Hortonworks Data Platform

Ranger Ambari Installation

(September 30, 2015)

docs.cloudera.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 2.1 User Interface (UI). Unlike the manual installation process, which requires you to perform a number of installation steps, installing Ranger using the Ambari UI is 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.

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

- Ranger Admin
- Ranger UserSync
- Ranger Key Management Service

After these components are installed and started, you can enable Ranger plugins by navigating to each individual Ranger service (HDFS, HBase, Hiveserver2, Storm, Knox, YARN, and Kafka) 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 or Apace Kafka requires you to enable Kerberos. To enable Kerberos on your cluster, see Enabling Kerberos Security in the Ambari Security Guide.

2. Installation Prerequisites

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

- A MySQL, Oracle, or PostgreSQL database instance is running and available to be used by Ranger.

  The Ranger installation will create two new users (default names: rangeradmin and rangerlogger) and two new databases (default names: ranger and ranger_audit).

- Configure the database instance for Ranger as described in the following sections.

  - Configuring MySQL for Ranger [2]
  - Configuring PostgreSQL for Ranger [3]
  - Configuring Oracle for Ranger [4]

2.1. Configuring MySQL for Ranger

1. You can use the MySQL root user to create the Ranger databases.

   Optionally, you can also create a non-root user to use to create the Ranger databases. For example, you would use the following series of commands to create the `rangerdba` user with password `rangerdba`.

   a. Log in as the root user, then use the following commands to create the `rangerdba` user and grant it adequate privileges.

   ```
   CREATE USER 'rangerdba'@'localhost' IDENTIFIED BY 'rangerdba';

   GRANT ALL PRIVILEGES ON *.* TO 'rangerdba'@'localhost';

   CREATE USER 'rangerdba'@'%' IDENTIFIED BY 'rangerdba';

   GRANT ALL PRIVILEGES ON *.* TO 'rangerdba'@'%';

   GRANT ALL PRIVILEGES ON *.* TO 'rangerdba'@'localhost' WITH GRANT OPTION;

   GRANT ALL PRIVILEGES ON *.* TO 'rangerdba'@'%' WITH GRANT OPTION;

   FLUSH PRIVILEGES;
   ```

   b. Use the `exit` command to exit MySQL.

   c. You should now be able to reconnect to the database as `rangerdba` using the following command:

   ```
   mysql -u rangerdba -prangerdba
   ```

   After testing the `rangerdba` login, use the `exit` command to exit MySQL.

2. Use the following command to confirm that the `mysql-connector-java.jar` file is in the Java share directory. This command must be run on the server where Ambari server is installed.

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

If the file is not in the Java share directory, use the following command to install the MySQL connector .jar file.

RHEL/CentOS/Oracle Linux

```
yum install mysql-connector-java*
```

SLES

```
zypper install mysql-connector-java*
```

3. Use the following command format to set the `jdbc/driver/path` based on the location of the MySQL JDBC driver .jar file.This command must be run on the server where Ambari server is installed.

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

For example:

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

2.2. Configuring PostgreSQL for Ranger

1. On the PostgreSQL host, install the applicable PostgreSQL connector.

RHEL/CentOS/Oracle Linux

```
yum install postgresql-jdbc*
```

SLES

```
zypper install -y postgresql-jdbc
```

2. Confirm that the .jar file is in the Java share directory.

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

3. Change the access mode of the .jar file to 644.

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

4. You can use the PostgreSQL root user to create the Ranger databases.

Optionally, you can also create a non-root user to use to create the Ranger databases. For example, you would use the following series of commands to create the `rangerdba` user and grant it adequate privileges.

Log in as the root user and enter:

```
echo "CREATE DATABASE $dbname;" | sudo -u $postgres psql -U postgres
```

```
echo "CREATE USER $rangerdba WITH PASSWORD '$passwd';" | sudo -u $postgres
 psql -U postgres
echo "GRANT ALL PRIVILEGES ON DATABASE $dbname TO $rangerdba;" | sudo -u
 postgres psql -U $postgres
```

Where:

- `$postgres` is the postgres user
- `$dbname` is the name of your PostgreSQL database

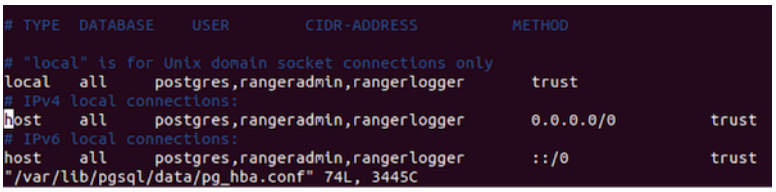
5. Use the following command format to set the `jdbc/driver/path` based on the location of the PostgreSQL JDBC driver .jar file. This command must be run on the server where Ambari server is installed.

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

For example:

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

6. Add allow access details for Ranger users:
   - change `listen_addresses='localhost'` to `listen_addresses='*' ('*' = any)` to listen from all IPs in `postgresql.conf`.
   - Make the following changes to the Ranger db user and Ranger audit db user in `pg_hba.conf`.



2.3. Configuring Oracle for Ranger

1. On the Oracle host, install the appropriate JDBC .jar file.
   - Download the Oracle JDBC (OJDBC) driver from http://www.oracle.com/technetwork/database/features/jdbc/index-091264.html.
   - For **Oracle Database 11g**: select Oracle Database 11g Release 2 drivers > ojdbc6.jar.
   - For **Oracle Database 12c**: select Oracle Database 12c Release 1 driver > ojdbc7.jar.
   - Copy the .jar file to the Java share directory. For example:

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

Note

Make sure the .jar file has the appropriate permissions. For examle:

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

2. You can use the Oracle root user to create the Ranger databases.

   Optionally, you can also create a non-root user to use to create the Ranger databases. For example, you would use the following series of commands to create the RANGERDBA user and grant it permissions using SQL*Plus, the Oracle database administration utility:

   ```
   # sqlplus sys/root as sysdba
   CREATE USER $RANGERDBA IDENTIFIED BY $RANGERDBAPASSWORD;
   GRANT SELECT_CATALOG_ROLE TO $RANGERDBA;
   GRANT CONNECT, RESOURCE TO $RANGERDBA;
   QUIT;
   ```

3. Use the following command format to set the `jdbc/driver/path` based on the location of the Oracle JDBC driver .jar file. This command must be run on the server where Ambari server is installed.

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

   For example:

   ```
   ambari-server setup --jdbc-db=oracle --jdbc-driver=/usr/share/java/ojdbc6.
   jar
   ```

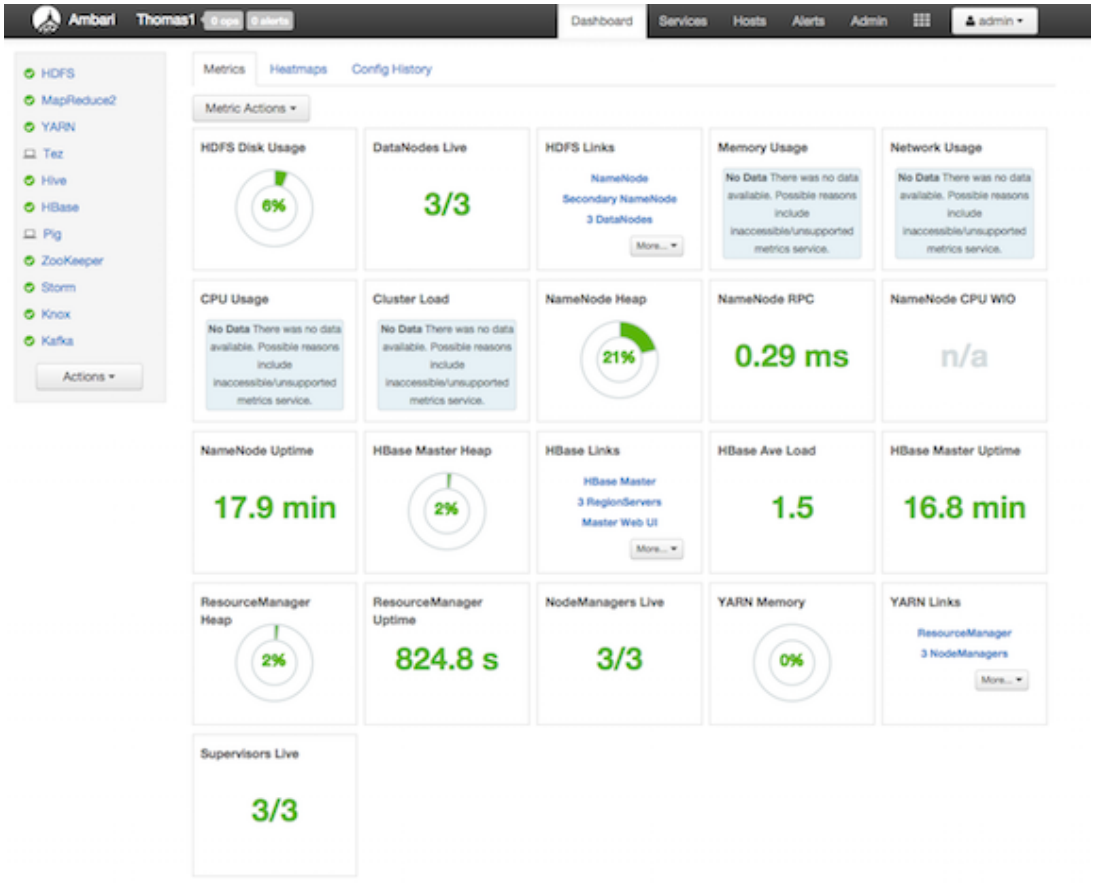
3. Ranger Installation

Use the following steps to install Ranger using Ambari.

3.1. Start the Installation

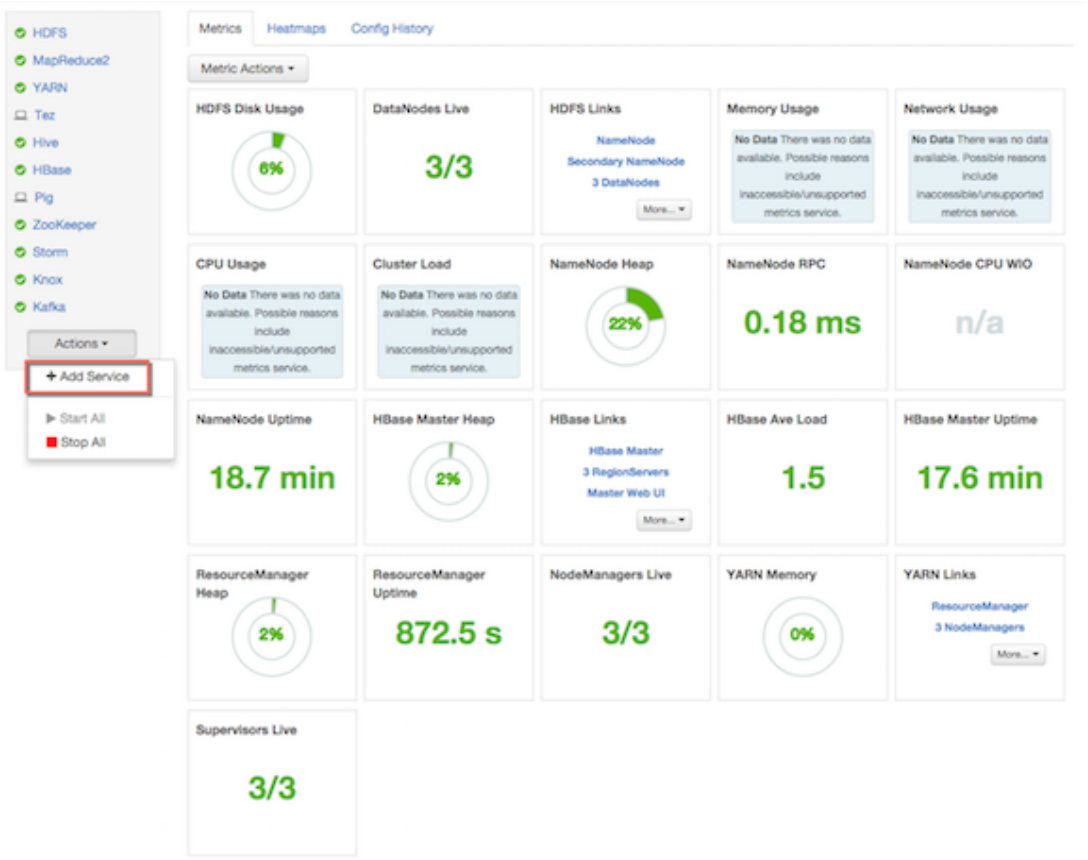
1. Log into your Ambari cluster with your designated user credentials. The main Ambari Dashboard page will be displayed.

Figure 3.1. Installing Ranger - Main Dashboard View



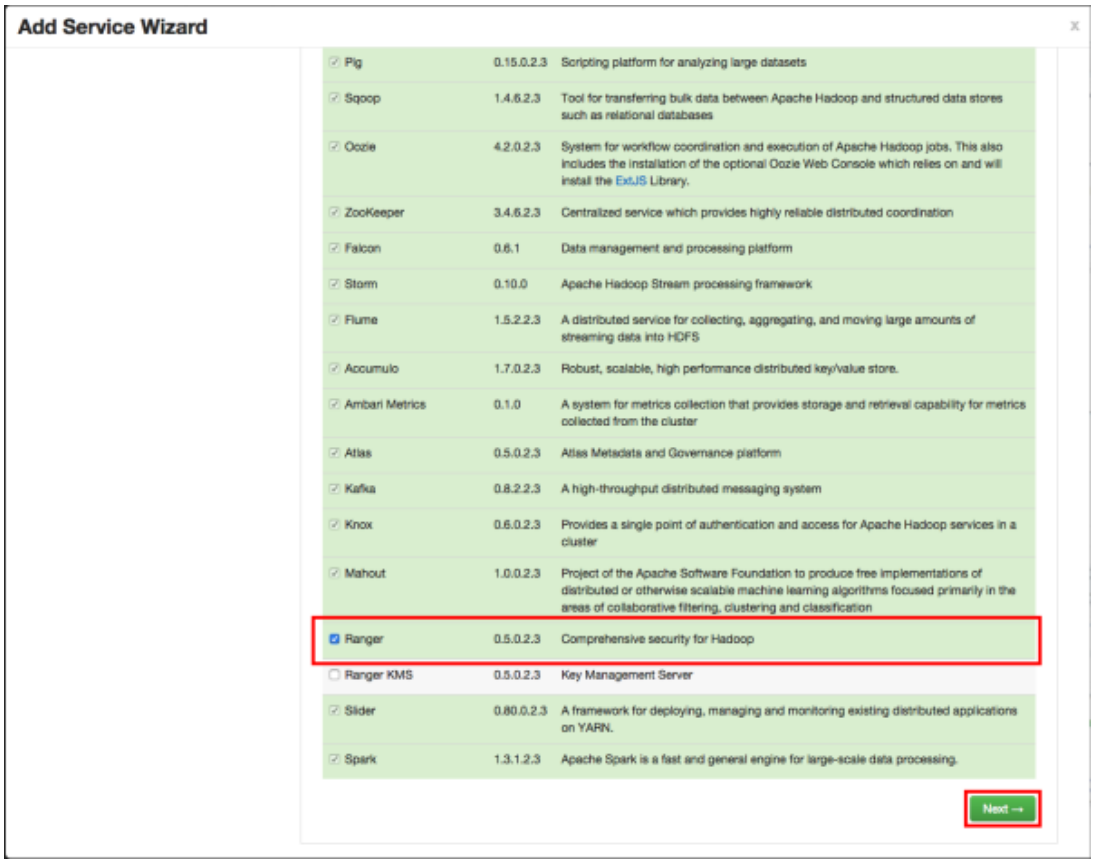
2. In the left navigation menu, click **Actions**, then select **Add Service**.

Figure 3.2. Installing Ranger - Add Service



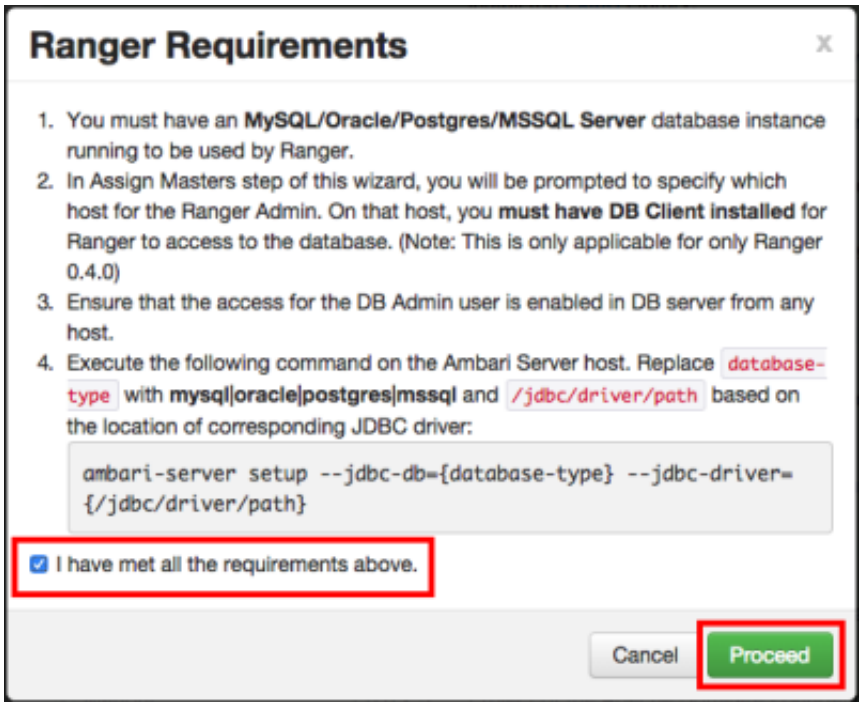
3. On the Choose Services page, select **Ranger**, then click **Next**.

Figure 3.3. Installing Ranger - Choose Service



4. The Ranger Requirements page appears. Ensure that you have met all of the installation requirements, then select the "I have met all the requirements above" check box and click **Proceed**.

Figure 3.4. Installing Ranger - Ranger Requirements



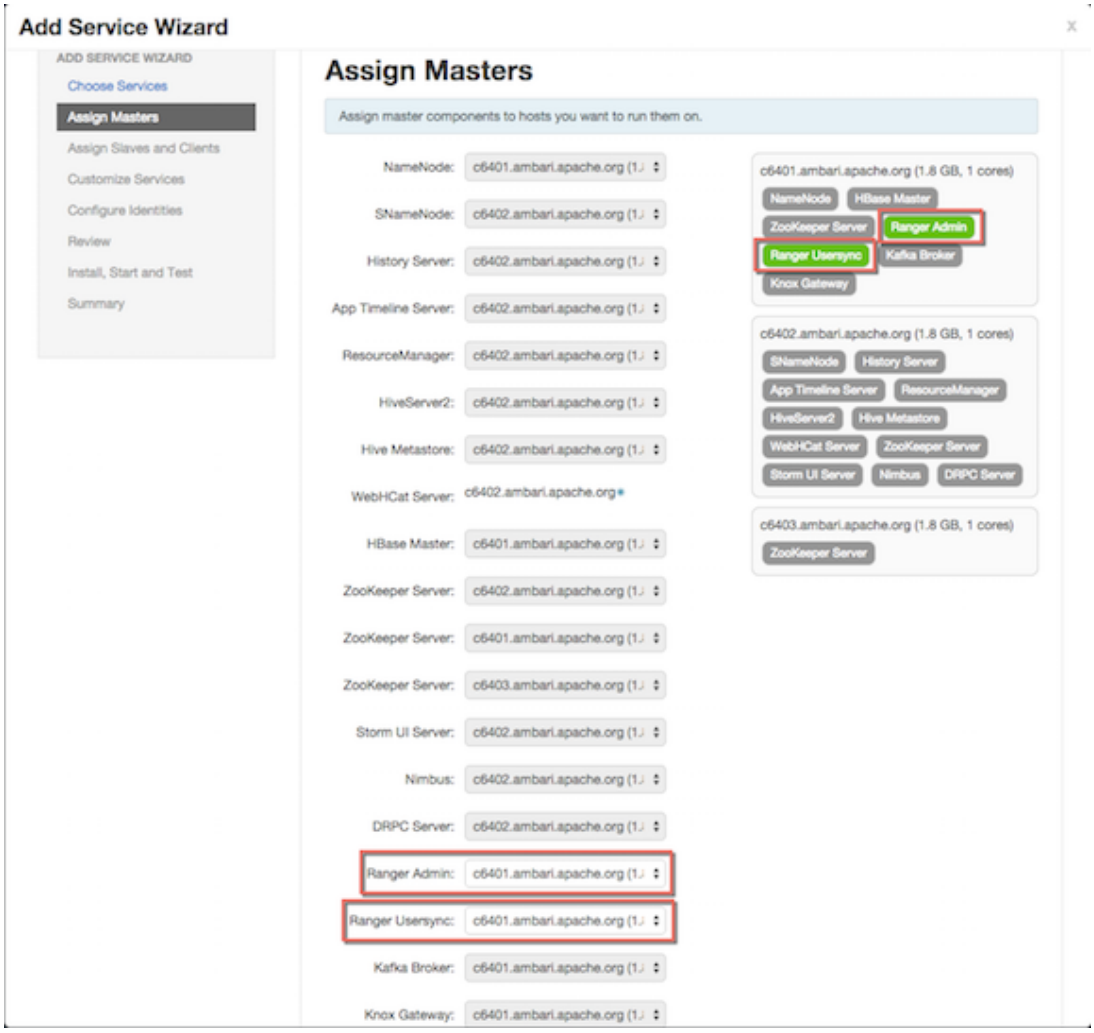
5. You are then prompted to select the host where Ranger Admin will be installed. This host should have DB admin access to the Ranger DB host and UserSync. Notice in the figure below that both the Ranger Admin and Ranger Usersync services will be installed on the primary node in the cluster (c6401.ambari.apache.org in the example shown below).

   Make a note of the Ranger Admin host for use in subsequent installation steps. Click **Next** when finished to continue with the installation.

Note

The Ranger Admin and Ranger Usersync services must be installed on the same cluster node.

Figure 3.5. Installing Ranger Assign Masters



6. The Customize Services page appears. These settings are described in the next section.

3.2. Customize Services

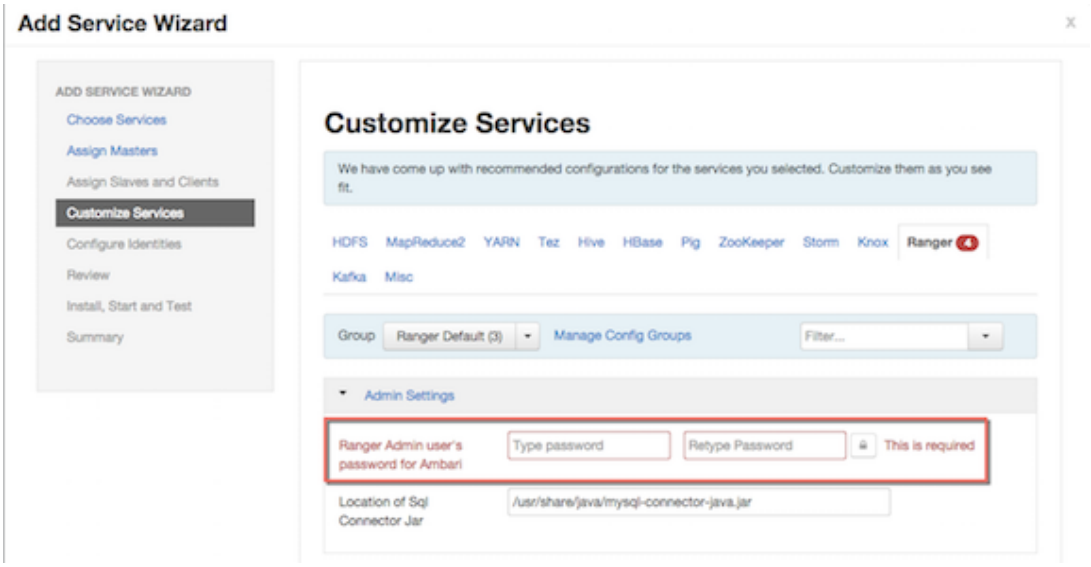
The next step in the installation process is to specify Ranger settings on the Customize Services page. You must specify all of the following settings on the Customize Services page before clicking **Next** at the bottom of the page to continue with the installation.

In this section:

- Configuring Usersync Settings [29]

3.2.1. Admin Settings

1. Under Admin Settings on the Customize Services page, type in the password for the user account used by Ambari. This password will only be used by the Ambari Agent, and will be used with the user name specified in the the Ranger configuration as `ranger_admin_username` under "Advanced ranger-env".



3.2.2. DB Settings

1. Under DB Settings on the Customize Services page, select the "DB Flavor" (installed database type) that you are using with Ranger. The "Location of SQL Connector Jar" box contains the path to the JDBC driver .jar file.

   A message box reminds you to make sure you have set the `jdbc/driver/path` based on the location of the JDBC driver .jar file for the installed Ranger database. If you have not already done so, set the JDBC driver path as described in the installation prerequisites.

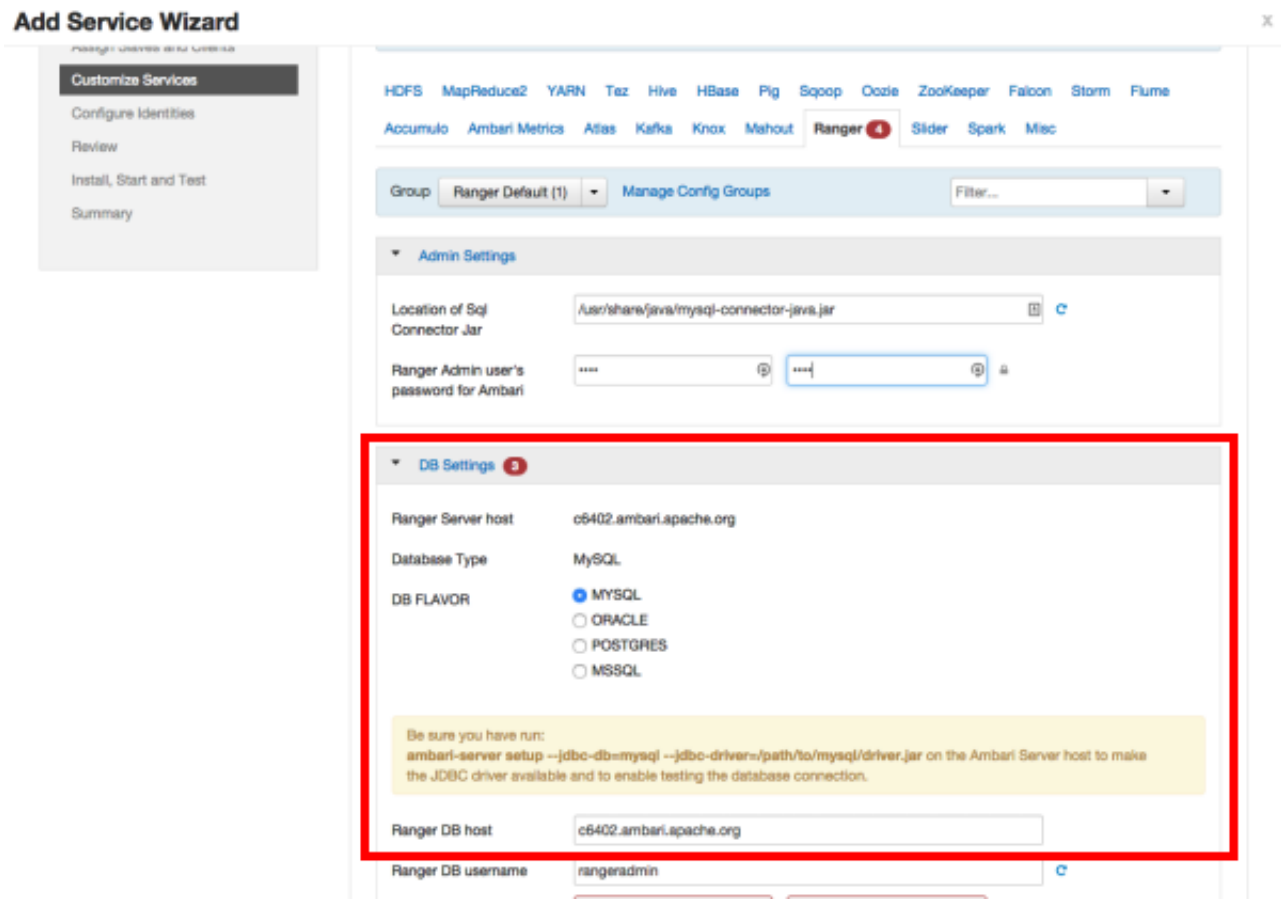


Table 3.1. Ranger DB Host

Database Type	Host	Example
MySQL	<HOST[:PORT]>	c6401.ambari.apache.org **or** c6401.ambari.apache.org:3306
Oracle	<HOST:PORT:SID>	c6401.ambari.apache.org:1521:ORCL
	<HOST:PORT/Service>	c6401.ambari.apache.org:1521/XE
PostgreSQL	<HOST[:PORT]>	c6401.ambari.apache.org **or** c6401.ambari.apache.org:5432
MS SQL	<HOST[:PORT]>	c6401.ambari.apache.org **or** c6401.ambari.apache.org:1433

2. Next, enter the user names and passwords for your Ranger database server. The following table describes these settings in more detail. You can use a MySQL, Oracle, or PostgreSQL database.

Note

The Ranger installation script requires DBA account credentials with privileges to create database (DB) users, and can assign privileges to DB users for READ/WRITE operations in Ranger Policy/Audit DB. However, if Ranger DB users are created before the Ranger installation, you do not need to provide the DB root user and password (you still need to enter some values due to Ambari UI validation, but they do not need to be the admin user details). See Pre-creating Ranger DB Users with the DBA Setup Script for information about pre-creating the users and DB using a separate step involving DB Admin. If users are pre-created, you should clear the "Setup DB and DB user" check box under "Advanced ranger-env" before proceeding with the installation.

Table 3.2. Ranger Database Settings

Configuration Property Name	Description	Default Value	Example Value	DB is automatically created by the Ranger installation?
DB host	The fully qualified domain name of the Ranger database server. For Oracle, Port/SID/ Service are added here as well. See the Ranger DB Host table above.		c6401.ambari.apache.org	Yes
Ranger DB root user	The Ranger database user that has administrative privileges to create database schemas and users.	root	root	Yes, if DB setup is not done
Ranger DB root password	The root password for the Ranger database user.	N/A	root	Yes, if DB setup is not done
Ranger DB name	The name of the Ranger Policy database. For Oracle the tablespace name should be given here.	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		PassWORd	Yes
Ranger Audit DB name	The name of the Ranger Audit database. This can be a different database in the same database. For Oracle the tablespace name should be given here.	ranger_audit	ranger_audit	Yes
Ranger Audit DB username	The username for the Ranger Audit database. This username performs	rangerlogger	rangerlogger	Yes

Configuration Property Name	Description	Default Value	Example Value	DB is automatically created by the Ranger installation?
	all audit logging operations.			
Ranger Audit DB password	The password for the Ranger Audit database.		rangerlogger	Yes

Note

For Oracle 11g Release 2 and Oracle 12c, the following format must be used for the Ranger DB host and the JDBC connect string:

- **Ranger DB host**

  Format:

  - If using a SID: `//hostname:port:SID`

    Example:

    `c6401.ambari.apache.org:1521:ORCL`

  - If using a service: `//hostname:port/SID`

    Example:

    `c6401.ambari.apache.org:1521/ORCL`

- **JDBC connect string**

  - If using a SID: `jdbc:oracle:thin@hostname:port:SID`

    Example:

    `jdbc:oracle:thin:@c6401.ambari.apache.org:1521:ORCL`

  - If using a service: `jdbc:oracle:thin@//hostname:port/Service`

    Example:

    `jdbc:oracle:thin:@//c6401.ambari.apache.org:1521/XE`

  Note that the Ambari UI will generate the string based on the value provided in the Ranger DB host. But currently it is generating the wrong connection string in certain cases (especially for Oracle DB). Hence you may need to replace the JDBC connection string as described above.

