# Cloudera Security

## **Authorization Using Sentry**

# **Enabling the Sentry Service Using Cloudera Manager**

## Prerequisites

CDH 5.1.x (or higher) managed by Cloudera Manager 5.1.x (or higher).

HiveServer2 and the Hive Metastore running with strong authentication.

1. For HiveServer2, strong authentication is either Kerberos or LDAP.
2. For the Hive Metastore, only Kerberos is considered strong authentication

Note:

If you want to override the Kerberos prerequisite for the Hive metastore, set the ***sentry.hive.testing.mode***property to ***true***to allow Sentry to work with weaker authentication mechanisms. Add the following property to the HiveServer2 and Hive metastore's sentry-site.xml

**Warning:** Cloudera strongly recommends against enabling this property in production. Use Sentry's testing mode only in test environments.

1. Impala 1.4.0 (or higher) running with strong authentication. With Impala, either Kerberos or LDAP can be configured to achieve strong authentication.
2. Implement Kerberos authentication on your cluster.
3. The Hive warehouse directory (/user/hive/warehouse or any path you specify as *hive.metastore.warehouse.dir*in your hive-site.xml) must be owned by the Hive user and group.
4. **Using the default Hive warehouse directory** - Permissions on the warehouse directory must be set as follows

For example:

$ sudo -u hdfs hdfs dfs -chmod -R 771 /user/hive/warehouse

$ sudo -u hdfs hdfs dfs -chown -R hive:hive /user/hive/warehouse

## Enabling the Sentry Service

### Disable impersonation for HiveServer2 in the Cloudera Manager Admin Console:

1. Go to the Hive service.
2. Click the **Configuration** tab
3. Select **Scope** > **HiveServer2**.
4. Select **Category** > **Main**.
5. Uncheck the **HiveServer2 Enable Impersonation** checkbox.
6. Click **Save Changes** to commit the changes.

### Enable the Hive user to submit YARN jobs.

1. Go to the YARN service.
2. Click the **Configuration** tab.
3. Select **Scope** > **NodeManager**.
4. Select **Category** > **Security**.
5. Ensure the **Allowed System Users** property includes the *hive* user. If not, add *hive*.
6. Click **Save Changes** to commit the changes.

### Block the Hive CLI user from accessing the Hive metastore:

1. Select the **Hive** service.
2. On the Hive service page, click the **Configuration** tab.
3. In the search well on the right half of the Configuration page, search for Hive Metastore Access Control and Proxy User Groups Override to locate the *hadoop.proxyuser.hive.groups* parameter and click the plus sign.
4. Enter *hive* into the text box and click the plus sign again.
5. Enter *hue* into the text box.
6. **Uncheck the Enable Sentry Authorization using Policy Files for Hive and Impala**
7. Click **Save Changes**.

## Enabling the Sentry Service for Hive

1. Go to the Hive service.
2. Click the **Configuration** tab.
3. Select **Scope** > **Hive (Service-Wide)**.
4. Select **Category** > **Main**.
5. Locate the **Sentry Service** property and select Sentry.
6. Click **Save Changes** to commit the changes.
7. Restart the Hive service.

## Enabling the Sentry Service for Impala

1. Go to the Impala service.
2. Click the **Configuration** tab.
3. Select **Scope** > **Impala (Service-Wide)**.
4. Select **Category** > **Main**.
5. Locate the **Sentry Service** property and select Sentry.
6. Click **Save Changes** to commit the changes.
7. Restart Impala.

## Enabling the Sentry Service for Solr

1. Go to the Solr service.
2. Click the **Configuration** tab.
3. Select **Scope** > **Solr (Service-Wide)**.
4. Select **Category** > **Main**.
5. Locate the **Sentry Service** property and select Sentry.
6. Click **Save Changes** to commit the changes.
7. Restart Solr.

