Hortonworks Data Platform

Ranger Ambari Installation

(Jun 9, 2015)

docs.hortonworks.com

Hortonworks Data Platform: Ranger Ambari Installation

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

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

Please visit the Hortonworks Data Platform page for more information on Hortonworks technology. For more information on Hortonworks services, please visit either the Support or Training page. Feel free to contact us directly to discuss your specific needs.

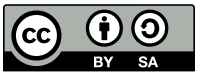


Table of Contents

List of Figures

List of Tables

1. Overview

Apache Ranger can be installed either manually using the Hortonworks Data Platform (HDP) or the Ambari 2.0 User Interface. Unlike the manual installation process, which requires you to perform a number of installation steps, installing Ranger using the Ambari UI is much simpler and easier. The Ranger service option will be made available through the Add Service wizard after the HDP cluster is installed using the installation wizard.

Note

Start the DataNode using the applicable commands in the "Controlling HDP Services Manually" section of Ambari Installation Guide

Once Ambari has been installed and configured, you then need only use the Add Service wizard to install the following components:

- Ranger Admin
- Ranger UserSync

After these components are installed and started, you can enable Ranger plugins by navigating to each individual Ranger service (HDFS, Hive, Knox, and Storm) and modifying the configuration under advanced ranger-service>-plugin-properties.

Note that when you enable a Ranger plugin, you will need to restart the component.

Note

Enabling Apache Storm requires you to enable Kerberos. Refer to the Configuring Kerberos Authentication for Storm section of *Installing HDP Manually* for more information on how to enable Kerberos for Storm.

2. Installation Prerequisites

Before you install Ranger, make sure your cluster meets the following requirements.

- A MySQL server of Oracle Server database instance running and available to be used by Ranger.
- For the Ranger Admin host, either MySQL Client or Oracle Client is installed so Ranger can access the database.
- The DBA Admin user (root in the case of MySQL or SYS for Oracle(is enabled in the database server from any host. To enable the DB Admin user, enter the following commands:

```
create user 'root'@'%' identified by 'rootpassword': --->only if this user
 does not
         exist grant all provileges on *.* to 'root'@'%' identified by
 'rootpassword' with grant
         option; flush privileges;
```

```
set password for 'root'@'localhost'=password \('rootpassword');
```

- Execute the following commands on the Ambari Server host:

  replace

  ```
  database-type
  ```

  with

  ```
  mysql
  ```

  or

  ```
  oracle
  ```

  , and /jdbc/driver/path based on the location of the MySQL or Oracle JDBC driver;

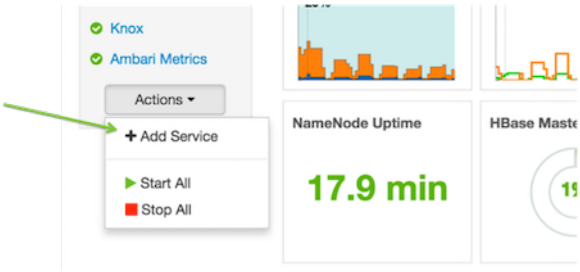
  ambari-server setup –jdbc-db={database-type} –jdbc-driver={/jdbc/driver/path}

3. Ranger Installation

Installing Ranger using the Ambari UI requires you to perform the following steps:

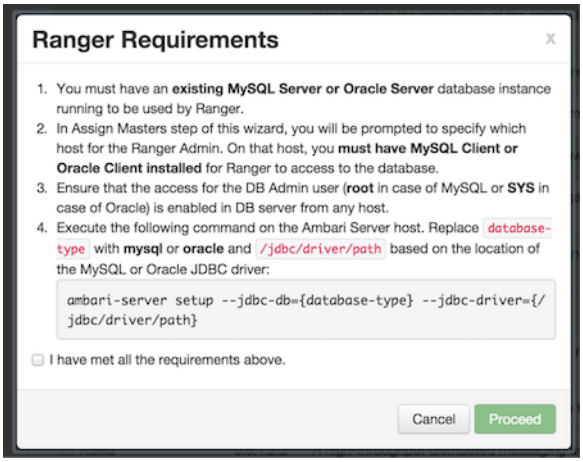
1. Log into the Ranger Admin interface with your user credentials.
2. In the left navigation page, click on the **Actions** button to display the Actions drop-down menu.
3. Click on **Add Service**.

Figure 3.1. Installing Ranger Add Service



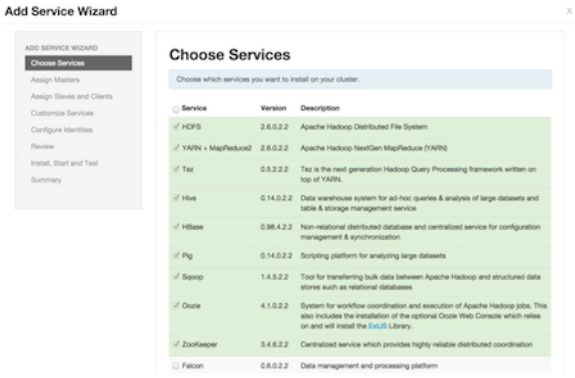
4. The Ranger Requirements page is then displayed. On this page, ensure you have met all of the requirements listed and click on the *I have met all the requirements above* checkbox. Select the **Proceed** box to continue with the installation and configuration.