  In an Oracle DB environment, if the JDBC connect string has been overridden to complete the ranger installation, the audit JDBC URLs must also be manually updated in order for DB auditing to work properly.

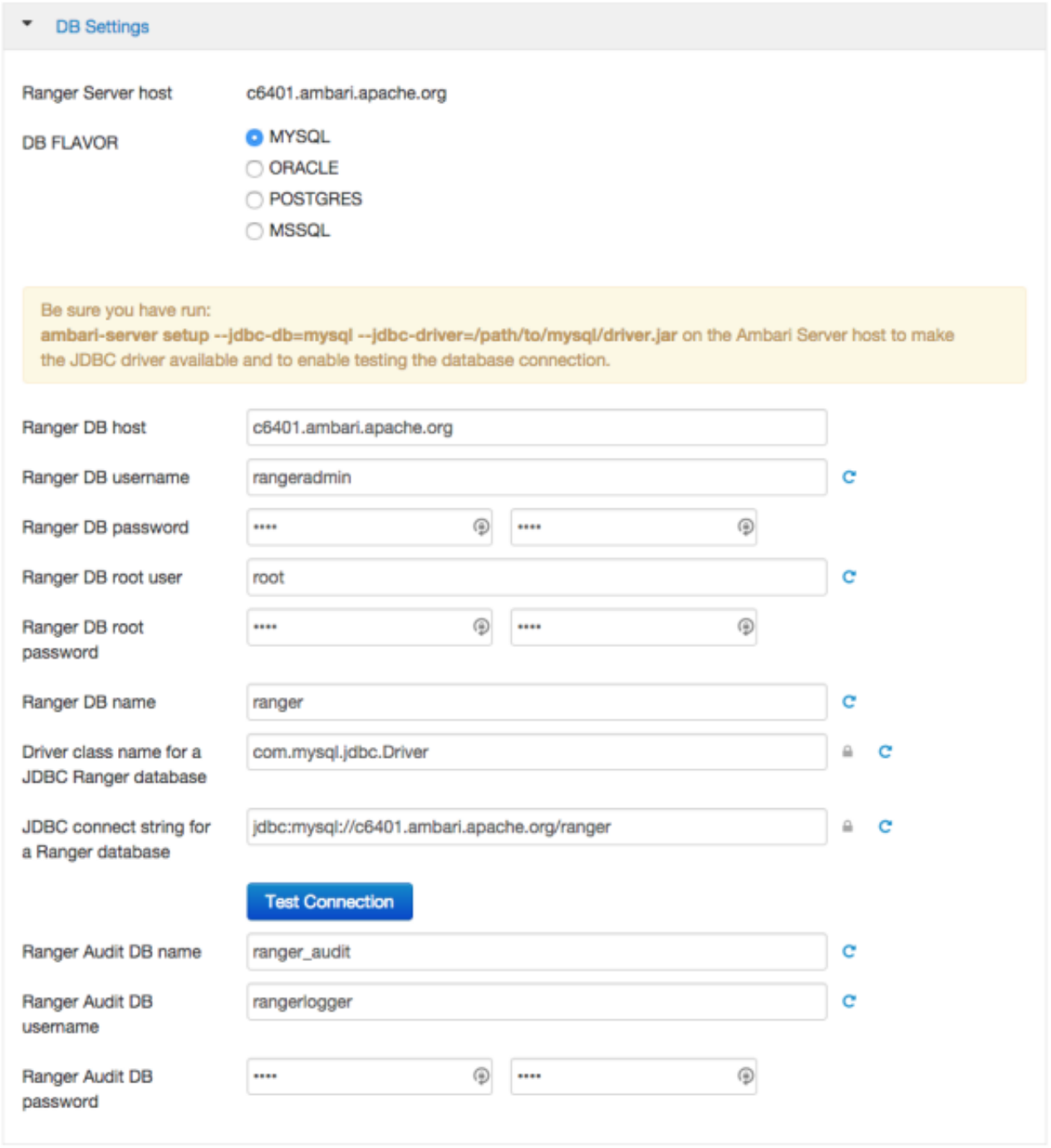
- `ranger.jpa.audit.jdbc.url` under "Advanced ranger-admin-site" (ranger admin setting)
- `xasecure.audit.destination.db.jdbc.url` under "Advanced ranger-<component>-audit" (ranger plugin setting)

The following images show examples of the DB Settings for each Ranger database type:

Note

To test the DB settings, click **Test Connection**. If a Ranger database has not been pre-installed, Test Connection will fail even for a valid configuration.

MySQL:

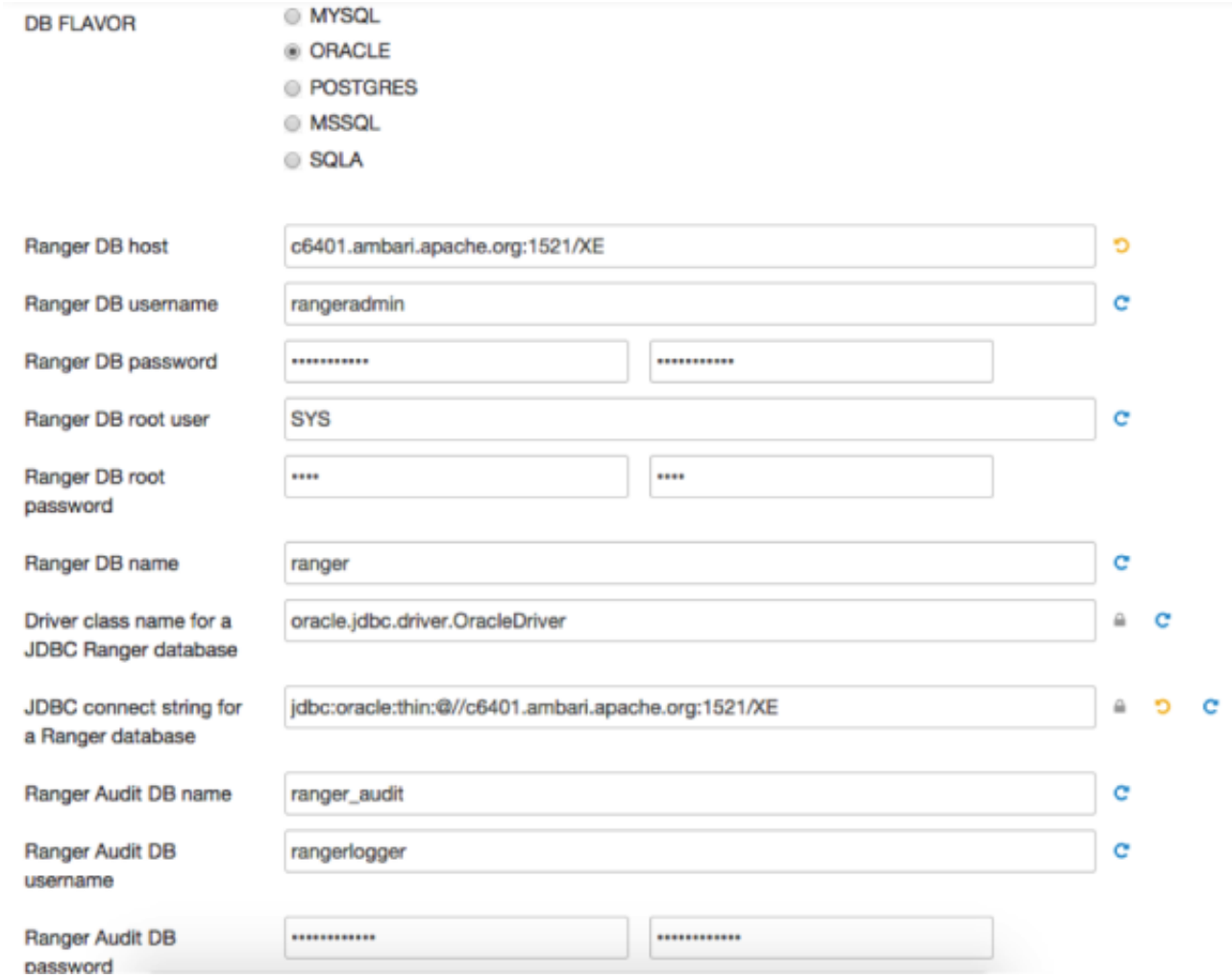


Oracle – if the Oracle instance is running with a service name:

Important

Note that the Ambari UI will generate the string based on the value provided in the Ranger DB host. But currently it is generating the wrong

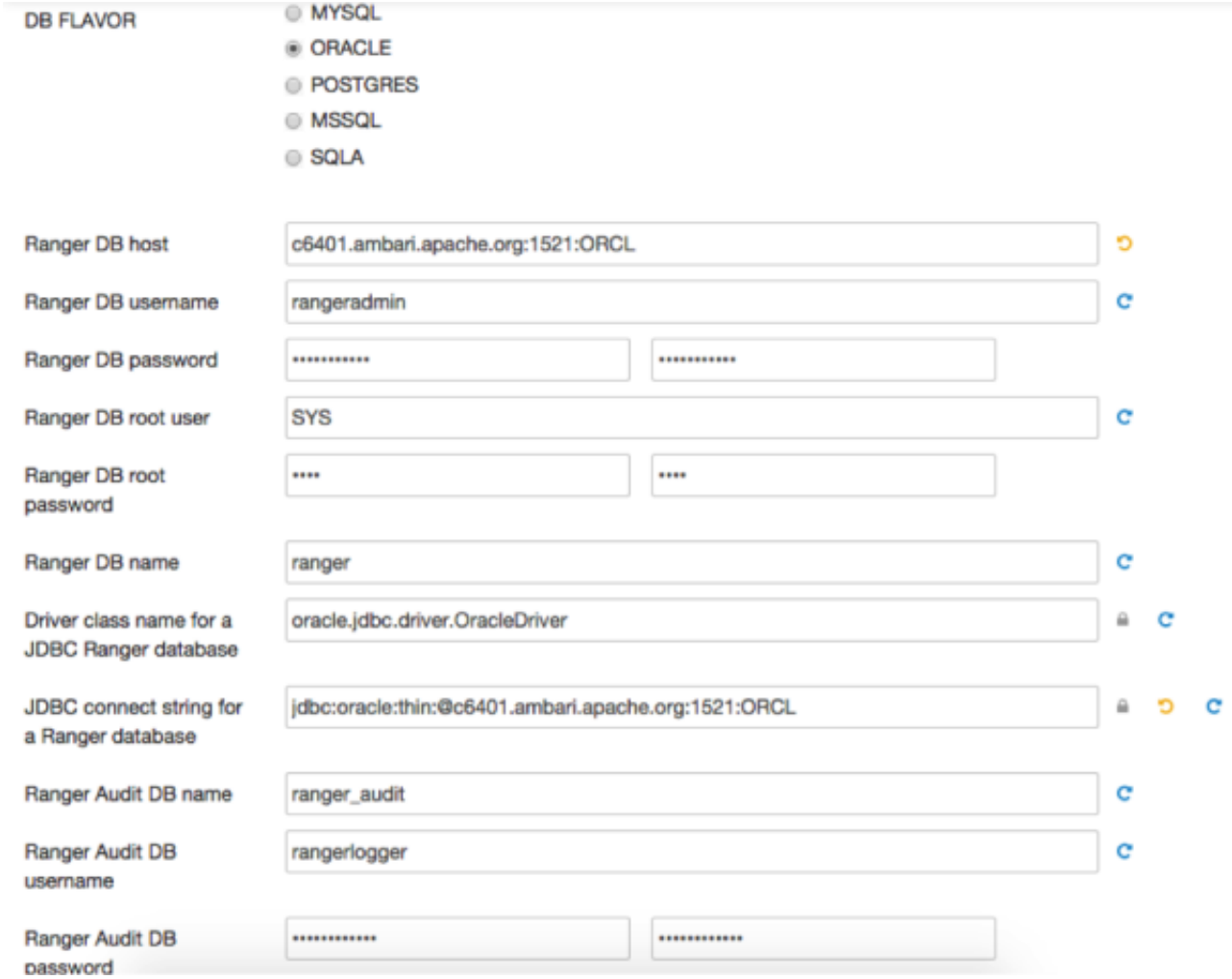
connection string for Oracle. Therefore you must replace the JDBC connection string as described above.



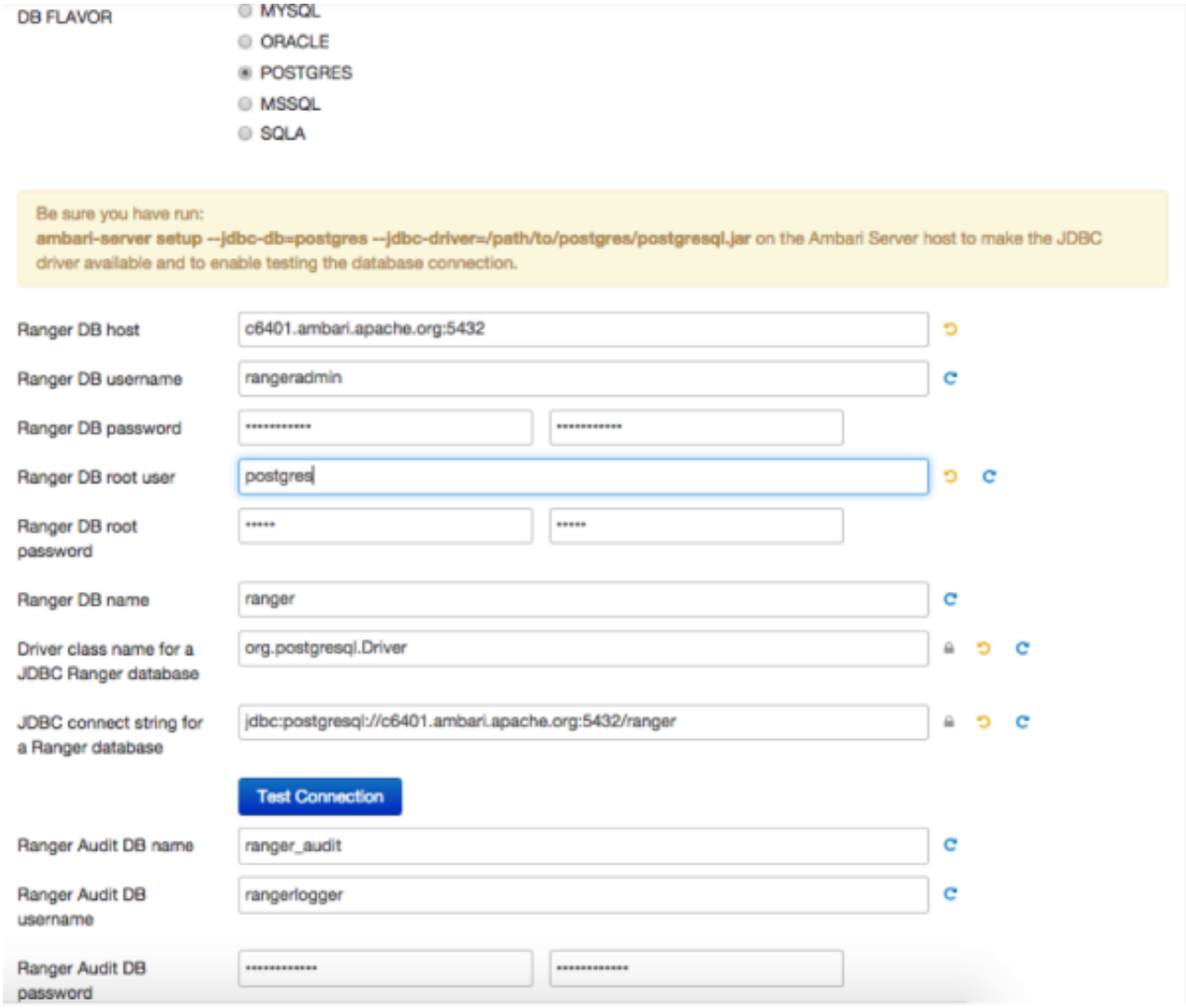
Oracle – if the Oracle instance is running with a SID:

Important

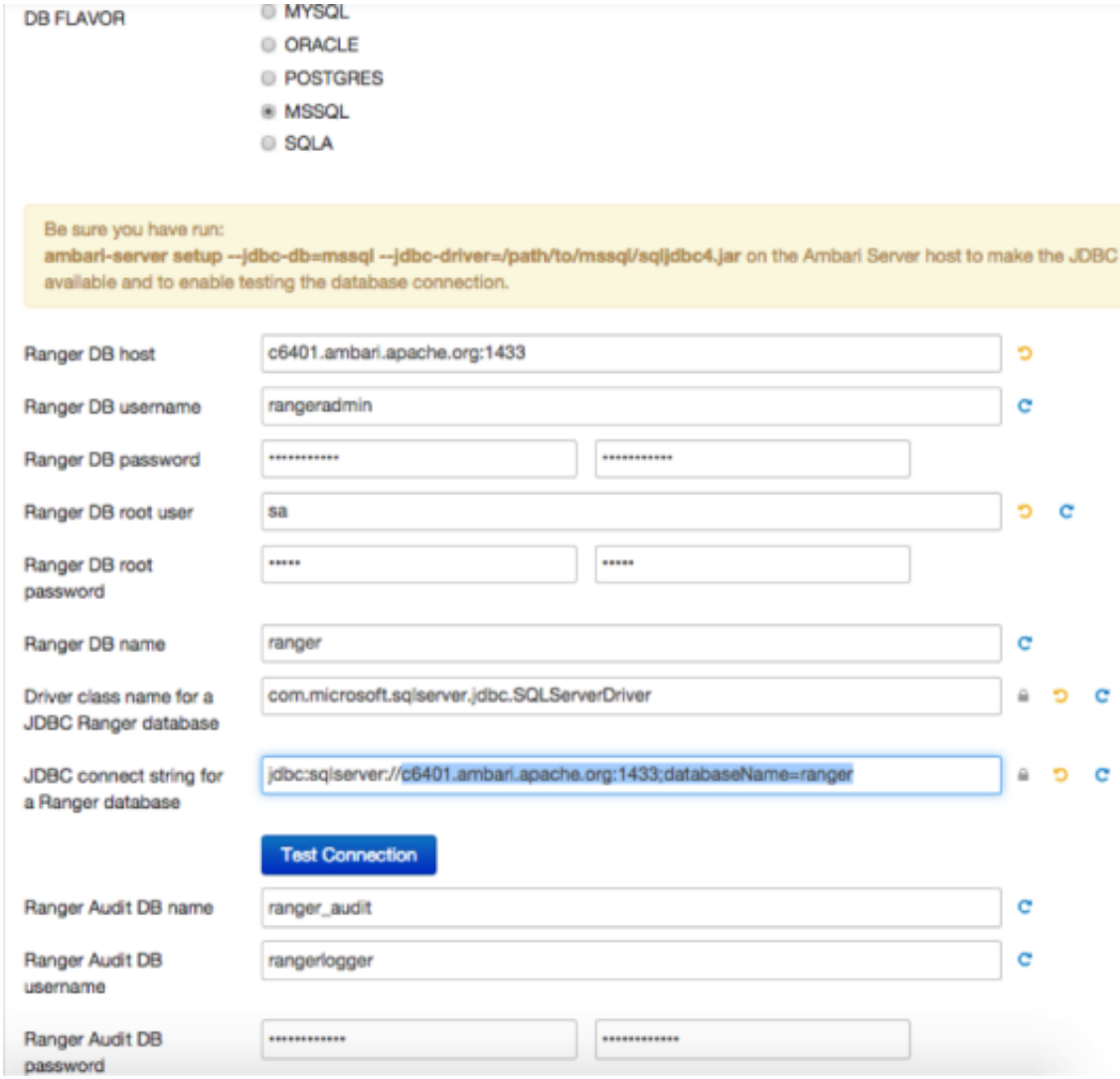
Note that the Ambari UI will generate the string based on the value provided in the Ranger DB host. But currently it is generating the wrong connection string for Oracle. Therefore you must replace the JDBC connection string as described above.


DB FLAVOR
MYSQL
ORACLE
POSTGRES
MSSQL
SQLA
Ranger DB host
c6401.ambari.apache.org:1521:ORCL
Ranger DB username
rangeradmin
Ranger DB password
Ranger DB root user
SYS
Ranger DB root password
Ranger DB name
ranger
Driver class name for a JDBC Ranger database
oracle.jdbc.driver.OracleDriver
JDBC connect string for a Ranger database
jdbc:oracle:thin:@c6401.ambari.apache.org:1521:ORCL
Ranger Audit DB name
ranger_audit
Ranger Audit DB username
rangerlogger
Ranger Audit DB password

PostgreSQL:


DB FLAVOR
MYSQL
ORACLE
POSTGRES
MSSQL
SQLA
Be sure you have run:
ambari-server setup --jdbc-db=postgres --jdbc-driver=/path/to/postgres/postgresql.jar on the Ambari Server host to make the JDBC driver available and to enable testing the database connection.
Ranger DB host
c6401.ambari.apache.org:5432
Ranger DB username
rangeradmin
Ranger DB password
Ranger DB root user
postgres
Ranger DB root password
Ranger DB name
ranger
Driver class name for a JDBC Ranger database
org.postgresql.Driver
JDBC connect string for a Ranger database
jdbc:postgresql://c6401.ambari.apache.org:5432/ranger
Test Connection
Ranger Audit DB name
ranger_audit
Ranger Audit DB username
rangerlogger
Ranger Audit DB password

MS SQL:



3.2.3. Configuring Ranger Settings

Once you have updated the DB Settings, you will then need to update your existing Ranger settings. The following figure shows the settings, and the table below describes each of these fields.

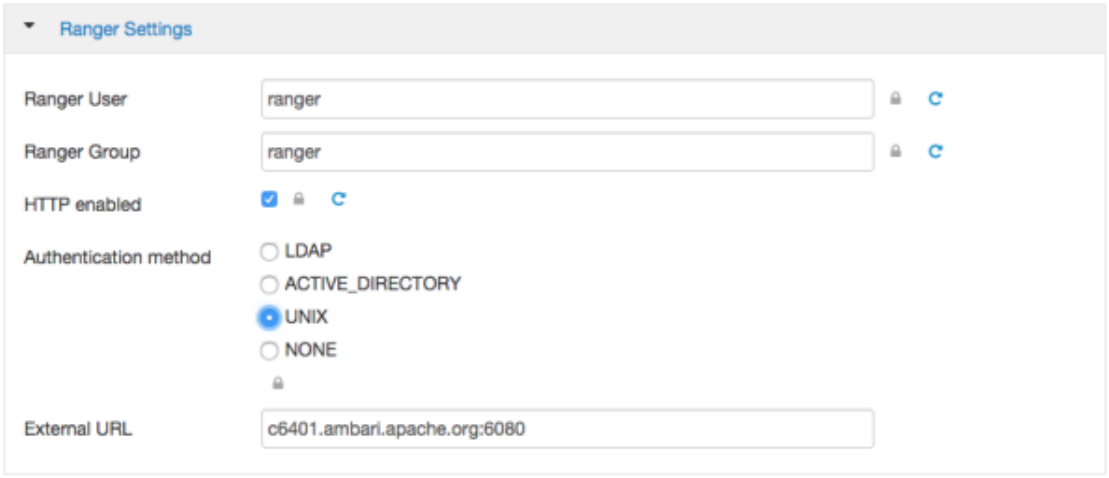


Table 3.3. Ranger Settings

Configuration Property	Description	Default Value	Example Value	Required?
Ranger User	The value used to create users and assign permissions. This is the OS level user that will be created and used to start the Ranger Admin and Ranger Usersync services.	ranger	ranger	Yes
Ranger Group	The value used to create groups and assign permissions. This is the OS level group that will be created and used to start the Ranger Admin and Ranger Usersync services.	ranger	ranger	Yes
HTTP Enabled	A check box that specifies whether or not HTTP authentication is enabled. If HTTP is not enabled, only HTTPS is allowed.	Selected	Selected	No
Authentication method	The type of authentication method used to log into the Policy Manager Only users created within the Policy Manager tool can log in. The available authentication methods are **LDAP**, **Active Directory**, **UNIX**, and **NONE**. If **NONE** is selected, Ranger uses the local user database for authentication, and only internal Ranger users can log in.	UNIX	None	Yes
External URL	The Ranger Policy Manager host.		http://<your_ranger_host>:6080	Yes

3.2.4. Configuring Ranger Authentication

3.2.4.1. UNIX Authentication Settings

The following figure shows the UNIX authentication settings, and the table below describes each of these properties.

Table 3.4. UNIX Authentication Settings

Configuration Property	Description	Default Value	Example Value	Required?
Allow remote Login	Flag to enable/ disable remote login via UNIX Authentication Mode.	TRUE	TRUE	No.
ranger.unixauth.service.hostname	The FQDN where the ranger-usersync module is running (along with the UNIX Authentication Service).	localhost	myunixhost.domain.com	Yes, if UNIX authentication is selected.
ranger.unixauth.service.port	The port number where the ranger-usersync module is running the UNIX Authentication Service.	5151	5151	Yes, if UNIX authentication is selected.

3.2.4.2. Active Directory Authentication Settings

This section describes how to configure settings for Active Directory authentication.

Note

In addition to these settings, you may also need to configure the Active Directory properties described in Configuring Usersync Settings.

3.2.4.2.1. AD Settings

The following figure shows the Active Directory (AD) authentication settings, and the table below describes each of these properties.

Table 3.5. Active Directory Authentication Settings

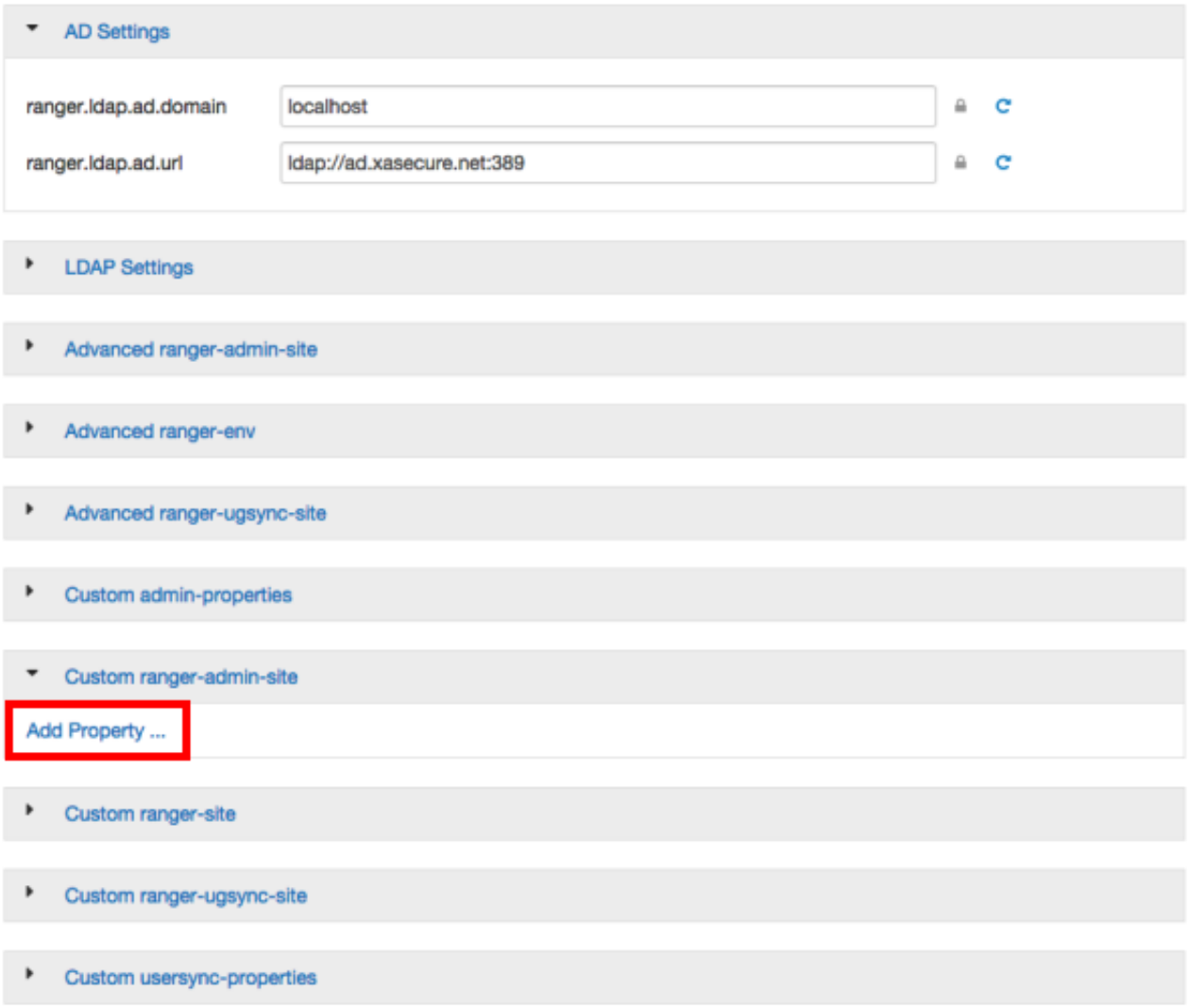
Configuration Property Name	Description	Default Value	Example Value	Required?
ranger.ldap.ad.domain	Server domain name (or IP address) where ranger-usersync module is running (along with the AD Authentication Service). The default value of "localhost" must be changed to the domain name.	localhost	example.com	Yes, if Active Directory authentication is selected.
ranger.ldap.ad.url	The URL and port number where ranger-usersync module is running the AD Authentication Service. The default value is a placeholder and must be changed to point to the AD server.	ldap://ad.xasecure.net:389	ldap://127.0.0.1:389	Yes, if Active Directory authentication is selected.

