Hortonworks Data Platform

Security Administration Tools Guide

(Oct 28, 2014)

docs.hortonworks.com

Hortonworks Data Platform : Security Administration Tools Guide

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 Tables

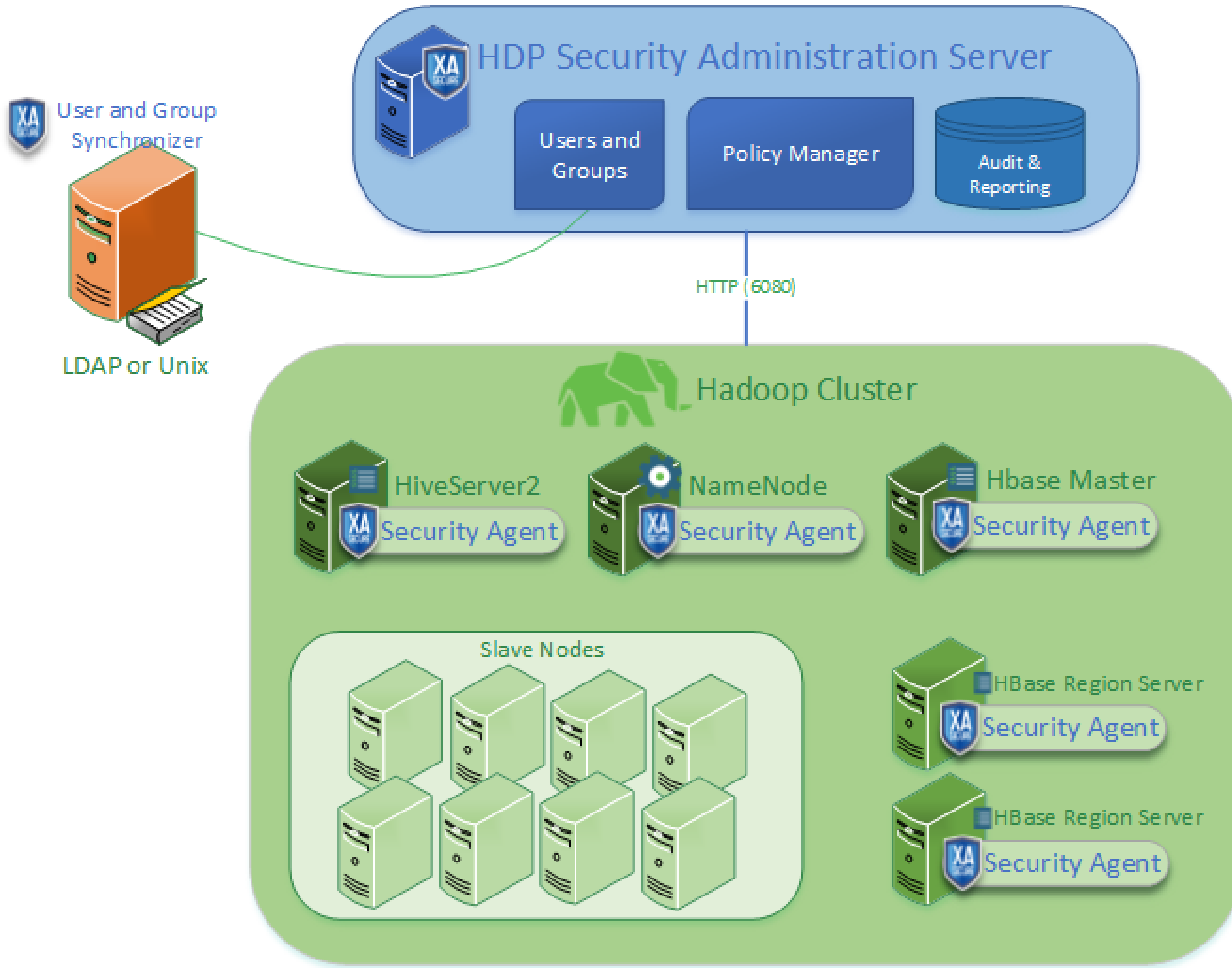
1. HDP Security Administration Overview

The HDP Security Administration provides the following security for Hadoop clusters:

- Authorization: Restricts access to explicit data as follows:
  - Fine-grained access control for HDFS, Hive, and Hbase
  - Role-based policies
  - Component-level enforcement
- Audit: Track and report on the following items in a central location:
  - Detailed access auditing for HDFS, Hive and Hbase
  - Admin action auditing
- Centralized Security Policies:
  - UI to centrally manage security policies
  - Delegated administration
  - Automated policy synchronization

1.1. HDP Security Administration Architecture

An HDP Security Administration deployment contains the following components:



- **HDP Security Administration server**: A central location to manage all security policies for Hadoop clusters, including access control, auditing, and reporting. It also provides delegated administration features to enable administration of policies for specific data to other users and groups.

- **User and Group Synchronizer**: Synchronizes user and group information between a UNIX server and the HDP Security Administration server. Allows the Unix system users on the host where the agent is installed to sign in to the Web UI with the same credentials as the local host.

- **Security Agent for HDFS**: Enforces the HDFS access control based on the policies managed on the HDP Security Administration server and provides audit and reporting HDFS activity.

- **Security Agent for Hive**: Enforces Hive (HiveServer2) access control based on the policies managed on the HDP Security Administration server and provides audit and reporting for Hive activity.

- **Security Agent for HBase**: Enforces HBase access control (via Hive2 service) based on the policies managed on the HDP Security Administration server and provides audit and reporting for HBase activity. Install an agent on the HBase Master and all HBase Regional servers.

The following table shows the ports used by the HDP Security Administration tools:

Table 1.1. Server and Agent Ports

Component	Listening Port	Connection to Port
HDP Security Administration server	6080[a] (HTTP)	3306 (JDBC/MySQL)
All Agents (HDFS, HBase and Hive)		6080* (HTTP)
User and Group Synchronization Agent	5151[b](Optional for remote Unix)	3306 (JDBC/MySQL)
MySQL	3306[c]	3306

[a]Ensure agent hosts can connect to the HDP SA server on port 6080.

[b] Make sure HDP Security Administration server can connect to port 5151 on the server were Unix Synchronization Service is installed.

[c]HDP Security Administrator server and agent servers should be able to connect to port 3306 on the server MySQL is installed. The agents insert the audit logs directly into the database

1.2. Performance Guidelines

- Policy Enforcement: Security Agents run within the process of NameNode, HiveServer2 and HBase Region Servers. It adds negligible overhead to the existing policy check and enforcement. The Security Agents can handle more than 50 simultaneous requests within less than 1.5 milliseconds.

  Recommendation: Limit the number of policies by grouping resources together and also where possible using wild cards or recursive options.

- Audits (log uploads to the server) : The Security Agent logs all access logs centrally to RDBMS. When MySQL is installed on a dedicated server with 4 Cores and 16 GB RAM, XASecure can handle up to 6500 logs/second with 375 concurrent requests. XASecure has inbuilt mechanism to log the event asynchronously without affecting the runtime performance of the cluster. If there is a sudden surge of event logs, XASecure will automatically buffer the logs and do deferred writing to database. If the surge of access requests lasts for longer period, then XASecure will throttle itself by discarding excess logs.

  Recommendation: For high-end systems, it is recommend that the database is properly tuned for memory caching and disk IO. It is also recommended to appropriately partition the database and archive historical data on regular intervals.

1.3. Download HDP Security Administration Tool Installers

The HDP Security Administration Suite is available to download from Hortonworks Add-ons page.

Download the components, as follows:

- HDP Security Administration server: Required for all deployments.

- UX-UserGroup Synchronizer: Optional. Provides Web UI authentication and automatically imports users and groups for policies.
- Security Agent for Hive: Only required if you are managing access or auditing HiveServer2.
- Security Agent for Hadoop: Only required if you are managing access or auditing HDFS.
- Security Agent for HBase: Only required if you are managing access or auditing HBase.

2. Install the HDP Security Administration Server

Install the HDP Security Administration on a Linux Server with at least 2 GB memory available for the HDP Security Administration web application. You can install the HDP Security Administration on a shared web application host. When in a test environment, you can also install the server on a node within the Hadoop cluster, such as the NameNode.

Note

Configure SSL after deploying the server and agents using the instructions in Configure SSL for Web UI and Server/Agent Communications.

2.1. Prerequisites

Before installing, ensure that you have met the following prerequisites::

- Hardware meets the minimum requirements, see System Requirements
- Oracle Java JDK 7 is installed, see Software Requirements
- MySQL Server and the `root` account credentials (that is the 'root'@'%' user id and password), see Database Requirements
- Root access to the hosts where you will be installing HDP Security Administration and/or the agents
- Download the JBDC driver for MySQL

2.1.1. System Requirements

Install the HDP Security Administration server on a Linux Server that has the following:

- Linux Host with at least 2 GB memory available for HDP Security Administration Web application
- Operating System: CentOS/RedHat, Ubuntu, or SuSe
- 2 GB of memory
- 10 GB disk space for HDP Security Administration logs
- Hadoop cluster (HDP) 2.1 or higher

Note

You can use a shared host for the HDP Security Administration server.

2.1.2. Software Requirements

The HDP Security Administration server requires:

- MySQL Server (hosted on the same system) or MySQL Client installed on the HDP Security Administration host.
- Oracle Java JDK version 7.x
- MySQL connector (JDBC driver)

The Security Agents require:

- MySQL connector (JDBC driver)

2.1.3. Database Requirements

The HDP Security Administration supports MySQL Server to store Policy, Auditing, and User data.

Installing HDP Security Administration requires the MySQL server hostname and root account credentials. The HDP Security Administration installation script creates the database and the db user automatically using the information you specify in the properties file.

After the installation of HDP Security Administration server, the MySQL database administrator must grant permission to the database user to access and write remotely from the NameNode, HiveServer2, and HBase (Master and Region Server) hosts.

2.2. Determine Authentication Method for Web UI

During the installation process, you will set up the authentication method for to the HDP Security Administration Web UI. The Web UI supports the following authentication methods:

- **Local HDP Security Administration Web UI user database**: Users and their credentials are stored in the HDP Security Administration database, and managed manually in the interface.
- **External LDAP** (supported services are OpenLDAP or AD): Users authenticate against an external LDAP service and their permission is determined by their group membership. Requires configuration during installation of the HDP Security Administration tools.
- **External Unix Server**: Users authenticate against an external Unix system using their credentials for that remote Unix system. Typically this is a server within the Hadoop cluster. This also requires configuration during both the installation of the HDP Security Administration tools and the installation of the Users and Groups Synchronizer Agent on the remote Unix System.

2.3. Install the HDP Security Administration Server

Install the HDP Security Administration server on a Linux host with at least 2 GB memory available for the Web application and at least 10 GB of diskspace for HDP Security Administration logs.

Note

You can install the HDP Security Administration on a shared web application host. Before installing ensure that the following prerequisites have been met, see Prerequisites.

2.3.1. Installation Set Up

Perform the following steps on the HDP Security Administration host.

1. Log on to the host as `root`.
2. Copy the installation file and extract as follows:
   a. Create a temporary directory, such as `/tmp/xasecure`:

      ```
      mkdir /tmp/xasecure
      ```

   b. Move the installation package to the temporary directory.

   c. Move the MySQL Connector Jar file to the temporary directory. Download the JAR from here.

   d. Extract the contents:

      ```
      tar xvf $xasecureinstallation.tar
      ```

   e. Go to the directory where you extracted the installation files:

      ```
      cd /tmp/xasecure/xasecure-$name-$build-version
      ```

3. Open the `install.properties` file for editing.
4. Define the parameters for the MySQL database setup:

Table 2.1. MySQL Database Install Parameters

Parameter	Value	Description
MYSQL_BIN	mysql	Specify the command to invoke MySQL. For example, `mysql`. This command is used by the script to invoke MySQL and connect to the database server.
MYSQL_CONNECTOR_JAR	*$path-to-mysql-connector*	Specify the absolute path on the local host to the JDBC driver for MySQL including filename.[a] For example, `/tmp/xasecure/mysql-connector-java.jar`
db_root_password	*$root-password*	The password for the root MySQL account. Used by the installation script to create the HDP SA database and database user.
db_host	*$mysql-host*	Host name of the system running MySQL server.
db_user	*$xadbuser*	Specify a name for the user account that the installer creates and is then used to write to the database.

Parameter	Value	Description
db_name	*$dbname*	Specify a name for the database that Installer creates during installation.
db_password	*$dbpassword*	Specify a password for the *$xadbuser* account created by the installer during installation.
audit_db_name	*$auditdb*	Specify a name for the audit database created by the installer during installation.
audit_db_user	*$auditdbuser*	Specify a name for the audit database account created by the installer during installation.
audit_db_password	*$auditdbupw*	Specify the password for the audit database account that the installer sets during installation.

[a] Download the JAR from here.

During installation, the script logs into the database, creates the HDP Security database named in the properties file, adds the user specified, and loads the MySQL tables.

Warning

DO NOT create the HDP Security database beforehand. If the database you specify already exists the HDP Security Administration tables are not added.

5. Define the HDP Security Administration Server URL, which is used Security Agents and users accessing the interface for Policies and Auditing:

Table 2.2. HDP Security Administration Server URL Parameters

Parameter	Value	Description
policymgr_external_url	*$url*	Specify the full URL to access the HDP Security Administration Web UI. For example, `http://pm-host:6080`.
policymgr_http_enabled	*$true-or-false*	Specify `true` to allow access to the HDP Security Administration Interface on HTTP or specify `false` to only allow HTTPS access to the interface.

6. In the *JAVA_HOME* parameter specify the path to the directory that contains the Java bin, for example:

```
#------------------------- JAVA CONFIG - BEGIN
 ----------------------------------

#
# Java Home path
#
JAVA_HOME='/usr/lib/jvm/jre-1.7.0-openjdk.x86_64'

#------------------------- JAVA CONFIG - END
 ----------------------------------
```

7. Use the following parameters and values in all configurations:

Table 2.3. Required Settings (for future enhancements)

Parameter	Value	Description
unix_user	`xasecure`	Parameter and value required in all configurations.
unix_group	`xasecure`	Parameter and value required in all configurations.

8. Use one of the following sets of parameters to define the Authentication for the HDP Security Administration Web UI:

   - Web UI administrators that are manually defined in the HDP Security Administration Web UI:

Table 2.4. HDP Security Administration Web UI Local Authentication Parameter

Parameter	Value	Description
remoteLoginEnabled	`false`	Specify `false` to manage users in the HDP Security Administration Web UI.

   - Web UI administrators authenticated against an external Unix Server:

Table 2.5. External Unix System Users Authentication Parameters

Parameter	Value	Description
authentication_method	`UNIX`	Specify `UNIX` to allow users to sign in to the HDP Security Administration Web UI using their credentials from an external Unix Server.
remoteLoginEnabled	`true`	Specify `true` to enabled remote login.
authServiceHostName	*$usersync-hostname*	Specify the remote Unix host name[a]
authServicePort	*$port*	Listening port of the Unix host where the UX-UserGroup Synchronizer will be installed, the default port is 5151.

[a] Requires installation of the UX-UserGroup Synchronizer.

Note

Requires installation of the User and Group Synchronizer Agent on the remote Unix Server.