Figure 3.2. Installing Ranger Ranger Requirements



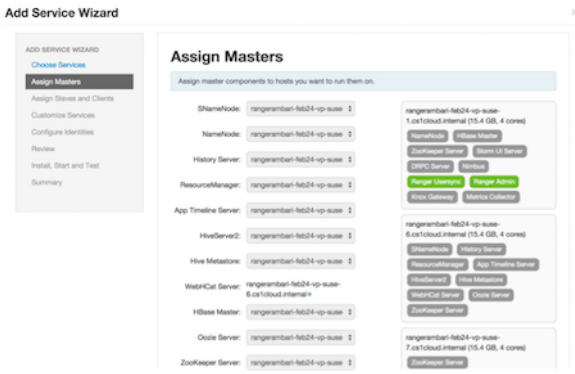
5. The Choose Services page is displayed. Select **Ranger** from the list of services.

Figure 3.3. Installing Ranger Choose Services



6. You are then prompted to select the host where Ranger Admin will be installed. This host should have DB admin access to the Ranger DB host and usersync. The figure below shows Ranger Admin and Usersync being installed on the same host.

Figure 3.4. Installing Ranger Assign Masters



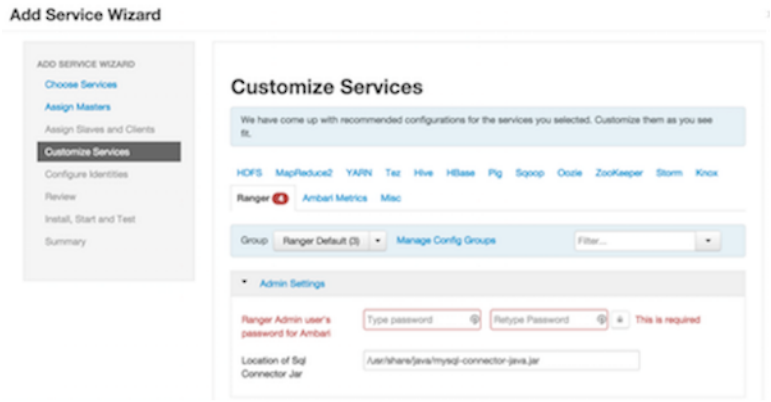
7. After selecting the host, update the properties listed in the sections below.

8. Once you have entered all the values in the Customize Services tab, review all the information and click **Deploy**.

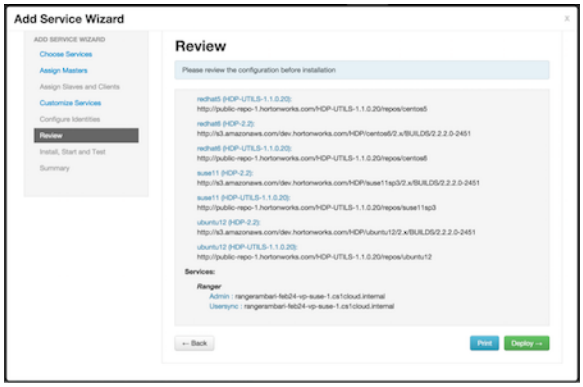
3.1. Admin Settings

In the Ranger Admin settings, be sure to set the user password for the Ranger Admin by specifying the password for the Ranger Admin user created from Ambari. This user can be used for any admin work from within Ambari.

Figure 3.5. Admin Settings Customize Services



Once you enter all the values in the Customize Services tab, review all the information and click **Deploy**.



3.2. Database Settings

In the Ranger Admin console, update the following properties shown in the figure and table below.

The table below describes each of the Ranger database settings fields that you can complete, Note that the fields designated in red are required fields.

Table 3.1. Ranger Database Settings

Configuration Property Name	Description	Default Value	Example Value	Required?
SQL Command Invoker				
Ranger DB host	The fully qualified hostname of the Ranger database server.		localhost	Yes
Ranger DB root user	The Ranger database user that has administrative privileges to create	root	root	Yes

Configuration Property Name	Description	Default Value	Example Value	Required?
	database schemas and users.			
Ranger DB root password	The root password for the Ranger database user (db_root_user).		rootPassWORd	Yes
Ranger DB name	The name of the Ranger Policy database	ranger	ranger	Yes
Ranger DB username	The username for the Policy database.	rangeradmin	rangeradmin	Yes
Ranger DB password	The password for the Ranger Policy database user		RangerAdmin PassWORd	Yes
Ranger Audit DB name	The name of the Ranger Audit database. This can be a different database in the same database.	ranger_audit	ranger_audit	Yes
Ranger Audit DB username	The username for the Ranger Audit database. This username performs all audit logging operatins.	rangerlogger	rangerlogger	Yes
Ranger Audit DB password	The password for the Ranger Audit Audit database.		RangerLogger PassWORd	Yes

If you are using Oracle, make sure you also update the following properties:

Table 3.2. Oracle Database Settings

Configuration Property Name	Description	Default Value	Example Value	Required?
DB FLAVOR	The database you want to use. Options are MySQL and Oracle.	MYSQL	MYSQL	Yes
SQL Command Invoker	The command used to invoke the SQL database.			No
SQL_CONNECTOR_JAR	The path to the Oracle JDBC driver. The database driver location for MySQL. If the Oracle database is used, copy the Oracle JDBC driver to */usr/share/java/ojdbc6.jar.* In Windows, only MySQL is supported.	*/usr/share/java/mysql-connector-java.jar*	*/usr/share/java/mysql-connector-java.jar*	Yes
ORACLE_HOME	The path to the folder for SQLPLUS			

3.3. Ranger Settings

In Ranger Settings, update the following fields shown in the figure and table below.