3.2.4.2.2. Custom ranger-admin-site Settings for Active Directory (Optional)

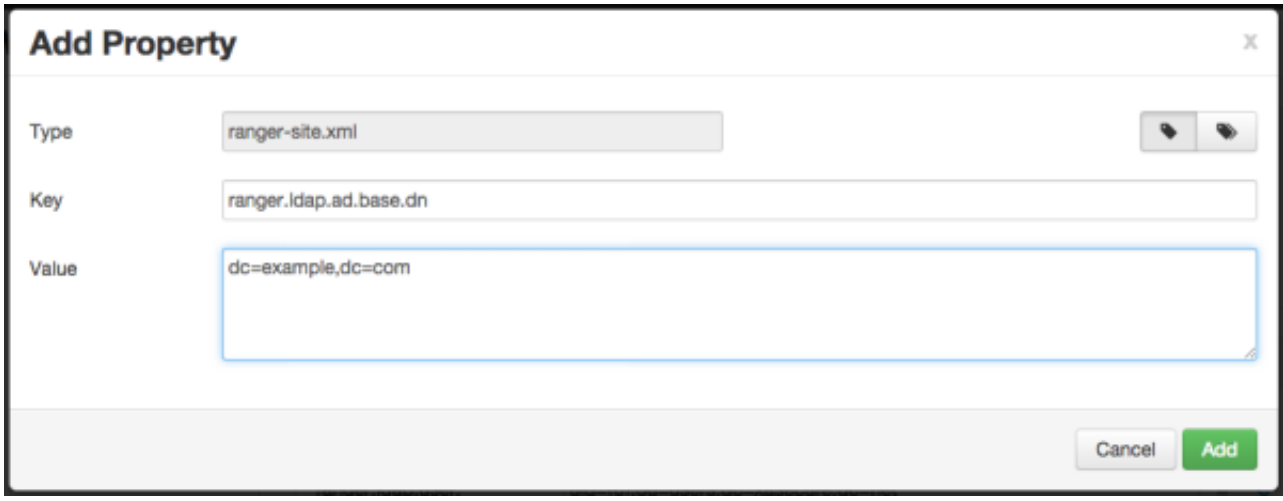
The following Custom ranger-admin-site settings for Active Directory authentication are optional.

To add a Custom ranger-admin-site property:

1. Select **Custom ranger-admin-site**, then click **Add Property**.



2. On the Add Property pop-up, type the property name in the **Key** box, type the property value in the **Value** box, then click **Add**.



The following figure shows the Custom ranger-admin-site settings required for Active Directory (AD) authentication, and the table below describes each of these properties.

Table 3.6. Active Directory Custom ranger-admin-site Settings

Custom Property Name	Sample Values for AD Authentication
ranger.ldap.ad.base.dn	dc=example,dc=com
ranger.ldap.ad.bind.dn	cn=adadmin,cn=Users,dc=example,dc=com
ranger.ldap.ad.bind.password	secret123!
ranger.ldap.ad.referral	follow | ignore | throw

There are three possible values for `ranger.ldap.ad.referral`: `follow`, `throw`, and `ignore`. The recommended setting is `follow`.

When searching a directory, the server might return several search results, along with a few continuation references that show where to obtain further results. These results and references might be interleaved at the protocol level.

- When this property is set to `follow`, the AD service provider processes all of the normal entries first, and then follows the continuation references.

- When this property is set to `throw`, all of the normal entries are returned in the enumeration first, before the `ReferralException` is thrown. By contrast, a "referral" error response is processed immediately when this property is set to `follow` or `throw`.
- When this property is set to `ignore`, it indicates that the server should return referral entries as ordinary entries (or plain text). This might return partial results for the search. In the case of AD, a `PartialResultException` is returned when referrals are encountered while search results are processed.

3.2.4.3. LDAP Authentications Settings

This section describes how to configure LDAP and Advanced ranger-ugsync-site settings for Active Directory authentication.

Note

In addition to these settings, you must also configure the LDAP properties described in Configuring Usersync Settings.

3.2.4.3.1. LDAP Settings

The following figure shows the LDAP authentication settings, and the table below describes each of these properties.

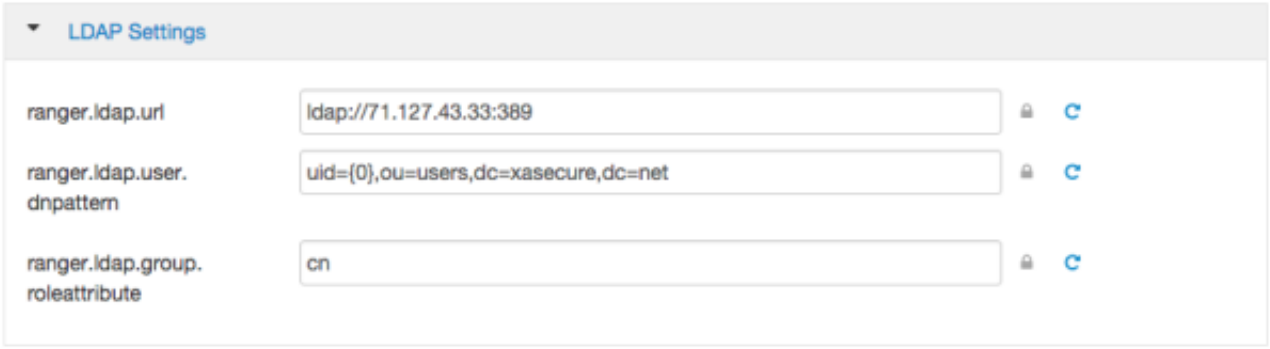


Table 3.7. LDAP Authentication Settings

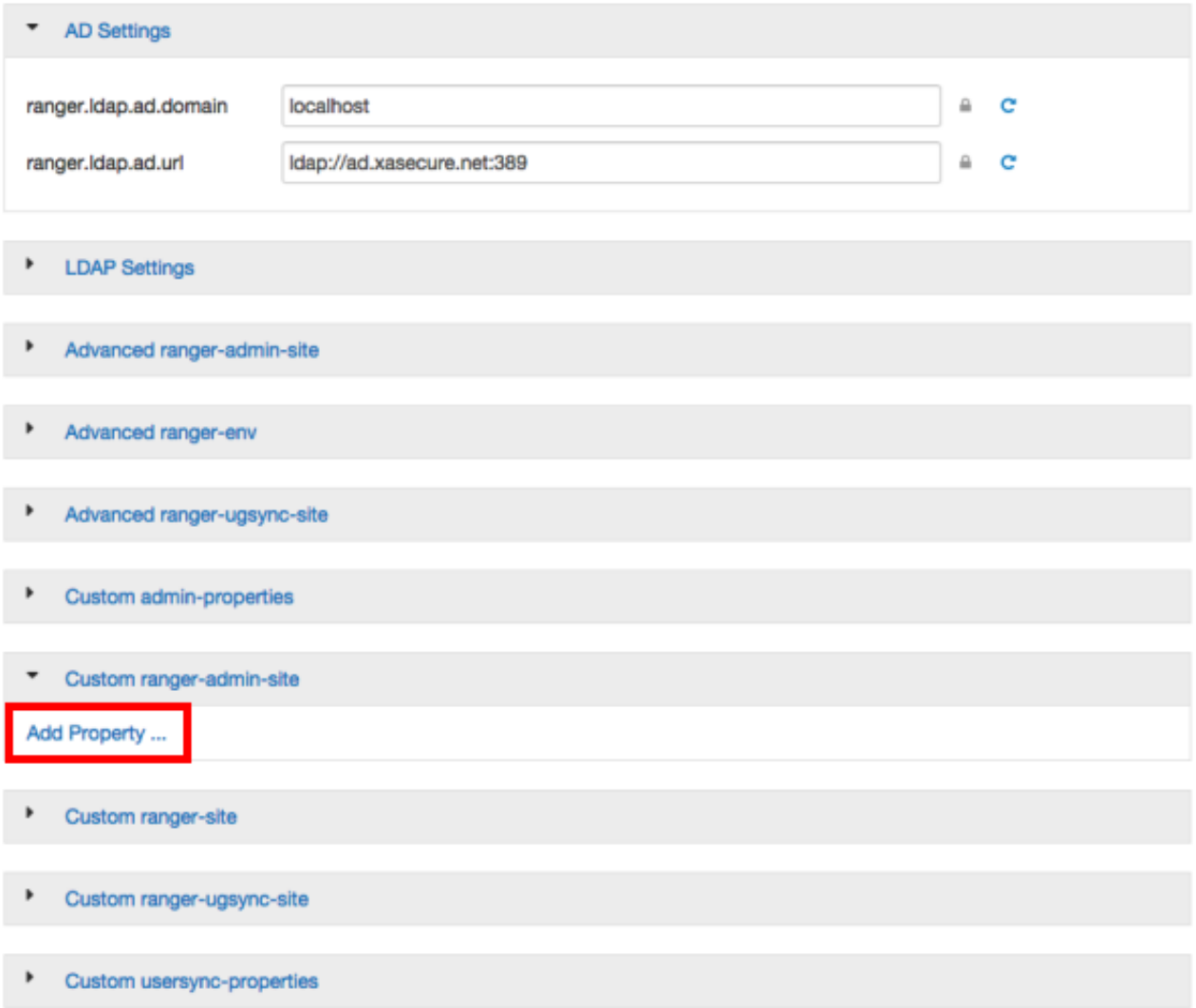
Configuration Property Name	Description	Default Value	Example Value	Required?
ranger.ldap.url	The URL and port number where ranger-usersync module is running the LDAP Authentication Service.	ldap://71.127.43.33:389	ldap://127.0.0.1:389	Yes, if LDAP authentication is selected.
ranger.ldap.user.dnpattern	The domain name pattern.	uid={0},ou=users, dc=xasecure,dc=net	cn=ldapadmin,ou=Users, dc=example,dc=com	Yes, if LDAP authentication is selected.
ranger.ldap.group.roleattribute	The LDAP group role attribute.	cn	cn	Yes, if LDAP authentication is selected.

3.2.4.3.2. Custom ranger-admin-site Settings for LDAP (Optional)

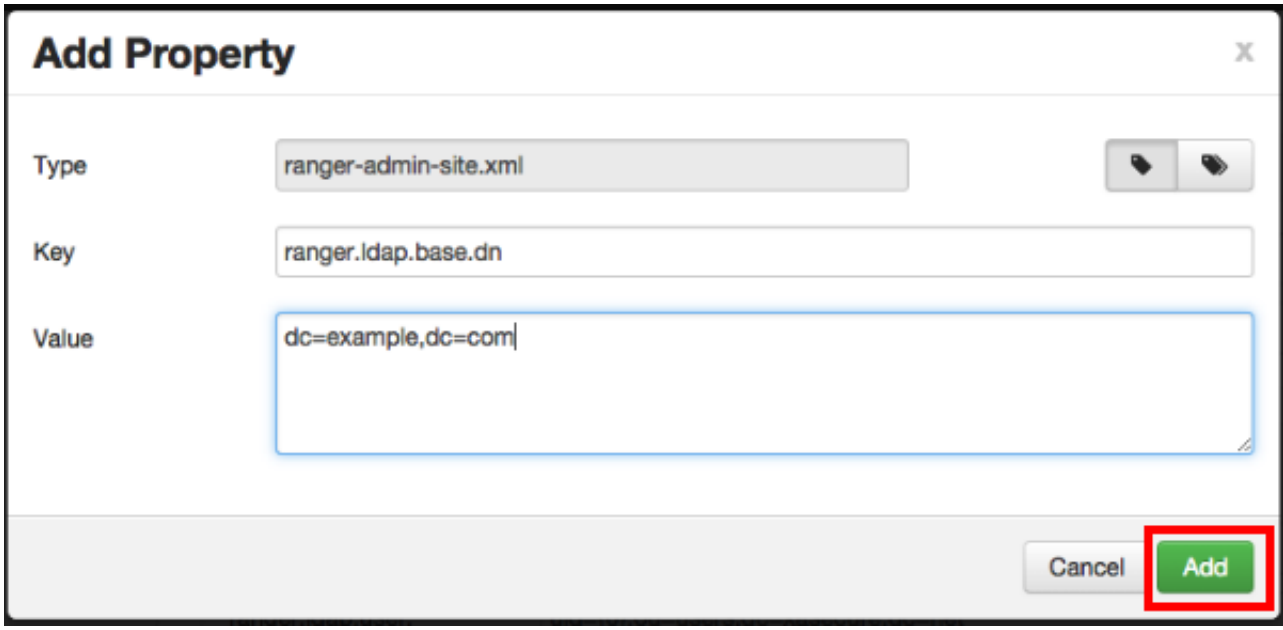
The following Custom ranger-admin-site settings for LDAP are optional.

To add a Custom ranger-admin-site property:

1. Select **Custom ranger-admin-site**, then click **Add Property**.



2. On the Add Property pop-up, type the property name in the **Key** box, type the property value in the **Value** box, then click **Add**.



The following figure shows the Custom ranger-admin-site settings required for LDAP authentication, and the table below describes each of these properties.

Table 3.8. LDAP Custom ranger-admin-site Settings

Custom Property Name	Sample Values for AD or LDAP Authentication
ranger.ldap.base.dn	dc=example,dc=com
ranger.ldap.bind.dn	cn=adadmin,cn=Users,dc=example,dc=com
ranger.ldap.bind.password	secret123!
ranger.ldap.referral	follow | ignore | throw

There are three possible values for `ranger.ldap.referral`: `follow`, `throw`, and `ignore`. The recommended setting is `follow`.

When searching a directory, the server might return several search results, along with a few continuation references that show where to obtain further results. These results and references might be interleaved at the protocol level.

- When this property is set to `follow`, the LDAP service provider processes all of the normal entries first, and then follows the continuation references.
- When this property is set to `throw`, all of the normal entries are returned in the enumeration first, before the `ReferralException` is thrown. By contrast, a "referral" error response is processed immediately when this property is set to `follow` or `throw`.
- When this property is set to `ignore`, it indicates that the server should return referral entries as ordinary entries (or plain text). This might return partial results for the search.

3.2.4.3.3. Advanced ranger-admin-site Settings

The following Advanced ranger-admin-site properties apply only to LDAP authentication.

Table 3.9. Active Directory Authentication Settings

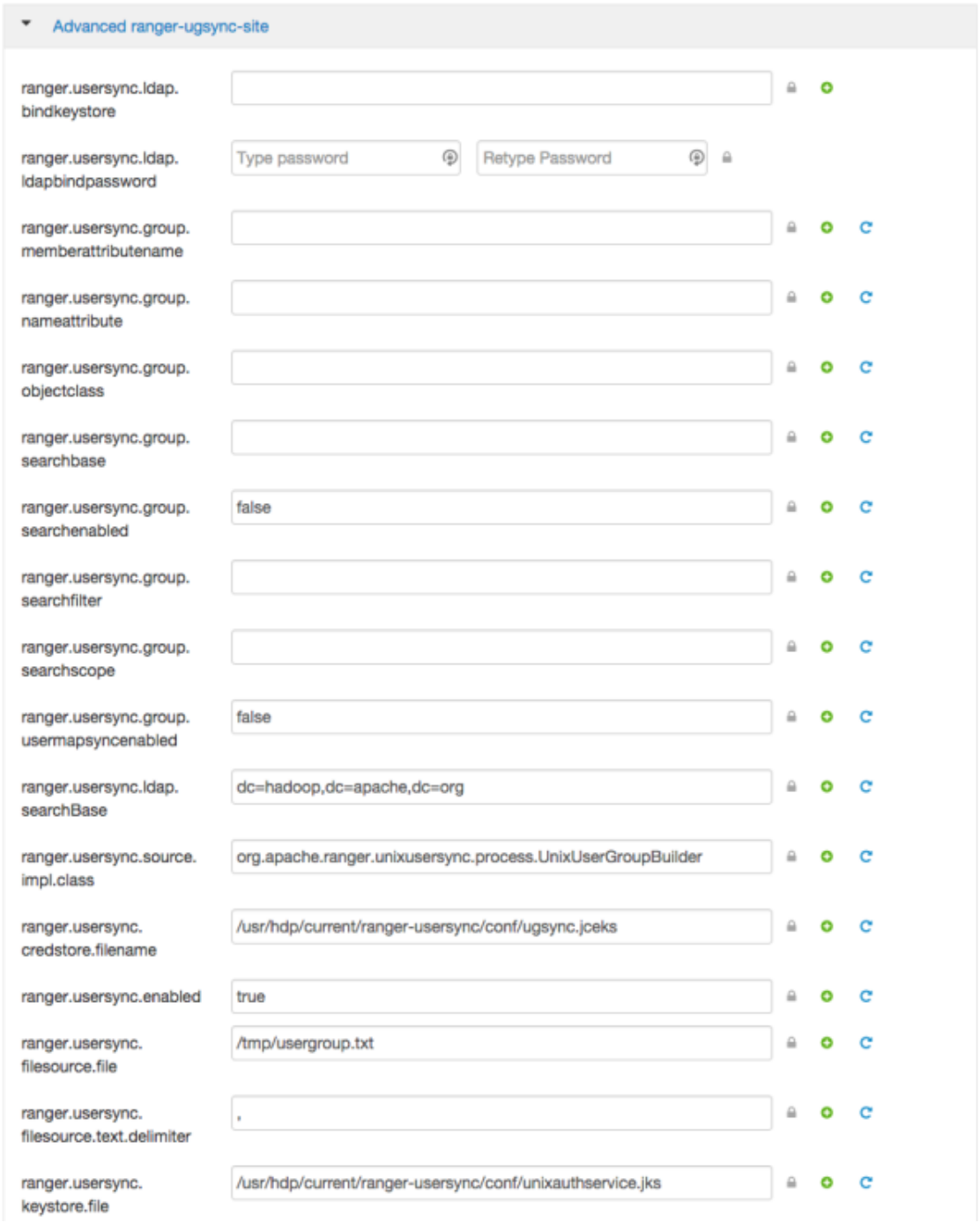
Property Name	Sample values for LDAP Authentication
ranger.ldap.group.searchbase	dc=example,dc=com
ranger.ldap.group.searchfilter	(member=cn={0},ou=Users,dc=example,dc=com)

3.2.5. Configuring Usersync Settings

Usersync pulls in users from UNIX, LDAP, or AD and populates Ranger's local user tables with these users.

3.2.5.1. UNIX Usersync Settings

If you are using UNIX authentication, the default values for the Advanced ranger-ugsync-site properties are the settings for UNIX authentication.



3.2.5.2. Required LDAP and AD Usersync Settings

If you are using LDAP authentication, you must update the following Advanced ranger-ugsync-site properties.

Table 3.10. LDAP Advanced ranger-ugsync-site Settings

Property Name	LDAP Value
ranger.usersync.ldap.bindkeystore	Set this to the same value as the `ranger.usersync.credstore.filename` property, i.e, the default value is `/usr/hdp/current/ranger-usersync/conf/ugsync.jceks`
ranger.usersync.ldap.bindalias	ranger.usersync.ldap.bindalias
ranger.usersync.source.impl.class	ldap

Table 3.11. AD Advanced ranger-ugsync-site Settings

Property Name	LDAP Value
ranger.usersync.source.impl.class	ldap

3.2.5.3. Additional LDAP and AD Usersync Settings

If you are using LDAP or Active Directory authentication, you may need to update the following properties, depending upon your specific deployment characteristics.

Table 3.12. Advanced ranger-ugsync-site Settings for LDAP and AD

Property Name	LDAP ranger-ugsync-site Value	AD ranger-ugsync-site Value
ranger.usersync.ldap.url	ldap://127.0.0.1:389	ldap://ad-conrowoller-hostname:389
ranger.usersync.ldap.binddn	cn=ldapadmin,ou=users, dc=example,dc=com	cn=adadmin,cn=Users, dc=example,dc=com
ranger.usersync.ldap.ldapbindpassword	secret	secret
ranger.usersync.ldap.searchBase	dc=example,dc=com	dc=example,dc=com
ranger.usersync.source.impl.class	org.apache.ranger. ladpusersync. process.LdapUserGroupBuilder	
ranger.usersync.ldap.user.searchbase	ou=users, dc=example, dc=com	dc=example,dc=com
ranger.usersync.ldap.user.searchscope	sub	sub
ranger.usersync.ldap.user.objectclass	person	person
ranger.usersync.ldap.user.searchfilter	Set to single empty space if no value. Do not leave it as “empty”	(objectcategory=person)
ranger.usersync.ldap.user.nameattribute	uid or cn	sAMAccountName
ranger.usersync.ldap.user.groupnameattribute	memberof,ismemberof	memberof,ismemberof
ranger.usersync.ldap.username.caseconversion	none	none
ranger.usersync.ldap.groupname.caseconversion	none	none
ranger.usersync.group.searchenabled *	false	false
ranger.usersync.group.usermapsyncenabled *	false	false

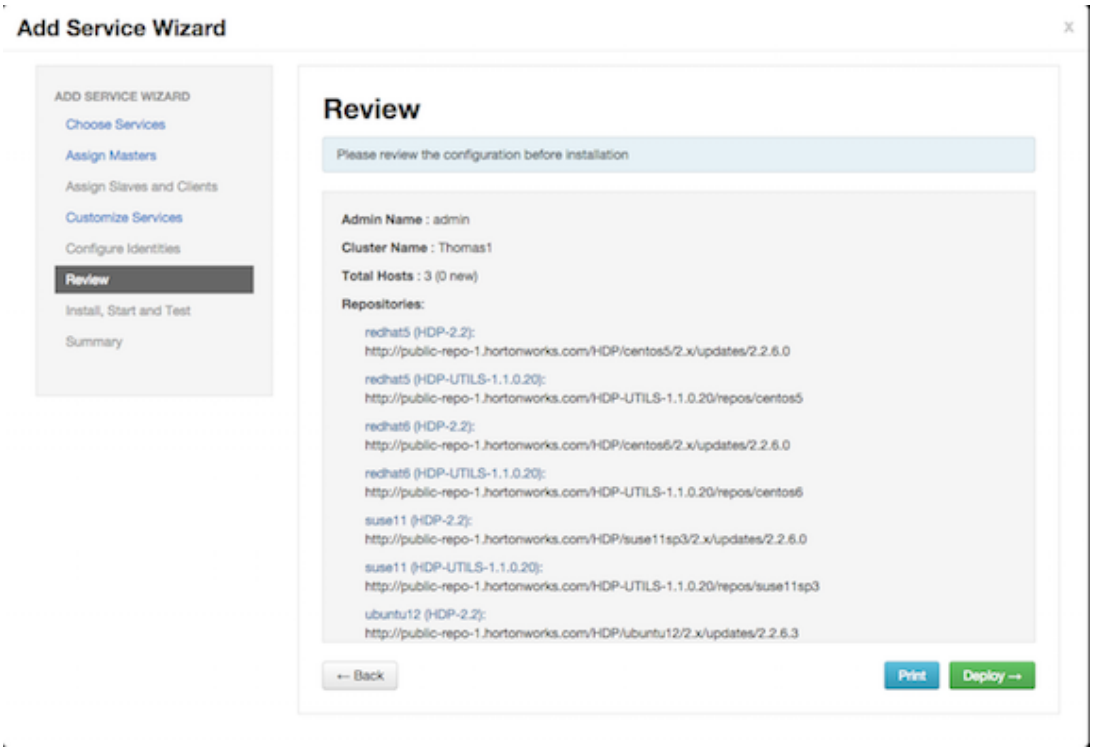
Property Name	LDAP ranger-ugsync-site Value	AD ranger-ugsync-site Value
ranger.usersync.group.searchbase *	ou=groups, dc=example, dc=com	dc=example,dc=com
ranger.usersync.group.searchscope *	sub	sub
ranger.usersync.group.objectclass *	groupofnames	groupofnames
ranger.usersync.group.searchfilter *	needed for AD authentication	(member=CN={0}, OU=MyUsers, DC=AD-HDP, DC=COM)
ranger.usersync.group.nameattribute *	cn	cn
ranger.usersync.group.memberattributename *	member	member
ranger.usersync.pagedresultsenabled *	true	true
ranger.usersync.pagedresultssize *	500	500

* Only applies when you want to filter out groups.

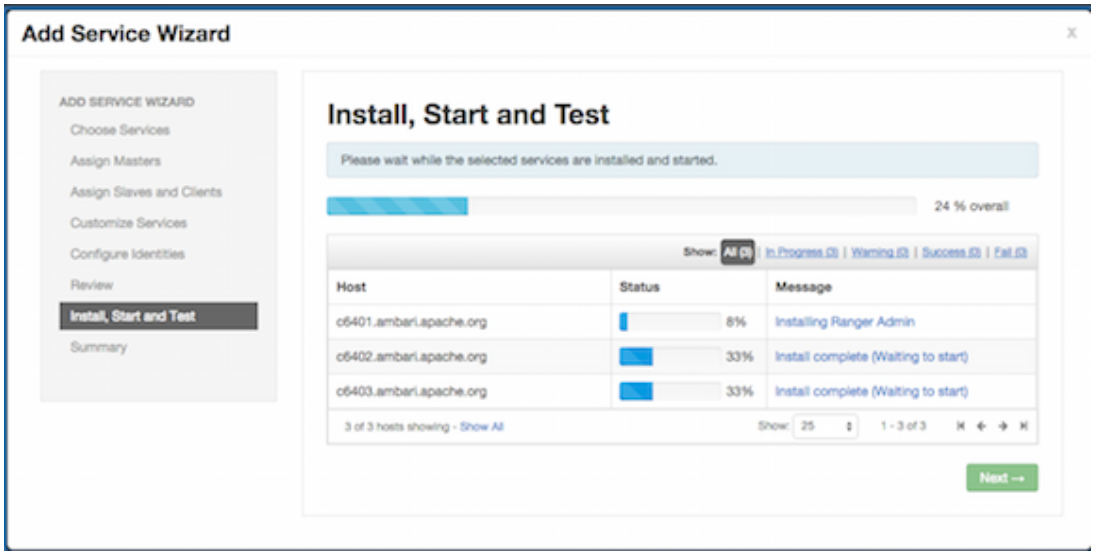
After you have finished specifying all of the settings on the Customize Services page, click **Next** at the bottom of the page to continue with the installation.

3.3. Complete the Ranger Installation

1. On the Review page, carefully review all of your settings and configurations. If everything looks good, click **Deploy** to install Ranger on the Ambari server.



2. When you click **Deploy**, Ranger is installed on the specified host on your Ambari server. A progress bar displays the installation progress.



3. When the installation is complete, a Summary page displays the installation details.

Note

If the installation fails, you should complete the installation process, then reconfigure and reinstall Ranger.

3.4. Configuring Ranger for LDAP SSL

3.4.1. Import the LDAP Cert into the Default Java TrustStore

1. If you are using a CA signed certificate for your LDAP authentication, the certificate should already be included in the default Java trustStore located at `$JAVA_HOME/jre/lib/security/cacerts` on all of your nodes, or at least on the NameNode and Ranger Admin/Usersync nodes.

2. There is no need to manually restart Ranger or perform any keytool imports.

3. If necessary you can import the CA cert to `$JAVA_HOME/jre/lib/security/cacerts`. If you are using a self-signed cert you can use the keytool to import it into `$JAVA_HOME/jre/lib/security/cacerts`.

3.4.2. Alternative Option

You can also use the following method when the self-signed cert is not in `$JAVA_HOME/jre/lib/security/cacerts`.

For Ranger Usersync:

1. Edit `/usr/hdp/current/ranger-usersync/ranger-usersync-services.sh`.

2. Add `java option > -Djavax.net.ssl.trustStore=/<path to the cacert>`.

For Ranger Admin:

1. Edit `/usr/hdp/current/ranger-admin/ews/ranger-admin-services.sh`.
2. Add parameter `-Djavax.net.ssl.trustStore=/<path to the cacert>` to the Java call in the script.

3.5. Pre-creating Ranger DB Users with the DBA Setup Script

You can set up Ranger users using a Hortonworks custom database script. The purpose of this script is to set up database (DB) users in environments where there is a separate database administrator managing the databases, and you do not want to provide database admininstrator credentials to Ranger for creating the database users.

To pre-create Ranger DB users using the `dba_script.py` script:

1. Download the Ranger rpm using the yum install command.

```
yum install ranger-admin
```

2. You should see one file named `dba_script.py` in the `/usr/hdp/current/ranger-admin` directory.
3. Execute the script by running the following command:

```
python dba_script.py
```

4. Pass all values required in the argument. These should include `db flavor`, `JDBC jar`, `db host`, `db name`, `db user`, and other parameters.

Note

If you would prefer not to pass runtime arguments, then simply update the `install.properties` file and then run the `python dba_script.py -q`. If you specify `-q` in a given argument, then the script will read all required information from the `install.properties` file

Note

If you would prefer to review the DDL SQL statements in the `dba_script.py` before executing them, run `python dba_script.py -d <path to save sql file>`. The `-d` option will save the SQL statements into the specified file. These statements can then be reviewed, customized by the DBA, and executed. If you use the `-d` option, you must alter the users' default tablespace to the correct one by executing the following statements. Replace the user names and tablespace with your actual values.

```
alter user ranger_admin_user default tablespace ranger_admin_db;
alter user ranger_audit_user  default tablespace ranger_audit_db;
```

Note

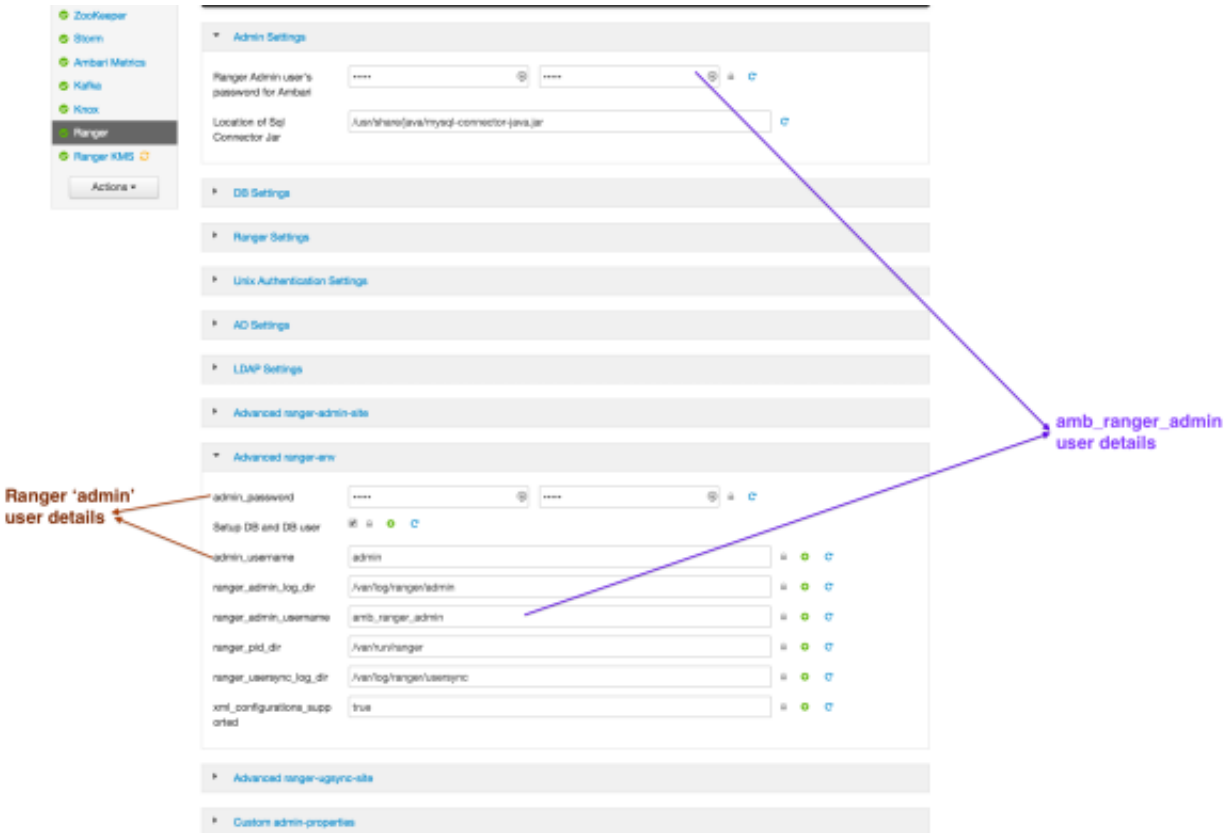
If DB users are pre-created using the dba_script.py as described above, you must clear the **Setup DB and DB user** check box under "Advanced ranger-env" before proceeding with the installation.