The following is an example allowing HDP Sandbox users to access HDP Security Administration Web UI:

```
# ------- UNIX User CONFIG ----------------
#
unix_user=xasecure
unix_group=xasecure

#
# ------- UNIX User CONFIG  - END ----------------
```

```
#

#
# UNIX authentication service for Policy Manager
#
# PolicyManager can authenticate using UNIX username/password
# The UNIX server specified here as authServiceHostName needs to be
 installed with xasecure-unix-ugsync package.
# Once the service is installed on authServiceHostName, the UNIX username/
password from the host <authServiceHostName> can be used to login into
 policy manager
#
# ** The installation of xasecure-unix-ugsync package can be installed
 after the policymanager installation is finished.
#
#LDAP|ACTIVE_DIRECTORY|UNIX|NONE
authentication_method=UNIX
remoteLoginEnabled=true
authServiceHostName=sandbox
authServicePort=5151
```

- Web UI administrators authenticated against an external LDAP (either OpenLDAP or Active Directory service):

Table 2.6. External LDAP Service Authentication Parameters

Parameter	Value	Description
authentication_method	LDAP	Specify LDAP to allow users to sign in to the HDP Security Administration Web UI using their credentials from an external LDAP service.
remoteLoginEnabled	true	Specify true to enabled remote login.
authServiceHostName	*$usersync-hostname*	Specify the LDAP service host name or IP address.[a]
authServicePort	*$port*	Listening port of the LDAP service, default port is 389.

[a] Requires installation of the UX-UserGroup Synchronizer.

The following is an example of the configuration parameters for OpenLDAP installed on HDP Sandbox:

```
# ------- UNIX User CONFIG ----------------
#
unix_user=xasecure
unix_group=xasecure

#
# ------- UNIX User CONFIG  - END ----------------
#

#
# UNIX authentication service for Policy Manager
#
# PolicyManager can authenticate using UNIX username/password
# The UNIX server specified here as authServiceHostName needs to be
 installed with xasecure-unix-ugsync package.
```

```
# Once the service is installed on authServiceHostName, the UNIX username/
password from the host <authServiceHostName> can be used to login into
 policy manager
#
# ** The installation of xasecure-unix-ugsync package can be installed
 after the policymanager installation is finished.
#
#LDAP|ACTIVE_DIRECTORY|UNIX|NONE
authentication_method=LDAP
remoteLoginEnabled=true
authServiceHostName=sandbox
authServicePort=389
```

9. Save the `install.properties` file.

The following example shows the HDP Security Administration server `install.properties` for a system that does not allow remote login of Web UI administrators:

```
#
# This file provides list of deployment variables for the Policy Manager Web
 Application
#
#------------------------- MYSQL CONFIG - BEGIN
 ----------------------------------

#
# The executable path to be used to invoke command-line MYSQL
#
MYSQL_BIN='mysql'

#
# Location of mysql client library (please check the location of the jar file)
#
MYSQL_CONNECTOR_JAR=/usr/share/java/mysql-connector-java.jar

#
# MYSQL password for the MYSQL root user-id
# ***************************************************************************
# ** If the password is left empty or not-defined here,
# ** it will be prompted to enter the password during installation process
# ***************************************************************************
#

db_root_password=hadoop
db_host=localhost

#
# MySQL UserId used for the XASecure schema
#
db_name=xasecure
db_user=xaadmin
db_password=hadoop

#
# MySQL UserId for storing auditlog infromation
#
# * audit_db can be same as the XASecure schema db
# * audit_db must exists in the same ${db_host} as xaserver database
 ${db_name}
```

```
# * audit_user must be a different user than db_user (as audit user has access
 to only audit tables)
#
audit_db_name=xasecure
audit_db_user=xalogger
audit_db_password=hadoop

#------------------------- MYSQL CONFIG - END
 ----------------------------------

#
# ------- PolicyManager CONFIG ----------------
#

policymgr_external_url=http://localhost:6080
policymgr_http_enabled=true

#
# ------- PolicyManager CONFIG - END ---------------
#


#
# UNIX authentication service for Policy Manager
#
# PolicyManager can authenticate using UNIX username/password
# The UNIX server specified here as authServiceHostName needs to be installed
 with xasecure-unix-ugsync package.
# Once the service is installed on authServiceHostName, the UNIX username/
password from the host <authServiceHostName> can be used to login into Policy
 Manager
#
# ** The installation of xasecure-unix-ugsync package can be installed after
 the policymanager installation is finished.
#

remoteLoginEnabled=false
authServiceHostName=
authServicePort=

#
# -----------------------------------------------------------
#

# ######  DO NOT MODIFY ANY VARIABLES BELOW #########################
#
# --- These deployment variables are not to be modified unless you understand
 the full impact of the changes
#
###################################################

app_home=$PWD/app
war_file=${PWD}/war/xa_portal.war
TMPFILE=$PWD/.fi_tmp
LOGFILE=$PWD/logfile
LOGFILES="$LOGFILE"

JAVA_BIN='java'
JAVA_VERSION_REQUIRED='1.7'
JAVA_ORACLE='Java(TM) SE Runtime Environment'
```

```
db_create_user_file=${PWD}/db/create_dev_user.sql
db_core_file=${PWD}/db/xa_core_db.sql
db_assert_file=${PWD}/db/reset_asset.sql
```

2.3.2. Run the HDP Security Administration Installation Script

After configuring the `install.properties` file, install the HDP Security Administration server as `root`:

1. Log on to the Linux system as root and go to the directory where you extracted the HDP Security Administration installation files:

   ```
   cd /tmp/xasecure/xasecure-policymgr-$build-version
   ```

2. Run the installation script:

   ```
   # ./install.sh
   ```

Once the `install.sh` execution is complete, the HDP Security Administration Web UI is accessible.

Using a web browser, go to the HDP Security Administration application at `http://$policymgr_host:6080`. If this is the first installation, sign in with the default account, `admin\admin`.

Caution

Change the `admin` user account password as soon as possible.

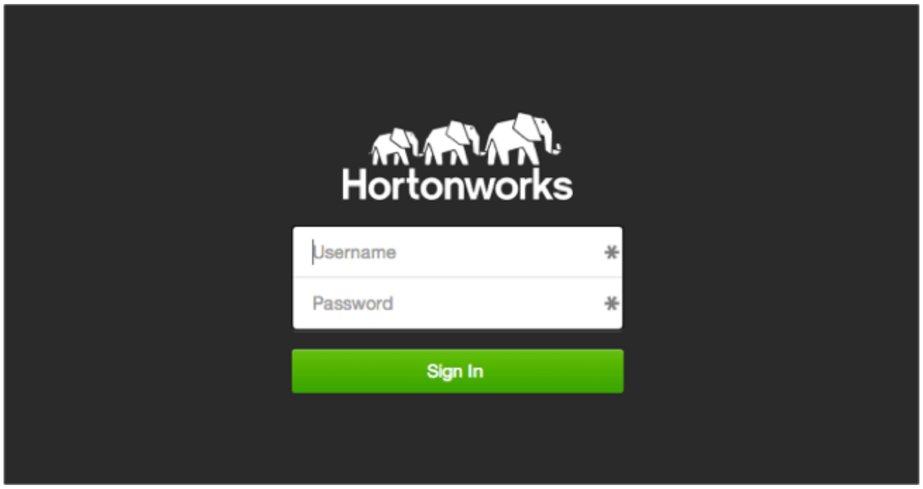
2.4. Change the Default Password

The HDP Security Administration Interface default port is `6080`.

To sign in and change the password:

1. Open a browser and type `http://policymgr-host:6080` in the address bar.

   The log in screen displays.



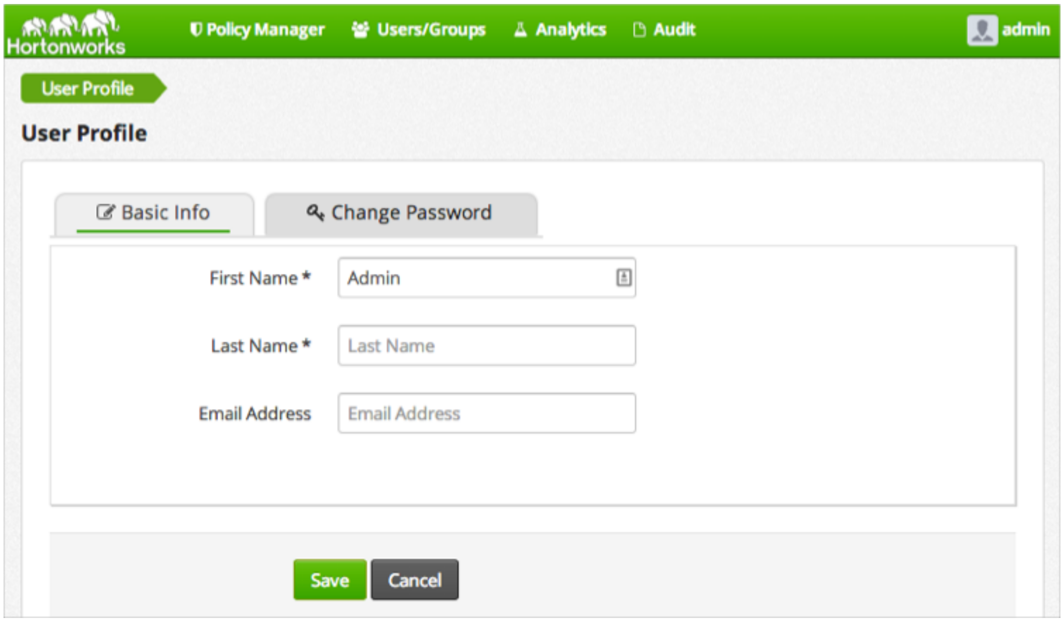
2. Enter the default account credentials. In the first field enter `admin` and in the second field `admin`.

3. Click **Sign In**.

   The HDP Security Administration Web UI Home page displays.

4. In the upper right corner, click **admin** > **Profile**.

   The Basic Info tab displays.



User Profile

First Name * Admin

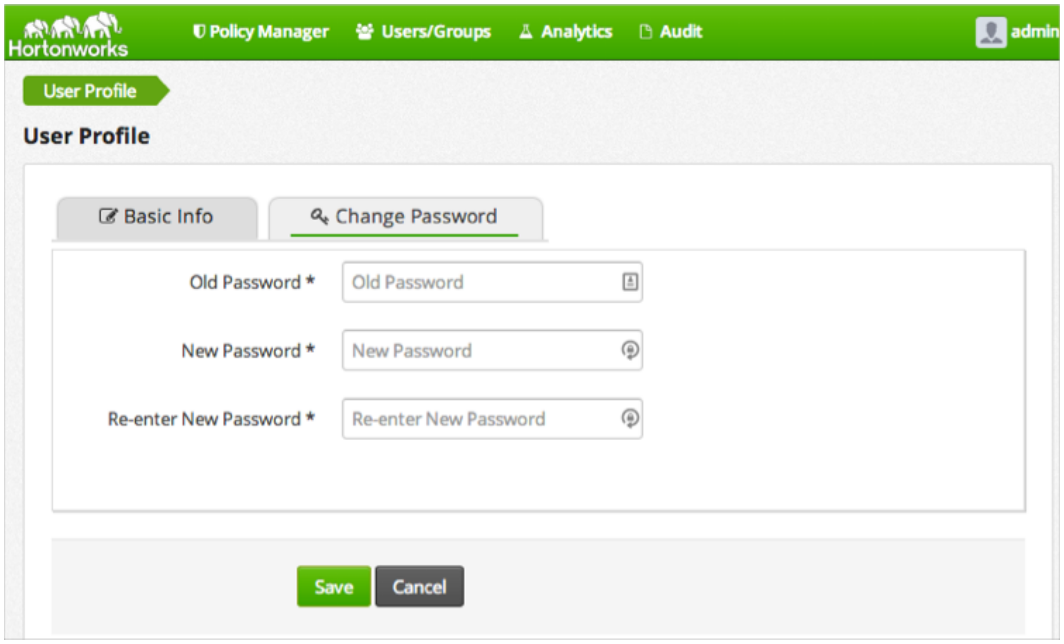
Last Name * Last Name

Email Address Email Address

Tip

Information on the admin profile cannot be changed.

5. Go the **Password** tab, type the old password and the new one to change the password.



6. Click **Save**.

Log out and then back in using the new password.

3. Setting Up the User and Group Agent

The HDP Security Administration tools have two types of users:

- **Web UI administrators**: Users who require access to the Web UI to manage Hadoop cluster Policies and Audit and Report on Hadoop cluster activity. The user and group synchronizer is required when authenticating Web UI Administrators against an external Unix Server.
- **Hadoop cluster users**: Users who require access to the Hadoop cluster data and therefore are named in ACL Policies created on the HDP Security Administrator Web UI. Use the User and Group Agent to synchronize accounts to use in policies from an external source such as a Unix Server or LDAP Service.

Tip

HDP Security Administration tools can be used to monitor Hadoop cluster activity without restricting access to data in HDFS, Hive, or HBase repositories. By default, when a Hadoop cluster repository is added to the HDP Security Administration, the repository the default setting allows all access.

3.1. Set up the User and Group Agent

Install the Unix User and Group Synchronizer (`uxugsync`) component after installing the HDP Security Administration server, see Install the HDP Security Administration Server. This component synchronizes users and groups from an external Unix host or LDAP service to the HDP Security Administration server. This agent is required when allowing remote authentication of Web UI administrators with a Unix System.

UX-UserGroup Synchronizer provides the following functionality:

- User and group data for creating policies
- Authentication for HDP Security Administration accounts using the same credentials as the external host where the synchronizer is installed

Note

- Before installing the UX-UserGroup Synchronizer verify that Java 7 JRE or JDK is installed by running the following command:

  ```
  java -version
  ```

- The user and group agent is not required when authenticating users against an external LDAP service.

3.1.1. Installation Set Up for Unix Authentication and User/Group Synchronization

To synchronize user and groups and/or allow users from a remote Unix system to log into the Web UI perform the following steps on the remote Unix host:

1. Log on to the host as root.
2. Copy the installation files to the target host and extract the files:

   a. Create a temporary directory, such as `/tmp/xasecure`:

   ```
   mkdir /tmp/xasecure
   ```

   b. Move the installation package into the temporary directory along with the MySQL Connector Jar.

   c. Extract the contents:

   ```
   tar xvf $xasecureinstallation.tar
   ```

   d. Go to the directory where you extracted the installation files:

   ```
   cd /tmp/xasecure/xasecure-$name-$build-version
   ```

3. Open the `install.properties` file for editing.
4. Set the UNIX remote authentication and user/group synchronization parameters:

Table 3.1. Unix Authentication and User/Group Sync Installation Parameters

Parameter	Value	Description
POLICY_MGR_URL	*$URL*	Complete URL including protocol and port to the HDP Security Administration server. For example, `http://policy-manager:6080`.
MIN_UNIX_USER_ID_TO_SYNC	*$integer*	Specify the minimum user ID level to synchronize with HDP Security Administration. Typically system users are created with IDs lower than 1000. For example, `1000`
SYNC_INTERVAL	*$minutes*	Specify the interval in minutes, the default when no value is set is `360`.
SYNC_SOURCE	unix	Specify `unix` to allow remote authentication and user/group synchronization for users and groups on the host system.

Example `install.properties` file for HDP Security Administration Server configured for UNIX authentication and UNIX user and group synchronization:

```
#
# The following URL should be the base URL for connecting to the policy
 manager web application
# For example:
#
#  POLICY_MGR_URL = http://policymanager.xasecure.net:6080
#

POLICY_MGR_URL = http://policymgr:6080


# Minumum Unix User-id to start SYNC.
# This should avoid creating UNIX system-level users in the Policy Manager
```

```
#
MIN_UNIX_USER_ID_TO_SYNC = 1000

# sync interval in minutes
# user, groups would be synced again at the end of each sync interval
# defaults to 5min if SYNC_SOURCE is unix
# defaults to 360min if SYNC_SOURCE is ldap
SYNC_INTERVAL =

# sync source,  only unix and ldap are supported at present
# defaults to unix
SYNC_SOURCE = unix
```

5. Save the `install.properties` file.

3.1.2. Installation Set Up for LDAP Service User/Group Synchronization

When synchronizing users from an LDAP service the agent can be installed on the HDP Security Administration server.

