# Hortonworks Data Platform

Apache Ambari Views

(April 3, 2017)

docs.cloudera.com

#### Hortonworks Data Platform: Apache Ambari Views

Copyright © 2012-2017 Hortonworks, Inc. All 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.

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

# **Table of Contents**

1. Understanding Ambari Views	1			
1.1. Views Terminology	1			
1.2. Understanding Views Development, Persona, Versions, and Deployment				
2. Administering Ambari Views	5			
2.1. Preparing Ambari Server for Views	5			
2.2. Running Ambari Server Standalone	7			
2.2.1. Prerequisites For Standalone Ambari Servers	7			
2.2.2. Setting Up Standalone Ambari Server Compared with Setting Up				
Operational Ambari Server	8			
2.2.3. Running Standalone Ambari Server Instances Behind a Reverse				
Proxy				
2.3. Configuring View Instances	9			
2.4. Creating View Instances	10			
2.5. Migrating View Instance Data	11			
2.6. Creating View URLs	. 11			
2.7. Setting View Permissions				
2.8. Configuring Views for Kerberos	. 12			
2.9. Migrating Hue Artifacts to Ambari Views	13			
2.9.1. Requirements for Hue-to-Views Migration	13			
2.9.2. Creating a HueToAmbari View Instance	. 14			
2.9.3. Migrate Hue Artifacts to an Ambari View	18			
2.10. Configuring Specific Views	. 21			
2.10.1. Configuring Capacity Scheduler View	21			
2.10.2. Configuring Files View	29			
2.10.3. Configuring Falcon View	. 33			
2.10.4. Configuring Hive View	39			
2.10.5. Configuring Pig View	49			
2.10.6. Configuring Slider View	57			
2.10.7. Configuring SmartSense View	58			
2.10.8. Configuring Storm View	60			
2.10.9. Configuring Tez View	62			
2.10.10. Configuring Workflow Manager View				
3. Using YARN Queue Manager View	74			
3.1. Setting up Queues	74			
3.2. Configuring Queues	79			
3.3. Enabling Preemption	. 80			
3.4. Setting YARN Queue Priorities	84			
3.5. Configuring Cluster Scheduler Settings				
3.6. Applying the Configuration Changes	87			
4. Using Files View	89			
5. Using Falcon View	. 90			
6. Using Hive View 2.0	. 91			
6.1. Query Tab	. 93			
6.2. Jobs Tab	99			
6.3. Tables Tab				
6.3.1. Creating Tables				
6.3.2. Uploading Tables				
6.4. Saved Queries Tab				

6.5. UDFs Tab	105
6.6. Settings Tab	106
7. Using Pig View	107
7.1. Writing Pig Scripts	107
7.2. Viewing Pig Script Execution History	108
7.3. User-Defined Functions (UDFs) Tab	108
8. Using Slider View	109
9. Using SmartSense View	
10. Using Storm View	111
10.1. Monitoring Storm Cluster Status: the Cluster Summary Page	111
10.2. Monitoring Topology Status: the Topology Summary Page	113
10.3. Looking Up Configuration Values: the Component Summary Page	115
11. Using Tez View	117
11.1. Understanding Directed Acyclic Graphs (DAGs), Vertices, and Tasks	118
11.2. Searching and Identifying Hive Queries	118
11.2.1. Analyzing the Details of Hive Queries	119
11.3. Identifying the Tez DAG for Your Job	122
11.4. Understanding How Your Tez Job Is Executed	
11.5. Identifying Causes of Failed Jobs	
11.6. Viewing All Failed Tasks	126
11.7. Using Counters to Identify the Cause of Slow-Performing Jobs	127
12. Using Workflow Manager View	130

# List of Figures

2.1. Configuring Views with your HDP Cluster	7			
2.2. Default Hive View Settings 4				
2.3. Default Hive View Cluster Configuration	44			
2.4. HDFS Service Page in Ambari	47			
2.5. Using the Filter to Search Advanced hdfs-site Settings	47			
2.6. Granting User Permissions to Hive Views				
2.7. Pig View Details and Settings	52			
2.8. Pig View Cluster Configuration				
2.9. HDFS Service Page in Ambari				
2.10. Using the Filter to Search Advanced hdfs-site Settings	. 55			
2.11. Granting User Permissions to Pig View	56			
2.12. Kerberos Settings for Pig View	57			
2.13. Tez View Create Instance Page	. 64			
2.14. Tez View Instance Page				
2.15. Granting User Permissions to Tez View	. 66			
2.16. Workflow Manager Kerberos Configuration for Oozie	. 70			
2.17. wfm-oozie-proxy-user.png				
6.1. Views Menu of Ambari				
6.2. Links to Hive-Related Views in Ambari	. 92			
6.3. Query Tab : Simple Query in Editor with "default" Database Containing No Tables				
Selected	94			
6.4. Query Editor				
6.5. Database and Table Pane				
6.6. SQL in Query Editor with Resulting Visual Explan Plan				
6.7. Details of a Map Join Node				
6.8. Jobs Tab of Hive View 2.0				
6.9. Table Manager	100			
6.10. Example of Information in the DDL Subtab				
6.11. Example of Storage Information Subtab				
6.12. Example of Detailed Information Subtab				
6.13. Example of Statistics Subtab				
6.14. Example of Creating a Table Form				
6.15. Saved Queries Tab				
6.16. UDF Tab				
6.17. Settings Tab with Example Key and Value for One Property				
	107			
7.2. Pig View Script History Tab	108			
7.3. Pig View UDFs Tab				
11.1. Views Menu of Ambari				
11.2. SQL Query Execution in Hive				
11.3. Hive Queries Tab Showing Unfiltered Results				
11.4. Details for a Successful Query with Links to Application and DAG Windows				
11.5. Total Timeline and Log Details of a Submitted Query				
11.6. Configurations Tab				
11.7. All DAGs View (Truncated Screenshot)				
11.8. Tez View Column Selector Dialog Box				
11.9. View Tab in Tez View				
11.10. DAG Details Window				

11.11. Tez View All Tasks Tab	126
11.12. Tez View DAG-Level Counters Tab	127
11.13. Tez View Vertex Swimlane Tab	127
11.14. Tez View Vertex Details Subtab	128
11.15. Tez View Vertex-Level Counters Tab	128
11.16. Tez View Task-Level Counters Tab	128

# **List of Tables**

2.1. Hive View Instance Details	42
2.2. Troubleshooting Hive Views Errors	48
2.3. Storm View Instance Details	62
2.4. Storm View Instance Settings	62
2.5. Cluster Configurations for Tez View	63
2.6. Cluster Configuration Values for Tez View in Ambari	64
2.7. Kerberos Settings for Tez View	67
2.8. Workflow Manager View Instance Details	72
11.1. Tez Job Status Descriptions	124

# List of Examples

2.1.	Substitute #USER#	 38
2.1.	Substitute #USER#	 38

# **1. Understanding Ambari Views**

Apache Ambari includes the Ambari Views Framework, which enables developers to create UI components, or Views, that "plug into" the Ambari Web interface. Ambari automatically creates and presents to users some instances of Views, if the service used by that View is added to the cluster. For example, if Apache YARN service is added to the cluster, the YARN Queue Manager View displays to Ambari web users. In other cases, the Ambari Admin user must manually create a view instance.

Developing and using Views enables you to extend and customize the Ambari web to meet your specific needs.

Using Views also extends your Ambari implementation to allow third parties to plug in new resource types, along with APIs, providers, and UIs to support them. Views are deployed on the Ambari Server, which enables Ambari Admins to create View instances and set access privileges for users and groups.

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

#### **More Information**

Ambari Views Framework

Views Terminology [1]

Understanding Views Development, Persona, Versions, and Deployment [2]

# **1.1. Views Terminology**

The following are Views terms you should be familiar with:

Views framework	The core framework that is used to develop a View: similar to a Java web application.	
View definition	The View resources and core View properties, such as name, version, and any necessary configuration properties. Ambari reads View definition during deployment.	
View package	A bundle of View client and server assets (and dependencies) that is ready to deploy to Ambari.	
View deployment	The process of instantiating a View instance in Ambari, which makes that View available to Ambari Admins for creating instances.	
View name	The unique identifier for a View. A View can have one or more versions. The name is defined in the View Definition (created by the View Developer) and built into the View Package.	
View version	The uniquely named version of a View. Multiple versions of a View (uniquely identified by View name) can be deployed to Ambari.	

View instance	The 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	The unique identifier of a specific instance of a View.
framework services	View context, instance data, configuration properties, and events

# **1.2. Understanding Views Development, Persona, Versions, and Deployment**

Views are basically web applications that can be "plugged in to" 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, and CSS, provide the UI for the view that is rendered in the Ambari web interface.

#### Development

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 Ambari Views framework is separate from Views themselves. The framework is a core feature of Ambari that you use to create, deploy, integrate, and manage your own, custom views.

You develop and deliver a view by performing the following tasks:

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

#### Persona

Three user persona interact with Views:

View developer Person who builds the front end and back end of a View and uses the framework services available during development. The developer

	creates 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 the Ambari web (basically, the end user).

#### Versions

Each View must have a unique name, although it can have one or more View versions. Each View name and version combination is a single *View package*. After a View package is deployed, Ambari Admins can create *View instances*, each of which is identified by a unique View instance name. The Ambari Admin can then set access permissions for each View instance.

#### Deployment

Views can be deployed and managed in the *operational Ambari Server*, the Ambari Server operating your cluster. Alternatively, Views can be deployed and managed in one or more separate *standalone Ambari Servers*. Running standalone Ambari Server instances is useful when users who will access views will not have (and should not) have access to the operational Ambari Server. You can run one or more separate standalone Ambari Server instances to scale-out your solution for handling a large number of users.

The following table provides details about the Ambari views currently available to you, including two that have Technical Preview status.

View	/iew Automatically Description Created?		HDP Stacks	Required Services
Using YARN Queue Manager View [74]	Yes	Provides a visual way to configure YARN capacity scheduler queue capacity.	HDP 2.3 or later	YARN
Using Files View [89]	Yes	Allows you to browse the HDFS file system.	HDP 2.2 or later	HDFS
Using Hive View 2.0 [91]	Yes	Exposes a way to find, author, execute and debug HDP 2.3 later		HDFS, YARN, Hive
Migrating Hue Artifacts to Ambari Views [13]	No	Supports migrating Hue artifacts to an Ambari View. late		Hue
Using Pig View [107]	No	Provides a way to author and execute Pig Scripts. HDI late		HDFS, Hive, Pig
Using Slider View [109]	No	A tool to help deploy and manage Slider-based applications.       HDP 2.2 or later         This view has been marked deprecated.       HDP 2.2 or later		HDFS, YARN
SmartSense	Yes	Allows you to capture bundles, set bundle capture schedule, and view and download captured bundles.	HDP 2.0 or later	SmartSense

View	Automatically Created?	Description	HDP Stacks	Required Services
topologies.		Supports monitoring Storm cluster status and topologies. This view has been marked deprecated.	HDP 2.5 or later	Storm
Using Tez View [117] Yes View information related to Tez jobs that are executing on the cluster. HDP 2.2.4.2 cluster		HDP 2.2.4.2 or later	HDFS, YARN, Tez	
Workflow Designer	No	This View is <b>Tech Preview</b> HDP 2.4 or later		Oozie

Subsequent chapters in this guide describe tasks performed by an Ambari Administrator to make Views available to users in their Ambari-managed cluster. This guide does not describe View development and packaging. You can learn more about the Ambari Views Framework from the following resources:

#### **More Information**

Running Ambari Server Standalone [7]

https://cwiki.apache.org/confluence/display/AMBARI/Views

https://github.com/apache/ambari/tree/trunk/ambari-views/examples

https://github.com/apache/ambari/tree/trunk/contrib/views

# 2. Administering Ambari Views

An Ambari Administrator should review this chapter and use the instructions in each of the following sections to configure an Ambari-managed cluster for using Views.

- Preparing Ambari Server for Views [5]
- Running Ambari Server Standalone [7]
- Configuring View Instances [9]
- Creating View Instances [10]
- Migrating View Instance Data [11]
- Creating View URLs [11]
- Setting View Permissions [12]
- Configuring Views for Kerberos [12]
- Migrating Hue Artifacts to Ambari Views [13]
- Configuring Specific Views [21]

#### Audience

Read this chapter if you have the Ambari Administrator role. You must have permissions in the Ambari Administrator role to perform tasks described in this chapter.

#### **More Information**

**Role Comparison Chart** 

## **2.1. Preparing Ambari Server for Views**

#### Prerequisites

Before you begin to work with Ambari views:

- Review the amount of memory available to the Ambari server that hosts views for your cluster.
- Review whether your Ambari server is configured for HTTPS.
- Increase Available Memory to the Ambari Views Server

You must increase the amount of memory available to the Ambari server hosting views. This is particularly true if you intend to deploy and use multiple views concurrently.

To increase the memory available to the Ambari Views server:

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 the following value:

-Xmx4096m -XX:PermSize=128m -XX:MaxPermSize=128m

3. Restart the server:

ambari-server restart

#### • Configure a Trust Store

*If your Ambari Server instance is configured for HTTPS*, you must configure a trust store so that the deployed views accept the certificate used by the Ambari Server during API communications.

To configure such a trust store:

#### Steps

1. On the Ambari Server, create a new keystore to contain the server's HTTPS certificate:

```
keytool -import -file <path_to_the_Ambari_Server's_SSL_Certificate> -alias
ambari-server -keystore ambari-server-truststore
```

- 2. When prompted, trust the certificate by typing yes.
- 3. Configure the server to use the new trust store:

```
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.
 [4] Setup truststore.
 [5] Import certificate to truststore.
_____
Enter choice, (1-5): 4
Do you want to configure a truststore [y/n] (y)? y
TrustStore type [jks/jceks/pkcs12] (jks): jks
Path to TrustStore file : <path to the ambari-server-truststore keystore>
Password for TrustStore:
Re-enter password:
Ambari Server 'setup-security' completed successfully.
```

4. Restart the server:

ambari-server restart

• (Optional) Increase Timeout Value

The views.request.read.timeout.millis property in /etc/ambari-server/ conf/ambari.properties sets the timeout value for requests made by Ambari views to non-ambari services, such as webHcat, or Hive. By default, views.request.read.timeout.millis is set to 10 seconds.

The views.ambari.request.read.timeout.millis property in /etc/ambari-server/ conf/ambari.properties sets the timeout values for requests made by Ambari views to Ambari services. By default, views.ambari.request.read.timeout.millis is set to 5 seconds.

Usually no action is required. However, if you experience timeouts, or long wait times, you can increase the values for each of these properties to lower response times.

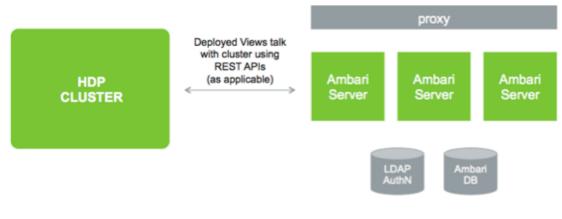
#### **More Information**

Set Up SSL for Ambari

# 2.2. Running Ambari Server Standalone

A recommended strategy that limits user access to your operational Ambari Server while managing a large number of Views users is to set up one or more standalone Ambari Servers. You can configure your operational Ambari Server as a remote cluster, then use the Remote Cluster option when configuring each view instance. A diagram of this strategy follows:

#### Figure 2.1. Configuring Views with your HDP Cluster



#### **More Information**

Registering a Remote Cluster

**Configuring View Instances** 

### 2.2.1. Prerequisites For Standalone Ambari Servers

When setting up multiple standalone Ambari Server instances, you must be aware of the following requirements:

- All Ambari Server instances should be the same version.
- All Ambari Server instances should access the same underlying database.

Ensure that it is *not* the same database that is being used by the Operational Ambari Server that is managing the HDP cluster.

- The Ambari database should be scaled and highly available, independent of Ambari Server.
- For an external authentication source such as LDAP or Active Directory, Ambari Server authentication should be identical for all instances.
- If the cluster that Views users access is Kerberos-enabled, you must configure Ambari and the views for Kerberos.
- You must run each standalone Ambari Server instance behind a reverse proxy.

#### **More Information**

Running Standalone Ambari Server Instances Behind a Reverse Proxy

# **2.2.2. Setting Up Standalone Ambari Server Compared with Setting Up Operational Ambari Server**

Setting up a standalone Ambari Server instance is very similar to setting up an operational Ambari Server instance. Many of the steps are the same, with one key exception: you do not install a cluster using a standalone server instance. A standalone Ambari Server instance does not manage a cluster and does not deploy or communicate with Ambari Agents; instead, a standalone Ambari Server runs as web server instance, serving views for users.

The following table compares the high-level tasks required to set up an operational Ambari Server and a standalone Ambari server:

	Operational Ambari Server	Standalone Ambari Server	
1	Install ambari-server package	Install ambari-server package	
2	Run ambari-server setup (DB, JDK)	Run ambari-server setup (DB, JDK)	
		Do not share the DB with an Operational Ambari Server.	
3	Configure external LDAP authentication	Configure external LDAP authentication	
4	Install cluster	Do not install cluster	
5	Deploy views	Deploy views	
6	Create and configure view instances	Create and configure view instances	
7		(Optional) Repeat for each Ambari Server instance	
8		(Optional) Set up proxy for Ambari Server instances	
9		(Optional) Set up SSL for Ambari	

#### **More Information**

Hortonworks Data Platform Apache Ambari Installation

**Configure external LDAP authentication** 

Set up proxy for Ambari Server instances

Set up SSL for Ambari

### **2.2.3. Running Standalone Ambari Server Instances Behind** a Reverse Proxy

If you require many users to access Ambari views, you should install and run multiple standalone Ambari Server instances behind a reverse proxy. In this case, the reverse proxy must honor *session affinity*, meaning that after a session is established, the reverse proxy routes each subsequent request to the same Ambari server instance. Depending on the reverse proxy implementation, you can achieve session affinity in several different ways, including hashing client IP and using the JSESSIONID header.



#### Important

Using a reverse proxy is supported only for standalone Ambari Server instances.

Using multiple, operational Ambari Sever instances behind a reverse proxy *is not* supported.

# **2.3. Configuring View Instances**

When you create a View instance, you specify some basic configuration information about the view and you configure the view to communicate with a cluster. Based on the resources managed by your Ambari Server, choose one of three options when completing the Cluster Configuration section; Local Cluster, Remote Cluster, or Custom. Use the following descriptions to guide your choice.

- Local Cluster If you are configuring a view instance in an Ambari Server that is also managing a cluster, you can select Local Cluster. When you select this option, Ambari automatically determines the cluster configuration properties required. **Remote Cluster** If your Ambari Server is not managing a cluster, then you must select either Remote Cluster or Custom. If you plan to configure a view to work with a cluster that is remote from an Ambari Server and that cluster is being managed by Ambari, you should select Remote Cluster. Registering a Remote Cluster enables the Remote Cluster option. When you select the Remote Cluster option for a view instance, Ambari automatically determines the cluster configuration properties required for the view instance. Be sure the Remote Cluster includes all services required for the view you are configuring.
- Custom If your cluster is remote from and not being managed by the Ambari Server running the view, you must select **Custom** and then manually configure the view to work with the cluster.

You can use the following table to help determine which options are available for view configuration:

If you are working in this scenario	Choose this option
Your cluster is managed by a local Ambari Server that is also running the view	Local Cluster
Your cluster is managed by Ambari and your cluster	Remote Cluster
is remote from the standalone Ambari Server running the view	
Your cluster is remote from the standalone Ambari Server running the view and your cluster is not managed by Ambari.	Custom

#### **More Information**

**Registering Remote Clusters** 

# **2.4. Creating View Instances**

To create a View instance:

#### Steps

- 1. On the Ambari Admin page, browse to a View and expand it.
- 2. Click Create Instance.
- 3. Provide the following information:

Item	Required?	Description
View version	Yes	Exact version to instantiate
Instance name	Yes	Name unique to the selected View
Display label	Yes	Readable display name of the View instance in Ambari Web
Description	Yes	Readable description of the View instance in Ambari Web
Visible	No	Whether the View is visible or not visible to the end-user in Ambari Web
		Use this property to temporarily hide a view from users.
Settings	Maybe	Depending on the View, a group of settings that can be customized
		If a setting is required, you are prompted to provide the required information.
Cluster configuration	Maybe	Depending on the View, you can choose a local or remote cluster, or manually configure a custom View.

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.

#### **More Information**

Registering Remote Clusters

Configuring View Instances [9]

# **2.5. Migrating View Instance Data**

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 view data from the previous version to the newer version.



#### Important

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

For example, consider a case to migrate from view INSTANCE A (the source view instance) to view INSTANCE B (the target view instance). In this case, you 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
```

The command values are as follows:

- AMBARI\_SERVER\_HOST is the Ambari Server host name 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 is 1.0.0 and TARGET\_VIEW\_VERSION is 1.0.1.

# **2.6. Creating View URLs**

After creating a View instance, you should create a URL by which to access it, based on the view name, version, and instance name. You can also create a short URL of your choosing. You can copy and embed View URLs to provide user access to specific view instances.

To create a View URL:

Steps

- 1. In the Ambari Admin page, browse to the View URLs section.
- 2. Click Create New URL.
- 3. Enter a URL name, select the view, select the instance, and (optionally) type a short URL.

Short URLs must include only lowercase, alphanumeric characters.

4. Click Save.

# **2.7. Setting View Permissions**

By default, a new View instance has no permissions set. An Ambari Admin must specify which users and groups can use the View or, on a local cluster, specify permissions based on cluster roles. An Admin can also set permissions other than those required to use a View.

To set permissions for users and groups to use a view:

#### Steps

- 1. Browse to a view and expand it.
- 2. Click the name of 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.

The Views 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 Admin interface and the Ambari Admin can set users and groups on these permissions.

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.

#### **More Information**

Understanding View Development, Persona, Versions, and Deployment

# **2.8. Configuring Views for Kerberos**

If the cluster that your views communicate with is Kerberos-enabled, you must:

- Configure all Ambari Server instances for Kerberos.
- Configure each view for Kerberos.
- Install the Kerberos client utilities on the Ambari Server so that Ambari can kinit.
  - RHEL/CentOS/Oracle Linux

yum install krb5-workstation

• SLES

zypper install krb5-client

• Ubuntu/Debian