3.6. Updating Ranger Admin Passwords

For the following users, if you update the passwords on the Ranger Configs page, you must also update the passwords on the Configs page of each Ambari component that has the Ranger plugin enabled. Individual Ambari component configurations are not automatically updated – the service restart will fail if you do not update these passwords on each component.

- Ranger Admin user – The credentials for this user are set in **Configs > Advanced ranger-env** in the fields labeled **admin_username** (default value: `admin`) and **admin_password** (default value: `admin`).
- Admin user used by Ambari to create repo/policies – The user name for this user is set in **Configs > Advanced ranger-env** in the field labeled **ranger_admin_username** (default value: `amb_ranger_admin`). The password for this user is set in **Configs > Admin Settings** in the field labeled **Ranger Admin user's password for Ambari**. This password is specified during the Ranger installation.

The following image shows the location of these settings on the Ranger Configs page:



4. Using Apache Solr for Ranger Audits

Apache Solr is an open-source enterprise search platform. Apache Ranger can use Apache Solr to store audit logs, and Solr can also to provide a search capability of the audit logs through the Ranger Admin UI.

Important

Solr must be installed and configured before installing RangerAdmin or any of the Ranger component plugins.

It is recommended that Ranger audits be written to both Solr and HDFS. Audits to Solr are primarily used to enable search queries from the Ranger Admin UI. HDFS is a long-term destination for audits – audits stored in HDFS can be exported to any SIEM system, or to another audit store.

Configuration Options

- Solr Standalone – Solr Standalone is only recommended for testing and evaluation. Solr Standalone is a single instance of Solr that does not require ZooKeeper.
- SolrCloud – This is the recommended configuration for Ranger. SolrCloud is a scalable architecture that can run as single node or as a multi-node cluster. It includes features such as replication and sharding, which are useful for high availability (HA) and scalability. With SolrCloud, you need to plan the deployment based on the cluster size.

The following sections describe how to install and configure Apache Solr for Ranger Audits:

4.1. Prerequisites

Solr Prerequisites

- Ranger supports Apache Solr 5.2 or higher.
- Apache Solr requires the Java Runtime Environment (JRE) version 1.7 or higher.
- 1 TB free space in the volume where Solr will store the index data.
- 32 GB RAM.

SolrCloud Prerequisites

- SolrCloud supports replication and sharding. It is highly recommended that you use SolrCloud with at least two Solr nodes running on different servers with replication enabled.

- SolrCloud requires Apache ZooKeeper.

4.2. Installing Solr

The recommended method for installing Solr is using the HDP Search Installer. Optionally, you can install Solr using the Solr for Ranger setup script.

4.2.1. Installing Solr with the HDP Search Installer (Recommended)

To install Solr, use the following command to run the HDP Search installer:

```
yum install lucidworks-hdpsearch
```

The HDP Search installer installs Solr in the `/opt/lucidworks-hdpsearch/solr` directory.

4.2.2. Installing Solr with the Setup Script (Optional)

To install Solr using the Solr for Ranger setup script, set the following properties in the `install.properties` file before running the `setup.sh` script described in the next two sections, Configuring Solr Standalone and Configuring Solr Cloud.

Note

If you have installed Solr using the HDP Search installer, there is no need to set these properties.

Table 4.1. Solr install.properties Values

Property Name	Value	Description
SOLR_INSTALL	true	If this is set to `true`, the setup.sh script will download and install the Solr package specified with `SOLR_DOWNLOAD_URL`.
SOLR_DOWNLOAD_URL	http://archive.apache.org/dist/lucene/solr/5.2.1/solr-5.2.1.tgz	The Solr download URL.
SOLR_INSTALL_FOLDER	/opt/solr	The Solr installation directory.

4.3. Configuring Solr Standalone

Use the following procedure to configure Solr Standalone.

1. Download the `solr_for_audit_setup_v3` file to the `/usr/local/` directory:

   ```
   wget https://issues.apache.org/jira/secure/attachment/12761323/
   solr_for_audit_setup_v3.tgz -O /usr/local/solr_for_audit_setup_v3.tgz
   ```

2. Use the following commands to switch to the `/usr/local/` directory and extract the `solr_for_audit_setup_v3` file.

```
cd /usr/local
tar xvf solr_for_audit_setup_v3.tgz
```

The contents of the .tgz file will be extracted into a `/usr/local/solr_for_audit_setup_v3` directory.

3. Use the following command to switch to the `/usr/local/solr_for_audit_setup_v3` directory.

```
cd /usr/local/solr_for_audit_setup
```

4. Use the following command to open the `install.properties` file in the vi text editor.

```
vi install.properties
```

Set the following property values, then save the changes to the `install.properties` file.

Table 4.2. Solr install.properties Values

Property Name	Value	Description
JAVA_HOME	<path_to_jdk>, for example: `/usr/jdk64/jdk1.8.0_60`	Provide the path to the JDK install folder. For Hadoop, you can check `/etc/hadoop/conf/hadoop-env.sh` for the value of JAVA_HOME. As noted previously, Solr only supports JDK 1.7 and higher.
SOLR_USER	solr	The Linux user used to run Solr.
SOLR_INSTALL_FOLDER	/opt/lucidworks-hdpsearch/solr	The Solr installation directory.
SOLR_RANGER_HOME	/opt/lucidworks-hdpsearch/solr/ranger_audit_server	The location where the Ranger-related configuration and schema files will be copied.
SOLR_RANGER_PORT	6083	The Solr port for Ranger.
SOLR_DEPLOYMENT	standalone	The deployment type.
SOLR_RANGER_DATA_FOLDER	/opt/lucidworks-hdpsearch/solr/ranger_audit_server/data	The folder where the index data will be stored. The volume for this folder should have at least 1 TB free space for the index data, and should be backed up regularly.
SOLR_LOG_FOLDER	/var/log/solr/ranger_audits	The folder for the Solr log files.
SOLR_MAX_MEM	2g	The memory allocation for Solr.

5. Use the following command to run the Solr for Ranger setup script.

```
./setup.sh
```

6. To start Solr, log in as the `solr` or `root` user and run the following command.

```
/opt/lucidworks-hdpsearch/solr/ranger_audit_server/scripts/start_solr.sh
```

When Solr starts, a confirmation message appears.

```
Started Solr server on port 6083 (pid=). Happy searching!
```

7. You can use a web browser to open the Solr Admin Console at the following address:

```
http:<solr_host>:6083/solr
```

Note

You can use the following command to stop Solr:

```
/opt/lucidworks-hdpsearch/solr/ranger_audit_server/scripts/
stop_solr.sh
```

4.4. Configuring SolrCloud

Use the following procedure to configure SolrCloud.

1. Download the `solr_for_audit_setup_v3` file to the `/usr/local/` directory:

```
wget https://issues.apache.org/jira/secure/attachment/12761323/
solr_for_audit_setup_v3.tgz -O /usr/local/solr_for_audit_setup_v3.tgz
```

2. Use the following commands to switch to the `/usr/local/` directory and extract the `solr_for_audit_setup_v3` file.

```
cd /usr/local
tar xvf solr_for_audit_setup_v3.tgz
```

   The contents of the .tgz file will be extracted into a `/usr/local/solr_for_audit_setup_v3` directory.

3. Use the following command to switch to the `/usr/local/solr_for_audit_setup_v3` directory.

```
cd /usr/local/solr_for_audit_setup
```

4. Use the following command to open the `install.properties` file in the vi text editor.

```
vi install.properties
```

   Set the following property values, then save the changes to the `install.properties` file.

Table 4.3. Solr install.properties Values

Property Name	Value	Description
JAVA_HOME	<path_to_jdk>, for example: `/usr/jdk64/jdk1.8.0_40`	Provide the path to the JDK install folder. For Hadoop, you can check `/etc/hadoop/conf/hadoop-env.sh` for the value of JAVA_HOME. As noted previously, Solr only supports JDK 1.7 and higher.
SOLR_USER	solr	The Linux user used to run Solr.
SOLR_INSTALL_FOLDER	/opt/lucidworks-hdpsearch/solr	The Solr installation directory.
SOLR_RANGER_HOME	/opt/lucidworks-hdpsearch/solr/ranger_audit_server	The location where the Ranger-related configuration and schema files will be copied.
SOLR_RANGER_PORT	6083	The Solr port for Ranger.

Property Name	Value	Description
SOLR_DEPLOYMENT	solrcloud	The deployment type.
SOLR_ZK	<ZooKeeper_host>:2181/ranger_audits	The Solr ZooKeeper host and port. It is recommended to provide a sub-folder to create the Ranger Audit related configurations so you can also use ZooKeeper for other Solr instances. Due to a Solr bug, if you are using a path (sub-folder), you can only specify one ZooKeeper host.
SOLR_SHARDS	1	If you want to distribute your audit logs, you can use multiple shards. Make sure the number of shards is equal or less than the number of Solr nodes you will be running.
SOLR_REPLICATION	1	It is highly recommend that you set up at least two nodes and replicate the indexes. This gives redundancy to index data, and also provides load balancing of Solr queries.
SOLR_LOG_FOLDER	/var/log/solr/ranger_audits	The folder for the Solr log files.
SOLR_MAX_MEM	2g	The memory allocation for Solr.

5. Use the following command to run the set up script.

```
./setup.sh
```

6. Run the following command **only once** from any node. This command adds the Ranger Audit configuration (including `schema.xml`) to ZooKeeper.

```
/opt/lucidworks-hdpsearch/solr/ranger_audit_server/scripts/
add_ranger_audits_conf_to_zk.sh
```

7. Log in as the `solr` or `root` user and run the following command to start Solr on each node.

```
/opt/lucidworks-hdpsearch/solr/ranger_audit_server/scripts/start_solr.sh
```

   When Solr starts, a confirmation message appears.

```
Started Solr server on port 6083 (pid=). Happy searching!
```

8. Run the following command **only once** from any node. This command creates the Ranger Audit collection.

```
/opt/lucidworks-hdpsearch/solr/ranger_audit_server/scripts/
create_ranger_audits_collection.sh
```

9. You can use a web browser to open the Solr Admin Console at the following address:

```
http:<solr_host>:6083/solr
```

Note

You can use the following command to stop Solr:

```
/opt/lucidworks-hdpsearch/solr/ranger_audit_server/scripts/
stop_solr.sh
```

5. Ranger Plug ins Overview

Ranger plugins can be enabled for several HDP services. This section describes how to enable each of these plugins. For performance reasons, it is recommended that you store audits in Solr and HDFS, and not in a database.

If you are using a Kerberos-enabled cluster, there are a number of additional steps you must follow to ensure that you can use the Ranger plugins on a Kerberos cluster.

The following Ranger plugins are available:

- HDFS [41]
- Hive [46]
- HBase [52]
- Kafka [56]
- Knox [60]
- YARN [66]
- Storm [70]

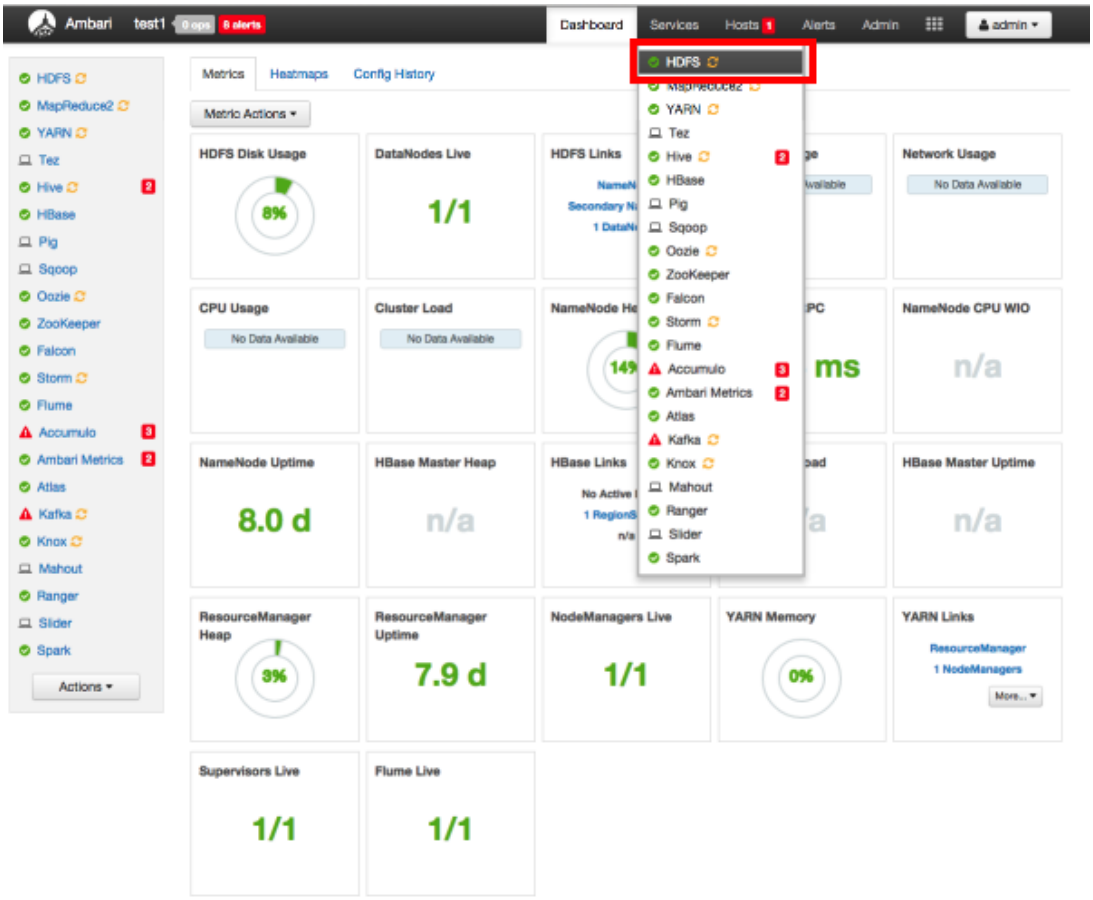
You can save Ranger audits to HDFS or Solr:

- Save Audits to HDFS [74]
- Save Audits to Solr [75]

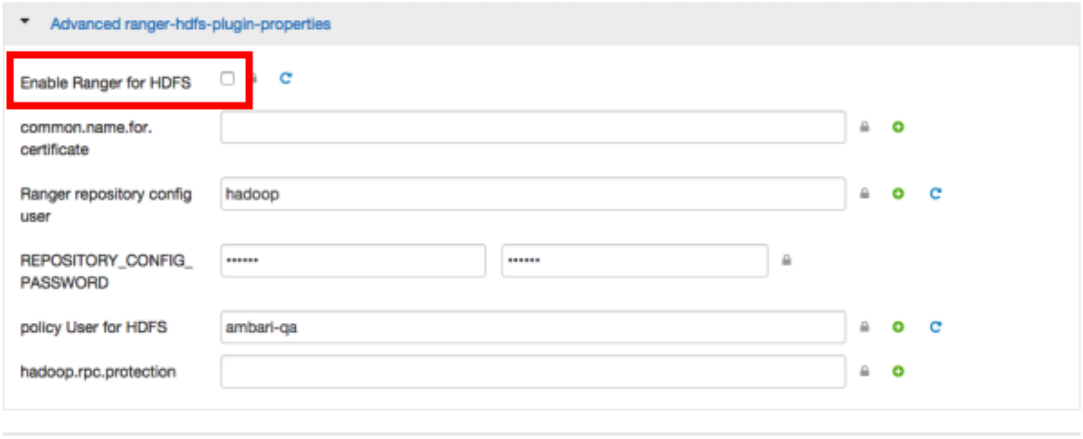
5.1. HDFS

Use the following steps to enable the Ranger HDFS plugin.

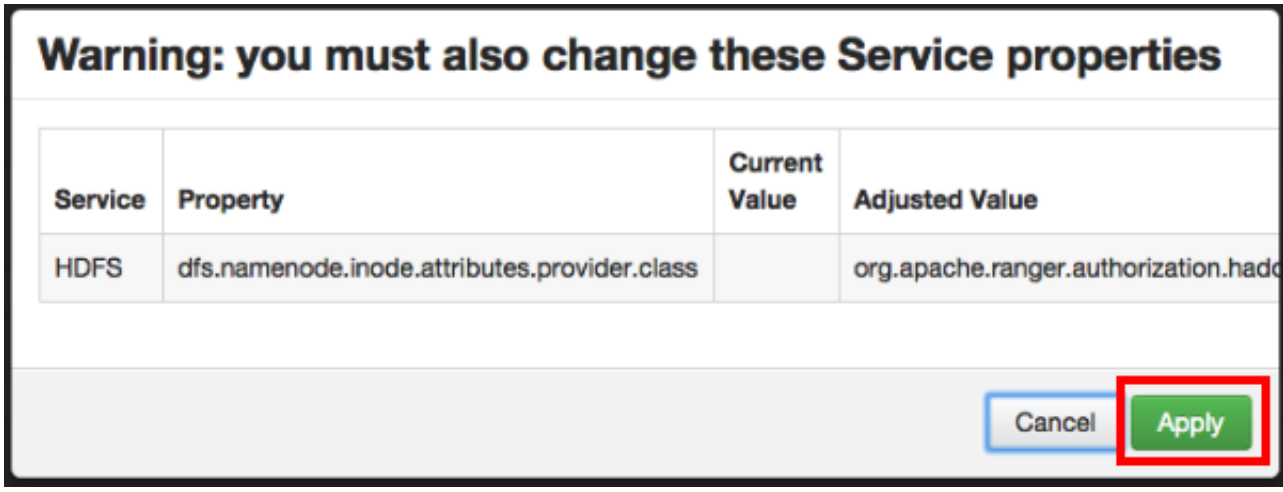
1. Select **HDFS** from the Services tab in the top menu.



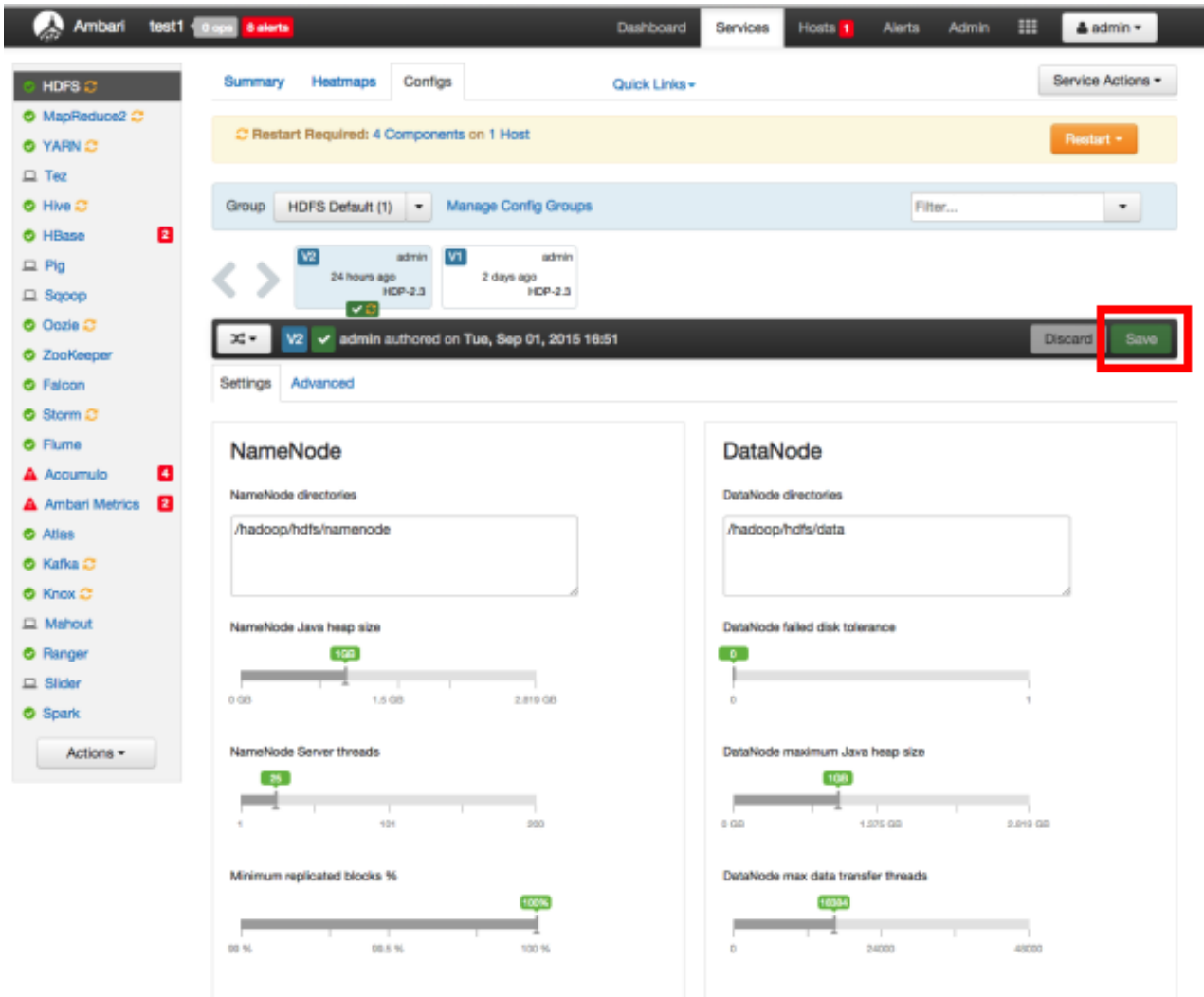
2. Click the **Configs** tab, then click the **Advanced** tab. Scroll down and click to open **Advanced ranger-hdfs-plugin-properties**.



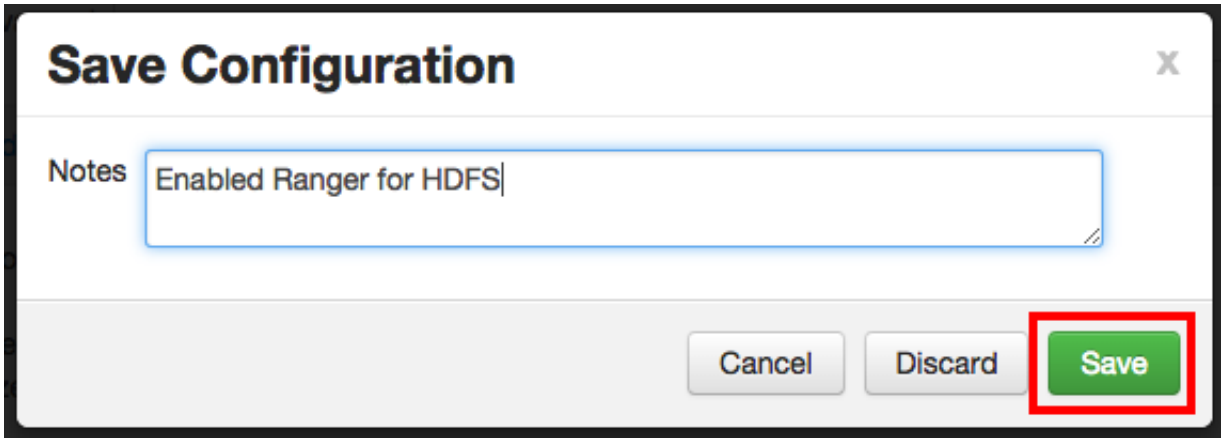
3. Select the **Enable Ranger for HDFS** check box. A Warning pop-up appears. Click **Apply** to save the property updates.



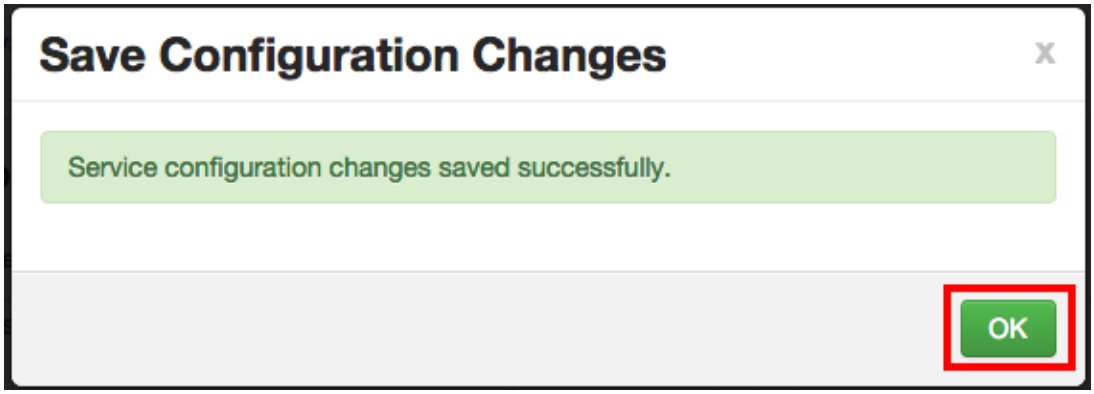
4. To save the configuration, click the green **Save** button on the black menu bar at the top of the page.



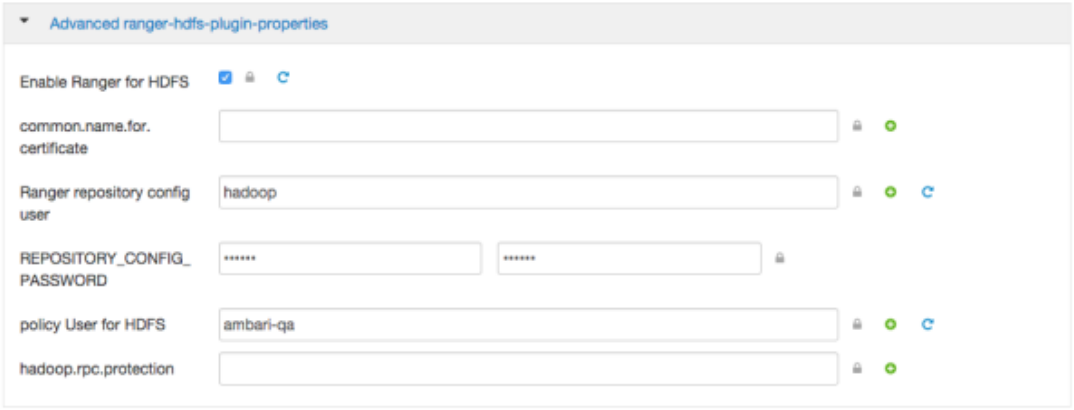
5. A Save Configuration pop-up appears. Type in a note describing the changes you just made, then click **Save**.



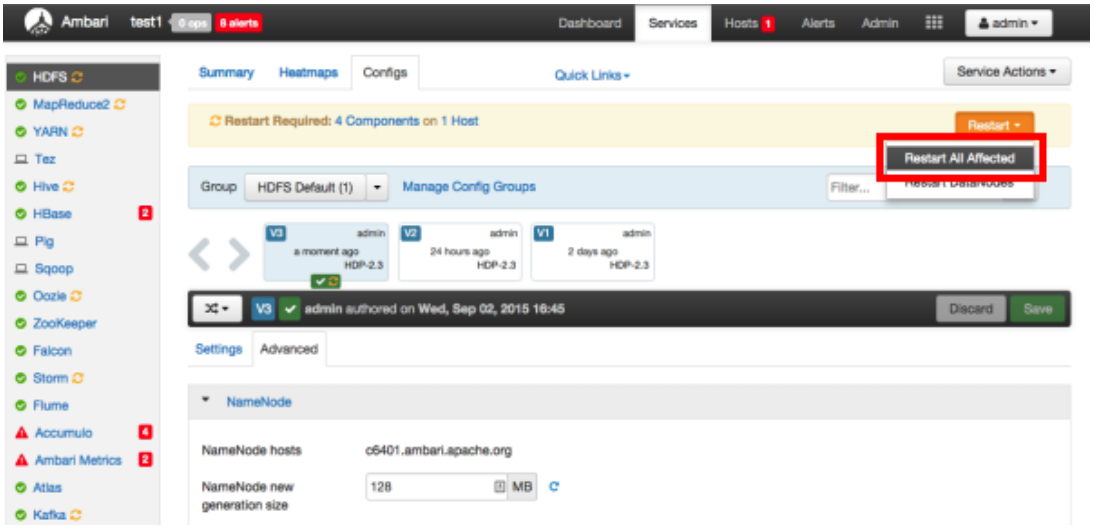
6. Click **OK** on the Save Configuration Changes pop-up.



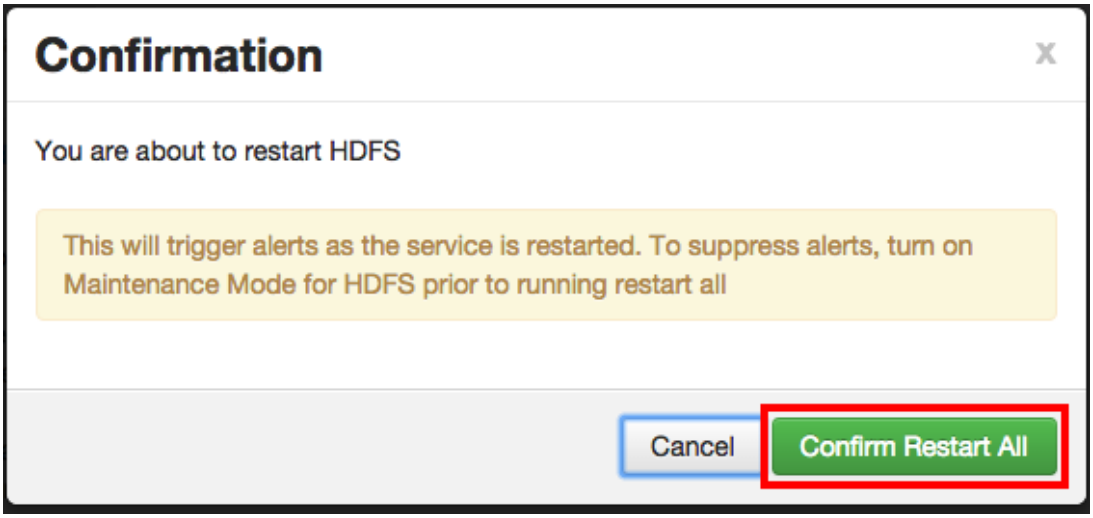
7. The new plugin properties for HDFS will be displayed.



8. A Restart Required message will be displayed at the top of the page. Click **Restart**, then select **Restart All Affected** to restart the HDFS service and load the new configuration.