Note

The LDAP configuration in the User and Group Synchronizer Agent is only used for synchronization. Authentication is configured during the installation of the HDP Security Administration Server,

To synchronize user and groups from an LDAP service:

1. Log on to the host as root.
2. Copy the installation files to the target host and extract the files:

   a. Create a temporary directory, such as `/tmp/xasecure`:

      ```
      mkdir /tmp/xasecure
      ```

   b. Move the installation package into the temporary directory along with the MySQL Connector Jar.

   c. Extract the contents:

      ```
      tar xvf $xasecureinstallation.tar
      ```

   d. Go to the directory where you extracted the installation files:

      ```
      cd /tmp/xasecure/xasecure-$name-$build-version
      ```

3. Open the `install.properties` file for editing.
4. Configure the LDAP user and group synchronization parameters:

Table 3.2. LDAP User/Group Sync Installation Parameters

Parameter	Value	Description
POLICY_MGR_URL	*$URL*	Complete URL including protocol and port to the HDP Security

Parameter	Value	Description
		Administration server. For example, `http://policy-manager:6080`.
`MIN_UNIX_USER_ID_TO_SYNC`	*`$integer`*	Specify the minimum user ID level to synchronize with HDP Security Administration. Typically system users are created with IDs lower than 1000. For example, `1000`
`SYNC_INTERVAL`	*`$minutes`*	Specify the interval in minutes, the default when no value is set is `360`.
`SYNC_SOURCE`	`ldap`	Specify `unix` to allow remote authentication and user/group synchronization for users and groups on the host system.
`SYNC_LDAP_URL`	*`$URL`*	Specify the full URL to the LDAP service, including port number. For example, `ldap://ldap-host:389`.[a]
`SYNC_LDAP_BIND_DN`	*`$userDN`*	Specify the user DN for the LDAP account to the LDAP service.
`SYNC_LDAP_BIND_PASSWORD`	*`$password`*	Specify the password for the LDAP account.
`SYNC_LDAP_USER_SEARCH_BASE`	*`$BaseDN`*	Specify the base DN for the user and groups search.
`SYNC_LDAP_USER_SEARCH_SCOPE`	`base`, `one` or `sub`	Specify the search type (base, one or sub) for the search.
`SYNC_LDAP_USER_OBJECT_CLASS`	*`$class`*	Specify the ObjectClass for users and groups to sync. For example, `person`.[b]
`SYNC_LDAP_USER_SEARCH_FILTER`	*`$filter`*	Specify the value to filter the search results on for synchronization. For example, `dept=engineer`.
`SYNC_LDAP_USER_NAME_ATTRIBUTE`	*`$attribute`*	Specify the attribute to return as the user or group name. This is the value synchronized.
`SYNC_LDAP_USERNAME_CASE_ CONVERSION`	`lower`	Converts the user name case on import. The possible values are `lower` or `upper`.
`SYNC_LDAP_GROUPNAME_CASE_ CONVERSION`	`lower`	Converts the group name case on import. The possible values are `lower` or `upper`.

[a] Only Active Directory and OpenLDAP are supported.

[b] The default is `person`.

Example `install.properties` file for HDP Security Administration Server configured for LDAP authentication and LDAP user and group synchronization:

```
#
# The following URL should be the base URL for connecting to the policy
 manager web application
# For example:
#
#  POLICY_MGR_URL = http://policymanager.xasecure.net:6080
#
POLICY_MGR_URL = http://policymgr:6080

#
# Minumum Unix User-id to start SYNC.
# This should avoid creating UNIX system-level users in the Policy Manager
#
MIN_UNIX_USER_ID_TO_SYNC = 1000

# sync interval in minutes
# user, groups would be synced again at the end of each sync interval
# defaults to 5min if SYNC_SOURCE is unix
# defaults to 360min if SYNC_SOURCE is ldap
SYNC_INTERVAL =

# sync source,  only unix and ldap are supported at present
# defaults to unix
SYNC_SOURCE = ldap

# ---------------------------------------------------------------
# The following properties are relevant only if SYNC_SOURCE = ldap
# ---------------------------------------------------------------

# URL of source ldap
# a sample value would be:  ldap://ldap.example.com:389
# Must specify a value if SYNC_SOURCE is ldap
SYNC_LDAP_URL = ldap://sandbox:389

# ldap bind dn used to connect to ldap and query for users and groups
# a sample value would be cn=admin,ou=users,dc=hadoop,dc=apache,dc-org
# Must specify a value if SYNC_SOURCE is ldap
SYNC_LDAP_BIND_DN = cn=admin,ou=users,dc=hadoop,dc=apache,dc-org

# ldap bind password for the bind dn specified above
# please ensure read access to this file  is limited to root, to protect the
 password
# Must specify a value if SYNC_SOURCE is ldap
# unless anonymous search is allowed by the directory on users and group
SYNC_LDAP_BIND_PASSWORD =

# search base for users
# sample value would be ou=users,dc=hadoop,dc=apache,dc=org
SYNC_LDAP_USER_SEARCH_BASE = ou=users,dc=hadoop,dc=apache,dc=org


# search scope for the users, only base, one and sub are supported values
# please customize the value to suit your deployment
# default value: sub
SYNC_LDAP_USER_SEARCH_SCOPE = sub

# objectclass to identify user entries
# please customize the value to suit your deployment
# default value: person
SYNC_LDAP_USER_OBJECT_CLASS = person

# optional additional filter constraining the users selected for syncing
# a sample value would be (dept=eng)
# please customize the value to suit your deployment
# default value is empty
SYNC_LDAP_USER_SEARCH_FILTER =

# attribute from user entry that would be treated as user name
# please customize the value to suit your deployment
# default value: cn
SYNC_LDAP_USER_NAME_ATTRIBUTE = cn
```

```
# UserSync - Case Conversion Flags
# possible values:  none, lower, upper
SYNC_LDAP_USERNAME_CASE_CONVERSION=lower
SYNC_LDAP_GROUPNAME_CASE_CONVERSION=lower
```

5. Save the `install.properties` file.

3.1.3. Run the Agent Installation Script

After configuring the `install.properties` file, install the agent as `root`:

1. Log on to the Linux system as root and go to the directory where you extracted the installation files:

   ```
   cd /tmp/xasecure/xasecure-$name-$build-version
   ```

2. Run the agent installation script:

   ```
   # ./install.sh
   ```

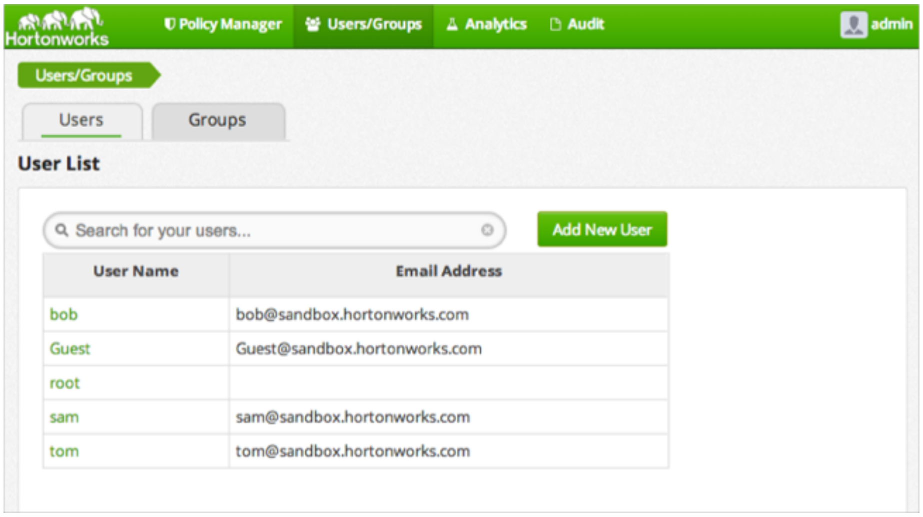
3.2. Verify User/Group Synchronizer

Once the synchronizer is installed, user and group information displays on the User/Group tab in the HDP Security Administration interface.

To verify that the user and groups uploaded:

1. Sign in to the Web UI.
2. Click **Users/Groups**.

   The Users tab displays.



User List

User Name	Email Address
bob	bob@sandbox.hortonworks.com
Guest	Guest@sandbox.hortonworks.com
root	
sam	sam@sandbox.hortonworks.com
tom	tom@sandbox.hortonworks.com

If the agent is not online, no user or group data displays.

4. Configure Repositories and Install Security Agents

HDP Security Administration tools allow you to audit activity and enforce access policies for up to ten different Hadoop clusters. Access Policies and Audited events are created and stored in the HDP Security Administration server and pushed to Security Agents installed on Hadoop cluster nodes.

The Security Agents integrate with data services in the Hadoop cluster to enforce access policies and audit activity. The agents are installed on cluster nodes as follows:

- HDFS Security Agent is installed on the NameNode host and in HA (High Availability) clusters also on the stand-by NN.
- Hive Security Agent is installed on the HiveServer2 host.
- HBase Security Agents are installed on each HBase Master and Region Server host.

4.1. Add HDFS Repositories

The HDFS repository contains access policies for the Hadoop cluster HDFS. The Security Agent integrates with the NameNode service on the NameNode host. The agent enforces the policy's configured in the HDP Security Administration Web UI and sends HDFS audit information to the portal where it can be viewed and reported on from a central location.

Warning

In Ambari managed environments additional configuration is required. Ensure that you carefully follow the steps outlined in the Configure Hadoop Agent to run in Ambari Environments.

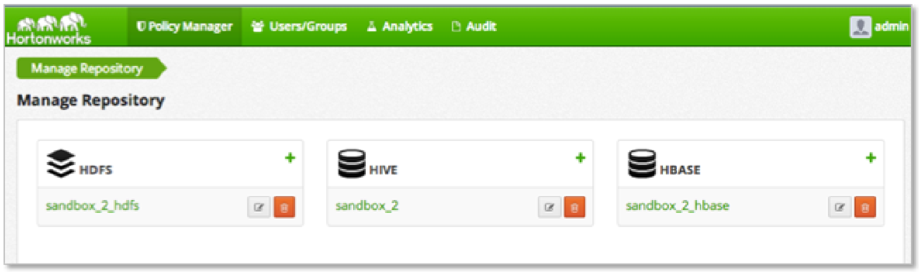
4.1.1. Add a HDFS Repository

Add HDFS repositories after the Hadoop environment is fully operational. During the initial set up of the repository, Hortonworks recommends testing the connection from the HDP Security Administration Web UI to the NameNode to ensure that the agent will be able to connect to the server after installation is complete.

4.1.1.1. Create a HDFS Repository

Before installing the agent on the NameNode, create a HDFS Repository as follows:

1. Sign in to the HDP Security Administration Web UI and click **Policy Manager**.

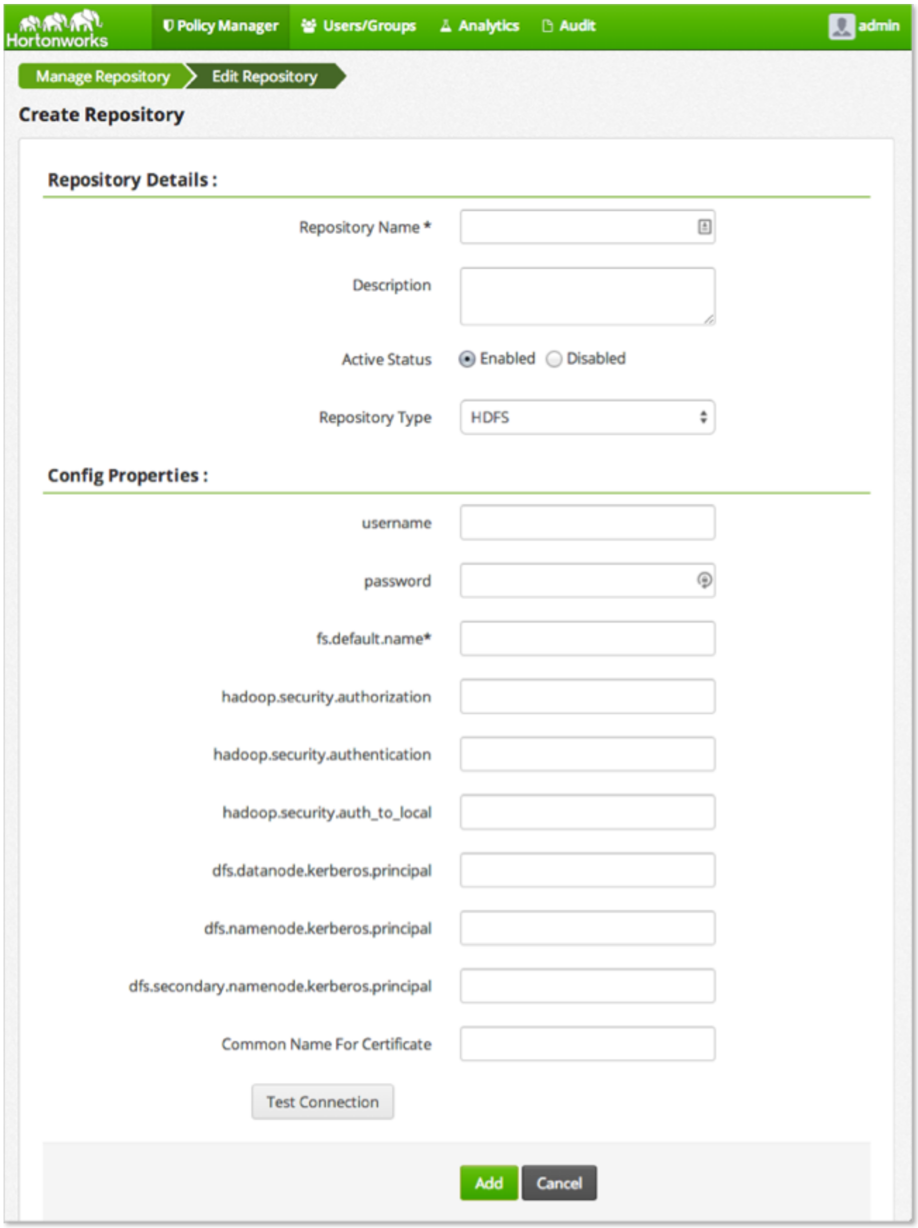


Manage Repository

HDFS	sandbox_2_hdfs
HIVE	sandbox_2
HBASE	sandbox_2_hbase

2. Next to HDFS, click the + (plus symbol).

   The Create Repository page displays.



Create Repository

Repository Details :

Repository Name *	
Description	
Active Status	● Enabled ○ Disabled
Repository Type	HDFS

Config Properties :

username	
password	
fs.default.name*	
hadoop.security.authorization	
hadoop.security.authentication	
hadoop.security.auth_to_local	
dfs.datanode.kerberos.principal	
dfs.namenode.kerberos.principal	
dfs.secondary.namenode.kerberos.principal	
Common Name For Certificate	

3. Complete the Repository Details:

Table 4.1. Policy Manager Repository Details

Label	Value	Description
Repository Name	*`$name`*	Specify a unique name for the repository, you will need to specify the same repository name in the agent installation properties. For example, `clustername_hdfs`.
Description	*`$description-of-repo`*	Enter a description up to 150 characters.
Active Status	`Enabled` or `Disabled`	Enable or disable policy enforcement for the repository.
Repository type	`HDFS`, `Hive`, or `HBase`	Select the type of repository, HDFS.
User name	*`$user`*	Specify a user name on the remote system with permission to establish the connection, for example `hdfs`.
Password	*`$password`*	Specify the password of the user account for connection.

