Views Technical Preview uses this the underlying Ambari Views framework to deliver views for Pig, Tez, Hive, Files and Capacity Scheduler .
Pig View	Provides a way to author and execute Pig scripts. See Configure the Pig View .
Hive View	Exposes a way to find, author and execute Hive queries. See Configure the Hive View .
Tez View	Introduced as part of Ambari 2.0 and HDP 2.2.4, it allows you to debug Hive Queries and Pig scripts. If configured, it can also be accessed from the Hive View. See Configure the Tez View .
Files View	Allows you browse the HDFS filesystem. See Configure the Files View .
Capacity Scheduler View	Provides a visual way to configure YARN capacity scheduler queue capacity.

	See Configure the Capacity Scheduler View .
--	---

System Requirements

The following lists the systems requirements for the Tech Preview:

- Ambari 2.0 + HDP 2.2.4 Cluster
- Install of HDFS, YARN + MapReduce2, Tez, Pig, Hive, ZooKeeper, Ambari Metrics

View Package Downloads

The following lists the software downloads for the **Ambari User Views Tech Preview**:

Item	Download URL
Ambari 2.0	Refer to the Ambari 2.0 documentation for information on installing and configuring Ambari. http://docs.hortonworks.com/HDPDocuments/Ambari-2.0.0.0/index.html
Files View	http://public-repo-1.hortonworks.com/HDP-LABS/Projects/Views/tp1/files-0.1.0-tp1.jar
Hive View	http://public-repo-1.hortonworks.com/HDP-LABS/Projects/Views/tp1/hive-0.2.0-tp1.jar
Pig View	http://public-repo-1.hortonworks.com/HDP-LABS/Projects/Views/tp1/pig-0.1.0-tp1.jar
Capacity Scheduler View	http://public-repo-1.hortonworks.com/HDP-LABS/Projects/Views/tp1/capacity-scheduler-0.3.0-tp1.jar
Tez View	Already packaged in Ambari 2.0.

Install Ambari and HDP

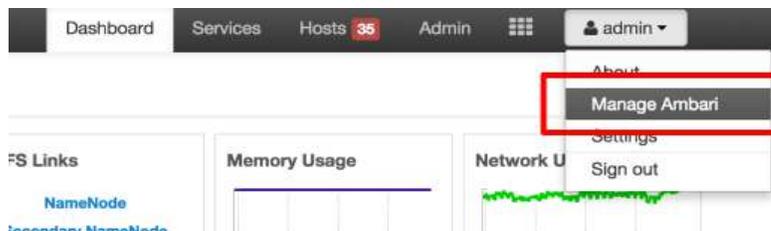
1. Download and install the Ambari 2.0 release.
<http://docs.hortonworks.com/HDPDocuments/Ambari-2.0.0.0/index.html>
2. Launch the Ambari **Cluster Install Wizard** and install an **HDP 2.2.4** cluster. **Be sure to include the following services at minimum: HDFS, YARN + MapReduce2, Tez, Hive, Pig, ZooKeeper and Ambari Metrics.**
3. Proceed to [Deploy the Views](#).

Deploy the Views

1. After [installing Ambari and HDP](#), download the [view packages](#) using the URLs above.
2. Copy the view packages to the Ambari Server in the views directory. For example:

```
cp *tpl.jar /var/lib/ambari-server/resources/views/
```
3. Restart the Ambari Server to deploy the views.

```
ambari-server restart
```
4. Once the views are deployed in Ambari Server, browse to the Ambari Web UI and login (as admin/admin):
<http://ambari.server:8080>
5. Go to the **Ambari Administration** interface by selecting **Manage Ambari**:



6. Follow configuration steps below specific to each view.
7. Refer to the Ambari Administration Guide for more information on Managing Views.

Configure the Files View

The **Files View** provides a convenient way to access HDFS through a web-based interface.

Configuring HDFS

You need to setup an HDFS proxy user for the Ambari daemon account. For example, if ambari-server daemon is running as `root`, you setup a proxy user for `root` in `core-site.xml` by clicking *HDFS -> Configs -> Custom core-site -> Add Property*:

```
hadoop.proxyuser.root.groups=*  
hadoop.proxyuser.root.hosts=*
```

Restart the required components as indicated by Ambari. If you want to add multiple views, you can restart services once after making changes for all views.