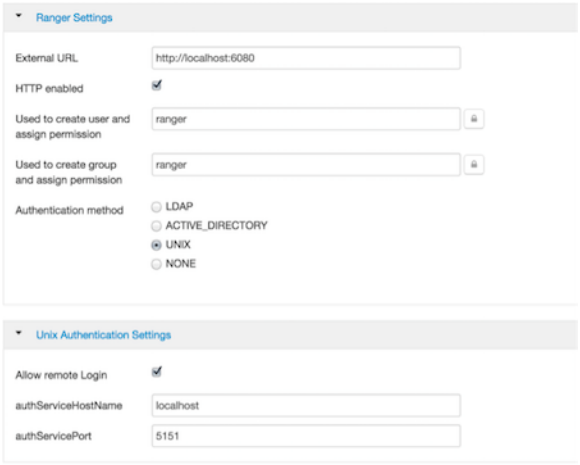


Table 3.3. Ranger Settings

Configuration Property Name	Description	Default Value	Example Value	Required?
External URL	The Ranger Admin host.			Yes
HTTP Enabled	Checkbox used to specify whether HTTP authentication should be enabled.			No
Used to create user and assign permisssion	Value used to create users and assign permissions.			Yes
Used to create group and assign permission	Value used to create groups and assign permissions.			Yes
Authentication method	The type of authentication method used to log into the Policy Admin tool. Only users created within the Policy Admin tool may log in. Types of authentication are: **LDAP**, **Active_Directory**, and **UNIX**	None	None	Yes
Allow remote login	Flag to enable/disable remote login via UNIX Authentication Mode.	TRUE	TRUE	Yes, if UNIX authentication_mode is selected.
authServiceHostName	Server Name (or IP address) where ranger-usersync module is running (along with UNIX Authentication Service).	localhost	myunixhost.domain. com	Yes, if UNIX authentication_method is selected.
authService Port	The port number where ranger-usersync module is running the UNIX Authentication Service.	5151	5151	Yes, if UNIX authentication_method is selected.

3.4. Advanced Usersync Properties

In the Advanced usersync Properties field, enter the following values in the specified fields.

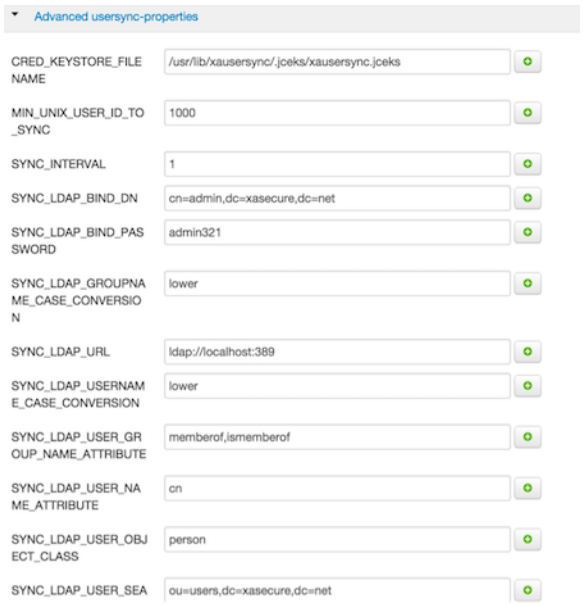


Table 3.4. Advanced Usersync Properties

Configuration Property Name	Description	Default Value	Example Value	Required
CRED_KEYSTORE FILENAME	Location of the file where the encrypted password is kept.	*/usr/lib/ xausersync/.jceks/ xausersync.jceks*	*/etc/ranger/usersync/ jceks/xausersync.jceks*	Yes, if SYNC_SOURCE is selected as LDAP.
MIN_UNIX_USER_ TO_SYNC	The UserId that is used to synchronize to the Ranger user database.	300 (UNIX), 1000 (LDAP)	1000	
SYNC_INTERVAL	Sepecifies the interval (in minutes) between the synchronization cycles. Note that the second sync cycle will NOT start until the first sync cycle is completed.		5	No
SYNC_SOURCE	Specifies where the user/group information is extracted to the put into the Ranger database. Specify whether you want to use UNIX or LDAP. UNIX retrieves the user information from */etc/passwd* file and retrieves group information from */ etc/group* file. LDAP retrieves the user		UNIX	No