## Enabling the Sentry Service for Hue

* Hue uses a Security app to make it easier to interact with Sentry.
* When you set up Hue to manage Sentry permissions, make sure that users and groups are set up correctly.
* **Every Hue user connecting to Sentry must have an equivalent OS-level user account on all hosts so that Sentry can authenticate Hue users.**
* **Each OS-level user should also be part of an OS-level group with the same name as the corresponding user's group in Hue.**

1. Go to the Hue service.
2. Click the **Configuration** tab.
3. Select **Scope** > **Hue (Service-Wide)**.
4. Select **Category** > **Main**.
5. Locate the **Sentry Service** property and select Sentry.
6. Click **Save Changes** to commit the changes.
7. Restart Hue.

* For more information on using the Security app, see [THIS](http://gethue.com/apache-sentry-made-easy-with-the-new-hue-security-app/).

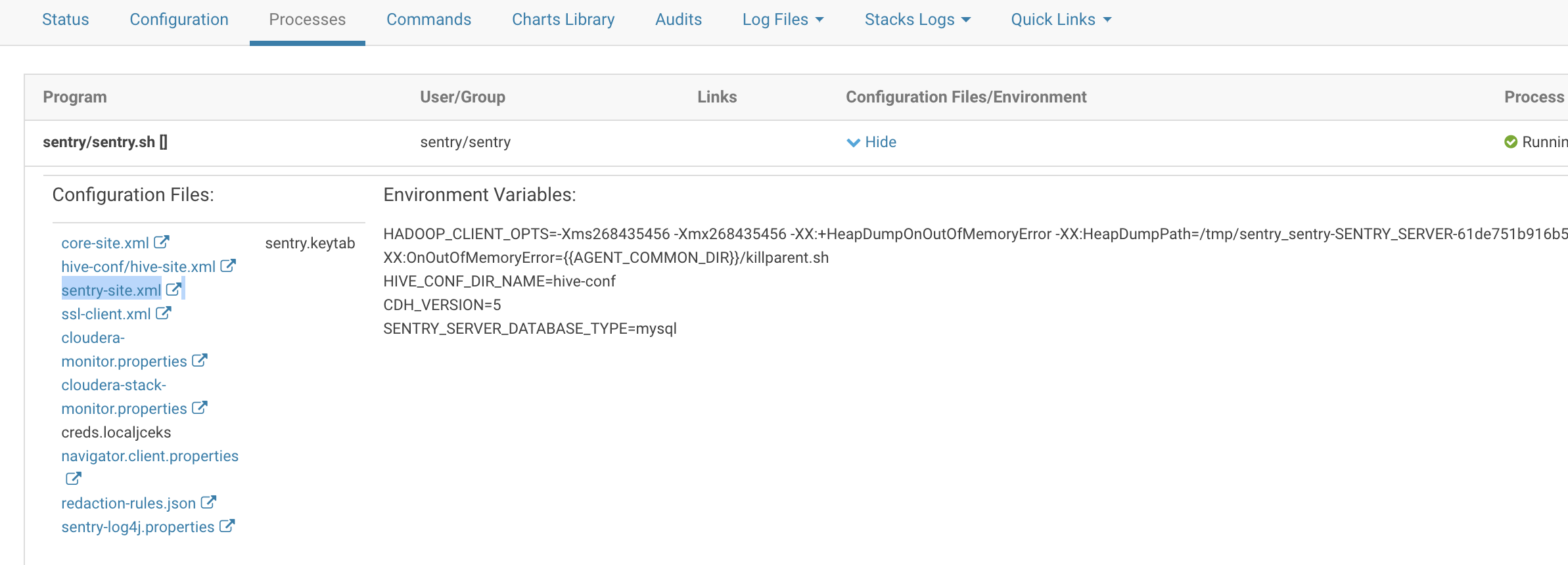
## Add Hive, Impala and Hue Groups to Sentry's Admin Groups

1. Go to the Sentry service.
2. Click the **Configuration** tab.
3. Select **Scope** > **Sentry (Service-Wide)**.
4. Select **Category** > **Main**.
5. Locate the **Admin Groups** property and add the hive, impala and hue groups to the list. If an end user is in one of these admin groups, that user has administrative privileges on the Sentry Server.
6. Add hive, impala and hue users in **“Allowed Connecting Users”** property
7. Click **Save Changes** to commit the changes.