Creating a View Instance

1. Browse to the Ambari Administration interface.
2. Click Views, expand the **Files** view and click **Create Instance**.
3. Enter the following information:

Property	Value
Instance Name	FILES_1
Display Name	MyFiles
Description	Browse HDFS files and directories
WebHDFS FileSystem URI	webhdfs://<HDFS -> configs -> Advanced hdfs-site -> dfs.namenode.http-address> E.g. webhdfs://c6401.ambari.apache.org:50070
WebHDFS Username	\${username}

4. Click **Save**, Give Permissions to the appropriate users and groups, and go the view the instance.

Configure the Pig View

The **Pig View** provides a web based interface to compose, edit and submit queries, download results, view logs and view history of job submissions.

Configuring HDFS

You need to setup an HDFS proxy user for the Ambari daemon account. For example, if ambari-server daemon is running as `root`, you setup a proxy user for `root` in `core-site` by adding and changing properties in `HDFS -> Configs -> Custom core-site`:

```
hadoop.proxyuser.root.groups=*
hadoop.proxyuser.root.hosts=*
```

You need to setup an HDFS proxy user for WebHCat. For example, if your WebHCat Server is running as `hcat`, you setup a proxy user for `hcat` in `core-site`:

```
hadoop.proxyuser.hcat.groups=*
hadoop.proxyuser.hcat.hosts=*
```

Configuring WebHCat

You need to setup a WebHCat proxy user for the Ambari daemon account. Select *Hive* -> *Configs* -> *Custom webhcat-site* -> *Add Property*. For example, if `ambari-server` daemon is running as `root`, you setup a proxy user for `root` in `webhcat-site`:

```
webhcat.proxyuser.root.groups=*
webhcat.proxyuser.root.hosts=*
```

Creating a View Instance

1. Browse to the Ambari Administration interface.
2. Click Views, expand the **Pig** view and click **Create Instance**.
3. Enter the following information:

Property	Value
Instance Name	PIG_1
Display Name	MyPig
Description	Save and execute Pig scripts
WebHDFS FileSystem URI	webhdfs://<HDFS -> configs -> Advanced hdfs-site -> dfs.namenode.http-address> E.g. webhdfs://c6401.ambari.apache.org:50070
WebHCat URL	From <i>Hive</i> -> <i>Configs</i> http://<WebHCat Server host>:<templeton.port>/templeton/v1

	E.g http://c6401.ambari.apache.org:50111/templeton/v1
Jobs HDFS Directory	/user/\${username}/pig/jobs
Scripts HDFS Directory	/user/\${username}/pig/scripts

4. Click **Save**, Give Permissions to the appropriate users and groups, and go the view the instance.

Configure the Hive View

Configuring HDFS

You need to setup an HDFS proxy user for the Ambari daemon account. For example, if ambari-server daemon is running as `root`, you setup a proxy user for `root` in core-site by clicking *HDFS -> Configs -> Custom core-site -> Add Property*:

```
hadoop.proxyuser.root.groups=*
hadoop.proxyuser.root.hosts=*
```

In addition, it is necessary to setup HDFS users for all Ambari users that you plan on using; otherwise you will receive permission errors on job submissions and file save operations.

Creating a View Instance

1. Browse to the Ambari Administration interface.
2. Click Views, expand the **Hive** view and click **Create Instance**.
3. Enter the following information:

Property	Value
Instance Name	HIVE_1
Display Name	MyHive
Description	Save and execute Hive SQL scripts
WebHDFS FileSystem URI	webhdfs://<HDFS -> configs -> Advanced hdfs-site -> dfs.namenode.http-address> E.g.

	webhdfs://c6401.ambari.apache.org:50070
WebHCat URL	From <i>Hive -> Configs</i> http://<WebHCat Server host>:<templeton.port>/templeton/v1 E.g. http://c6401.ambari.apache.org:50111/templeton/v1
Jobs HDFS Directory	/user/\${username}/hive/jobs
Scripts HDFS Directory	/user/\${username}/hive/scripts
Hive Server 2 Host	Click <i>Hive -> Summary -> HiveServer2</i> and note the host name E.g. c6401.ambari.apache.org
Hive Server 2 Port	< <i>Hive -> Configs-> Hive Server Port</i> > E.g. 10000
Hive Auth	auth=NONE;user=\${username}
yarn.ats.url*	http://<YARN -> Configs -> yarn.timeline-service.webapp.address> E.g. http://c6401.ambari.apache.org:8188

4. Click **Save**. Give Permissions to the appropriate users and groups, and go the view the instance.



The **Tez View** integrates with the Hive View. Please install the [Tez View](#) when you install the Hive View.

Configure the Capacity Scheduler View

The **Capacity Scheduler View** provides a UI to manage the YARN capacity scheduler queues eliminating cumbersome editing of the underlying XML file typically used to configure the capacity scheduler. The view uses the Ambari REST API to read and modify the capacity scheduler configuration file.