4. Complete the security settings for the Hadoop cluster, the settings must match the values specified in the `core-site.xml` file as follows:

Table 4.2. Repository HDFS Required

Label	Value	Description
fs.default.name	*`$hdfs-url`*	HDFS URL, should match the setting in the Hadoop `core-site.xml` file. For example, `hdfs://sandbox.hortonworks.com:8020`
hadoop.security.authorization	`true` or `false`	Specify the same setting found in the core-site.xml.
hadoop.security.authentication	`simple` or `kerberos`	Specify the type indicated in the `core-site.xml`.
hadoop.security.auth_to_local	*`$usermapping`*	Must match the setting in the core-site.xml file. For example: `RULE:[2:$1@$0]([rn]m@.*)s/.*/yarn/ RULE:[2:$1@$0](jhs@.*)s/.*/mapred/ RULE:[2:$1@$0]([nd]n@.*)s/.*/hdfs/ RULE:[2:$1@$0](hm@.*)s/.*/hbase/ RULE:[2:$1@$0](rs@.*)s/.*/hbase/ DEFAULT`
dfs.datanode.kerberos.principal	*`$dn-principal`*	Specify the Kerberos DataNode principal name.
dfs.namenode.kerberos.principal	*`$nn-principal`*	Specify the Kerberos NameNode principal name.
dfs.secondary.namenode.kerberos.principal	*`$secondary-nn-principal`*	Specify the Kerberos Secondary NN principal name.
Common Name For Certificate	*`$cert-name`*	Specify the name of the certificate.

5. Click **Test Connection**.

   If the server can connect to HDFS, the connection successful message displays. If the connection fails, go to the troubleshooting appendix.

6. After making a successful connection, click **Save**.

4.1.1.2. Install the HDFS Agent on NameNode

Install the agent on the NameNode Host as `root` (or sudo privileges). In HA Hadoop clusters, you must also install an agent on the Secondary NN.

4.1.1.2.1. Installation Set Up

Perform the following steps on the Hadoop NameNode host.

1. Log on to the host as `root`.
2. Create a temporary directory, such as `/tmp/xasecure`:

   ```
   mkdir /tmp/xasecure
   ```

3. Move the package into the temporary directory along with the MySQL Connector Jar.
4. Extract the contents:

   ```
   tar xvf $xasecureinstallation.tar
   ```

5. Go to the directory where you extracted the installation files:

   ```
   cd /tmp/xasecure/xasecure-$name-$build-version
   ```

6. Open the `install.properties` file for editing.
7. Change the following parameters for your environment:

Table 4.3. HDFS Agent Install Parameters

Parameter	Value	Description
POLICY_MGR_URL	*$url*	Specify the full URL to access the Policy Manager Web UI. For example, `http://pm-host:6080`.
MYSQL_CONNECTOR_JAR	*$path-to-mysql-connector*	Absolute path on the local host to the JDBC driver for mysql including filename.[a] For example, `/tmp/xasecure/`
REPOSITORY_NAME	*$Policy-Manager-Repo-Name*	Name of the HDFS Repository in the Policy Manager that this agent connects to after installation.
XAAUDIT.DB.HOSTNAME	*$XAsecure-db-host*	Specify the host name of the MySQL database.
XAAUDIT.DB.DATABASE_NAME	*$auditdb*	Specify the audit database name that matches the *audit_db_name* specified during the web application server installation.
XAAUDIT.DB.USER_NAME	*$auditdbuser*	Specify the audit database name that matches the *audit_db_user* specified during the web application server installation
XAAUDIT.DB.PASSWORD	*$auditdbupw*	Specify the audit database name that matches the *audit_db_password*

Parameter	Value	Description
		specified during the web application server installation.

[a] Download the JAR from here.

8. Save the `install.properties` file.

Note

If your environment is configured to use SSL, modify the properties following the instructions in Set Up SSL for HDFS Security Agent.

4.1.1.2.1.1. Example HDFS Agent Installation Properties

The following is an example of the Hadoop Agent `install.properties` file with the MySQL database co-located on the XASecure host:

```
#
# Location of Policy Manager URL
#
#
# Example:
# POLICY_MGR_URL=http://policymanager.xasecure.net:6080
#


POLICY_MGR_URL=http://xasecure-host:6080

#
# Location of mysql client library (please check the location of the jar file)
#
MYSQL_CONNECTOR_JAR=/usr/share/java/mysql-connector-java.jar

#
# This is the repository name created within policy manager
#
# Example:
# REPOSITORY_NAME=hadoopdev
#

REPOSITORY_NAME=sandbox


#
# AUDIT DB Configuration
#
#  This information should match with the one you specified during the
 PolicyManager Installation
#
# Example:
# XAAUDIT.DB.HOSTNAME=localhost
# XAAUDIT.DB.DATABASE_NAME=xasecure
# XAAUDIT.DB.USER_NAME=xalogger
# XAAUDIT.DB.PASSWORD=
#
#

XAAUDIT.DB.HOSTNAME=xasecure-host
XAAUDIT.DB.DATABASE_NAME=xaaudit
XAAUDIT.DB.USER_NAME=xaaudit
XAAUDIT.DB.PASSWORD=password

#
# SSL Client Certificate Information
#
# Example:
# SSL_KEYSTORE_FILE_PATH=/etc/xasecure/conf/xasecure-hadoop-client.jks
# SSL_KEYSTORE_PASSWORD=clientdb01
# SSL_TRUSTSTORE_FILE_PATH=/etc/xasecure/conf/xasecure-truststore.jks
# SSL_TRUSTSTORE_PASSWORD=changeit
#
#
# IF YOU DO NOT DEFINE SSL parameters, the installation script will
 automatically generate necessary key(s) and assign appropriate values
# ONLY If you want to assign manually, please uncomment the following
 variables and assign appropriate values.

# SSL_KEYSTORE_FILE_PATH=
# SSL_KEYSTORE_PASSWORD=
# SSL_TRUSTSTORE_FILE_PATH=
# SSL_TRUSTSTORE_PASSWORD=
```

4.1.1.2.2. Run the Agent Installation Script

After configuring the `install.properties` file, install the agent as `root`:

1. Log on to the Linux system as root and go to the directory where you extracted the installation files:

   ```
   cd /tmp/xasecure/xasecure-$name-$build-version
   ```

2. Run the agent installation script:

   ```
   # ./install.sh
   ```

4.1.1.2.3. Verify that Agent is Connected

Connected Agents display in the HDP Security Administration Web UI.

Note

Agents may not appear in the list until after the first event occurs in the repository.

To verify that the agent is connected to the server:

1. Log in to the interface using the admin account.

2. Click **Audit** > **Agent**.

4.1.1.2.4. Configure HDFS Agent to run in Ambari Environments

On Hadoop clusters managed by Ambari, change the default HDFS settings to allow the agent to enforce policies and report auditing events. Additionally, Ambari uses its own startup scripts to start and stop the NameNode server. Therefore, modify the Hadoop configuration script to include the Security Agent with a NameNode restart.

To configure HDFS properties and NameNode startup scripts:

1. Update HDFS properties from the Ambari Web Interface as follows:

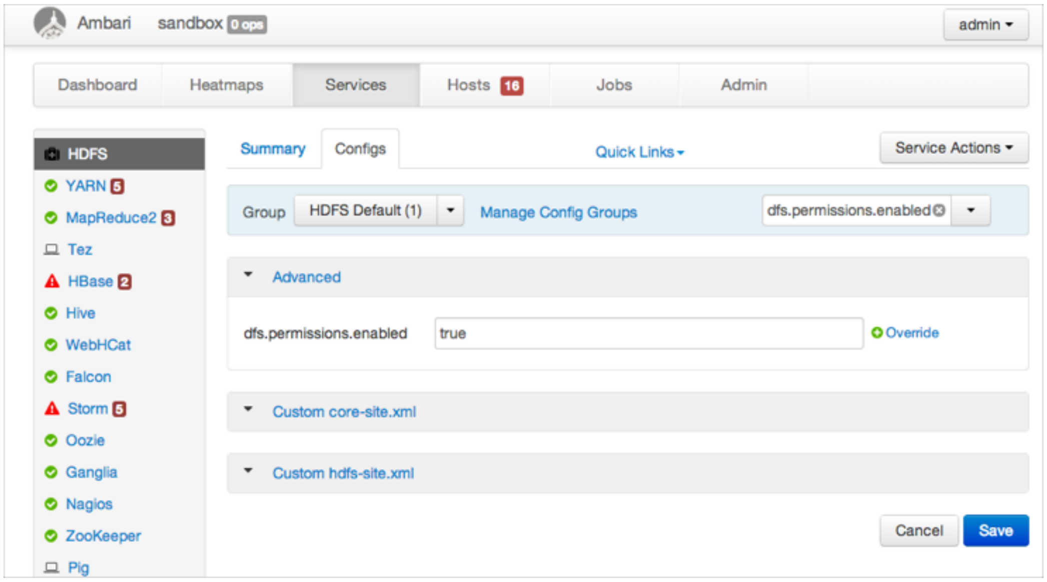
   a. On the Dashboard, click **HDFS**.

      The HDFS Service page displays.

   b. Go to the **Configs** tab.

   c. In Filter, type `dfs.permissions.enabled` and press enter.

      The results display. This property is located under Advanced.



Group: HDFS Default (1)

dfs.permissions.enabled

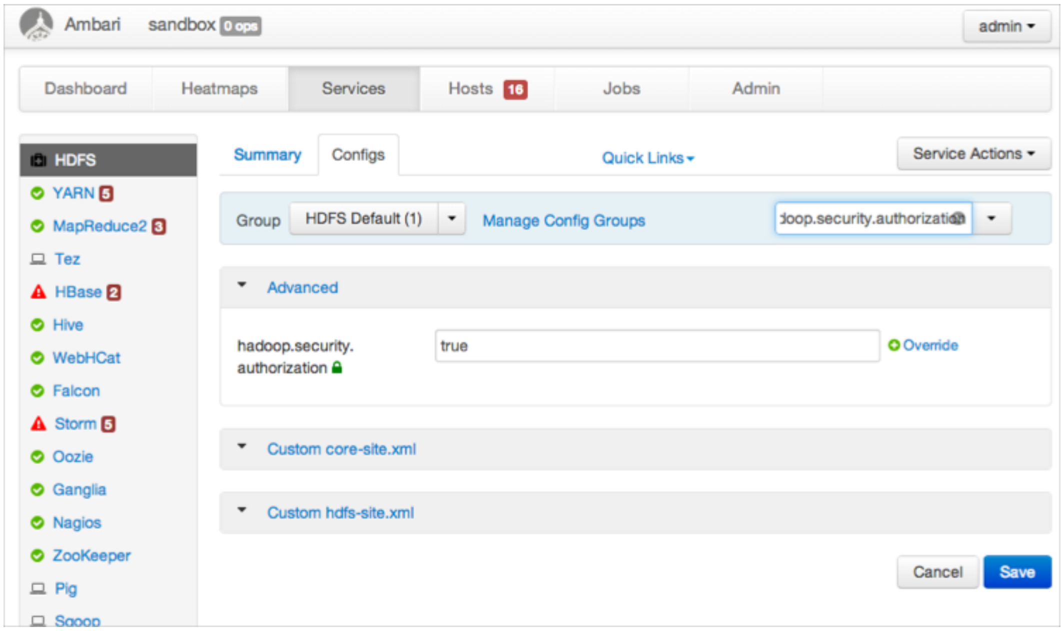
Advanced

dfs.permissions.enabled: true

   d. Expand Advanced, then change `dfs.permissions.enabled` to `true`.

   e. In Filter, type `hadoop.security.authorization` and press enter.

      Under the already expanded Advanced option, the parameter displays.



hadoop.security.authorization: true

f. Change `hadoop.security.authorization` to `true`.

g. Scroll to the bottom of the page and click **Save**.

At the top of the page, a message displays indicating the services that need to be restarted.

Warning

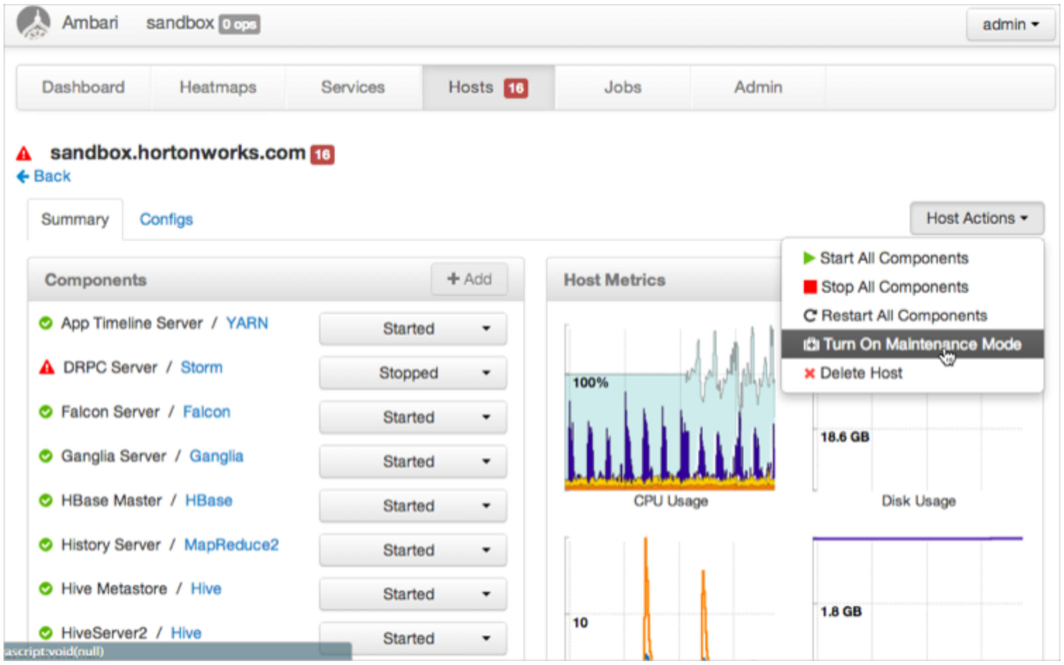
Do not restart the services until after you perform the next step.

2. Change the Hadoop configuration script to start the Security Agent with the NameNode service:

a. In the Ambari Administrator Portal, click **HDFS** and then **NameNode**.

The NameNode Hosts page displays.

b. Click **Host Actions** and choose **Turn on Maintenance Mode**.



Wait for the cluster to enter maintenance mode.

c. SSH to the NameNode as the `root` user.

d. Open the `hadoop-config.sh` script for editing and go to the end of the file. For example:

```
vi /usr/lib/hadoop/libexec/hadoop-config.sh
```

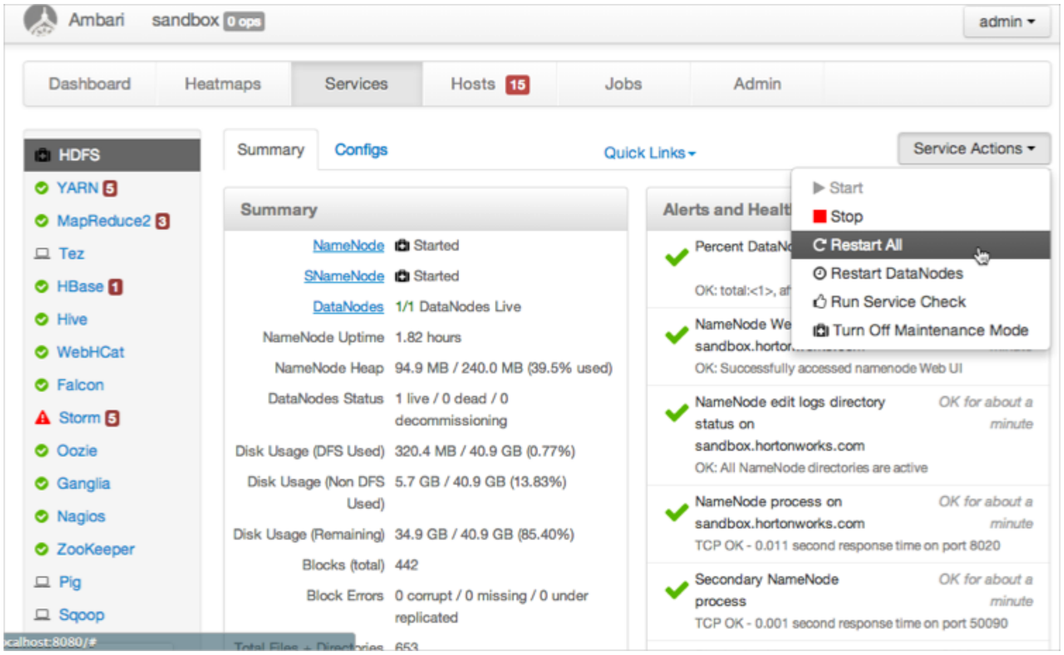
e. At the end of the file paste the following statement:

```
if [ -f  ${HADOOP_CONF_DIR}/xasecure-hadoop-env.sh ]
then
        .  ${HADOOP_CONF_DIR}/xasecure-hadoop-env.sh
fi
```

This adds the Security Agent for Hadoop to the start script for Hadoop.

f. Save the changes.