```
apt-get install krb5-user krb5-config
```

• If a view requires HDFS or WebHCat to be configured for a proxy user, you must use the primary Kerberos principal as that user, instead of the ambari-server daemon user.

For example, if you configure Ambari Server for Kerberos principal ambariserver@EXAMPLE.COM, this value would be ambari-server.

Follow specific instructions to configure each view for Kerberos, and the cluster for Kerberos access from the view.

#### **More Information**

Configuring Ambari and Hadoop for Kerberos

Configuring Specific Views

## **2.9. Migrating Hue Artifacts to Ambari Views**

The Ambari Views Framework provides Ambari users a rich, GUI experience to utilize HDP components. Alternative to Ambari Views, HDP users have leveraged the open source web interface Hue to utilize HDP components.

Ambari 2.5 includes the Hue-to-Views Migration tool, which is specifically designed to migrate existing Apache Hue artifacts to Ambari views.

The following sections describe requirements and how to migrate your existing Hue artifacts to Ambari views:

- Requirements for Hue-to-Views Migration [13]
- Creating a HueToAmbari View Instance [14]
- Migrate Hue Artifacts to an Ambari View [18]

### **2.9.1. Requirements for Hue-to-Views Migration**

#### Prerequisites

- Hue service must have a network connection to an Ambari Server serving the Hue-to-Views migration tool
- Install a standalone Ambari Server v2.5. Ambari Server must be enabled as a views server
- Hue database types supported:
  - mysql
  - oracle
  - postgresql

Access rights must be granted to the Hue back-end database.

• In Ambari, create all users in Ambari equivalent to Hue users for each view.

Grant permission for each view user to use their respective views. Each view user must log in to that view, before migrating artifacts.

#### Supported Artifacts, Versions, and Expectations

The Hue-to-Views migration tool supports migrating the following artifacts:

Hive

- Saved Queries
- Query History
- User-defined function (UDF) and JAR artifacts

Pig

- Saved scripts
- Pig Jobs
- User-defined function (UDF) and JAR artifacts

#### Versions

Ambari 2.5.x supports migrating Hue version 2.6.1 artifacts into Ambari 2.5.x

#### Limitations

• The Hue-to-Views Migration tool does not validate scripts.

Scripts maintain the same status in the Ambari Views Framework as they had in Hue.

• The Hue-to-Views migration tool does not support high availability (HA).

You must provide the current, active namenode for the target cluster regarding the Webhdfs URI for Ambari.

- Some hive queries, such as mysql version 0.4.0, may fail to migrate, and cause the Hue-to-Views migration to stop.
- Hue-to-Views migrations must be done on a single cluster. Both Hue and Views must point to the same cluster.

### **2.9.2. Creating a HueToAmbari View Instance**

The Hue-to-Views migration tool instantiates an Ambari view named HUETOAMBARI\_MIGRATION, but does not create a view instance automatically, when you install Ambari 2.5.0.

To create a HueToAmbari view instance that appears in the Ambari Views menu:

#### Steps

1. In Ambari Web, click admin > Manage Ambari.

- 2. On the Ambari Admin page, click Views.
- 3. In View Name, browse to HUETOAMBARI\_MIGRATION then expand it.
- 4. Click Create Instance.

Roles	View Name In:	stances
Go to Dashboard	SCAPACITY-SCHEDULER	1.0.0 (1)
Versions	> FILES	1.0.0 (1)
Remote Clusters	⊁ HIVE	1.0.0 (0) , 1.5.0 (1)
	> HORTONWORKS_SMARTSENS	E 1.3.0.0-22 (1)
III Views	✓ HUETOAMBARI_MIGRATION	1.0.0 (0)
Views		+ Create Instance
View URLs		

5. On **Views/Create Instance**, provide required values for the instance name, display name, and description.

View	HUETOAN	IBARI_MIGRATION
Version	1.0.0	*
Details		
In	stance Name*	HueToAmbari
I	Display Name*	HueToAmbari
	Description*	example Hue-to-Ambari view instance

6. Provide required values for all Hue-to-Views migration settings.

Settings	
Hue Http URL	Enter Hue Server http URL
Hue Server hostname	Enter Hue Server Hostname
Ambari http URL	Enter Ambari Server http URL
Ambari Server hostname	Enter Ambari Server Hostname
Webhdfs URI(Hue)	Enter Webhdfs URI of Hue
Webhdfs URI(Ambari)	Enter Webhdfs URI of Ambari
Hue Database Driver	Enter Hue Database Driver
Hue JDBC URL	Enter Hue JDBC Url
Hue Database Username	Enter Username for Hue DB
Hue Database Password	Enter Password for Hue DB
Ambari Database Driver	Enter Ambari Database Driver
Ambari JDBC URL	Enter Ambari JDBC Url
Ambari Database Username	Enter Ambari DB Username
Ambari Database Password	Enter Ambari DB Password
Kerberos enabled on Ambari cluster?(y/n)	y/n
principal name (if kerberos is enabled)	Please enter the principal name is kerberos is enabled

The following table shows example values for all required Hue-to-Views migration settings, using a standard, Ambari-managed, 3-node cluster.

- Hue Server url : c6401.ambari.apache.org
- Hue NameNode URI: c6402.ambari.apache.org
- NameNode port: 50070
- Hue Database Name(mysql): Huedb
- Hue Database username(mysql): hue
- Hue Database Password(mysql): hue
- Ambari Hostname: c6402.ambari.apache.org
- Ambari Database Name (postgresql): ambari
- Ambari Database username (postgresql): ambari
- Ambari Database Password (postgresql): bigdata

Property	Description	Syntax	Example
Hue Http URL	Https url where Hue server is located	<hue http="" url=""></hue>	http://c6401.ambari.apache.org:8000/
Hue Server hostname	hostname on which Hue runs (the host from which you migrate data)	<hue hostname=""></hue>	c6401.ambari.apache.org
Ambari http URL	http url of the Ambari Server	<ambari http="" url=""></ambari>	http://c6402.ambari.apache.org:8080/
Ambari Server hostname	hostname of the Ambari server	<ambari hostname=""></ambari>	c6402.ambari.apache.org
Webhdfs URI (Hue)	Namenode URI of Hue	webhdfs:// <hostname>:50070</hostname>	webhdfs:// c6402.ambari.apache.org:50070
Webhdfs URI (Ambari)	NameNode URI of Ambari	webhdfs:// <hostname>:50070</hostname>	webhdfs:// c6402.ambari.apache.org:50070
Hue Database Driver	JDBC Driver to access Hue DB	<db driver=""></db>	com.mysql.jdbc.Driver
Hue JDBC URL	JDBC Url to access Hue DB	jdbc: <dbtype>:// <hostname>/<db name&gt;</db </hostname></dbtype>	jdbc:mysql://c6401.ambari.apache.org/ huedb
Hue Database Username	Hue Database Username	<db username=""></db>	hue
Hue Database Password	Hue Database Password	<db password=""></db>	hue
Ambari Database Driver	JDBC Driver to access Ambari DB	<db driver=""></db>	org.postgresql.Driver
Ambari JDBC URL	JDBC Url to access Ambari DB	jdbc: <dbtype>:// <hostname>/<db name&gt;</db </hostname></dbtype>	jdbc:postgresql:// c6402.ambari.apache.org:5432/ambari
Ambari Database Username	database username for Ambari	<db username=""></db>	ambari
Ambari Database Password	database password for the Ambari database	<db password=""></db>	bigdata
Kerberos enabled on Ambari cluster? (y/n)	(y/n) for Kerberos	<y n=""></y>	n
Principal name (if Kerberos enabled)	If Kerberos is enabled, you must provide Principal Name	<principal username&gt;</principal 	ambari-cl1

#### More Information

https://docs.oracle.com/javase/7/docs/api/java/sql/DriverManager.html

7. Click Save.

The new, HUETOAMABARI view displays in the list of Ambari Views. To use the new view, click **Go To Dashboard**.

#### **More Information**

Creating View Instances [10]

Migrate Hue Artifacts to an Ambari View [18]

### **2.9.3. Migrate Hue Artifacts to an Ambari View**

An administrator-level user can use the HueToAmbari view instance to migrate existing Hue 2.6.1 artifacts into Ambari views.

#### Prerequisites

- 1. Review the requirements for Hue-to Views migration and complete all prerequisites.
- 2. Create a view instance, using the Hue-To-Views migration tool.

In the following example, this instance is called HueMigration view.

#### Steps

1. On your standalone views server, using the **Ambari Admin** page, create an administrator-level user.

Grant permission for the admin user to use the HueMigration view.

2. Log in to the HueMigration view as the administrator user.

Logging in initializes the database for the new view.

3. Wait while the view performs database checks.

# Welcome to HueMigration View

### Please wait...

~	Hue Http Test	
→	Hue Webhdfs Test	
~	Hue Database Test	
~	Ambari Database Test	

4. Transfer artifacts.

To transfer artifacts:

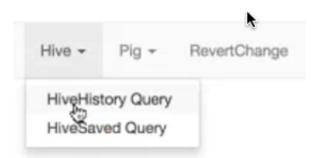
#### Steps

a. In Ambari Web, click Views > HueMigration.

The HueMigration view shows Hive and Pig menus that list existing Hue artifacts.

The RevertChange option allows you to clear any migration settings before you submit them.

b. From each artifact menu, click an artifact name.



A migration settings page displays options for migrating the artifacts.

- c. In the migration settings page, specify:
  - a user name and instance name to which the artifacts will migrate
  - a date range (using Start and End dates) from which to migrate artifacts

Saved Query Migration	
User Name *	hue ×
Instance Name *	Hivehue -
Start Date	2017-02-08
End Date	2017-02-14
Submition	

#### d. Click Submit.

When the migration completes, a migration report displays.

# Migration Report

Parameters	Status
Number of Query Transferred	2
Total Number of Queries	2
Total Time Taken	543ms
Hue Users	hue
Ambari Instance Name(Target)	Hivehue

- e. Review the migration report to confirm artifacts transferred.
- 5. Verify that the artifacts for each user have transferred from Hue to each of the Hive and Pig views.

For each view and user.

#### Steps

- a. Log in to your Views server, as the view user.
- b. Open a view to which you migrated artifacts.
- c. Verify that queries, scripts and UDFs migrated successfully.

#### **More Information**

Requirements for Hue-to-Views Migration [13]

Running Ambari Server Standalone [7]

Creating a HueToAmbari View Instance [14]

Setting View Permissions [12]

## **2.10. Configuring Specific Views**

Ambari configures and deploys most views automatically, for each service added to a cluster.

Depending on your environment, each View (and associated service) may require some additional configuration or troubleshooting.

#### **More Information**

Configuring Capacity Scheduler View [21]

Configuring Files View [29]

Configuring Falcon View [33]

Configuring Hive View [39]

Configuring Pig View [49]

Configuring Slider View [57]

Configuring SmartSense View [58]

Configuring Storm View [60]

Configuring Tez View [62]

Configuring Workflow Manager View [67]

### **2.10.1. Configuring Capacity Scheduler View**

#### Prerequisites

Capacity Scheduler View requires that the cluster is managed by Ambari – the view utilizes the Ambari Server API.

#### **More Information**

Creating a Capacity Scheduler View Instance [22]

Troubleshooting Capacity Scheduler View [29]

#### 2.10.1.1. Creating a Capacity Scheduler View Instance

When you deploy a cluster using Ambari, a Capacity Scheduler View instance is automatically created. If you do not need to reconfigure the Ambari-created cluster, proceed to use the YARN Queue Manager View.

If you have deployed your cluster manually, or if you need to re-configure the Ambaricreated YARN Queue Manager View, you can use the information in this section to create and configure a view instance.

To set up a Capacity Scheduler / YARN Queue Manager view instance:

#### Steps

						Cthe	er Bookmarks
Dashboard	Services	Hosts	Alerts	Admin		💄 admin 👻	
_						About	
						Manage An	nbari
						Settings	
						Sign out	
HDFS Links		Memory U	sage	N	etwork U	Isage	

1. In the **Ambari Web** top menu, click **admin > Manage Ambari**.

2. On the Ambari Admin page, click Views.

🚕 Ambari		III 🔺 admin 🗸
Clusters	Welcome to Areache Archeri	
test_cluster1 ©	Welcome to Apache Ambari	
Permissions	Monitor your cluster resources, manage who can access the clus	ter, and customize views for Ambari users.
Go to Deshboard	Operate Yo	ur Cluster
Versions	Manage the configuration of your cluster	and monitor the health of your services
III Views	_	
Views	Manage Permissions	Go to Dashboard
LUser + Group Management		
Users		
	Manage Users + Groups	Deploy Views
Groups	Manage the users and groups that can access Ambari	Create view instances and grant permissions
	1	
	Users Groups	Views
	Users Groups	ViewS

3. On the **Views** page, click **CAPACITY-SCHEDULER**, then click **Create Instance**.

🚕 Ambari				admin +
Clusters	Views		Search	Q
test_cluster1 © Permissions	View Name	Instances		
Go to Dashboard	❤ CAPACITY-SCHE	DULER 0.4.0 (1)		
Versions		YARN Queue 0.4.0 Manager	Manage YARN Capacity Sche	duler
III Views		+ Create Instance		
Views	> FILES	0.2.0 (0)		
LUser + Group Management	> HIVE	0.4.0 (0)		
Users	> PIG	0.1.0 (0)		
Groups	> SLIDER	2.0.0 (0)		
	> TEZ	0.7.0.2.3.0.0-812 (1)		

4. In the **Details** box on the **Create Instance** page, type an instance name, display name, and a description for the view.



#### Note

The instance name cannot contain spaces or special characters.

- 5. In the **Cluster Configuration** box on the **Create Instance** page, configure the view to communicate with the HDP cluster.
  - For HDP clusters that are local (managed by the local Ambari Server), click the Local Ambari Managed Cluster option, then click the local cluster name.
  - To configure the view to work with HDP clusters that are remote (not part of this Ambari Server instance), click the **Custom** option, then enter the remote Ambari cluster API URL and the Ambari cluster user name and password.
- 6. Click **Save** at the bottom of the page.

🔬 Ambari			🛦 admin 👻			
Ciusters	Views / Create In	istance				
test_cluster1 C Permissions Go to Dashboard	View CAPA Version 0.4.0	CITY-SCHEDULER				
Versions	Details	· · ·				
III Views	Instance Nar	e Capacity_Scheduler_1				
Views	Display Nar	Capacity Scheduler 1				
LUser + Group Management	Descripti	Capacity Scheduler configuration 1				
Users Groups		Visible				
	Cluster Configuration					
	Local Amberi Managed Cluster     Cluster Name test_cluster1					
	Custom	L* http://ambarl.server.8080/api/v1/clusters/MyCluster				
	Operator Usernam	e* djones				
	Operator Passwor	d*	9			
		Cancel	Save			

7. A Capacity Scheduler View instance is created, and the configuration page for the instance appears.

🝌 Ambari				🛓 admin +
Ciusters	Views / Capacity Scl	heduler 1 Go to instance	1	Delete Instance
test_cluster1 © Permissions Go to Dashboard	View CAPACITY-	SCHEDULER		
Versions				
III Views	Details			/ Edit
Views	Instance Name	Capacity_Scheduler_1		
LUser + Group Management	Display Name	Capacity Scheduler 1		
Users Groups	Description	Capacity Scheduler configuration 1		
		✓ Visible		
	Permissions			
	Permission Grant per	mission to these users	Grant permission to these groups	
	Use Add Us			
	Cluster Configuration			∕ Edit
	<ul> <li>Local Ambari Managed Clus</li> <li>Cluster Name</li> </ul>	ter test_cluster1 \$		
	Custom			
		http://ambari.server:8060/api/v1/ch	usters/MyCluster	
	Ambari Cluster URL* Operator Username*	admin		

#### More Information

Using YARN Queue Manager View [74]

### 2.10.1.2. User Permissions for YARN Queue Manager View

To add users and groups to a YARN Queue Manager view instance:

#### Steps

1. On the Capacity Scheduler view instance configuration page, click **Add User** in **Permissions**.

🚕 Ambari				🛔 admin 🕶
Clusters	Views / Capac	city Scheduler 1 Go to instance		Delete Instance
test_cluster1 © Permissions Go to Dashboard Versions		APACITY-SCHEDULER		
III Views	Details			∕ Edt
Views	Instance	Name Capacity_Scheduler_1		
LUser + Group Management	Display	Name Capacity Scheduler 1		
Users Groups	Desc	ription Capacity Scheduler configuration 1		
		Visible		
	Permissions			
	Permission	Grant permission to these users	Grant permission to these groups	
	Use	Add User	Add Group	
	Cluster Configuratio	on		🖌 Edit
	Local Ambari Managed Cluster     Cluster Name     test_cluster1     \$			
	Custom	r URL* http://ambari.server:8080/api/v1/clus	ters/MyCluster	

2. In **Use**, enter user names, then click the blue check mark to add the users. You can use the same method to add groups in **Add Group**.

🝌 Ambari					🛔 admin 👻
Ciusters	Views / Cap	bacity Sc	heduler 1 to instance	De	iete Instance
test_cluster1 C Permissions Go to Dashboard	View	CAPACITY-	SCHEDULER		
Versions					
III Views	Details				✓ Edit
Views	Instar	nce Name	Capacity_Scheduler_1		
LUser + Group Management	Disp	play Name	Capacity Scheduler 1		
Users Groups	De	escription	Capacity Scheduler configuration 1		
uroque			⊘ Visible		
	Permissions				
	Permission Grant permission to these users Grant permission to these				
	Use	bamith	× djones ×	Add Group	
	Cluster Configur	Cluster Configuration			
	Local Ambari Managed Cluster     Cluster Name     test_cluster1     \$				
	Custom				
	Ambari Clu	uster URL*	http://ambari.server:8080/api/v1/clusters	/MyCluster	

3. After you have finished adding users and groups, click **Go to instance** at the top of the page to open the YARN Queue Manager view instance.

🝌 Ambari					🛓 admin 👻
Clusters	Views / Cap		Delete Instance		
test_cluster1 @	View	CAPACITY	SCHEDULER		
Go to Dashboard	Version	0.4.0			
Versions					
III Views	Details				🖌 Edit
Views	Instan	ce Name	Capacity_Scheduler_1		
LUser + Group Management	Displ	ay Name	Capacity Scheduler 1		
Users	De	scription	Capacity Scheduler configuration 1		
Groups			Visible		
	Permissions				
	Permission	Grant per	mission to these users	Grant permission to these groups	
	Use	bsmith	djones	product_management	
	Cluster Configura	ation			/ Edit
					₽ EOR
	<ul> <li>Local Ambari I</li> <li>Clus</li> </ul>	Managed Clu iter Name	test_cluster1 0		
	0.0				
	Custom				
	Ambari Clus	iter URL*	http://ambari.server:8080/api/v1/clus	ters/MyCluster	
	Operator Us	sername*	admin		

4. The Capacity Scheduler view instance page appears.

Ambari test_cluster1	Dops 1 divi		Dashboard	Services	Hosts	Alerts	Admin	 🛓 admin 🕶
+ Add Queue	Actions +	Click on a queue to the left for det	tails.					
root (100%)	× .							
default (100%)	×							
Scheduler	× .							
Maximum 10000 Applications								
Maximum AM 20 9 Resource	4							
Node Locality 40 Delay								
Calculator org.apa	the hadoop yarr							
Martin								
Versions								
vt Current 45 years ag	0 load							

# 2.10.1.3. Troubleshooting Capacity Scheduler View

If you encounter an issue where the configurations cannot be applied from Capacity Scheduler View, you should go to the local Ambari Server instance managing the cluster and directly edit the Capacity Scheduler configuration from the YARN configuration page.

In the local Ambari instance, using **Ambari Web**, browse to **>Services > YARN**, then click the **Configs** tab. On the **Advanced** tab, expand **Scheduler**.

yarn.resourcemanager.	org.apache.hadoop.yarn.server.resourcemanager.scheduler.capacity.CapacityScheduler		0	C
scheduler.class				
Capacity Scheduler	yarn.scheduler.capacity.maximum-am-resource-percent=0.2			
	yarn.scheduler.capacity.maximum-applications=10000			
	yarn.scheduler.capacity.node-locality-delay=40			
	yarn.scheduler.capacity.queue-mappings-override.enable=false			
	yarn.scheduler.capacity.root.accessible-node-labels=*	•	C	
	yarn.scheduler.capacity.root.acl_administer_queue=*	×	0	
	yarn.scheduler.capacity.root.capacity=100			
	yarn.scheduler.capacity.root.default.acl_submit_applications=*			
	yarn.scheduler.capacity.root.default.capacity=100			
	yarn.scheduler.capacity.root.default.maximum-capacity=100			

In **Scheduler** you can edit the underlying configurations for the YARN Queue Manager and fix any issues you may encounter.

# 2.10.2. Configuring Files View

This section describes how to configure a Files View instance and use the File browser UI to access HDFS.

- Configuring Your Cluster for Files View [30]
- Creating and Configuring a Files View Instance [31]
- Troubleshooting [33]



# Important

It is critical that you prepare your Ambari Server for hosting views. It is strongly recommended you increase the amount of memory available to your Ambari Server, and that you run additional, standalone Ambari Servers to host the views. See and for more information.

### **More Information**

Preparing Ambari Server for Views [5]

Running Ambari Server Standalone [7]

# 2.10.2.1. Configuring Your Cluster for Files View

For Files View to access HDFS, the Ambari Server daemon hosting the view needs to act as the proxy user for HDFS. This allows Ambari to submit requests to HDFS on behalf of Files View users.

If you are running views in an operational Ambari server (one that is operating the cluster) Ambari does this setup by default. You should verify that the setup described in the following subsections has been completed. If you are running views on a standalone server, you must setup proxy user settings manually.

To set up an HDFS proxy user for the Ambari Server daemon account, you need to configure the proxy user in the HDFS configuration. This configuration is determined by the account name the **ambari-server** daemon is running as. For example, if your ambari-server is running as **root**, you set up an HDFS proxy user for **root** with the following:

#### Steps

- 1. In Ambari Web, browse to Services > HDFS > Configs.
- 2. On the Advanced tab, browse to the Custom core-site section.
- 3. Click Add Property...
- 4. Enter the following custom properties:

hadoop.proxyuser.root.groups="us**#is**"tice the ambari-server daemon account name hadoop.proxyuser.root.hosts=ambariot is part of the property name. Be sure to modify server.hostname this property name for the account name you are running the ambari-server as. For example, if you were running ambari-server daemon under the account name ambariusr, you would use the following properties instead:

hadoop.proxyuser.ambariusr.group**Sirf'ulsærty**, if you have configured Ambari Server for hadoop.proxyuser.ambariusr.hosts **Kærlbæri**s, be sure to modify this property name for server.hostname the primary Kerberos principal user. For example, if ambari-server is setup for Kerberos using principal ambari-server@EXAMPLE.COM, you would use

the following properties instead:

hadoop.proxyuser.ambariserver.groups="users" hadoop.proxyuser.ambariserver.hosts=ambariserver.hostname

5. Save the configuration change and restart the required components as prompted by Ambari.

# **More Information**

Configure Ambari Server for Kerberos

# 2.10.2.2. Creating and Configuring a Files View Instance

To create and configure a Files View instance:

#### Steps

- 1. Browse to the Ambari Admin page.
- 2. Click Views, expand the Files View, and click Create Instance.
- 3. In **Details**, enter the following values for View instance properties:

Property	Description	Value
Instance Name	This is the Files view instance name. This value should be unique for all Files view instances you create. This value cannot contain spaces and is required.	FILES_1
Display Name	This is the name of the view link displayed to the user in Ambari Web.	MyFiles
Description	This is the description of the view displayed to the user in Ambari Web.	Browse HDFS files and directories.
Visible	This checkbox determines whether the view is displayed to users in Ambari Web.	Visible or Not Visible

- 4. Information that you provide in **Settings and Cluster Configuration** depends on your environment; specifically, whether:
  - your cluster is Kerberos-enabled or not
  - NameNode HA is enabled or not
  - your Files View instance is being configured in an operational or a standalone Ambari server

Refer to the following table for instructions to complete the Files View configuration:

Kerberos Enabled	NameNode HA Enabled	Operational Ambari Server	Standalone Ambari Server
No	No	Settings: defaults	Settings: defaults
No	Yes	Cluster Configuration: Local	Cluster Configuration: Custom
Yes	No	Settings : Kerberos	
		Cluster Configuration : Custom	
Yes	Yes	Settings: Kerberos	
		Cluster Configuration: Custom	

The Local Ambari Managed Cluster Configuration option is enabled in the Ambari Admin page *only* if you are managing a cluster in an Operational Ambari Server.

### **More Information**

Cluster Configuration: Local [32]

Cluster Configuration: Custom [32]

# Kerberos Settings [32]

Running Ambari Server Standalone [7]

# 2.10.2.2.1. Kerberos Settings

### Prerequisites

Before setting up Kerberos for Files View, you must first set up Kerberos for Ambari by configuring the Ambari Server daemon with a Kerberos principal and keytab.

#### Steps

After you have set up Kerberos for Ambari, in **Files View > Settings**, enter the following properties:

Property	Description	Example Value
WebHDFS Username	This is the username the view will access HDFS as. Leave this default value intact to represent the authenticated view user.	\${username}
WebHDFS Authorization	This is the semicolon-separated authentication configuration for WebHDFS access.	auth=KERBEROS;proxyuser=ambari-server **This property is only needed if the view is Custom Configured or Ambari Server is Kerberized before 2.4.0.

With a Kerberos setup, the proxy user setting should be the primary value of the Kerberos principal for Ambari Server. For example, if you configured Ambari Server for Kerberos principal **ambari-server@EXAMPLE.COM**, this value would be **ambari-server**.

#### **More Information**

Configuring Views for Kerberos

# 2.10.2.2.2. Cluster Configuration: Local

The Local Ambari Managed Cluster Configuration option is enabled on the Ambari Admin page if you are managing a cluster with Ambari. When enabled, you can choose this option and Ambari will automatically configure the view based on how the cluster is configured.

When you configure the view using the Local option, the Files View communicates with HDFS based on the fs.defaultFS property (for example: hdfs://namenode:8020). The View also determines whether NameNode HA is configured and adjusts accordingly.

# 2.10.2.2.3. Cluster Configuration: Custom

These properties are required if using Custom configuration.

<b>Required Properties</b>	Description	Example Value
WebHDFS FileSystem URI	The WebHDFS FileSystem URI in the format webhdfs:// <host>:<http_port></http_port></host>	webhdfs://namenode:50070

These properties are required if your cluster is configured for NameNode HA.

Property	Description	Example Value
Logical name of the NameNode cluster	Comma-separated list of nameservices.	hdfs-site/dfs.nameservices

Property	Description	Example Value
		For example:
		nameservice
List of NameNodes	Comma-separated list of NameNodes for	hdfs-site/dfs.ha.namenodes
	a given nameservice.	For example:
		namenode1,namenode2
First NameNode RPC Address	RPC address for first name node.	hdfs-site/dfs.namenode.rpc-address. [nameservice].[namenode1]
Second NameNode RPC Address	RPC address for second NameNode.	hdfs-site/dfs.namenode.rpc-address. [nameservice].[namenode2]
First NameNode HTTP (WebHDFS) Address	WebHDFS address for first NameNode.	hdfs-site/dfs.namenode.http-address. [nameservice].[namenode1]
Second NameNode HTTP (WebHDFS) Address	WebHDFS address for second NameNode.	hdfs-site/dfs.namenode.http-address. [nameservice].[namenode2]
Failover Proxy Provider	The Java class that HDFS clients use to contact the Active NameNode.	hdfs-site/ dfs.client.failover.proxy.provider. [nameservice]

# 2.10.2.2.4. Troubleshooting

Error	Solution
<b>500</b> Usernames not matched: name=root != expected=ambari- server	If your cluster is configured for Kerberos, double-check WebHDFS Authorization setting and confirm the "proxyuser=" part of the string is set to the Ambari Server principal name. For example:
	auth=KERBEROS;proxyuser=ambari-server
500 User: ambari-server is not allowed to impersonate admin	HDFS has not been configured for Ambari as a proxy user.
<b>500</b> SIMPLE authentication is not enabled. Available:[TOKEN, KERBEROS]	If your cluster is configured for Kerberos, you cannot use the Local Cluster Configuration option. You must use the Custom Cluster Configuration option and enter the WebHDFS FileSystem URI.
	For example:
	webhdfs://namenode:50070

### **More Information**

Kerberos Settings [32]

**Configuring Your Cluster** 

Cluster Configuration: Custom [32]

# 2.10.3. Configuring Falcon View

Hadoop administrators can use **Falcon View** to centrally define, schedule, and monitor data management policies. **Falcon** uses those definitions to auto-generate workflows in Apache Oozie.

This section describes:

• Configuring Your Cluster for Falcon View [34]

• Installing and Configuring Falcon View [35]

# 2.10.3.1. Configuring Your Cluster for Falcon View

For Falcon View to access HDFS, the Ambari Server daemon hosting the view must act as the proxy user for HDFS. This allows Ambari to submit requests to HDFS on behalf of the Falcon View users. This is critical since Falcon View stores metadata about the user Falcon entity definitions. Falcon View users must have a user directory set up in HDFS.

# 2.10.3.1.1. Set up HDFS Proxy User

To set up an HDFS proxy user for the Ambari Server daemon account, you need to configure the proxy user in the HDFS configuration. This configuration is determined by the account name the **ambari-server** daemon is running as. For example, if your ambari-server is running as **root**, you set up an HDFS proxy user for **root**.

#### Steps

- 1. In Ambari Web, browse to Services > HDFS > Configs.
- 2. Under the Advanced tab, navigate to the Custom core-site section.
- 3. Click Add Property... to add the following custom properties:

hadoop.proxyuser.root.groups="users" hadoop.proxyuser.root.hosts=ambari-server.hostname

Notice the ambari-server daemon account name root is part of the property name. Be sure to modify this property name for the account name you are running the ambariserver as. For example, if you were running ambari-server daemon under an account name of ambariusr, you would use the following properties instead:

```
hadoop.proxyuser.ambariusr.groups="users"
hadoop.proxyuser.ambariusr.hosts=ambari-server.hostname
```

Similarly, if you have configured Ambari Server for Kerberos, be sure to modify this property name for the **primary Kerberos principal** user. For example, if ambari-server is setup for Kerberos using principal **ambari-server@EXAMPLE.COM**, you would use the following properties instead:

```
hadoop.proxyuser.ambari-server.groups="users"
hadoop.proxyuser.ambari-server.hosts=ambari-server.hostname
```

4. Save the configuration change and restart the required components as indicated by Ambari.

# **More Information**

Ambari Server for Kerberos

# 2.10.3.1.2. Set up HDFS User Directory

Falcon View stores user metadata in HDFS. By default, the location in HDFS for this metadata is /user/\${username} where \${username} is the username of the currently logged in user that is accessing Falcon View.



# Important

Since many users leverage the default Ambari admin user for getting started with Ambari, the /user/admin folder needs to be created in HDFS. Therefore, be sure to create the admin user directory in HDFS using these instructions prior to using the view.

To create user directories in HDFS, do the following for each Falcon View user:

### Steps

- 1. Connect to a host in the cluster that includes the HDFS client.
- 2. Switch to the hdfs system account user.

su - hdfs

3. Using the HDFS client, make an HDFS directory for the user. For example, if your username is admin, you would create the following directory.

hadoop fs -mkdir /user/admin

4. Set the ownership on the newly created directory. For example, if your username is admin, you would make that user the directory owner.

```
hadoop fs -chown admin:hadoop /user/admin
```

# **2.10.3.2. Installing and Configuring Falcon View**

You must manually copy the .jar file for Falcon View, then configure Ambari to access the view. You can install Falcon View in a secure or an unsecure cluster. If using a secure cluster, Ambari and Falcon must be properly configured with Kerberos.

# Prerequisites

• Apache Falcon must have been installed and configured, and be deployed in Ambari.

For an Ambari-managed installation, Falcon is included as a default service.

- The users and groups for Falcon must exist in Ambari prior to installing the Falcon View.
- Falcon must have been configured as a proxy super user in the <code>oozie-site</code> properties and in the HDFS <code>core-site</code> properties.

#### Steps

- 1. Copy the Falcon View falcon-ambari-view.jar file from the Falcon server / webapp directory to the Ambari server /views directory.
  - If the Falcon and Ambari servers are on the same host, use the copy command:

```
cp /usr/hdp/current/falcon-server/server/webapp/falcon-ambari-view.jar /
var/lib/ambari-server/resources/views/
```

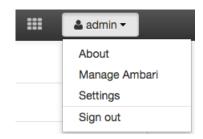
• If the Falcon server is on a remote host, use the secure copy command for your operating system.

A key pair might be required. See your operating system documentation for more information about remote copies.

2. Restart the Ambari server.

[root@DataMovementDocs-1 ~]# ambari-server restart

3. In Ambari Web, browse to *user\_name* > Manage Ambari.



- 4. Under Deploy Views, click Views, then click Falcon > Create Instance in the Views list.
- 5. Provide the required Details information.

Instance Name: 250 characters, no spaces, no special characters Display Name: 250 characters, including spaces; no special characters; can be the same as the Instance Name

Description: 140 characters max, including spaces; special characters allowed



# Note

If you enter more than the allowed number of characters, you might see the error message Cannot create instance: Server Error.

6. Select a cluster configuration.

The Local and Remote fields populate with the names of available clusters. The authentication type for the cluster is automatically recognized.

To use a custom cluster location, enter the Falcon service URI and authentication type of simple or kerberos.

7. Click Save.

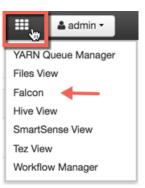
The Permissions section displays at the bottom of the Views page.

- 8. (Optional) Set the permissions for access to the view.
- 9. Hover over the Views icon to verify that your Falcon View is available in the menu.



# Note

Do *not* click on the Falcon link yet. You must make additional configuration changes before you can access the Falcon View.



10.Click the Ambari icon to return to the Dashboard window, then click the **Falcon** service and the **Configs** tab.

0	HDFS	ø	Summary	Configs	Quick Links-	Service Actions
۲	YARN	C				
ø	MapReduce2	0	Summary			No alert
	Tez			Falcon Serve	Started No alerts	
۲	Hive			Falcon Client	2 Falcon Clients Installed	
۲	HBase	C				
	Pig					
	Sqoop					
•	Oozie					
0	ZooKeener					
0	Falcon					
U	Flume					
0	Ambari Infra					

11Scroll to the Falcon startup.properties section, locate the \*.application.services field, and enter the following services immediately above the line org.apache.falcon.metadata.MetadataMappingService:

org.apache.falcon.service.GroupsService,\

org.apache.falcon.service.ProxyUserService,

12Add the proxy user for hosts and groups in the **Custom falcon-runtime.properties** section.

The proxy user is the user that the Falcon process runs as, typically Falcon.

- a. Click Add Property.
- b. Add the following key/value pairs.

Substitute #USER# with the proxy user configured for the Ambari server.

• Key=\*.falcon.service.ProxyUserService.proxyuser.#USER#.hosts, Value=\*

These are the hosts from which *#USER#* can impersonate other users.

Key=\*.falcon.service.ProxyUserService.proxyuser.#USER#.groups, Value=\*

These are the groups that the users being impersonated must belong to.

## Example 2.1. Substitute #USER#

```
In the key/value pairs above, if the #USER# is "falcon", enter
*.falcon.service.ProxyUserService.proxyuser.falcon.hosts.
```

The wildcard value=\* (asterisk) is used to allow impersonation from any host or of any user. If you don't use the wildcard character, enter the appropriate host or group values.

Add Property			
Туре	falcon-runtime.properties.xml		
Кеу	*.falcon.service.ProxyUserService.proxyuser.falcon.groups		
Value	*		
	4		
	Cancel		

13.Click Save on the information bar at the top of the Configs page.

<>	V18         admin         V17         admin         V16         admin         V15         admin         V14         admin         V13         a           8 hours ago HDP-2.5         8 hours ago HDP-2.5         9 hou						
X •     V18 •     admin authored on Fri, Aug 12, 2016 13:19     Discard     Save							
- Fak	▼ Falcon Server						

If you try to leave the page without clicking Save, you see a Warning message. Click **Save** in the Warning dialog box.

A Restart Required message displays at the top of the Falcon Configs page.

14.Click Restart > Restart All Affected to restart the Falcon services.

15.When the restart completes, verify that you can access the Falcon View by clicking **Falcon** in the **Views** menu.

### **More Information**

Adding a Service to your Hadoop cluster

Installing Apache Falcon

Managing Users and Groups

# 2.10.4. Configuring Hive View

This section describes:

- Upgrading Your Hive View [39]
- Configuring Your Cluster for Hive View [40]
- Creating a Hive View Instance [42]
- Troubleshooting Hive View [48]

### Prerequisites

• Prepare your Ambari Server for hosting views.

It is strongly recommended that you:

- increase the amount of memory available to your Ambari Server
- run a standalone Ambari Server to host the views
- Install Tez View when you install Hive View. Tez View integrates with Hive View.

#### **Hive View Versions**

With the release of Apache Ambari 2.5.0, two Hive View versions install as part of your Hortonworks Data Platform distribution:

#### Hive View 1.5

#### Hive View 2.0

Both versions are JDBC-based. You can run both views simultaneously, use only one of the views, or upgrade your data from the older view to the newer view.



# Important

In the Ambari UI and in this documentation going forward, *Hive View* refers to version 1.5 of the view and *Hive View 2.0* is the term to differentiate the new version of the view from the previous version.

#### **More Information**

Preparing Ambari Server for Views [5]

Running Ambari Server Standalone [7]

Using Tez View [117]

# 2.10.4.1. Upgrading Your Hive View

If you are upgrading from Apache Ambari 2.4.0 to Apache Ambari 2.5.0 and want to upgrade and migrate the data and queries from Hive View to Hive View 2.0, first create a new instance of the Hive View and then migrate your queries.

#### Migrating your queries into the new view

Create a new Hive View 2.0 instance and then migrate the saved queries from the Hive View 1.5 instance to the new instance. To do that, run the following curl command.

```
curl -v -u admin:admin -X PUT -H X-Requested-By:1 http://<host/ip ambari
server>:8080/api/v1/views/<view name>/versions/<version of target view>/
instances/<instance name of target view>/migrate/<version of source
view>/<instance name of source view>
```

For information on where to get the specific parameters listed in the curl command, refer to the following figure:

Hive View 2.0

Visible

View	HIVE	
Version	2.0.0	
Details		
Details		
Inst	tance Name	AUTO_HIVE20_INSTANCE

/main/view/HIVE/auto\_hive20\_instance C

This view instance is auto created when the Hive service is added to a cluster.

# **2.10.4.2. Configuring Your Cluster for Hive View**

**Display Name** 

Description

Short URL

For Hive View to access HDFS, the Ambari Server daemon hosting the view needs to act as the proxy user for HDFS. This allows Ambari to submit requests to HDFS on behalf of Hive View users. This is critical since the Hive View will store metadata about their user Hive queries in HDFS. This also means users that access Hive View must have a user directory setup in HDFS.

If you are running views in an operational Ambari server (one that is operating the cluster) Ambari does this setup by default. You should verify that the HDFS proxy user and user directory settings are correct. If you are running views on a standalone server, you must set up proxy user settings manually.



# Important

*For clusters with wire encryption enabled*: You must configure a Hive session parameter to use Hive View:

- 1. Click the drop-down menu of your username profile (top right corner of window).
- 2. Click Manage Ambari > Views > Hive > Hive View or Hive View 2.0.
- 3. Click the **Edit** icon in the Settings part of the window.
- 4. In the the Hive Session Parameters field, enter sslTrustStore=/etc/security/serverKeys/ hivetruststore.jks;trustStorePassword=your\_password, inserting a real password in place of your\_password. If you do not specify a password, the default password that gets assigned is changeit.
- 5. Click Save.

### **More Information**

Set up HDFS Proxy User [41]

Set up HDFS User Directory [42]

# 2.10.4.2.1. Set up HDFS Proxy User

To set up an HDFS proxy user for the Ambari Server daemon account, you must configure the proxy user in the HDFS configuration. This configuration is determined by the account name the **ambari-server** daemon is running as. For example, if your ambari-server is running as **root**, you set up an HDFS proxy user for **root** with the following:

#### Steps:

- 1. In Ambari Web, browse to **Services > HDFS > Configs**.
- 2. Under the Advanced tab, browse to the Custom core-site section.
- 3. Click Add Property... to add the following custom properties:

```
hadoop.proxyuser.root.groups="users"
hadoop.proxyuser.root.hosts=ambari-server.hostname
```

Notice the **ambari-server** daemon account name root is part of the property name. Be sure to modify this property name for the account name you are running the ambariserver as. For example, if you were running **ambari-server** daemon under an account name of **ambariusr**, you would use the following properties instead:

```
hadoop.proxyuser.ambariusr.groups="users"
hadoop.proxyuser.ambariusr.hosts=ambari-server.hostname
```

Similarly, if you have configured Ambari Server for Kerberos, be sure to modify this property name for the **primary Kerberos principal** user. For example, if ambari-server is setup for Kerberos using principal **ambari-server@EXAMPLE.COM**, you would use the following properties instead:

```
hadoop.proxyuser.ambari-server.groups="users"
hadoop.proxyuser.ambari-server.hosts=ambari-server.hostname
```

4. Save the configuration change and restart the required components as indicated by Ambari.

### **More Information**

Ambari Server for Kerberos

# 2.10.4.2.2. Set up HDFS User Directory

The Hive View stores user metadata in HDFS. By default, the location in HDFS for this metadata is /user/\${username} where \${username} is the username of the currently logged in user that is accessing the Hive View.



# Important

Since many users leverage the default Ambari admin user for getting started with Ambari, you must create the /user/admin folder in HDFS using these instructions before using Hive view.

To create user directories in HDFS, for each Hive View user:

#### Steps

- 1. Connect to a host in the cluster that includes the HDFS client.
- 2. Switch to the hdfs system account user.

su - hdfs

3. Using the HDFS client, make an HDFS directory for the user. For example, if your username is admin, you would create the following directory.

hadoop fs -mkdir /user/admin

4. Set the ownership on the newly created directory. For example, if your username is admin, you would make that user the directory owner.

```
hadoop fs -chown admin:hadoop /user/admin
```

# 2.10.4.3. Creating a Hive View Instance

#### Steps

- 1. Click Manage Ambari to open the Ambari Admin user interface.
- 2. Click Views > Hive > Create Instance.
- 3. On the Create Instance page, select the **Version**. If multiple Hive View JAR files are present, choose one.
- 4. Enter the following view instance details:

# Table 2.1. Hive View Instance Details

Property	Description	Example Value
Instance Name	This is the Hive view instance name. This value should be unique for all	AUTO_HIVE_INSTANCE

Property	Description	Example Value
	Hive view instances you create. This value cannot contain spaces and is required.	
Display Name	This is the name of the view link displayed to the user in Ambari Web.	Hive View
Description	This is the description of the view displayed to the user in Ambari Web.	Auto-created when the Hive service is deployed.
Short URL	Alternative to full URL to quickly navigate to the Hive View.	Auto-created when the Hive service is deployed.
Visible	This checkbox determines whether the view is displayed to users in Ambari Web.	Visible or Not Visible

- 5. The **Settings** and **Cluster Configuration** options depend on a few cluster and deployment factors in your environment. Typically, you can accept the default **Settings** unless you are using the Hive View with a Kerberos-enabled cluster.
- 6. Click Save.

# **More Information**

Settings and Cluster Configuration [43]

# 2.10.4.3.1. Settings and Cluster Configuration

Ambari configures Hive View settings automatically when you choose to add the Hive service.



# Tip

To use Hive View with a Hive LLAP (interactive query) environment, set the **Use Interactive Mode** property setting to true.

# **Figure 2.2. Default Hive View Settings**

Settings	🖍 Edit	
Hive Session Parameters	transportMode=http;httpPath=cliservice	
WebHDFS Username	\${username}	
WebHDFS Authentication	auth=SIMPLE	
Instance name of Tez view		
Scripts HDFS Directory*	/user/\${username}/hive/scripts	
Jobs HDFS Directory*	/user/\${username}/hive/jobs	
Default script settings file*	/user/\${username}/.\${instanceName}.defaultSettings	

An example default Hive View cluster configuration is shown in the following figure:

# Figure 2.3. Default Hive View Cluster Configuration

Cluster Configuration		🖍 Edit
Local Cluster		
Cluster Name	mycluster \$	
Remote Cluster		
Cluster Name	\$	
Custom		
HiveServer2 JDBC Url*	jdbc:hive2://127.0.0.1:10000	
Hive Metastore directory	/apps/hive/warehouse	
WebHDFS FileSystem URI*	webhdfs://namenode:50070	
Logical name of the NameNode cluster		
List of NameNodes		
First NameNode RPC Address		
Second NameNode RPC Address		
First NameNode HTTP (WebHDFS) Address		
Second NameNode HTTP (WebHDFS) Address		
First NameNode HTTPS (WebHDFS) Address		
Second NameNode HTTPS (WebHDFS) Address		
Failover Proxy Provider		
Umask	022	
Auth To Local		
YARN Application Timeline Server URL*	http://yarn.ats.address:8188	
YARN ResourceManager URL*	http://yarn.resourcemanager.address:8088	

If required for migrating view instances, find and modify the following cluster configuration settings, using Ambari Web. HiveServer2 JDBC URL Click Hive > Summary to view the URL, displayed at the bottom of the Summary list. For example: jdbc:hive2:// c6403.ambari.apache.org:2181,c6401.ambari.apache.org:2181,c6402.an Hive Metastore directory Click Hive > Configs > Advanced > General. For example, /apps/hive/warehouse WebHDFS FileSystem URI\* Click HDFS > Configs > Advanced >Advanced hdfs-site For example dfs.nameserviceid.http-address For HA: Click HDFS > Configs > Advanced > Advanced hdfs-site > dfs.nameservice.id. When you enter the value in the view definition, pre-pend "webhdfs://" to the value you find in the advanced HDFS configuration settings. For example, webhdfs:// c6401.ambari.apache.org:50070 or webhdfs:// nameserviceid Logical Name of the NameNode cluster List of NameNodes First NameNode RPC Address Click HDFS > Configs > Advanced > Advanced hdfs-site > dfs.namenode.rpc-address. See the first address in the list. For example, c6401.ambari.apache.org Second NameNode RPC Address Click HDFS > Configs > Advanced > Advanced hdfs-site > dfs.namenode.rpc-address. See the second address in the list. For example, c6402.ambari.apache.org First NameNode HTTP Click HDFS > Configs > Advanced > Advanced hdfs-site > dfs.namenode.http-address (WebHDFS) Address See the first address in the list. For example, c6401.ambari.apache.org Second NameNode HTTP Click HDFS > Configs > Advanced > Advanced hdfs-site (WebHDFS) Address > dfs.namenode.http-address See the second address in the list. For example,

c6402.ambari.apache.org

First NameNode HTTPS (WebHDFS) Address	Click HDFS > Configs > Advanced > Advanced hdfs-site > dfs.namenode.https-address
	See the first address in the list. For example, c6401.ambari.apache.org
Second NameNode HTTPS (WebHDFS) Address	Click HDFS > Configs > Advanced > Advanced hdfs-site > dfs.namenode.https-address
	See the second address in the list. For example, c6402.ambari.apache.org
Failover Proxy Provider	Click HDFS > Configs > Advanced > Advanced hdfs-site > dfs.client.failover.proxy.provider[nameservice]
Umask	Click HDFS > Configs > Advanced > Advanced hdfs-site > fs.permissions.umask-mode
	The default value is 022. Do not change this value unless you are sure that you understand the effects of changing the value on your Hive View cluster. The umask property defines the file mode creation mask, which controls how file permissions are configured in new files.
Auth To Local	Click HDFS > Configs > Advanced > Advanced core-site > hadoop.security.auth_to_local
YARN Application Timeline Server URL*	Click YARN > Configs > Advanced > Application Timeline Server > yarn.timeline- service.webapp.address. When you enter the value in the view definition, pre-pend "http://" to the value you find in the YARN advanced configuration settings. For example, http://c6401.ambari.apache.org:8188
YARN ResourceManager URL*	Click YARN > Configs > Advanced > Advanced yarn- site > yarn.resourcemanager.webapp.address. When you enter the value in the view definition, pre-pend "http://" to the value you find in the YARN advanced configuration settings. For example, http:// c6401.ambari.apache.org:8088
For NameNode High Availability	

The following values must be entered for primary and secondary NameNodes:

First NameNode RPC Address, or Second NameNode RPC Address	Select the primary or secondary NameNode to view settings from that host in the cluster. When you enter the value in the view definition, pre-pend "http://" to the value you find in the <b>advanced hdfs-site</b> settings. For example, http://c6401.ambari.apache.org:8020
First NameNode HTTP (WebHDFS) Address, or Second	Click HDFS > Configs > Advanced > Advanced hdfs- site > dfs.namenode.http-address. When you enter the value in the view definition, pre-pend "http://" to the

NameNode HTTP (WebHDFS)	value you find in the advanced hdfs-site settings. For
Address	example, http://c6401.ambari.apache.org:50070

To get First NameNode RPC Address values:

#### Steps

1. In Ambari Web, browse to HDFS > Summary page. Click NameNode (primary) or SNameNode (secondary) to view the host page:

# Figure 2.4. HDFS Service Page in Ambari

0	HDFS	Summary He	atmaps	Configs	Quick Links -	Service Actions
۲	YARN					
۲	MapReduce2	Summary		_		No aler
	Tez	1	NameNode	Started No alerts	Disk Remaining	1.1 TB / 1.3 TB (91.10%)
0	Hive	<u>S</u>	NameNode	Started No alerts	Blocks (total)	738
	HBase		odes Status	ava starteu 3 live / 0 dead / 0 decommission		0 corrupt replica / 0 missing / 0 under replicated
	Pig	Jou	urnalNodes	0/0 JournalNodes Live	Total Files + Directories	947
Ô	Oozie	NE	SGateways	0/0 Started	Upgrade Status	No pending upgrade
0	ZooKeeper	NameNo	ode Uptime	6.12 days	Safe Mode Status	Not in safe mode
0	Storm	Namel	Node Heap	102.9 MB / 1011.3 MB (10.2% u	used)	
•	Ambari Infra	Disk Usage (	(DFS Used)	5.1 GB / 1.3 TB (0.40%)		
_	Ambari Metrics	Disk Usage (Non	DFS Used)	109.1 GB / 1.3 TB (8.50%)		

- 2. On the host page, click **Configs > Advanced**.
- 3. Enter rpc in the Filter search well at the top right corner of the page or browse to the **Advanced hdfs-site** settings to find the dfs.namenode.rpc-address value that you can enter into the Hive View definition. Here is an example of using the Filter to locate a value:

# Figure 2.5. Using the Filter to Search Advanced hdfs-site Settings

Summary Heatma	ps Configs	Quick Links*		Service Actions -	
Group Default (3)	Manage Config Gro	rpc	0 •		
< > <sup>1</sup> 7 day	admin s ago HDP-2.5				
🗶 - 🔽 🖌 adm	in authored on Mon, Oct	03, 2016 09:50		Discard Save	
Settings Advanced					
<ul> <li>Advanced hdfs-si</li> </ul>	te				
dfs.namenode.rpc- address	c6401.ambari.apa	che.org:8020		dfs.namenode.rpc-address RPC address that handles all	clients requests.

# **More Information**

Kerberos Setup for Hive View [48]

### 2.10.4.3.2. User Permissions for Hive Views

After saving the Hive View instance definition, grant permission on the view for the set of users who can use the view:

# Figure 2.6. Granting User Permissions to Hive Views

Permissions		
Permission	Grant permission to these users	Grant permission to these groups
Use	admin	Add Group
Local Cluster Po	ermissions	
	<ul> <li>Cluster Administrator</li> </ul>	
	<ul> <li>Cluster Operator</li> </ul>	
	Service Operator	
	<ul> <li>Service Administrator</li> </ul>	
	Cluster User Check All   Clear All	

# 2.10.4.3.3. Kerberos Setup for Hive View

### Prerequisites

Set up basic Kerberos for the Ambari server that manages Views.

#### Steps

After you have set up Kerberos on the Ambari Views server, you must manually set the following Hive View property:

WebHDFS Authentication	auth=KERBEROS;proxyuser= <ambari-principal·< th=""></ambari-principal·<>	
	name>	

This property is only needed if the view is Custom Configured or Ambari Server is Kerberized before 2.4.0.

# **More Information**

Set Up Kerberos for Ambari Server

# 2.10.4.4. Troubleshooting Hive View

# Table 2.2. Troubleshooting Hive Views Errors

Error	Solution
User: root is not allowed to impersonate admin	HDFS has not been configured for Ambari as a proxy user.
E090 HDFS020 Could not write file /user/admin/hive/jobs/ hive-job-1-2015-10-30_02-12/query.hql [HdfsApiException]	,

# **More Information**

Set up HDFS Proxy User [41]

Set up HDFS User Directory [42]

# 2.10.5. Configuring Pig View

This section describes:

- Configuring Your Cluster for Pig View [49]
- Creating a Pig View Instance [52]

# **2.10.5.1. Configuring Your Cluster for Pig View**

For Pig View to access HDFS, the Ambari Server daemon hosting the view needs to act as the proxy user for HDFS. This allows Ambari to submit requests to HDFS on behalf of the users using the Pig View. This is critical since Pig View will store metadata about the user Pig scripts. This also means users that will access Pig View must have a user directory setup in HDFS. In addition, Pig View uses WebHCat to submit Pig scripts so the View needs a proxy user for WebHCat.

If you are running views in an operational Ambari server (one that is operating the cluster) Ambari does this setup by default. You should verify that the HDFS and WebHCat proxy user and the HDFS user directory settings are correct. If you are running views on a standalone server, you must setup proxy user settings manually.

# **More Information**

Set up HDFS Proxy User [49]

Set up WebHCat Proxy User [50]

Set up HDFS User Directory [51]

# 2.10.5.1.1. Set up HDFS Proxy User

To set up an HDFS proxy user for the Ambari Server daemon account, you need to configure the proxy user in the HDFS configuration. This configuration is determined by the account name the **ambari-server** daemon is running as. For example, if your ambari-server is running as **root**, you set up an HDFS proxy user for **root** with the following:

Steps

- 1. In Ambari Web, browse to Services > HDFS > Configs.
- 2. Under the Advanced tab, navigate to the Custom core-site section.
- 3. Click Add Property... to add the following custom properties:

```
hadoop.proxyuser.root.groups="users"
hadoop.proxyuser.root.hosts=ambari-server.hostname
```

Notice the **ambari-server** daemon account name root is part of the property name. Be sure to modify this property name for the account name you are running the ambari-server as. For example, if you were running **ambari-server** daemon under an account name of **ambariusr**, you would use the following properties instead:

```
hadoop.proxyuser.ambariusr.groups="users"
hadoop.proxyuser.ambariusr.hosts=ambari-server.hostname
```

Similarly, if you have configured Ambari Server for Kerberos, be sure to modify this property name for the **primary Kerberos principal** user. For example, if ambari-server is setup for Kerberos using principal **ambari-server@EXAMPLE.COM**, you would use the following properties instead:

```
hadoop.proxyuser.ambari-server.groups="users"
hadoop.proxyuser.ambari-server.hosts=ambari-server.hostname
```

4. Save the configuration change and restart the required components as indicated by Ambari.

#### **More Information**

Ambari Server for Kerberos

# 2.10.5.1.2. Set up WebHCat Proxy User

You must set up an HDFS proxy user for WebHCat and a WebHCat proxy user for the Ambari Server daemon account.

To setup the HDFS proxy user for WebHCat :

#### Steps

- 1. In Ambari Web, browse to **Services > HDFS > Configs**.
- 2. Under the Advanced tab, navigate to the Custom core-site section.
- 3. Click Add Property... to add the following custom properties:

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

4. Save the configuration change and restart the required components as indicated by Ambari.

To setup a WebHCat proxy user for the Ambari Server daemon account, you need to configure the proxy user in the WebHCat configuration. This configuration is determined by the account name the **ambari -server** daemon is running as. For example, if your ambari - server is running as **root**, you set up an WebHCat proxy user for **root** with the following:

#### Steps

1. In Ambari Web, browse to Services > Hive > Configs.

2. Under the Advanced tab, navigate to the Custom webhcat-site section.

3. Click Add Property... to add the following custom properties:

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

Notice the **ambari-server** daemon account name root is part of the property name. Be sure to modify this property name for the account name you are running the ambari-server as. For example, if you were running **ambari-server** daemon under an account name of **ambariusr**, you would use the following properties instead:

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

Similarly, if you have configured Ambari Server for Kerberos, be sure to modify this property name for the **primary Kerberos principal** user. For example, if ambari-server is setup for Kerberos using principal **ambari-server@EXAMPLE.COM**, you would use the following properties instead:

```
webhcat.proxyuser.ambari-server.groups=*
webhcat.proxyuser.ambari-server.hosts=*
```

4. Save the configuration change and restart the required components as indicated by Ambari.

#### **More Information**

Ambari Server for Kerberos

# 2.10.5.1.3. Set up HDFS User Directory

The Hive View stores user metadata in HDFS. By default, the location in HDFS for this metadata is /user/\${username} where \${username} is the username of the currently logged in user that is accessing the Hive View.



# Important

Since many users leverage the default Ambari admin user for getting started with Ambari, the /user/admin folder needs to be created in HDFS. Therefore, be sure to create the admin user directory in HDFS using these instructions prior to using the view.

To create user directories in HDFS, for each PigView user :

#### Steps

- 1. Connect to a host in the cluster that includes the HDFS client.
- 2. Switch to the hdfs system account user.

su - hdfs

3. Using the HDFS client, make an HDFS directory for the user. For example, if your username is admin, you would create the following directory.

```
hadoop fs -mkdir /user/admin
```

4. Set the ownership on the newly created directory. For example, if your username is admin, you would make that user the directory owner.

```
hadoop fs -chown admin:hadoop /user/admin
```

# 2.10.5.2. Creating a Pig View Instance

To create a Pig View Instance:

# Steps

- 1. Browse to the Ambari Admin interface.
- 2. Click Views, expand the Pig View, and click Create Instance.
- 3. On the **Create Instance** page, select **Version**. If multiple Pig View jars are present, choose one.
- 4. Enter the Details and Settings. The Instance Name appears in the URI, the Display Name appears in the Views drop-down list, and the Description helps multiple users identify the view:

# Figure 2.7. Pig View Details and Settings

liew	PIG	
ersion	1.0.0	•
Details		
	Instance Name	ETLPig
	Display Name	ETL Pig
	Description	Pig View for ETL team
		2 Visible
Settings		
	IDFS Usemame	\$(usemame)
Webh	IDFS Username S Authentication	\$(username) auth=SIMPLE
WebH		
WebHDF3 WebHDF3	S Authentication	
WebHDFS WebHDFS Web	S Authentication HCat Username	auth=SIMPLE

5. Scroll down, and enter the Cluster Configuration information, which tells the Pig View how to access resources in the cluster. For a cluster that is deployed and managed by Ambari, select Local Ambari Managed Cluster:

Cluster Configuration	
Local Ambari Managed Clu	uster
Cluster Name	MyCluster \$
Custom	
VebHDFS FileSystem URI*	webhdfs://namenode:50070
Logical name of the NameNode cluster	
List of NameNodes	
First NameNode RPC Address	
Second NameNode RPC Address	
First NameNode HTTP (WebHDFS) Address	
Second NameNode HTTP (WebHDFS) Address	
Failover Proxy Provider	
WebHCat Hostname*	webhcat-host.example.com
WebHCat Port*	50111

Figure 2.8. Pig View Cluster Configuration

6. Click **Save**, give Permissions to the appropriate users and groups, and click **Go to instance** at the top of the page to go to the view instance.

# **2.10.5.3. Getting Correct Configuration Values for Manually-Deployed Clusters**

If you have manually deployed your cluster, you must enter cluster configuration values in the Pig View Create Instance page. The following table explains where you can find cluster configuration settings in Ambari.

Scripts HDFS Directory*	/user/\${username}/pig/scripts
Jobs HDFS Directory*	/user/\${username}/pig/jobs
WebHDFS FileSystem URI*	Click HDFS > Configs > Advanced hdfs-site > dfs.namenode.http-address. When you enter the value in the view definition, pre-pend webhdfs:// to the value you find in the advanced HDFS configuration settings. For example, webhdfs:// c6401.ambari.apache.org:50070
WebHCat Hostname*	Click <b>Hive &gt; Configs &gt; Advanced &gt; WebHCat Server</b> <b>&gt; WebHCat Server host</b> to view the hostname. For example, c6402.ambari.apache.org
WebHCat Port*	Click <b>Hive &gt; Configs &gt; Advanced &gt; Advanced webhcat-</b> <b>site &gt; templeton.port</b> to view the port number. For example, 50111

### For NameNode High Availability

The following values must be entered for primary and secondary NameNodes:

First NameNode RPC Address or Second NameNode RPC Address	Select the primary or secondary NameNode to view settings from that host in the cluster. When you enter the value in the view definition, pre-pend $http://to$ the value you find in the advanced hdfs-site settings. For example, http://c6401.ambari.apache.org:8020
First NameNode HTTP (WebHDFS) Address or Second NameNode HTTP (WebHDFS) Address	Click HDFS > Configs > Advanced > Advanced hdfs- site > dfs.namenode.http-address. When you enter the value in the view definition, pre-pend http:// to the value you find in the advanced hdfs-site settings. For example, http://c6401.ambari.apache.org:50070

To get First NameNode RPC Address values:

### Steps

1. In Ambari Web, browse to the **HDFS Summary** page. Click **NameNode** (primary) or **SNameNode** (secondary) to view the host page:

# Figure 2.9. HDFS Service Page in Ambari

• HDFS	Summary	Heatmaps	Configs	Quick Links-	Service Actions
YARN					
MapReduce	Summary				No alert
🖵 Tez		NameNode	Started No alerts	Disk Remaining	1.1 TB / 1.3 TB (91.10%)
Hive		SNameNode	Started No alerts	Blocks (total)	738
HBase	c	DataNodes Status	3/3 Staned 3 live / 0 dead / 0 decommissi		0 corrupt replica / 0 missing / 0 under replicated
Pig			0/0 JournalNodes Live	Total Files + Directories	947
Oozie		NFSGateways	0/0 Started	Upgrade Status	No pending upgrade
2 ZooKeeper	Na	ameNode Uptime	6.12 days	Safe Mode Status	Not in safe mode
Storm		NameNode Heap	102.9 MB / 1011.3 MB (10.2%	6 used)	
Ambari Infra	Disk U	Isage (DFS Used)	5.1 GB / 1.3 TB (0.40%)		
Ambari Met		(Non DFS Used)	109.1 GB / 1.3 TB (8.50%)		

- 2. On the host page, click **Configs > Advanced**.
- 3. Enter rpc in the Filter search well at the top right corner of the page or navigate to the **Advanced hdfs-site** settings to find the dfs.namenode.rpc-address value that you can enter into the Pig View definition. Here is an example of using the Filter to locate a value:

Figure 2.10. Using the Filter to Search Advanced hdfs-site Settings

Summary Heatmaps Configs	Quick Links+		Service Actions -
Group Default (3)  Manage (	Config Groups	rpc	•
7 days ago HDP-2.5			
X - V1 - admin authored on I	Mon, Oct 03, 2016 09:50		Discard Save
Settings Advanced			
<ul> <li>Advanced hdfs-site</li> </ul>			
dfs.namenode.rpc- c6401.ar	mbari.apache.org:8020	dfs.nar	menode.rpc-address
address	inginabaonaroitironen Y	PPC a	ddress that handles all

# 2.10.5.4. User Permissions for Pig View

After saving the Pig View instance definition, grant view permissions for all Pig View users:

Views / M	y Pig \	View Go to Instance	Delete Instance
View Version	PIG 1.0.0		
Details			🖊 Edit
Instance	Name	MyPigView	
Display	Name	My Pig View	
Desc	ription	description	
		🗟 Visible	
Permissions			
Permission	Grant p	ermission to these users	Grant permission to these groups
Use			
Settings			✓ Edit

Figure 2.11. Granting User Permissions to Pig View

# 2.10.5.5. Kerberos Setup for Pig View

# Prerequisites

Set up basic Kerberos for the Ambari views server.

#### Steps

After you have set up Kerberos on the Ambari Views server, you must manually set the following Pig View property:

WebHDFS Authentication auth=KERBEROS;proxyuser=<ambari-principalname>

> This property is only needed if the view is Custom Configured or Ambari Server is Kerberized before 2.4.0.

For example, see the following figure:

# Figure 2.12. Kerberos Settings for Pig View

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

# **More Information**

Set Up Kerberos for Ambari Server

# 2.10.6. Configuring Slider View

To configure Slider View:

### Steps

- 1. From the Ambari Admin interface, browse to the Views section.
- 2. Click to expand the **Slider** view and click **Create Instance**.
- 3. Enter the instance name, the display name and description.
- 4. Enter the configuration properties for your cluster.

Property	Description	Example
Ambari Server URL (required)	The Ambari REST URL to the cluster resource.	http://ambari.server:8080/api/v1/ clusters/MyCluster
Ambari Server Username (required)	The username to connect to Ambari. Must be an Ambari Admin user.	admin
Ambari Server Password (required)	The password for the Ambari user.	password
Slider User	The user to deploy slider applications as. By default, the applications will be deployed as the "yarn" service account user. To use the current logged-in Ambari user, enter \${username}.	joe.user or \${username}
Kerberos Principal	The Kerberos principal for Ambari views. This principal identifies the process in which the view runs. Only required if your cluster is configured for Kerberos. Be sure to configure the view principal as a proxy user in core-site.	view-principal@EXAMPLE.COM
Kerberos Keytab	The Kerberos keytab for Ambari views. Only required if your cluster is configured for Kerberos.	/path/to/keytab/view- principal.headless.keytab

5. Save the view.

# 2.10.7. Configuring SmartSense View

This chapter describes:

- Configuring Your Cluster for SmartSense View [58]
- Creating a SmartSense View Instance [58]
- Using SmartSense View [110]

# 2.10.7.1. Configuring Your Cluster for SmartSense View

When you deploy a cluster with Ambari, a SmartSense View instance is automatically created as long as an Ambari Agent is deployed on the host running the Ambari Server.

If an Ambari Agent is *not* installed on the Ambari Server host, the view will not be automatically created, and you will have to add a SmartSense instance manually.

Before accessing SmartSense View, you should enter your SmartSense user ID, account name (both are available in the Hortonworks support portal in the **Tools** tab), and email address in the SmartSense service configuration properties.

#### **More Information**

Creating a SmartSense View Instance [58]

# 2.10.7.2. Creating a SmartSense View Instance

To create a SmartSense view instance manually:

Steps

- 1. Browse to the Ambari Admin interface.
- 2. Click Views, expand the HORTONWORKS\_SMARTSENSE menu, and click Create Instance:

🚕 Amberl			_	_	_		▲ admin +
Clusters		Views				Search	Q,
mycluster Permissions	ø	View Name	Instances				
Go to Dashboard		> CAPACITY-SCHEDULER	1.0.0 (1)				
Versions		> FILES	1.0.0 (1)				
		> HIVE	1.0.0 (1)				
III Views		HORTONWORKS_SMARTSE	1.3.0.0-1505 INSE	(1)			
		Se	nartSense View	1.3.0.0-1505		SmartSense - Hortonwor	ks SmartSe
LUser + Group Management			+ Create Instance				
Users		1					
Groups		▶ PIG	1.0.0 (0)				
		SLIDER	2.0.0 (0)				
		> TEZ	0.7.0.2.3.4.0	460 (1)			

3. On the **Create Instance** page, select the **Version**. If multiple SmartSense View jars are present, choose one.

Clusters	Views / Create Insta	ance	
mycluster (5 Permissions Go to Dashboard Versions	View HORTONY Version 1.3.0.0-10 Details	NORKS_SMARTSENSE	
Vews	Instance Name	SmartSense_View	0
User + Group Management	Display Name Description	SmartSenseView( My SmartSense View	
Users Groups		Vable	
	Settings		
	hst.server.url*	http://o6401.ambari.apache.org:9000/	
	hst.server.username	admin	
	hst.server.password*		9

4. Enter the following view instance details:

Instance Name	This is the SmartSense view instance name. This value should be unique for all SmartSense view instances you create. This value cannot contain spaces and is required.
Display Name	This is the name of the view link displayed to the user in Ambari Web.
Description	This is the description of the view displayed to the user in Ambari Web.

5. Enter the following view instance settings:

hst.server.url	This is the HST server URL. This should be <pre>http:// <hst_host>:9000/</hst_host></pre>
hst.server.username	The default username is 'admin'.
hst.server.password	Unless changed after installation, the default password is 'admin'.

6. Click Save.

# 2.10.8. Configuring Storm View



# Important

This view has been marked deprecated.

## Prerequisites

Storm View requires that the cluster is managed by Ambari; the view utilizes the Ambari Server API.

Prepare your Ambari Server for hosting views.

It is strongly recommended that you:

- increase the amount of memory available to your Ambari Server
- run a standalone Ambari Server to host the views

To create a Storm View instance:

### Steps

1. Browse to the Ambari Admin interface: from the dashboard, open the administrator account menu and click **Manage Ambari**:

🝌 Ambari					Dashboard Services	Hosts Alerts	Admin	ш	🛦 admin 🔹
O HDFS	8	Metrics	Heatmaps	Config History					About Manage Amba
A YARN MapReduce2	۰	Metric Actio	ns• L	ist 1 hour •					Settings Sign out
I Tez		HDFS Disk Usi	age	DataNodes Live	HDFS Links	Memory Usage		Network L	Isage
A Hive	Ð				NameNode	1 I I I	2	39.0 KB	4
Pig Pig		63%		3/3	Secondary NameNode 3 DataNodes	1.8 GB		00.0 100	
ZooKeeper					More	1.0 00		19.5 KB	

2. Click Views, expand the Storm\_Monitoring menu, and click Create Instance:

Clusters	Views o		Search
storm (	3 View Name	Instances	
Go to Dashboard	> CAPACITY-SCHEDULI	ER 1.0.0 (0)	
Versions	> FILES	1.0.0 (0)	
Remote Clusters	> HIVE	1.0.0 (0) , 2.0.0 (0)	
	> PIG	1.0.0 (0)	
Wews	> SLIDER	2.0.0 (0)	
Views	✓ Storm_Monitoring	0.1.0 (1)	
View URLs	Storm View	0.1.0	Storm View (auto-created)
LUser + Group Management	+ Create Inst	tance	
Users	> TEZ	0.7.0.2.5.0.0-655 (0)	
Groups	> ZEPPELIN	1.0.0 (0)	

3. On the **Create Instance** page, select the **Version**. If multiple Storm View versions are present, choose one.

🚕 Ambari			🛔 admin -		
Clusters	Views / Create Instance				
Ighdp250 C Roles Go to Dashboard Versions Remote Clusters	View Storm_M Version 0.1.0 Details	fonitoring -			
III Views Views View URLs	Instance Name* Display Name* Description*	Storm_View1 Storm/View Storm View			
LUser + Group Management		S SUMP			
Users Groups	Settings				
Coups	Storm Hostname*	storm-host-node1.com 8744			
		Cancel	Save		

4. Enter the following view instance details:

Property	Description	Example Value
Instance Name	This is the Storm view instance name. This value should be unique for all Storm view instances you create. This value cannot contain spaces, and it is a required setting.	Storm_View1
Display Name	This is the name of the view link displayed to the user in Ambari Web.	StormView
Description	This is the description of the view displayed to the user in Ambari Web.	StormView

# Table 2.3. Storm View Instance Details

5. Enter the following view instance settings:

# **Table 2.4. Storm View Instance Settings**

Property	Description	Example Value
Storm Hostname	This is the hostname where the Storm UI Server is running.	storm-host-nodel.com
Storm Port	This is the port where the Storm UI Server is listening.	8744

Settings depend on cluster and deployment factors in your environment. You can typically leave the default settings unless you are using Storm View with a Kerberosenabled cluster.

# 6. Click Save.

# **More Information**

Preparing Ambari Server for Views [5]

Running Ambari Server Standalone [7]

Set Up Kerberos for Ambari Server

Configuring Views for Kerberos

# **2.10.9. Configuring Tez View**

This sections describes:

- Configuring Your Cluster for Tez View [62]
- Creating or Editing a Tez View Instance [63]

# 2.10.9.1. Configuring Your Cluster for Tez View

When you deploy a cluster with Ambari, a Tez View instance is automatically created. However, you must verify that the configurations listed in the following table have been correctly set. If you have manually deployed your cluster, you must set the properties listed in the following table to configure your cluster before you create Tez View on your standalone Ambari server.

To configure your cluster for Tez View:

### Steps

1. Confirm the following configurations are set:

# Table 2.5. Cluster Configurations for Tez View

Component	Configuration	Property	Comments
YARN	yarn-site.xml	yarn.resourcemanager. system-metrics- publisher.enabled	Enable the generic history service in the Timeline Server. Verify that this property is set to true.
YARN	yarn-site.xml	yarn.timeline- service.enabled	Enable the Timeline Server for logging details. Verify that this property is set to true.
YARN	yarn-site.xml	yarn.timeline- service.webapp.address	Value must be the IP:PORT on which the Timeline Server is running.

2. If you changed any settings, you must restart the YARN ResourceManager and the Timeline Server for your changes to take effect.

# **2.10.9.2. Creating or Editing a Tez View Instance**

Depending on whether you must create a new Tez View instance for a manually deployed cluster or modify an Ambari-created Tez View, using one of the following procedures:

To modify a Tez View instance on an Ambari-managed cluster:

# Steps

- 1. Browse to the Ambari Admin interface.
- 2. Click Views and expand the Tez View.
- 3. On the **Create Instance** page, change the appropriate configuration parameters.
- 4. Select Local Ambari-Managed Cluster:

iew	TEZ				
ersion	0.7.0.2.3.	0.0-2108 \$			
Details					
	Instance Name				E
	Display Name				
	Description				
		Visible			
Cluster Cor	nfiguration				
O Local A	mbari Managed Clu	uster			
	Cluster Name	MyCluster	\$		
Custom	1				
YARN Time	eline Server URL	yam.timeline-servi	ce.hostname:8188		
YARN R	esourceManager URL	yam.resourceman	ager.hostname:8088		

# Figure 2.13. Tez View Create Instance Page



# Important

Secure clusters that use wire encryption (SSL/TSL) cannot use the **Local Ambari Managed Cluster** option. Instead you must configure the view manually.

5. Click **Save**, grant Permissions on the view , then click **Go to instance** to use the view.

To create a new Tez View instance for a manually-deployed cluster:

#### Steps

- 1. Browse to the Ambari Admin interface.
- 2. Click Views, expand the Tez View, and click Create Instance.
- 3. On the **Create Instance** page, select the **Version**.
- 4. Enter the Details (required). The Instance Name appears in the URI, the Display Name appears in the Views drop-down list, and the Description helps multiple users identify the view.
- 5. Scroll down to the Cluster Configuration, verify that **Custom** is checked and enter the following values, which tell the Tez View how to access resources in the cluster:

# Table 2.6. Cluster Configuration Values for Tez View in Ambari

Property	Value
	The URL to the YARN Application Timeline Server, used to provide Tez information. Typically, this is the yarn.timeline-service.webapp.address property that is specified in the etc/hadoop/conf/ yarn-site.xml.

Property	Value
	When you enter the value in the view definition, pre- pend "http://" to the value you find in the yarn-site.xml file. For example, http:// <timeline server<br="">host&gt;:8188</timeline>
	For wire encryption-enabled clusters:
	Set this based on the value of <pre>yarn.timeline- service.webapp.https.address</pre> in <pre>yarn- site.xml</pre>
	When you enter the value in the view definition, pre- pend "https://" to the value. For example, https:// <timeline host="" server="">:8190</timeline>
YARN ResourceManager URL (required)	The URL to the YARN ResourceManager, used to provide YARN Application data. Typically, this is the yarn.resourcemanager.webapp.address property that is specified in the etc/hadoop/conf/ yarn-site.xml.
	When you enter the value in the view definition, pre- pend "http://" to the value you find in the yarn-site.xml file. For example, http:// <resourcemanager host&gt;:8088</resourcemanager 
	<b>Important:</b> If YARN ResourceManager HA is enabled, provide a comma-separated list of URLs for all the Resource Managers.
	For wire encryption-enabled clusters:
	Set this based on the value of yarn.resourcemanager.webapp.https.address in yarn-site.xml
	When you enter the value in the view definition, pre- pend "https://" to the value. For example, https:// <resourcemanager host="">:8090</resourcemanager>

- 6. Click Save and grant Permissions on the view.
- 7. At the top of the view instance configuration page, click **Go to instance**.
- 8. When your browser is at the view instance page, copy the URL for the Tez View from your browser address bar:

### Figure 2.14. Tez View Instance Page

À Ambari 20clu	ster • Cops 7 slerts		Da	shboard Services	s Hosts <mark>3</mark> Alerts	Admin III	📥 admin -
😭 All Dags							
0					Last refreshed at 06	Mar 2015 17:21:39	C Refresh
Number of rows 10						Page 1 💮 🤇	30 🗘
Number of rows 10 Dag Name	a. Id	Submitter	Status	Start Time	End Time	Page 1 💮 🕻 Duration	Applice
		Submitter	Status	Start Time	End Time		

9. In tez-site.xml, specify the URL that you copied in Step 8 as the value for the tez.tez-ui.history-url.base property, and save the file.

10Restart the HiveServer2 daemon to make sure that your changes to tez-site.xml take effect.



#### Important

If your cluster is configured for Kerberos, you must set up Ambari Server for Kerberos for the Tez View to access the ATS component.

#### **More Information**

User Permissions for Tez Views [66]

Using Tez View [117]

Kerberos Setup for Tez View [67]

#### 2.10.9.2.1. User Permissions for Tez Views

After saving the Tez View instance definition, grant permission on the view for all Tez View users:

#### Figure 2.15. Granting User Permissions to Tez View

iews / Te			
rsion	0.7.0.2.3.	3.0-2108	
Details			≠ Edt
In	stance Name	TEZ_CLUSTER_INSTANCE	
	lisplay Name	Tez View	
	Description	Monitor and debug all Tez jobs, submitted by Hive	queries and Pig scripts (auto-created)
		Visible	
	Grant per		mission to these groups
Permissions Permission Use	Grant per	mission to these users Grant per	mission to these groups
Permission Use	Add U	mission to these users Grant per	04p
Use Cluster Confi Local Ant	add U guration	mission to these users Grant per	04p
Permission Use Cluster Config & Local Ant	Add U	mission to these users Grant per	04p
Permission Use Cluster Config & Local Ant	add U guration	mission to these users Grant per	04p
Permission Use Cluster Config © Local Amt	guration pari Managed Ciu Cluster Name	mission to these users Grant per	



## Note

To grant access to all Hive and Pig users, create a group that contains these users, and then grant permission to use the Tez View to that group.

#### **More Information**

Managing Users and Groups

#### 2.10.9.2.2. Kerberos Setup for Tez View

#### Prerequisites

Set up basic Kerberos for views, on the Ambari Views server.

After you have set up basic Kerberos, Tez View requires that you set the following configuration properties:

Steps

1. On the timeline server host, set the following values for properties in the YARN configuration for Ambari-managed clusters or the <code>yarn-site.xml</code> for manually deployed clusters:

#### Table 2.7. Kerberos Settings for Tez View

Property	Value
yarn.timeline-service.http- authentication.proxyuser. <ambari- principal-name&gt;.hosts</ambari- 	*
yarn.timeline-service.http- authentication.proxyuser. <ambari- principal-name&gt;.users</ambari- 	*
yarn.timeline-service.http- authentication.proxyuser. <ambari- principal-name&gt;.groups</ambari- 	*
Timeline HTTP Auth	kerberos
RM HTTP Auth	kerberos



## Note

Tez View will not work in a kerberized cluster, if Timeline HTTP Auth and RM HTTP Auth properties are not set to kerberos.

For example, if the Kerberos principal used for the Ambari server is ambariservice@EXAMPLE.COM, replace ambari-service with <ambari-principalname>.

2. Restart the Timeline Server so your configuration changes take effect.

#### **More Information**

Set Up Kerberos for Ambari Server

# **2.10.10. Configuring Workflow Manager View**

Before you can access Workflow Manager, you must complete several configuration tasks.

See the following content:

Section 2.10.10.1, "Configuring Your Cluster for Workflow Manager View" [68]

Section 2.10.10.2, "Kerberos Setup for Workflow Manager" [70]

Section 2.10.10.3, "Creating and Configuring a Workflow Manager View Instance" [71]

# 2.10.10.1. Configuring Your Cluster for Workflow Manager View

For Workflow Manager View to access HDFS, the Ambari Server daemon hosting the view must act as the proxy user for HDFS. This allows Ambari to submit requests to HDFS on behalf of the Worklow Manager View users.

Each Worklow Manager View user must have a user directory set up in HDFS.

If the cluster is configured for Kerberos, ensure that the Section 2.8, "Configuring Views for Kerberos" [12] has been completed.

#### 2.10.10.1.1. Set up HDFS Proxy User



# Note

If you previously set up the proxy user for another View, you can skip this task.

To set up an HDFS proxy user for the Ambari Server daemon account, you need to configure the proxy user in the HDFS configuration. This configuration is determined by the account name the **ambari-server** daemon is running as. For example, if your ambari-server is running as **root**, you set up an HDFS proxy user for **root**.

#### Steps

- 1. In Ambari Web, browse to Services > HDFS > Configs.
- 2. Under the Advanced tab, navigate to the Custom core-site section.
- 3. Click Add Property... to add the following custom properties:

```
hadoop.proxyuser.root.groups="users"
hadoop.proxyuser.root.hosts=ambari-server.hostname
```

Notice the ambari-server daemon account name root is part of the property name. Be sure to modify this property name for the account name you are running the ambariserver as. For example, if you were running ambari-server daemon under an account name of ambariusr, you would use the following properties instead:

```
hadoop.proxyuser.ambariusr.groups="users"
hadoop.proxyuser.ambariusr.hosts=ambari-server.hostname
```

Similarly, if you have configured Ambari Server for Kerberos, be sure to modify this property name for the **primary Kerberos principal** user. For example, if ambari-server is setup for Kerberos using principal **ambari-server@EXAMPLE.COM**, you would use the following properties instead:

```
hadoop.proxyuser.ambari-server.groups="users"
hadoop.proxyuser.ambari-server.hosts=ambari-server.hostname
```

4. Save the configuration change and restart the required components as indicated by Ambari.

#### **More Information**

Ambari Server for Kerberos

### 2.10.10.1.2. Set up HDFS User Directory

You must set up a directory for each user that accesses the Workflow Manager View. Workflow Manager View stores user metadata in the user directory in HDFS.

#### About This Task

By default, the location in HDFS for the user directory is  $/user/\${username}$ , where  $\${username}$  is the username of the currently logged in user that is accessing Workflow Manager View.



#### Important

Since many users leverage the default Ambari *admin* user for getting started with Ambari, you should create a /user/admin directory if one does not exist, in addition to directories for other Workflow Manager View users.

#### Steps

- 1. Connect to a host in the cluster that includes the HDFS client.
- 2. Switch to the HDFS system account user.

#### su - hdfs

- 3. If working on a secure Kerberos cluster:
  - a. Destroy any existing Kerberos tickets:

#### kdestroy

If no ticket is found, you get an error message that you can ignore: No credentials cache file found while destroying cache

b. Obtain a Kerberos ticket-granting ticket:

```
kinit -kt /etc/security/keytabs/hdfs.headless.keytab hdfs
```

4. Using the HDFS client, make an HDFS directory for the user.

For example, if your username is *wfmadmin*, you would create the following directory.

hadoop fs -mkdir /user/wfmadmin

5. Set the ownership on the newly created directory.

For example, if your username is *wfmadmin*, the directory owner should be *wfmadmin:hadoop*.

hadoop fs -chown wfmadmin:hadoop /user/wfmadmin

6. Log in as root user.

su -

7. Access the Kerberos administration system.

kadmin.local

8. Create a new principal and password for the user.

You can use the same user name and password that you used for HDFS directory.

addprinc -pw [wfmadmin-password] wfmadmin@EXAMPLE.COM

9. Repeat steps 2 through 8 for any additional Workflow Manager View users.

## 2.10.10.2. Kerberos Setup for Workflow Manager

If you install Ambari using Kerberos, the Kerberos settings for Oozie that are required for Workflow Manager are configured automatically. The image below shows what the settings should look like.

#### Figure 2.16. Workflow Manager Kerberos Configuration for Oozie

<ul> <li>Advanced oozie-site</li> </ul>			
oozie.authentication. kerberos.name.rules	RULE:[1:\$1@\$0](ambari-qa-cl1@EXAMPLE.COM)s/.*/ambari-qa/RULE:[1:\$1@\$0](cstm-	•	
oczie.authentication.type	kerberos	0	c
oozie.service.Hadoop AccessorService. kerberos.enabled	true	0	c
Custom cozie-site			
oozie.authentication. kerberos.keytab 🔒	/etc/security/keytabs/spnego.service.keytab	•	
oozie.authentication. kerberos.principal	HTTP/_HOST@EXAMPLE.COM	•	
oozie.service.Hadoop AccessorService. kerberos.principal 🖨	oozie/_HOST@EXAMPLE.COM	0	
cozie.service.Hadoop AccessorService.keytab. file	/etc/security/keytabs/oozie.service.keytab	•	
Add Property			

### 2.10.10.2.1. Set up Proxy User for Oozie

If you are using Kerberos, you must configure the proxy user for Oozie. Workflow Manager uses Oozie as its scheduling engine.

#### Steps

1. In Ambari Web, browse to **Services > Oozie > Configs**.

- 2. Expand the **Custom oozie-site** section.
- 3. Click Add Property... to add the following custom properties:

oozie.service.ProxyUserService.proxyuser.[ambari-server-cl1].groups=\*

oozie.service.ProxyUserService.proxyuser.[ambari-server-cl1].hosts=\*

Replace *ambari-server-cl1* with the server principal name used when configuring Kerberos.

#### Figure 2.17. wfm-oozie-proxy-user.png

<ul> <li>Custom oozie-site</li> </ul>			
oozie.service.	•	•	•
ProxyUserService. proxyuser.ambari-server- cl1.groups			
oozie.service.	•	0	•
ProxyUserService. proxyuser.ambari-server- cl1.hosts			
Add Property			

4. Save the configuration change and restart the required components as indicated by Ambari.

#### **More Information**

Set Up Kerberos for Ambari Server

# 2.10.10.3. Creating and Configuring a Workflow Manager View Instance

You can configure multiple WFM View instances. You might want to have multiples instances if you want to assign different users and permissions for each instance. You can also have instances that run locally and others that run remotely.

Steps

- 1. Click Manage Ambari to open the Ambari Admin user interface.
- 2. Click Views, expand the Workflow\_Manager View, and click Create Instance.
- 3. On the Create Instance page, select the Version.

If multiple View versions are present, choose one.

4. Enter the following view instance Details:

Property	Description	Example Value
Instance Name	This is the Workflow Manager View instance name and must be unique for all instances you create. The instance name cannot be modified after the instance is created.	wfm_local_instance This value cannot contain spaces or special characters other than an underscore.
Display Name	This is the name of the view link displayed to the user in Ambari Web. The display name can be modified after the instance is created.	WFM View This value can contain spaces and underscores, but no other special characters.
Description	This is the description of the view displayed to the user in Ambari Web.	Local instance of WFM
Visible	This checkbox determines whether the view is displayed to users in Ambari Web.	Visible or Not Visible

## Table 2.8. Workflow Manager View Instance Details

- 5. Under Cluster Configuration, select the Local Cluster or Remote Clusterinstance type and select the cluster name.
- 6. Review the remaining settings and make changes as desired.

The Settings and Cluster Configuration options depend on a few cluster and deployment factors in your environment. Typically, you can accept the default Settings unless you are using the Workflow Manager View with a Kerberos-enabled cluster.

7. Click Save.

The instance is created and a success message displays.

- 8. Scroll to the **Permissions** section at the bottom of the Views configuration form.
- 9. Grant permission on the Workflow Manager View for the set of users and groups who can access the view instance.

Permission	Grant permission to these users	Grant permission to these groups
Jse	admin	Add Group
Local Cluster P	ermissions	
	<ul> <li>Cluster Administrator</li> </ul>	
	<ul> <li>Cluster Administrator</li> <li>Cluster Operator</li> </ul>	
	Cluster Operator	

#### More Information

For descriptions of the View instance types, see Section 2.3, "Configuring View Instances" [9].

You can access the Workflow Manager documentation in the Workflow Management guide.

# 3. Using YARN Queue Manager View

The Yarn Capacity Scheduler allows for multiple tenants in an HDP cluster to share compute resources according to configurable workload management policies.

The YARN Queue Manager View is designed to help Hadoop operators configure these policies for YARN. In the View, operators can create hierarchical queues and tune configurations for each queue to define an overall workload management policy for the cluster.

The YARN Queue Manager View is designed to help hadoop operators configure workload management policies for YARN. In YARN Queue Manager View, operators can create hierarchical queues and tune configurations for each queue to define an overall workload management policy for the cluster.

#### Next Steps

- Setting up Queues [74]
- Configuring Queues [79]
- Enabling Preemption [80]
- Setting YARN Queue Priorities [84]
- Configuring Cluster Scheduler Settings [86]
- Applying the Configuration Changes [87]

# **3.1. Setting up Queues**

To set up Capacity Scheduler queues on a view instance.

#### Steps

1. On the YARN Queue Manager view instance configuration page, click Add Queue.

The queue will be added under the top level, or "root" queue. A "default" queue already exists under the root queue.

To return to a *previously created* YARN Queue Manager view instance:

- a. Click **Views** on the Manage Ambari page.
- b. Click **CAPACITY-SCHEDULER**.
- c. Click the applicable YARN Queue Manager view instance, then click **Go to instance** at the top of the page.

🔬 Ambari test,	_cluster1 (Brank 1 elect		Dashboard	Services	Hosts	Alerts	Admin	 🛦 admin 👻
+ Add Queue	Actions -	Click on a queue to the left for	details.					
root (100%)	~							
default (10	0%) 🗸							
Scheduler	× .							
Maximum Applications	10000							
Maximum AM Resource	20 %							
Node Locality Delay	40							
Calculator	org.apachs.hadoop.yarr							
Mandana								
Versions								
vi Current 4	6 years ago load							

2. Type in a name for the new queue, then click the green check mark to create the queue. In the following example, we're creating the "Engineering" queue.

🔬 Ambari test_	cluster1 (Capa <mark>1</mark>	alert		Dashboard	Services	Hosts	Alerts	Admin	 🛓 admin 👻
Engineering		× ~	Click on a queue to the left for	details.					
root (100%)		× .							
default (10	096)	× .							
Scheduler		~							
Maximum Applications	10000								
Maximum AM Resource	20 96								
Node Locality Delay	40								
Calculator	org.apache.hadoop	yam							
Versions									
v1 Current 4	6 years ago	load							

3. The "Engineering" queue is added, and its configuration page appears.

🔬 Ambari 🛛 test_cluster 💼	an Rateta		Dashboard Services H	losts Alerts	Admin	🔺 admin 🔹
+ Add Queue	C Actions -	Engineering 2 root.Engineering				×
default (100%)	# ¥	Capacity	Level Total	1	100%	
Engineering (2%)	© © V	Engineering Cepacity: 0 %	Max Capacity: 0	% O	C the	le node labels
Maximum Applications 10000			V Show Peer Level Queues			
Maximum AM Resource 20	96	Access Control and Status	Resources	1		
Node Locality Delay 40		State Purving Stopped		User Limit Factor	1	
Calculator Defaul	Resource Calculator Ø	Administer Anyone Custom	M	nimum User Limit	100 96	
Queue Mappings		Queue Effective Anyone	Maxir	num Applications	Inherited	
Queue Mappings Oto	bled	Administrators	Maxim	um AM Resource	Inheritz 56	
		Submit Anyone Custom Applications		Priority	0	
Versions		Effective Users Anyone		Ordering Policy	FIFO \$	
vi Current version	1 load					

4. The sum of queue capacities at any level in the YARN Queue Manager configuration must total 100%. Here the default queue is already set to 100%. Therefore, if we try to set the "Engineering" queue capacity to 60%, error messages appear warning that the total at this level is 160%.

Ambari test_clu	ster (Days Daters				Dashboard	Services	Hosts	Alerts	Admin		A admir
+ Add Queue	C Ad	tions -	Engineering	7							х
default (100%	0	• • •	Capacity			Level Total			189%		
Engineering (	3016) (	▲ □	Engineering Cepacity: 60	%	- 🖸 Max C	apacity: 60	96			O Enabl	e node labels
Maximum Applications	10000				v Show Peer	Level Queues					
Maximum AM Resource	20 96		Access Control a	nd Status		Resour	ces				
Node Locality Delay	40		State	Running Stopped			User Li	nit Factor	1		
Calculator	Default Resource Calo	ulator 8	Administer	Anyone Custom			Minimum	Jser Limit	100	96	
Queue Mappings			Queue	e Argone		м	aximum Ap	plications	Inhorite		
Queue Mappings Override	C Disabled		Administraters			Ma	simum AM	Resource	Inhorite	96	
			Submit Applications	Anyone Custom				Priority	0		
Versions			Effective Users	Anyone			Order	ing Policy	FIFO \$		
Current	version1	load									

5. If we click the "default" queue and set its capacity to 0%, the Level Total bar at the top of the page lists the total queue capacity at this level as 60%.

🧟 Amberi test_clu	stor (Exec Extents				Dashboard	Services	Hosts	Alerts	Admin	ш	🛦 admin 🔹
+ Add Queue	C Actio	ns - /	default 🗷								×
default (0%) 🛽			Capacity			Level Total		60%			
Engineering (6	10%) 🥔	(A 0)	default Capacity: 0	% O	- 💿 Max Ca	pacity: 100	96			O Enabl	e node labels
Scheduler Maximum Applications	10000	~			v Show Peer i	avel Queues					
Maximum AM Resource	20 96		Access Control a	ind Status		Resour	005				
Node Locality Delay	40		State	Nurving Stopped			User Lir	nit Factor	1		
Celculator	Default Resource Calcula	tor 8	Administer Queue	Anyone Custom			Minimum	Aser Limit	100	96	
Queue Mappings			Effective	· Anyona		M	oximum Ap	plications	Inherb	d	
Queue Mappings Override	O Disabled		Administrators			Ma	dimum AM	Resource	Interts	96	
			Submit Applications	Anyone Custom				Priority	0		
Versions			Effective Users	· Anyone			Order	ing Policy	0		
vi Current	version1	load									

6. To add more queues at the root level, click the **root** queue, then click **Add Queue**. In the following example, we have added a "Support" queue set to 10% of the level capacity, and a "Marketing" queue set to 30%. The root-level queue capacities now total 100%.

+ Add Queue	C Actions -	Marketing 2		3
default (0%)		Capacity	Level Total	·
Engineering (60%) Marketing (50%) Support (10%)	* 0 * 0	Marketing Capacity: 20 (%)	Max Capacity: 40 %	C Enable node labels
Scheduler	×	Access Control and Status	Resources	
Adaximum AM Resource 20 % Node Locality Delay 40	II te Calculator a	State     Rurwing     Stopped       Administer Guese     Anyone     Custom       Effective Administraters     Anyone     Custom       Submit Applications     Anyone     Custom       Effective Users     Image: Custom     Image: Custom	Maximum AM Resource	00 96 herited herite 96

- 7. To save your configuration, click **Actions > Save Only**.
- 8. On the **Notes** pop-up, enter an optional description of your changes, then click **Save**.

Each version is retained and listed in the Versions box.

Ambari test_cluster 🚮	en Calarta		Dashbor	rd Services	Hosts Al	erta Admir		A admin
	Actions - Restart ResourceManager Refresh Queues	Marketing Capacity		Level Total		100%		×
Engle Bothold		Marketing Capacity: 30 %	M	ax Capacity: 40	16		OBN	ble node labels
<ul> <li>oopport (rowy)</li> </ul>	- •		v Show	Peer Level Queues				
Scheduler	× .	Access Control and St	atus	Resour	ces			
Maximum Applications 10000		State	wing Stopped		User Limit Fr	etter 1		
Maximum AM Resource 20	96		yone Custom		Minimum User I	Limit 100	96	
Node Locality Delay 40		Queue Effective	Angenes		aximum Applica	line	fled	
Calculator Detaut	Resource Calculator 8	Administraters		Ma	ximum AM Resc	Inter Inter	ni 96	
Queue Mappings		Submit An Applications	yone Custom		Pri	0		
Queue Mappings Override	bied		Anyona		Ordering P	PIPO PIPO		
Versions								
Current a few seconds	ago load							
version 1	load							

9. To build a queue hierarchy, click a top level queue, then click Add Queue.

In the following example, the "qa" and "development" queues have been added under the "Engineering" queue.

Ambari test_c	luster • Bages Bailerts		De	ashboard Services Hosts Alerts	Admin III 🔺 admir
+ Add Queue	Actio	18 -	qa 🍘		3
root (100%)		~	root.Engineering.qa		
default (0%)	0	<i>•</i> •	Capacity	Level Total	100%
Engineering	(6016)	×	Q2		C Enable node labels
<ul> <li>develo</li> </ul>	pment (20%)	ø 🗸	Capacity: 80 %	Max Capacity: 100 %	0
• qa (80		••		Show Peer Level Queues	
Marketing (     Support (10		a v a v	Access Control and Status	Resources	
	~1		State Running Stopped	User Limit Factor	1
Scheduler		× .	Administer Anyone Custom	Minimum User Limit	100 %
Maximum Applications	10000 🗄		Queue	Maximum Applications	Inherited
Maximum AM Resource	20 %		Effective Administraters	Maximum AM Resource	Inheritz 96
Node Locality Delay	40		Submit Anyone Custom	Priority	0
Calculator	Default Resource Calcul	ator \$	Applications Anyone	Ordering Policy	FIFO 8
Queue Mappings			Effective Users		
Queue Mappings Override	C Disabled				
Versions					
	rw seconds ago	load			
-	sinutes ago	load			
1 ver	sion1	load			

# **3.2. Configuring Queues**

To configure a queue:

#### Steps

- 1. Click the queue name.
- 2. Set the following queue parameters:



## Note

Hold the cursor over a parameter name to display a description of the parameter.

#### Capacity

- Capacity The percentage of cluster resources available to the queue. For a sub-queue, the percentage of parent queue resources.
- Max Capacity The maximum percentage of cluster resources available to the queue. Setting this value tends to restrict elasticity, as the queue will be unable to utilize idle cluster resources beyond this setting.
- Enable Node Labels Select this check box to enable node labels for the queue.

#### Access Control and Status

- State Running is the default state. Setting this to Stopped lets you gracefully drain the queue of jobs (for example, before deleting a queue).
- Administer Queue Click **Custom** to restrict administration of the queue to specific users and groups.
- Submit Applications Click **Custom** to restrict the ability to run applications in the queue to specific users and groups.

#### Resources

- User Limit Factor The default value of "1" means that any single user in the queue can at maximum only occupy the queue's configured capacity. This prevents users in a single queue from monopolizing resources across all queues in a cluster. Setting the value to "2" would restrict the queue's users to twice the queue's configured capacity. Setting it to a value of 0.5 would restrict any user from using resources beyond half of the queue capacity.
- Minimum User Limit This property can be used to set the minimum percentage of resources allocated to each queue user. For example, to enable equal sharing of the queue capacity among five users, you would set this property to 20%.
- Maximum Applications This setting enables you to override the Scheduler Maximum Applications setting. The default setting is Inherited (no override).

- Maximum AM Resource This setting enables you to override the Scheduler Maximum AM Resource setting. The default setting is Inherited (no override).
- Priority An integer value that sets a relative priority for a queue. The default value is 0 for all queues. Setting this to a higher value gives a queue access to cluster resources ahead of queues with lower priorities. In order for YARN Queue Priorities to be applied, you must enable preemption. For more information see Setting YARN Queue Priorities.

The following image shows the example "Engineering" queue with these settings specified:

Ambari test_cluster 🚛 3 alurts		Dashboard Services Hosts Alerts Admin III 🔺 admin -
Add Queue     Actions -     mot (100%)	Engineering 2	×
default (0%) 🚺 🖉 🗸	Capacity	Level Total , 100%
🚥 🖛 Engineering (80%)	Engineering	C Enable node labels
e development (20%) 🖉 🗸	Capacity: 60 56	Max Capacity: 60 96
💶 qa (80%) 🥔 🗸		V Show Peer Level Queues
Marketing (30%)     Ø     ✓     Support (10%)     Ø     ✓	Access Control and Status	Resources
Scheduler 🗸	State Purning Stopped	User Limit Factor 1
achequier 🗸	Administer Queue Anyone Custom	Minimum User Limit 20 %
Maximum Applications 10000	Effective Arguna	Maximum Applications Inholted
Maximum AM Resource 20 56	Administraters	Maximum AM Resource Inherite 96
Node Locality Delay 40	Submit Anyone Custom Applications	Priority 0
Calculator Default Resource Calculator I	Effective Users	
Queue Mappings		
Gueue Mappings Disabled Override		
Versions		
st Correct 3 minutes ago load		
T days ago load		
7 days ago load     version1 load		

#### **More Information**

Enabling Preemption [80]

Setting YARN Queue Priorities [84]

Configuring Cluster Scheduler Settings [86]

# **3.3. Enabling Preemption**

#### **About This Task**

When using YARN queues, a scenario can occur in which a queue has a guaranteed level of cluster resources, but must wait to run applications because other queues are utilizing all of the available resources. If Preemption is enabled, higher priority applications do not have to wait because lower priority applications have taken up the available capacity. With Preemption enabled, under-served queues can begin to claim their allocated cluster resources almost immediately, without having to wait for other queues' applications to finish running.



#### Note

For more information about Preemption, see Better SLAS Via Resource Preemption in the YARN Capacity Scheduler.

#### Steps

1. On the Ambari dashboard, select **YARN > Configs**. Under YARN Features, click **Preemption**. The button label changes to indicate that Preemption is enabled.

🔬 Ambari test_clu	uster • 0 ops 7 olerts	Dashboard Services Hos	ts Alerts Admin 🌐 🚨 admin 🔻
O HDFS	Summary Heatmaps Configs	Quick Links+	Service Actions +
YARN     MapReduce2     Tez     Heve     Hese     Pig     Oozie     ZooKeeper	Group Default (1)   Manage Config C  Admin 7 days ago FCP-2.5  4 day  C  There is 1 configuration change in 1 service Si	admin rs ago HDP-2.6 HDP-2.6 VI ad HCP-2.6 HCP-2.6 HCP- tar 06, 2017 10:19	Filter
Bitorm     Ambari Infra     Ambari Infra     Ambari Metrics     Attas     Kafka     Knox     SmartSense     Sitter     Actions →	Settings Advanced  Memory Node  Memory allocated for all YARIN containers on a nor  Output O ME Device Memory 2070 ME 2000 ME 2070 ME	de Minimum Container Size (Memory)	YARN Features Node Labels Dentified Pre-emption Free/Med
	CPU Node CPU Scheduling Disabled CPU Isolation	Container Minimum Container Size (VCores)	

 Select the Advanced tab, then scroll down and select Custom yarn-site. Click Add Property and use the Add Property pop-up to add a new property in the following format:

yarn.resourcemanager.monitor.capacity.preemption.total\_preemption\_per\_round= <(memory-of-one-NodeManager)/(total-cluster-memory)>

This is the maximum percentage of resources preempted in a single round. You can use this value to restrict the pace at which Containers are reclaimed from the cluster. After computing the total desired preemption, the policy scales it back to this limit. This should be set to (memory-of-one-NodeManager)/(total-cluster-memory). For example, if one NodeManager has 32 GB, and the total cluster resource is 100 GB, the total\_preemption\_per\_round should set to 32/100 = 0.32. The default value is 0.1 (10%):

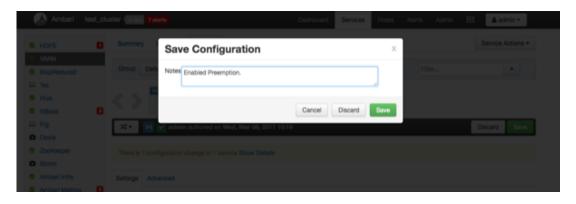
```
yarn.resourcemanager.monitor.capacity.preemption.total_preemption_per_round=
0.1
```

3. Click Add Property again and use the Add Property pop-up to add the following Custom yarn-site property:

yarn.resourcemanager.monitor.capacity.preemption.natural\_termination\_factor= 1.0

Similar to total\_preemption\_per\_round, you can apply this factor to slow down resource preemption after the preemption target is computed for each queue (for example, "give me 5 GB back from queue-A"). For example, if 5 GB is needed back, in the first cycle preemption takes back 1 GB (20% of 5GB), 0.8 GB (20% of the remaining 4 GB) in the next, 0.64 GB (20% of the remaining 3.2 GB) next, and so on. You can increase this value to speed up resource reclamation. The recommended value for this parameter is 1.0, meaning that 100% of the target capacity is preempted in a cycle.

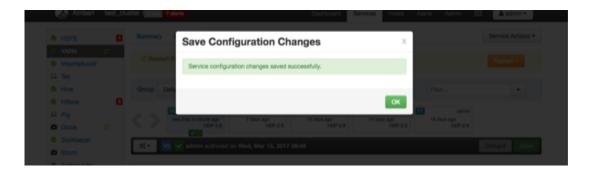
4. Click **Save** to save the new configuration settings. On the Save Configuration pop-up, type a description of the changes in the Notes box, then click **Save**.



5. On the Dependent Configurations pop-up, click **OK** to accept the recommended value of true for the yarn.scheduler.capacity.ordering-policy.priority-utilization.underutilized-preemption.enabled property.

	Conner States Transfer						
Dependent (	Configurations						2
Recommended	Changes						
	iguration changes, Ambari is recomm all checked configuration changes to				retain the Current Val	ue.	
Property		Service	Config Group	File Name	Current Value	Recommende	ed Value
	apacity.ordering-policy.priority-utiliz ad-preemption.enabled	YARN	Default	yam-site	false	true	
							Cancel OK
	Memory					YARN Features	

6. Click **OK** on the Save Configuration Changes pop-up.



7. Select **Restart > Restart All Affected** to restart the YARN service and load the new configuration.

🛛 🚕 Ambari test_clus	tor • O open 7 storts	Dashboard Services Ho	sts Alerts Admin 🎫 🔺 admin 🔹
O HDFS	Summary Heatmaps Configs	Quick Links+	Service Actions -
YARN     MapReduce2	C Restart Required: 4 Components on 1 Host		Restart -
Tez     Hive	Group Default (1)   Manage Config Group	ups	Pestart All Affected Filter, Nestart Nodewanagers
HBase   Pig   Oozie		o 14 days ago 14 days ago	dmin 19 admin 19 days ago 1-2.6 HDP-2.6
ZooKeeper     Storm	X • V5 V admin authored on Wed, Mar 1		Discard Save
Ambari Infra     Ambari Metrics	Settings Advanced		
<ul><li>Atlas</li><li>Kafka</li></ul>	Memory		YARN Features
C Knox C SmartSense	Node Memory allocated for all YARN containers on a node	Container Minimum Container Size (Memory)	Node Labels Disabled
Actions -	0 MB 2040 MB 7970 MB	0 MB 1500 MB 2072 MB	Pre-emption
		Maximum Container Size (Memory)	Enabled
		0 MB 1500 MB 5072 MB	
	CPU		
	Node CPU Scheduling	Container Minimum Container Size (VCores)	

8. Click **Confirm Restart All** on the confirmation pop-up to confirm the YARN restart.

O HDFS	Summary	Confirmation	Х		Service Actions *
	C Restart B	You are about to restart YARN			
	Group Defa	This will trigger alerts as the service is restarted. To suppress alerts, turn on Maintenance Mode for YARN prior to running restart all		Film.	•
O HBase 🔲	2.5			admin 19 days app	
Oozie     ZooKeeper		Cancel Confirm Restart A	a	109-24	
Storm     Amberi Infra	20 + 1/3				Discard Save

9. After YARN restarts, Preemption will be enabled. Other components may also require a restart.

# **3.4. Setting YARN Queue Priorities**

#### About This Task

Even with preemption enabled, there are some use cases where applications might not have access to cluster resources without setting priorities:

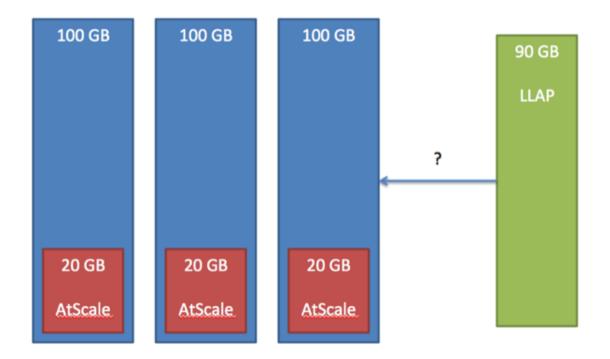
- Long-running applications Without setting priorities, long-running applications in queues that are under capacity and with lower relative resource usage may not release cluster resources until they finish running.