9. Click **Confirm Restart All** on the confirmation pop-up to confirm the HDFS restart.



10. After HDFS has restarted, the Ranger plugin for HDFS is enabled.

Note

In order to access HDFS folders in previous versions of HDP, access permissions also had to be granted in Ranger to the applicable parent folders. As of HDP-2.3, it is no longer required to grant access permissions to the parent folder.

For example, for the folder path `../customer/data/marketing`:

- In previous versions, to grant access to the `/customer/data/marketing` folder, you were required to grant Execute permission in Ranger for both the `/customer` and `/customer/data` folders, along with a Read or Write permission for the `/customer/data/marketing` folder.
- As of HDP-2.3, it is no longer necessary to grant Execute permission to the parent folders.

For more details, see RANGER-357.

5.2. Hive

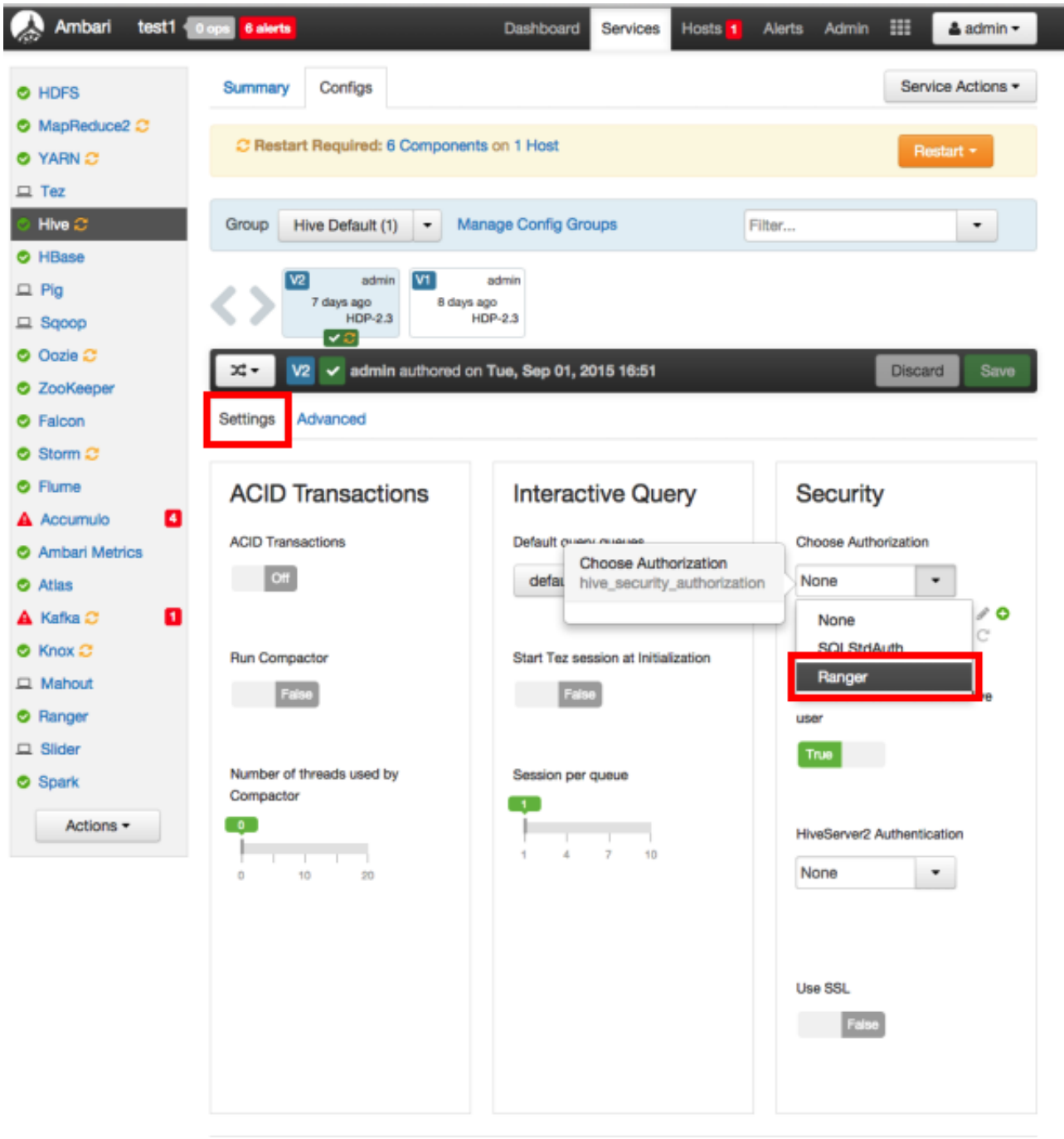
Important

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.

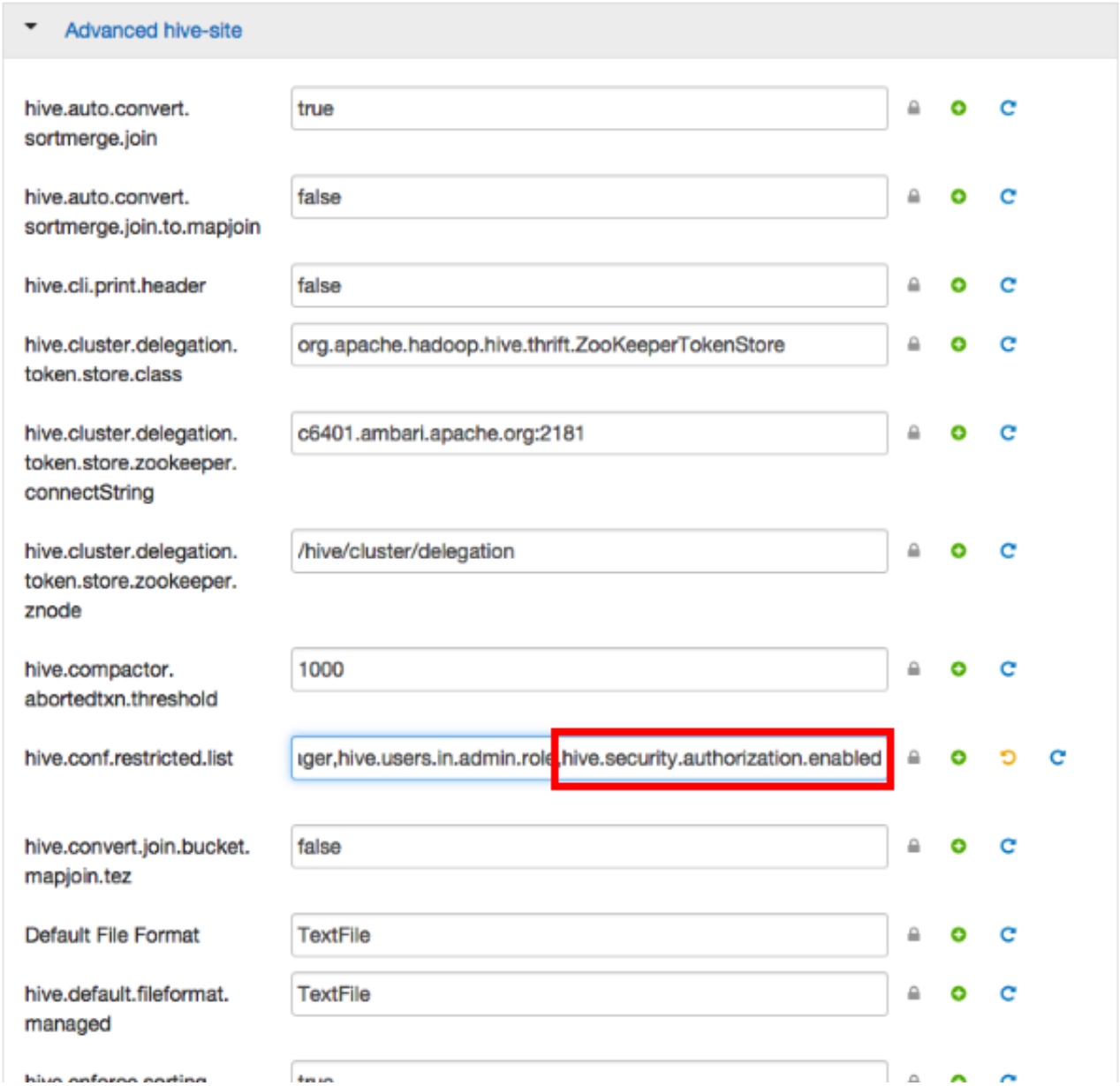
Use the following steps to enable the Ranger Hive plugin.

1. Select **Hive** from the Services tab in the top menu.

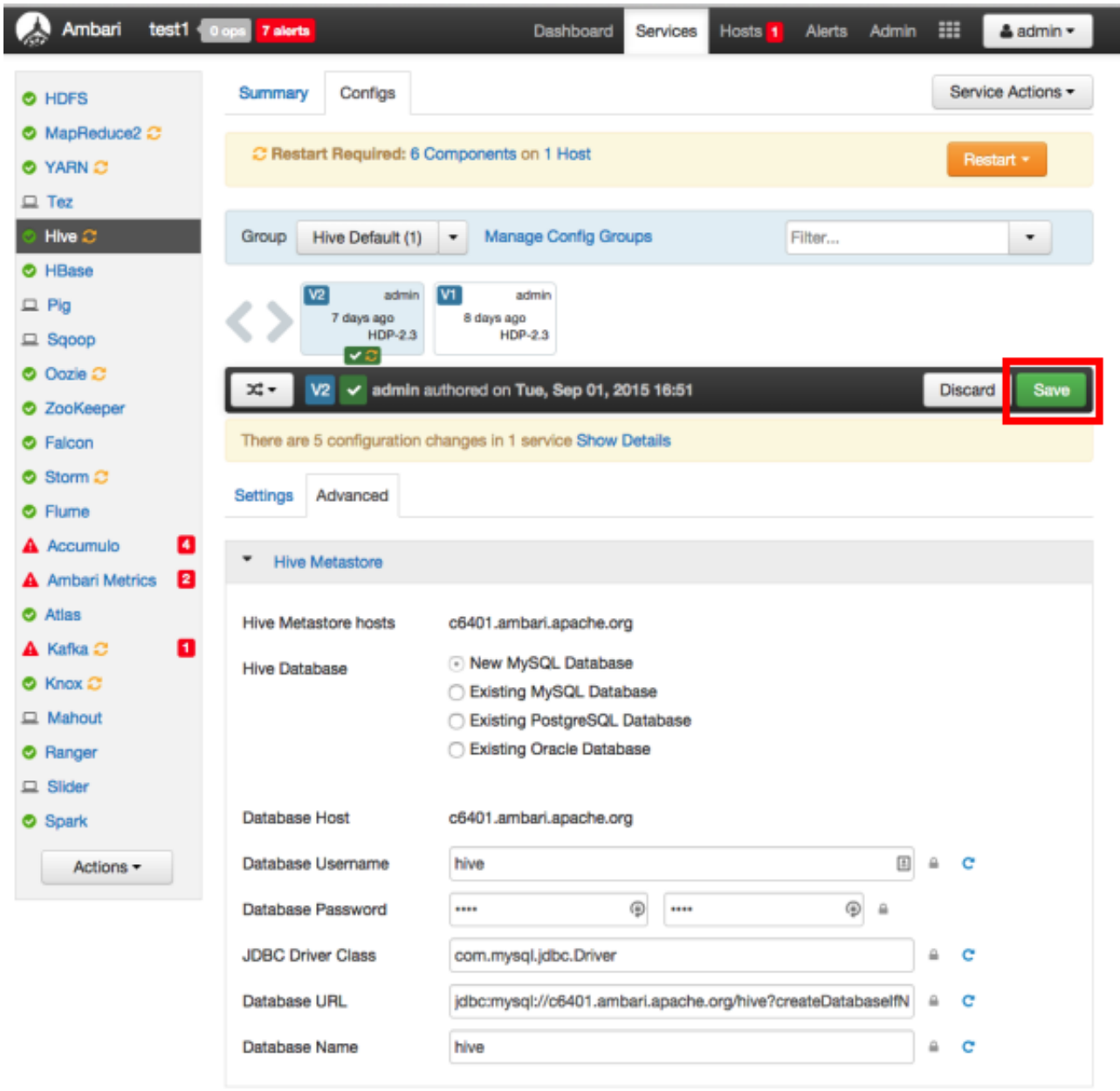
2. Click the **Configs** tab, then click the **Settings** tab. Use the drop-down in the Security box to select **Ranger**.



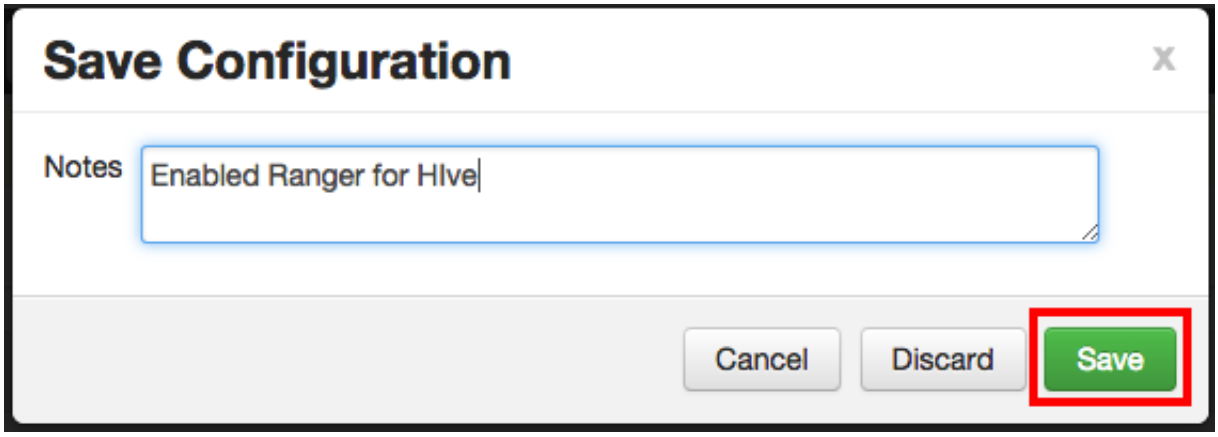
3. On the **Advanced** tab, click to open **Advanced hive-site**. Click inside the **hive.conf.restricted.list** box. Use the right-arrow button to scroll to the end of the comma-separated list of properties. Add the `hive.security.authorization.enabled` property to the end of the list.



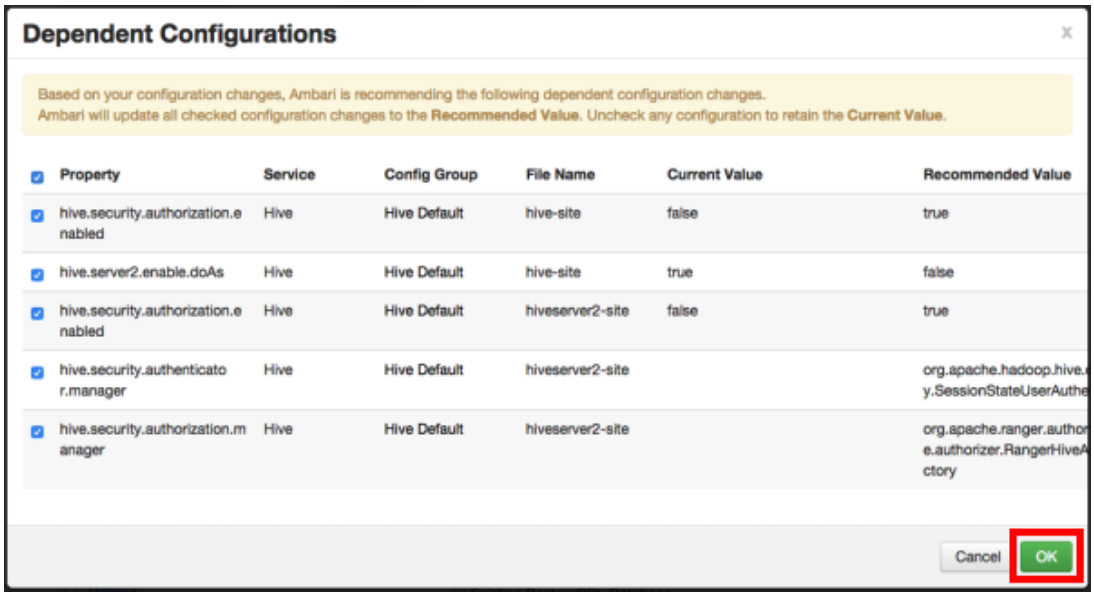
4. To save the configuration, click the green **Save** button on the black menu bar at the top of the page.



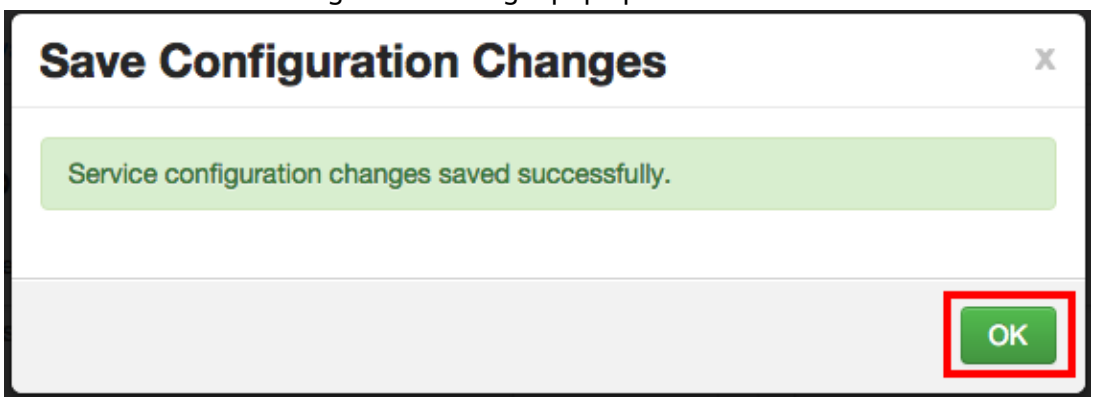
5. A Save Configuration pop-up appears. Type in a note describing the changes you just made, then click **Save**.



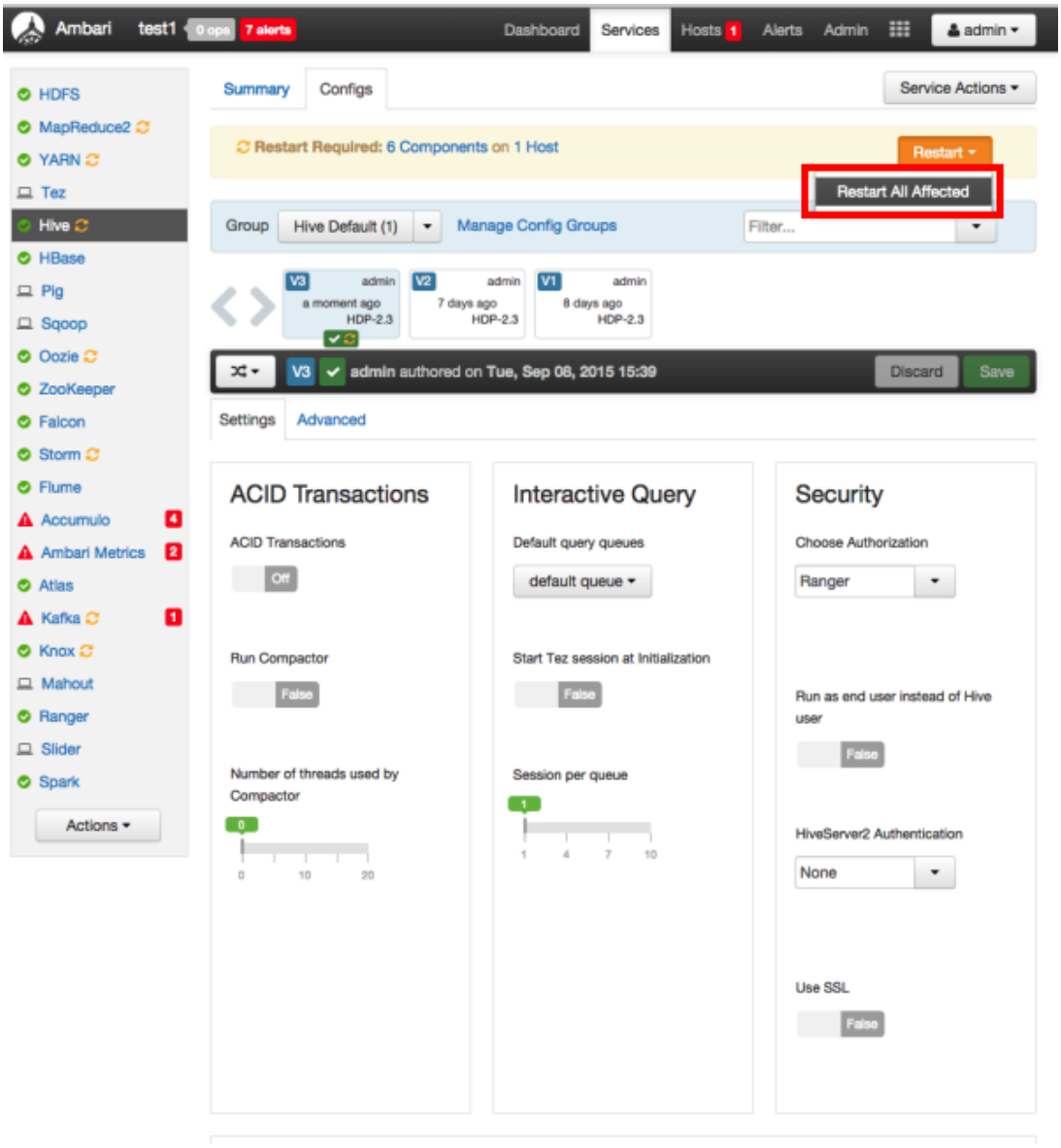
6. The configuration changes will be listed on the Dependent Configurations pop-up. Click **OK** to confirm the changes.



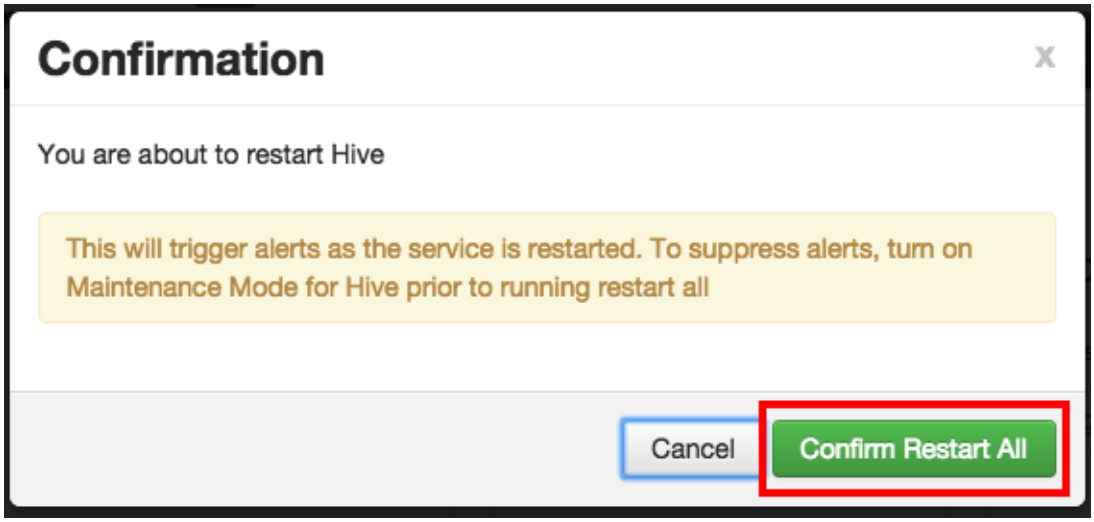
7. Click **OK** on the Save Configuration Changes pop-up.



8. A Restart Required message will be displayed at the top of the page. Click **Restart**, then select **Restart All Affected** to restart the Hive service and load the new configuration.



9. Click **Confirm Restart All** on the confirmation pop-up to confirm the Hive restart.



10. After Hive has been restarted, the Ranger plugin for Hive will be enabled.

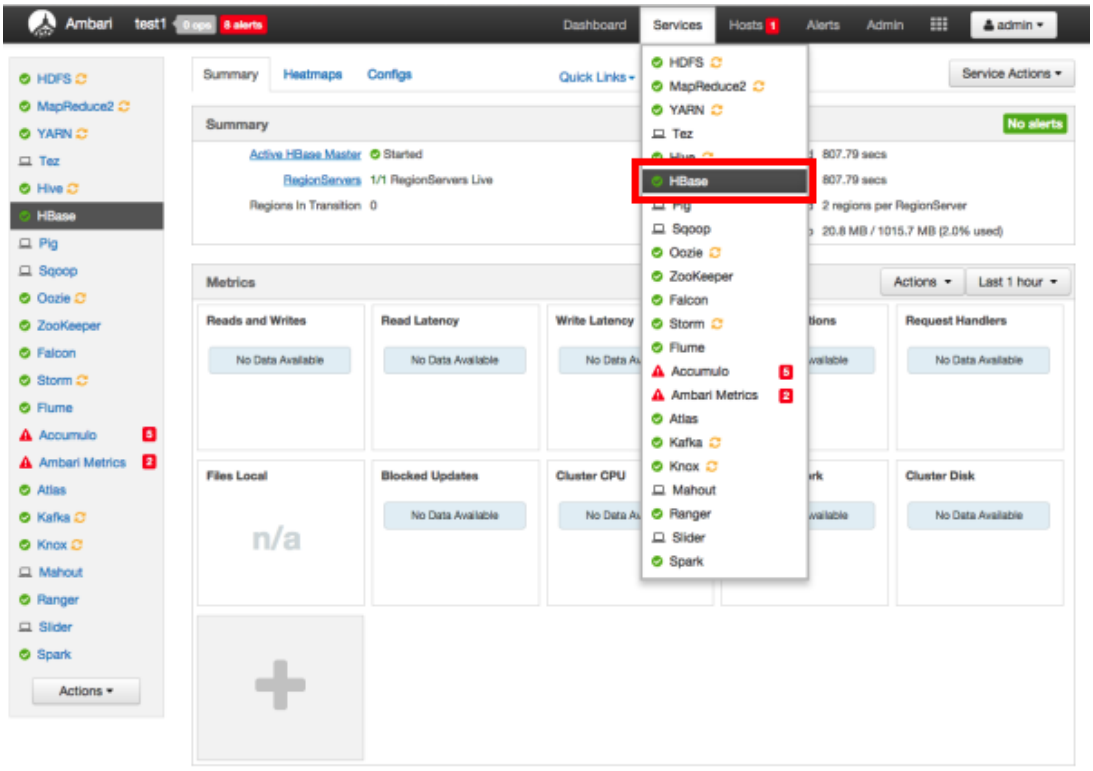
5.3. HBase

Note

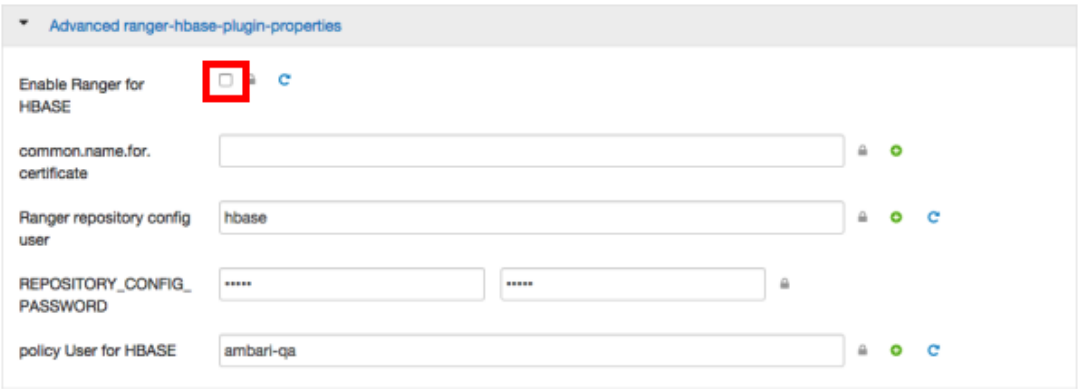
When HBase is configured with Ranger, and specifically XASecure Authorizer, you may only grant and revoke privileges.

Use the following steps to enable the Ranger HBase plugin.

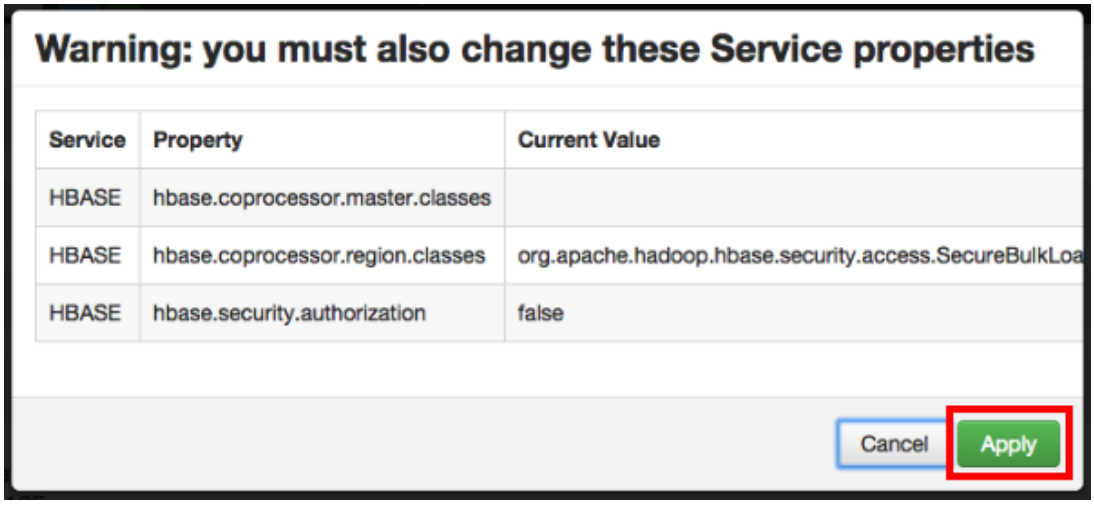
1. Select **HBase** from the Services tab in the top menu.