3. In the Ambari Administrative Portal, click **Services** > **Service Actions** > **Restart All**.



Summary

NameNode	Started
SNameNode	Started
DataNodes	1/1 DataNodes Live
NameNode Uptime	1.82 hours
NameNode Heap	94.9 MB / 240.0 MB (39.5% used)
DataNodes Status	1 live / 0 dead / 0 decommissioning
Disk Usage (DFS Used)	320.4 MB / 40.9 GB (0.77%)
Disk Usage (Non DFS Used)	5.7 GB / 40.9 GB (13.83%)
Disk Usage (Remaining)	34.9 GB / 40.9 GB (85.40%)
Blocks (total)	442
Block Errors	0 corrupt / 0 missing / 0 under replicated
Total Files + Directories	653

Percent DataN
OK: total:<1>, af

NameNode We
sandbox.horto
OK: Successfully accessed namenode Web UI

NameNode edit logs directory status on sandbox.hortonworks.com — OK for about a minute
OK: All NameNode directories are active

NameNode process on sandbox.hortonworks.com — OK for about a minute
TCP OK - 0.011 second response time on port 8020

Secondary NameNode process — OK for about a minute
TCP OK - 0.001 second response time on port 50090

Wait for the services to completely restart.

4. Click **Services** > **Service Actions** > **Turn off Maintenance Mode**.

   It may take several minutes for the process to complete. After confirming all the services restart as expected, perform a few simple HDFS comments such as browsing the file system from Hue.

4.1.1.2.5. Restart NameNode In non-Ambari Environments

The HDFS Agent is integrated with the NameNode Service. Before your changes can take effect you must restart the NameNode service.

1. On the NameNode host machine, execute the following command:

   ```
   su -l hdfs -c "/usr/lib/hadoop/sbin/hadoop-daemon.sh stop namenode"
   ```

   Ensure that the NameNode Service stops completely.

2. On the NameNode host machine, execute the following command:

   ```
   su -l hdfs -c "/usr/lib/hadoop/sbin/hadoop-daemon.sh start namenode"
   ```

   Ensure that the NameNode Service starts correctly.

4.1.1.3. Test HDFS Configuration

After completing the setup of the HDFS Repository and agent, perform a few simple tests to ensure that the agent is auditing and reporting events to the HDP Security Administration Web UI. By default, the repository allows all access and has auditing enabled.

1. Log into the Hadoop cluster.

2. Type the following command to display a list of items at the root folder of HDFS:

```
hadoop fs -ls /
Found 6 items
drwxrwxrwx   - yarn   hadoop          0 2014-04-21 07:21 /app-logs
drwxr-xr-x   - hdfs   hdfs            0 2014-04-21 07:23 /apps
drwxr-xr-x   - mapred hdfs            0 2014-04-21 07:16 /mapred
drwxr-xr-x   - hdfs   hdfs            0 2014-04-21 07:16 /mr-history
drwxrwxrwx   - hdfs   hdfs            0 2014-06-17 15:05 /tmp
drwxr-xr-x   - hdfs   hdfs            0 2014-04-22 07:21 /user
```

3. Sign in to the Web UI and click **Audit**.

   The Big Data page displays a list of events for the configured Repositories.

4. Click **Search** > **Repository Type** > **HDFS**.

   The list filters as you make selections.

4.2. Add Hive Repositories

HDP Security Administration tools support access control and auditing for Hive repositories in Hadoop clusters.

4.2.1. Create a Hive Repository

Before installing the agent on the HiveServer2 host set up a repository in the Policy Manager.

Important

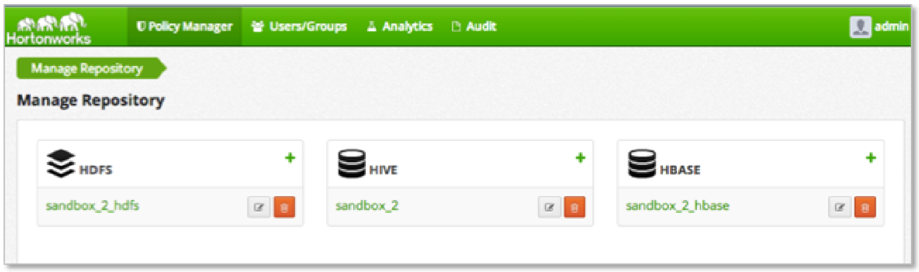
For Hive connection information, see HiveServer2 Clients, JDBC.

To create a Hive Repository:

1. Sign in to the HDP Security Administrator Web UI as an administrator.

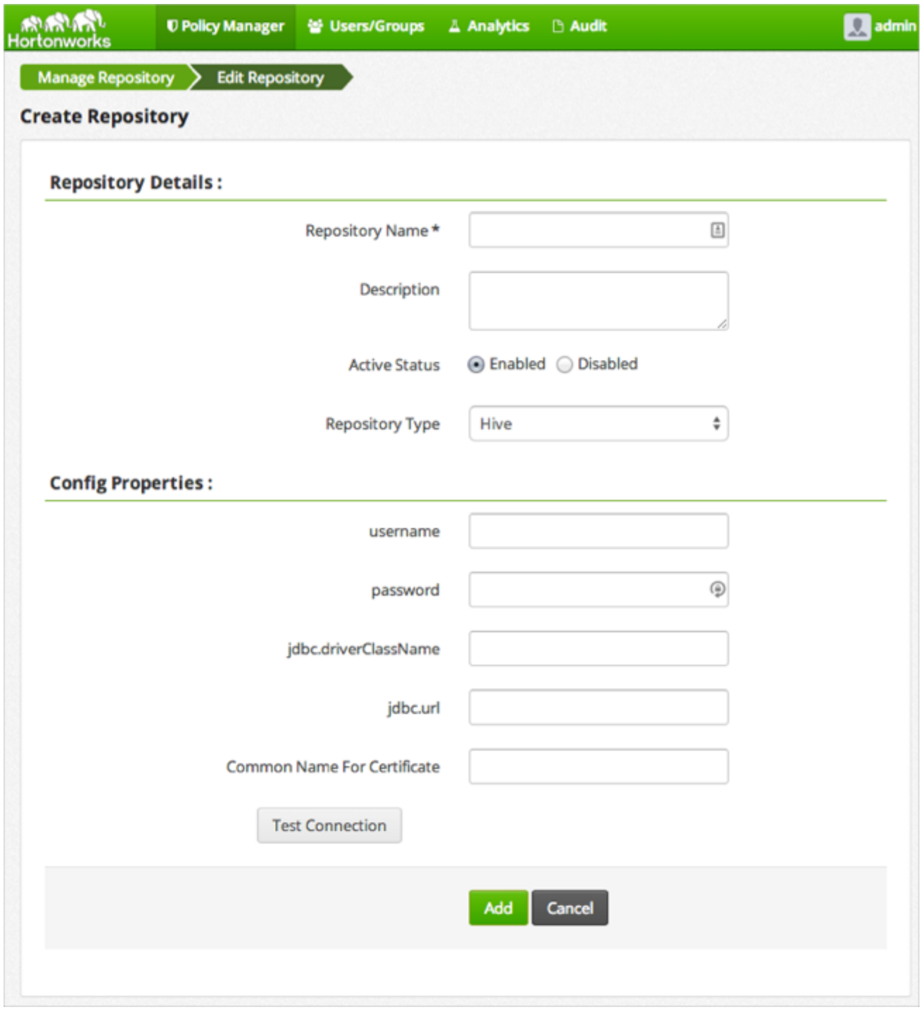
2. Click **Policy Manager**.

   The Manage Repository page displays.



3. Next to Hive, click the green plus symbol.

   The Create Repository page displays.



Create Repository

Repository Details :

Repository Name *

Description

Active Status ● Enabled ○ Disabled

Repository Type Hive

Config Properties :

username

password

jdbc.driverClassName

jdbc.url

Common Name For Certificate

4. Complete the required settings with the following information:

Table 4.4. Hive Repository Details

Label	Value	Description
Repository Name	*$name*	Specify a unique name for the repository, you will need to specify the repository name in the agent installation properties. For example, `clustername_hive`.
Description	*$description-of-repo*	Enter a description up to 150 characters.
Active Status	`Enabled` or `Disabled`	Enable or disable policy enforcement for the repository.
Repository type	HDFS, `Hive`, or `HBase`	Select the type of repository, Hive.
User name	*$user*	Specify a user name on the remote system with permission to establish the connection with the hive, for example `hive`.
Password	*$password*	Specify the password of the user account for connection.
jdbc.driverClassName	*$classname*	Specify the full classname of the driver used for Hive connections.

Label	Value	Description
		The default HiveServer2 classname is `org.apache.hive.jdbc.HiveDriver`.
jdbc.url	*`$jdbc:hive2://hiveserver-host:port/db`*	Specify the complete connection URL, including port (default port is 10000) and database name. For example on sandbox, `jdbc:hive2://sandbox:10000/`.

5. Click **Test Connection**.

   If the server can establish a connection with HiveServer using the information you provided a success message displays.

6. After the connection is successful, click **Save**.

4.2.2. Install the Hive Agent on the HiveServer2 Host

After creating the Hive Repository in the Policy Manager, install the agent on the HiveServer2 host.

> **Note**
>
> If you are using Beeswax on Hue to run Hive queries, you must also install the Hive agent on the Hue server host.

4.2.2.1. Installation Set Up

Perform the following steps on the HiveServer2 host.

1. Log on to the host as `root`.

2. Create a temporary directory, such as `/tmp/xasecure`:

   ```
   mkdir /tmp/xasecure
   ```

3. Move the package into the temporary directory along with the MySQL Connector Jar.

4. Extract the contents:

   ```
   tar xvf $xasecureinstallation.tar
   ```

5. Go to the directory where you extracted the installation files:

   ```
   cd /tmp/xasecure/xasecure-$name-$build-version
   ```

6. Open the `install.properties` file for editing.

7. Change the following parameters for your environment:

Table 4.5. Hive Agent Install Parameters

Parameter	Value	Description
`POLICY_MGR_URL`	*`$url`*	Specify the full URL to access the Policy Manager Web UI. For example, `http://pm-host:6080`.

Parameter	Value	Description
MYSQL_CONNECTOR_JAR	*$path-to-mysql-connector*	Absolute path on the local host to the JDBC driver for mysql including filename.[a] For example, `/tmp/xasecure/`
REPOSITORY_NAME	*$Policy-Manager-Repo-Name*	Name of the HDFS Repository in the Policy Manager that this agent connects to after installation.
XAAUDIT.DB.HOSTNAME	*$XAsecure-db-host*	Specify the host name of the MySQL database.
XAAUDIT.DB.DATABASE_NAME	*$auditdb*	Specify the audit database name that matches the *audit_db_name* specified during installation.
XAAUDIT.DB.USER_NAME	*$auditdbuser*	Specify the audit database name that matches the *audit_db_user* specified during installation
XAAUDIT.DB.PASSWORD	*$auditdbupw*	Specify the audit database name that matches the *audit_db_password* specified during installation

[a] Download the JAR from here.

8. Save the `install.properties` file.

Note

If your environment is configured to use SSL, modify the properties following the instructions in Set Up SSL for Hive Security Agent.

4.2.2.2. Run the Agent Installation Script

After configuring the `install.properties` file, install the agent as `root`:

1. Log on to the Linux system as root and go to the directory where you extracted the installation files:

```
cd /tmp/xasecure/xasecure-$name-$build-version
```

2. Run the agent installation script:

```
# ./install.sh
```

4.2.2.3. Restart the Hive Service

After installing the agent in an environment that does NOT have Ambari, manually restart the Hive services as follows:

1. Stop Hive. Execute this command on the Hive Metastore and Hive Server2 host machine.

```
ps aux | awk '{print $1,$2}' | grep hive | awk '{print $2}' | xargs kill >/
dev/null 2>&1
```

2. Start Hive Metastore. On the Hive Metastore host machine, execute the following command:

```
su - hive -c "env HADOOP_HOME=/usr JAVA_HOME=/usr/jdk64/jdk1.6.0_31 /tmp/
startMetastore.sh /var/log/hive/hive.out /var/log/hive/hive.log /var/run/
hive/hive.pid /etc/hive/conf.server"
```

where, *$HIVE_LOG_DIR* is the directory where Hive server logs are stored. For example, /var/logs/hive.

3. Start HiveServer2. On the Hive Server2 host machine, execute the following command:

```
su - hive -c "env JAVA_HOME=/usr/jdk64/jdk1.6.0_31 /tmp/startHiveserver2.
sh /var/log/hive/hive-server2.out /var/log/hive/hive-server2.log /var/run/
hive/hive-server.pid /etc/hive/conf.server"
```

   where *$HIVE_LOG_DIR* is the directory where Hive server logs are stored. For example, /var/logs/hive.

4.2.3. Configure Hive in Ambari Environments

Follow the configuration steps in environments where Hive is managed by the Ambari Server:

- Modify the Ambari Hive Startup Script
- Configure Hive

4.2.3.1. Modify the Ambari Hive Startup Script

Remove the HiveServer configuration string from the Ambari Hive startup script.

Note

Ambari starts and stops the HiveServer2 using a built in script. In order to start and stop HiveServer2 with the integrated Security Agent, you must comment out the HiveServer configuration string.

1. Log into the Ambari Server Linux host using the Ambari account.

2. Open the Ambari Hive startup script for editing:

```
cd /var/lib/ambari-server/resources/stacks/HDP/2.0.6/services/HIVE/package/
templates
vi startHiveserver2.sh.j2
```

3. Comment out the following line by prepending a # at the beginning of the line as follows:

```
# HIVE_SERVER2_OPTS="${HIVE_SERVER2_OPTS} –hiveconf hive.
security.authenticator.manager=org.apache.hadoop.hive.ql.security.
SessionStateUserAuthenticator -hiveconf hive.security.authorization.
manager=org.apache.hadoop.hive.ql.security.authorization.plugin.sqlstd.
SQLStdHiveAuthorizerFactory”
```

   And add the following line to replace it:

```
HIVE_SERVER2_OPTS="${HIVE_SERVER2_OPTS} –hiveconf hive.security.
authenticator.manager=org.apache.hadoop.hive.ql.security.
SessionStateUserAuthenticator”
```

4. Restart the Ambari Server from the command line as follows:

```
su -l ambari -c "/etc/init.d/ambari-server stop"
su -l ambari -c "/etc/init.d/ambari-server start"
```

5. On each node in the cluster, restart the Ambari Agents:

```
su -l ambari -c "/etc/init.d/ambari-agent stop"
su -l ambari -c "/etc/init.d/ambari-agent start"
```

After the Ambari Server and Agents finish rebooting, update the Hive Configuration with the required settings.

4.2.3.2. Configure Hive

After changing the Ambari Hive startup script and restarting the Ambari Server from the command line, perform the following steps to configure Hive server for agent integration.

1. Log into the Ambari Web UI, and click **Hive** > **Config**.

   Note

   To find a property, type the name in the Filter field and press enter; if the parameter exists, it is returned under the common or advanced list. Click the arrow key to expand the lists to see the settings.