Creating a View Instance

1. Browse to the Ambari Administration interface.

2. Click Views, expand the **Capacity-Scheduler** view and click **Create Instance**.
3. Enter the following information:

Property	Value
Instance Name	CS_1
Display Name	MyQueueManager
Description	Browse and manage YARN Capacity Scheduler queues
Ambari Cluster URL	http://<Ambari Host>:8080/api/v1/clusters/<clusterName> E.g. http://c6401.ambari.apache.org:8080/api/v1/clusters/fo
Operator Username	<operatorUsername>
Operator Password	<operatorPassword>

4. Click **Save**, Give Permissions to the appropriate users and go the view the instance.

Configure the Tez View

Tez is an execution engine that can be used for Hive and Pig. The **Tez View** provides the ability to understand and debug Tez task execution, during and after execution.



The **Tez View** is included with Ambari 2.0. Separate download is not required as it is GA and comes pre-deployed in Ambari 2.0

Creating a View Instance

1. Browse to the Ambari Administration interface.
2. Click Views, expand the **Tez** view and click **Create Instance**.
3. Enter the following information:

Property	Value
Instance Name	TEZ_1
Display Name	MyTez
Description	Find, understand and debug Tez jobs
YARN Timeline Server URL*	<i>http://<YARN -> Configs -> yarn.timeline-service.webapp.address</i> E.g. http://c6401.ambari.apache.org:8188
YARN ResourceManager URL*	<i>http://<YARN -> Configs -> yarn.resourcemanager.webapp.address</i> E.g. http://c6401.ambari.apache.org:8088

4. Click **Save**. Give Permissions to the appropriate users and groups, and go the view the instance.

Configuring Ambari User Views with a Secure Cluster

This document describes how to configure the Ambari User Views to work with a Kerberos-enabled cluster.

- [Create Ambari Server Proxy User Principal and Keytab](#)
- [Setup Ambari Server for Kerberos with Proxy User](#)
- [Configure Cluster Services for Ambari Proxy User](#)
- [Configure Ambari User Views for Kerberos Authentication](#)

Create Ambari Server Proxy User Principal and Keytab

Using the Kerberos `kadmin` utility [in the case of MIT KDC], create the Ambari Service principal that will act as the Proxy User. The name used for the principal is arbitrary. For this example, we will use:

```
ambari-user/ambari-server-host@EXAMPLE.COM
```



The Hive View requires that the principal name is comprised of three parts: the primary user, the instance and the realm (separated by “/” and “@”). For example: `primary/instance@realm`.

A sample session with `kadmin` could look as follows:

```
root@kdc-1 ~]# kadmin.local
Authenticating as principal root/admin@EXAMPLE.COM with password.
kadmin.local: add_principal -pw hadoop ambari-user/ambari-server-host
WARNING: no policy specified for ambari-user/ambari-server-host@EXAMPLE.COM;
defaulting to no policy
Principal "ambari-user/ambari-server-host@EXAMPLE.COM" created.
```

Next, generate a keytab for the newly created principal:

```
kadmin.local: xst -norandkey -k ambari-user.keytab
ambari-user/ambari-server-host@EXAMPLE.COM
Entry for principal ambari-user/ambari-server-host@EXAMPLE.COM with kvno 1,
encryption type aes256-cts-hmac-sha1-96 added to keytab
WRFILE:ambari-user.keytab.
Entry for principal ambari-user/ambari-server-host@EXAMPLE.COM with kvno 1,
encryption type aes128-cts-hmac-sha1-96 added to keytab
WRFILE:ambari-user.keytab.
Entry for principal ambari-user/ambari-server-host@EXAMPLE.COM with kvno 1,
encryption type des3-cbc-sha1 added to keytab WRFILE:ambari-user.keytab.
```

```
Entry for principal ambari-user/ambari-server-host@EXAMPLE.COM with kvno 1,
encryption type arcfour-hmac added to keytab WRFILE:ambari-user.keytab.
Entry for principal ambari-user/ambari-server-host@EXAMPLE.COM with kvno 1,
encryption type des-hmac-sha1 added to keytab WRFILE:ambari-user.keytab.
Entry for principal ambari-user/ambari-server-host@EXAMPLE.COM with kvno 1,
encryption type des-cbc-md5 added to keytab WRFILE:ambari-user.keytab.
```

Copy the generated keytab (`ambari-user.keytab` in this case) to the Ambari Server host. Keytabs are by convention, typically stored in `/etc/security/keytabs` but the location is arbitrary.

```
/etc/security/keytabs/ambari-user.keytab
```