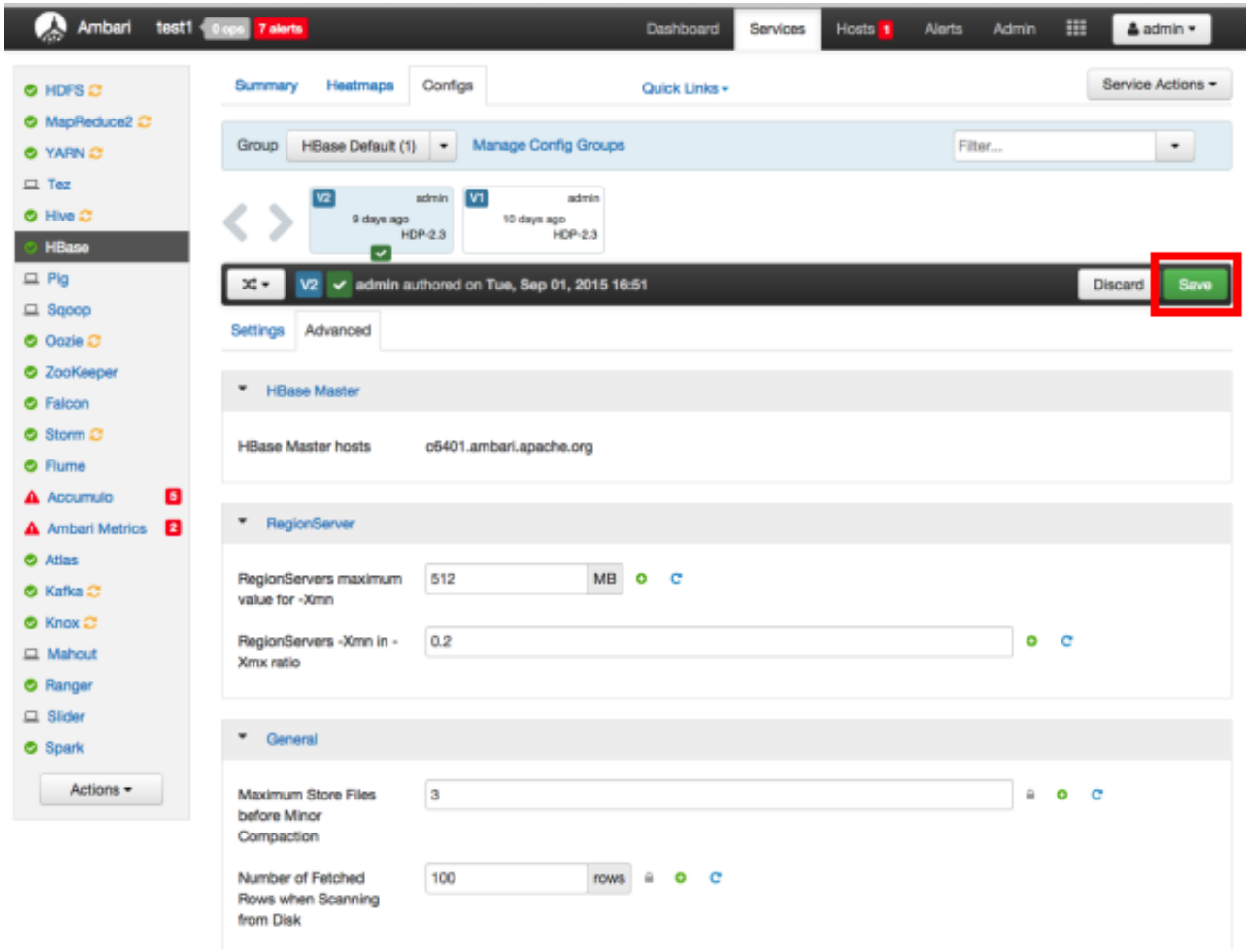
2. Click the **Configs** tab, then click the **Advanced** tab. Scroll down and click to open **Advanced ranger-hbase-plugin-properties**.



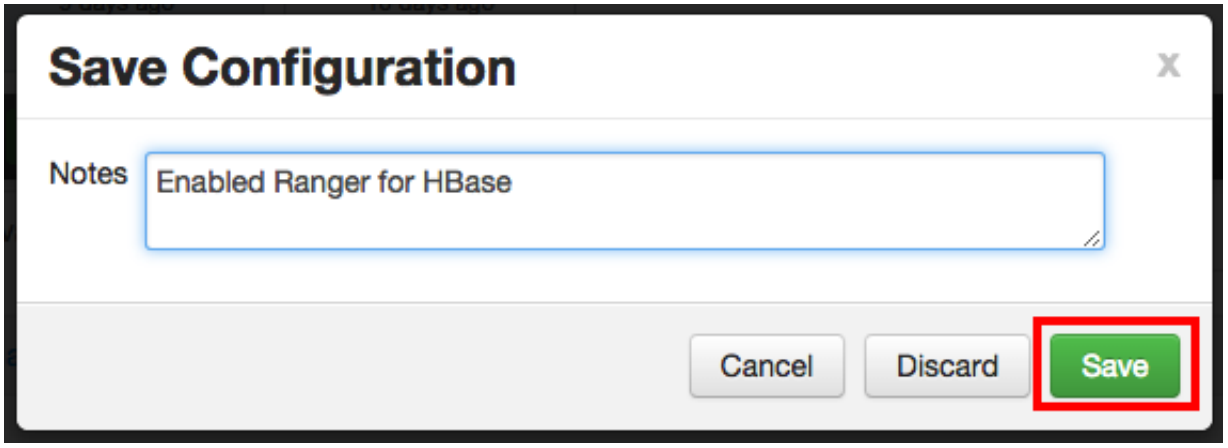
3. Select the **Enable Ranger for HBASE** check box. A Warning pop-up appears. Click **Apply** to save the property updates.



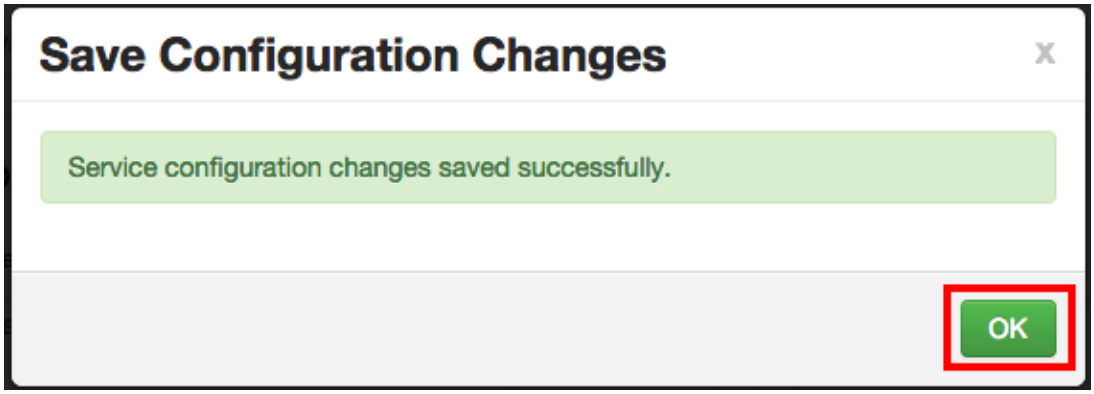
4. To save the configuration, click the green **Save** button on the black menu bar at the top of the page.



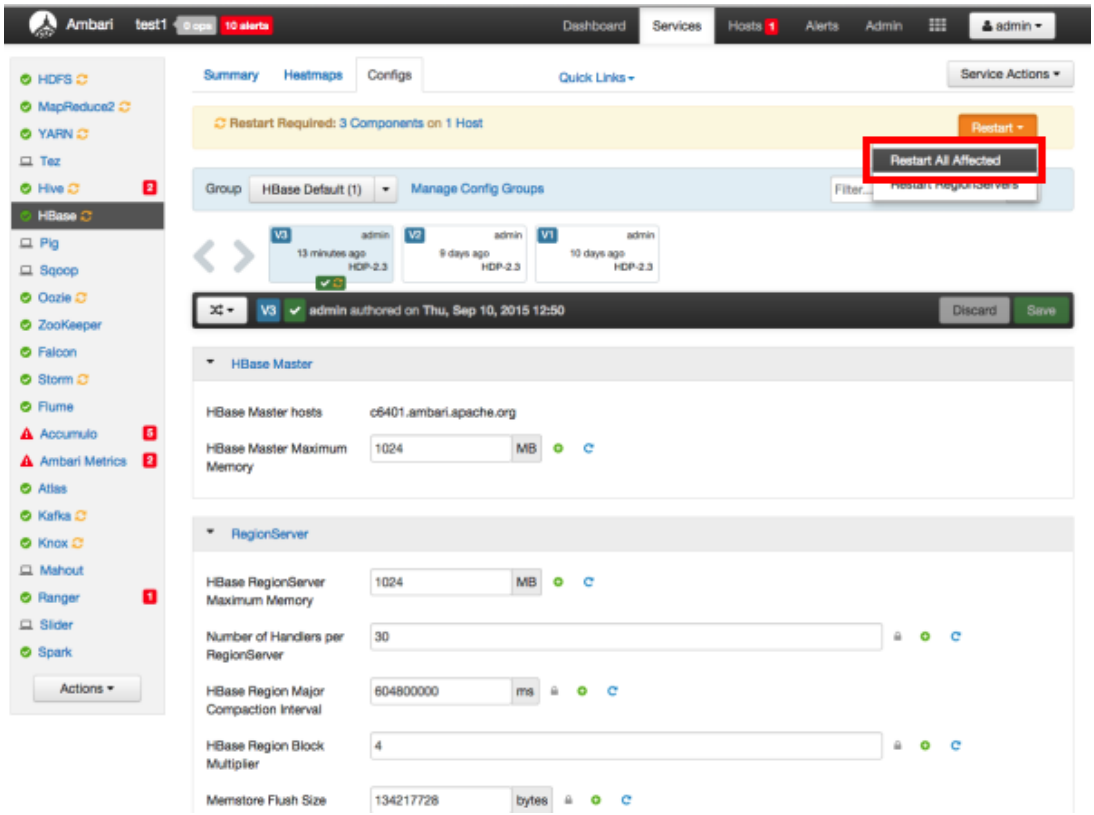
5. A Save Configuration pop-up appears. Type in a note describing the changes you just made, then click **Save**.



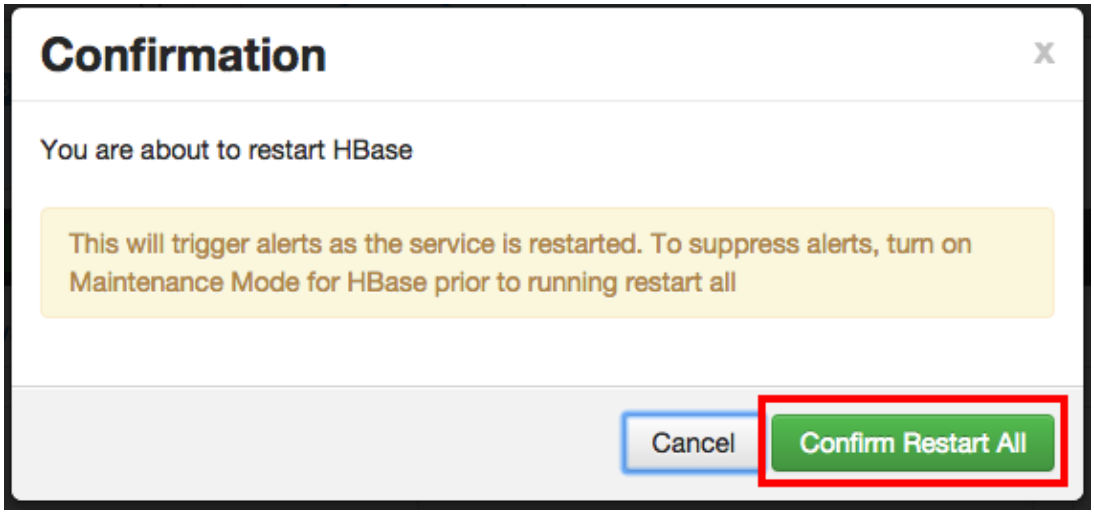
6. Click **OK** on the Save Configuration Changes pop-up.



7. A Restart Required message will be displayed at the top of the page. Click **Restart**, then select **Restart All Affected** to restart the HBase service and load the new configuration.



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

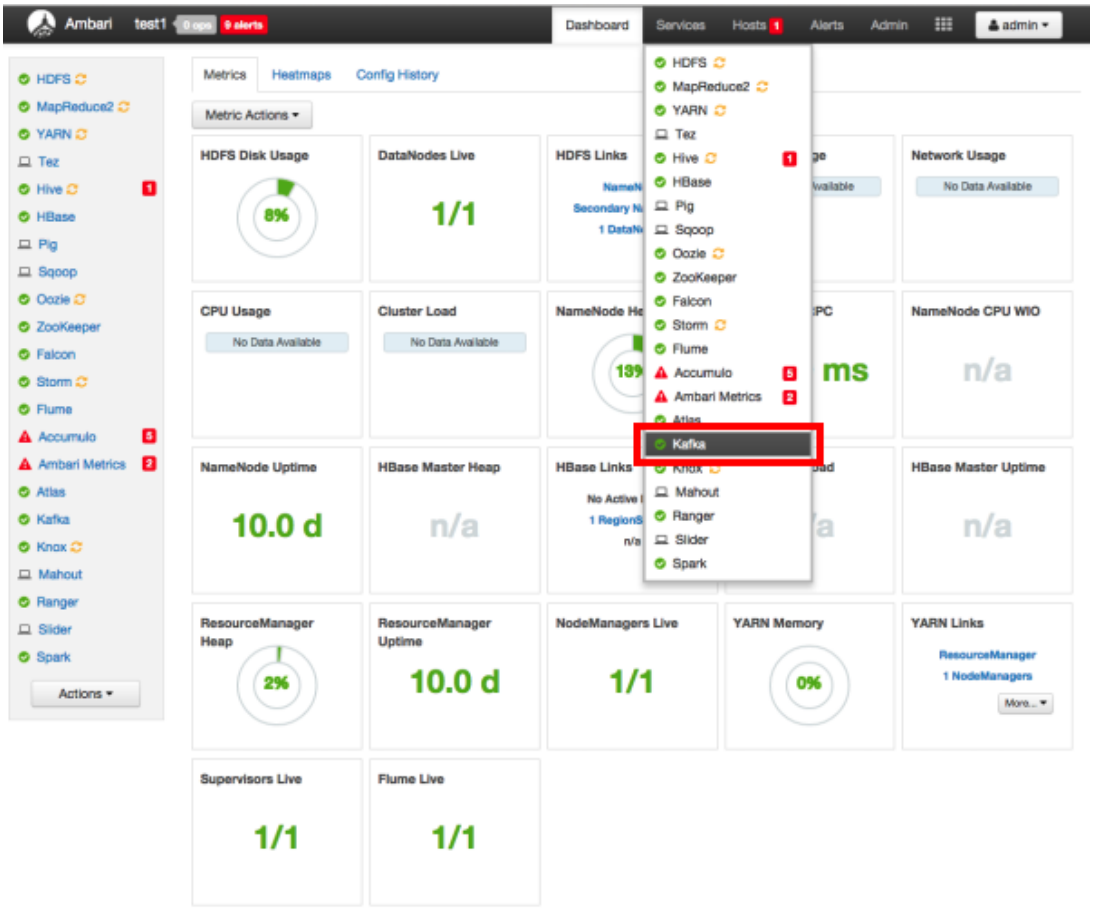


9. After HBase has been restarted, the Ranger plugin for HBase will be enabled.

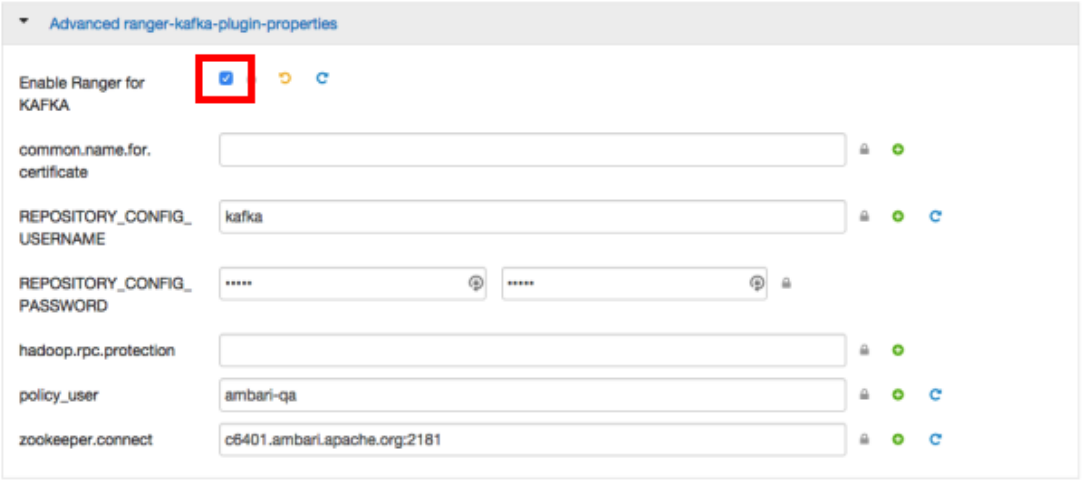
5.4. Kafka

Use the following steps to enable the Ranger Kafka plugin.

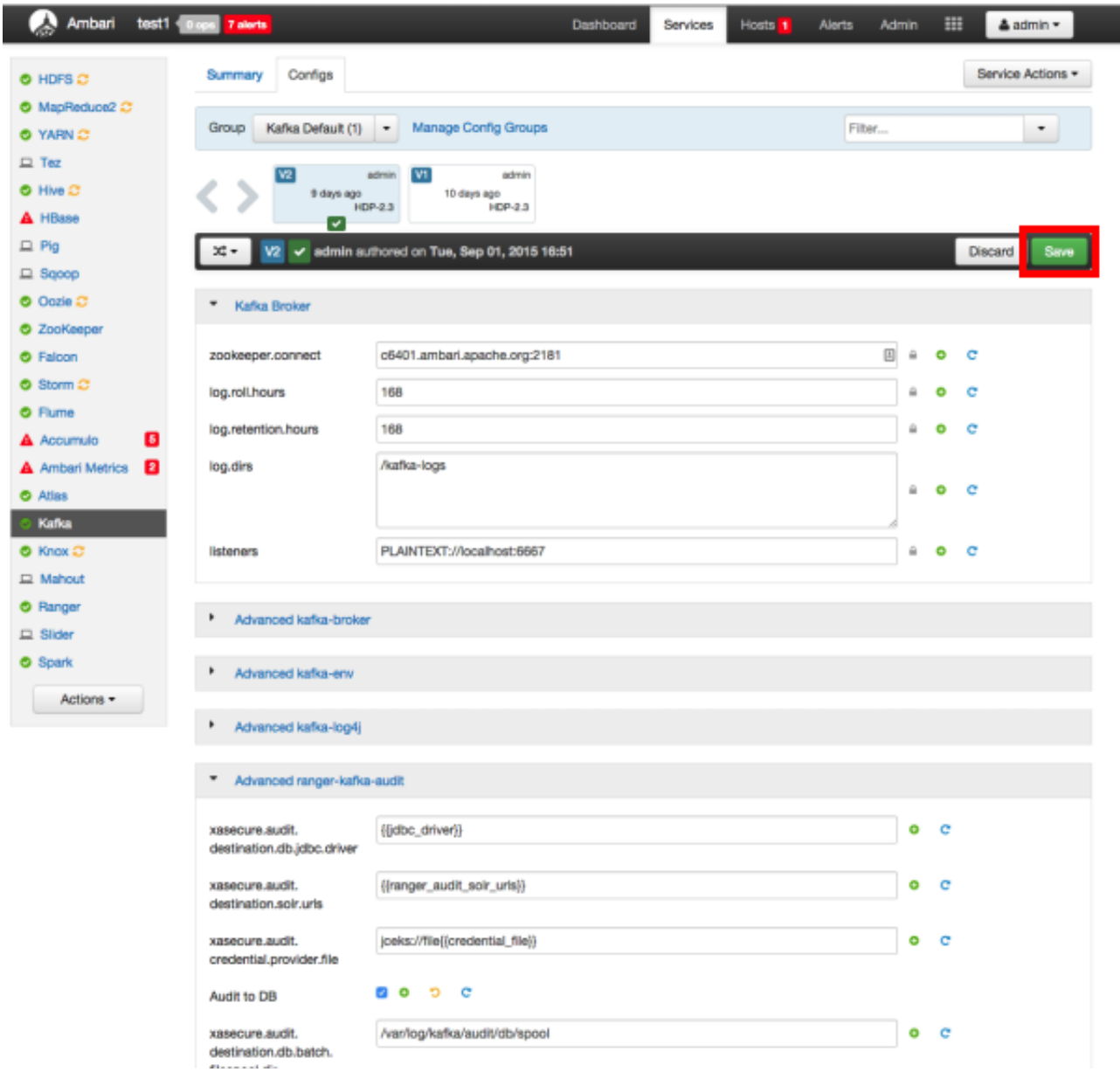
1. Select **Kafka** from the Services tab in the top menu.



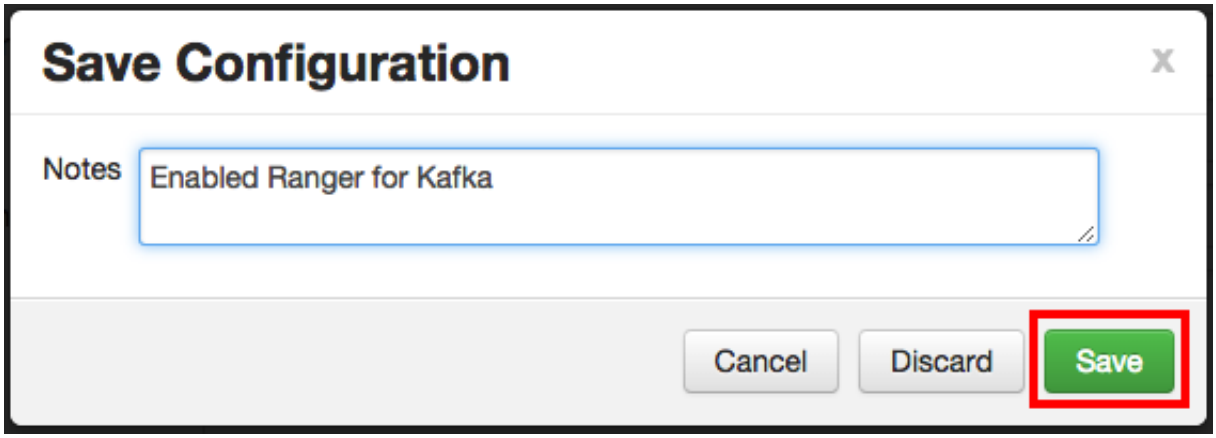
2. Click the **Configs** tab, then scroll down and click to open **Advanced ranger-kafka-plugin-properties**. Select the **Enable Ranger for KAFKA** check box.



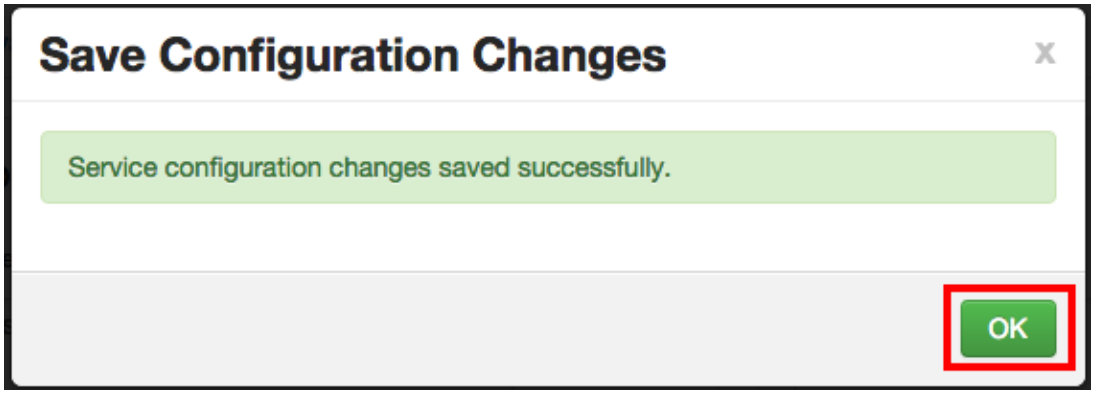
3. To save the configuration, click the green **Save** button on the black menu bar at the top of the page.



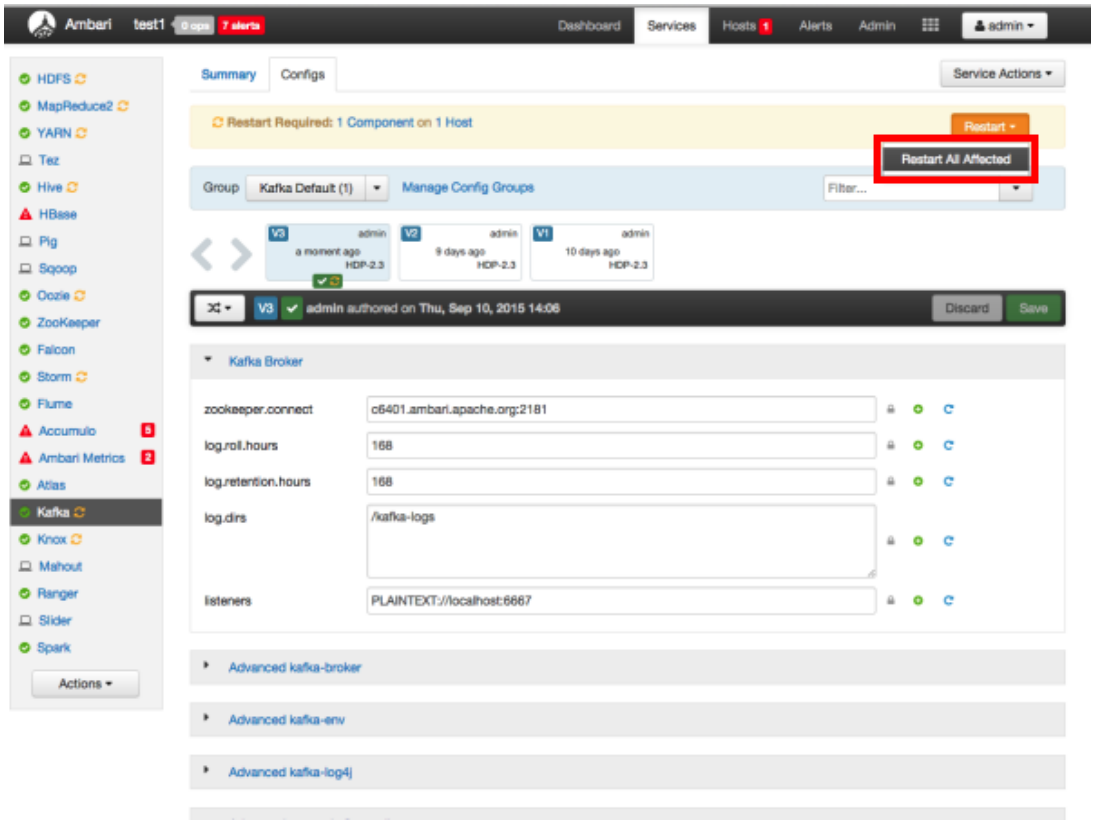
4. A Save Configuration pop-up appears. Type in a note describing the changes you just made, then click **Save**.



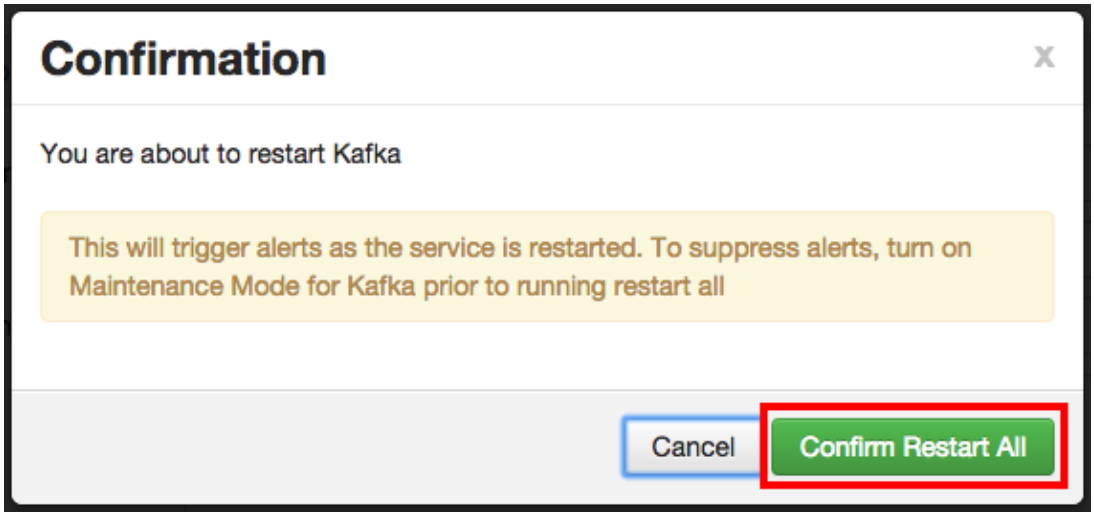
5. Click **OK** on the Save Configuration Changes pop-up.



6. A Restart Required message will be displayed at the top of the page. Click **Restart**, then select **Restart All Affected** to restart the Kafka service and load the new configuration.



7. Click **Confirm Restart All** on the confirmation pop-up to confirm the Kafka restart.

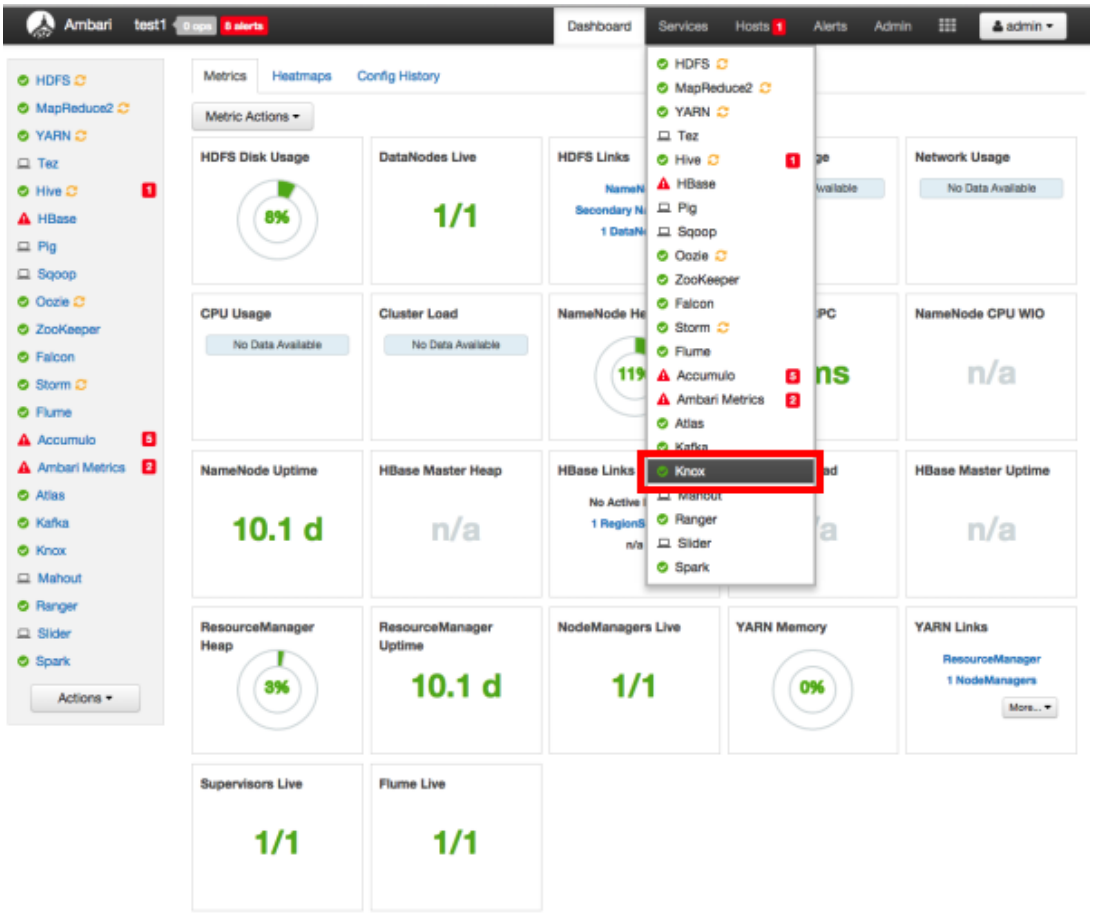


8. After Kafka has been restarted, the Ranger plugin for Kafka will be enabled.

5.5. Knox

Use the following steps to enable the Ranger Knox plugin.