Configuration Property Name	Description	Default Value	Example Value	Required
	information from the LDAP service.			
SYNC_LDAP_BIND_DN	The LDAP bind domain name used to connect to LDAP and query for users and groups.		cn=admin, ou=users, dc=hadoop, dc=apache dc-org	Yes, if SYNC_SOURCE is selected as LDAP.
SYNC_LDAP_ BIND_PASSWORD	The LDAP bind password fro the bind domain name specified in the SYNC_LDAP_BIND_DN		LdapAdminPassWORd	Yes, if SYNC_SOURCE is selected as LDAP.
SYNC_LDAP_GROUP NAME_ CASE_CONVERSION	Converts all group names to lower/upper case.	lower	lower	No (defaults to lower)
SYNC_LDAP_URL	The URL of the source LDAP.		Ldap:// ldap.example.com:389	Yes, if SYNC_SOURCE is selected as LDAP.
SYNC_LDAP_ USERNAME CASE_CONVERSION	Converts all usernames to lower/upper case. Lower=Usernames are converted to lower case when the username is saved to the Ranger database. Upper=Usernames are converted to upper case when the username is saved to the Ranger database.	lower	lower	No (defaults to lower)
SYNC_LDAP_USER_ GROUP_NAME_ ATTRIBUTE	An attribute from the user entry whose values would be treated as groups values to be pushed into the Policy Manager database. You can provide multiple attribute names, separated by a comma.	memberofismemberof	memberofismemberof	No (defaults to memberof, ismemberof)
SYNC_LDAP_USER_ NAME_ATTRIBUTE	An attribute from the user entry that is treated as a username.	cn	cn	No (defaults to cn)
SYNC_LDAP_USER_ OBJECT_CLASS	An objectclass used to identify user entries.	person	person	No (defaults to person)
SYNC_LDAP_USER_ SEARCH_FILTER	An additional optional filter constraining the users selected for syncing.		(dept=eng)	No (defaults to an empty string)

4. Ranger Plugins Overview

There are a number of Ranger plugins that must be enabled before using Ranger. The sections below describe how to enable each of these plugins.

Make sure the JDBC driver is available in the hosts where the Ranger plugin will be installed (e.g. the NameNode host in case of HDFS plugin, Hbase master and region server in case of HBase plugin) in the specified path during Ranger Admin installation. For example, create the folder */usr/share/java* and copy *mysql-connector-java.jar* if it is not already there.

If you are using a Kerberos-enabled cluster, there are a number of steps you need to follow to ensure you can use the different Ranger plugins on a Kerberos cluster. These plugins are:

1. HDFS
2. Hive
3. HBase
4. Knox

4.1. HDFS

Ranger plugins are enabled from the Ranger service itself. To enable the ranger HDFS plugin, perform the steps described below.

1. Select HDFS from the service and click on the **Configs** tab.
2. Navigate to *advanced ranger-hdfs-plugin-properties* and select the **Enable Ranger for HDFS** checkbox.
3. Select audit settings (Audit to DB or Audit to HDFS) and enter values accordingly. Note that only if **Audit to HDFS** is selected, settings related to that config will be shown. Refer to the table shown below for the different audit settings you can modify.
4. Save the configuration.
5. Ambari will display a restart indicator. Restart the HDFS component.
6. After the component is restarted, the Ranger plugin for HDFS will be enabled.

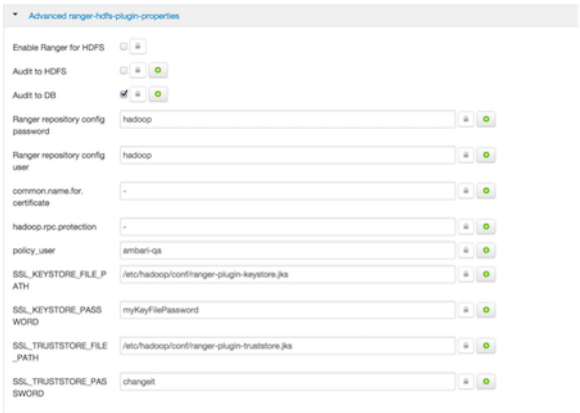


Table 4.1. HDFS Plugin Configuration Properties

Configuration Property Name	Description	Default Value	Example Value	Required?
Enable Ranger for HDFS	Flag used to enable/disable Hive funcitonality for Ranger.	FALSE		Yes
Audit to HDFS	Flag used to enable/ disable HDFS audit logging. If HDFS audit logging is turned off, it will not log any access control to HDFS.	FALSE		Yes
Audit to DB	Flag to enable/ disable database audit logging. If the database audit logging is turned off, it will not log any access to the database.	FALSE		Yes
Ranger repository config password				
Ranger repository config user				
common.name. for.certificate				
hadoop.rpc.protection	Configuration parameter used to control the quality of protection in the Hadoop cluster. Options are: **Authentication**, **Integrity**, and **Privacy**.		auth-int	No
policy_user				
SSL_KEYSTORE_ FILE_PATH	Java Keystore Path where the SSL key for the plugin is stored. This is used only if SSL is enabled between the Policy Admin Tool and Plugin. If SSL is not enabled, leave the default value as is - do not set as EMPTY is SSL is not used.	/etc/hadoop/conf/ ranger-plugin-keystore.jks	/etc/hadoop/conf/ ranger-plugin-keystore.jks	Yes, if only SSL is emanled
SSL_KEYSTORE_ PASSWORD	Password associated with SSL Keystore. Is used only if SSL is enabled between Policy Admin Tool and Plugin; if SSL is not enabled, leave the default value as is - do not set as EMPTY if SSL is not used.	None	None	Yes, if SSL is enabled.

Configuration Property Name	Description	Default Value	Example Value	Required?
SSL_KEYSTORE_ FILEPATH	Java Keystore Path where the trusted certificates are stored for verifying SSL connections to the Policy Admin Tool. Is used only if SSL is enabled between the Policy Admin Tool and Plugin; if SSL is not enabled, leave the default value as is - do not set as EMPTY if SSL is not used.	/etc/hadoop/conf/ranger-plugin-truststore.jks	/etc/hadoop/conf/ranger-plugin-truststore.jks	Yes, if SSL is enabled.
SSL_TRUSTSTORE_ PASSWORD	Password associated with Truststore file. Is used only if SSL is enabled between the Policy Admin Tool and Plugin; if SSL is not enabled, leave the default value as is - do not set as EMPTY if SSL is not used.	None	None	Yes, if SSL is enabled.