2. Update the following properties as follows:

   - **Property name**: `hive.security.authorization.manager`

     New Value: `com.xasecure.authorization.hive.authorizer.XaSecureAuthorizer`

   - **Property name**: `hive.security.authorization.enabled`

     New Value: `true`

3. Filter for the `hive.exec.pre.hooks` property.

   Add the HDP Security hook after the existing value by inserting a comma followed by `com.xasecure.authorization.hive.hooks.XaSecureHivePreExecuteRunHook`.

   For example, if the existing value is `org.apache.hadoop.hive.ql.hooks.ATSHook` the new value with the HDP Security hook is:

   ```
   org.apache.hadoop.hive.ql.hooks.ATSHook,com.xasecure.authorization.hive.
   hooks.XaSecureHivePreExecuteRunHook
   ```

4. Search for the *`hive.exec.post.hooks`* property.

   Add the HDP Security hook after the existing value by inserting a comma followed by `com.xasecure.authorization.hive.hooks.XaSecureHivePostExecuteRunHook`.

   For example if the existing value is `org.apache.hadoop.hive.ql.hooks.ATSHook` the new value with the HDP Security hook is:

```
org.apache.hadoop.hive.ql.hooks.ATSHook,com.xasecure.authorization.hive.
hooks.XaSecureHivePostExecuteRunHook
```

5. Expand **Custom hive-site.xml**, and add the following properties:

Table 4.6. Custom hive-site.xml Properties

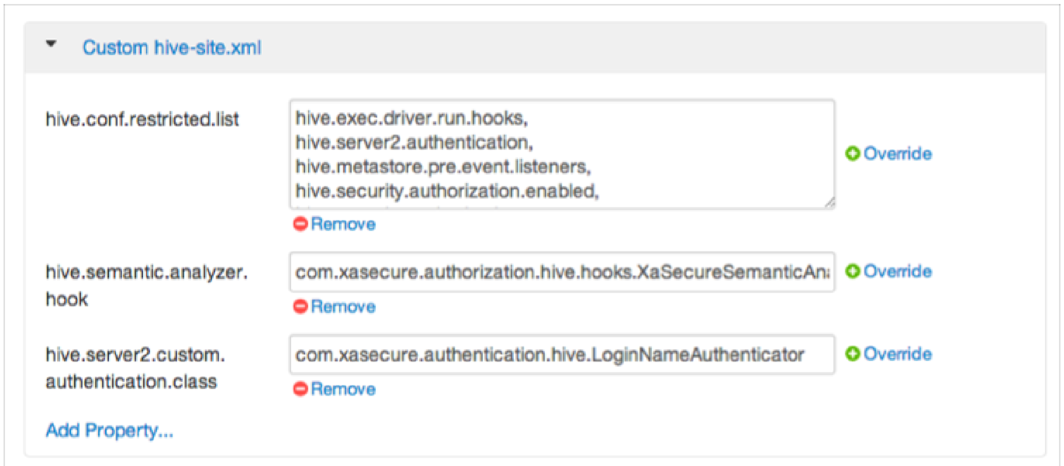
Key	Value
hive.semantic.analyzer.hook	com.xasecure.authorization.hive.hooks.XaSecureSemanticAnalyzerHook
hive.server2.custom.authentication.class	com.xasecure.authentication.hive.LoginNameAuthenticator
hive.conf.restricted.list	hive.exec.driver.run.hooks, hive.server2.authentication, hive.metastore.pre.event.listeners, hive.security.authorization.enabled,hive.security.authorization.manager, hive.semantic.analyzer.hook, hive.exec.post.hooks

Note

For each property, click **Add Property**, enter Key and Value shown in the table above, then click **Add**.

6. After all the properties have been updated and added, scroll to the bottom and click **Save**.

The settings display under **Custom hive-site.xml**.



Custom hive-site.xml

hive.conf.restricted.list: hive.exec.driver.run.hooks, hive.server2.authentication, hive.metastore.pre.event.listeners, hive.security.authorization.enabled,

hive.semantic.analyzer.hook: com.xasecure.authorization.hive.hooks.XaSecureSemanticAn

hive.server2.custom.authentication.class: com.xasecure.authentication.hive.LoginNameAuthenticator

When properties change, the affected services must be restarted. A Restart option displays.

7. Click **Restart** > **Restart all**.

4.2.4. Verify that Agent is Connected

Connected Agents display in the HDP Security Administration Web UI.

Note

Agents may not appear in the list until after the first event occurs in the repository.

To verify that the agent is connected to the server:

1. Log in to the interface using the admin account.
2. Click **Audit** > **Agent**.

4.3. Add HBase Repositories

HBase agents integrate with the HBase Master and HBase Region Servers.

Note

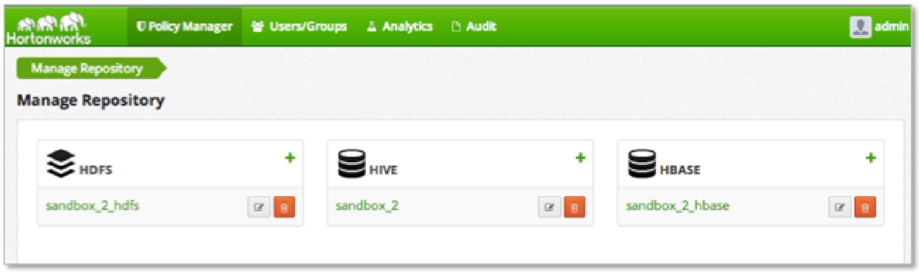
When adding an HBase Repository you must install the Security Agent for HBase on the HBase Master and each of the HBase Region Servers in your cluster and ensure that the configuration settings are the same on each Region Server.

4.3.1. Create a HBase Repository

Before installing the agent on the HBase Regional Servers, create an HBase Repository as follows:

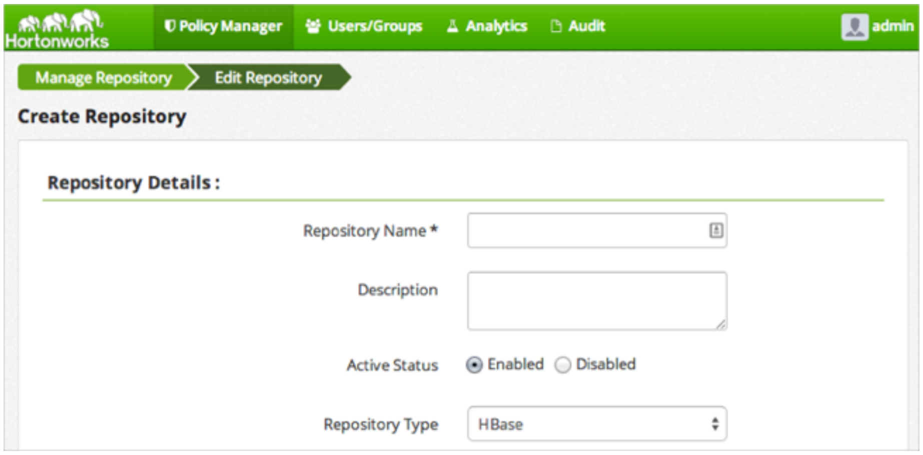
1. Sign in to the HDP Security Administration Web UI.
2. Click **Policy Manager**.

   The Manage Repository page displays.



3. Next to HBase, click the + (plus symbol).

   The Create Repository page displays.



Create Repository

Repository Details :

Repository Name *

Description

Active Status ● Enabled ○ Disabled

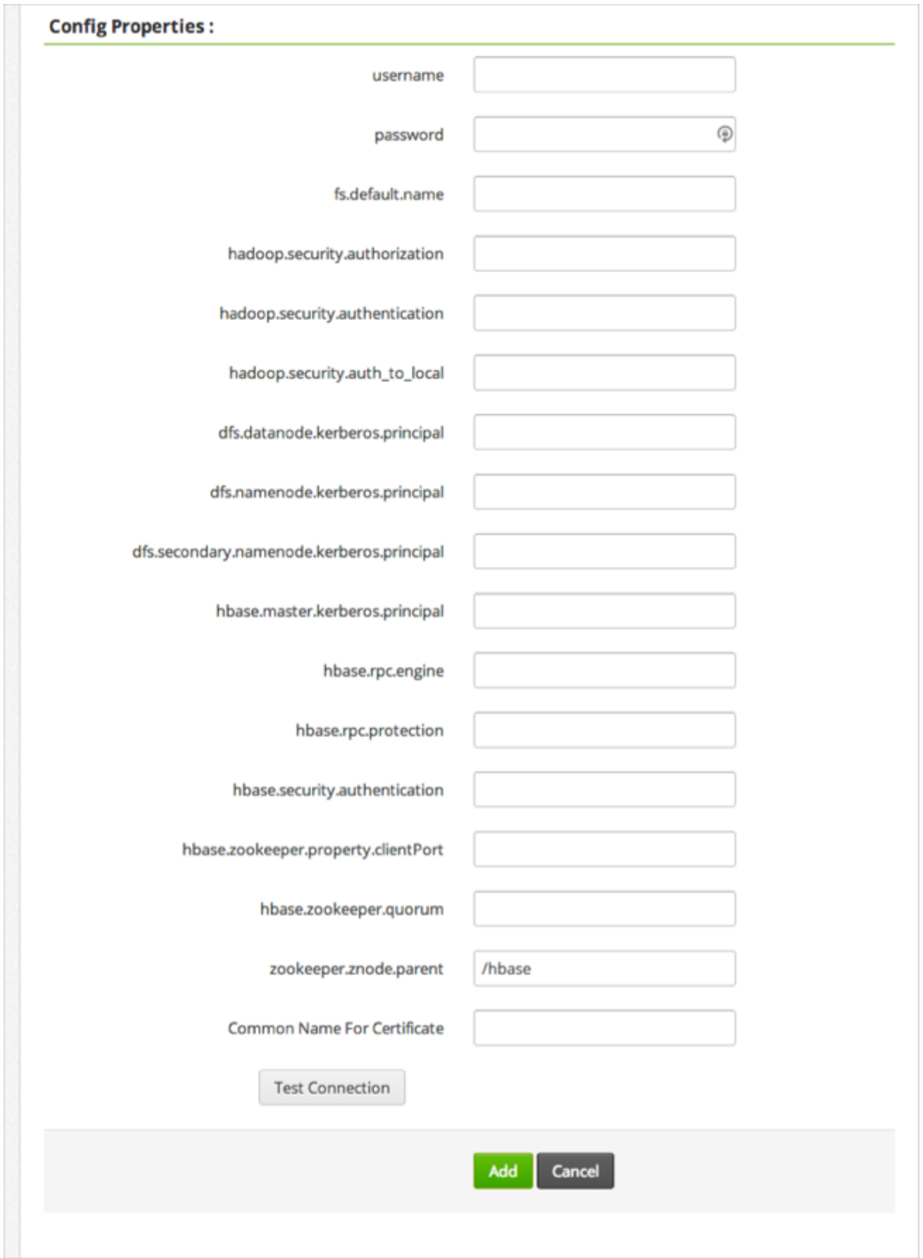
Repository Type HBase

4. Complete the Repository Details with the following information:

Table 4.7. HBase Repository Details

Label	Value	Description
Repository Name	*$name*	Specify a unique name for the repository, you will need to specify the same repository name in the agent installation properties. For example, `clustername_hbase`.
Description	*$description-of-repo*	Enter a description up to 150 characters.
Active Status	`Enabled` or `Disabled`	Enable or disable policy enforcement for the repository.
Repository type	`HDFS`, `Hive`, or `HBase`	Select the type of repository, HBase.
User name	*$user*	Specify a user name on the remote system with permission to establish the connection, for example `hbase`.
Password	*$password*	Specify the password of the user account for connection.

5. Complete the HBase Configuration:



Config Properties :

username	
password	
fs.default.name	
hadoop.security.authorization	
hadoop.security.authentication	
hadoop.security.auth_to_local	
dfs.datanode.kerberos.principal	
dfs.namenode.kerberos.principal	
dfs.secondary.namenode.kerberos.principal	
hbase.master.kerberos.principal	
hbase.rpc.engine	
hbase.rpc.protection	
hbase.security.authentication	
hbase.zookeeper.property.clientPort	
hbase.zookeeper.quorum	
zookeeper.znode.parent	/hbase
Common Name For Certificate	

The settings must match the values specified in the `core-site.xml` and `hbase-site.xml` file as follows:

Table 4.8. HBase Configuration

Label	Value	File
fs.default.name	*`$hdfs-url`*	`core-site.xml` For example, `hdfs://sandbox.hortonworks.com:8020`
hadoop.security.authorization	`true`	`core-site.xml` If this field is false, then change to true

Label	Value	File
		in core-site before you continue.
hadoop.security.authentication	`simple` or `kerberos`	`core-site.xml`
hadoop.security.auth_to_local	*`$usermapping`*	`core-site.xml`For example: `RULE:[2:$1@$0]([rn]m@.*)s/.*/yarn/ RULE:[2:$1@$0](jhs@.*)s/.*/mapred/ RULE:[2:$1@$0]([nd]n@.*)s/.*/hdfs/ RULE:[2:$1@$0](hm@.*)s/.*/hbase/ RULE:[2:$1@$0](rs@.*)s/.*/hbase/ DEFAULT`
dfs.datanode.kerberos.principal	*`$dn-principal`*	Specify the Kerberos DataNode principal name.
dfs.namenode.kerberos.principal	*`$nn-principal`*	Specify the Kerberos NameNode principal name.
dfs.secondary.namenode.kerberos.principal	*`$secondary-nn-principal`*	Specify the Kerberos Secondary NN principal name.
hbase.master.kerberos.principal	*`$hbase-principal`*	Specify the Kerberos principal for the HBase Master.
hbase.rpc.engine	org.apache.hadoop.hbase.ipc.SecureRpcEngine	`hbase-site.xml`
hbase.rpc.protection	`PRIVACY`	`hbase-site.xml`
hbase.security.authentication	`simple`	`hbase-site.xml`
hbase.zoopkeeper.property.clientPort	`2181`	`hbase-site.xml`
hbase.zookeeper.quorom		`hbase-site.xml`
zookeeper.znode.parent	`/hbase`	`hbase-site.xml`
Common Name For Certificate	*`$cert-name`*	Specify the name of the certificate.

Note

The blank fields are optional.

6. Click **Test Connection**.

   If the server can connect to HBase, the connection successful message displays.

   HDP Security Administration server connects to HBase and lists the tables. Hortonworks recommends creating the repository and installing the agent after HBase contains data. If HBase connection fails (and tables exist), go to the troubleshooting appendix.

7. After making a successful connection, click **Save**.

The repository is created with an open access Policy, that is auditing is enabled and all users are allowed to access the resources. Complete the installation of the agent and do a few simple access test before configuring policies to ensure that the solution is working properly.

4.3.2. Installation Set Up

Use same installation properties file to install the Security Agent for HBase. Install the agent on *all* of the following HBase hosts:

- HBase Master host
- All HBase Region Server hosts

1. Log on to the host as `root`.
2. Create a temporary directory, such as `/tmp/xasecure`:

   ```
   mkdir /tmp/xasecure
   ```

3. Move the package into the temporary directory along with the MySQL Connector Jar.
4. Extract the contents:

   ```
   tar xvf $xasecureinstallation.tar
   ```

5. Go to the directory where you extracted the installation files:

   ```
   cd /tmp/xasecure/xasecure-$name-$build-version
   ```

6. Open the `install.properties` file for editing.
7. Change the following parameters for your environment:

Table 4.9. Hive Agent Install Parameters

Parameter	Value	Description
POLICY_MGR_URL	*$url*	Specify the full URL to access the Policy Manager Web UI. For example, `http://pm-host:6080`.
MYSQL_CONNECTOR_JAR	*$path-to-mysql-connector*	Absolute path on the local host to the JDBC driver for mysql including filename.[a] For example, `/tmp/xasecure/`
REPOSITORY_NAME	*$Policy-Manager-Repo-Name*	Name of the HDFS Repository in the Policy Manager that this agent connects to after installation.
XAAUDIT.DB.HOSTNAME	*$XAsecure-db-host*	Specify the host name of the MySQL database.
XAAUDIT.DB.DATABASE_NAME	*$auditdb*	Specify the audit database name that matches the *audit_db_name* specified during installation.
XAAUDIT.DB.USER_NAME	*$auditdbuser*	Specify the audit database name that matches the *audit_db_user* specified during installation.