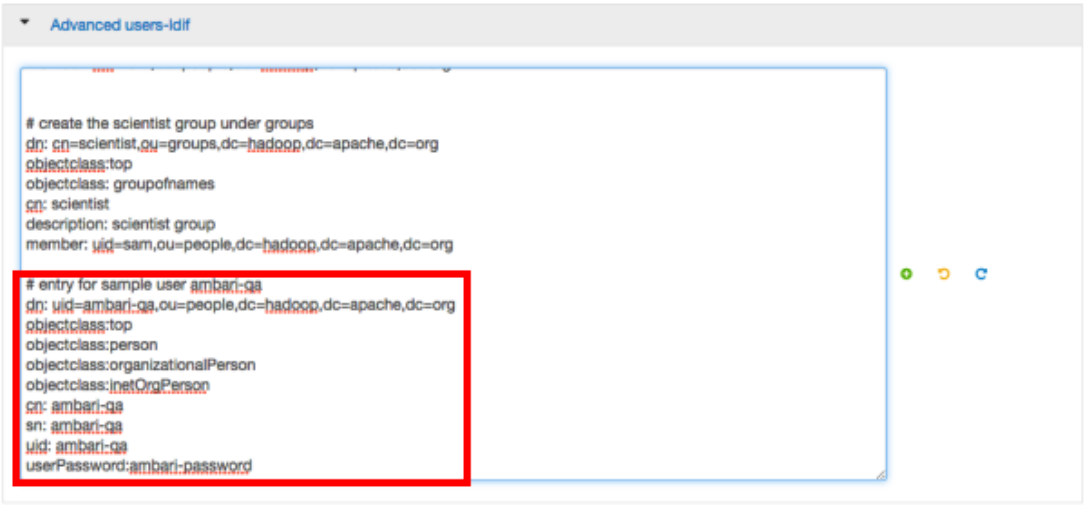
1. Select **Knox** from the Services tab in the top menu.



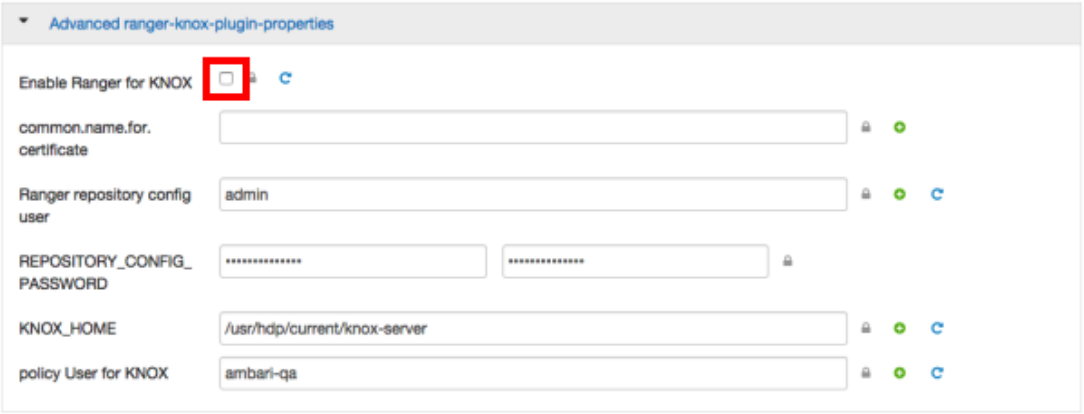
2. Click the **Configs** tab, then scroll down and click to open the **Advanced users-ldif** text box. Scroll down to the bottom of the text box and add the following lines of code:

```
# entry for sample user ambari-qa
dn: uid=ambari-qa,ou=people,dc=hadoop,dc=apache,dc=org
objectclass:top
objectclass:person
objectclass:organizationalPerson
objectclass:inetOrgPerson
cn: ambari-qa
sn: ambari-qa
uid: ambari-qa
userPassword:ambari-password
```

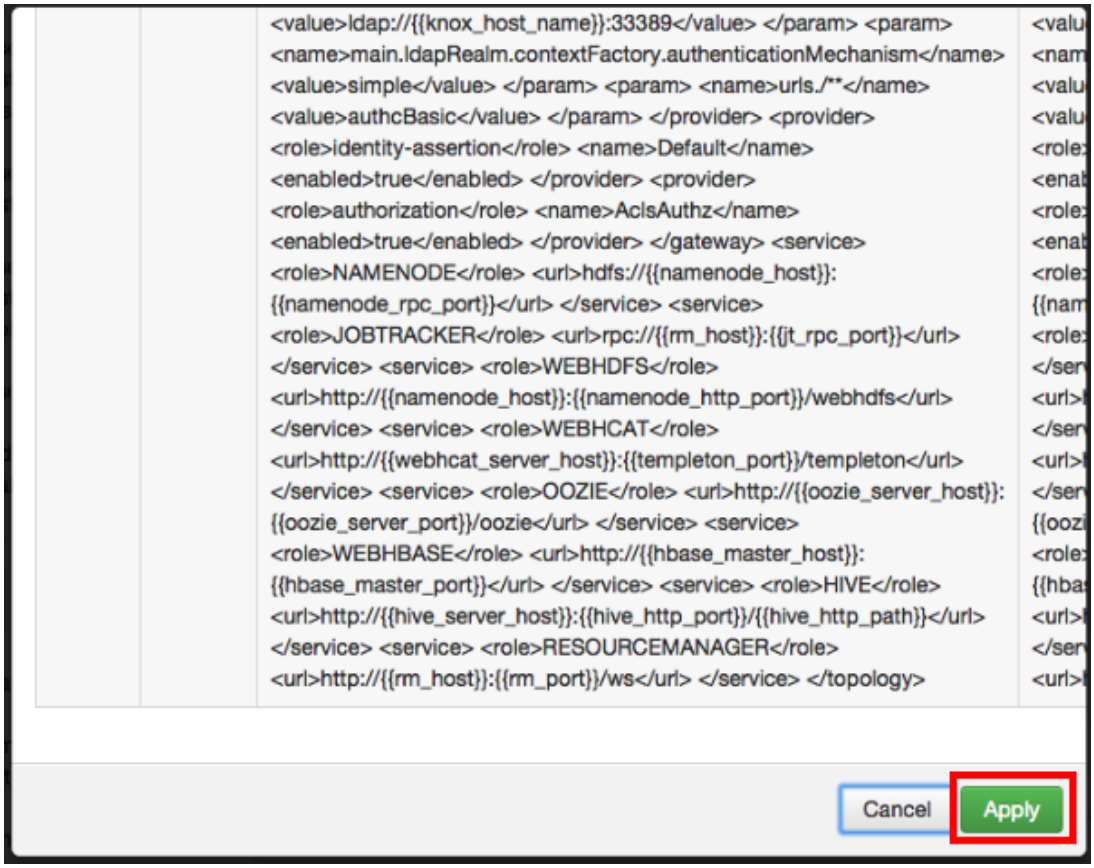
For example:



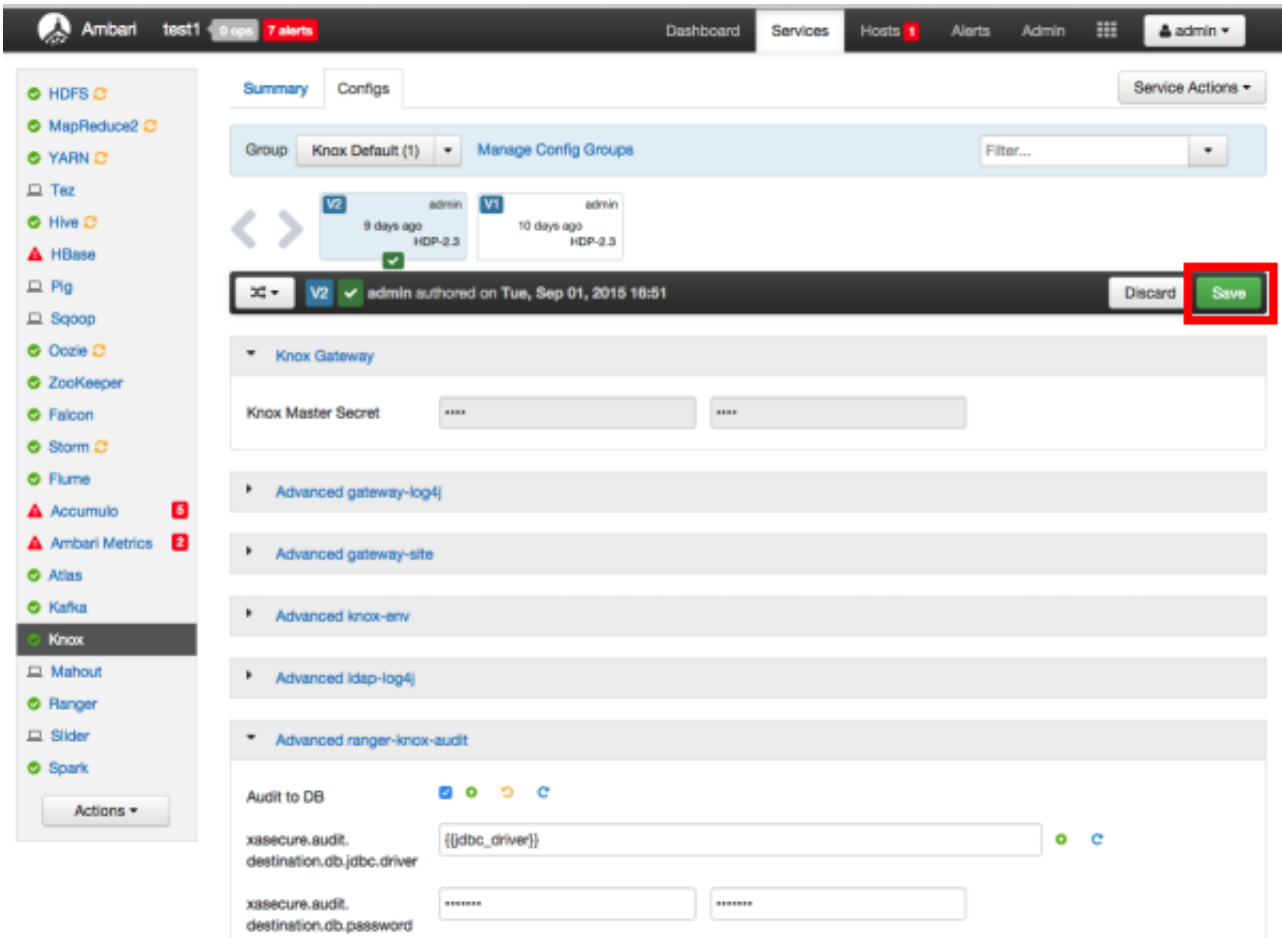
3. Click the **Configs** tab, then click the **Advanced** tab. Scroll down and click to open **Advanced ranger-knox-plugin-properties**.



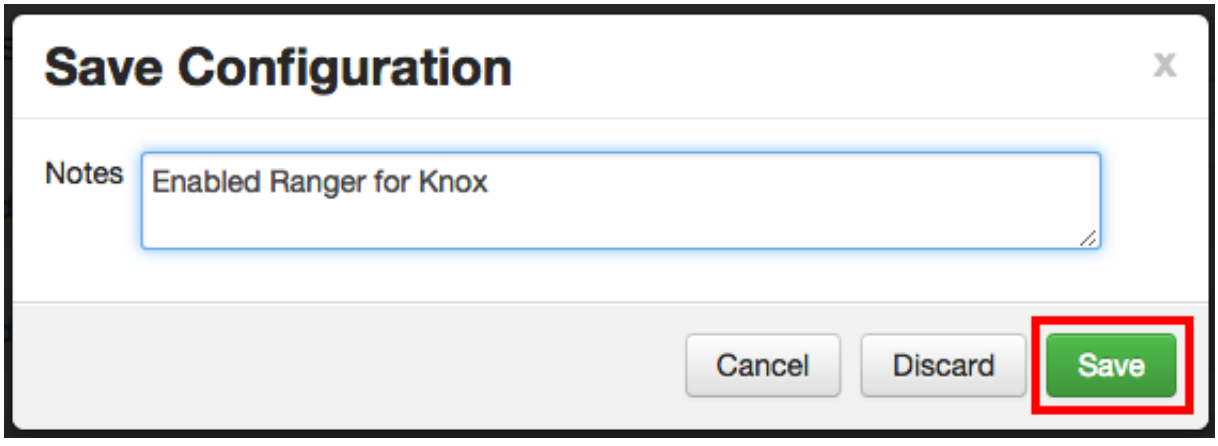
4. Select the **Enable Ranger for KNOX** check box. A Warning pop-up appears. Click **Apply** to save the property updates.



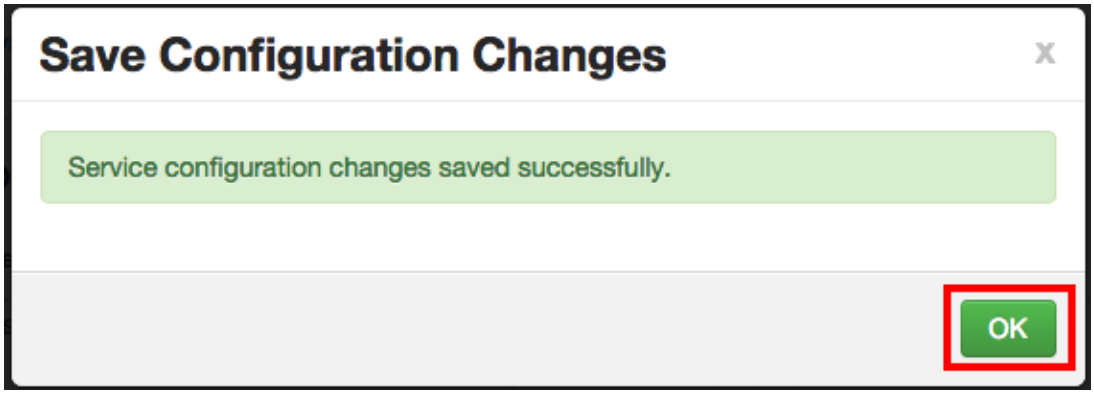
5. To save the configuration, click the green **Save** button on the black menu bar at the top of the page.



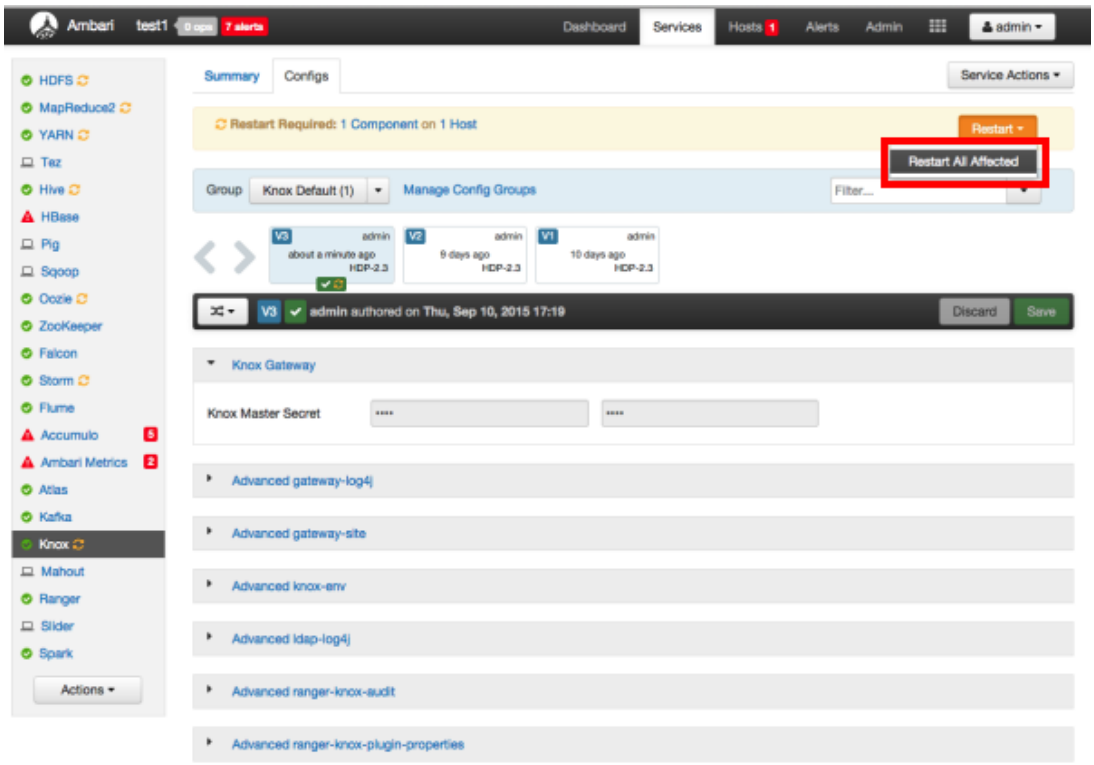
6. A Save Configuration pop-up appears. Type in a note describing the changes you just made, then click **Save**.



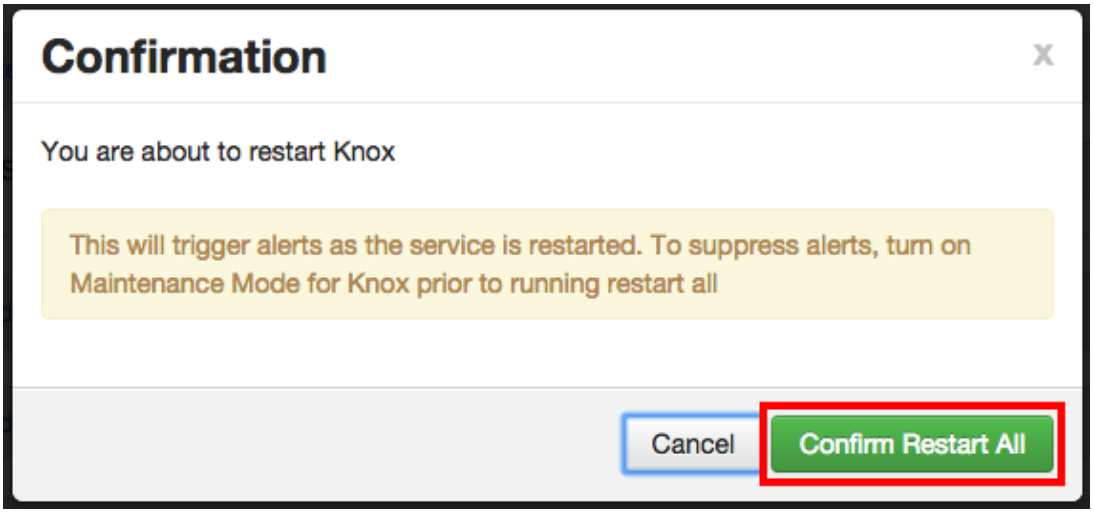
7. Click **OK** on the Save Configuration Changes pop-up.



8. A Restart Required message will be displayed at the top of the page. Click **Restart**, then select **Restart All Affected** to restart the Knox service and load the new configuration.



9. Click **Confirm Restart All** on the confirmation pop-up to confirm the Knox restart.

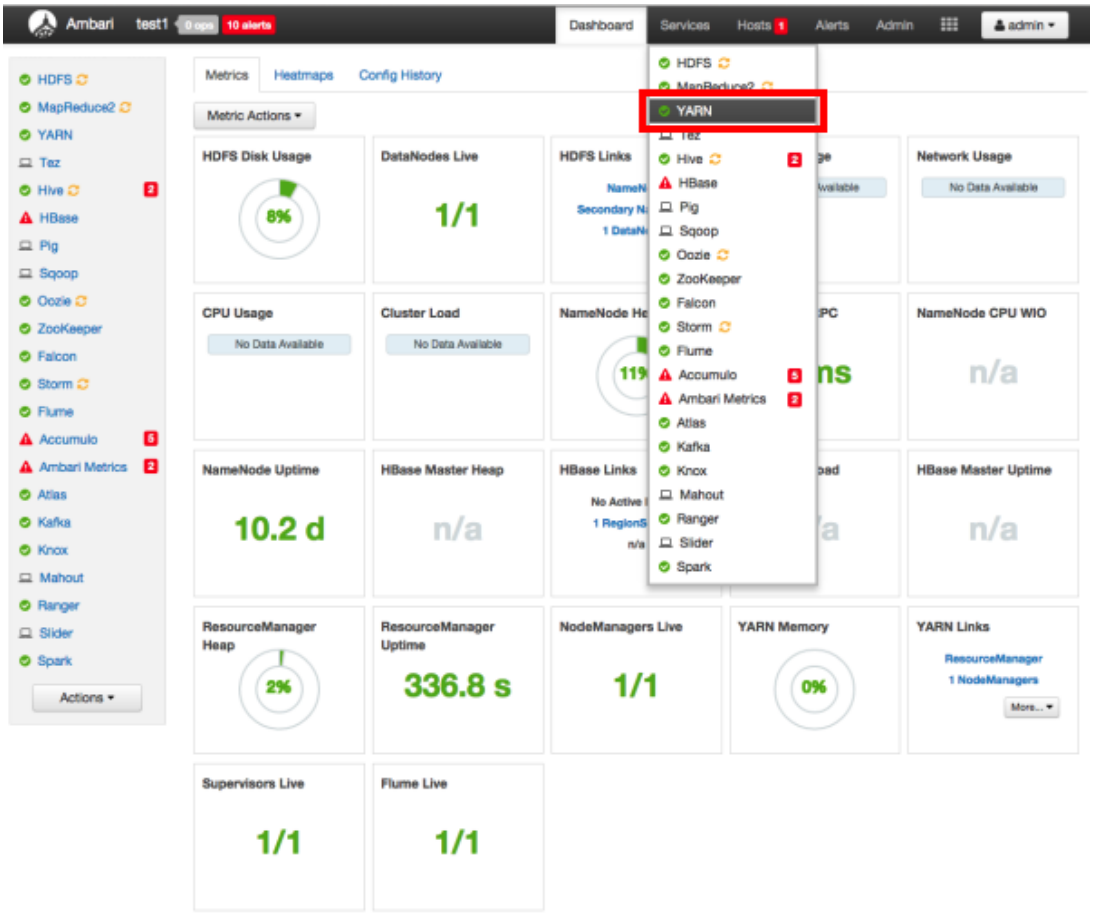


10.After Knox has been restarted, the Ranger plugin for Knox will be enabled.

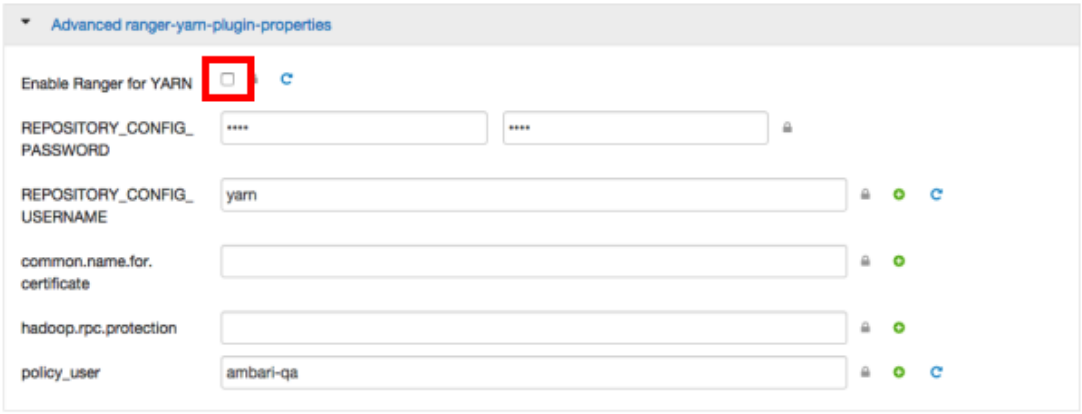
5.6. YARN

Use the following steps to enable the Ranger YARN plugin.

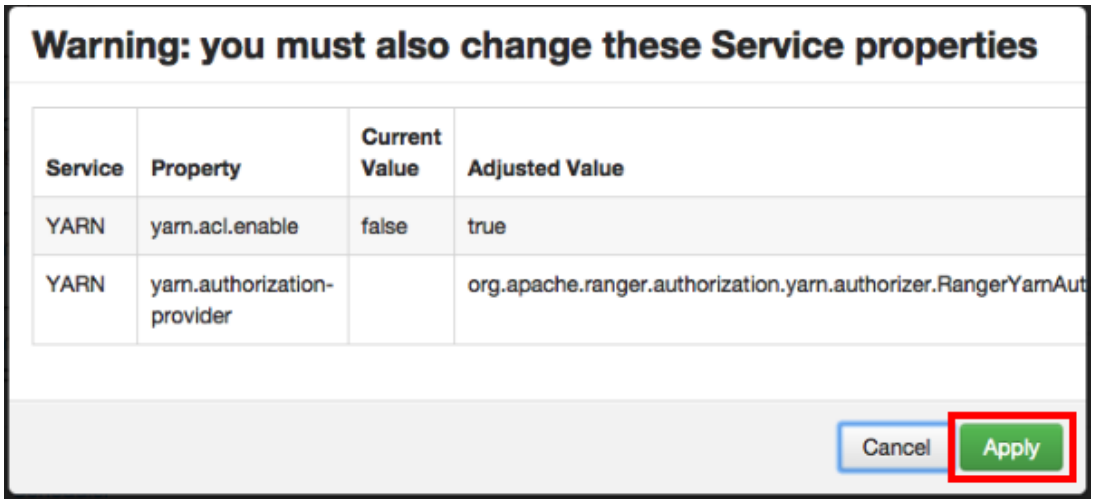
1. Select **YARN** from the Services tab in the top menu.



2. Click the **Configs** tab, then click the **Advanced** tab. Scroll down and click to open **Advanced ranger-yarn-plugin-properties**.



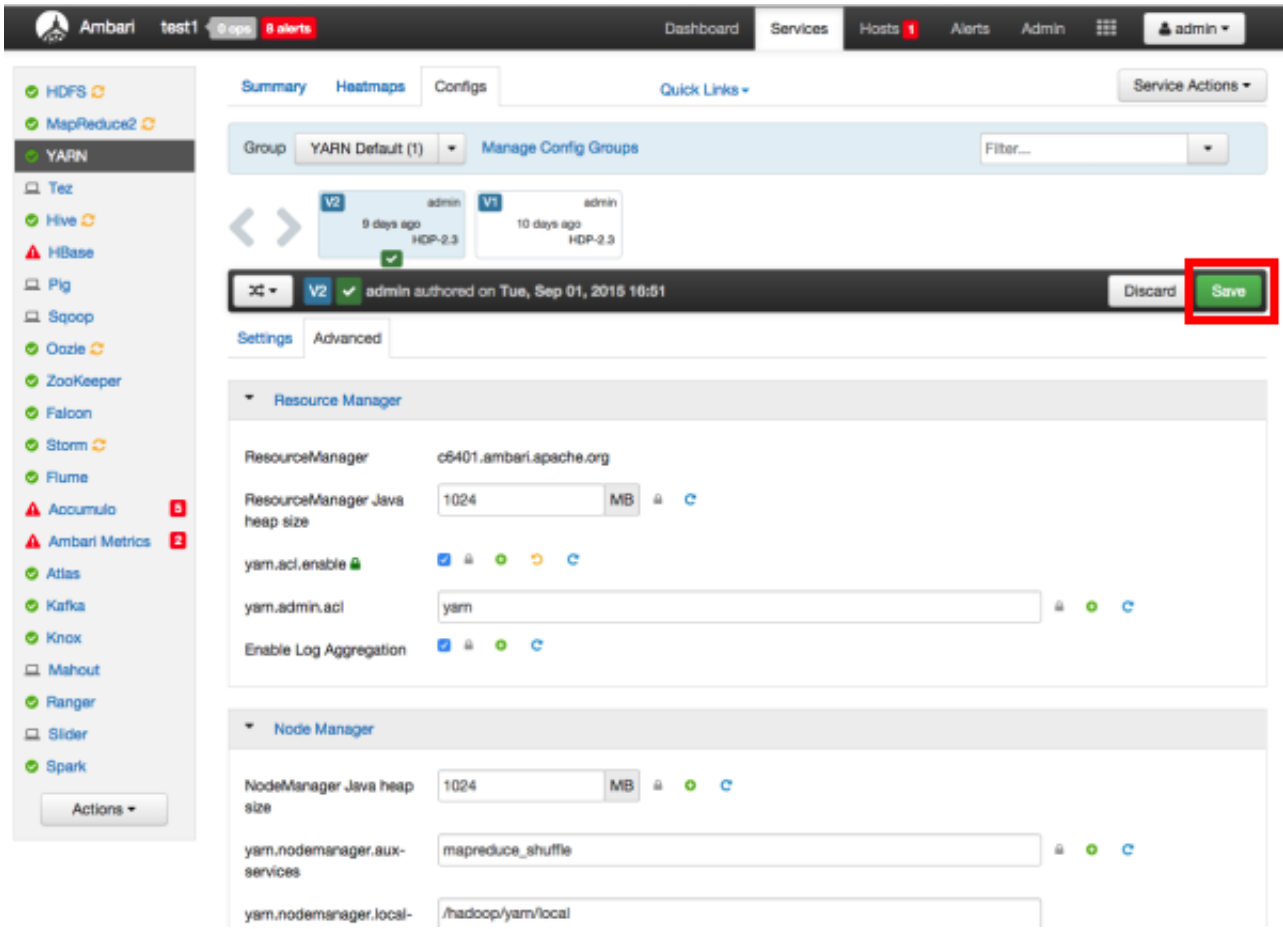
3. Select the **Enable Ranger for YARN** check box. A Warning pop-up appears. Click **Apply** to save the property updates.