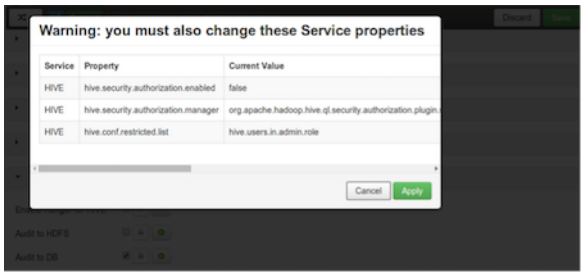
4.2. Hive

Important

You should not use the Hive CLI after enabling the Ranger Hive plugin. The Hive CLI is not supported in HDP-2.2.0 and higher versions, and may break the install or lead to other unpredictable behavior. Instead, you should use the HiveServer2 Beeline CLI.

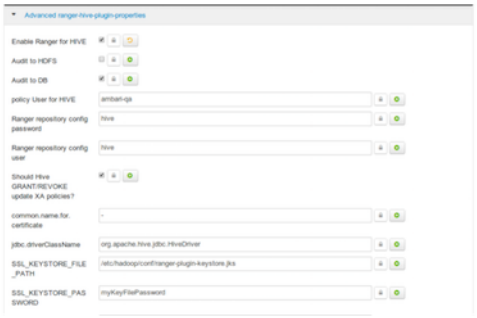
To enable the Ranger Hive plugin, perform the steps described below.

1. Navigate to the Hive service.
2. Click on the **Config** tab.
3. In the Config tab, navigate to *advanced ranger-hive-plugin-properties*.
4. Enter values in the fields listed in the Ranger Hive Settings table shown below.
5. Make sure to select the **Enable Ranger for Hive** check box.
6. When you select this check box, a warning dialog will appear.



7. Click **Apply** to save these changes.

8. Ambari will present a restart indicator. Restart the Hive component.



9. Enter values in the fields listed in the Ranger Hive Settings table shown below.

Table 4.2. Hive Plugin Configuration Properties

Configuration Property Name	Description	Default Value	Example Value	Required?
Enable Ranger for Hive	Flag used to enable/disable Hive functioanlity for Ranger.	FALSE		Yes
Audit to HDFS	Flag used to enable/ disable Hive audit logging. If Hive audit logging is turned off, it will not log any access control to HDFS.	FALSE		Yes
Audit to DB	Flag to enable/ disable database audit logging. If the database audit logging is turned off, it will not log any access control to database.	FALSE		Yes
Policy User for Hive				
Ranger repository config password				
Should Hive GRANT/ REVOKE uspdate XA policies?	Checkbox that provides the ability for the XAAgent to update the policies based on the grant/ revoke commands from the Hive client.			
common.name. for.certificate				
jdbc.driverClassName				
SSL_KEYSTORE_ FILE_PATH	Java Keystore path where SSL key for the plugin is stored.	*/etc/hive/conf/ ranger-plugin-keystore.jks*	*/etc/hive/conf/ ranger-plugin-keystore.jks*	Yes, if SSL is enabled.

Configuration Property Name	Description	Default Value	Example Value	Required?
SSL_KEYSTORE_ PASSWORD	Password associated with the SSL Keystore. This is only used if SSL is enabled between Policy Admin Tool and Plugin. If SSL is not enabled, leave the default value as is - do not set as EMPTY if SSL not used.	None	None	Yes, if SSL is enabled.
SSL_TRUSTSTORE_ FILE_PATH	The Java Keystore path where the trusted certificates are stored for verifying the SSL connection to the Policy Admin Tool. This is used only if SSL is enabled between the Policy Admin Tool ad Plugin. If SSL is not enabled, leave the default value as is - do not set as EMPTY if SSL is not used.	/etc/hive/conf/ranger-plugin-truststore.jks	/etc/hive/conf/ranger-plugin-truststore.jks	Yes, if SSL is enabled.
SSL_TRUSTSTORE_ PASSWORD	The password associated with the Truststore file. This is used only if SSL is enabled between the Policy Admin Tool and Plugin. If SSL is not enabled, leave the default value as is - do not set as EMPTY if SSL is not used.	None	None	Yes, if SSL is enabled.

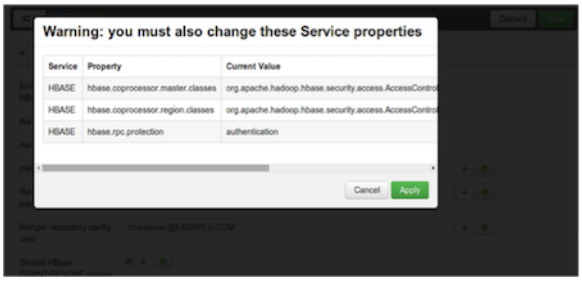
4.3. HBase

To enable the Ranger HBase plugin, perform the steps described below.

1. Navigate to the HBase service.

2. Click on the Config tab and navigate to advanced *ranger-hbase-plugin-properties*. Refer to the Ranger HBase Properties table for information on modifying these properties.

3. Make sure to select the **Enable Ranger for HBase** checkbox.
4. When you select the checkbox, a warning dialog popup will be opened.