Setup Ambari Server for Kerberos with Proxy User

Once the keytab is located on the Ambari Server host, stop the Ambari Server:

```
ambari-server stop
```

Next, configure Ambari Server to kinit as the Proxy User principal. This is accomplished by running “`ambari-server setup-security`” from the command line and following the prompts for “Setup Ambari Kerberos JAAS configuration”. For example:

```
[root@ambari-server-host]# ambari-server setup-security
Using python /usr/bin/python2.6
Security setup options...
=====
Choose one of the following options:
 [1] Enable HTTPS for Ambari server.
 [2] Encrypt passwords stored in ambari.properties file.
 [3] Setup Ambari kerberos JAAS configuration.
=====
Enter choice, (1-3): 3
Setting up Ambari kerberos JAAS configuration to access secured Hadoop daemons...
Enter ambari server's kerberos principal name (ambari@EXAMPLE.COM):
ambari-user/ambari-server-host@EXAMPLE.COM
Enter keytab path for ambari server's kerberos principal:
/etc/security/keytabs/ambari-user.keytab
Ambari Server 'setup-security' completed successfully.
```

Next, start Ambari Server:

```
ambar-server start
```

Configure Cluster Services for Ambari Proxy User

Just as you would configure proxyuser in the non-kerberos case, ensure that you are allowing the ambariuser to act as proxyuser for the required services. Please refer to the view configuration document for details.

In addition to the above configurations, you should also add to hive-site.xml:

```
hive.server2.enable.impersonation = true
```

Configure Ambari User Views for Kerberos Authentication

Depending on the View you are configuring, following the instructions below:

- [Files View](#)
- [Hive View](#)
- [Pig View](#)
- [Capacity Scheduler View](#)

Files View

The Files view requires that `auth=Kerberos;proxyuser=<ambari-principal>`; for example:

Properties		Edit
WebHDFS FileSystem URI*	<input type="text" value="webhdfs://erik-views-1.c.pramod-thangali.internal:50070"/>	
WebHDFS Username	<input type="text" value="{username}"/>	
webhdfs.auth	<input type="text" value="auth=KERBEROS;proxyuser=ambariuser"/>	



There must be an ambari-user HDFS user.

Hive View

Hive requires that WebHDFS Authentication is set to:

`auth=KERBEROS;proxyuser=<ambari-principal>` and that “Hive Authentication” is set to KERBEROS AND that the principal is set to the same principal as what is specified in hive-site.xml for `hive.server2.authentication.kerberos.principal`. For example:

Pig View

Pig View requires little extra configuration to function with a kerberized cluster:

webhdfs Authentication needs to be set to

```
auth=KERBEROS;proxyuser=<ambari-user-principal>
```

For example:

Properties Edit	
WebHDFS FileSystem URI*	webhdfs://erik-views-1.c.pramod-thangali.internal:50070
WebHDFS Username	\${username}
WebHDFS Authentication	auth=KERBEROS;proxyuser=ambariuser
WebHCat URL*	http://erik-views-3.c.pramod-thangali.internal:50111/templeton/v1
WebHCat Username	\${username}
Dataworker Username	\${username}
Scripts HDFS Directory*	/user/\${username}/pig/scripts
Jobs HDFS Directory*	/user/\${username}/pig/jobs
Meta HDFS Directory	/user/\${username}/pig/store

Capacity Scheduler View

The Capacity Scheduler view does not require any additional configuration to function with a secured cluster.

Learn More About Views

- Learn more about Apache Ambari here:
<http://ambari.apache.org/>
- Learn more about Ambari Views here:
<https://cwiki.apache.org/confluence/display/AMBARI/Views>
- Browse view example code here:
<https://github.com/apache/ambari/tree/trunk/ambari-views/examples>
- Learn more about the Pig View code here:
<https://github.com/apache/ambari/tree/trunk/contrib/views/pig>
- Learn more about the Files View code here:
<https://github.com/apache/ambari/tree/trunk/contrib/views/files>
- Learn more about the Hive View code here:
<https://github.com/apache/ambari/tree/trunk/contrib/views/hive>
- Learn more about the Capacity Scheduler View code here:
<https://github.com/apache/ambari/tree/trunk/contrib/views/capacity-scheduler>

Questions, Issues, Feedback?