Parameter	Value	Description
XAAUDIT.DB.PASSWORD	*$auditdbupw*	Specify the audit database name that matches the *audit_db_password* specified during installation.

[a] Download the JAR from here.

8. Save the `install.properties` file.

Note

If your environment is configured to use SSL, modify the properties following the instructions in Set Up SSL for HBase Security Agents.

The following is an example of the HBase `install.properties`:

```
#
# Location of Policy Manager URL
#
#
# Example:
# POLICY_MGR_URL=http://policymanager.xasecure.net:6080
#

POLICY_MGR_URL=http://policymgr:6080

#
# Location of mysql client library (please check the location of the jar file)
#
MYSQL_CONNECTOR_JAR=/usr/share/java/mysql-connector-java.jar

#
# This is the repository name created within policy manager
#
# Example:
# REPOSITORY_NAME=hbasedev
#

REPOSITORY_NAME=sandbox_2_hbase

#
# AUDIT DB Configuration
#
#  This information should match with the one you specified during the
 PolicyManager Installation
#
# Example:
# XAAUDIT.DB.HOSTNAME=localhost
# XAAUDIT.DB.DATABASE_NAME=xasecure
# XAAUDIT.DB.USER_NAME=xalogger
# XAAUDIT.DB.PASSWORD=
#
#

XAAUDIT.DB.HOSTNAME=xasecure
XAAUDIT.DB.DATABASE_NAME=xasecure
XAAUDIT.DB.USER_NAME=xasecure
XAAUDIT.DB.PASSWORD=hadoop
```

```
#
# SSL Client Certificate Information
#
# Example:
# SSL_KEYSTORE_FILE_PATH=/etc/xasecure/conf/xasecure-hadoop-client.jks
# SSL_KEYSTORE_PASSWORD=clientdb01
# SSL_TRUSTSTORE_FILE_PATH=/etc/xasecure/conf/xasecure-truststore.jks
# SSL_TRUSTSTORE_PASSWORD=changeit

#
# IF YOU DO NOT DEFINE SSL parameters, the installation script will
 automatically generate necessary key(s) and assign appropriate values
# ONLY If you want to assign manually, please uncomment the following
 variables and assign appropriate values.
```

4.3.3. Run the Agent Installation Script

After configuring the `install.properties` file, install the agent as `root`:

1. Log on to the Linux system as root and go to the directory where you extracted the installation files:

   ```
   cd /tmp/xasecure/xasecure-$name-$build-version
   ```

2. Run the agent installation script:

   ```
   # ./install.sh
   ```

4.3.4. Restart the HBase Service (Manual HDP Installation)

Changes to the properties require a restart of the HBase services.

To restart HBase:

1. Execute this command on the HBase Master host machine:

   ```
   su -l hbase -c "/usr/lib/hbase/bin/hbase-daemon.sh --config /etc/hbase/conf
    stop master; sleep 25"
   ```

2. Execute this command on all RegionServers:

   ```
   su -l hbase -c "/usr/lib/hbase/bin/hbase-daemon.sh --config /etc/hbase/conf
    stop regionserver"
   ```

3. Execute this command on the HBase Master host machine:

   ```
   su -l hbase -c "/usr/lib/hbase/bin/hbase-daemon.sh --config /etc/hbase/conf
    start master; sleep 25"
   ```

4. Execute this command on all RegionServers:

   ```
   su -l hbase -c "/usr/lib/hbase/bin/hbase-daemon.sh --config /etc/hbase/conf
    start regionserver"
   ```

4.3.5. Configure HBase Properties (Ambari Deployment)

HDP Security Administration requires that the following properties are set in the `hbase-site.xml`. Configure these properties and restart Hbase before creating a repository in the Policy Manager.

Table 4.10. Custom hbase-site.xml Parameters

Key	Value
hbase.security.authorization	true
hbase.coprocessor.master.classes	com.xasecure.authorization.hbase.XaSecureAuthorizationCoprocessor
hbase.coprocessor.region.classes	org.apache.hadoop.hbase.security.token.TokenProvider,org.apache.hadoop.hba
hbase.rpc.engine	org.apache.hadoop.hbase.ipc.SecureRpcEngine
hbase.rpc.protection	PRIVACY

4.3.5.1. Update and Add Properties with Ambari

Use these instructions to update the Hbase properties in the Ambari UI.

1. Log into the Ambari Web UI, and click **HBase** > **Config**.

   ### Note

   To find a parameter, type the parameter name in the Filter field and press enter; if the parameter exists, it is returned under list. Click the arrow key to expand the lists and see the parameter settings.

2. Update the following properties from the Ambari Default Value to the HDP Security required values:

   ### Table 4.11. HBase Parameter Values

HBase Property	Ambari Default Value	HDP Security Required Value
`hbase.security.authorization`	`false`	`true`

3. Expand **Custom hbase-site.xml**, and add the following properties:

   ### Table 4.12. Custom hbase-site.xml Properties

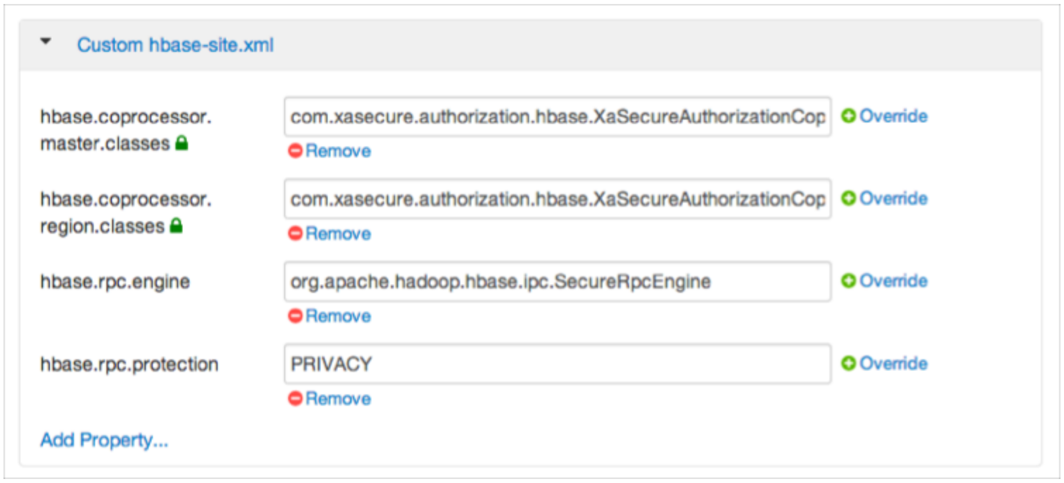
Key	Value
hbase.coprocessor.master.classes	com.xasecure.authorization.hbase.XaSecureAuthorizationCoprocessor
hbase.coprocessor.region.classes	org.apache.hadoop.hbase.security.token.TokenProvider,org.apache.hadoop.h
hbase.rpc.protection	PRIVACY

   ### Note

   For each property, click **Add Property**, enter Key and Value shown in the table above, then click **Add**.

4. After all the properties have been updated or added, click **Save**.

   The **Custom hbase-site.xml** properties display.



Custom hbase-site.xml

hbase.coprocessor.master.classes: com.xasecure.authorization.hbase.XaSecureAuthorizationCop

hbase.coprocessor.region.classes: com.xasecure.authorization.hbase.XaSecureAuthorizationCop

hbase.rpc.engine: org.apache.hadoop.hbase.ipc.SecureRpcEngine

hbase.rpc.protection: PRIVACY

When properties change, the affected services must be restarted. A Restart option appears.

5. Click **Restart** > **Restart all**.

4.3.6. Verify that Agent is Connected

Connected Agents display in the HDP Security Administration Web UI.

> **Note**
>
> Agents may not appear in the list until after the first event occurs in the repository.

To verify that the agent is connected to the server:

1. Log in to the interface using the admin account.

2. Click **Audit** > **Agent**.

4.3.7. Test HBase Access and Auditing

After the repository is set up and you have verified that the agent is connected to the server, perform a few basic HBase test as outlined below:

1. Open a browser and go to `http://hue-host:8888`.

2. Click on the **Hue Shell** icon in the navigation pane.

3. Click **HBase Shell**.

   The prompt displays.

   ```
   hbase(main):001:0>
   ```

4. At the prompt type `list`.

   ```
   hbase(main):001:0> list
   ```

```
list
TABLE
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/lib/hadoop/lib/slf4j-log4j12-1.7.5.
jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/lib/zookeeper/lib/slf4j-log4j12-1.6.
1.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an
 explanation.
ambarismoketest
test
2 row(s) in 4.9490 seconds

=> ["ambarismoketest", "test"]
```

The XASecure HBase agent reports the activity to the server.

> **Note**
>
> If the HBase command fails with the following Zookeeper error, restart HBase with the root user account from the command line and retest.
>
> ```
> ERROR: Can't get master address from ZooKeeper; znode data ==
>  null
> ```

5. Sign in to the Web UI and click **Audit**.

   The Big Data page displays a list of events for the configured Repositories.

6. Click **Search** > **Repository Type** > **HBase**.

   The list filters as you make selections.

4.4. Change Repository Configuration

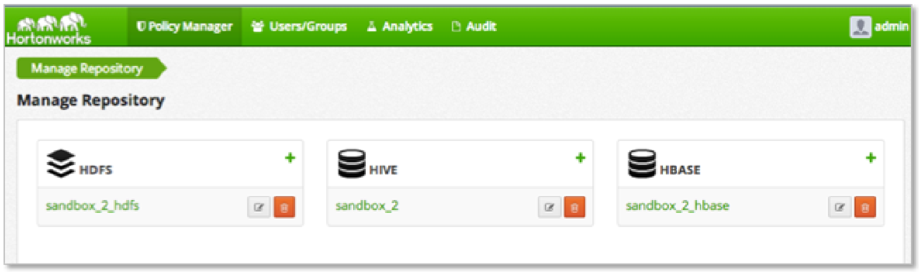
You can edit repository details, including the configuration properties using the edit icon next to a repository name.

To change the settings:

1. Sign in to the HDP Security Administration Web UI.

2. Click **Policy Manager**.

   The Repository Details page displays.



3. Click the **Edit** icon next to the Repository name.

   The Repository Details page displays.

4. Change the settings and click **Save**.

Note

Changing the repository name does not break the link between the agent and the repository. The name is updated on the corresponding Audit and Reporting pages.

4.5. Remove a Repository Configuration

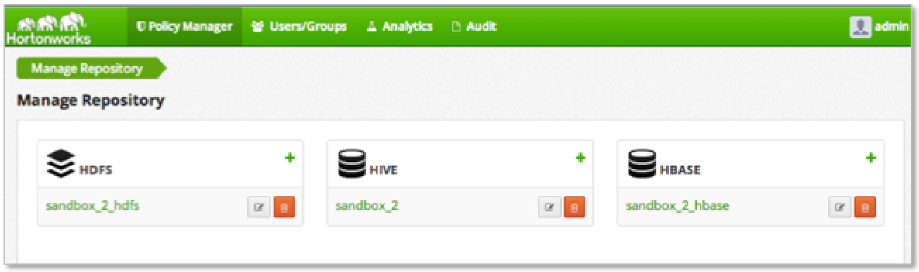
Deleting a repository only hides it from the Administration Web UI and does not completely remove it from the system.

To remove a repository:

1. Sign in to the HDP Security Administration Web UI.

2. Click **Policy Manager**.

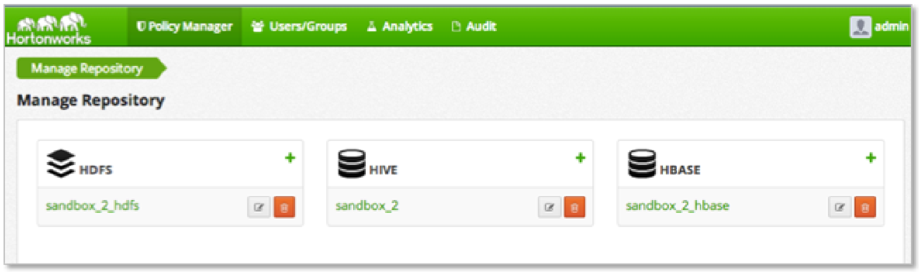
   The Repository Details page displays.



3. Next to the repository name, click the Trash icon.

5. Configure Policies

The Policy Manager is accessible from the main menu bar. The home page shows a list of tools supported by HDP Security Administration server. Clicking a particular repository name opens toward the Policy list for the repository.



5.1. Policy Overview

Policies limit access to Hive and HBase repositories to White Listing users, that is once a repository is created and the agent installed, only users who have been granted permission can access the resources. The Security Agent intercepts requests to the resource and checks the user against the policies of the repository and determines if the user matches any rules that grant them access to the resource.

If no rules explicitly grant access, the following occurs:

- **HDFS**: The request is passed through and the user can access the resource if permitted to do so by the HDFS local policies.
- **Hive and HBase** : The request is rejected.

5.2. Add a Policy

Policies define who can access which resources within a Repository. Policies can only be written for known Users and Groups, that is users and groups that have already been defined in the HDP Security Administration Web UI, either by the User and Groups Synchronizer or manually entered.

To add a Policy:

1. Click **Policy Manager** > **Repository Name** > **Add New Policy**.

   The Create Policy page displays.

2. Complete the Policy Details:

   Table 5.1. Policy Details

Field	Description
HDFS: **Resource Path** or Hive/HBase Tables and Columns	For HDFS, enter a comma separated list of paths for the policy. For example, `/apps/tez/qa,/apps/tez/`

Field	Description
	`production`. For Hive and HBase, start typing the table name and select the tables you want to add. In the path, you can use regular expression to match multiple directory (or table/column/column family names), for example, `/apps/tez/qa*` matches all subdirectories of `/apps/tez` that being with 'qa'.
Description	Enter text that describes the policy, only visible from the Policy Manager UI.
Recursive	Select Yes to grant permission to all subdirectories of the specified path.
Audit Logging	Select Yes to log activity to the directory to the Audit and Reporting facility of the HDP Security Administration tools.

3. Complete the User and Group Details:

Table 5.2. Policy Details

Field	Description
Group Permission	Click the + sign to select a group from the Users and Groups list. If the group is not listed, it must be added to the server that the User and Group Synchronizer polls for accounts. If the user or group was recently added, it will appear after the next *`sync_interval`*.
User Permission	Click the + sign to select a user from the Users and Groups list. If the user is not listed, it must be added to the server that the User and Group Synchronizer polls for accounts. If the user or group was recently added, it will appear after the next *`sync_interval`*.
Policy Status	Select Enabled to enforce the Policy, or Disabled to keep a copy of the Policy without enforcing it.

4. Click **Save**.

5.3. Remove a Policy

Removing a policy from the Web UI, removes the policy from both the HDP Security Administration server and the corresponding agent on the Repository host.

To remove a Policy:

1. Click **Policy Manager** > **Repository Name** .

   The Policy list displays.

2. Click the trash icon at the end of the row.

The policy change synchronizes within a few seconds with the agent and is removed from both the server and the agent.

5.4. Disable a Policy

Disabling a policy in the Web UI, removes the policy from the corresponding agent on the Repository host.

To remove a Policy:

1. Click **Policy Manager** > **Repository Name** .

   The Policy list displays.

2. Click the Edit icon near the end of the row.

3. Change the Policy Status to Disabled.

4. Click **Save**.

The policy change synchronizes within a few seconds with the agent and is removed from both the server and the agent.

5.5. Enable/Disable Audit Logging

You can disable only auditing (and leave the policy active). When auditing is disabled, repository activity is no longer recorded by the HDP Security Administration tools. Hadoop cluster logging still occurs and is available in the configuration locations.