5. Click on the **Apply** button to save the changes.
6. Ambari will display a Restart indicator. Restart the Ranger HBase component.

Table 4.3. Ranger HBase Properties

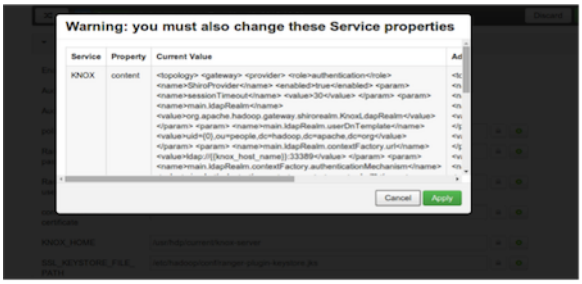
Configuration Property Name	Description	Default Value	Example Value	Required
Enable Ranger for HBASE	Flag used to enable/ disable HBase functionality for Ranger.	FALSE		Yes
Audit to HDFS	Flag used to enable/ disable HBase audit logging. If HBase audit logging is turned off, it will not log any access control to HBase.	FALSE		Yes
Audit to DB	Flag to enable/ disable database audit logging. If the database audit logging is turned off, it will not log any access control to the database.			
Policy User for HBASE				
Ranger repository config password				

Configuration Property Name	Description	Default Value	Example Value	Required
Ranger repository config user				
Should HBase GRANT/REVOKE update XA policies?	Checbox that provides the ability for the XA Agent to update the policies based on the grant/ revoke commands from the HBase client.	TRUE	TRUE	Yes
common.name. for.certificate				
SSL_KEYSTORE_ FILE_PATH	Java Keystore path where the SSL key for the plugin is stored. This is only used if SSL is enabled between the Policy Admin Tool and plugin. If SSL is not enabled, leave the default value as is - do not set as EMPTY if SSL is not used.	*/etc/hbase/conf/ ranger-plugin-truststore.jks*	*/etc/hbase/conf/ ranger-plugin-truststore.jks*	Yes, if SSL is enabled
SSL_KEYSTORE_ PASSWORD	Password associated with the SSL Keystore. This is only used if SSL is enabled between the Policy Admin Tool and plugin. If SSL is not enabled, leave the default value as is - do not set as EMPTY if SSL is not enabled.	myKeyFilePassword	MyKeyFilePassword	Yes, if SSL is enabled
SSL_KEYSTORE_ FILE_PATH	Java Keystore path where the trsuted certificates are stored for verifying SSL connection to the Policy Admin Tool. This is used only if SSL is enabled between the Policy Admin Tool and plugin. If SSL is not enabled, leave the default value as is - do not set as EMPTY.	*/etc/hbase/conf/ ranger-plugin-truststore.jks*	*/etc/hbase/conf/ ranger-plugin-truststore.jks*	Yes, if SSL is enabled
SSL_TRUSTSTORE_ PASSWORD	Password associated with the Truststore file. This is used only if SSL is enabled between the Policy Admin tool and plugin. If SSL is not enabled, leave the default value as is - do not set as EMPTY.	changeit	changeit	Yes, if SSL is enabled.

4.4. Knox

To enable the Ranger Knox plugin, perform the steps described below.

1. Navigate to the Knox service.
2. Click on the **Config** tab and navigate to advance ranger-knox-plugin-properties and modify the values in the Knox Plugin Properties table shown below.
3. Make sure to select the **Enable Ranger for Knox** checkbox.
4. When you select the checkbox, a warning dialog popup will be opened.



5. Click on the **Apply** button to save the changes.
6. Ambari will display a Restart indicator.
7. Restart the Ranger Knox component.

Table 4.4. Knox Plugin Properties

Configuration Property Name	Description	Default Value	Example Value	Required?
Enable Ranger for KNOX	Flag used to enable/ disable Knox functionality for Ranger.	FALSE		Yes
Audit to HDFS	Flag used to enable/ disable Knox audit logging. If Knox audit logging is turned off, it will not log any access control to Knox.	FALSE		Yes
Audit to DB	Flag to enable/ disable database audit logging. If the database audit logging is turned off, it will not log any access control to database.	FALSE		Yes
policy User for Knox				
Ranger repository config password				
Ranger repository config user				

Configuration Property Name	Description	Default Value	Example Value	Required?
common.name for.certificate				
KNOX_HOME				
SSL_KEYSTORE_ FILE_PATH	The Java Keystore path wwhere the SSL key for the plugin is stored. This is only used if SSL is enabled between the Policy Admin tool and plugin.	*/etc/knox/conf/ ranger-plugin-truststore.jks*	*/etc/knox/conf/ ranger-plugin-truststore.jks*	Yes, if SSL is enabled
SSL_KEYSTORE_ PASSWORD	The password associated with SSL Keystore. This is only used if SSL is enabled between the Policy Admin tool and plugin. If SSL is not enabled, leave the default value as is - do not set as EMPTY if SSL is not enabled	MyKeyFilePassword	MyKeyFilePassword	Yes, if SSL is enabled
SSL_TRUSTSTORE_ FILE_PATH	The Java Keystore path where the trusted certificates are stored for verifying SSL connection to the Policy Admin tool. This is only used if SSL is enabled between the Policy Admin tool and plugin. If SSL is not enabled, leave the default value as is - do not set as EMPTY if SSL is not enabled.	*/etc/knox/conf/ ranger-plugin-truststore.jks*	*/etc/knox/conf/ ranger-plugin-truststore.jks*	Yes, if SSL is enabled
SSL_TRUSTSTORE_ PASSWORD	The password associated with the truststore file. This is only used if SSL is enabled between the Policy Admin tool and plugin. If SSL is not enabled, leave the default value as is - do not set as EMPTY if SSL is not enabled.	changeit	changeit	Yes, if SSL is enabled