- Use the Ambari User Views Technical Preview Forum here:
<http://hortonworks.com/community/forums/forum/ambari-user-views/>

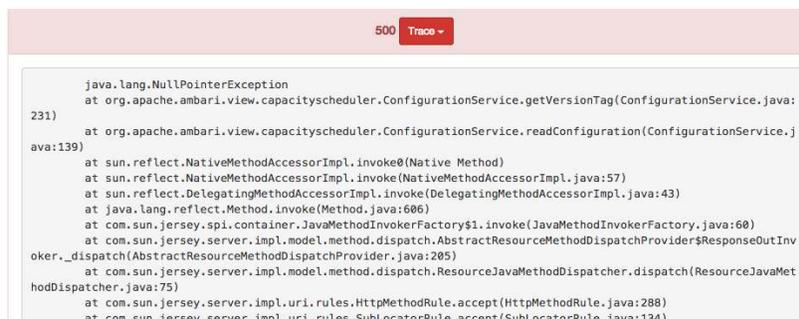
Troubleshooting

Issue: When executing a Hive query from the View:

```
{"message":"Error in creation: org.apache.hadoop.security.authorize.AuthorizationException: User: root is not allowed to impersonate admin","status":500,"trace":""}
```

Solution: Configure the Hadoop proxy user configuration for the account running the Ambari Server.

Issue: When loading the Capacity Scheduler View



The screenshot shows a web browser interface with a red header bar containing the text "500 Trace". Below the header is a white box containing a Java stack trace. The stack trace starts with "java.lang.NullPointerException" and lists several frames from the Ambari Capacity Scheduler view, including "ConfigurationService.getVersionTag", "ConfigurationService.readConfiguration", and various Jersey framework classes like "NativeMethodAccessorImpl", "DelegatingMethodAccessorImpl", "JavaMethodInvokerFactory", "AbstractResourceMethodDispatchProvider", "ResourceJavaMethodDispatcher", "HttpMethodRule", and "SubLocatorRule".

```
500 Trace -  
  
java.lang.NullPointerException  
at org.apache.ambari.view.capacityscheduler.ConfigurationService.getVersionTag(ConfigurationService.java:231)  
at org.apache.ambari.view.capacityscheduler.ConfigurationService.readConfiguration(ConfigurationService.java:139)  
at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)  
at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:57)  
at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)  
at java.lang.reflect.Method.invoke(Method.java:606)  
at com.sun.jersey.spi.container.JavaMethodInvokerFactory$1.invoke(JavaMethodInvokerFactory.java:68)  
at com.sun.jersey.server.impl.model.method.dispatch.AbstractResourceMethodDispatchProvider$ResponseOutInvoker._dispatch(AbstractResourceMethodDispatchProvider.java:205)  
at com.sun.jersey.server.impl.model.method.dispatch.ResourceJavaMethodDispatcher.dispatch(ResourceJavaMethodDispatcher.java:75)  
at com.sun.jersey.server.impl.uri.rules.HttpMethodRule.accept(HttpMethodRule.java:288)  
at com.sun.jersey.server.impl.uri.rules.SubLocatorRule.accept(SubLocatorRule.java:134)
```

Solution: Check the view configuration and confirm the Ambari Cluster URL and credentials provided are correct for your cluster.

Appendix: Users and Permissions

Set up Ambari and HDP Users

You need to provide user access to both Ambari and the HDP cluster. In typical cluster deployments, users will be setup at the OS level and Ambari to use a common LDAP directory. You can refer to the Ambari documentation for more information about configuring Ambari for LDAP authentication. In lieu of that configuration being available, this section describes how to setup a local OS and Ambari user.

1. As the Ambari Admin, log into the Ambari Administration interface.
2. Browse to **Manage Users + Groups > Users**.
 - a. Click on **Create Local User**.
 - b. Enter information about the user and click **Save**.

[Users](#) / Create Local User



Username

Type

Status

Ambari Admin

Password

3. Refer to the Ambari Administration Guide for more information on Managing Users + Groups.
4. To configure the user with an OS level account that can access the cluster, first add the user to the OS. For example: for a user 'rohit' on RHEL/CentOS:

```
adduser rohit
```
5. Setup an HDFS workspace and permissions for the user:

```
hdfs dfs -mkdir /user/rohit
hdfs dfs -chown rohit:hadoop /user/rohit
hdfs dfs -chmod 755 /user/rohit
```

Manage View Access Permissions

You can control user access to views by setting permissions on a view instance.

1. As the Ambari Admin, log into the Ambari Administration interface.
2. Browse to **Views** and expand the specific view that you want to manage permissions.
3. Add the user (screenshot below):

Permissions		
Permission	Grant permission to these users	Grant permission to these groups
Use	<input type="text" value="rohit"/> <input type="text" value="rohit"/> <input type="button" value="x"/> <input checked="" type="checkbox"/>	<input type="text" value="Add Group"/>

4. Click the **check** to save the change.
5. Refer to the Ambari Administration Guide for more information on Managing Views.