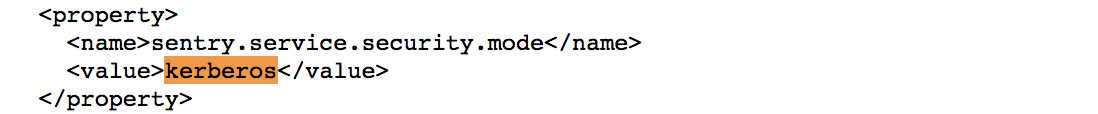
## Verify Security mode is set to Kerberos:

Goto Sentry > Instance > Click “Sentry Server” > Processes >>

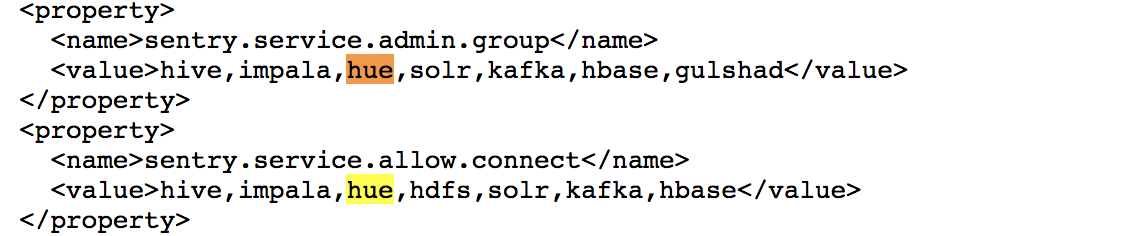
You will get below screen, Open ***sentry-site.xml*** and verify “*sentry.service.security.mode*” is set to Kerberos.



Ex:



Below properties should contain hue, hive groups (these are admin groups)



## **On Teminal, generate tgt with hive user keytab and principal**

kinit -kt hive.keytab hive/<hostname>@REALM

### Login to Beeline

beeline

!connect jdbc:hive2://cdh2.c.velvety-glazing-195106.internal:10000/;principal=hive/cdh2.c.velvety-glazing-195106.internal@MIT.DABSTERINC.COM.COM

### Created the role admin.

create role myadmin;

### Assign the role to a group.

GRANT ROLE myadmin TO GROUP hadoopadmins;

GRANT ROLE myadmin TO GROUP hadoopdev;

After these steps all users within the group administrators are allowed to manage hive privileges

### Granted privileges to admin role.

GRANT ALL ON SERVER server1 TO ROLE myadmin WITH GRANT OPTION;

### Create a sample database;

Create database testDB;

Show databases;

### Grant all privileges to admin

GRANT ALL on database testDB to role myadmin;

### Create a sample table, try accessing it before and after granting different privilege’s

Use testDB;

CREATE External TABLE employee(eid int, name string) ROW FORMATE DELIMITED FIELDS TERMINATED BY ‘\t’ LINES TERMINATED BY ‘\n’ STORED AS TEXTFILE;

### Commands in Sentry

Similarly, you can use below syntax to play around with Sentry:

CREATE ROLE [role\_name];

DROP ROLE [role\_name];

GRANT ROLE role\_name [, role\_name]

TO GROUP <groupName> [,GROUP <groupName>]

REVOKE ROLE role\_name [, role\_name]

FROM GROUP <groupName> [,GROUP <groupName>]

GRANT SELECT(column\_name) ON TABLE table\_name TO ROLE role\_name;

REVOKE SELECT(column\_name) ON TABLE table\_name FROM ROLE role\_name;

GRANT SELECT ON DATABASE jranalyst1 TO ROLE analyst\_role;

SHOW CURRENT ROLES;