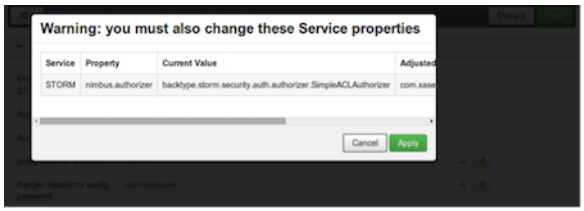
4.5. Storm

Before you can use the Storm plugin, you must first enable Kerberos on your cluster. To enable Kerberos on your cluster:

1. Add a system (OS) user *stormtestuser*.
2. Make sure this user is synced to Ranger Admin (under *users/groups* tab in the Ranger Admin UI).

3. Create a Kerberos principal by entering the following command:
   - `kadmin.local -q 'addprinc -pw stromtestuser stormtestuser@example.com'`
4. After applying Kerberos setup and creating the user/principal, navigate to the Storm service and click on the **Config** tab.
5. Navigate to *advanced ranger-storm-plugin-properties* and modify the properties shown in the table below.

6. Select the Enable Ranger for Storm checkbox.
7. Under the same Config tab, set `common.name.for.certificate` as blank.
8. When you select the checkbox, a warning dialog popup window will be opened.



9. Click on the **Apply** button to save the changes.
10. Ambari will display a Restart indicator.
11. Restart the Ranger Storm component.

Table 4.5. Storm Plugin Properties

Configuration Property Name	Description	Default Value	Example Value	Required?
Enable Ranger for STORM	Flag used to enable/ disable Storm functionality for Ranger.	FALSE		Yes
Audit to HDFS	Flag used to enable/ disable Storm audit logging. If Storm audit logging is turned off, it will not log any access control to Storm.	FALSE		Yes

Configuration Property Name	Description	Default Value	Example Value	Required?
Audit to DB	Flag to enable/ disable database audit logging. If the database audit logging is turned off, it will not log any access control to database.	FALSE		Yes
policy User for Storm				
Ranger repository config password				
Ranger repositoy config user				
common.name. for.certificate				
SSL_KEYSTORE_ FILE_PATH	The Java Keystore path where the SSL key for the plugin is stored. This is only used if SSL is enabled between the Policy Admin tool and plugin. If SSL is not enabled, leave the default value as is - do not set as EMPTY if SSL is not enabled.	*/etc/storm/conf/ ranger-plugin-truststore.jks*	*/etc/storm/conf/ ranger-plugin-truststore.jks*	Yes, if SSL is enabled
SSL_KEYSTORE_ PASSWORD	The password associated with SSL Keystore. This is only used if SSL is enabled between the Policy Admin tool and plugin. If SSL is not enabled, leave the default value as is - do not set as EMPTY if SSL is not enabled.	myKeyFilePassword	myKeyFilePassword	Yes, if SSL is enabled
SSL_TRUSTSTORE_ FILE_PATH	The Java Keystore path where the trusted certificates are stored for the Policy Admin tool. This is only usedif SSL is enabled between the Policy Admin tool and plugin. If SSL is not enabled, leave the default value as is - do not set as EMPTY if SSL is not enabled.	*/etc/storem/conf/ ranger-plugin-truststore.jks*	*/etc/storm/conf/ ranger-plugin-truststore.jks*	Yes, if SSL is enabled
SSL_TRUSTSTORE_ PASSWORD	The password associated with the truststore file. This is used only if SSL is enabled between the Policy Admin tool and plugin. If SSL is not enabled, leave the default value as is -	changeit	changeit	Yes, if SSL is enabled.

Configuration Property Name	Description	Default Value	Example Value	Required?
	do not set as EMPTY if SSL is not enabled.			

5. Ranger Plugins - Kerberos Overview

If you are using a Kerberos-enabled cluster, there are a number of steps you need to follow to ensure you can use the different Ranger plugins on a Kerberos cluster. These plugins are:

1. HDFS
2. Hive
3. HBase
4. Knox

5.1. HDFS

To enable the Ranger HDFS plugin on a Kerberos-enabled cluster, perform the steps described below.

1. Create the system (OS) user *hdfsuser*. Make sure this user is synced to Ranger Admin (under *users/groups* tab in the Ranger Admin User Interface.
2. Create a Kerberos principal for *hdfsuser* by entering the following command:
   - `kadmin.local -q 'addprinc -pw hdfsuser hdfsuser@example.com`
3. Navigate to the HDFS service.
4. Click on the **Config** tab.
5. Navigate to *advanced ranger-hdfs-plugin-properties* and update the properties listed in the table shown below.