To disable auditing:

1. Click **Policy Manager** > **Repository Name** .

   The Policy list displays.

2. Click the Edit icon near the end of the row.

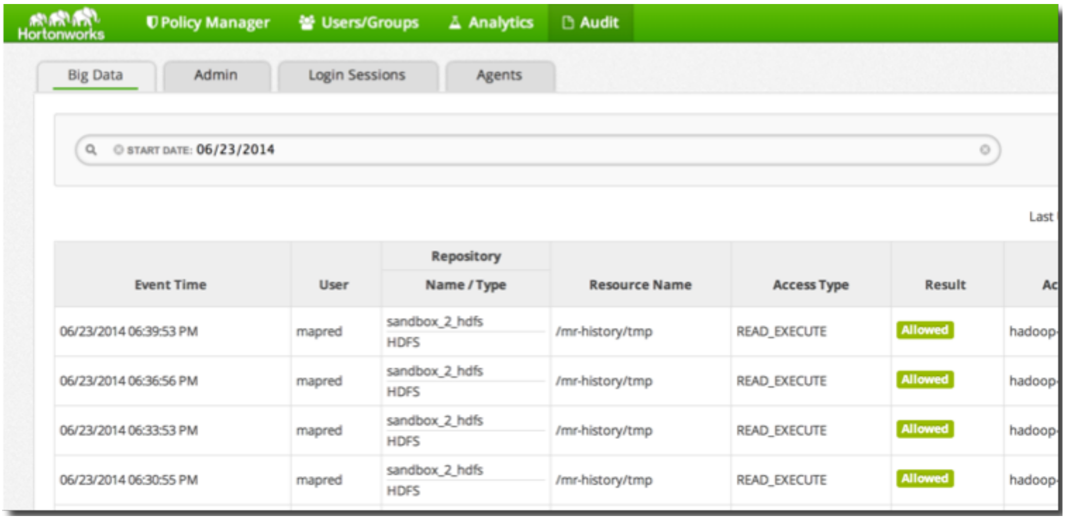
3. Change the Audit Logging to off.

4. Click **Save**.

The policy change synchronizes within a few seconds with the agent tops uploading activity data to the server.

6. Audit

The HDP Secure Administration tools provide audit logs for activity on the Hadoop cluster repositories as well as in the Administration Web UI.



START DATE: 06/23/2014

Last

		Repository				
Event Time	User	Name / Type	Resource Name	Access Type	Result	Ac
06/23/2014 06:39:53 PM	mapred	sandbox_2_hdfs HDFS	/mr-history/tmp	READ_EXECUTE	Allowed	hadoop
06/23/2014 06:36:56 PM	mapred	sandbox_2_hdfs HDFS	/mr-history/tmp	READ_EXECUTE	Allowed	hadoop
06/23/2014 06:33:53 PM	mapred	sandbox_2_hdfs HDFS	/mr-history/tmp	READ_EXECUTE	Allowed	hadoop
06/23/2014 06:30:55 PM	mapred	sandbox_2_hdfs HDFS	/mr-history/tmp	READ_EXECUTE	Allowed	hadoop

The activity data is separated into the following tabs:

- **Big Data**: Provides Repository activity data for all Policies that have Audit set to On. The default repository Policy is configured to log all user activity within the Repository. This default policy does not contain user and group access rules.
- **Admin**: Contains all events for the HDP Security Administration Web UI, including Repository, Policy Manager, Log in, etc.
- **Login Sessions**: Contain all log events to the configured Repositories.
- **Agents**: Shows the upload history of the Security Agents.

7. Configure SSL for Web UI and Server/ Agent Communications

Configuring SSL (HTTPS) for the HDP Security Administration Web UI and server/agent communication, requires a JKS formatted private key and CA X509 certificate for the HDP Security Administration Server host.

- Install and Configure SSL on HDP Security Administration Server
- Install and Configure SSL on Security Agents

7.1. Install and Configure SSL on HDP Security Administration Server

Hortonworks recommends configuring SSL after HDP Security Administration server and agents are fully configured and tested.

Note

These steps require a private key for HDP Security Administration server and a valid CA X509 Certificate in JKS format. For more details on obtaining a certificate, see http://tomcat.apache.org/tomcat-7.0-doc/ssl-howto.html#Certificates.

1. Log on to the HDP Security Administration server as root.

2. Install the certificate in the key store following the instructions outlined in http://tomcat.apache.org/tomcat-7.0-doc/ssl-howto.html#Certificates.

3. Edit the `/usr/lib/xapolicymgr/ews/xapolicymgr.properties` as follows:

   a. Comment out the following line to disable the HTTP service port:

   ```
   #http.service.port=6080
   ```

   b. Uncomment the following line to enable the HTTPS service port:

   ```
   https.service.port=6080
   ```

   ### Note

   Modify the port number as required.

4. Add the certificate key store information:

   ```
   https.attrib.keyAlias=$KeyAlias_From_JKS_file
   https.attrib.keystorePass=$KeyStore_Password_for_JKS_file
   https.attrib.keystoreFile=$Absolute_Path_JKS_file
   ```

5. Restart the HDP Security Administration service as follows:

```
service xapolicymgr stop
service xapolicymgr start
```

7.2. Install and Configure SSL on Security Agents

To set up the Security Agent to communicate using HTTPS, perform these steps on each system where an agent is installed.

- Set Up SSL on the HDFS Security Agent
- Set Up SSL on the Hive Security Agent
- Set Up SSL on the HBase Security Agent

7.2.1. Set Up SSL on the HDFS Security Agent

The Security Agent for HDFS is installed on the NameNode. Perform these steps on the NameNode host.

> **Note**
>
> These steps require a private key for the HDP Security Agent (for client SSL verification) and a valid CA X509 Certificate in JKS format.

1. Change the HDP Security Administration Server URL from HTTP to HTTPS in the Security Agent configuration file:

   a. Open the configuration file for editing, `/etc/hadoop/conf/xasecure-hdfs-security.xml`.

   b. Change the value in the xasecure.hdfs.policymgr.url property from *http* to *https* and update the port as required.

   For example, if the current value is http://*$hostname*:6080/service/assets/policyList/*$repository_name* change it to https://*$hostname*:6080/service/assets/policyList/*$repository_name*.

2. Define the SSL policymgr.clientssl properties in the Security Agent SSL configuration file, `/etc/hadoop/conf/xasecure-policymgr-ssl.xml` as follows:

   ```
   xasecure.policymgr.clientssl.keystore = $JKS_file
   xasecure.policymgr.clientssl.keystore.password = $keystore_password
   xasecure.policymgr.clientssl.truststore = $CA_certificate_file
   ```

3. After saving the configuration, restart the NameNode.

   a. On the NameNode host machine, execute the following command:

   ```
   su -l hdfs -c "/usr/lib/hadoop/sbin/hadoop-daemon.sh stop namenode"
   ```

   Ensure that the NameNode Service stops completely.

   b. On the NameNode host machine, execute the following command:

```
su -l hdfs -c "/usr/lib/hadoop/sbin/hadoop-daemon.sh start namenode"
```

Ensure that the NameNode Service starts correctly.

7.2.2. Set Up SSL on the Hive Security Agent

The Security Agent for Hive is installed on the HiveServer2 host. Perform these steps on the HiveServer2 host. If Security Agents are installed on HiveCli hosts, repeat these steps on each of the hosts.

Note

These steps require a private key for the HDP Security Agent (for client SSL verification) and a valid CA X509 Certificate in JKS format.

1. Change the HDP Security Administration Server URL from HTTP to HTTPS in the Security Agent configuration file:

   a. Open the configuration file for editing, `/etc/hive/conf.server/xasecure-hive-security.xml`.

   b. Change the value in the xasecure.hdfs.policymgr.url property from *http* to *https* and update the port as required.

      For example, if the current value is http://*`$hostname`*:6080/service/assets/policyList/*`$repository_name`* change it to https://*`$hostname`*:6080/service/assets/policyList/*`$repository_name`*.

2. Define the SSL policymgr.clientssl properties in the Security Agent SSL configuration file, `/etc/hive/conf.server/xasecure-policymgr-ssl.xml` as follows:

   ```
   xasecure.policymgr.clientssl.keystore = $JKS_file
   xasecure.policymgr.clientssl.keystore.password = $keystore_password
   xasecure.policymgr.clientssl.truststore = $CA_certificate_file
   ```

3. After saving the configuration, restart the Hive.

   a. Stop Hive. Execute this command on the Hive Metastore and Hive Server2 host machine.

      ```
      ps aux | awk '{print $1,$2}' | grep hive | awk '{print $2}' | xargs kill
       >/dev/null 2>&1
      ```

   b. Start Hive Metastore. On the Hive Metastore host machine, execute the following command:

      ```
      su - hive -c "env HADOOP_HOME=/usr JAVA_HOME=/usr/jdk64/jdk1.6.0_31 /tmp/
      startMetastore.sh /var/log/hive/hive.out /var/log/hive/hive.log /var/run/
      hive/hive.pid /etc/hive/conf.server"
      ```

      where, *`$HIVE_LOG_DIR`* is the directory where Hive server logs are stored. For example, `/var/logs/hive`.

   c. Start HiveServer2. On the Hive Server2 host machine, execute the following command:

```
su - hive -c "env JAVA_HOME=/usr/jdk64/jdk1.6.0_31 /tmp/startHiveserver2.
sh /var/log/hive/hive-server2.out /var/log/hive/hive-server2.log /var/
run/hive/hive-server.pid /etc/hive/conf.server"
```

where *$HIVE_LOG_DIR* is the directory where Hive server logs are stored. For example, `/var/logs/hive`.

7.2.3. Set Up SSL on the HBase Security Agents

The Security Agents for HBase repositories are installed on the HBase Master and all HBase Regional Servers in the cluster. Perform these steps on all the HBase Security Agent hosts.

Note

These steps require a private key for the HDP Security Agent (for client SSL verification) and a valid CA X509 Certificate in JKS format.

1. Change the HDP Security Administration Server URL from HTTP to HTTPS in the Security Agent configuration file:

   a. Open the configuration file for editing, `/etc/hbase/conf/xasecure-hbase-security.xml`.

   b. Change the value in the xasecure.hdfs.policymgr.url property from *http* to *https* and update the port as required.

      For example, if the current value is http://*$hostname*:6080/service/assets/policyList/*$repository_name* change it to https://*$hostname*:6080/service/assets/policyList/*$repository_name*.

2. Define the SSL policymgr.clientssl properties in the Security Agent SSL configuration file, `/etc/hbase/conf/xasecure-policymgr-ssl.xml` as follows:

   ```
   xasecure.policymgr.clientssl.keystore = $JKS_file
   xasecure.policymgr.clientssl.keystore.password = $keystore_password
   xasecure.policymgr.clientssl.truststore = $CA_certificate_file
   ```

3. After saving the configuration, restart the HBase services.

   a. Execute this command on the HBase Master host machine:

      ```
      su -l hbase -c "/usr/lib/hbase/bin/hbase-daemon.sh --config /etc/hbase/
      conf stop master; sleep 25"
      ```

   b. Execute this command on all RegionServers:

      ```
      su -l hbase -c "/usr/lib/hbase/bin/hbase-daemon.sh --config /etc/hbase/
      conf stop regionserver"
      ```

   c. Execute this command on the HBase Master host machine:

      ```
      su -l hbase -c "/usr/lib/hbase/bin/hbase-daemon.sh --config /etc/hbase/
      conf start master; sleep 25"
      ```

   d. Execute this command on all RegionServers:

```
su -l hbase -c "/usr/lib/hbase/bin/hbase-daemon.sh --config /etc/hbase/
conf start regionserver"
```

8. Troubleshoot Agent and Server Connections

Verify connectivity between the agent host and HDP Security Administration server.

8.1. Test HDP Security Administration Server URL

In order to connect to the HDP Security Administration server, both the HTTP (default 6080) and to uploads/pulls information using the repository named in the connection URL.

Use telnet to test the connection:

1. Check the URL for the portal in the agent configuration file, `xasecure-`*`$service-name`*`-security.xml`. This file is located in the configuration directory of the Hadoop service (`/etc/`*`$servicename`*`/conf`).

   For example:

   ```
   more /etc/hbase/conf/xasecure-hbase-security.xml
   ......
   <property>
      <name>xasecure.hbase.policymgr.url</name>
      <value>http://policymgr:6080/service/assets/policyList/sandbox_2_hbase</
   value>
      <description>
    Location where XASecure Role Based Authorization Info is
    located.
      </description>
   </property>
   .......
   ```

2. Telnet to the HTTP port from the agent host to the HDP Security Administration server:

   ```
   telnet policymgr 6080
   Trying 192.168.56.101...
   Connected to policymgr.
   Escape character is '^]'.
   ```

   ### Note

   If either of the connections fail, then check your firewall and SELinux settings.

3. If you are able to connect, verify the repository name matches on the server and agent:

   a. Sign in to the HDP Security Administration Web UI.

   b. Click **Policy Manager** > **Manage Repository**. The name of the repository in the UI must match the name shown at the end of the `xasecure.hbase.policymgr.url`.

      For example, this agent sends information to the HBase repository named `sandbox_2_hbase`:

```
<value>http://policymgr:6080/service/assets/policyList/sandbox_2_hbase</
value>
```

8.2. Test Remote Connection to MySQL

The Security Agents connect directly to the HDP Security Administration database. Using the connection information you provided in the `install.properties` file, manually verify that the mysql database is accessible:

1. Log into the agent host as `root`.

2. Open the install.properties file and find the connection information, for example:

   ```
   XAAUDIT.DB.HOSTNAME=poliymgr
   XAAUDIT.DB.DATABASE_NAME=xasecure
   XAAUDIT.DB.USER_NAME=xasecure
   XAAUDIT.DB.PASSWORD=hadoop
   ```

3. Telnet to the MySQL port from the agent host to the HDP Security Administration host:

   ```
   telnet policymgr 3306
   Trying 192.168.56.101...
   Connected to policymgr.
   Escape character is '^]'.
   ```

4. Using the same information, connect to the MySQL database from the command line:

   ```
   mysql -u$XAAUDIT.DB.USER_NAME -p$XAAUDIT.DB.PASSWORD -h$XAAUDIT.DB.
   HOSTNAME $XAAUDIT.DB.DATABASE_NAME
   ```

   For example:

   ```
   mysql -uxasecure -phadoop -hpolicymgr xasecure
   Reading table information for completion of table and column names
   You can turn off this feature to get a quicker startup with -A

   Welcome to the MySQL monitor.  Commands end with ; or \g.
   Your MySQL connection id is 4069
   Server version: 5.1.73 Source distribution

   Copyright (c) 2000, 2013, Oracle and/or its affiliates. All rights reserved.

   Oracle is a registered trademark of Oracle Corporation and/or its
   affiliates. Other names may be trademarks of their respective
   owners.

   Type 'help;' or '\h' for help. Type '\c' to clear the current input
    statement.
   ```

If the connection is rejected, verify MySQL user name and password as well as the permissions for the user to connect remotely.

8.3. Uninstall Security Agent

The same basic un-install steps apply to all Security Agents.

To un-install an XAAgent:

1. Login to the host as the `root` user.
2. Go to the `/etc/xasecure` directory for the type of agent you are uninstalling, that is either `hdfs`, `hive` or `hbase`:

   ```
   cd /etc/xasecure/$type
   ```

3. Run the uninstaller script:

   ```
   ./uninstall.sh
   ```

   The agent is removed from the system.

If the `/etc/xasecure` for the agent type does not exist, the agent can also be uninstalled using the installation package.

Use the following steps to uninstall Security Agent:

1. Copy the install tar file to a temporary directory on the host (for example, `/tmp/xasecure`).
2. Expand the tar file in to the temporary directory and go to that directory:

   ```
   cd /tmp/xasecure
   tar xvf xasecure$name-$buildversion.tar
   ```

3. Run the un-install script in as follows:

   ```
   ./uninstall.sh
   ```