- Applications that require large containers The issue with long-running applications is exacerbated for applications that require large containers. With short-running applications, previous containers may eventually finish running and free cluster resources for applications with large containers. But with long-running services in the cluster, the large containers may never get sufficiently large resources on any nodes.
- Hive LLAP Hive LLAP (Low-Latency Analytical Processing) enables you to run Hive queries with low-latency in near real-time. To ensure low-latency, you should set the priority of the queue used for LLAP to a higher priority, especially if your cluster includes long-running applications.



## Note

To set the queue used for Hive LLAP, select **Hive > Config > Settings** on the Ambari dashboard, then select a queue using the **Interactive Query Queue** drop-down. For more information, see the Hive Performance Tuning guide.

For example, the following figure shows a 3-node cluster with long-running 20 GB containers. The LLAP daemons require 90 GB of cluster resources, but preemption does not occur because the available queues are under capacity with lower relative resource usage. With only 80 GB available on any of the nodes, LLAP must wait for the long-running applications to finish before it can access cluster resources.



#### Prerequisites



In order for YARN Queue Priorities to be applied, you must enable preemption.

#### Steps

1. On the YARN Queue Manager view, select a queue, then enter a priority in the **Priority** box under Resources. All queues are set to a priority of 0 by default. Higher numbers indicate higher priority.

🚕 Ambari test_e	oluster (Bases 1 der)			Dashboard Services Hosts Alerts Admin III 🔺 ad	imin •
+ Add Queue	C Ad	lions -	default 2 root.default		×
ee 🕳 default (80)	6)	•	Capacity	Level Total 100%	
Engineering     develo     ga (80	pment (20%)	*	default Capacity: 60 96	Max Capacity: 100 56	Jeis
Marketing (	10%)	- 1		V Show Peer Level Queues	
Support (10			Access Control and Status	Resources	
Scheduler		~	State Ranning Stopped	User Limit Factor	
			Administer Anyone Custom	Minimum User Limit 100 96	
Maximum Applications Maximum AM Resource	10000 [] 20 96		Effective Administrators	Maximum Applications Inherited Maximum AM Resource Inherite 96	
Node Locality Delay	40		Submit Anyone Custom	Priority 2	
Calculator Queue Mappings	Default Resource Calc	ulator 0	Effective Users Anyone	Ordering Policy I	_
Queue Mappings Override	O Disabled				
Versions					
vel Current	8 days ago	load			
vð.	15 days ago	load			
-	15 days ago	load			
with the	version1	load			

2. Select Actions > Save and Refresh Queues to save the priority setting.



# **3.5. Configuring Cluster Scheduler Settings**

You can use the Scheduler box to set global capacity scheduler settings that apply to all queues.

+ Add Queue Actio	n5 *		Engineering @		1
root (100%)		× .	root.Engineering		
defauit (60%)	.0	× .	Capacity	Level Total 100%	
Engineering (20%)		<b>~</b>	Engineering		C Enable node labe
<ul> <li>development (20%)</li> </ul>		× .	Cepecity: 20 %	Max Capacity: 60 %	-
qa (80%)	"	× .		Show Peer Level Queues	
<ul> <li>Marketing (10%)</li> </ul>	"	× .			
< Bupport (10%)	"	× .	Access Control and Status	Resources	
Scheduler		~	State Running Stopped	User Limit Factor	
			Administer Queue Anyone Custom	Minimum User Limit 20	96
Maximum Applications 10000			Effective	Maximum Applications Interte	d
Maximum AM Resource 20 %			Administrations	Maximum AM Resource Inhorito	96
Node Locality Delay 40			Submit Anyone Custom	Priority 0	
Calculator Default Resource Calcul	ator	•	Effective Users		
Gueue Mappings					
Gueue Mappings Override					
	-	-	•		

The following Scheduler global parameters are available:

- Maximum Applications To avoid system-thrash due to an unmanageable load caused either by malicious users, or accidentally the Capacity Scheduler enables you to place a static, configurable limit on the total number of concurrently active (both running and pending) applications at any one time. This property is used to set this limit, with a default value of 10,000.
- Maximum AM Resource The limit for running applications in any specific queue is a fraction of this total limit, proportional to its capacity. This is a hard limit, which means that once this limit is reached for a queue, any new applications submitted to that queue will be rejected, and clients will have to wait and retry later.
- Node Locality Delay The number of missed scheduling cycles after which the scheduler attempts to schedule rack-local containers.
- Calculator The method by which the scheduler calculates resource capacity across resource types.
- Queue Mappings You can use this box to specify default queue mapping based on user or group.
- Queue Mappings Override Select this box to enable override of default queue mappings.

# **3.6. Applying the Configuration Changes**

You can use the Actions menu to apply configuration changes made to the queue hierarchy.

Depending on the configuration changes made, the Actions menu will guide you to the options available to apply the changes.

For changes that are not valid and cannot be applied, the **Actions** button will turn red, and the menu will not appear.

+ Add Queue	C Actions -
root (100%)	~
	(A /
cheduler	~

For configuration changes that can be applied dynamically (without restarting the YARN ResourceManager), the Actions Menu will guide you to **Save and Refresh Queues**.

+ Add	d Queue C Actions -
re	0° Save and Restart ResourceManage
	Save and Refresh Queues
	Save Only
	Download config

For configuration changes that require a restart of the YARN ResourceManager, the Actions Menu will guide you to **Save and Restart ResourceManager**.

+ Add	Queue 🕫 Actions -
- rc	\$ Save and Restart ResourceManager
	Save Only
_	🛓 Download config

# **4. Using Files View**

**Files View** provides a convenient way to access files and folders in your cluster's file system, using a browser-style user interface. Ambari creates a Files View instance and one default Files View automatically during cluster deployment. The default Files View points to the HDFS file system.

You can also create a custom view from the Files View instance that points to an Amazon Web Services (AWS) S3 file system bucket.

To create a custom Files View to work with files in S3, provide the bucket URL, and valid, AWS credentials in the **Settings > View Configs** field when you create the view instance that accesses an S3 bucket. You must enter the AWS credentials as a single string containing no space characters using the following syntax:

fs.defaultFS=s3a://<bucketURL>/;fs.s3a.access.key=<validaccesskeystring>;fs.s3a.secret.key<validsecretkeys

\_\_\_

Files View supports the following tasks:

- Moving Files/ Folders within your file system.
- Copying Files/Folders within your file system.
- Uploading files from a local system
- Modifying permissions of files and folders

#### **More Information**

Creating View Instances [10]

**About Access Keys** 

# **5. Using Falcon View**

Apache Falcon solves enterprise challenges related to Hadoop data replication, business continuity, and lineage tracing by deploying a framework for data management and processing. The Falcon framework can also leverage other HDP components, such as Apache Pig, Apache Hadoop Distributed File System (HDFS), Apache Sqoop, Apache Hive, Apache Spark, and Apache Oozie. Falcon enables this simplified management by providing a framework to define and manage backup, replication, and data transfer.

Hadoop administrators can use the **Falcon View** to centrally define, schedule, and monitor data management policies. **Falcon** uses those definitions to auto-generate workflows in Apache Oozie.

#### **More Information**

Data Movement and Integration

# 6. Using Hive View 2.0



## Important

This chapter focuses on Hive View 2.0. For information about how to use earlier versions of Hive View, see Using the Hive View in version 2.4.2 of the HDP Apache Ambari Views Guide.

Hive is a data warehouse infrastructure built on top of Hadoop. It provides tools to enable data ETL, a mechanism to put structures on the data, and the capability to query and analyze large data sets that are stored in Hadoop. **Hive View** is designed to help you author, optimize, and execute Hive queries.

Use Hive View to:

- Browse databases
- Write queries or browse query results in full-screen mode, which can be particularly helpful with complex queries or large query results
- Manage query execution jobs and history
- View existing databases, tables, and their statistics
- Create tables and export table DDL to source control
- View visual explain plans



# Tip

Consider using the Tez View of Ambari as a complement to Hive View if you are using the Tez execution engine. The Tez View helps you debug Hive queries by displaying performance metrics and envisioning the Directed Acyclic Graph (DAG) process at runtime.

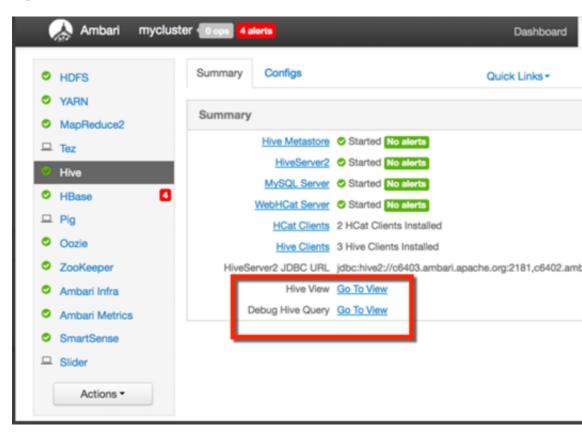
There are two simple ways to access either a Hive View or the Tez View in Ambari:

• Click the views tile in the upper right corner, and select the Hive View or Tez View that you want to use:



• Alternatively, you can click a **Hive View** hyperlink or the **Debug Hive Query** hyperlink (which opens the Tez View) on the **Summary** tab when the Hive Service is selected in Ambari:

#### Figure 6.2. Links to Hive-Related Views in Ambari



#### **More Information**

Query Tab [93]

Jobs Tab [99]

Tables Tab [100]

Saved Queries Tab [105]

UDFs Tab [105]

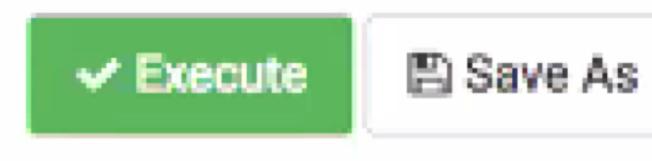
Settings Tab [106]

# 6.1. Query Tab

When you open Hive View 2.0, you first see the **Query** tab. The **Query** tab helps you write, plan, execute, and save queries with embedded tools for locating Hive databases and tables of your system in the same window.









#### Important Features of the Query Tab

- Use the **Browse** button to find databases. Click on a database to enable it as a selectabe database in Hive View. The **Browse** button acts as a filter in case you are working with more databases than you want to work with in Hive View.
- Click one or more databases in the database and tables pane (*not* in the **DATABASE** field) to designate active databases. All queries in the current tab are then run against the active database or databases.
- The database and tables pane displays all the tables of the active databases in a single view.
- The Query Editor field and an active **RESULTS** tab can be toggled between compact and full-screen mode. The screenshot below shows the compact mode. Full-screen mode enlarges the query editor field, which can help you author large, complex SQL queries or browse large query results.
- Use multiple Worksheet tabs to work on and analyze queries in parallel.

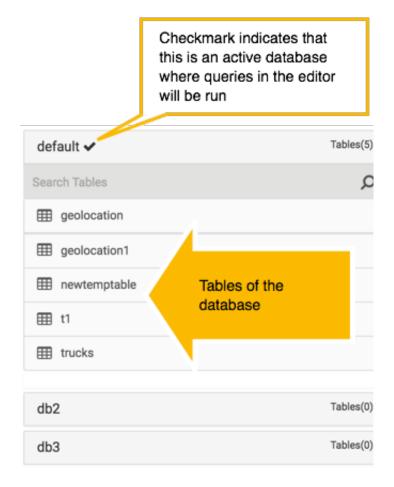
#### Query Editor and Database and Table Pane

Use the **Browse** button in the Query Editor to find and select databases. After you select databases and make them active in the view, they appear in the database and table pane.

# Figure 6.4. Query Editor

Worksheet1 DATABASE Select or searc	ch database/schema	Click on + to new Works	and the second s		 	
× 🛢 defaul	t × S db2 × S	db3				
	atabases sele om Browse bu	and the second se				
				=		
✓ Execute	🖺 Save As	Insert UDF 🗸 🥱	o Visual Explain			
RESULTS	≣ LOG	% VISUAL EX	PLAIN 🥠	TEZ UI		

### Figure 6.5. Database and Table Pane



#### How to Use the Query Editor

- Enter queries in the blank field. To run a specific query, highlight it, and click **Execute**. All queries contained in a Worksheet tab execute sequentially, and they run in the same session.
- Click the **Results** tab to view the results after a query is exeucuted.
- Similarly, click the Logs tab to view the log from an executed query.
- When the first query is executed in a Worksheet, a Tez session is started. Click the **Tez UI** tab to see details about the DAG execution.
- Click Save as to save your query.
- Double-click the **Worksheet** tab to rename the query, click **OK**, and then **Save as** to save the query with the new name.
- Click on the + symbol to open a new worksheet tab. Queries executed from the new worksheet tab will execute in a different session. Queries from different worksheets can execute in parallel.

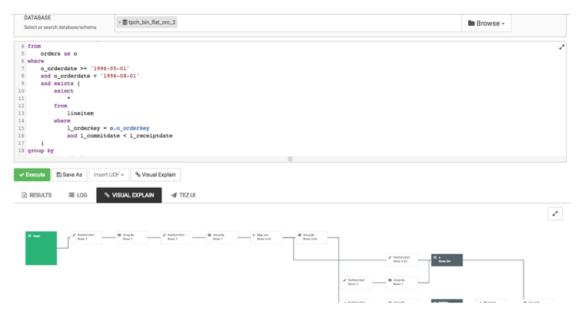
- Click the double arrow icon in the upper right corner of the Query Editor to expand the Worksheet area and cover Database Explorer. Click the icon again to collapse the Worksheet and make Database Explorer available again.
- Click the icon at the bottom of the Worksheet window and drag it down to expand the authoring space.

#### Visual Explain Plan

Hive View 2.0 includes a simple, graphical visualization of the Hive execution plan that focuses on key information that can help you spot expensive operations in your query.

Clicking the Visual Explain tab launches a visual explain plan.

#### Figure 6.6. SQL in Query Editor with Resulting Visual Explan Plan



The visual explain plan illustrates the execution of the query, regardless of whether you run the query or not. So you can use the Visual Explain feature truly as a planning tool.



#### Important

The workflow sequence of the visual explain plan is plotted beginning on the right side and proceeding to the left.

The nodes of the explain plan contain kernels of information that help you forecast or troubleshoot execution problems. Key information of the plan includes:

- *Execution "cost"*: The number of processed rows for each node of the visual explain plan is displayed, which gauges the resource demands of the various runtime processes of an executing query.
- Labeling of runtime processes for easy comparison: Each node is identified by type of operation or process. For example, you can see which nodes represent "cheap" Map Join activity and compare them to "expensive" Partition or Sort operations.

Tip

If your query performs Partition or Sort operations on a large number of records, the query runtime is relatively long. In this case, consider tuning or rewriting the query.

Click on a node to view detailed information about SQL operators:

## Figure 6.7. Details of a Map Join Node



#### Debugging Hive Query Execution Using the Tez UI

Query execution can be debugged using the embedded Tez UI. To access the Tez UI, click the **TEZ UI** tab at the bottom of the Query Editor window.



## Important

Clicking the Tez UI in Hive View instantiates an Ambari Tez View for the current query. Therefore, for more information about how to use the Tez UI, see Using Tez View.

# 6.2. Jobs Tab

You can view the history of all queries that the current user executed in the Hive View instance on the **Jobs** tab. In addition, activity in the Hive View that does not execute queries is logged, such as generating a visual explain plan.

The **History** tab of Hive View in Ambari 2.4.2 and earlier versions was a predecessor to the **Jobs** tab. The functionality of the **Jobs** tab is similar to the **History** tab. The main differences are:

• The **Stop execution** button that was on the **History** tab is not on the **Jobs** tab. Instead, you can stop an executing query by clicking the **Stop** button on the **Query** tab.

• The **History** tab displayed *all* queries that were run by the current user on the databases selected in Hive View, even if the user executed the queries outside Hive View (such as in a JDBC or ODBC client). The **Jobs** tab only shows activity of the current user in Hive View.

#### Figure 6.8. Jobs Tab of Hive View 2.0

HIVE					+ NEW JOB +	NEW TABLE		
A QUERY	A JOBS III TABLES 🗞 SAV	A NOTIFICATIONS						
6 ALL	6 SUCCEEDED			Search title	Mar 8, 2017 - Mar 15, 2017			
Job Id	Title	status	Start time		Duration	Action		
56	LogisticsFW	SUCCEEDED	20 hours ago		26	12		
select count(*) from trucks t join geolocation g on (g.driverid = t.driverid and g.truckid = t.truckid);								
55	Worksheet1	SUCCEEDED	a day ago		2	10		
54	Worksheet1	SUCCEEDED	a day ago		0	10		
53	Worksheet1	SUCCEEDED	a day ago		0	2		
select * from trucks limit 5;								

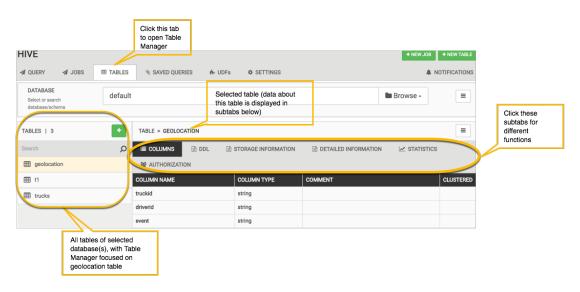
Note the following details about the way the UI operates:

- The **Title** column displays query names. If no name was assigned to the query, the worksheet number appears.
- Click the icon in the **Action** column to toggle between compact and expanded view of each job. Expanding a job on this tab shows the SQL statement for the action. For example, in the Jobs Tab screenshot above, Job 56 and Job 53 are expanded.

# 6.3. Tables Tab

Click the **Tables** tab to access the Table Manager. The Table Manager is one central place to view, create, delete, and manage tables of whichever databases that you select after clicking the **Browse** button. By default, the Table Manager opens with the **Columns** subtab enabled.

#### Figure 6.9. Table Manager



#### **Gathering Information about a Table**

In the **DDL**, **Storage Information**, and **Detailed Information** subtabs, you can view properties, storage, and other information of the particular table selected in the left-side pane.

#### HIVE + NEW JOB + NEW TABLE A QUERY I TABLES A NOTIFICATIONS A JOBS #- UDFs SETTINGS DATABASE default Browse -≡ Select or se TABLE > GEOLOCATION ≡ TABLES | 3 + COLUMNS DDL STORAGE INFORMATION 1 CREATE TABLE 'geolocation'( 2 'truckid' string, 1 'driverid' string, ρ DETAILED INFORMATION MAUTHORIZATION geolocation `driverid` string, `event` string, `latitude` string, `longitude` string, `city` string, `state` string, ⊞ t1 III trucks Tatate string, 'velocity' string, 'diling\_ind' string) 'arg.apache.hadoop.hive.ql.io.orc.OrcSerde' | I STORED AS INPUTCOMAT 'org.apache.hadoop.hive.ql.io.orc.OrcInputFormat'

#### Figure 6.10. Example of Information in the DDL Subtab

#### Figure 6.11. Example of Storage Information Subtab

TABLES   3	+	TABLE > GEOLOCA	ILE > GEOLOCATION								
Search	Q	I COLUMNS	DDL.	STORA	3E INFORMATION	DETAILED INFORMATION	M STATISTICS	볼 AUTHORIZATION			
geolocation		INFORMATION			VALUE						
⊞ t1		SerDe Library	De Library org.apache.hadoop.hive.ql.io.orc.OrcSerde								
I trucks		Input Format			org.apache.hadoop.hive.ql.io.orc.OrcInputFormat						
		Output Format			org.apache.hadoop.hive.ql.io.orc.OrcOutputFormat						
		Compressed			No						
		Number of Buckets			-1						
		Bucket Columns	ucket Columns								
		Sort Columns	ort Columns								
		Daramatere									

#### **Figure 6.12. Example of Detailed Information Subtab**

TABLES   3	+	TABLE > GEOLOC	SEOLOCATION								
Search	Q	E COLUMNS	DDL.	STORAGE INFORMATION		M STATISTICS	출 AUTHORIZATION				
geolocation		INFORMATION		VALUE							
⊞ t1		Database Name		default							
I trucks		Owner		admin							
		Create Time		Thu Jan 12 09:58:31 UTC 2017							
		Last Access Time		UNKNOWN							
		Retention		0							
		Table Type		MANAGED_TABLE							
		Location	.ocation hdfs://dipayan-hive-test-1.novalocal:8020/apps/hive/warehouse/geolocation								

#### **Running and Viewing Statistics of a Table**

To assess how big a table is and to estimate the execution duration of queries that run on the table, you might find the **Statistics** subtab to be helpful. The subtab also calculates statistics on each column of the table if you want to gather this information.

If statistics do not appear on this subtab, click the **Compute** button. The button changes to **Recompute** after you first calculate statistics. You might click **Recompute** if you want to change whether or not column information is included or if you change the table in some way after the initial calculation of statistics.

In the following screenshot, **include columns** must be selected and the **Recompute** button clicked to display column statistics.

#### Figure 6.13. Example of Statistics Subtab

HIVE		+ NEW JOB + NEW TABLE
A QUERY A JOBS III TABLES	SAVED QUERIES A- UDFs Ø SETTINGS	A NOTIFICATIONS
DATABASE Select or search database/schema		Browse -
TABLES   3	TABLE > GEOLOCATION	=
Search D	E COLUMNS DDL E STORAGE INFORMATION DETAIL	LED INFORMATION
geolocation	STATISTICS	✓ Recompute include columns
Ⅲ t1	STATISTICS	
I trucks	TABLE STATISTICS	
	STATS NAME	VALUE
	Number of Files	1
	Number of Rows	8000
	Raw Data Size	7112000
	Total Size	43039

### **6.3.1. Creating Tables**

You can add a table to a database by either clicking the + symbol or the **NEW TABLE** button.

If you create a table, the Table Manager lets you build the table column by column and helps you use advanced features like ACID, partitioning, and bucketing. The form that appears is dynamic. For example, if you designate the DATA TYPE for a column as **DECIMAL**, the SIZE fields show two editable fields that are appropriate for a decimal data type (Precision and Scale).

Depending on the column data type that you choose, click the configuration wheel in the ADVANCED column of the form to select whether the column is Partitioned or Clustered. If you select Partitioned, the Table Manager adds a **Partitions** tab in the Table Manager. The **Partitions** tab helps provide an overview of all partitioned columns in the table.



#### Tip

Notice that the Create Table mode also includes the **ADVANCED** and **TABLE PROPERTIES** subtabs, which enable you to further customize the new table. The **ADVANCED** subtab lets you make a table transactional (that is, enable ACID properties), change the default location of the Hive table in HDFS, change the file format from the default ORC format. If you select the TEXTFILE format, you can also configure the row format (for example, separator values, null values, escape values). Figure 6.14. Example of Creating a Table Form

IVE						+ NEW	JOB + NEW TAB
QUERY 刘 JOBS 🌐 TABL	LES % S	AVED QUERIES n- UDFs	SETTINGS				A NOTIFICATIO
DATABASE Select or search database/schema	tpch_bin_f	flat_orc_2				Browse -	≡
ABLES   8	•	TABLE > CREATE TABLE					LUPLOAD TABLE
earch	Q						
		Name outomer contex	*				
I customer		Name customer_contac	4				
		Name customer_contact					
customer					SZE	ADVANCED	ACTION
customer     lineitem		COLUMNS ADVANC	ED TABLE PROPERTIES	•	SIZE		ACTION × Delete
田 customer 田 lineitem 田 nation		COLUMNS ADVANC	ED TABLE PROPERTIES		SIZE 20		
customer cu		III COLUMNS () ADVANC COLUMN NAME Id firstname	ED TABLE PROPERTIES	•	20	•	× Delete × Delete
customer  custo		III COLUMNS ADVANC	ED TABLE PROPERTIES	-		٥	× Delete

### 6.3.2. Uploading Tables

To access the Upload Table functionality, click the + symbol or the **NEW TABLE** button.

Ç

Tip

With Hive View 2.0, you can change more properties of an existing table while performing the Upload Table operation. Earlier versions of Hive View only let you change column names and data types while uploading a table.

In the Upload Table window, you can upload files which contain the rows of an Apache Hive table. The Upload Table command supports various input file formats. On uploading, it creates a new Hive table with the data.

#### **Input File Formats:**

CSV, XML, and JSON files are supported for input.

CSV

Supported types are:

- CSV with custom field delimiter (default is comma,)
- Quote character (default is double quote ")
- Escape character (default is backslash \)

The row delimiter must be \n or \r or \r\n. If **Is first row header?** option is selected, then first row of the file is treated as column names. During preview this can be changed by clearing this field but other delimiters should not be changed during the preview. The number of columns in the table and their order is defined by the first line of the file, irrespective of whether it represents column names or not. If there are extra columns in line 2 onwards, they are ignored. If there are lesser columns in line 2 onwards, then the rest of the columns are treated as null values.

#### XML

The format of the contents in the XML file should be as shown below:

1	10	h	le	1
-	ιa	S	10	1

	<row></row>
	<col name="col1Name"/> row1-col1-Data
	<col name="col2Name"/> row1-col2-Data
	<col name="col3Name"/> row1-col3-Data
	<col name="col4Name"/> row1-col4-Data
<row></row>	
	<col name="col1Name"/> row2-col1-Data
	<col name="col2Name"/> row2-col2-Data
	<col name="col3Name"/> row2-col3-Data
	<col name="col4Name"/> row2-col4-Data

<

The root tag must be . Inside there can be any number of <row> tags representing one row of the table. Inside each <row> tag there can be any number of <col> tags representing columns of the row. Each <col> tag must have a "name" attribute, which will be treated as the name of column. Column values should be within the <col> tag. The names, number and order of columns are decided by the first <row> entry. The names of column and datatypes can be changed during the Preview.

#### JSON