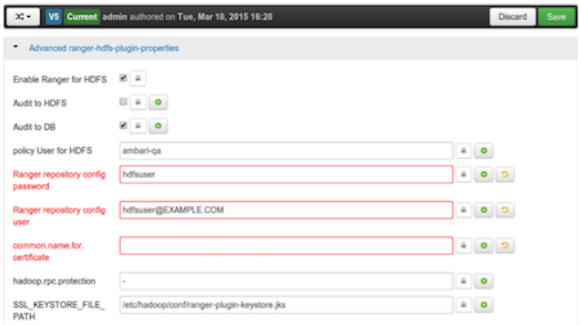


Table 5.1. HDFS Plugin Properties

Configuration Property Name	Value
Ranger repository config user	hdfsuser@example.com
Ranger repository config password	hdfsuser
common.name.for.certificate	blank

6. After updating these properties, click **Save** and restart the HDFS service.

5.2. Hive

Important

You should not use the Hive CLI after enabling the Ranger Hive plugin. The Hive CLI is not supported in HDP-2.2.0 and higher versions, and may break the install or lead to other unpredictable behavior. Instead, you should use the HiveServer2 Beeline CLI.

To enable the Ranger HBase plugin on a Kerberos-enabled cluster, perform the steps described below.

1. Create the system (OS) user *hiveuser*. Make sure this user is synced to Ranger Admin (under *users/groups* tab in the Ranger Admin UI).
2. Create a Kerberos principal for *hiveuser* by entering the following command:
   - `kadmin.local -q 'addprinc -pw hiveuser hiveuser@example.com`
3. Navigate to the Hive service.
4. Click on the **Config** tab and navigate to *advanced ranger-hive-plugin-properties*.
5. Update the following properties with the values listed in the table below.

Table 5.2. Hive Plugin Properties

Configuration Property Name	Value
Ranger repository config user	hiveuser@example.com
Ranger repository config password	hiveuser
common.name.for.certificate	blank

6. After updating these properties, click **Save** and then restart the Hive service.

5.3. HBase

To enable the Ranger HBase plugin on a Kerberos-enabled cluster, perform the steps described below.

1. Create the system (OS) user *hbaseuser*. Make sure this user is synced to Ranger Admin (under *users/groups* tab in the Ranger Admin UI).
2. Create a Kerberos principal for *hbaseuser* by entering the following command:
   - `kadmin.local -q 'addprinc -pw hbaseuser hbaseuser@example.com`
3. Navigate to the HBase service.
4. Click on the **Config** tab and go to *advanced ranger-hbase-plugin-properties*.
5. Update the following properties with the values listed in the table below.

Table 5.3. HBase Plugin Properties

Configuration Property Name	Value
Ranger repository config user	hbaseuser@example.com
Ranger repository config password	hbaseuser
common.name.for.certificate	blank

6. After updating these properties, click **Save** and then restart the HBase service.

5.4. Knox

To enable the Ranger Knox plugin on a Kerberos-enabled cluster, perform the steps described below.

1. Create the system (OS) user *knoxuser*. Make sure this user is synced to Ranger Admin (under *users/groups* tab in the Ranger Admin UI).

2. Create a Kerberos principal for *knoxuser* by entering the following command:

   - `kadmin.local -q 'addprinc -pw knoxuser knoxuseer@example.com`

3. Navigate to the Knox service.

4. Click on the **Config** tab and navigate to *advanced ranger-knox-plugin-properties*.

5. Update the following properties with the values listed in the table below.

Table 5.4. Knox Plugin Properties

Configuration Property Name	Value
Ranger repository config user	knoxuser@example.com
Ranger repository config password	knoxuser
common.name.for.certificate	blank

6. After updating these properties, click **Save** and then restart the Knox service.

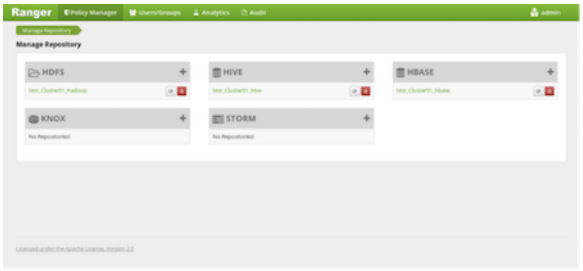
7. Open the Ranger Admin UI by entering the following information:

   - http://ranger-host>:6080

   - **username/password** - *admin*/*admin*. or use *username* as shown in *advanced ranger-env* under the **Config** tab of the Ranger service, and *password* as shown in **Admin Settings**.

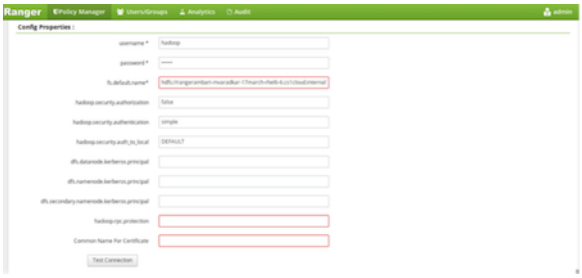
8. After you have successfully logged into the system, you will be redirected to the Policy Manager page.

Figure 5.1. Knox Policy Manager



9. Click on the repostory (clusterName_hadoop) **Edit** option under the HDFS box.

Figure 5.2. Knox Repository Edit



10. Update the following properties listed in the table below under the Config Properties section:

Table 5.5. Knox Configuration Properties

Configuration Property Name	Value
fs.default.name	hdfs
hadoop.rpc.protection	blank
common.name.for.certificate	blank

11. Click on **Named Test Connection**. You should see a *Connected Successfully* dialog boxappear.

12. Click **Save**.

6. About HDP

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.**

For more information on Hortonworks technology, Please visit the Hortonworks Data Platform page. 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.