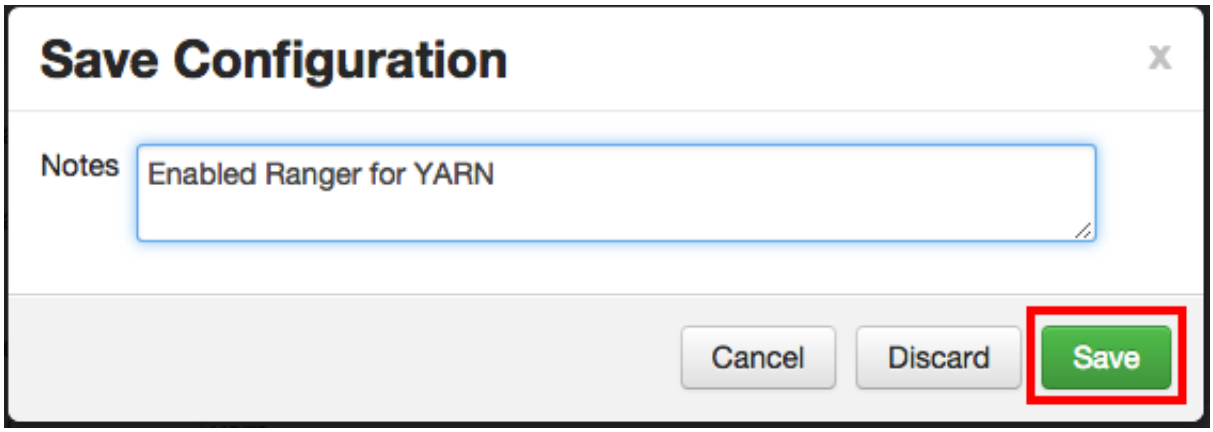
Note

Enabling Ranger for YARN sets the yarn.acl.enable property to true. This enables fallback to native YARN ACLs if there is no Ranger policy.

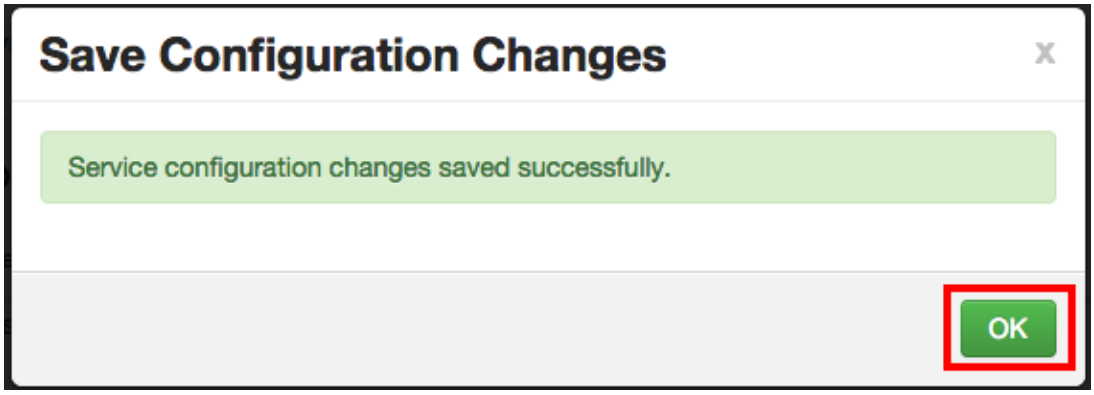
4. To save the configuration, click the green **Save** button on the black menu bar at the top of the page.



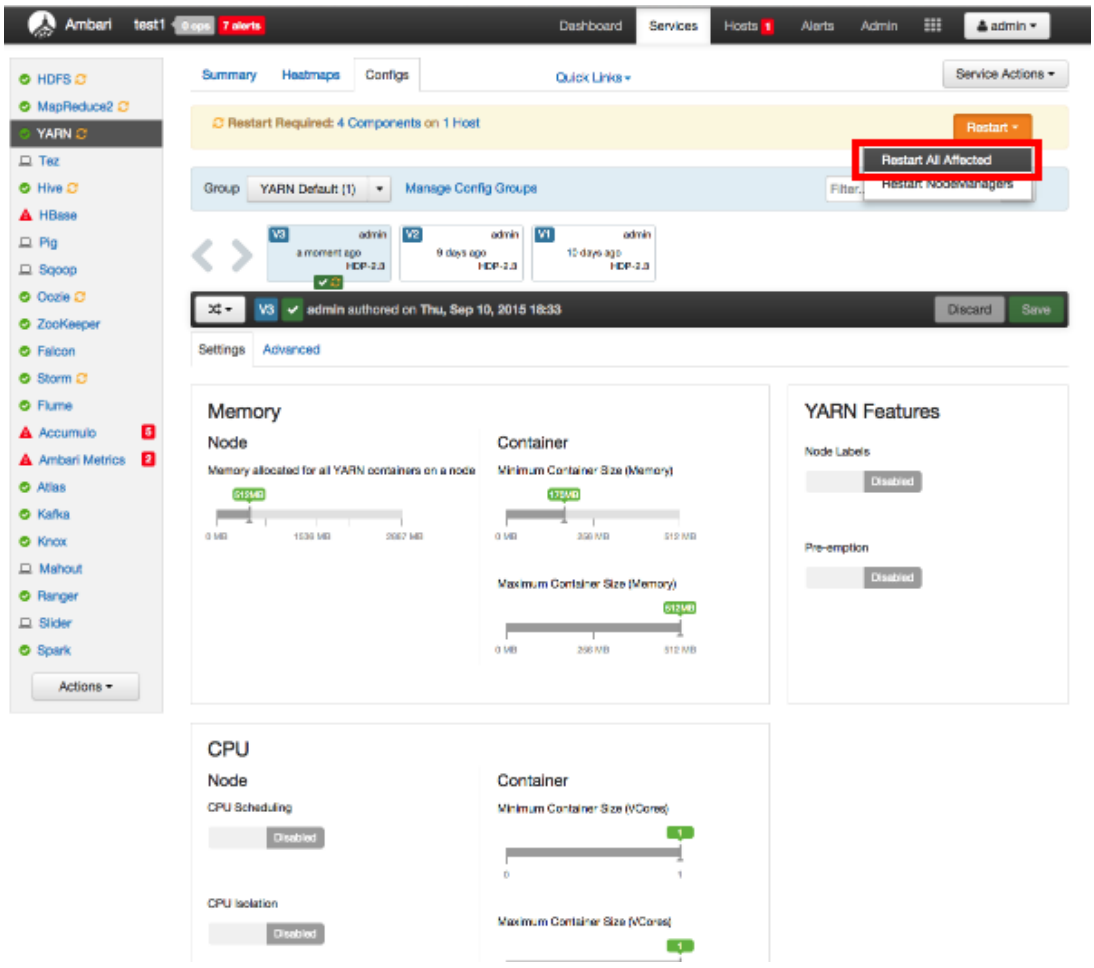
5. A Save Configuration pop-up appears. Type in a note describing the changes you just made, then click **Save**.



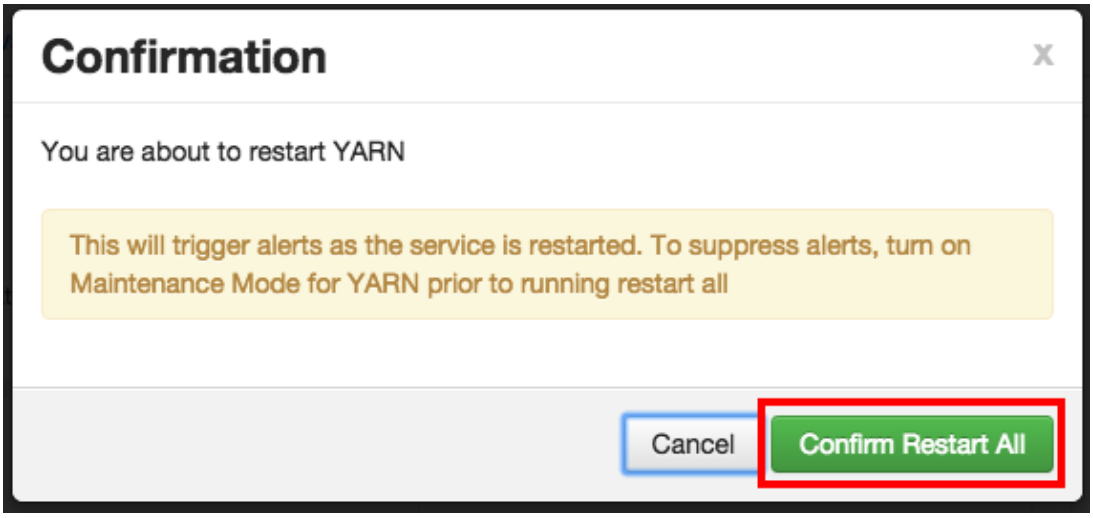
6. Click **OK** on the Save Configuration Changes pop-up.



7. A Restart Required message will be displayed at the top of the page. Click **Restart**, then select **Restart All Affected** to restart the YARN service and load the new configuration.



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



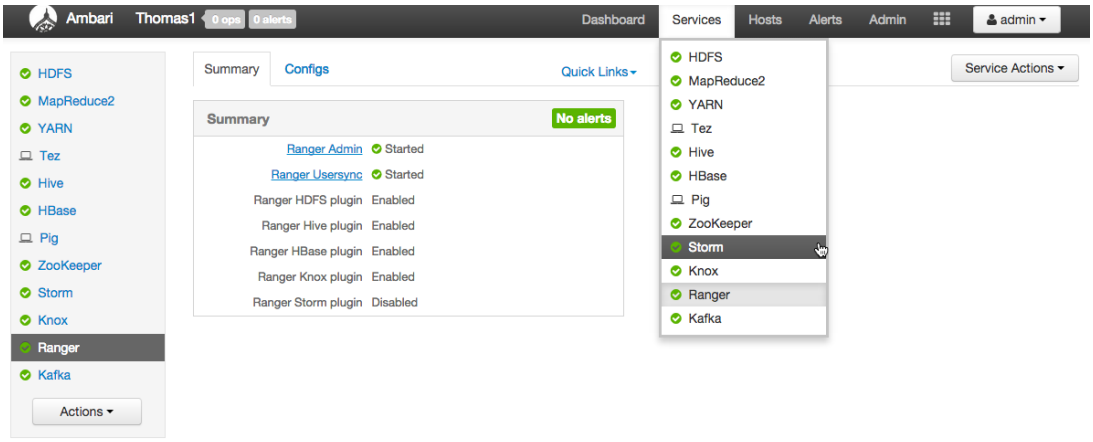
9. After YARN has been restarted, the Ranger plugin for YARN will be enabled.

5.7. Storm

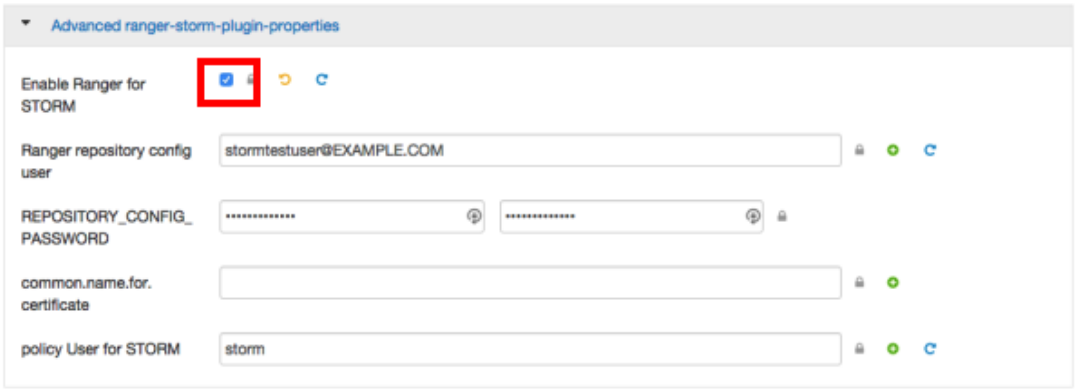
Before you can use the Storm plugin, you must first enable Kerberos on your cluster. To enable Kerberos on your cluster, see Enabling Kerberos Security in the Ambari Security Guide.

Use the following steps to enable the Ranger Storm plugin.

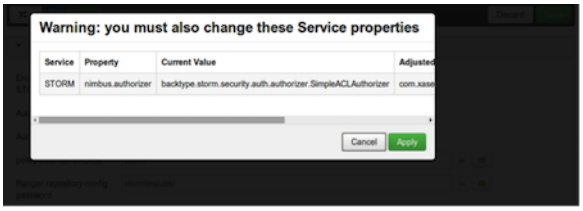
1. Select **Storm** from the Services tab in the top menu.



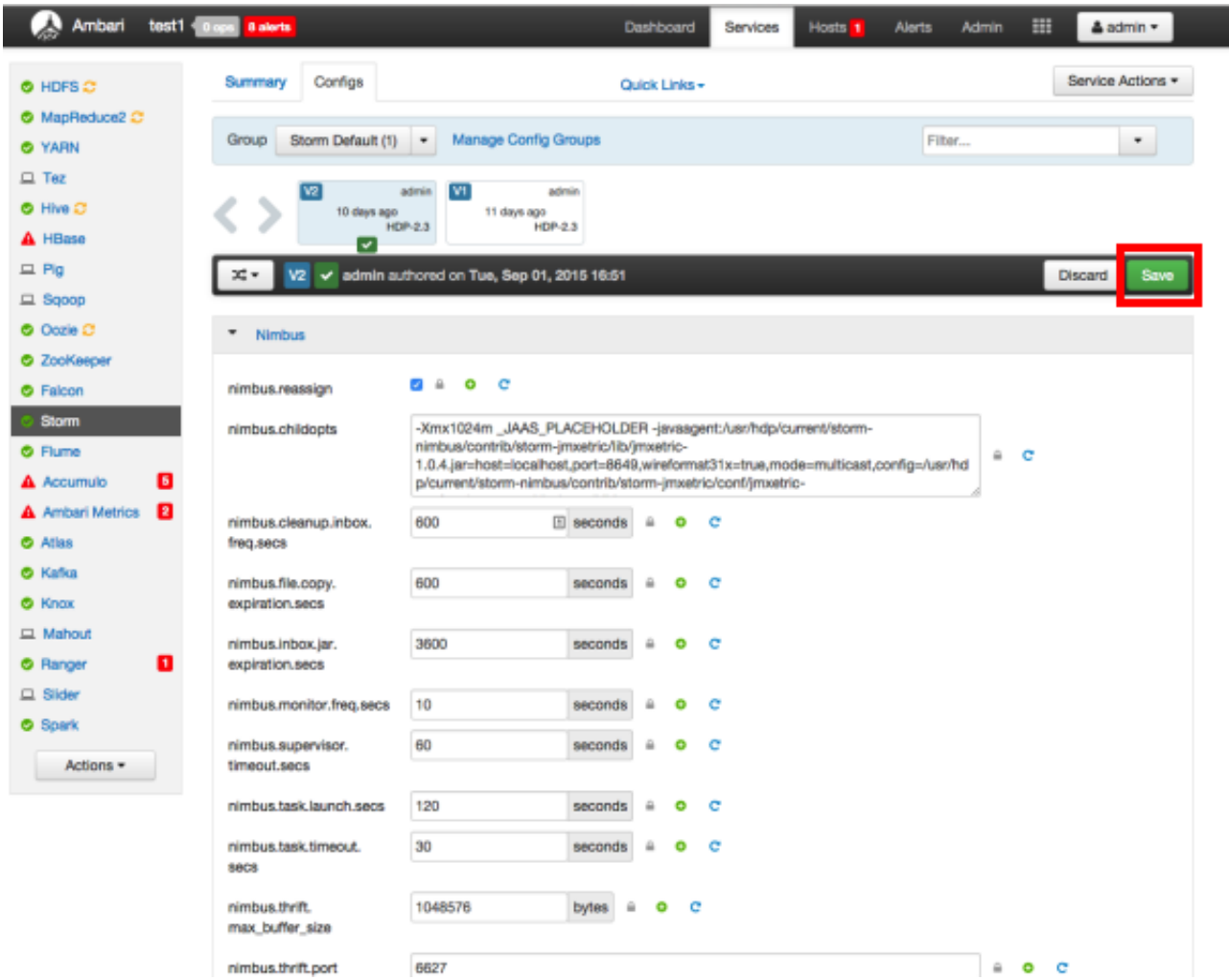
2. Click the **Configs** tab, then click the **Advanced** tab. Scroll down and click to open **Advanced ranger-storm-plugin-properties**.



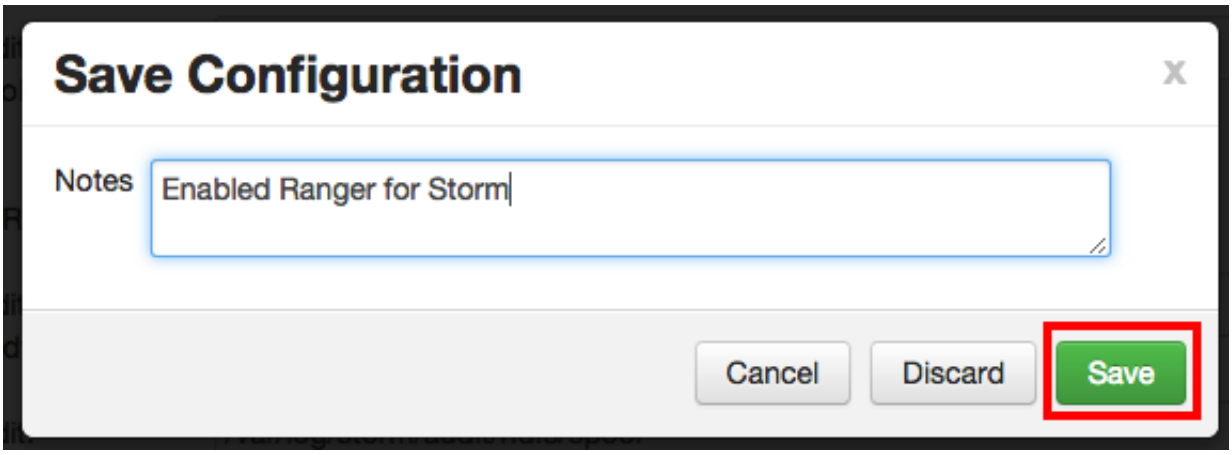
3. Select the **Enable Ranger for STORM** check box. A Warning pop-up appears. Click **Apply** to save the property updates.



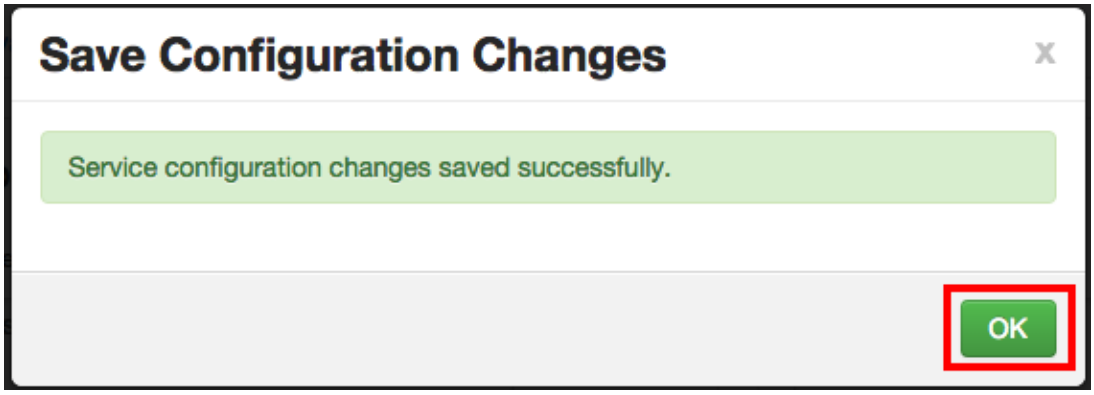
4. To save the configuration, click the green **Save** button on the black menu bar at the top of the page.



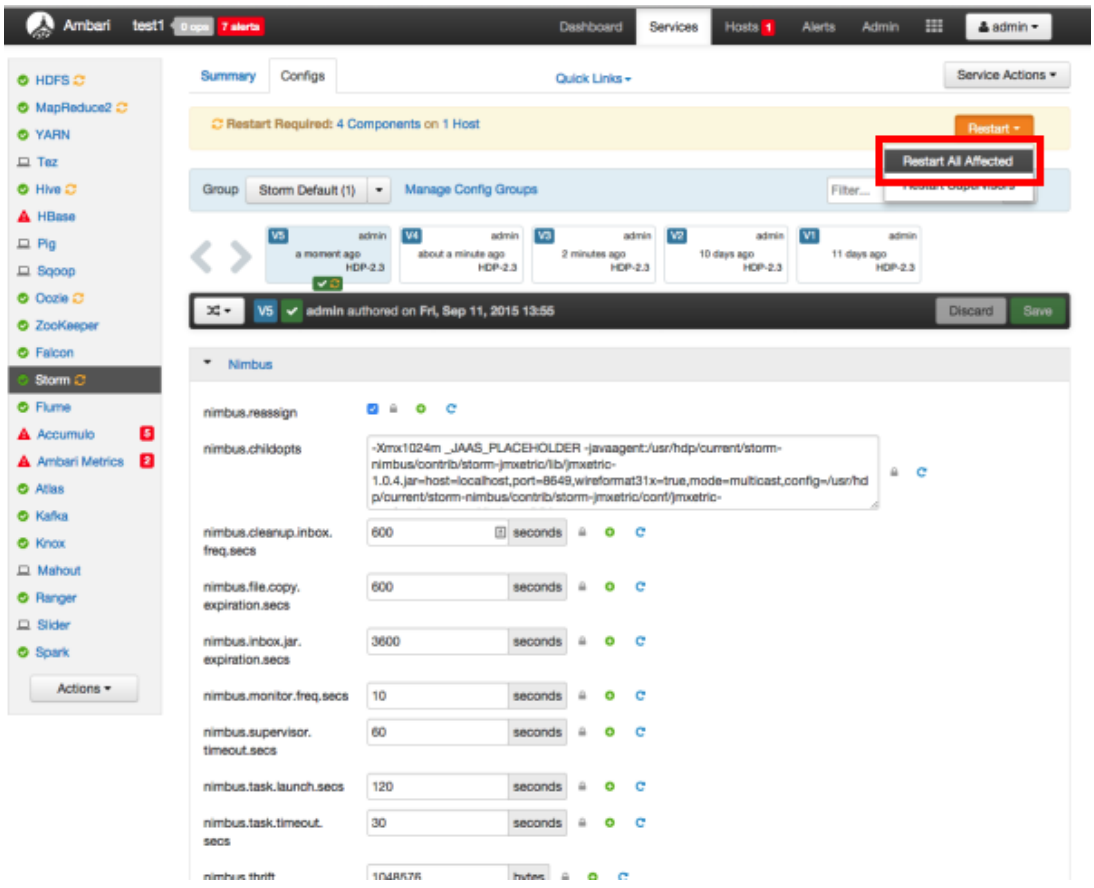
5. A Save Configuration pop-up appears. Type in a note describing the changes you just made, then click **Save**.



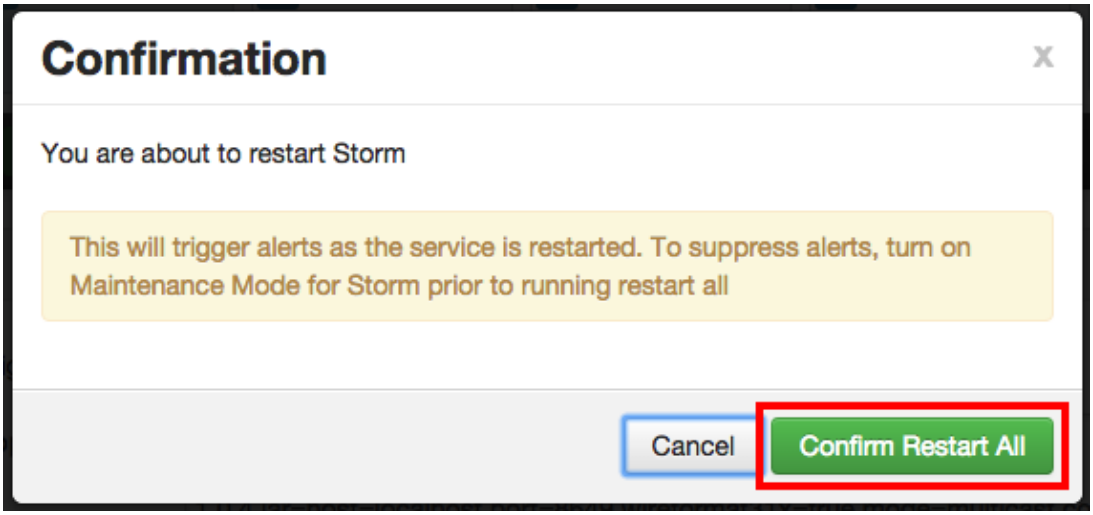
6. Click **OK** on the Save Configuration Changes pop-up.



7. A Restart Required message will be displayed at the top of the page. Click **Restart**, then select **Restart All Affected** to restart the Storm service and load the new configuration.



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



9. After Storm has been restarted, the Ranger plugin for Storm will be enabled.

5.8. Save Audits to HDFS

The following steps show how to save Ranger audits to HDFS for HBase. You can use the same procedure for other components.

1. From the Ambari dashboard, select the HBase service. On the Configs tab, scroll down and select **Advanced ranger-hbase-audit**. Select the **Audit to HDFS** check box.
2. Set the HDFS path where you want to store audits in HDFS:

   ```
   xasecure.audit.destination.hdfs.dir = hdfs://
   $NAMENODE_FQDN:8020/ranger/audit
   ```

   Refer to the `fs.defaultFS` property in the **Advanced core-site** settings.

   Note

   For NameNode HA, `NAMENODE_FQDN` is the cluster name. In order for this to work, `/etc/hadoop/conf/hdfs-site.xml` needs to be linked under `/etc/<component_name>/conf`.

3. Enable the Ranger plugin for HBase.
4. Make sure that the plugin sudo user should has permission on the HDFS Path:

   ```
   hdfs://NAMENODE_FQDN:8020/ranger/audit
   ```

   For example, we need to create a Policy for Resource : `/ranger/audit`, all permissions to user hbase.

5. Save the configuration updates and restart HBase.
6. Generate some audit logs for the HBase component.
7. Check the HFDS component logs on the NameNode:

```
hdfs://NAMENODE_FQDN:8020/ranger/audit
```

Note

For a secure cluster, use the following steps to test audit to HDFS for STORM/KAFKA/KNOX:

- In core-site.xml set the hadoop.proxyuser.<component>.groups property with value " * " or service user.
- For the Knox plugin there is one additional property to add to core-site.xml. Add hadoop.proxyuser.<component>.users property with value " * " or service user (i.e knox).
- Link to `/etc/hadoop/conf/core-site.xml` under `/etc/<component_name>/conf`.
- Verify the service user principal.
- Make sure that the component user has permissions on HDFS.

5.9. Save Audits to Solr

You can save and store Ranger audits to Solr if you have installed and configured the Solr service in your cluster.

It is recommended that Ranger audits be written to both Solr and HDFS. Audits to Solr are primarily used to enable queries from the Ranger Admin UI. HDFS is a long-term destination for audits – audits stored in HDFS can be exported to any SIEM system, or to another audit store.

To save Ranger audits to Solr:

1. From the Ambari dashboard, select the Ranger service. On the Configs tab, scroll down and select **Advanced ranger-admin-site**. Set the following property values:
   - `ranger.audit.source.type = solr`
   - `ranger.audit.solr.urls = http://solr_host:6083/solr/ranger_audits`
   - `ranger.audit.solr.username = ranger_solr`
   - `ranger.audit.solr.password = NONE`
2. Restart the Ranger service.
3. After the Ranger service has been restarted, you will then need to make specific configuration changes for each plugin to ensure that the plugin's data is captured in Solr.
4. For example, if you would like to configure HBase for audits to Solr, perform the following steps:

- Select the Audit to Solr checkbox in Advanced ranger-hbase-audit.
- Enable the Ranger plugin for HBase.
- Restart the HBase component.

5. Verify that the Ranger audit logs are being passed to Solr by opening one of the following URLs in a web browser:

   http://{RANGER_HOST_NAME}:6080/index.html#!/reports/audit/bigData

   http://{SOLR_HOST}:6083/solr/ranger_audits

6. 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 [77]
2. Hive [78]
3. HBase [78]
4. Knox [79]

6.1. HDFS

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

1. Create the system (OS) user `rangerhdfslookup`. 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 `rangerhdfslookup` by entering the following command:
   - `kadmin.local -q 'addprinc -pw rangerhdfslookup rangerhdfslookup@example.com`

Note

A single user/principal (e.g., rangerrepouser) can also be created and used across services.

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.

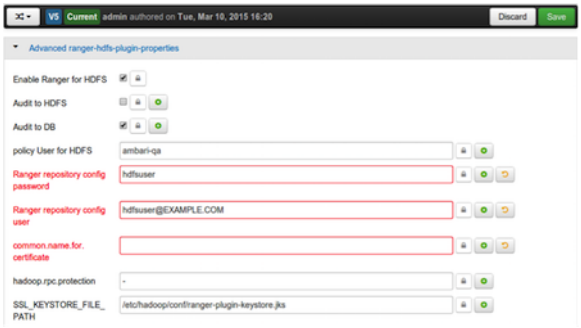


Table 6.1. HDFS Plugin Properties

Configuration Property Name	Value
Ranger repository config user	rangerhdfslookup@example.com

Configuration Property Name	Value
Ranger repository config password	rangerhdfslookup
common.name.for.certificate	blank

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

6.2. 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 `rangerhivelookup`. Make sure this user is synced to Ranger Admin (under *users/groups* tab in the Ranger Admin UI).
2. Create a Kerberos principal for `rangerhivelookup` by entering the following command:
   - `kadmin.local -q 'addprinc -pw rangerhivelookup rangerhivelookup@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 6.2. Hive Plugin Properties

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

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

6.3. HBase

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

1. Create the system (OS) user `rangerhbaselookup`. Make sure this user is synced to Ranger Admin (under *users/groups* tab in the Ranger Admin UI).
2. Create a Kerberos principal for `rangerhbaselookup` by entering the following command:

- `kadmin.local -q 'addprinc -pw rangerhbaselookup rangerhbaselookup@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 6.3. HBase Plugin Properties

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

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

6.4. Knox

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

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

2. Create a Kerberos principal for `rangerknoxlookup` by entering the following command:

   - `kadmin.local -q 'addprinc -pw rangerknoxlookup rangerknoxlookup@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 6.4. Knox Plugin Properties

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

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

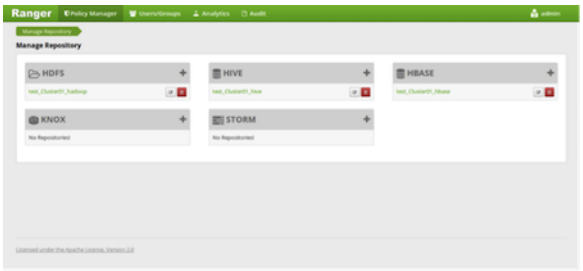
7. Open the Ranger Admin UI by entering the following information:

   - 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**.

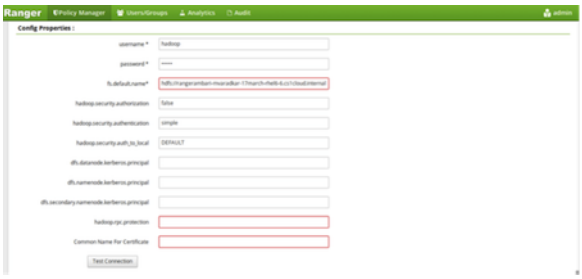
8. After you have successfully logged into the system, you will be redirected to the Policy Manager page.

Figure 6.1. Knox Policy Manager



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

Figure 6.2. Knox Repository Edit



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

Table 6.5. 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**.