```
[ { "collName" : "value-1-1", "col2Name" : "value-1-2"}, { "collName" :
"value-2-1", "col2Name" : "value-2-2"}]
```

The file should contain a valid JSON array containing any number of JSON objects. Each JSON object should contain column names as property and column values as property values. The names, number and order of columns in the table are decided from the first object of the JSON file. The names and datatype of column can be edited during the preview step. If some JSON objects have extra properties then they are ignored. If they do not have some of the properties then null values are assumed. Note that the extension of files cannot be ".json"

To import a file into Hive View:

#### Steps

- 1. Select the input file format file type by specifying CSV, XML, or JSON.
- 2. If the File Type is CSV, you can select the **Field Delimiter**, the **Escape Character**, and the **Quote Character** values in the drop-down menus. Also, you can click the **Is first row** header? box if you want to enable this feature.

- a. If Stored as is TEXTFILE, then a gear next to it is enabled and you can click it to select **Fields Terminated By**, and **Escape By** to be used in creation of the Hive table.
- b. If Stored as is NOT TEXTFILE, another option **Contains endlines?** is enabled. If the column values in your file contain endline characters, ("\n" newline, ASCII 10 or "\r" carriage return, ASCII 13) then you must check this field for proper handling otherwise unexpected results might occur. Endline characters are not supported in TEXTFILE format.
- 3. Expand the Select File Source section to pick the table to upload. Click **Upload from HDFS** or **Upload from Local**.
- 4. If you clicked **Upload from Local**, you can choose the file from your local machine. Otherwise, enter the full HDFS path and click **Preview**. The file is partially read from client's browser or HDFS and the preview is generated with a suggested table name, column names, column data types and 10 rows from the data file.
- 5. (Optional) Change the suggested table name, column names, column data types, and many other table DDL and properties in the **Columns**, **Advanced**, and **Table Properties** subtabs.
- 6. Click the **Create** table. The actual table and temporary table (stored as TEXTFILE) are created. After this the data rows from the file are inserted into the temporary table followed by insertion from temporary table to actual table.
- 7. On success the temporary table is deleted and workflow completes.

In case of failure, an error is reported and the temporary table and actual tables are deleted. You can see the error message by clicking the message icon at the top right. Clicking again on the message icon brings back the Upload Table page. You can perform any changes required and click **Upload** again to upload the same file or restart the process by selecting a different file.

## 6.4. Saved Queries Tab

The Saved Queries tab shows all the queries that have been saved by the current user. Click the gear icon to the right of the query list to view the history of a query or to delete it:

#### Figure 6.15. Saved Queries Tab

HIVE								+ NEW JOB + NEW TABLE
A QUERY	A JOBS	I TABLES	⊗ SAVED QUERIES	∯- UDFs	SETTINGS			A NOTIFICATIONS
Preview			Title			Database	Owner	Action
select count(	*) from trucks t j	oin geoloca 🔖	Logist	ticsFW		default	admin	0

## 6.5. UDFs Tab

User-defined functions (UDFs) can be added to queries by pointing to a JAR file on HDFS and indicating the Java classpath, which contains the UDF definition. After the UDF is

added here, an **Insert UDF** button appears in the Query Editor that enables you to add the UDF to your query:

#### Figure 6.16. UDF Tab

HIVE			+ NEW JOB + NEW TABLE
A QUERY A JOBS	I TABLES & SAVED QUERIES	+ UDFs • SETTINGS	A NOTIFICATIONS
UDF Name	UDF Class Name	Owner	+ NEW UDF Action
UDF Name	UDF Class Name	File Resource	
UDF Ngame	UDF Class Name	File Resource	v Add UDF Cancel

## 6.6. Settings Tab

Use the **Settings** tab to append settings to queries that you execute in Hive View.

#### Figure 6.17. Settings Tab with Example Key and Value for One Property

HIVE						+ NEW JOB + NEW TABLE
A QUERY	A JOBS	I TABLES	ℜ SAVED QUERIES	nh- UDFs	SETTINGS	A NOTIFICATIONS
KEY					VALUE	ACTIONS
hive.execution	n.engine				tez	C2 Edit 🔮 Delete
				•		🖺 update 🛛 🗙 cancel
					+ A/	dd New

## 7. Using Pig View

Apache Pig is a scripting platform for processing and analyzing large data sets. Pig was designed to perform extract-transform-load (ETL) operations, raw data research, and iterative data processing. **Pig View** provides a web-based interface to compose, edit, and submit Pig scripts, download results, and view logs and the history of job submissions.

Use Pig View to:

- Write Pig scripts
- Execute Pig scripts
- Add user-defined functions (UDFs) to Pig scripts
- View the history of all Pig scripts run by the current user

### 7.1. Writing Pig Scripts

Navigate to the Pig View instance Scripts page, and click **New Script** in the upper right corner of the window. Name the script in the New Script dialog box, click **Create**, and enter your script into the editor. After you have written the script, you can use the execute button on the upper right to run it. Check the box that is adjacent to the execute button to use Tez instead of the default MapReduce engine.

The following figure shows a running Pig script:

#### Figure 7.1. Pig Script Running in Pig View

Ambari M	AyCluster (Dage Daterts			Dashboard	Services	Hosts Ale	rts Admin	-	▲ ambari-qa *			
B Pig_ETL		koript History										
🖉 🗈 Sere	F	Pig_ETL_1 🥒					O Execut	te on Tez	Execute +			
O (100)	/ P	Ghelper+ UDF helper+			/user/a	mbari-qa/pig/	scripts/piget	11-2015-0	6-15_02-55.pig			
e Deve		<pre>1 batting = load 'batting.cav' using PigStorage(','); 2 runs = FOREACE batting GENERATE 60 as playerDD, 61 as year, 68 as runs; 3 grp_data = GOREACE grp_data GENERATE group as grp_HAX(runs.runs) as max_runs; 4 max_runs = FOREACE grp_data GENERATE group as grp_HAX(runs.runs) as max_runs; 5 join_max_run = JOIN max_runs by (50, max_runs), runs by (year,runs); 6 join_data = FOREACE join_max_run GENERATE 50 as year, 52 as playerID, 51 as runs; 7 dump join_data;</pre>										
	An	guments			-							
	1	his pig script has no argu	ments defined.				Pig argume	nt	+ Add			

## **7.2. Viewing Pig Script Execution History**

The History tab shows the history of Pig scripts run by the current user. A particular script in history can be clicked to open it in a new Script tab to view its details:

#### Figure 7.2. Pig View Script History Tab

- 🔬 A	mbari MyCluster (Doos D	alerts		Dashboard	Services	Hosts	Alerts	Admin	ш	4	mbari-qa	•
Ø	Scripts	History										
se	UDFs	Date	Script	Status		Duration		Actions				
0	History	2015-06-15 08:00	PIg_ETL_1	RUNNING				8 Delete				
							Sł	10w: 10	• 1	- 1 of 1	*	÷

## 7.3. User-Defined Functions (UDFs) Tab

UDFs can be added to Pig scripts by clicking **Create UDF** in the upper right corner of the UDFs window. In the Create UDF dialog box, point to a UDF in the system by specifying the name and path:

#### Figure 7.3. Pig View UDFs Tab

Ambari MyCluster 🛛 🚥 🛛	alerta	Dashboard	Services	Hosts	Alerts	Admin	ш	🛦 ambari-qa •
🚯 Scripts	UDFs							+ Create UDF
🗲 UDFs	Name	Path		Owner				
History	No UDFs to display							

# 8. Using Slider View

Slider is a framework for deploying and managing long-running applications on YARN. When applications are packaged using Slider for YARN, **Slider View** can be used to help deploy and manage those applications from Ambari.



#### Important

This view has been marked deprecated.

## 9. Using SmartSense View

SmartSense View allows Hortonworks support subscription customers to capture diagnostic data for two purposes:

- To receive recommendations on performance, security, and operational changes based on your server hardware, HDP services deployed, and your use cases.
- To quickly capture diagnostic information about services and hosts when working with support to troubleshoot a support case.

You use SmartSense View to do the following tasks:

- Capture a bundle
- Set a bundle capture schedule
- View and download captured bundles
- View SmartSense recommendations

## **10. Using Storm View**

Storm provides a real-time, scalable, and distributed solution for data streamed from realtime sources such as machine sensors, supporting data ingestion, processing, and real-time response. Typical use cases include automated systems that respond to sensor data by notifying support staff, or an application that places a proximity-based advertisement on a consumer's smart phone.

Use Storm View for the following types of operations:

- Monitor Storm cluster status and review configuration settings.
- Monitor Storm topologies, review configuration settings, perform topology actions such as Activate, Deactivate, and Kill, and perform topology rebalancing to increase worker JVMs and component parallelism.
- Access component metrics, debug logs, and jstack outputs; debug and profile worker JVMs.



#### Important

This view has been marked deprecated.

# **10.1. Monitoring Storm Cluster Status: the Cluster Summary Page**

The landing page for Storm view shows current cluster status and nimbus configuration.

It shows the available nimbus host(s), and for a nimbus HA, denotes which host is a leader. It also shows all available supervisor hosts and currently deployed topologies. Here is an example landing page:

EXECUTOR	TASKS		Topology Listing				
	<b>2</b> 8	28	Topology Name	50	eus.	Uptie	ve .
SUPERVISOR	51	ors	wordcount		TIME:	52m 2	
$\bigcirc$	1		Supervisor Summary				(
(100%)	7	5%	Host	Slots	CPU	Memory	Uptime
$\bigcirc$			o6601.ambarl.apache.org	200 %	(0%)	54 N	1h 10m 7s
Nimbus Summary		G	c6602.ambari.apache.org				1h 9m 18s
Host:Port	Status	Uptime		50%	(0×)	(27%)	
:6602.ambari.apache.org;6627	Lander	1h 10m 38s					

Apache Storm - v1.0.1.2.5.0.0-733

The lower left section of the summary page shows resource utilization of supervisors:

Supervisor Summary				ø
Host	Slots	CPU	Memory	Uptime
c6601.ambari.apache.org	100 %	0%	54%	1h 10m 7s
c6602.ambari.apache.org	50%	0%	27%	1h 9m 18s

The upper right section shows the current status of the deployed topology:

Topology Listing		ø
Topology Name	Status	Uptime
wordcount	ACTIVE	31s

Click on the "Nimbus Configuration" section to list Storm configuration settings:

Nimbus Configuration	0
Search By Key	Q.
Key	Value
client.jartransformer.class	org.apache.storm.hack.StormShadeTransformer
drpc.invocations.port	3773
logviewer.max.per.worker.logs.size.mb	2048
nimbus.blobstore.class	org.apache.storm.blobstore.LocalFsBlobStore
nimbus.childopts	-Xmx1024m-javaagent./usr/hdp/current/storm-nimbus/contrib/storm-jmxetric/lb/jmxetric- 1.0.4.jar=host=localhost.port=8649.wireformat31x=true_mode=multicast.config=./usr/hdp/current/storm-nimbus/contrib/storm- jmxetric/conf/jmxetric-conf.xml.process=Nimbus_JVM
resource.aware.scheduler.eviction.strategy	org.apache.storm.scheduler.resource.strategies.eviction.DefaultEvictionStrategy
scheduler.display.resource	false
storm.cluster.mode	distributed
storm.group.mapping.service	org.apache.storm.security.auth.ShellBasedGroupsMapping
storm.messaging.netty.client_worker_threads	1
storm.messaging.netty.server_worker_threads	1
storm.thrift.transport	org.apache.storm.security.auth.SimpleTransportPlugin
supervisor.localizer.cache.target.size.mb	10240
superv/sor.run.worker.as.user	false
topologybuiltin.metrics.bucket.size.secs	60
topology.max.error.report.per.interval	5

# **10.2. Monitoring Topology Status: the Topology Summary Page**

The topology summary page contains metrics and directed acyclic graphs (DAG) that show deployed topology components and topology debugging features.

You can select which window for which to review metrics. By default the view will show metrics for "All Time."

OLOGY SUMMARY	TOPOLOGY ST	TATS				
ID: wordcount-2-1466189929	Window	Emitted	Transferred	Complete Latency (ms)	Acked	Falled
Owners storm					PECKEN	1 8000
Status: ACTIVE	10m Os	65240	35000	0		
Uptime: 3m 27s	3h Om Os	65240	35000	0		
Workers: 3	5d Oh Om Os	65240	35000	0		
Executors: 28 Tasks: 28						
Memory: 2496	All time	65240	35000	0		
rdcount						
<u>ج</u>		4				

On the right side above panel, there are several topology actions buttons. These buttons allow you to perform several actions: Activate (highlights when topology status is deactivated), Deactivate, Rebalance, Kill, and Changing log level.

Window All time	System Summary	OFF	Debug	OFF	▶ ■ ⊕ ● D
-----------------	----------------	-----	-------	-----	-----------

#### Rebalancing a Topology

To adjust the number of workers for the topology and the parallelism of each component in the topology, use the rebalance button.

	Rebalance Topolo	gy			×
-	Workers*:	0	• 4		
Y	spout*:	5		1	
it-2-14661899	count*:	12		1	r (ms)
	split*:	8		1	
	Wait Time*:	30		1	
				Close Save	
	_	_		Close Save	

#### Changing the Logging Level of a Running Topology

This feature facilitates topology debugging, by allowing you to temporarily enable debug log level and see any issues in a topology.

To use this feature, edit the Logger to update the class name for which you would like to add a log level.

Modify the logger levels for topology. Note that applying a setting	Change Log Level restarts the timer in the workers. To	o configure the root logger, us	e the name ROOT.	
Logger	Level	Timeout	Expires At	Action
comyour organization LoggerName	ALL	- 30		

For example, if you would like to see debug logs in the count bolt of the sample word count topology supplied with Storm, add the classname as org.apache.storm.starter.WordCount.

#### Sampling Events in a Running Topology

This feature allows users to debug and see the events that are flowing through the topology, essentially sampling events from a running topology and storing them in a log file.

To use this feature, turn the Debug switch to "On":

Window	Alttime	•	System Summary	OFF	Debug	OFF		Þ = 4	0 0
		- 0					· ·		

	you really want to debug pling percentage.	g this topology	? If yes, pleas	e, specify ×			• D
TOPOLOGY SUMMARY							
ID: wordcount-2-14661899				No Yes	(ma)	Acked	Failed
Owner: storm							
Status: ACTIVE	104105	411209	20072				
Uptime: 25m4s	3h Om Os	889920	477240	0			
Workers: 3							
Executors: 28	1d Oh Om Os	889920	477240	0			

The event logger will sample the given percentage of incoming tuples and write them to the log for users to see the incoming tuples at each stage of topology. We recommend that you not set this to a higher percentage, because it can fill up the logs on disk very quickly.

# **10.3. Looking Up Configuration Values: the Component Summary Page**

On the Component Summary page, you can drill down to a individual component in a topology to see relevant stats for the component and access debug logs and jstack outputs.

Window	All time		System	Summary	OFF	Þ	ebug OFF			•
OMPON	ENT SUMM	ARY		SPOUT STAT	5					
	ID: spout			Window	Emitted	Transferred	Complete Latency	(ms)	Acked	Failed
	agy: wordcov	nt –		3h Om Os	94360	94360	0,000		0	0
Decut				1d Oh Om Os	94360	94360	0,000		0	0
	ska: S			All time	94360	94360	0,000		0	0
	we events			10m 0s	29806	29806	0,000		0	0
	ats ( All tir	ne)	_							
Search by 1			۹							
Gream	Er	vitted	Transferred		Complete Lat	ency (ms)		Acked	Falle	N
efault	94	360	94360		0,000			0	0	
ecutor !	Stats ( All t	ime )								
secutor : Search by i		ime )	٩							

You can also debug and profile a worker JVM, by choosing the rightmost button on the Component Summary Page:

Window All time • System Summary OFF Debug OFF

The popup window shows all worker processes running the particular spout. You can select the worker processes to take the jstack output or Heap dump, and selectively restart a worker JVM.

		- 8
Host:Port	Executor Id	ms
c6601.ambari.apache.org:6700	[25-25], [28-28]	
c6602.ambari.apache.org:6700	[26-26]	
c6601.ambari.apache.org:6701	[27-27], [24-24]	

## **11. Using Tez View**

Tez is a framework for building high performance batch and interactive data processing applications. Apache Hive and Pig use the Tez framework. When you run a job such as a Hive query or Pig script using Tez, you can use Tez View to track and debug the execution of that job.

Tez provides a framework that enables human-interactive response times with Apache Hive queries and Apache Pig data transformations. Tez View helps you find specific Hive queries and their performance metrics, even in environments where there are hundreds or more queries running daily, when you need to troubleshoot or tune data analytic applications. Both sortable metrics and graphic visualizations enable you to understand and debug submitted Tez jobs, such as Hive queries or Pig scripts, that are executed using the Tez execution engine.

#### To open the Tez View:

- 1. Open Ambari.
- 2. Click the views tile in the upper right corner of the window:

#### Figure 11.1. Views Menu of Ambari



3. Select Tez View.

The following sections describe using Tez View to manage Hive and Pig tasks:

- Understanding Directed Acyclic Graphs (DAGs), Vertices, and Tasks [118]
- Searching and Identifying Hive Queries [118]
- Identifying the Tez DAG for Your Job [122]
- Understanding How Your Tez Job Is Executed [124]
- Identifying Causes of Failed Jobs [125]

- Viewing All Failed Tasks [126]
- Using Counters to Identify the Cause of Slow-Performing Jobs [127]

# 11.1. Understanding Directed Acyclic Graphs (DAGs), Vertices, and Tasks

To explain DAGs, vertices, and tasks, consider how Hive SQL queries are compiled and converted into a Tez execution graph also known as a DAG. A *DAG* is a collection of vertices where each vertex executes a fragment of the query or script. Directed connections between vertices determine the order in which they are executed. For example, the vertex to read a table must be run before a filter can be applied to the rows of that table.

As another example, consider when a vertex reads a user table. This table can be very large and distributed across multiple computers and multiple racks. Reading the table is achieved by running many tasks in parallel. The following figure shows the execution of a SQL query in Hive:

#### SELECT \* FROM customers SQL Query GROUP BY state Calcite Cost Based Optimizer **Hive Physical Optimizer** Tez Execution DAG Global Group By: state Table Scan: customers Vertex Local Group By: state Table Scan: Tasks customers. Table customers: distributed over many machines HDFS

#### Figure 11.2. SQL Query Execution in Hive

## **11.2. Searching and Identifying Hive Queries**

The landing page of the Tez View has a tabbed layout, with the **Hive Queries** tab as the default tab. While displaying all the query details in a tabular format with pagination support, the UI also provides options to search based on various parameters.

The search criteria of the **Hive Queries** tab are:

- Query ID
- User
- DAG ID
- Tables Read
- Tables Written
- App ID
- Queue
- Execution Mode

Search returns hits when there are exact matches with the criteria that is entered on the top of the tab. Therefore, you must enter each search string as a complete value in the search criteria fields. You can enter multiple values (table names) in the **Tables Read** filter and **Tables Written** filters.

#### Figure 11.3. Hive Queries Tab Showing Unfiltered Results

Hive Queries	All DAGs							Last refres
Query ID:	User:	DAG ID:	AG ID: Tables Read: T		App ID:	Queue:	Execution Mode:	
Search	Search	Search	Search	Search Search Search		Search	Search Q	
Query ID		User	Status	Query	DAG ID		Tables Read	Tables
hdfs_201703022	20426_bedb299b-3	3 hdfs	V SUCCEEDED	INSERT INTO TABL	dag_148837	7368398_0082_1	default.valuestmp	o default
hdfs_201703022	215315_dc251bl2-2.	hdfs	V SUCCEEDED	INSERT INTO TABL	dag_148837	7368398_0081_1	default.valuestmp	default
hdfs_201703021	92253_f9eb88d2-a.	hdfs	V SUCCEEDED	INSERT INTO TABL	dag_148837	7368398_0061_1	default.valuestmp	default
hdfs_201703011	40718_1a94a076-2	hdfs	✓ SUCCEEDED	INSERT INTO TABL	dag_148837	4743517_0002_14	default.valuestmp	default
hdfs_201703011	40716_3080e929-b.	hdfs	V SUCCEEDED	INSERT INTO TABL	dag_148837	4743517_0002_13	default.valuestmp	default
hdfs_201703011	40714_584642dc-3	hdfs	V SUCCEEDED	INSERT INTO TABL	dag_148837	4743517_0002_12	default.valuestmp	default

### **11.2.1.** Analyzing the Details of Hive Queries

Click the relevant link in the **Query ID** column for a query that you want to investigate. A window with three tabs containing information about the query is displayed.

#### **Details Tab**

The **Details** tab, the first of the three tabs, is displayed after clicking a **Query ID** link.

ref

# Figure 11.4. Details for a Successful Query with Links to Application and DAG Windows

Details		Hive Details		
Query ID	hdfs_20170302220426_bedb299b-32c7-4171-b76f-71bb6f5d1da7	Tables Read	default.values_tmp_table_2	
User	hdfs	Tables Written	default.sample_tmp	
Status	V SUCCEEDED	Client Address	172.22.76.27	
Start Time	03 Mar 2017 03:34:26	Execution Mode	TEZ	
End Time	03 Mar 2017 03:34:49	Hive Address	172.22.76.27	
Duration	22s 896ms	Client Type	CLI	
Application ID	application_1488377368398_0082			
DAG ID	dag_1488377368398_0082_1			
Session ID	Not Available!			
Thread Name	main			
Queue	default			

#### **Total Timeline View**

Click the **Timeline** tab to get a visual representation of Hive performance logs. The view represents the following pre-execution, runtime, and post-execution phases of a query:

- Pre-execution and DAG construction
- DAG submission
- DAG runtime
- Post-execution

Duration data about each phase are distilled into more granular metrics based on query execution logs.

A search-enabled table with raw performance log names and their major values is displayed under the timeline visualization.

Query Details	Timeline Configurat	ions							Last re
P		DAG Submission	1						DA
E		Submit Dag				Subm			
Pre-Execution +	DAG construction : 386m	BAG Submission : 1	3s 167ms	DAG Runtime :	9s 87ms	Post Executi	on : 7ms		
Compile	275ms	Submit Dag	12s 458ms	Run Dag	9s 87ms	Post ATS H	look	0ms	
Parse	2ms	Submit To Running	709ms			Remove Fil	les	2ms	
Build Dag	109ms					Rename O	r Move Files	5ms	
Search Raw Perf Log Na	ame	S	earch	\$	Value				
RemoveTempOrD	OuplicateFiles				2ms				
TezCreateVertex.	Map 1				107ms				
TezSubmitToRunn	ningDag				709ms				
partition-retrievin	9				Oms				
serializePlan						5ms			
TezSubmitDag						12s 458ms			
ezRunVertex.Ma	p 1				4s 251ms				
runTasks						22s 614ms			
lezBuildDag semanticAnalyze					109ms 269ms				

#### Figure 11.5. Total Timeline and Log Details of a Submitted Query

#### **Configurations Tab**

Click the **Configurations** tab to see a list of configuration properties and settings that are used in the Hive query. You can use this tab to verify that configuration property values align with your expectations.



#### Tip

By default, only configuration property names that contain the substring tez are listed. Use the **Search** field to change the search criteria.

#### Figure 11.6. Configurations Tab

Query Details Timeline Configurations		Last		
tez	Search			
Configuration Name		Configuration Value		
nive.convert.join.bucket.mapjoin.tez		false		
nive.tez.exec.print.summary		false		
nive.server2.tez.default.queues		default		
nive.tez.task.scale.memory.reserve.fraction.max		0.5		
nive.cluster.delegation.token.store.zookeeper.con	tez-ui-1.openstacklocal:2181			
ffs.namenode.http-address	tez-ui-1.openstacklocal:50070			
nive.in.tez.test	false			
nive.tez.task.scale.memory.reserve.fraction		-1.0		
arn.timeline-service.webapp.address		tez-ui-1.openstacklocal:8188		
nive.tez.dynamic.partition.pruning.max.event.size		1048576		
varn.timeline-service.address		tez-ui-1.openstacklocal:10200		
arn.resourcemanager.webapp.address		tez-ui-1.openstacklocal:8088		
nive.tez.enable.memory.manager		true		
arn.timeline-service.webapp.https.address	tez-ui-1.openstacklocal:8190			
arn.resourcemanager.admin.address	tez-ui-1.openstacklocal:8141			
arn.resourcemanager.resource-tracker.address		tez-ui-1.openstacklocal:8025		
varn.resourcemanager.hostname		tez-ui-1.openstacklocal		
varn.resourcemanager.webapp.https.address		tez-ui-1.openstacklocal:8090		

## **11.3. Identifying the Tez DAG for Your Job**

Click **All DAGs** to view the a list of jobs sorted by time, listing the latest jobs first. You can search for a job using the following search criteria fields:

- DAG Name
- Id (DAG identifier)
- Submitter (user who submitted the job)
- Status (job status)
- Application ID (application identifier)
- Caller ID

100%

✓ SUCCEEDED

DAG Name:	ID:		Submitter:	Status:
Search	Search		Search	All
Dag Name	ld	Submitter	Status	Progress
INSERT INTO TABL	dag_148420332822	admin		100%
ANALYZE TABLE `de	dag_148420332822	admin		100%
select count(*) from	dag_148420332822	admin		100%
select count(*) from	dag_148420332822	admin		100%
select count(*) from	dag_148420332822	admin	✓ SUCCEEDED	100%

#### Figure 11.7. All DAGs View (Truncated Screenshot)

#### Selecting the Columns That Appear in Search Results

select count(") from ... dag\_148420332822... admin

To select which columns are included in the Tez View search results, click the gear icon to the right of the search tool bar. A Column Selector dialog box appears where you can select which columns appear in the search results. Select the columns, and click **Ok** to return to Tez View:

#### Figure 11.8. Tez View Column Selector Dialog Box

Column Selector	×	Last refreshed at 17 Mar 2017 13:47:19 C Refresh				
Filter						
Select All			1 10 Rows 🛊 🛔			
Dag Name						
Ø Id	n Id	Queue	Caller ID			
Submitter	1_1484	20 default	hive_201703160422,	Click this		
Status	1_1484	20 default	hive_201703160356.	gear icon to		
✓ Progress	1_1484	20 default	hive_201703151853.			
Ø Start Time	1_1484	20 default	hive_201703151852,			
End Time	1_1484	20 default	hive_201703142142.	. selector		
Duration	1_1484	20 default	hive_201703142034.			
Application Id				1		
Queue						
Caller ID						
Caller Context						
Logs						
FileSystem - FILE_BYTES_READ						
FileSystem - FILE_BYTES_WRITTEN						



To search for columns, use the search filter at the top of the Column Selector dialog box. Click **Select All** to include all columns in your search results and uncheck it to clear all of your column selections.

#### Understanding Tez View Job Status

The following table explains the job status field that is returned for all search results returned in Tez View:

#### **Table 11.1. Tez Job Status Descriptions**

Status	Description
Submitted	The DAG is submitted to Tez but is not running.
Running	The DAG is currently running.
Succeeded	The DAG completed successfully.
Failed	The DAG failed to complete successfully.
Killed	The DAG was stopped manually.
Error	An internal error occurred when executing the DAG.

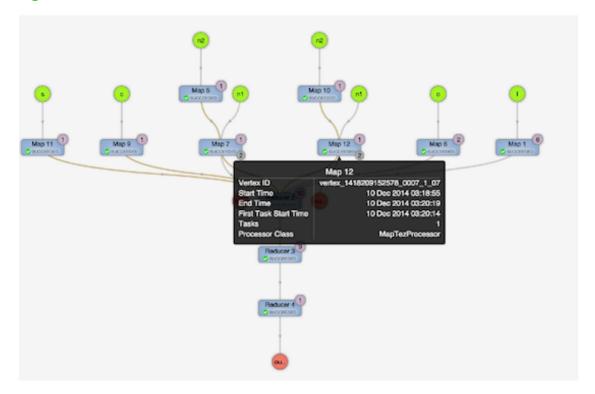
## **11.4. Understanding How Your Tez Job Is Executed**

Tez View enables you to gain insight into the complexity and the progress of executing jobs.

The View tab shows the following:

- DAG graphical view
- All vertices
- Tasks per vertex on top right of the vertex
- Failed vertices display in red, successful vertices display in green
- Mouse over vertices to view timeline details

Figure 11.9. View Tab in Tez View



The View Tab enables you to investigate the vertices that have failures or are taking a long time.

## **11.5. Identifying Causes of Failed Jobs**

Tez View enables you to quickly find and report errors. When a Tez task fails, you must:

- Identify why the task failed
- Capture the reason for task failure

The DAG Details tab

When a Tez task fails, the **DAG Details**tab explains the failure. You can download the data of the DAG to further examine the cause of failure. Starting with Tez View in Ambari 2.5.0, the downloader has the following features:

- A progress bar during the download process is displayed and includes messaging about the download status
- If the data download is not completely successful, some information rather than a failure message without any DAG details is provided. The available data is stored as JSON file in the zip archive.
- If the downloader fails to completely capture any data, the downloader re-attempts to download data. The maximum number of re-attempted downloads is 3. The main reason for this feature is to address the scenario where ATS is briefly offline.

#### Figure 11.10. DAG Details Window

	DAG	Details	DAG Counters	Graphical View	All Vertices	All Tasks	All TaskAttempt
0					Last refreshed at 1	4 Jun 2015 10	2:19:01 C Refre
DAG Details							
A Download o	data						
Application Id	application_1434230750579_0006						
Entity Id	dag_1434230750579_0006_1						
User	ambari-ga						
Status	FAILED [ Failed Tasks ] [ Failed TaskAttempts ]						
Start Time	14 Jun 2015 10:17:13						
End Time	14 Jun 2015 10:17:23						
Duration	9 secs						
Diagnostics							
> Task failed, > TaskAtte > Contr > Con Exit Stac a	texName=Map 1, vertexid=vertex_1434230750579_00 taskid=task_1434230750579_0006_1_00_000000, dia mpt 0 failed, info= inter container_1434230750579_0006_01_000002 finis ntainer failed, exitCode=1. Exception from container- tainer id: container_1434230750579_0006_01_000002 code: 1 k trace: ExitCodeException exitCode=1: t org.apache.hadoop.util.Shell.runCommand(Shell.javi t org.apache.hadoop.util.Shell.runCommand(Shell.javi t org.apache.hadoop.util.Shell.runCommand(Shell.javi t org.apache.hadoop.util.Shell.runCommand(Shell.javi t org.apache.hadoop.util.Shell.runCommand(Shell.javi	gnostics= shed with launch. 2 a:545)	diagnostics set to				

### **11.6. Viewing All Failed Tasks**

Multiple task failures may occur. The Tez View All Tasks tab enables you to view all tasks that failed and examine the reason and logs for each failure. Logs for failed tasks, but not for aborted tasks are available to download from this tab:

#### Figure 11.11. Tez View All Tasks Tab

Ambari 1	wyci	uster <b>10000 D</b>					1	Dashbo	ard Services	Hosts	Alerts	Admin 🚦	<b>≜</b> admi
🖀 All DAGs	/ D	AG [ hive_2015	06142	14445_ec5b3d	:31-962	2f-4303-a9	10-950897	abd09	9:1]				
					DA	G Details	DAG Cou	ters	Graphical View	v Al Ve	tices	I Taska All 1	TaskAttempts
0										Last refrest	ed at 14 Ju	n 2015 22:40:58	2 Refresh
Status:FAILED				Search							First	t 1 Last - 1	Pows 25
ask Index	0	Vertex Name	0	Status	0	Start Time	0	End Ti	me 0	Duration	0	Actions	Logs
000000 00		Map 3		FAILED		14 Jun 2015	14444	14.1-	2015 14:45:06	7 secs		counters attemp	pts Not /

# **11.7. Using Counters to Identify the Cause of Slow-Performing Jobs**

Tez View shows counters so you can understand why a task performs more slowly than expected. Counters help you better understand the task size and enable you to locate anomalies. Elapsed time is one of the primary counters to look for.

Counters are available at the DAG, vertex, and task levels. As of Tez View in Ambari 2.5.0, the following Hive LLAP counter information is displayed:

Group: org.apache.hadoop.hive.llap.counters.LlapIOCounters Counters: CACHE\_HIT\_BYTES, CACHE\_MISS\_BYTES, METADATA\_CACHE\_HIT, METADATA\_CACHE\_MISS

#### Figure 11.12. Tez View DAG-Level Counters Tab

Ambari MyCluster Dops Balerta	Dashboard Services Hosts Alerts Admin 🗰 🔺 admin 🗸
All DAGs / DAG [ OrderedWordCount ]	
	DAG Details DAG Counters Graphical View All Vertices All Tasks All TaskAttempts
0	Last refreshed at 14 Jun 2015 22:43:13 C Refresh
Counter Name	Counter Value
Search	
org.apache.fez.common.counters.DAGCounter	
NUM_SUCCEEDED_TASKS	3
TOTAL_LAUNCHED_TASKS	3
DATA_LOCAL_TASKS	1
AM_CPU_MILLISECONDS	1,550
AM_GC_TIME_MILLIS	196
File System Counters	
FILE_BYTES_READ	225
FILE_BYTES_WRITTEN	161

As of Tez View in Ambari 2.5.0, the **Vertex Swimlane** tab is also available. The information here is about vertex processor details for Hive and Pig.

#### Figure 11.13. Tez View Vertex Swimlane Tab

DAG Details	DAG Counters	Graphical View	All Vertices	All Tasks	All Task Attempts	Vertex Swimlane	Last refre	Auto Refresh shed at 17 Mar 2017 23:16:1	
scope-19 오	o						100% (	>	- • 2 0
Consolidated	0 Milliseconds	830	1660	2490 Sta Sta Du De Fin Tat Su Ru Pe	st Task Start Time	scope-1	9 vertex_1488377368398_0106_1_00 SUCCEEDED 17 Mar 2017 22:58:20 17 Mar 2017 22:58:20 17 Mar 2017 22:58:24 8 69 11m reater_than_four(2,27),movie(-1,-1) 17 Mar 2017 22:58:24 1 1 Not Availabble Not Availabble Not Availabble PigProcessor	6640 7470	834

If you want to view the information about the vertex processing in a different format, open a vertex to see the **Vertex Details** sutab. A **Description** pane at the bottom of the window is from the user payload of the respective vertex in the DAG plan.

#### Figure 11.14. Tez View Vertex Details Subtab

Vertex Details	Vertex Counters Tasks Task	Auto Refresh C Refresh		
Details		Stats		
Vertex ID	vertex_1488377368398_0105_1_00	Total Tasks	1	
Vertex Name	scope-19	Succeeded Tasks	1 Succeeded	
Processor Class	PigProcessor	Failed Tasks	0	
Status	✓ SUCCEEDED	Killed Tasks	0	
Progress	100%	First Task Start Time	17 Mar 2017 22:58:24 [ task_1488377368398_0105_1_00_000000 ]	
Start Time	17 Mar 2017 22:58:20	Last Task Finish Time	17 Mar 2017 22:58:28 [ task_1488377368398_0105_1_00_000000 ]	
End Time	17 Mar 2017 22:58:28	Average Duration	3s 954ms	
Duration	8s 361ms	Minimum Duration	3s 954ms [ task_1488377368398_0105_1_00_000000 ]	
		Maximum Duration	3s 954ms [ task_1488377368398_0105_1_00_000000 ]	
Description				
movies[1,9],mov	ies_greater_than_four[2,27],movies[-1,-	1] 0		

#### Figure 11.15. Tez View Vertex-Level Counters Tab

Ambari MyCluster ( 0 cps) 0 alerts	Dashbo	and Services	Hosts	Alerts	Admin	=	admin •
All DAGs / DAG [ OrderedWordCount ] / Vertex [ Tokenizer ]							
	Vertex Details	Vertex Counters	Tasks	Task /	Attempts	Sour	ces & Sinks
0		L	ist refreshe	d at 14 Ju	in 2015 22:	43:41	C Refresh
Counter Name	Counter Value						
Search							
org.apache.tez.common.counters.DAGCounter							
DATA_LOCAL_TASKS			1				
File System Counters							
FILE_BYTES_READ			32				
FILE_BYTES_WRITTEN			89				

#### Figure 11.16. Tez View Task-Level Counters Tab

Ambari MyCluster and Eaterts	Dashboard	Services	Hosts	Alerts	Admin	ш	admin •	
All DAGs / DAG [ OrderedWordCount ] / Vertex [ Tokenizer ] / Ta	ask [ 00_000000 ]							
		1	ask Details	Task	Counters	Tas	k Attempts	
0		L	ust refreshe	d at 14 Ju	in 2015 22:4	\$3:59	C Refresh	
Counter Name		Counter Value						
Search								
org.apache.tez.common.counters.DAGCounter								
DATA_LOCAL_TASKS			1					
File System Counters								
FILE_BYTES_READ			32					
FILE_BYTES_WRITTEN			89					

#### **Monitoring Task Progress for Jobs**

The Tez View shows task progress by increasing the count of completed tasks and total tasks. This enables you to identify the tasks that might be "hung" and to understand more about long-running tasks.

# **12. Using Workflow Manager View**

Ambari includes Workflow Manager View, which supports creating, scheduling, and monitoring jobs on a Hadoop cluster. Hadoop administrators can easily design and visualize workflows in the UI as flow graphs.

Workflow Manager is based on the Apache Oozie workflow engine that allows users to connect and automate the execution of big data processing tasks into a defined workflow. Workflow Manager integrates with the Hortonworks Data Platform (HDP) and supports Hadoop jobs for Hive, Sqoop, Pig, MapReduce, Spark, and more. In addition, it can be used to perform Java, Linux shell, distcp, SSH, email, and other operations.

You can access the Workflow Manager documentation in the Workflow Management guide on the Hortonworks documentation website.